Fast 3D recognition for forensics and counter-terrorism applications

RODRIGUES, Marcos <http://orcid.org/0000-0002-6083-1303> and ROBINSON, Alan

Fast 3D Recognition for Forensics and Counter-Terrorism Applications

Marcos A Rodrigues and Alan Robinson, Sheffield Hallam University, Sheffield, UK

Abstract

The development of advanced techniques for fast 3D reconstruction and recognition of human faces in unconstrained scenarios can significantly help the fight against crime and terrorism. We describe a 3D solution developed within Sheffield Hallam University that satisfies a number of important requirements such as operating close to real-time, high accuracy in recognition rates, and robust to local illumination. Experimental results in 3D face recognition are reported and two scenarios are provided that can be used to exploit the outcomes of this research for forensic analysis and for flagging potential threats in counter-terrorism.

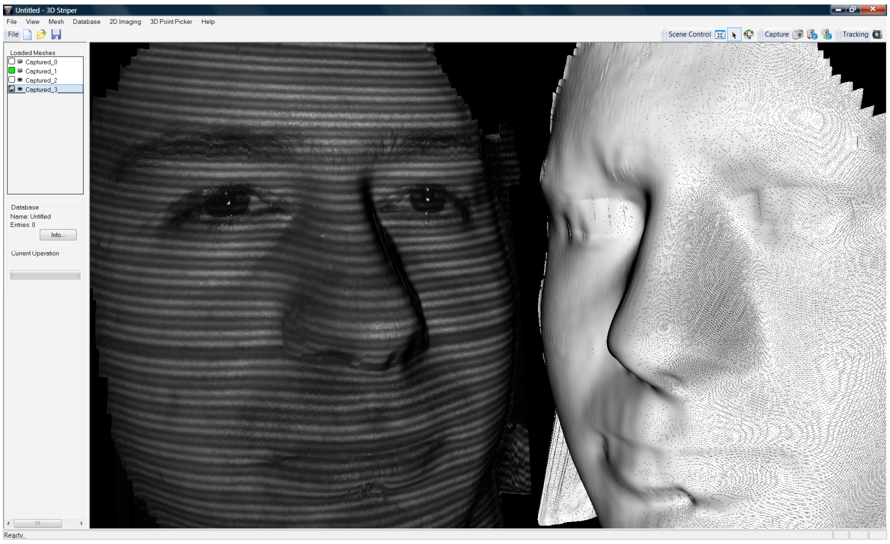
Introduction

Some essential security requirements to societies across the world include the need to manage heterogeneous sources of data and perform advanced recognition and reasoning techniques in real-time to detect complex abnormal behaviour and potential security threats. At Sheffield Hallam University we have focused on 3D modeling and recognition techniques together with associated hardware development that can be applied in a real-time security scenario. The proposed concept of 3D CCTV is forward looking into technological needs for increased recognition rates. It lends itself to the integration of existing 2D databases and standard CCTV data with unique generation and manipulation of 3D footage. It has the potential to greatly improve the effectiveness of CCTV gathering, as it will make it possible to recover 3D information from video sequences in a novel and effective way.

We have been developing and patenting unique methods of acquiring 3D models using uncoded or lightly-coded structured light scanning – each model being recorded within one single video frame (40ms). The structured light principle is well known: project a pattern of light onto the target surface, and record the deformed pattern in a camera with a known spatial relationship to the projector. We have developed structured light systems-for modelling faces and many other surfaces in industrial and medical applications [1—7].

The advantages of structured light scanning over stereo vision methods are numerous for instance: 1) the computationally intensive Correspondence Problem is avoided as only one image is used and the problem is then shifted towards the less computational intensive stripe indexing problem, 2) there is an explicit connected graph that makes surface reconstruction straightforward, 3) the density of the data can be controlled in the pattern design, and 4) smoothly undulating, featureless surfaces can be easily measured and this is not the case with stereo vision. The key disadvantage of structured light methods is that as the distance between projector, target and camera become greater, the light intensity reaching the camera becomes weaker. This means that it may be difficult to discriminate a dense light pattern, especially if a colour-coding scheme is used.

In order to realize non-intrusive 3D CCTV we developed techniques to project patterns of stripes in the NIR (near infrared) spectrum, which are invisible to the naked eye. The technique involves placing a NIR projector and two cameras in a known geometric relationship. One camera operates in the visible spectrum collecting standard CCTV footage, while the other camera operates in NIR providing means to recover the 3D structure of any desired frame or sequence of frames. Both visible and NIR spectrum cameras continuously save to disk. The difference between the two sets of data is that the visible spectrum camera contains normal texture of objects in the world, while the NIR camera images are illuminated by a NIR pattern of stripes as shown in Fig. 1.



Name: Untitled
Entries: 0

Fig. 1 Left, an example 3D model with NIR stripes on, right, wire mesh 3D model.

