Face Recognition with real-time stereo

Jason Brooks and Gaurav Gujral and John Morris
Department of Electrical and Computer Engineering
The University of Auckland, Auckland, New Zealand

***Abstract*—We investigated the feasibility of using high resolution 3D face data to enhance existing recognition techniques. After isolating regions of interest (ROI) on the face, we located six key feature points - the eye corners, nose tip and the nose bridge - using the 3D data in their ROI. Since the stereo hardware used can operate at video frame rates, we determined distributions for the X,Y and Z coordinates of distances of each feature point from the nose tip and stored these distributions in our data base. Testing three faces against our database of 3D geometries *only* showed clear discrimination and demonstrated that acquisition of multiple 3D face images has clear potential for improving face recognition rates.**

I. Introduction

Automated face recognition is one of the holy grails of computer vision: thousands of studies[1] have attempted to find techniques that can be used in host of applications ranging from detection of terrorists trying to board aircraft to control of access to secure areas. New reviews appear almost yearly [1], [2], [3], [4], [5]. Most studies have focused on single camera 2D images and, although impressive recognition rates are often claimed, they rely on controlled conditions, *e.g.* subject facing the camera, good lighting and unoccluded (no glasses, hats, beards, ...) faces. This motivated us to see whether a high resolution stereo system could provide better match rates. We report some preliminary experiments to build 3D face models based on the high resolution data obtained from the Auckland real-time stereo hardware [6].

A. Selected Previous work

One study in a relatively uncontrolled environment - Palm Beach International Airport - correctly identified only 47% out of 958 faces [7]. Many different techniques have been used for 2D face recognition: the eigenfaces and Hidden Markov Model techniques are the most widely used. They work by extracting landmarks on the face, such as the inter-eye distance and the jaw length [8]. However, these measurements have low accuracy under non-ideal conditions, making them generally unsuitable for security applications, especially outdoors where lighting changes significantly impact matching accuracy [4].

B. Aims

We hypothesized that

- Auckland's real-time stereo system, which can produce rectified left and right images, disparity and occlusion maps for 768×1024 pixel images at 30 frames per second (fps) with $\sim 1\%$ depth resolution [9], would provide sufficient additional data to make a significant improvement in recognition rates in uncontrolled environments.
- Feature points would be more accurately located using disparity or depth data than image intensities.
- By using a rapidly acquired sequence of 3D images, some simple signal averaging would improve the quality of 3D models acquired.
- By allowing the target's head to move while we captured 3D data, we would be able to recover a more complete model of the subject's head.
- By storing distributions of values for key biometrics, such as eye separation, rather than individual values we would produce a more useful correct match probability.

[1]The corpus on face recognition was growing by more than 400 paper per annum in 2007.

II. Stereo System

A. Optical configuration

For our experiments, we estimated the average dimensions of a human head and doubled them - assuming that the target head would lie within a $340 \times 500 \times 400$ m volume and that we would need a depth resolution of ± 2.5 mm over most of that volume to adequately measure key features. Preliminary calculations showed that a canonical stereo configuration (optical axes of both cameras parallel and perpendicular to the base line between them and scan lines parallel) would not provide the target depth resolution over that volume with our 768×1024 pixel cameras. However, a verging axis configuration [10] - optical axes 'verged' to meet at a fixation point in the scene (see Figure 1) - provides our target depth resolution - see Table I.

B. Stereo Hardware

The stereo rig in Auckland's Photogrammetry Laboratory has two monochrome Sentech CL83 cameras connected to an Altera EP3S2 FPGA on a ProcStar III card which rectifies the images, generates disparity and occlusion maps and transfers them to a host computer via a PCI-express bus. Gimel'farb's Symmetric Dynamic Programming Stereo (SDPS) algorithm [11] was used to compute disparities and visibility states. The hardware returns four 'images': rectified left and right images, disparity and the occlusion maps - see Figure 2: the disparity map has 128 levels [6] and is the one that would be seen by a Cyclopæan eye centred between the cameras. It allows only one change in disparity per scan line pixel - generating triangular depth profiles through occluded regions rather than large jumps [12]. A detailed description of the hardware has been published [13], [6].

