3D Face Recognition Using Multiview Keypoint Matching

Michael Mayo, Edmond Zhang
Department of Computer Science, University of Waikato, New Zealand
{mmayo, ez1}@cs.waikato.ac.nz

Abstract

A novel algorithm for 3D face recognition based point cloud rotations, multiple projections, and voted keypoint matching is proposed and evaluated. The basic idea is to rotate each 3D point cloud representing an individual's face around the x, y *or* z *axes, iteratively projecting the 3D points onto multiple 2.5D images at each step of the rotation. Labelled keypoints are then extracted from the resulting collection of 2.5D images, and this much smaller set of keypoints replaces the original face scan and its projections in the face database. Unknown test faces are recognised firstly by performing the same multiview keypoint extraction technique, and secondly, the application of a new weighted keypoint matching algorithm. In an extensive evaluation using the GavabDB 3D face recognition dataset (61 subjects, 9 scans per subject), our method achieves up to 95% recognition accuracy for faces with neutral expressions only, and over 90% accuracy for face recognition where expressions (such as a smile or a strong laugh) and random face-occluding gestures are permitted.*

1. Introduction

Face recognition is the one of the most challenging pattern recognition problems. It humbles the most powerful of computers, and renders the most sophisticated of algorithms intractable. Psychologists, cognitive scientists, and computer vision scientists have invested decades of research into solving this problem, with some tremendous advances – yet face recognition is still largely not understood. Human beings, on the other hand, know nothing consciously about *how* face recognition is performed, yet they solve this problem adeptly and routinely every day of their lives.

In this paper, we contribute to the state-of-the-art in 3D face recognition by proposing a novel method for recognition based on matching and voting keypoints that are extracted from multiple 2.5D views of each 3D face. Our method is evaluated on the GavabDB database [1] of 3D faces.

This paper essentially has three key novel contributions.

Figure 1. Examples of different 2.5D frontal face images generated after rotating the same 3D point cloud about the x axis by (a) -10°, (b) 0°, and (c) +10°.

Firstly, rather than taking a single 2.5D projection, we rotate the point cloud incrementally about its centre of mass, along one or more of the x, y, and z axes. Then, rather than taking only a single 2.5D projection, we take multiple projections.

Figure 1 depicts examples of some 2.5D views taken by rotating the same 3D point cloud. This, as our results show, provides significant additional features and dramatically increases face recognition accuracy.

The second main contribution is our new method for keypoint matching. Traditionally, keypoints such as the Scale Invariant Feature Transform (SIFT) [2] are matched as follows: if a test image contains at least three (or some other constant number of) keypoint matches with some target object, then the test image is considered to contain, somewhere in the image, that object.

In face recognition, however, this matching method is not feasible because faces are all, more or less, visually very similar. Keypoint matching between two faces yields many more matches than it would if the two objects were clearly distinct. The standard approach to keypoint matching therefore results in poor recognition performance when it comes to faces.

We propose instead to match all keypoints in a test image against all keypoints taken from the multiple 2.5D views of each labelled point cloud. In other words, we first of all rotate each training point cloud to project multiple 2.5D views, and then we extract the keypoints from each view. We then combine all of the keypoints from all of the views

978-0-7695-3718-4/09 $25.00 © 2009 IEEE
DOI 10.1109/AVSS.2009.11

into one complete set of labelled keypoints. The original point cloud and the set of 2.5D images which are now no longer needed can be discarded.

In order to recognise a individual's face, then, we firstly project the face's point cloud onto one or more 2.5D images as we did with the training faces, extracting the keypoints. The closest matching labeled keypoint for each unlabeled test keypoint is then determined, before each test keypoint "votes" in a weighted fashion on the class or identity of the face. All test keypoints, therefore, are utilised in the matching process rather than just a fixed number.

The third and final contribution is our evaluation result. On the GavabDB database, we demonstrate over 95% accuracy on recognition of faces with neutral and smiling expressions, and over 90% accuracy on average for situations where stronger expressions (such as an accentuated laugh) and random face-occluding gestures are permitted. This is a new state-of-the-art result on this challenging dataset.

Additionally, we do not rely on any expensive preprocessing techniques such as face detection and cropping, and facial feature detection (eyes, ears etc) that other systems employ. Our results therefore represent the minimum performance achievable using only our proposed algorithm, and further improvements are possible via more preprocessing.

