Video, Vision and Graphics (2003)

Coding 3D facial models for mugshot applications

John Robinson and Justen Hyde

Department of Electronics, University of York, York, United Kingdom

Abstract

Three-dimensional information about a human face may have some correlation with the colour information present in its flat texture image. In order to maximise the available information for human identification of faces, a variety of coding schemes based on Binary Tree Predictive Coding 5 (BTPC5) are proposed and evaluated against similar schemes applied to the JPEG coder. The results of these schemes are presented quantatively with some discussion of the subjective results.

1. Introduction

Three-dimensional mugshots – that is, facial images represented with colour and depth – may be more reliable for identity checks by humans than conventional 2D photos which lack much of the shape information used by humans in face recognition[1]. The extent to which implied depth information is included in sketch drawings by shading, and the common use of three-quarters views in portraiture supports this theory. Work aimed towards the computerised generation of cartoons of human faces has also shown a reliance on the preservation of shape from shading information[2,3]. With a 3D head model and appropriate rendering and manipulation tools, a user can obtain full face, profile and three-quarters views or any other angle desired. Arguably, this addition of explicit depth information provides cues to expose impersonation. We are interested in assessing the effectiveness of 3D mugshots, and in developing efficient methods for their display (e.g., animation of head movement requiring no direct user control of view – essentially providing a video of a moving head). Other work has attempted to reconstruct three dimensional data from two dimensional images[4], which can be considered (albeit indirectly) a method of coding. This approach assumes that no explicit three dimensional data is available at the source. In this paper, however, we report a method for coding such 3D pictures efficiently. We are aiming to represent the colour and geometry of the face in very few bits, so that such information can be encoded on printed media. Ultimately we seek to represent all the information for the generation of pictures like those in figure 1 in a two-dimensional glyph code. We describe our approach to this problem together with promising initial results.

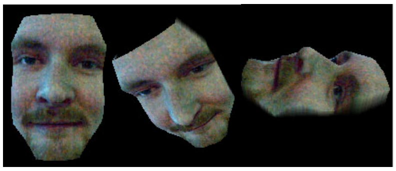

Figure 1: *Multiple views of a face can be obtained with no extra transmission of information*

2. Approach

2.1 Simplification of model – 3d mesh to depthmap

Coding a full three dimensional model of a head involves the manipulation of a huge amount of data. However, the face only occupies one side of the head and contains most of the recognition information. This allows us to treat 3D face coding as a 2.5D problem. The three dimensional model of the face can be reconstructed from depth information only – i.e., by placing the camera directly in front of the model, and taking a snapshot of the distance back from the camera plane at each point on the model, a "depthmap" is created which can be later used to recover a 3D model of the captured projection. This results in a four component image of a face.

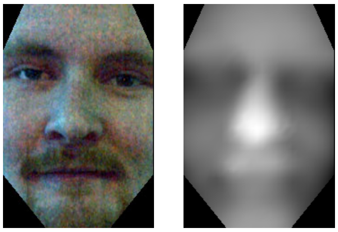

Figure 2: *Colour and depth maps of the full face projection of a 3D facial model*

2.2 Coding

The 2.5D image has four components - red, green, blue and depth. Our approach is to exploit dependencies between the channels to reduce the amount of data to be coded. Almost all picture coding schemes apply a colour transform which serves to reduce the correlation between components. Most often this is simply a conversion from RGB to luminance and two chrominance channels, and usually, after transformation, the components are coded independently. We, however, propose to use "Binary Tree Predictive Coding (BTPC)"[5,6], because it provides two-level of inter-component coding that can be extended naturally from the conventional three colour channels to four channels. In BTPC the components are first transformed with the statistically-optimal Karhunen-Loeve Transform (KLT). The highest-energy component is coded first, then successive components are coded relative to the previous ones. This is possible in BTPC because it is a predictive scheme: a poor prediction in an early component can be modified to a better prediction for later components if they have the same local shape properties. In the experiments below we compare this approach with a JPEG-derived alternative which also uses the KLT for pre-processing but does not include the second stage of dependent component coding.