The way the 3D CCTV concept operates on human faces is highlighted as follows. A set of face detection and eye tracking routines have been implemented [7] which operate on the visible spectrum camera. As soon as a face is detected that satisfies predefined conditions (width and height larger than a minimum number of pixels) the projector stripes are switched on and an NIR image is taken. Both images are saved to disk, the NIR contains stripe information that allows 3D reconstruction in real time (i.e. within a video frame of 40ms) and the visible spectrum image contains texture information that can be overlaid onto the 3D model.

Steps in 3D Face Reconstruction and Recognition

Concerning general research in 3D face recognition, the availability of 3D models and the format in which they are presented are not convenient for research aiming at fast recognition rates. While the Face Recognition

Grand Challenge FRGC [8] has allowed the wider research community to test recognition algorithms from standard 2D and 3D databases, a severe limitation is that it was not designed to cater for real-time requirements. The FRGC database is standardized such that an application can load pre-formatted data for feature extraction and recognition. 3D data were reconstructed from human subjects taken from a frontal, but arbitrary view point and, given that these are large files containing the structure of the vertices in 3D, this rules out the possibility of testing algorithms in a real-time scenario. Therefore, while 3D data were profitably used to test recognition algorithms in the FRGC, the process does not represent a natural way in which 3D facial recognition systems are to be deployed. The 3D CCTV concept described here provides a contribution towards solving real-time issues in 3D face recognition.

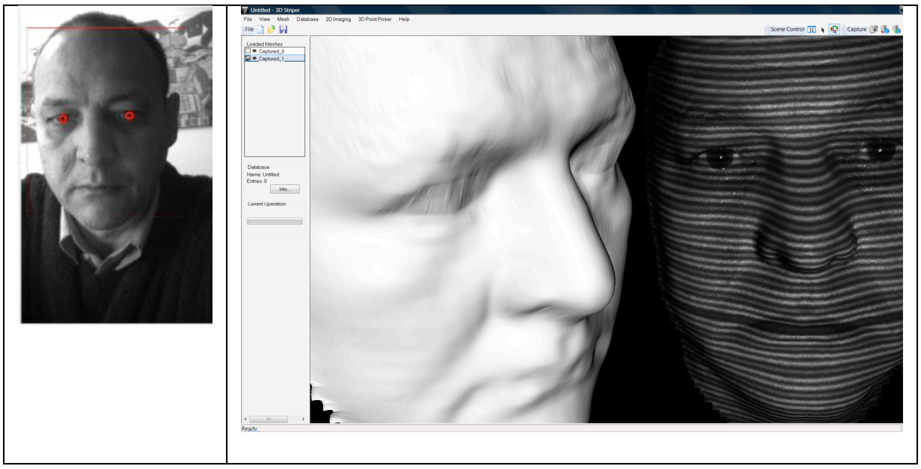
There are prescribed steps that need to be performed in order to achieve fully automatic 3D face recognition based on vision systems:

- 2D tracking and filtering: face and eye tracking; image filtering; image correspondence (stereo) or projection pattern detection (structured light methods)
- 3D reconstruction and post-processing: generation of 3D point cloud and mesh triangulation; noise removal; 3D hole filling; mesh smoothing (optional); mesh subdivision (optional); pose normalization; feature extraction
- Enrolment and recognition: features are enrolled in a database for subsequent identification (one-to-many) or verification (one-to-one) recognition using appropriate algorithms

These steps are described in the following sections.

2D Tracking and Filtering

We use OpenCV face and eye tracking routines developed by Intel's Micro-computer Research Lab [9], which proved to be consistent and reliable provided that a number of constraints are specified. The general problem with such detection techniques is the number of false positives. For instance, on any image there could be various detected faces and some might not be real faces. The same problem happens with eye detection; the routines normally detect more eyes than there are in the scene. In order to solve this problem a number of constraints are defined: first, there should be only one face detected in the image and the face width must be larger than a certain threshold (300 pixels in our case); second, there should be only one left and only one right eye detected in the image, and these must be within the region of interest set by the face detection; third, the position of the face and eyes must not have moved more than a set threshold since last detection (10 pixels in our case) so to avoid inconsistent shots caused by rapid motion.



Name: Untitled
Entries: 0

Fig. 2 Left, eye tracking in 2D using the visible spectrum camera. Right, a striped NIR image is taken and converted into 3D.

Fig 2 shows the visible spectrum camera continuously tracking and detecting (possibly multiple) faces and eyes, but only when the above conditions are satisfied a shot is taken. In this way, the system is apparently idle until someone places their face in front of the camera. When face and eyes are positively identified a near-infrared line pattern is projected onto the subject and a shot is taken and reconstructed in 3D as shown in the right of Fig. 2.

