Issues in Fast 3D Reconstruction from Video Sequences

MARCOS A. RODRIGUES, ALAN ROBINSON and WILLIE BRINK
Geometric Modelling and Pattern Recognition Group
Sheffield Hallam University, Sheffield UK, www.shu.ac.uk/gmpr

Abstract: In this paper, we discuss methods for incorporating data acquired as 3D surface scans of human faces into applications such as 3D animation and biometric 3D facial recognition. In both applications the challenge is to accurately and consistently find predefined features such as the corners of the eyes and the tip of the nose. In the field of biometry, if 3D face recognition is to compete with 2D methods, these features must be found to an accuracy greater than 1:1000. In multimedia, the greatest problem occurs with animated 3D faces, where very small inaccuracies are clearly seen in moving faces. Therefore any inconsistencies must be found and rectified. Our work starts by providing a high-speed, accurate 3D model, and then developing methods to recognise the required features.

Key–Words: 3D reconstruction, 3D modelling, 3D measurement, 3D face recognition, 3D animation

1 Introduction

As techniques for recording 3D surfaces improve, two important challenges emerge: how to immerse the 3D model into the environment of the application, and how to achieve continuity when the 3D model is recorded over the fourth dimension of time. These challenges are faced variously in fields such as biometric face recognition, industrial inspection and reverse engineering, and media applications.

The problem of immersion occurs because the initial model acquired by the 3D recording device typically has no information about the nature of the modelled surface; it may be a human face, a building or a hairline crack in an engine part. The application, on the other hand, will need to recognise features on the surface and then map those features to corresponding points in the reference model. For instance, for 3D face recognition the eyes, nose and other features such as front and chicks must be located, and for immersive gaming the player‘s head, hands, and feet must map to a synthetic figure within the game environment. This problem can be tackled by attaching easily recognisable optical markers to relevant parts of the subject, as in motion capture systems, or by employing software algorithms to find the required features from the initial data model. Clearly the former method is simpler, but its intrusiveness is prohibitive in many applications such as biometric security. The latter method of finding features from the surface model can be a difficult task.

Problems with continuity are likely to occur when the 3D model is recorded over time. This is partly because errors in the surface model are often not consistent, and while they may be unnoticeable in a single frame of the recording, they will jump dramatically from frame to frame in 3D. For similar reasons, post-processing work such as hole-filling must be carefully designed to provide frame continuity. Therefore for our core activities of face recognition and immersive media, we are working towards solutions to the problems of mapping features to the reference model, and improving the consistency and continuity of the model over time.

The paper is structured as follows. In Section 2 we introduce our multiple stripe method for fast 3D reconstruction. In Section 3 we present one of our core applications which is feature extraction for 3D face recognition and model rectification through hole filling and smoothing. In Section 4 we describe the frame continuity problem and an effective approach to deal with it focusing on hole filling. Finally, in Section 5 we summarise and discuss our work in the context of 3D face recognition and media applications.

2 Fast 3D Reconstruction

Our existing research into 3D scanning uses a novel uncoded structured light method [6], which projects a pattern of evenly-spaced white stripes onto the subject, and records the deformation of the stripes in a video camera placed in a fixed geometric relationship to the stripe projector. A camera and projector configuration is depicted in Fig 1.

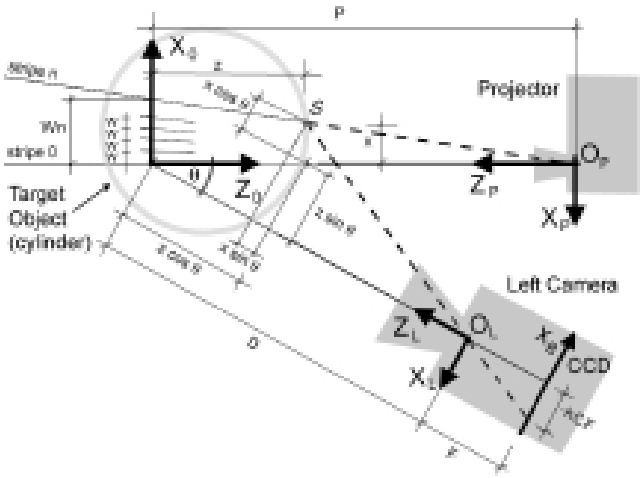

Fig 1: The scanner viewed down the Y axes. 3D reconstruction is achieved by corresponding stripe indices in both image and projector spaces.