The KLT can also be applied to an ensemble of head images where the feature vector for decorrelation is not the four-component pel, but the whole image (i.e. a 4xNxM-dimensional vector for images of N rows and M columns). This is similar to the Principal Components Analysis approach that has been widely applied for automatic face detection and recognition[7,8]. We have not adopted a full KLT at the image level because, as yet, our training and test sets are very small, however, we are able to remove some of the correlation between faces, simply by subtracting the mean head image from particular inputs before coding. Our experiments below test the different alternatives for this.

3. Method

3.1 Data capture

The data set used has been captured using a 3-d camera which records both texture data (using a standard digital camera) and geometric information (using a stereo vision system enhanced by light pattern projection). The data was captured by Tom Heseltine of the University of York Computer Science Department. The camera records a three dimensional image of the subject in the same way as a normal camera records a two dimensional image (i.e., the subject stands before the camera for a second whilst a photograph is taken). Whilst this is less accurate than scanning the person, it is also much less time consuming and intrusive.

3.2 Coding schemes applied

Figures 3 and 4 show the four different types of coding we have investigated. In each case the KLT is used to preprocess the colour and depth information and in each case BTPC is used as the final coding engine. The KLT converts four components of colour and depth into four channels, the most significant of which is quantised most accurately, and the least significant of which is quantised the most coarsely. The KLT is integrated into BTPC, but has been adapted for the four-channel situation. We have also used similar structures with JPEG as the coder, and in these cases have used a separate KLT stage.

3.3 The four coder configurations

To test whether the inclusion of depth information in the KLT process significantly alters the distribution of colour information in the transformed channels, we performed tests where an unmodified depth map was coded alongside KLT transformed colour components, where the least energy colour component is subject to the most extreme quantization. Test coders 1 and 3 use this method of depth information inclusion. Test coders 2 and 4 treat the depth channel no differently to the colour channels.

Features such as eyes are regions of high interest, but their colour information may be swamped by the less significant but more common flesh tones. To retain this information, difference from mean depth and colour images are

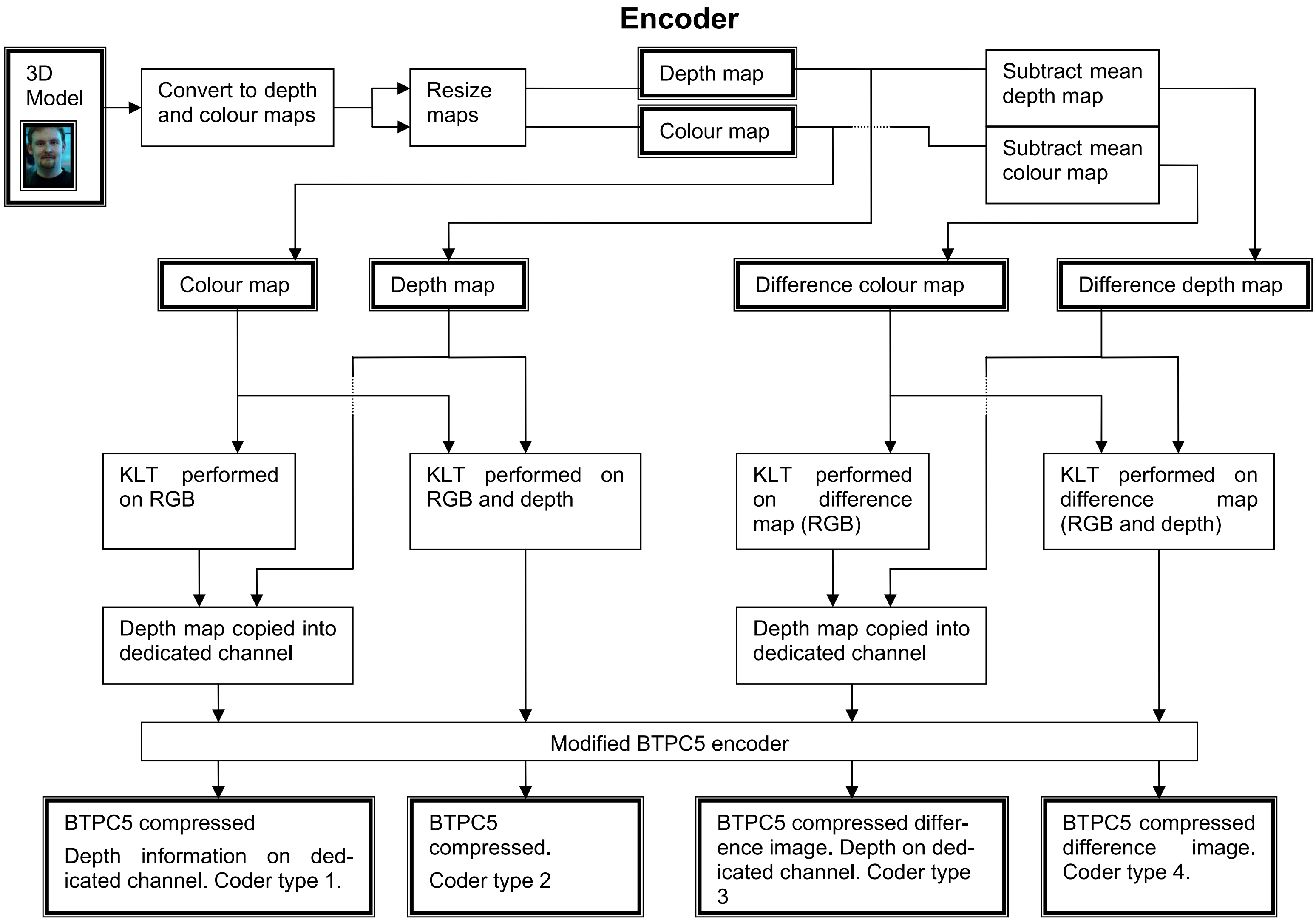
Encoder
3D Model
Convert to depth and colour maps
Resize maps
Depth map
Colour map
Subtract mean depth map
Subtract mean colour map
Colour map
Depth map
Difference colour map
Difference depth map
KLT performed on RGB
KLT performed on RGB and depth
KLT performed on difference map (RGB)
KLT performed on difference map (RGB and depth)
Depth map copied into dedicated channel
Depth map copied into dedicated channel
Modified BTPC5 encoder
BTPC5 compressed Depth information on dedicated channel. Coder type 1.
BTPC5 compressed. Coder type 2
BTPC5 compressed difference image. Depth on dedicated channel. Coder type 3
BTPC5 compressed difference image. Coder type 4.