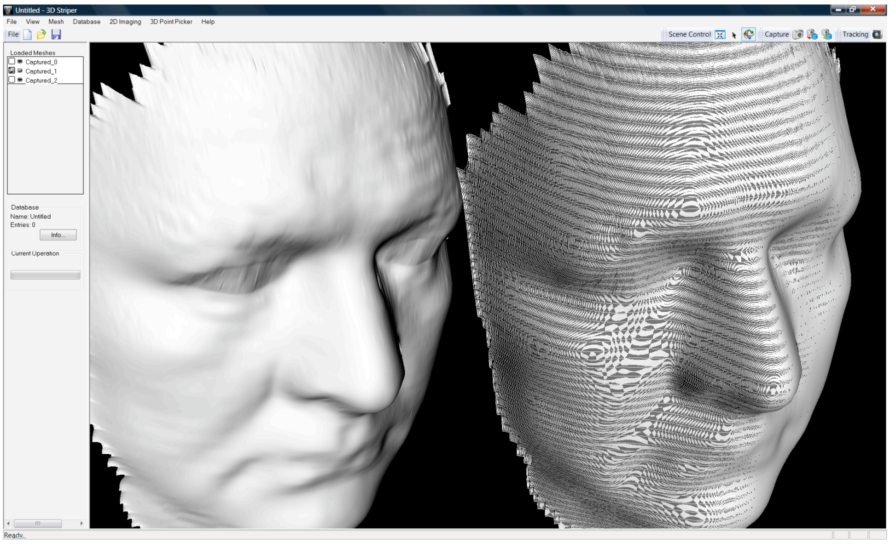
The next step in the process is to apply 2D image filters on the image that contains the stripe patterns namely a median filter followed by a weighted mean filter. This enables the detection of the stripe patterns in the image. Given that we know the geometry of the camera and projector, by knowing the stripe indices we can now fully reconstruct in 3D by trigonometry. While 3D models are shown in Figures 1 and 2, details of the process have been published in [10].

3D Reconstruction and Post-Processing

3D reconstruction is achieved by mapping the image space to system space (camera + projector) in a Cartesian coordinate system. We have developed a number of successful algorithms to deal with the mapping as described in [2, 10]. Once this mapping is achieved, a 3D point cloud is calculated and the output is triangulated using the connectivity of the vertices.

Once the surface shape has been modeled as a polygonal mesh, a number of 3D post-processing operations are required: hole filling, mesh subdivision, smoothing, and noise removal. There are several techniques that can be used to fill in gaps in the mesh such as the ones discussed in [11, 12, 13]. From the techniques we considered, we tend to focus on three methods namely bilinear interpolation, Laplace, and polynomial interpolation. We found that the most efficient method for real-time operation is bilinear interpolation [14].

The next step is mesh subdivision that, depending on the recognition algorithm to be used may or may not be required. Our research indicates that mesh subdivision is strongly advisable. The reason is that we sample the mesh for recognition where the boundaries of the data are set based on vertex positions so increasing the density of the mesh will provide more accurate sample boundaries. We use a polynomial interpolation of degree 4 across the stripe patterns and this increases the mesh density while making features more discernible. This is demonstrated in Figure 3, where in the subdivided mesh the region around the eyes and lips are better delineated.



Untitled - 3D Striper

Loaded Meshes

☐ Captured_0

☑ Captured_1

☐ Captured_2

Database

Name: Untitled

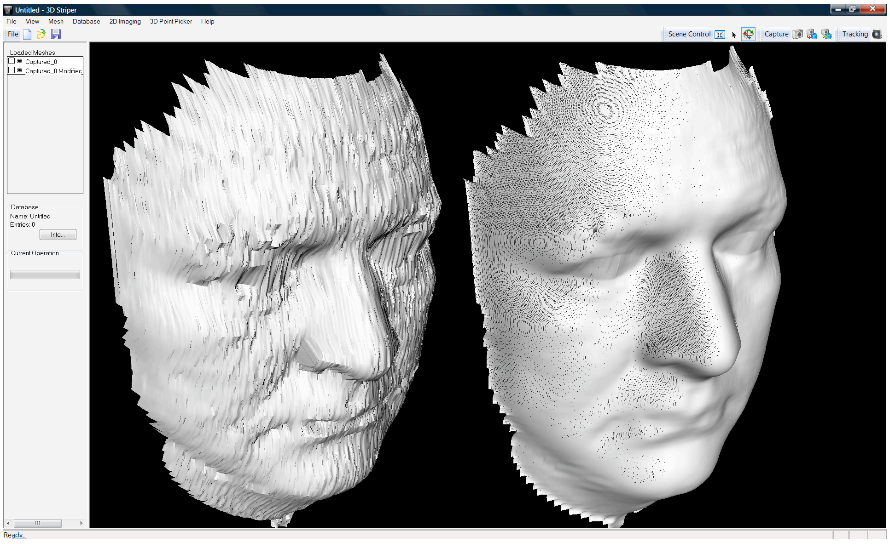
Entries: 0

Current Operation

Ready.