A detail from a video frame is depicted in Fig 2 (top) clearly showing the deformed stripes. One problem that our research has successfully tacked is referred to as the *indexing problem* which is to find the corresponding stripe indices in both image and projector spaces. Even for continuous surfaces the problem can be severe as small discontinuities in the object can give rise to un-resolvable ambiguities in 3D reconstruction. When there are large discontinuities over the object as shown in Fig 2 (bottom, points a, b and c belong to the same stripe) and these are distributed at many places the problem is particularly severe.

Despite such difficulties, the advantage of this over stereo vision methods [4] is that the stripe pattern provides an explicitly connected mesh of vertices, so that the polyhedral surface can be rendered without the need for surface reconstruction algorithms. Also, a smoothly undulating and featureless surface can be more easily measured by structured light than by stereo vision methods. These advantages for single frame scanning are even more important for 4D applications such as animation and immersive game playing.

Once the surface shape has been modelled as a polygonal mesh, we return to the video image, take the colour of the reflected white stripe at each pixel that maps to a vertex, and colour the vertex (or triangle) accordingly. The final model therefore contains the (*x, y, z*) coordinates and their corresponding RGB (red, green, blue) values for each vertex, and the face can be visualised as in Fig 3. This shows the original bitmapped image (the stripes are too fine to be discernible in the left picture), and five arbitrary poses of the 3D colour-mapped 3D model.

For 4D media applications we record the face over time using a video recording. If the data are compressed as normally happens in an MPEG-2 recording, or if the video is recorded onto tape causing a loss of resolution and increased signal to noise ratio, then the bitmap image is degraded to the point where it may become unusable. In the work reported here we record from a Canon XM1 Firewire video camera into the capture software (iMovie) on an Apple computer as a sequence of .bmp files (24 bit, 768x576 pixels). Frames can then be processed one at a time.

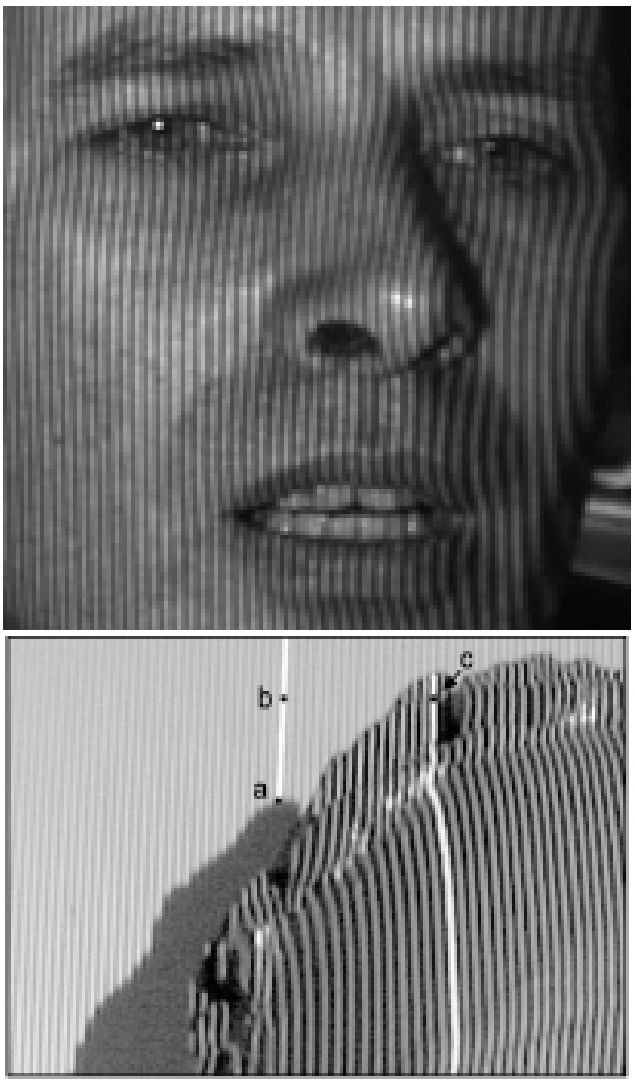

Fig 2: Top, detail from a bitmap, showing the stripes deforming across the face. Bottom, the indexing problem due to large discontinuities.

Fig 3: Visualisation of 3D data.