2. Background

In this section, we give brief overviews of the fields of 3D face recognition, and the idea of keypoints and keypoint matching.

2.1. 3D Face Recognition

3D face recognition, as opposed to the more traditional 2D face recognition, uses a 3D camera such as a laser range sensor to image a person's face. Whereas traditional optical cameras return a 2D intensity image, laser range sensors typically return a "point cloud" in 3D space. Often these points are arranged in strips in (x,y) space, with the z coordinate of each point indicating its depth.

Point clouds can be projected onto 2.5D images. A 2.5D image is, simply, a mapping of the 3D point cloud to a grey scale image in which the intensity is inversely proportional to the depth.

Previously, researchers have investigated face recognition from a single 2.5D projection of a 3D point cloud [3]. The main problem with this approach is that 2.5D images lose critical information about the face. That is, a point cloud is a true three-dimensional structure, yet a 2.5D image is still only a two-dimensional structure. If some facial features are obscured by rotation before the 2.5D image is projected, then the image consequently will lack that feature. And even if an image feature is captured in a 2.5D projection, it may look different subject to slight variations in the 3D pose of the point cloud.

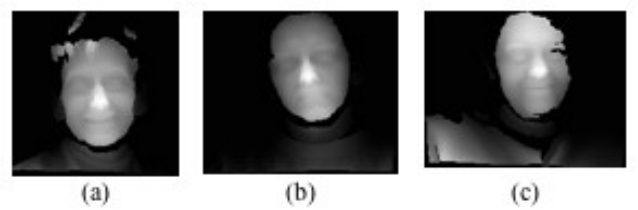

Figure 2. Examples of 2.5D projections taken of three different subjects.

It is beyond the scope of this paper to provide a detailed survey of the area of 3D face recognition here, but the interested reader is directed to a recent survey paper [3].

2.2. SIFT Keypoint Matching

A good image keypoint, according to Lowe [2], must be highly distinctive and have a low probability of mismatch. It should be tolerant to image noise and changes in illumination, and it should also be uniform in the presence of scaling, rotation, minor changes in viewing direction, and local distortions.

The SIFT descriptor has, for many years, been the most well-known and popular choice of keypoint, because it best satisfies all of these criteria. It also operates only on a grey scale representation of an image, making it suitable therefore for 2.5D images, where there is no colour information to lose.

Briefly, a SIFT descriptor for a small image patch, for example of size 4×4, is computed from the gradient vector histograms of the pixels in the patch. There are 8 possible gradient directions per pixel, and therefore the total size of the SIFT descriptor is $4\times4\times8=128$ elements. This feature vector is normalized to enhance invariance to changes in illumination, and transformed in other ways to ensure invariance to scale and rotation as well.

Although many possible keypoints at different locations in an image could be computed, only the most distinctive and invariant ones useful for matching are actually retained. These often fall on edges, corners, points, or other "interesting" parts of the image; and they can be off many different sizes and orientations as well.

3. Multiview Keypoint Voting Algorithm

In this section, we describe our proposed approach in detail.

3.1. 3D Point Cloud to 2.5D Image Set Conversion

SIFT keypoints can only be extracted from 2D images. Our approach involves converting 3D point clouds into 2D

images, where image intensity represents depth, i.e. the images are 2.5D. Keypoints can then be extracted from these images.

Algorithm 1 gives the basic steps employed to achieve this for a single projection.

Algorithm 1 3D point to 2.5D image conversion.

Input: A 3D point cloud

1: Compute the extrema of the point cloud along each of the three axes, obtaining X_{min}, X_{max}, Y_{min}, Y_{max}, Z_{min}, Z_{max}
2: Create a 2D image of width $\frac{X_{max}-X_{min}}{2}$ and height $\frac{Y_{max}-Y_{min}}{2}$
3: Scale the z-value of the points in the cloud to the range 1...255
4: Project points onto the 2D image pixels, setting each pixel to the scaled z value. Pixels that do not have any 3D points projected on to them are set to zero.

Output: A 2.5D image

For each point cloud, Algorithm 1 is executed multiple times as we rotate the point cloud incrementally about its *x*, *y*, and *z* axes. We compute a new 2.5D image with each step of the rotation, so that by the end of this process, multiple 2.5D images have been extracted.