Fig. 3 Post-processing showing a sub-divided mesh (left) and non-subdivided (right)

We use two smoothing techniques namely moving average and Gaussian smoothing. Moving average is performed through cycling across and then along the stripes. The average of every three points is estimated and the middle point is replaced by the average. In this way the boundary vertices remain anchored. Gaussian smoothing is iteratively estimated by considering 8 neighbours for each vertex. A convolution mask of size 3x3 is applied and the centre vertex is perturbed depending on the values of its neighbours. The difference between each neighbour and the centre vertex is recorded and averaged as ΔV. A multiplier factor is provided (L) which determines how much of this average should be used to correct the value of the centre vertex; i.e., it defines the momentum of the error correction. We use L=0.9 and the number of iterations is set to maximum 35. In each iteration cycle i, the centre vertex V is corrected by $V_i = V_{i-1} + L\Delta V$. The effects of smoothing are depicted in Figure 4. It could be argued that smoothing has the effect to remove features from the model, such as the area around the lips is more pronounced in the non-smoothed model. However, non-smooth meshes can have severe spikes that impair recognition. From our experiments, we conclude that the best sequence for smoothing operations are Gaussian smoothing followed by a moving average.



□ Captured_0
□ Captured_0 Modifiec

Database
Name: Untitled
Entries: 0

Fig. 4 The effects of smoothing.

Noise removal is mainly concerned with the region around the eyes, as considerable amount of noise exist due to eyelashes and unwanted reflections as it can be seen from the raw unsmoothed model on the left on Fig 4. A natural solution would be to replace the vertices in the eye by a spherical surface centred somewhere behind the face model. By experimentation, we chose the centre of the sphere at a position 40mm behind the face model in the same Z-axis as the centre of each eye. An elliptical mask is marked centred on each eye, and all vertices within the elliptical surface have their values replaced by their spherical counterparts. This however, resulted in unnatural looking models. A second solution, which is conceptually simpler, is to punch an elliptical hole centred at each eye and then fill in the holes with bilinear interpolation. This has proved to work well for the face models and it is the solution that is adopted here.

Pose Normalization and Sampling

All 3D models need to be brought to a standard pose to allow consistent recognition. We have chosen a standard pose depicted in Fig 5 where the origin is placed at the tip of the nose. In this pose, the X-axis is aligned with the horizontal position of the eyes, the Y-axis forms a constant angle of π/10 with a point located on the front between the two eyes, and the Z-axis points away from the model such that all depths are negative.

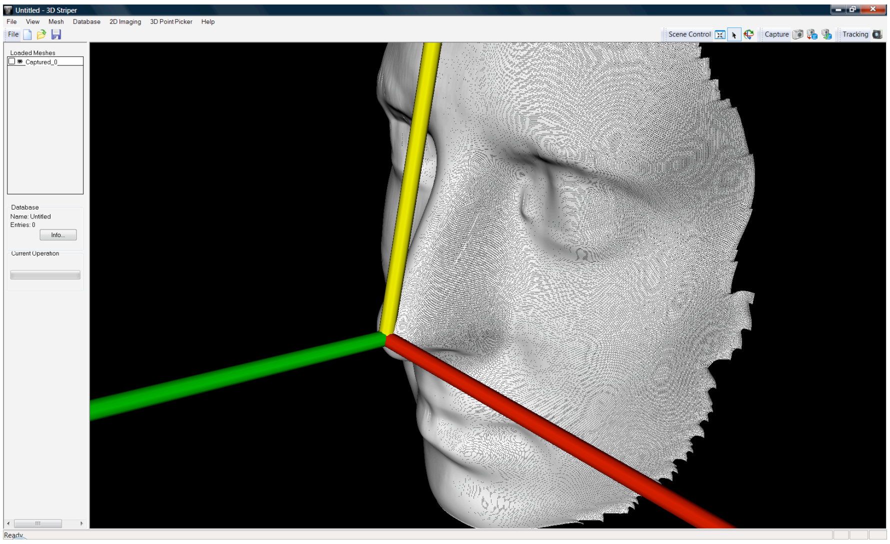


Fig. 5. The standard pose with the origin at the tip of the nose

The algorithm to achieve this standard pose is described as follows [1] (given that we know the position of the eyes (E_1 and E_2) in 3D:

1. Estimate the angle ß$_1$ in the XY-plane between E_1 and E_2
2. Centered on E_1 rotate the mesh around the Z-axis by angle ß$_1$: Rot(z, ß$_1$)
3. Estimate the angle ß$_2$ in the YZ-plane between E_1 and E_2
4. Centered on E_1 rotate the mesh around the Y-axis by angle ß$_2$: Rot(y, ß$_2$)
5. Find the intersection point on the mesh (above the eyes, on the front) of the arcs centered on E_1 and E_2 with radius 0.75 of the inter-ocular distance. Mark this point as F
6. Find the tip of the nose. This is defined as the highest point on the mesh below eye positions within a search threshold of one inter-ocular distance. Mark this point as T
7. Estimate the angle ß$_3$ described by F, T, and the Y-axis

8. Centered on *T*, rotate the mesh around the X-axis by (π /10 — ß$_3$): Rot(*x*, π /10 — ß$_3$)
9. After this rotation, the highest point on the mesh defining *T* might have slightly changed. Repeat steps 6, 7 and 8 until (π /10 — ß$_3$) is below a set threshold.

900 points are sampled for recognition defined within the boundaries of four planes: two horizontal planes (top and bottom) and two vertical planes (left and right) set the limits for sampling the 3D structure. All calculations are performed in 3D and the sampled points form the feature vector that uniquely characterizes a face and it is used for recognition. Figure 6 shows a face structure together with the sampled points.