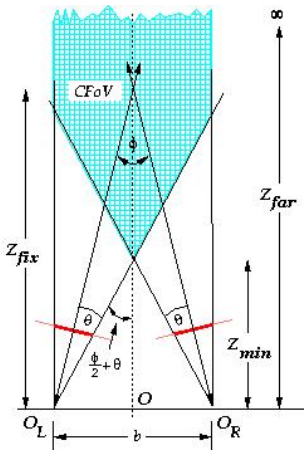


Fig. 1: Converging axis configuration
$O_{L|R}$: optical centre of the left|right camera
O: centre of baseline b
ϕ: vergence angle
Z_{min}: distance from O to the start of the CFoV
θ: half angle of view of the cameras
Z_{fix}: distance from O to the fixation point
Z_{far}: distance from O to the limit of the CFoV (∞ for $\phi < 2\theta$)

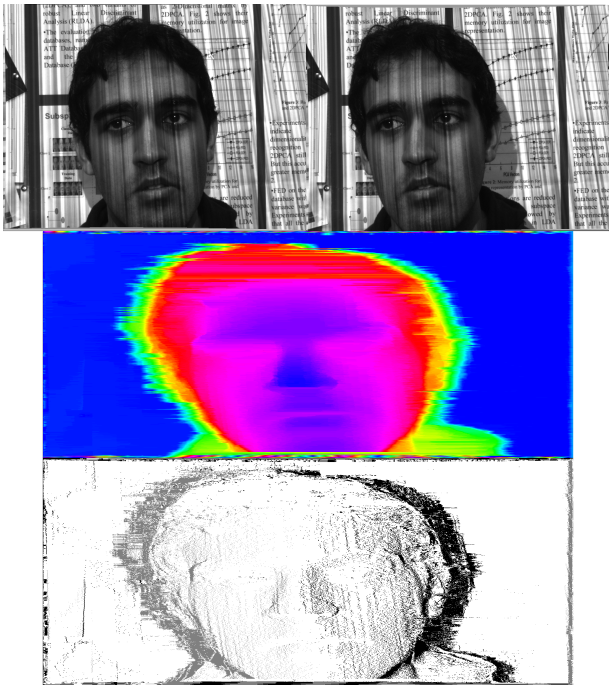


Fig. 2: Image set returned by the hardware: from top to bottom: Rectified left and right images, Disparity map (shown in false colour) and Occlusion map (white = binocular, grey = monocular left, black = monocular right)

.

TABLE I: Final system parameters

Baseline	b	100mm
Focal Length	f	9mm
Vergence Angle	ϕ	5°
Depth resolution $Z = 400mm$ $Z = 800mm$	ΔZ	 2.2 mm 3.8 mm

III. Stereo Configuration

To obtain accurate disparity maps of a face, we configured the system so that an entire face could fit inside the CFoV and the depth resolution was high enough to distinguish between facial features.

IV. Capturing Faces

We used active illumination to improve the matching: patterns of random width vertical lines were projected onto the subject's face. We used a visible colour pattern in these experiments, but the cameras used are sensitive in the near infrared (*i.e.* they have no IR block filter) and an 'invisible' (and less intrusive for a human subject) near IR pattern could be used[2]. We captured some images without the projected pattern and some with an RG665 (665nm long pass) filter in an attempt to enhance the contrast of the skin relative to the background. This filter's cut point is close to the hæmoglobin absorbance and thus it selects a spectral region where skin has a high reflectance [14].
For each test face, 3-4 seconds of video was captured: some faces were deliberately moved in this time interval.
The projected pattern improves the depth maps considerably (confirming previous work [12]) and images without it - initially taken to help improve 2D recognition - were not used due to the noisy disparity maps. They were initially taken to help improve 2D face recognition. The images with the visible block filter were also not used as it was found that the ratio of the intensity of the skin to the background did not improve. With the skin filter the ratio was 1:1.16 and without it the ratio was 1:1.50. Apparently skin's infrared reflectance is similar to that of many other common surfaces.