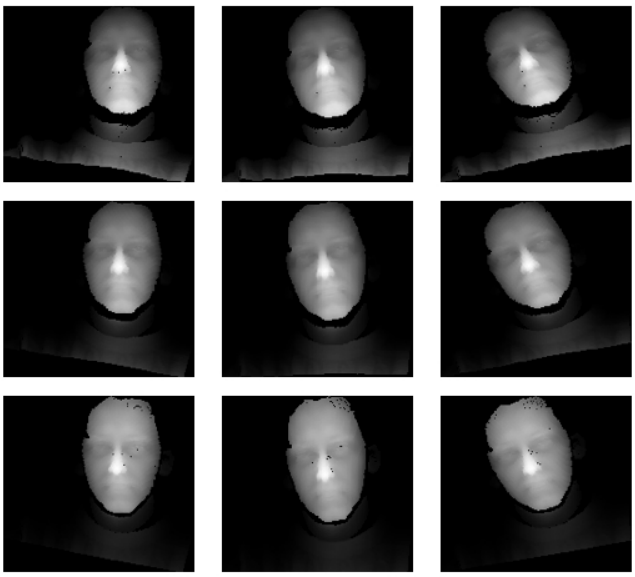

Figure 3. Examples of different 2.5D images generated after rotating the same 3D point cloud about the x (the rows) and z (the columns) axes in increments of 10°.

Figure 3 gives some examples of different 2.5D images formed from the same point cloud, after rotating it about the *x* and *z* axes in increments of 10°.

Keypoints are then detected and extracted from this set of 2.5D images rather than from a single image. By extracting keypoints in this way, more quality keypoints can be found, and the same feature (for example, a nose), can be captured by keypoints at many different viewing angles.

3.2. Multiview Keypoint Voting

Keypoints are extracted from both the labeled training faces and the unlabeled testing faces according to the process described in the previous section. Once the keypoints are extracted from all of the views, the original 3D and 2.5D data can then be discarded, and the keypoints are combined into a single set representing one face.

In order to classify the test faces, we use a novel keypoint voting algorithm depicted in Algorithm 2.

Algorithm 2 The keypoint voting algorithm.

Inputs: (i) A set *K* of labeled keypoints extracted from the training images. Keypoints are labeled by the class of the image they came from; (ii) A set *T* of unlabeled keypoints extracted from a single test face using the keypoint extraction method.

1: **for each** $t \in T$ **do**
2: Find the closest matching $k \in K$ according to the Euclidean distance function, $dist(t, k)$
3: Assign the label of k to t
4: Set the weight of t to $\frac{1}{dist(t,k)}$
5: **end for**
6: Each $t \in T$ then votes for its labeled class with its weight.
7: The final classification of the image is the class with the greatest total vote.

Output: A classification for the unknown test face.

Algorithm 2 does this: the set of test keypoints are matched to the training keypoints, and the best matching training keypoint according to Euclidean distance is determined.

The test keypoint is then given both a class, which is the same class as its best matching training keypoint, and a weight, defined as its inverse Euclidean distance from its best match. A test keypoint that closely matches a training keypoint, therefore, would have a very high weight; conversely, a poorly matching keypoint would have low weight.

Each test keypoint then "votes" on the final classification of the entire point cloud. The vote is a simple summation of the weights of the test keypoints by class. The vector of total weights is then normalised and returned as a probability distribution.

The unknown face can then be either classified according to the class with the highest probability, or rejected if the highest probability is not sufficient. In our evaluations, we always classified a test face according to its most probable identity.

Scan ID	Description
frontal1	frontal head orientation, neutral expression
frontal2	frontal head orientation, neutral expression
frontal3	frontal head orientation, strong smiling expression
frontal4	frontal head orientation, accentuated laugh
frontal5	frontal head orientation, random gesture occluding face
up	frontal but looking up (+35°), neutral expression
down	frontal but looking down (-35°) , neutral expression
right	right head profile (+90°), neutral expression
left	left head profile (-90°), neutral expression

Table 1. Description the nine different images of each individual in the GavabDB dataset, reproduced and enhanced from [1].

4. Evaluation

We evaluated our 3D face recognition algorithm in three different experiments. The main question was whether our method of rotating point clouds to project multiple 2.5D views really works, and if so, which axes of rotation produced the best performance. We describe first of all the dataset used for the experiments, and then experiments and results themselves.