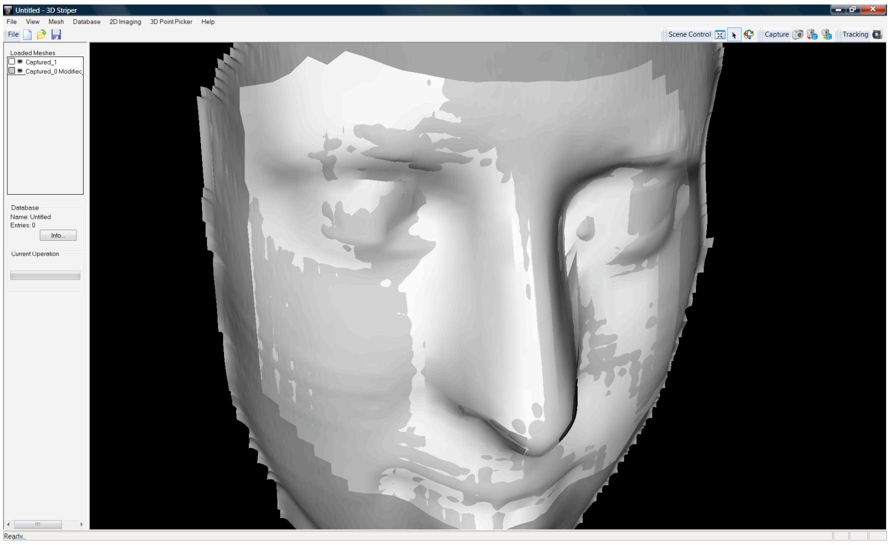


Fig. 6. Sampled data points from the face are marked lighter.

The delimiting planes for sampling (top, bottom, left, right) are defined as:

- $\prod_{\text{Top}}$: parallel to XZ-plane at point (0, 1.3(E_2 — E_1), 0)
- $\prod_{\text{Bottom}}$: parallel to XZ-plane at point (0, -0.66(E_2 — E_1), 0)
- $\prod_{\text{Left}}$: parallel to YZ-plane at point (-(E_2 — E_1), 0, 0)
- $\prod_{\text{Right}}$: parallel to YZ-plane at point ((E_2 — E_1), 0, 0)

Enrolment and Recognition

3D recognition algorithms based on eigenvector decomposition [15] were tested on a 3D database where 300 models were used as follows. First 100 subjects were enrolled in the database. Then a second (unseen) model of each person was used for testing identification and set the threshold for verification allowing the lowest possible FAR (false acceptance rate). Figure 7 shows an example of output recognition: the image on the left is the unknown face and the image on the right is the closest in the database found by the algorithm. A distance measure of 32 is indicated, and these measures are used to estimate the correct threshold for minimum FAR.

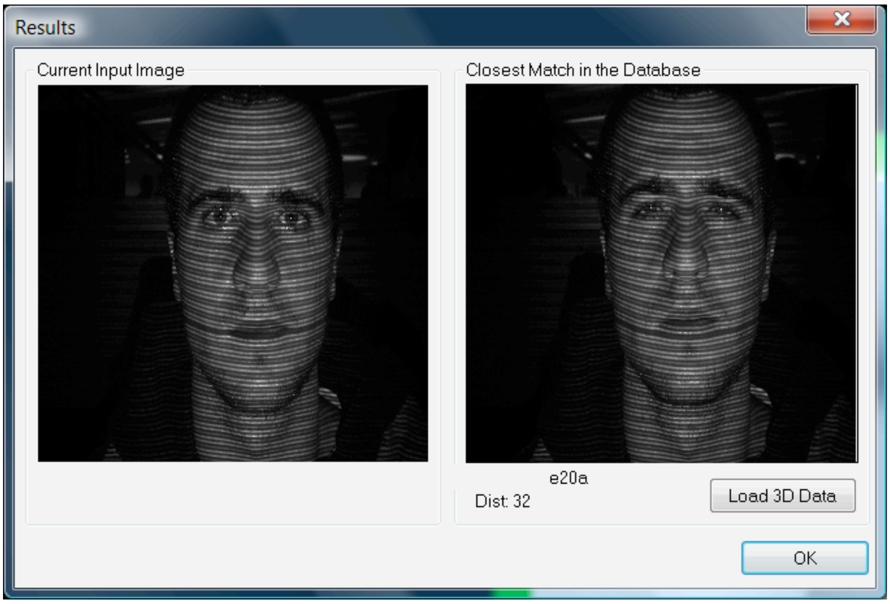
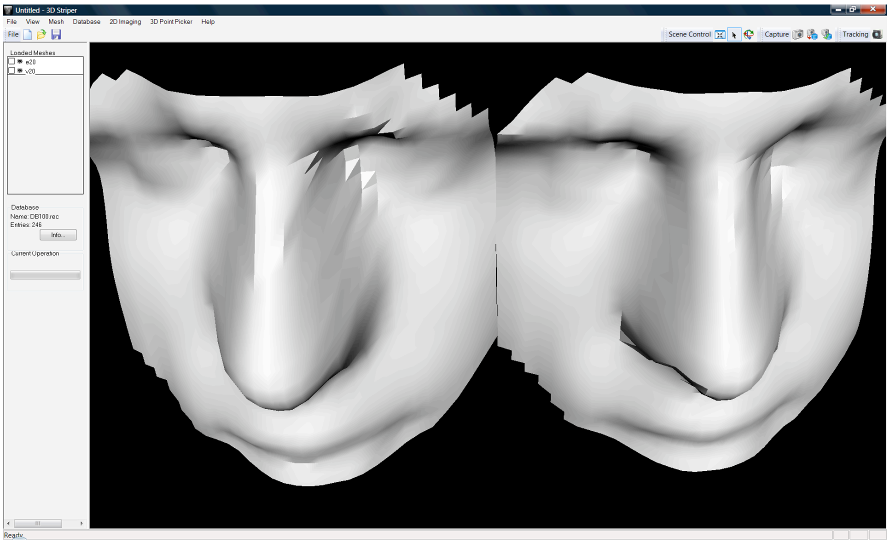