3 3D Recognition

Here we focus on the problem of human face recognition, as the theoretical and practical issues relate to robust 3D face recognition can be translated to a number of other applications. Much research has been undertaken in the area of 2D face recognition [5], [1] while 3D is incipient. It is often said that 3D face recognition has the potential for greater accuracy than 2D techniques, as 3D is invariant to pose and illumination and incorporates important face descriptors embedded within the 3D features [1]. The challenges to improved 3D face recognition for real-time applications reflect the shortcomings of current methods:

1. the need for fast and accurate 3D sensor technology,
2. improved algorithms to take into consideration variations in size and facial expression, and
3. improved methodology and datasets allowing algorithms to be tested on large databases, thus removing bias from comparative analyses.

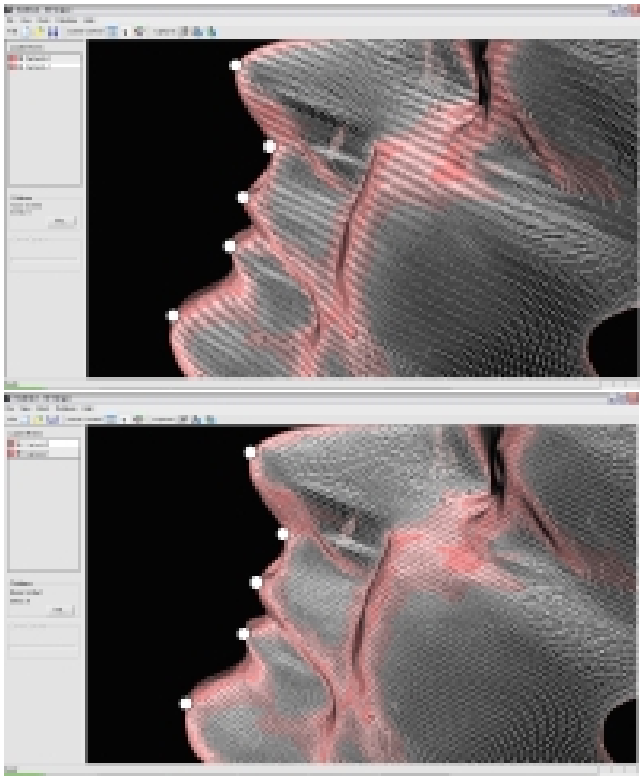

Fig 4: Top, a model with 33K vertices and 65K polygons with only white stripes being processed (stripes are clearly visible in the model). The markers represent estimated feature points on the 3D surface. Bottom: doubling the mesh density improves the accuracy of feature measurement.

Our fast 3D scan method meets challenges 1 and 3 above. The technique allows a 3D model to be acquired in 40 milliseconds with high accuracy. Fig 4 top shows a model acquired by processing white stripes only. By processing the black stripes, the number of vertices and polygons are doubled as depicted at the bottom. We have also developed subdivision algorithms based on polynomial interpolation and the density of the mesh can then be adjusted to any desired value.

Many approaches to 3D face recognition are based on 2.5D (e.g. [3] and references therein) and most try to compare scans through registration such as ICP (iterative closest point estimation) and their variants [7]. Performing recognition by comparing 3D scans through registration in this way becomes impractical for many reasons. First, there are too many data points leading to exponential time consuming algorithms and this can only work if one is to search a relatively small data base. Second, there is a practical constraint on registration, as it works best when models are acquired with scanning devices with the same characteristics. There is also an issue of defining what is a match in terms of global error between two surfaces and, perhaps equally important, which exactly are the data points being used to define a match.

This leads us naturally to feature point extraction as the most likely solution to 3D recognition. The problem of 3D recognition can thus be stated as:

1. Define a set of stable measures $m_i (i = 1, 2, \ldots, n)$ on a 3D scan and build a vector $M = (m_1, m_2, \ldots, m_n)^T$ that uniquely characterises a given vector
2. Build a matrix Ω of vectors M where the index of M points to the identity of the scanned object: $\Omega = (M_1, M_2, \ldots, M_s)^T$ where s is the total number of scans in the database
3. Define a method to identify a given scanned vector M with the most similar vector in the database (e.g. Principal Components Analysis).

In biometric 3D face recognition, the exact position of the eyes is essential. Because of the presence of eye lashes, the region is prone to unwanted noise (e.g. Fig 5 top). The solution involves first creating elliptical holes on the eyes then filing it with a spherical surface. The major and minor axes of the elliptical hole are defined proportional to the distance between the eyes, which is available from a 2D eye detection procedure. Once a 2D position is available, its 3D counterpart is easily determined. Holes are filled in by solving the Laplace differential equations over the hole then smoothed out using a Gaussian smooth filter as indicated in Fig 5.