A. Deciding on Feature Points

We selected six feature points for this study - four eye corners, bridge of the nose and tip of the nose - focussing on points that do not change with facial expression. The eyebrows were not taken as they can be moved too easily. No points were taken from the mouth area as lips are routinely deformed. The ear is also a stable feature point but was not used as fine details of the inner ear shape cannot be captured without the subject's head turned. Although our approach gathers data from several viewpoints by allowing the subject's head to move, for this preliminary study, we focused on points that could be identified in almost every image. Other points can be readily added to our scheme to enhance recognition accuracy.

[2]A Kinect projects a suitable pattern

V. Locating Feature Points

We used widely available routines from the openCV library [15]. We also used a C++ library - the CITR classes - which provided routines to interface directly with the stereo hardware and files that it generates.

A. Isolating Regions of Interest

We first isolated areas of interest. We used a Haar classifier on the left image to locate the face. This classifier searches for regions in an image which match a template. It compares the intensities of adjacent rectangular regions with a previously trained Haar cascade for a match [16]. We used OpenCV's cascades for the face, eyes and nose.

We first isolated the face region and then located the eyes and nose inside the face region. If more than one region was found, we only took the region with the largest area. To reduce computation time, it only searched for faces larger than 350×350 pixels, noses larger than 60×60 pixels and eyes larger than 25×15 pixels. To find the eyes and nose, the face was split to reduce search time: *e.g.* only the top left half was searched for the left eye. An example of regions found by the Haar cascade is shown in Figure Figure 3.

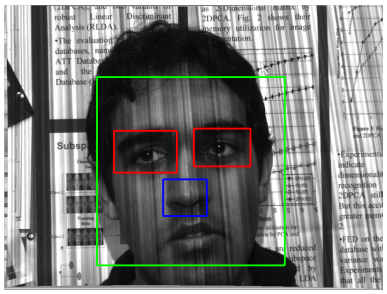


Fig. 3: Isolating regions of interest

The Haar cascade would locate these regions, even when the face was slightly turned. However, it would sometimes fail on images with movement blur. Stronger illumination - enabling the aperture time to be reduced - would alleviate this problem. With near IR active illumination, subjects would not notice this. The detected eye region sometimes included the eyebrow, which we removed by cropping the top 1/3 of the eye region.

B. Eye Corners

We first applied existing 2D routines for finding edges and corners to the left image. A Canny edge detector usually failed to outline the eye, because the projected pattern generated many additional edges. If we removed the projected pattern, the quality of the disparity maps dropped significantly.

Following Boulay [17], we also tried the openCV function `cvGoodFeaturesToTrack`, which locates strong corners by finding the minimum eigenvalues of the covariance matrix of the derivative of each pixel in an $N \times N$ neighbourhood. Before using this function, it was necessary to equalise the histogram in the eye region to increase the contrast between the eye and skin and to highlight the edges.

Fig. 5: Finding inner (left) and outer (right) eye corners

`cvGoodFeaturesToTrack` was then applied: the corners detected in a test image are circled in Figure 4.

(a) Before Histogram (b) After Histogram (c) Corners Found

Fig. 4: Finding eye corners with `cvGoodFeaturesToTrack`

This method successfully located the eye corners most of the time. However there was a problem with distinguishing these points from other corners that were detected, such as the iris and vertical lines from the projected pattern. Usually isolating the left and rightmost points and ensuring there is a minimum distance between them was enough to find the corners. However, for example, if there was a vertical line from the projected pattern before the eye corner, it would usually detect this point instead for all frames in the video. It is also vunerable to eye closures.

Low reliability of 2D methods prompted us to use depth information alone to find eye corners. We first split the eye into two regions and analysed each region individually to find inner and outer corners.

1) Inner Eye Corner: We examined at the gradient from the nose going towards the eye. Each row was searched horizontally starting from the top of the nose area. Our algorithm follows the nose downwards until it reaches a minimum which indicates it has reached the eye. A 3×3 window of all surrounding disparity levels at this point is taken, averaged and then stored as the inner eye corner. If another point is found on the next row, the 3×3 window is taken again and checked against the previous point found. If it has a lower disparity, then the new inner eye corner is updated. The 3×3 window reduces the effect of disparity errors, *e.g.* an incorrectly matched pixel may have a lower disparity than its actual depth. This process is illustrated[3] in Figure 5.