Figure 3: *Schematic diagram for four encoder alternatives*

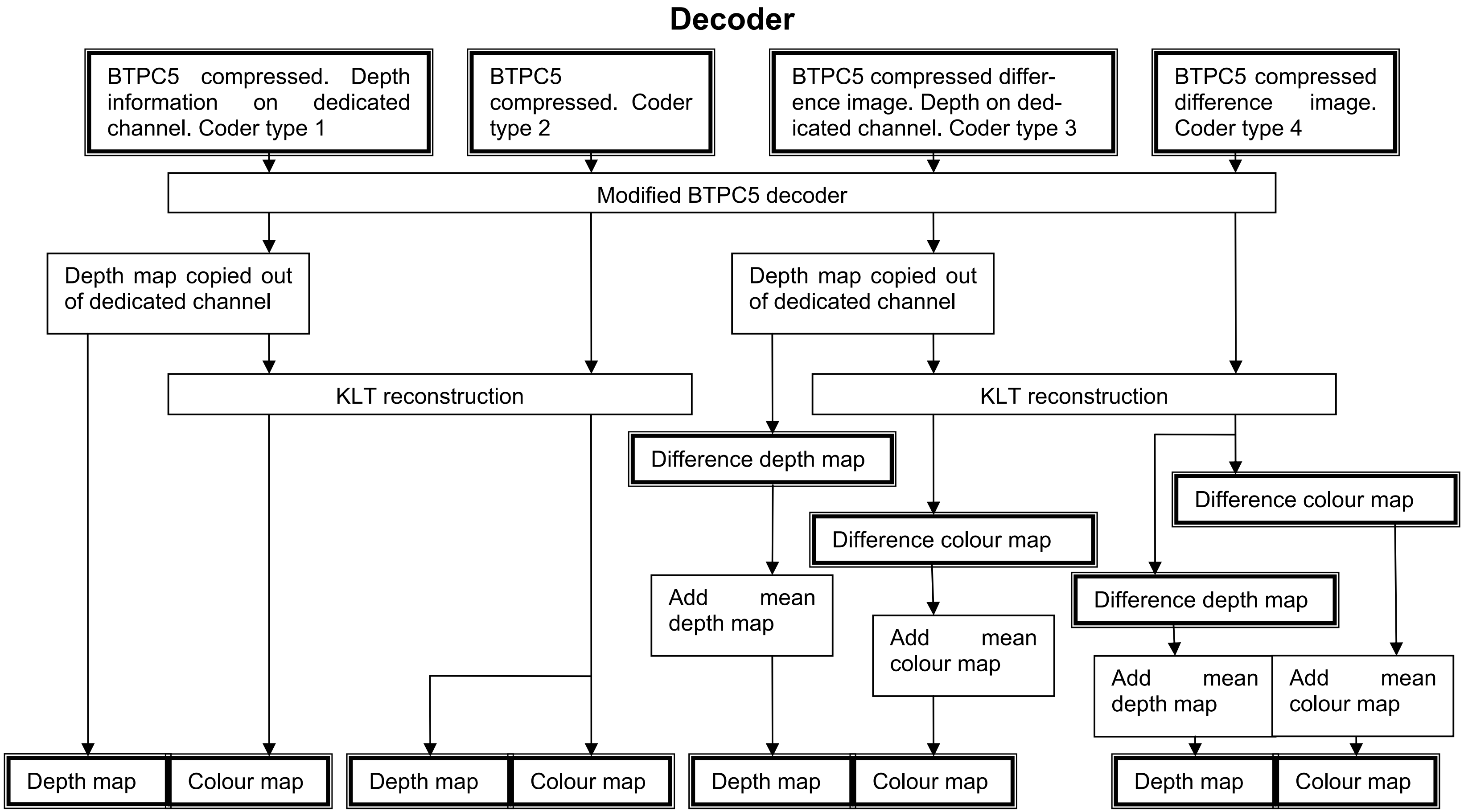
Decoder
BTPC5 compressed. Depth information on dedicated channel. Coder type 1
BTPC5 compressed. Coder type 2
BTPC5 compressed difference image. Depth on dedicated channel. Coder type 3
BTPC5 compressed difference image. Coder type 4
Modified BTPC5 decoder
Depth map copied out of dedicated channel
Depth map copied out of dedicated channel
KLT reconstruction
KLT reconstruction
Difference depth map
Difference colour map
Difference colour map
Difference depth map
Add mean depth map
Add mean colour map
Add mean depth map
Add mean colour map
Depth map
Colour map
Depth map
Colour map
Depth map
Colour map
Depth map
Colour map

Figure 4: *Schematic diagram for four decoder alternatives*

generated and coded rather than the actual images. Coders 3 and 4 use difference images, whilst 1 and 2 code the picture directly.

Coder type	1	2	3	4
Dedicated depth channel	✓	✗	✓	✗
Difference images	✗	✗	✓	✓

Images of three sizes have been tested: Full size (250, 168), quarter size (125, 84) and sixteenth size (62, 42). In the results section we concentrate on the first of these.

4. Evaluation

As with all image coding schemes, quantitative measures (MSE or PSNR) do not necessarily correspond to subjective test results. This is especially true if the system is envisaged as ultimately providing an extra tool for human identity recognition applications. However, quantitative analysis allows us to make comparisons across a large number of images and qualities and we therefore present summary results first.

4.1 Quantative Evaluation

The success of compression for each image can be evaluated based mean square error (MSE) (or Peak Signal-to-Noise Ratio (PSNR)) and bits per pixel (BPP) (or coded file size or compression ratio). The file size of the coded images, and hence the bit rate, must be as low as possible to allow the use of the system on printed media. MSE can be calculated by summing the individual MSEs for the depth and texture images, and should also be as low as possible. The source images for the following graphs can be found in section 4.2.