4.1. Dataset

The GavabDB dataset [1] is one of the available public benchmark datasets for 3D face recognition. The problem with most other 3D face datasets is that they contain only limited variability. For example, some datasets contain variations in head orientation, but the variations are quite limited; conversely, others contain scans of faces with expressions, but the expressions are mild.

The GavabDB dataset, in contrast, was deliberately designed with the intent of introducing considerable variability in head position, orientation, and facial expression. It is therefore one of the most challenging 3D face recognition datasets.

In terms of specification, the dataset consists of scans from 61 different individuals (45 male, 16 female), with nine different images of each individual, giving a total of 549 images. Only two of the images per individual are frontal and expression-neutral; the remainder consist of strong variations in pose and expression. The descriptions of each scan are given in Table 1.

4.2. Experiment Overview

In each experiment, we used *only* one or two facial scans per subject for training; the remaining seven or eight scans were reserved for testing. This represents a low proportion of training data (11% or 22% of the total data respectively), but in practice it is a realistic scenario, as the cost of obtaining many 3D face images for each individual is likely to be very high.

In Experiment 1, therefore, we used for training data only one of the frontal scans with a neutral expression, specifically, the *frontal1* scans as described in Table 1. This gave a total of 61 training images. Testing was then carried out on the remaining point clouds.

In Experiment 2, we increased the amount of training data to two point clouds per subject, selecting the neutral-expression *frontal1* images along with an image with a smiling expression, namely the *frontal3* images (see Table 1). This brought the total number of point clouds used for training to 122 out of the total 549 scans.

Finally, in Experiment 3, we were interested in seeing if machine learning could further improve our recognition rates. Our method already reduces each 3D point cloud to a set comprising a hundred or so 128-dimensional keypoints. We wanted to know if a machine learning algorithm could effectively build a "model" of the keypoints, which could then label unknown keypoints in a more efficient manner than direct matching.

4.3. Experiment 1

In this experiment, we used a single neutral-expression 3D face per subject for training, and tested on the remaining scans. Table 2 depicts the results.

We first of all used our keypoint matching algorithm to perform face recognition without any rotations at all – that is, we generate only one 2.5D image per training and testing face, and match them using the keypoint voting algorithm. This represents the baseline case as depicted in the second column of the table. Experimental results are given in the remaining columns.

The table also provides average recognition rates by scan type. For example, after training on the *frontal1* scans, the recognition rate for *frontal4* scans (an accentuated laugh) was 57.38%. The results also give the overall average recognition rate for frontal scans, which in the baseline case is 66.81%. Finally, the recognition rates for the non-frontal scans (such as subject looking up, or down, or when the subject is in profile), as well as an overall average for all the test cases, is also given.

Unsurprisingly, the recognition rates for side profiles when the system is trained only on frontal face images is low, though still above the $\frac{1}{61}$ or 1.6% classification rate that would be expected due to pure chance. We have included these results for the sake of completeness.

We tested many different types of point cloud rotation in this experiment. In each column of Table 2, the training images were rotated in increments of 10° starting at -10°

Test Data	Baseline	Training Images Rotated						Training and Test Images Rotated					
		$\pm 10^\circ x$	$\pm 10^\circ y$	$\pm 10^\circ z$	$\pm 10^\circ xy$	$\pm 10^\circ xz$	$\pm 10^\circ yz$	$\pm 10^\circ x$	$\pm 10^\circ y$	$\pm 10^\circ z$	$\pm 10^\circ xy$	$\pm 10^\circ xz$	$\pm 10^\circ yz$
frontal2	81.97	81.97	83.61	85.25	88.52	86.89	90.16	86.89	90.16	90.16	90.16	91.80	**95.08**
frontal3	70.49	78.69	75.41	78.69	81.97	**85.25**	**85.25**	83.61	80.33	**90.16**	86.89	86.89	**90.16**
frontal4	57.38	68.85	63.93	73.77	80.33	**81.97**	75.41	81.97	80.33	81.97	88.52	90.16	**93.44**
frontal5	57.38	62.30	67.21	65.57	62.30	**72.13**	70.49	65.57	72.13	67.21	75.41	**80.33**	78.69
frontal average	*66.81*	*72.95*	*72.54*	*75.82*	*78.28*	***81.47***	*80.25*	*79.51*	*80.74*	*82.38*	*85.25*	*87.30*	***89.34***
up	16.39	21.31	19.67	21.31	**27.87**	24.59	19.67	24.59	29.51	22.95	**39.34**	**39.34**	27.87
down	24.59	22.95	**31.15**	24.59	27.87	**31.15**	24.59	32.79	32.79	21.31	45.90	**49.18**	39.34
left	6.56	3.28	3.28	3.28	8.20	**9.84**	3.28	3.28	8.20	13.11	13.11	11.48	**14.75**
right	4.92	8.20	4.92	4.92	**9.84**	8.20	4.92	9.84	6.56	6.56	**14.75**	11.48	13.11
overall average	*30.96*	*43.44*	*48.57*	*43.65*	***50.00***	*44.67*	*49.18*	*48.36*	*56.76*	*50.00*	***57.58***	*46.72*	*56.56*