2) Outer Eye Corner: The outer eye corner was found by following the gradient on the side of the face until it reached the eye. This upwards gradient was followed until it starts to flatten out, indicating that it is approaching the eye. When the gradient starts increasing again, this point is a potential outer eye corner. As with the inner eye corner, each row is searched horizontally and the average disparity over a 3×3

[3]Face models generated using FaceGen [18]

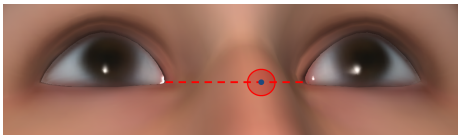


Fig. 6: Horizontal search for nose bridge

window stored at each detection. The point with the lowest average disparity is taken as the outer eye corner. This process is shown in Figure 5.

3) Nose Tip: The nose tip was found after isolating the nose region with the Haar cascade. We assume that points with the highest disparities form the nose tip area. The centre bottommost point in this area is taken as the nose tip. This method may fail when the face is tilted too much to one side.

4) Nose Bridge: Since the nose bridge is not guaranteed to be located in any of the regions of interest found by the Haar cascades, we found the nose tip and inner eye corners first. The area between the inner eye corners and above the nose tip is then searched. Two searches are made: a horizontal and a vertical one.

a) Horizontal Search: The area between the two inner eye corners is searched row by row for the highest disparity point. If multiple points are found, the centremost point is taken. The point it finds may not necessarily be the nose bridge, but lies along the line joining the nose tip to the nose bridge - see Figure 6.

b) Vertical Search: The angle between the nose tip and the point found in the horizontal search is found and a line drawn between these two points. The search area for this line is the length of the nose as detected by the Haar cascade, with the centre point being the point found in the horizontal search. Searching starts from the top of this line, when the gradient starts decreasing, the line is approaching the nose bridge. When the gradient starts increasing again, that point is taken as a potential nose bridge. As before, the lowest average disparity over a 3×3 window is the nose bridge. Figure 7 shows this process.

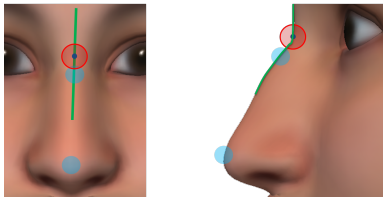


Fig. 7: Vertical search for nose bridge

VI. Generating 3D Model

Once all the feature points have been found, they were mapped into real world co-ordinates using the reprojection matrix found in the calibration step.

Before mapping the feature points, we generated a 3D point cloud of the face area using the real world measurements. This gives an idea of how accurate the depth on the face is.

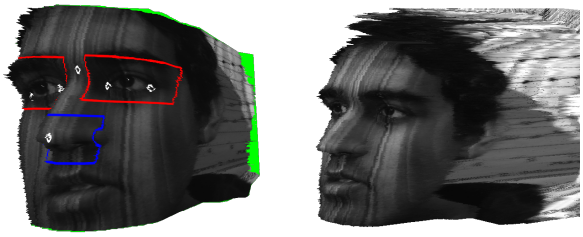


Fig. 8: Left: Feature points mapped to face; Right: Captured face after textures were reapplied

An example of a 3D point cloud generated by our program is shown[4] in Figure 8.

Textures were then reapplied to the point cloud to produce a 3D image. Intensities for each point in the disparity map were extracted from the left and right images using [11]:

$$x_L = \frac{(x+d)}{2} \quad x_R = \frac{(x-d)}{2} \tag{1}$$

where x is disparity map co-ordinate, d its disparity and $x_{L|R}$ coordinate of corresponding pixel in the left‖right image. Visibility states from the occlusion map were used: if the pixel was binocular or monocular left, x_L was used; if it was monocular right, x_R was used.

A. Mapping Feature Points

All the feature points were converted to real world co-ordinates and mapped onto the 3D model - see Figure 8.

This was then repeated for the remaining frames. Once all the frames have been processed, an image with good feature point detection was chosen as a reference image. All other images were registered to the nose tip in this image.