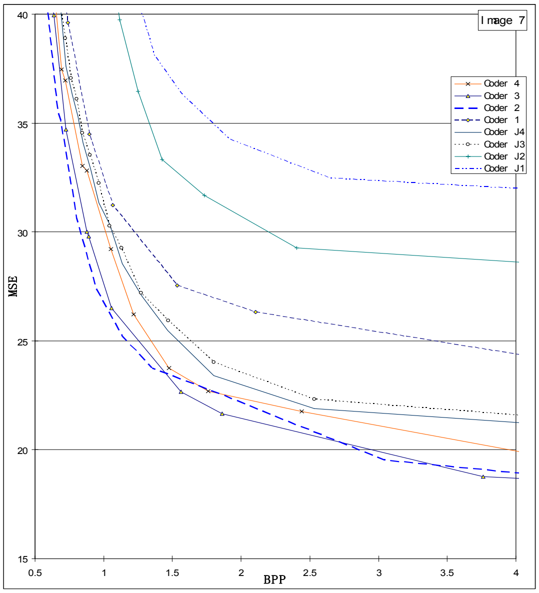

Figure 5 *Performance figures for the eight coders on input face 7.*

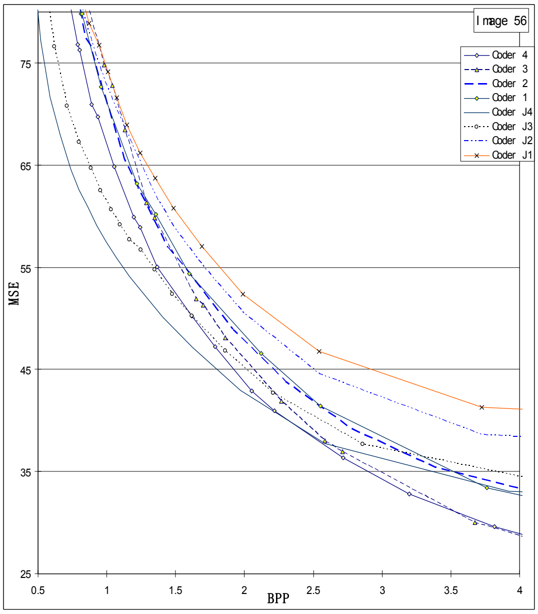

Figure 6 *Performance figures for the eight coders on input face 56.*

Figures 5 and 6 show coders J1 and J2 (the JPEG versions of coders 1 and 2) performing consistently poorly, though all of the other schemes have large variation on their relative performance. However, when the mean of a large number of tests is inspected, it becomes clear that, in general, coders 1 and 3 also perform very poorly. This implies that the use of difference images appreciably improves coding for JPEG, but has less significance for BTPC. Conversely, the independent coding of depth has little effect for JPEG, but is significantly worse than integrated coding of depth and colour in BTPC.