Table 2. Experiment 1 results. The *frontal1* images are used for training.

and ending at +10°. The axes of rotation were either x, y or z (yielding three 2.5D images per scan) or a pair of axes (e.g. x and z), yielding nine 2.5D images – see Figure 3 for an example.

We also tested the idea of rotating the test faces as well as the training faces in order to obtain more unlabeled test keypoints – and the results of these runs are given in the second set of columns in Table 2.

The results clearly show that, by and large, rotating both the training and test faces about the y and z axes yield the most accurate recognition rates. For neutral face recognition, the success rate reaches 95.08% – quite an increase over the baseline of 81.97%. The average recognition rate for all the frontal images, where expressions and gestures are permitted, reaches 89.34% when rotation is utilised compared to 66.81% in the baseline case.

These results also compare very favourably to other methods evaluated on the same dataset, for example, Moreno et al. [5], who reported 78% accuracy on frontal face recognition.

4.4. Experiment 2

In the second experiment, we tested the idea that "more is better" by increasing the amount of training data per individual. The facial images in which the individuals are smiling (the *frontal3* images) were added to the nuetral-expression *frontal1* images as training data. We then repeated the same experiment as in the case of Experiment 1. Table 3 gives the results.

Overall, the table shows frequent increases in recognition rates, and when the test images are not rotated as well (the left columns of the table), there is an accuracy boost of approximately 10% on average for frontal face recognition. However, the best result from Experiment 1 is never significently exceeded.

Of interest is the recognition rate for the test images with the subject laughing (the *frontal4* images). The best case recognition rate for this is 95.16% in this experiment, showing that training on faces with a smiling expression is conducive to also recognizing laughing faces.

4.5. Experiment 3

In Experiment 3, we extracted the keypoints in the normal way, but instead of performing direct matching for classification, we instead used machine learning.

The idea behind this was to see if a model built by a classifier, which would in theory be much smaller in terms of storage requirements than the keypoints set, could label the unlabeled keypoints in the test images more effectively than the direct matching process that we employed in Experiments 1 and 2.

Test Data	Baseline	MultiClass C4.5 $\pm 10^\circ yz$	1 vs. All C4.5 $\pm 10^\circ yz$
frontal2	59.02	**83.16**	80.33
frontal3	40.98	70.49	**73.77**
frontal4	27.87	**68.85**	65.57
frontal5	29.51	**57.38**	57.38
frontal average	*39.35*	***70.08***	*69.26*
up	13.11	26.23	**27.87**
down	13.11	**34.43**	31.15
left	9.84	**16.39**	14.75
right	9.84	13.11	**14.75**
overall average	*25.41*	***46.31***	*45.70*

Table 4. Experiment 3 results.The *frontal1* images are used for training.

The particular classifier we chose was C4.5 [4], a powerful and often utilized decision tree learner.

In the first case (MultiClass), we built a single decision tree from all of the keypoints extracted from all of the training faces. For problems with a large number of classes (61 in our case), this can result in a very large decision tree being constructed.

In the second case (1 vs. All), we built one decision tree per subject, with the positive class being those keypoints extracted from the subject's face image, and the negative class being all of the other keypoints. Unlabeled keypoints were then predicted by averaging the predictions of each of the individual trees. This method is more scalable to a large number of individuals, as it produces many small trees rather than one large tree.