Registration used a simple algorithm in which the 3D images were first translated so that the nose tips matched and then rotated in the yaw, pitch and roll directions until a best fit the reference image points. Rotations from -10° to 10 ° with a step size of 0.5° were tested. We can use this simple (and fast!) approach because the head is severely constrained (our subjects are instructed not to break their necks during experiments) and does not move significantly from frame to frame. Using a finer step moved the position by < 1 mm. The best fit was defined as the rotation that gives the smallest sum of Euclidean distances between the image feature points and the reference feature points. Registered feature points from one sequence of 100 images is shown in Figure Figure 9.

B. Feature point distributions

The distribution for each coordinate of each feature point was determined. These distributions determine the range of acceptable values for each feature point in the recognition phase. The vector displacements for each feature point from the nose tip were stored. The distributions of displacement X, Y and Z components were stored rather than, for example, the

[4]3D Model viewed using MeshLab [19]

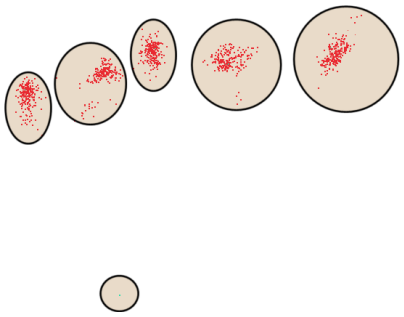


Fig. 9: Registered feature points - classes (*e.g.* inner eye, *etc.*) are circled

Euclidean distance because two points in quite different places could still have the same Euclidean distance from the origin. Individual coordinates have a greater discrimination ability. The distributions for the inner left eye corner for an example face are shown in Figure 10.

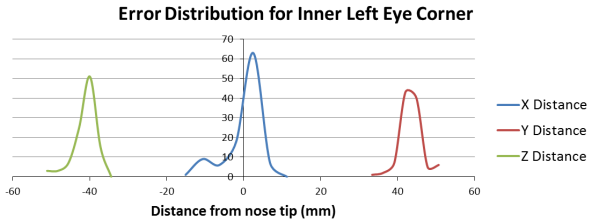


Fig. 10: Distribution of X, Y and Z distances from the inner left eye corner to the nose tip. The reference face is chosen rather arbitrarily as the one in which the feature points are found with high confidence: it may not be facing the cameras directly.

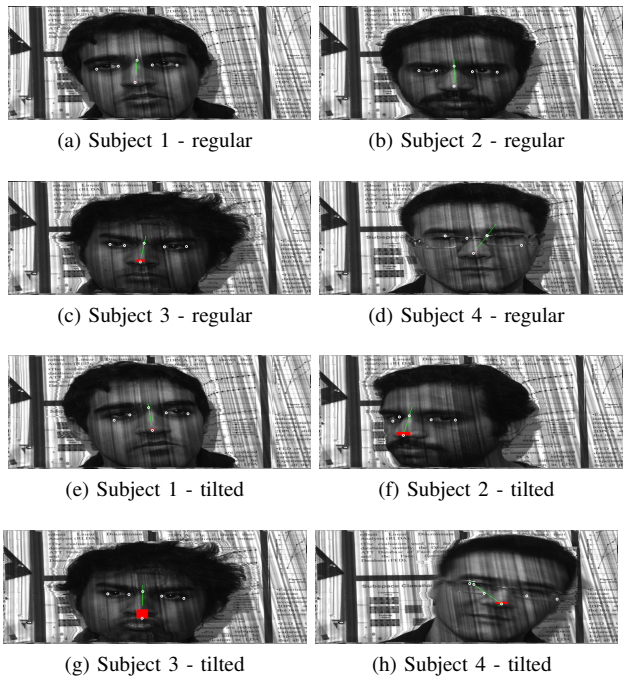
We observed that the errors were approximately normally distributed for a number of image sets, so only the mean, μ, and standard deviation, σ, for each displacement component was stored.

VII. Recognition

To test recognition, new face data was captured and the feature points determined as described above. Then the feature points on the new face were rotated to obtain a best fit to the database image as the two faces may be be in quite different poses. Once the test image is rotated, X, Y and Z displacements from the nose tip were compared to the (μ, σ) of the corresponding measurement in the database image. A 'probability' was then assigned to each measurement depending on distance from the mean. The mean was assigned as a 100% match: outside 2σ, the probability was set to 0%. Other values were linearly interpolated between these two points. These probabilities were then averaged to generate a final match score for the new face against all other faces already in the database.