Fig. 7. Example of recognition showing the closest match and a distance measure

Figure 8 is the 3D counterpart to Figure 7: here the 3D models generated and used by the recognition algorithm are shown. The model on the right was used for enrolment and the model on the left is the unknown model.

The recognition results for the 100 unseen models was100%, that is, the algorithm always retrieved the correct identity for the closest match. Upon studying the distance measures obtained by the one-to-many identification it was noted that no distance was greater than 50. In order to set the correct threshold for minimum FAR, we then tested the database with models that were not enrolled in the database. It is expected that the closest match to these models (there is always a closest match) would have a distance larger than 50. In fact, this was verified and the system has proved to work with no FAR if the threshold was set to 50.



Database
Name: DB100.rec
Entries: 246

Fig. 8. The 3D models generated and used by the algorithm corresponding to the images shown in Figure 7.

We tested the algorithms in a simulated environment to verify its real-time performance from eye tracking to 3D reconstruction and recognition and logging the results with time stamps onto an HTML file. We used a Sony Vaio computer, Intel Core 2 Duo, 2.4GHz, 4GB memory. It has been shown that the methods presented here lend themselves to real-time operation as, from the moment the eye tracking algorithms lock on the eyes, it takes 1second 200millisencods to perform the following operations:

- Take a shot in 2D
- Run image filters (median and weighted mean filters)
- Detect the stripes in the image
- Convert the stripes into a 3D point cloud
- Triangulate the point cloud

- Perform hole filling on the mesh
- Determine the location of eyes in 3D
- Punch a hole in the eyes, fill with bilinear interpolation
- Find tip of the nose
- Normalize pose with origin at the tip of the nose
- Determine the sampling planes and sample the mesh
- Replace noisy points by reflection
- Search the database for the closest match
- Display results of recognition on screen
- Save to log file: time stamp, current image, closest match
- Continue eye tracking and repeat the sequence

Application Scenarios to Forensics and Counter-Terrorism

Having demonstrated the efficacy of the approach in terms of robust reconstruction and recognition and its real-time credentials, we turn our attention to possible applications to forensics and counter-terrorism – in addition to the most obvious application of identity verification for access control.

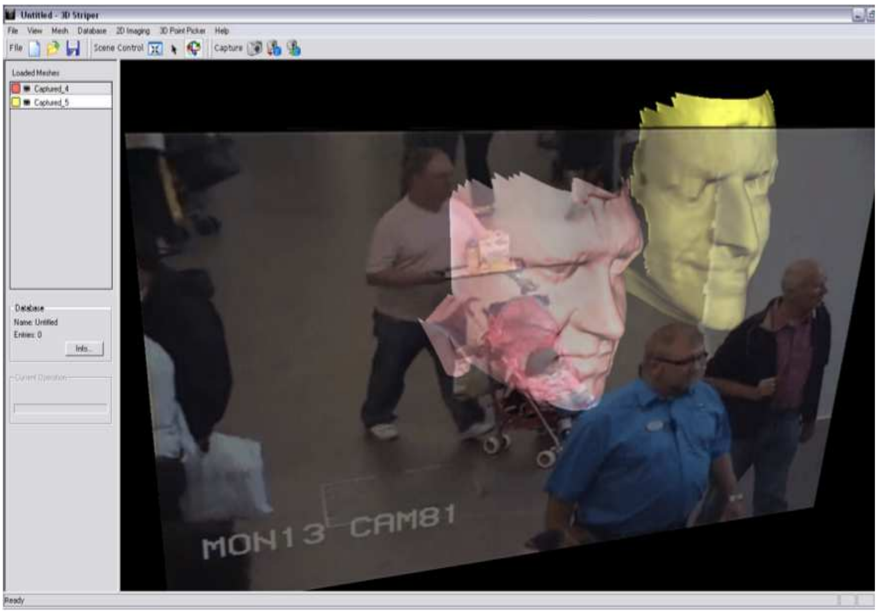
3D-2D Identification from Standard 2D Facial Databases