Test Data	Baseline	Training Images Rotated						Training and Test Images Rotated		
		$\pm 10^\circ x$	$\pm 10^\circ y$	$\pm 10^\circ z$	$\pm 10^\circ xy$	$\pm 10^\circ xz$	$\pm 10^\circ yz$	$\pm 10^\circ x$	$\pm 10^\circ y$	$\pm 10^\circ z$
frontal2	85.48	87.10	90.32	87.10	**93.55**	**93.55**	90.32	90.32	91.94	**93.55**
frontal4	83.87	83.87	87.10	88.71	85.48	**91.94**	90.32	93.55	**95.16**	93.55
frontal5	77.42	75.81	80.65	80.65	82.26	**85.48**	82.26	80.65	**87.10**	82.26
frontal average	*82.25*	*82.26*	*86.02*	*85.49*	*87.10*	***90.32***	*87.63*	*88.17*	***91.40***	*89.79*
up	29.03	38.71	35.48	30.65	**45.16**	41.94	32.26	**37.10**	35.48	33.87
down	25.81	33.87	32.26	32.26	**43.55**	38.71	33.87	**45.16**	**45.16**	37.10
left	6.45	**11.29**	**11.29**	4.84	**11.29**	**11.29**	**11.29**	11.29	9.68	**14.52**
right	9.68	12.90	9.68	9.68	12.90	**14.52**	11.29	**17.74**	11.29	9.68
overall average	*45.39*	*49.08*	*49.54*	*47.70*	*53.46*	***53.92***	*50.23*	***53.69***	***53.69***	*52.08*

Table 3. Experiment 2 results. The *frontal1* and *frontal3* images are used for training.

The results of this experiment are given in Table 4. Although there is significant improvement over the baseline case with no rotation, there is no improvement over the results achieved in Experiments 1 and 2.

5. Conclusion

The results of all of the experiments demonstrate significant accuracy increases over the baseline scenarios in all cases. Especially pleasing is the high recognition rate for neutral-expression frontal 3D face recognition.

3D face recognition in many ways is a challenging pattern recognition problem. One of the main reasons for this is simply the volume of data acquired in each instance of a facial scan. In the GavabDB dataset, for example, each face consists of 10,000-20,000 3D points. More modern laser range scanners such as the multi-modal 3D/2D camera developed by Payne et al. [6] can sample images at the much higher resolution of 500×500 – giving an upper limit of 250,000 3D points!

Multiple projection-based approaches such as ours are one means of reducing the amount of data. There are two types of data reduction in our algrithm: firstly, the point cloud is reduced to a set of 2.5D views; and secondly, the views are replaced by a set of labelled keypoints. The accurate recognition results show quite clearly that we are not discarding significant features during these steps – only the redundant information is discarded.

Future work in this area will investigate two main avenues of enhancement to our algorithm. Firstly, we are interested in the spatial arrangements of keypoints. In the present algorithm, keypoints are matched regardless of their relative positions. Relative proximities and spatial relationships between keypoints, however, must also contain useful information. For example, two very similar but nonetheless different faces may be distinguishable only because the distance between their eyes is slightly different relative to the position of their nose. In the present approach that considers only matches, these two faces could not be distinguished.

The second avenue for future research is to further reduce the quantity of keypoints that are extracted. Presently, there are approximately a hundred keypoints extracted per 2.5D image. For matching purposes, however, many of those keypoints could be discarded. The main question is: which ones?

To conclude, we have presented and evaluated a new approach to 3D face recognition based on voted keypoint matching across multiple views. We are greatly encouraged by our results and plan to continue research in this exciting area.

6. Acknowledgements

Thanks to A. B. Moreno and A. Sanchez for the time and effort needed to develop the GavabDB dataset of 3D facial scans, and also the anonymous AVSS reviewers for their helpful comments.

References

[1] Moreno A. and Sanchez A.: GavabDB: A 3D Face Database. In Workshop on Biometrics on the Internet, pages 77–85, Vigo (2004) 1, 4

[2] Lowe D.: Distinctive image features from scale-invariant keypoints International Journal of Computer Vision, 60, 2 (2004), pp. 91-110. 1, 2

[3] Abate, A. F., Nappi M., Riccio D. and Sabatino G.: 2D and 3D face recognition: A survey. Pattern Recognition Letters 28 1885–1906 (2007) 2

[4] Quinlan R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA. 5

[5] Moreno A., Sanchez A., Velez J. and Diaz F.: Face recognition using 3D surface-extracted descriptors. In Proc. Irish Machine Vision and Image (IMVIP'03) (2003). 5

[6] Payne A., Dorrington A., Cree M. and Carnegie A.: Characterizing an image intensifier in a full-field image ranging system. IEEE Sens. J. 8 17631770, (2008). 6