A. Running time

Measurements were taken on a Dell XPS^{TM} M1330 computer with Intel Core 2 Duo (2 cores at 2.2GHz), L1 cache - 32KB instruction, 32 KB data per core, L2 cache - 4 MB per core, and 2GB RAM. Based on averages over 100 images of different faces, it took 320 ms to locate facial features and 725 ms for best fit to reference.



(a) Subject 1 - regular (b) Subject 2 - regular

(c) Subject 3 - regular (d) Subject 4 - regular

(e) Subject 1 - tilted (f) Subject 2 - tilted

(g) Subject 3 - tilted (h) Subject 4 - tilted

Fig. 11: Four subjects in regular and angled poses

TABLE II: Average σ for X, Y and Z measurements

Feature	X (mm)	Y (mm)	Z (mm)
Left Eye Inner Corner	3.25	3.15	3.18
Left Eye Outer Corner	2.57	3.18	3.05
Right Eye Inner Corner	2.73	2.10	2.35
Right Eye Outer Corner	2.90	3.03	3.07
Nose Bridge	1.50	3.25	3.19

Feature point detection accuracy is demonstrated in Figure 11. White circles are drawn around the detected feature points. The red box on the nose represents the highest disparity points in the nose area and the green line shows the line between the nose tip and bridge.

First, glasses prevented the detection of all feature points as seen in subject 4 in Figures 11. Ignoring subject 4, for subjects facing the camera with a blank expression, feature points were detected correctly even when the head is slightly tilted. The eye corner detection however sometimes detects a point below the eye when the face is being tilted in the yaw direction as can be seen in subjects 1 and 2 in Figure 11. The nose tip and bridge are detected accurately except for the nose tip when there is a higher disparity point in the image, *e.g.* when subject 3 pouts his lips in Figure 11.

Table II shows the average standard deviation for each feature point from its mean value (subject 4 was excluded).

We undertook some basic recognition tests with 3 faces: 10%

of the data was randomly selected for testing and the remainder put in the database. Match scores are shown in Table III - 100% indicates perfect match.

TABLE III: Match (confidence) scores for three different faces

	S1 (10%)	S2 (10%)	S3 (10%)
S1 (90%)	**85%**	53%	43%
S2 (90%)	44%	**87%**	33%
S3 (90%)	63%	69%	**84%**

VIII. Limitations

A limitation of our system is that the active illumination requires a pattern to be projected on the users face, which takes away the non-intrusive nature of the system. However, since the images are all monochrome and good images were obtained with the visible block filter (RG665) in place, an 'invisible' (to the human eye) infrared illumination pattern should produce similar results.

Our experiments showed some limitation for face movement. Our system handled tilt in the roll direction well but was less tolerant of movement in the pitch and yaw directions. Additional work on feature point detection from the disparity maps is indicated.

As might be expected, feature point detection failed when facial accessories such as glasses are worn. We did not take this in to account in this preliminary study.

IX. Conclusions

We established a technique for modelling a face in 3D using distributions of positions of feature points relative to a reference point (nose tip). We use conventional Haar classifiers to locate each face on the original left image. However conventional 2D techniques for finding feature points did not work - mainly because the projected patterns interfered with the classifiers - so we developed techniques to determine individual points by following paths in the disparity map. Rather than store some representative locations of each feature point (compressing the database entry to a handful of 3D displacements) we propose storing the distributions of the X,Y and Z co-ordinates of the feature points acquired from long sequences of images. When matching a test image against the database, we can now obtain a score that the test image matches the stored one, giving a reliability to each possible match. Currently our prototype takes about 1s to process an image - or significantly slower than real time. This is not a problem for creating the feature distributions that make up the database of known faces as (potentially hundreds of) images of a subject can be streamed to disc for later processing. In recognition mode, it can still capture - in a very short time - many images of a subject from slightly different viewpoints due to natural motion of the subject's head and thus increase the reliability of the 'match or not match' question.