The idea in 3D-2D recognition is that accuracy can be improved over 2D-2D recognition by using a 3D subject against a standard 2D database. The comparison can take place either in 3D space, where the 2D data are reconstructed in 3D, or in 2D space, where 3D data are projected in 2D. This cross- dimensionality comparison is a classic computer vision problem, but advantage can be taken of the restricted nature of the application (human faces only). The 3D CCTV concept as proposed here allows 3D facial models to be reconstructed from saved footage: their position and orientation of the 3D facial models can then be automatically manipulated to match existing 2D photographs of suspects in a database at a higher accuracy than a 2D-2D comparison for challenging datasets in terms of pose and lighting. Incomplete facial models (say, a face seen from only one side) can be pose-normalized and, through mesh reflection, the unseen side of the face can be reconstructed together with the reflected texture.

3D-2D Identification from Standard 2D CCTV Video Sequences

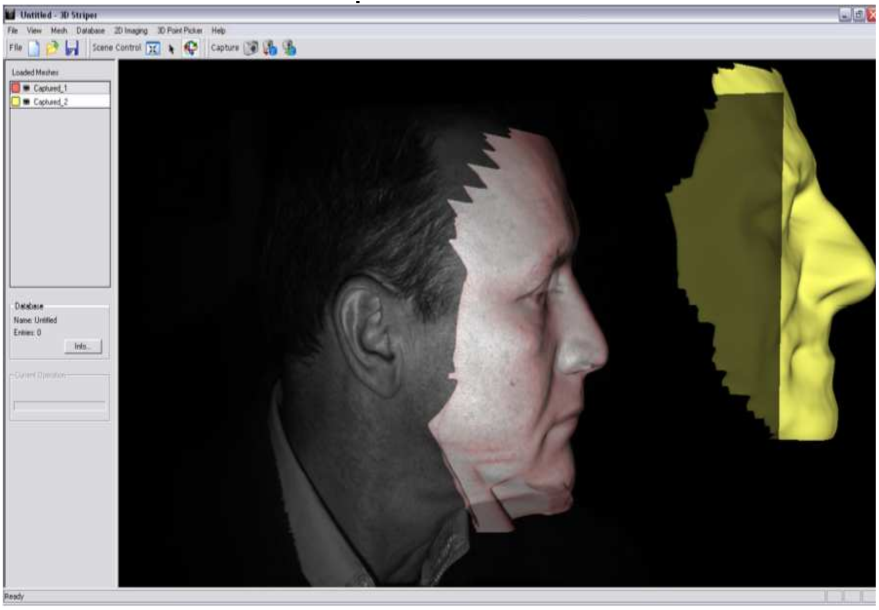
The purpose is to compare standard 2D CCTV with 3D models acquired either from 3D CCTV as discussed above or 3D acquisitions from a person in custody. The method proposed here can enable substantial improvements in detection rate of CCTV footage. The Metropolitan Police in London states that the identification rate from CCTV is about 20% (personal communication). The 3D CCTV concept has the potential to substantially increase identification rates through the integration of 2D and 3D data. This can be achieved by projecting 2D video footage onto a plane in a 3D environment and by providing the means to manipulate 3D models over the projective plane. In this way for instance, 2D profiles can be compared to 3D model profiles.

The simulated pictures below illustrate the concept where Figure 9 shows a 2D CCTV standard footage to be compared with 3D models by fitting the 3D models by a profile view in 2D. Figures 10 and 11 elaborate on this concept by displaying a 2D image onto the projective plane. 3D facial models of suspects are overlaid on the same screen. Through a combination of rotation, translation, and zooming, a 3D profile can be fitted onto the 2D footage. In the example of Figure 10, the model fits the person's profile. In figure 11, by using the "wrong" 3D model, coloured yellow (lighter), no amount of transformation can fit the profile. Such methods can be used off-line and can be used both for positive identification and for elimination purposes.



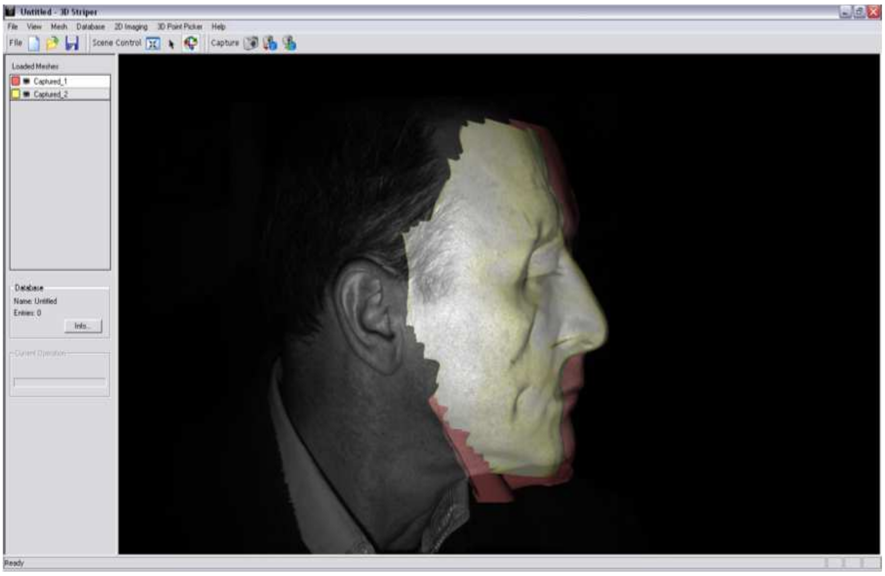
Database
Name: Untitled
Entries: 0

Fig. 9. Standard CCTV footage is projected on a 3D screen allowing the comparison of profiles