Once the location of the eyes are corrected, the face pose is normalised and the highest point is determined as the tip of the nose. The two eye positions and

the tip of the nose define the vertical symmetry plane and all other points on the face are defined in relation to this plane. We have defined a total of 84 distinct measures on the face (e.g. eyes, front, nose tip, tip of lip, cheeks, and so on). These points are estimated automatically and some that belong to the vertical symmetry plane are indicated in Fig 4. A number of extra measurements can be made between such features such as distances and ratios in addition to area, volume, perimeter, and various types of diameters such as breath and length.

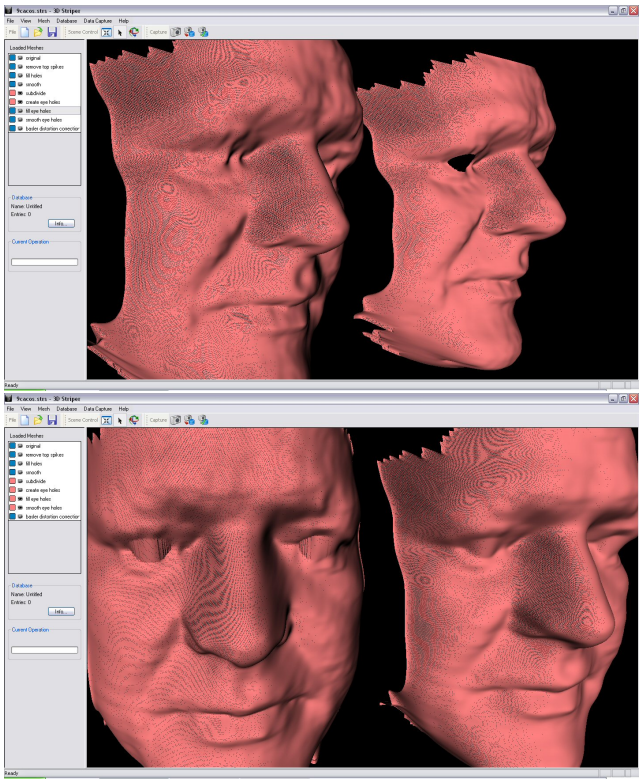

Fig 5: Rectifying the eye region: first create holes on the eyes (top right) then fill in using a Laplace hole filling algorithm (bottom left) and smooth out using a convolution average filter or Gaussian smooth algorithm (bottom right).

4 The Frame Continuity Problem

Here we will restrict our discussion to the problems associated with incorporating the 3D recording of a human face into a graphics environment. In the previous section we have discussed methods for identifying salient features in the human face. It is then a well-researched task to map those features onto a reference model by interpolating the positions of the vertices, and this gives appealling results, especially if an anti-aliased texture map is included. Unfortunately, we have noticed that when this is done over time to produce a moving face, many frame continuity errors occur, in the texture map and the surface shape. These problems are added to the fact that frame continuity problems will also occur in the initial recording, before it has been remapped. Solutions are inevitably heuristic, and we outline a number of specific issues here:

- Finding ill-defined features. The nose is a typical problem as the possible locations for the tip of the nose cover a wide area. In our animation test we perch a pair of glasses on the nose of the face model, and over time the glasses will jump around if the nose is inconsistently defined. Defining a fast, accurate and robust procedure for finding the tip of the nose is becoming an important goal in both biometric and media applications.
- Hole-filling. Standard techniques use splines and curvature methods, but these methods must be redefined in 4-space to avoid jitters and wobbles of the filled hole over time. For instance, Fig 7 shows three models from consecutive frames with clear evidence of inconsistencies around the nose, mouth and eyes that, despite most of the surface being well defined, must be solved to give an appealling result.
- Dealing with boundaries. A serious problem allied to hole-filling occurs at the edges of the initial recorded surfaces that are not defined as holes to be filled, such as around the mouth and nostrils. If the subject is speaking there will be an inconsistent delineation of the inside of the lips, as shown in Fig 6 source image (top) and 3D model (bottom). While using make-up helps to define the required edges, the ideal solution is to use software algorithms to recognise the lips.

Hole filling methods [9], [10], [11] or stripe connecting methods in our context, range from simply connecting the edges of the holes with straight lines and planes, to using curves such as cubic splines and Beziers, to our "hole-patching method" which replaces the hole with the patch on the symmetrically opposite side of the face. These techniques can work well for a rigid surface, but when the surface is moving the problem becomes much harder, because the interpolation of missing surface may be inconsistent from surface to surface. For single frames, i.e. rigid surfaces, we currently use a simple straight line interpolation, which if the controls are set correctly will produce satisfactory results. So the first step here is to try the same method over a number of frames.