We abandoned well established 2D recognition techniques because of the interference from the active illumination which was key to obtaining good depth maps and focused on the use of the disparity maps - unique to this study. We observe that simple modifications such as turning on the active illumination for every second frame would enable us to combine established techniques with our accurate depth data to improve recognition rates. Also facial additions such as glasses and beards are easily detected with our 3D data and thus eliminated from matching.

References

[1] N. B. Kachare and V. S. Inamdar, "Article: Survey of face recognition techniques," *International Journal of Computer Applications*, vol. 1, no. 1, pp. 29–33, February 2010, published By Foundation of Computer Science.

[2] R. Jafri and H. R. Arabnia, "A survey of face recognition techniques," *JIPS*, vol. 5, no. 2, pp. 41–68, 2009. [Online]. Available: http://dx.doi.org/10.3745/JIPS.2009.5.2.041

[3] X. Zhang and Y. Gao, "Face recognition across pose: A review," *Pattern Recogn.*, vol. 42, pp. 2876–2896, November 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1563046.1563061

[4] A. F. Abate, M. Nappi, D. Riccio, and G. Sabatino, "2D and 3D face recognition: A survey," *Pattern Recognition Letters*, vol. 28, no. 14, pp. 1885 – 1906, 2007.

[5] W. Zhao, R. Chellappa, A. Rosenfeld, and P. Phillips, "Face recognition: A literature survey," *ACM Computing Surveys*, pp. 399–458, 2003.

[6] J. Morris, K. Jawed, G. Gimel'farb, and T. Khan, "Breaking the 'ton': Achieving 1% depth accuracy from stereo in real time," in *Image and Vision Computing NZ*, D. Bailey, Ed. IEEE CS Press, 2009.

[7] "Facial Recognition System Test (Phase I) Summary," Palm Beach County Department of Airports, 2002.

[8] Electronic Privacy Information Center, "Face Recognition," Retrieved September 10, 2011 from: http://epic.org/privacy/facerecognition/.

[9] K. Jawed, J. Morris, T. Khan, and G. Gimel'farb, "Real time rectification for stereo correspondence," in *7th IEEE/IFIP Intl Conf on Embedded and Ubiquitous Computing (EUC-09)*, J. Xue and J. Ma, Eds. IEEE CS Press, 2009, pp. 277–284.

[10] K. Jawed and J. Morris, "Verging axis stereophotogrammetry," in *Proc PSIVT'2011*. Springer-Verlag LNCS, 2011.

[11] G. L. Gimel'farb, "Probabilistic regularisation and symmetry in binocular dynamic programming stereo," *Pattern Recognition Letters*, vol. 23, no. 4, pp. 431–442, 2002.

[12] A. Woodward, D. An, Y. Lin, P. Delmas, G. Gimel'farb, and J. Morris, "An evaluation of three popular computer vision approaches for 3-d face synthesis," in *Structural, Syntactic, and Statistical Pattern Recognition*, ser. Lecture Notes in Computer Science, D.-Y. Yeung, J. T. Kwok, A. L. N. Fred, F. Roli, and D. de Ridder, Eds., Aug. 2006, pp. 270–278. [Online]. Available: http://dx.doi.org/10.1007/11815921_29

[13] J. Morris, K. Jawed, and G. Gimel'farb, "Intelligent vision: A first step - real time stereovision," in *Advanced Concepts for Intelligent Vision Systems (ACIVS'2009)*, ser. LNCS, J. Blanc-Tallon, Ed., vol. 5807. Springer, 2009, pp. 355–366.

[14] E. Angelopoulou, "The Reflectance Spectrum of Human Skin," Department of Computer & Information Science, University of Pennsylvania, Tech. Rep., 1999.

[15] "Welcome - OpenCV Wiki," Retrieved September 10, 2011 from: http://opencv.willowgarage.com/wiki/.

[16] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–511 – I–518 vol.1.

[17] G. Boulay, "Eye Pose Tracking & Gaze Estimation," Master's thesis, Institut National des Sciences Appliques de Lyon, France, 2008.

[18] "FaceGen Modeller: 3D Face Generator," Retrieved September 11, 2011 from: http://www.facegen.com/modeller.htm.

[19] "MeshLab, a tool developed with the support of the 3D-CoForm project," Retrieved September 11, 2011 from: http://meshlab.sourceforge.net/.