Database
Name: Untitled
Entries: 0

Fig. 10. The 3D model fits the 2D profile obtained from standard CCTTV



Name: Untitled

Entries: 0

Fig. 11. No amount of transformation can fit a wrong 3D model (lighter) onto the 2D profile

3D-3D Identification flagging in continuous mode in counter-terrorism applications

The main idea behind this application is to flag previously seen faces by continuously generating new models, enrolling into the database, and performing identification. A mobile robot platform can be used or a stationary 3D camera positioned at a strategic location in public places such as shopping malls and airports. The system is set into a fully automatic mode tracking faces and eyes, performing identification and enrollment. When a face is found in the database within a given threshold, the system will immediately flag this. It can be used to check the movements of suspected agents in a theatre of war or within a civilian environment in a shopping mall.

Conclusion

We have presented methods for real-time 3D face recognition from face and eye tracking in 2D to fast 3D reconstruction, feature extraction, identification and verification. Based on geometry alone, the reported recognition accuracy is a perfect 100% with zero FAR. We have used 300 distinct models in our experiments. Equally significant, we have shown that the process from 2D tracking to 3D recognition takes only 1second 200milliseconds per subject and thus, can be used in a real-time scenario given the speed and accuracy of the 3D recognition.

While the methods presented here have direct application in face recognition tasks for access control, we have discussed the possible application into a number of interesting domains such as integrating the proposed 3D CCTV concept with standard 2D CCTV in person identification scenarios such as forensic investigations. We also discussed the issue of operating in continuous identification and flagging mode for applications in counter-terrorism and intelligence.

Future work among others includes the design and implementation of mesh compression methods and algorithms enabling fast network-based 3D recognition systems.

References

[1] M.A. Rodrigues, A. Robinson, Real-Time 3D Face Recognition using Line Projection and Mesh Sampling. Eurographics Workshop on 3D Object Retrieval (2011) H. Laga, T. Schreck, A. Ferreira, A. Godil, and I. Pratikakis (Editors), pp 1—8.

[2] W. Brink, A. Robinson, M. Rodrigues, Indexing Uncoded Stripe Patterns in Structured Light Systems by Maximum Spanning Trees, BMVC 2008, Leeds, UK, 1-4 Sep 2008

[3] M.A. Rodrigues, A. Robinson, W. Brink, Fast 3D Reconstruction and Recognition, 8th WSEAS Int Conf on Signal Processing, Computational Geometry & Artificial Vision, Rhodes, 2008, p15-21.

[4] M.A. Rodrigues, A. Robinson, W. Brink, "Issues in Fast 3D Reconstruction from Video Sequences", Lecture Notes in Signal Science, Internet and Education, Proceedings of 7th WSEAS International Conference on MULTIMEDIA, INTERNET & VIDEO TECHNOLOGIES (MIV '07), Beijing, China, September 15-17, 2007, pp 213-218.

[5] M. Rodrigues, A. Robinson, L. Alboul, W. Brink,"3D Modelling and Recognition", WSEAS Transactions on Information Science and Applications, Issue 11, Vol 3, 2006, pp 2118-2122.

[6] A. Robinson, L. Alboul and M.A. Rodrigues,"Methods for Indexing Stripes in Uncoded Structured Light Scanning Systems", Journal of WSCG, 12(3), 2004, pp 371-378

[7] M.A. Rodrigues and Alan Robinson, *Image Processing Method and Apparatus*, European Patent Office, Patent GB2426618, 29 Nov 2006. Also published as WO2005076196 (A1), GB2410876 (A).

[8] FRGC, (2005). The Face Recognition Grand Challenge, http://www.frvt.org/FRGC/

[9] Bradski, G.R and V. Pisarevsky (2000). Intelapos' Computer Vision Library: applications in calibration, stereo segmentation, tracking, gesture, face and object recognition. Computer Vision and Pattern Recognition. Proceedings. IEEE Conference on Volume 2, 796 – 797.

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

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

[12] Wang, J. and M. M. Oliveira (2003). 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.

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

[14] Rodrigues, M.A, and A.Robinson (2009). Novel Methods for Real-Time 3D Facial Recognition. ATINER 5th Int Conf on Computer Sc and Info Sys, Athens, Greece, 27-30 July 2009.

[15] Turk, M.A. and A. P. Pentland (1991). Face Recognition Uisng Eigenfaces. *Journal of Cognitive Neuroscience* **3** (1): 71–86.