Fig 6 shows the principles involved, for two successive frames shown left to right. At the top are de-

tails of the bitmap image showing the mouth slightly opening from the left frame to the right frame. The bitmap is similar to that seen in Fig 2, with the white stripes as recognised by the algorithms emphasised in the image. In this case the algorithm stops when it reaches the lips, but in an uneven manner due to the occlusions presented by the lips.

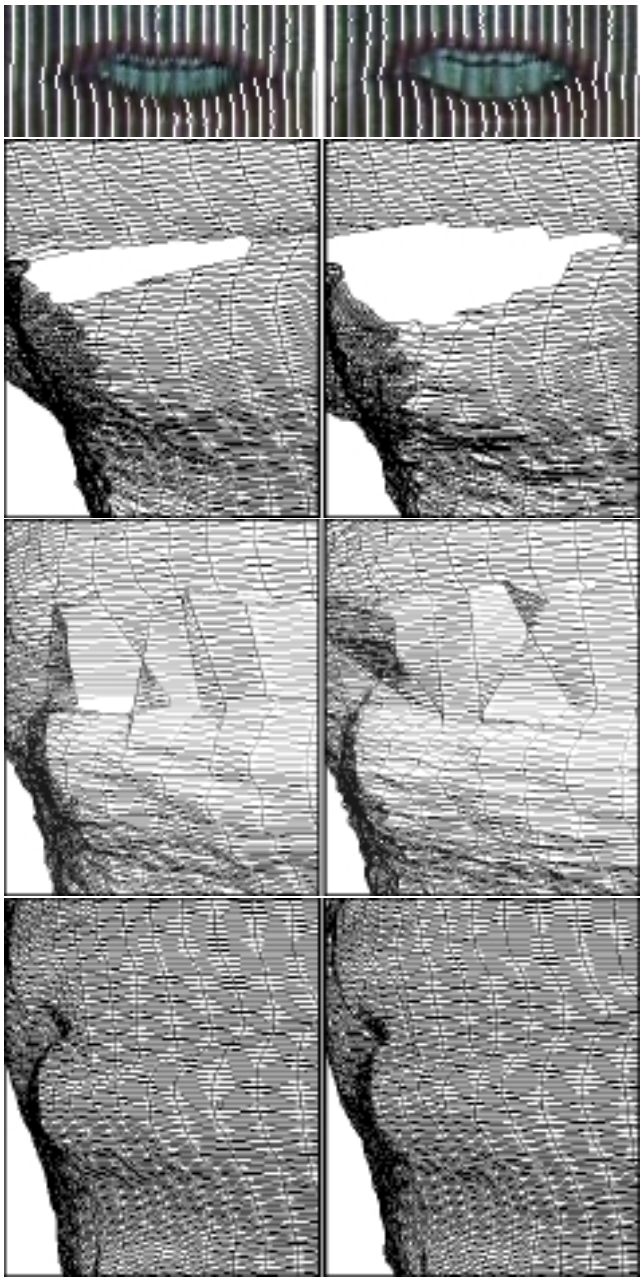

Fig 6: Model of two successive frames, showing detail of lips. Top is the original source image, with recognised stripes emphasised. Below is the resulting 3D model, in the middle with automatic straight line interpolation across the mouth and at the bottom with interpolated hole filling.

The generated meshes are derived via the mapping between each pixel on the white stripe and its corresponding vertex in the mesh. It is clear that the uneven lip boundary is also present in the mesh, and over the course of the animated sequence this produces unattractive jitters. The problem here is that the viewer is very critical of temporal changes which do not flow smoothly, and will be disturbed by surface edges which jitter from frame to frame. One solution is to fill in the hole in a way which smoothly follows from frame to frame. The holes in the mesh correspond exactly to the missing parts of the stripes, so that if those stripes can be connected in some way, then the homeomorphism will allow the vertices to be similarly connected in the mesh (note that this does not include projector occlusions). This is a simpler and potentially more accurate alternative to surface reconstruction from the polygonal mesh, which is the more usual approach, and requires a 3D interpolation rather than the 2D interpolation which can be used here.

Fig 6 middle shows a simple automatic straight line filler added. This is a worst case example showing lines which almost meet, and in general an inconsistent spacing between the stripes. The problem occurs partly because the edge of the lips, which is the point at which the straight line should join with its continuation on the other side of the mouth, is irregular, and partly because there the teeth provide a specular surface which reflects the stripes in unpredictable ways. It has been found that even if the stripes are prevented from irregular changes of direction as shown here, if the spacing between stripes is not smoothly continuous, the results will be unacceptable due to interframe jitters.

Therefore our solution is to introduce some interpolation between frames, and to keep the spacing between the stripes as even as possible. The key issue here is that this solution is only relevant for the mouth, where a straight line between the lips gives satisfactory results; it is not appropriate for, say the side of the nose. This means that we must segment the face, approximately as shown by the rectangles in the top of Fig 6 to constrain the hole filling region. The results of this interpolated line filling can be seen at the bottom of Fig 6 which shows a much smoother and more consistent filling, which provides smooth results between frames. This will also allow a better colour mapping to be achieved, with the proviso that problems may occur if the interpolation maps to the wrong colour.

As indicated earlier, this interpolation method is not suitable for the common occurrence of an occlusion at one side of the nose, as shown in Fig 7. Instead

we assume that the other side of the nose will not be occluded, and find a symmetrical patch of shape and texture which can be reflected and used. This method uses techniques from the face recognition methods, and again requires careful control to avoid frame jitters.

Fig 7: Successive frames with occlusions at the side of the nose and inconsistent boundaries at the lips.

5 Conclusions

This paper has discussed methods for incorporating data acquired as 3D surface scans of human faces into applications such as biometry and multimedia. We start by introducing our current method of fast 3D acquisition using multiple stripes which allows 3D reconstruction from a single 2D video frame. This lends the technique suitable for capturing moving objects such as a moving face in multimedia applications.

In both biometric and multimedia applications the challenge is to accurately and consistently find predefined features such as the corners of the eyes and the tip of the nose. In multimedia, the greatest problem occurs with animated 3D faces, where very small inaccuracies are clearly seen in moving faces. We define this as the frame continuity problem and present our solution to effectively deal with the problem. The method involves finding ill-defined features, applying hole filling techniques that minimise frame continuity problems and finally dealing with boundaries.

In the field of biometry, if 3D face recognition is to compete with 2D methods, facial features must be found to an accuracy greater than 1:1000. We discuss the problem of reliably defining the eyes in 3D as these are probably the most important biometric features for face recognition. The presence of eye lashes creates a noisy region around the eyes and these impair measurements and detection of other features that depend on knowing the exact position of the eyes. We present an affective solution of creating elliptical holes around the eyes, then filling the holes using either Laplace hole filling or bilinear interpolation, then smoothing over using a Gaussian smoothing algorithm.

References:

[1] K. Bowyer, K. Chang, and P. Flynn. A survey of 3d and multi-modal 3d+2d face recognition. *ICPR 2004*, pages 324–327, Cambridge 2004.

[2] M. Dong and R. Kotharib. Feature subset selection using a new definition of classifiability. *Pattern Recognition Letters*, 24:1215–1225, 2003.

[3] X. Lu, A. Jain, and D. Colbry. Matching 2.5d face scans to 3d models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):31–43, 2006.

[4] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London*, B:301–328, 1979.

[5] T. Nagamine, T. Uemura, and I. Masuda. 3d facial image analysis for human identification. *ICPR 1992*, pages 324–327, the Netherlands 1992.

[6] A. Robinson, L. Alboul, and M. Rodrigues. Methods for indexing stripes in uncoded structured light scanning systems. *Journal of WSCG*, 12(3):371–378, February 2004.

[7] M. Rodrigues, R. Fisher, and Y. Liu. Registration and fusion of range images. *CVIU Computer Vision and Image Understanding*, 87(1-3):1–131, July 2002.

[8] M. Rodrigues, A. Robinson, L. Alboul, and W. Brink. 3d modelling and recognition. *WSEAS Transactions on Information Science and Applications*, 3(11):2118–2122, 2006.

[9] L. S. Tekumalla and E. Cohen. A hole filling algorithm for triangular meshes. *tech. rep. University of Utah*, December 2004.

[10] J. Wang and M. M. Oliveira. A hole filling strategy for reconstruction of smooth surfaces in range images. *XVI Brazilian Symposium on Computer Graphics and Image Processing*, pages 11–18, October 2003.

[11] J. Wang and M. M. Oliveira. Filling holes on locally smooth surfaces reconstructed from point clouds. *Image and Vision Computing*, 25(1):103–113, January 2007.