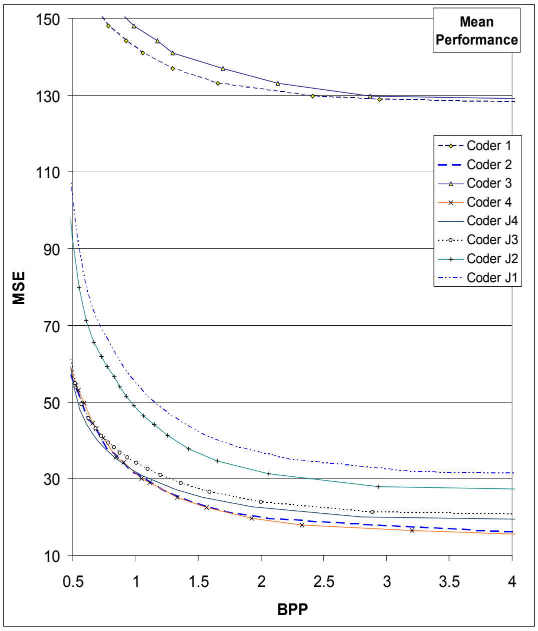

Figure 7. *Mean performance over entire test set.*

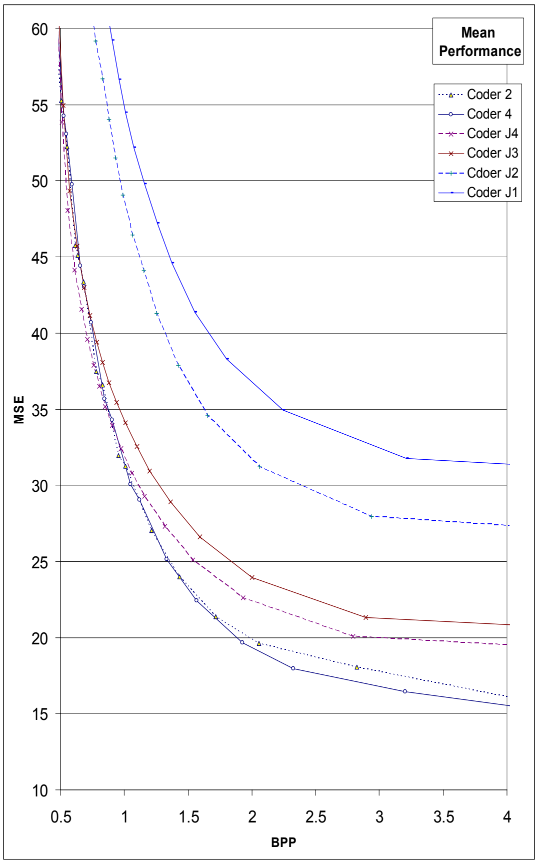

Figure 8. *Mean performance over entire test set. Expanded view of MSEs below 60.*

4.2 Subjective Evaluation

Ultimately, the system must be evaluated against file size and human recognisability rather than MSE, which is itself a poor estimator of the perceptual degradation present in a lossily coded image. This is a difficult and time consuming evaluation criterion to generate data for, and so has not been explored at this time. However, we show in figures 9 and 10 some examples of decoded pictures for the eight alternative approaches coding to approximately 1.5 BPP.

In figure 9, the poor illumination in the source image has caused much of the colour information to be lost – the eyes, for example, have almost completely lost their colour. This removes much of the benefit gained in mean subtraction, as there is little distinct colour information to retain. However, the JPEG compressed image – whilst retaining a comparable MSE – is clearly showing more visible signs of distortion then the BTPC5 compressed faces. Image 56 codes relatively inefficiently using the BTPC5 compressor below 2 BPP compared with the JPEG compressor. However, as figure 10 shows, this quantitative error is not duplicated in subjective perception of the output image.

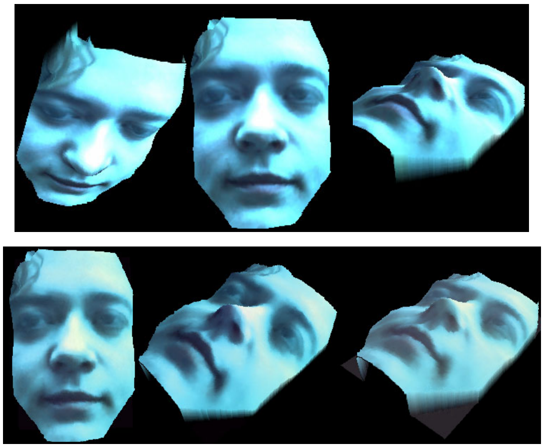

Figure 9 *Top: Three views of uncoded face "Image 7". Bottom, left to right**:** Coder 3, producing 1.53 BPP at 28.84 MSE, Coder 2, producing 1.35 BPP at 23.76 MSE and Coder J4, producing 1.47 BPP at 25.47 MSE*

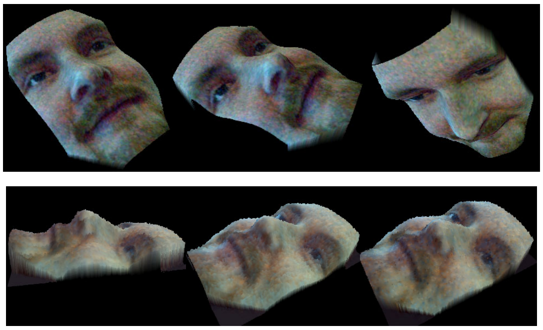

Figure 10 *Top: Three views of uncoded face "Image 56". Bottom, left to right, Coder J4 (1.40 BPP,50.07 MSE) , J3 (1.48 BPP, 52.37 MSE) and 4 (1.61 BPP, 50.25 MSE).*

5. Conclusion

We have demonstrated that both BTPC 5 and JPEG can code three dimensional face images to a reasonable standard, and that the subtraction of the mean colour and depth faces provides a substantial improvement in performance when using the JPEG encoder. The BTPC 5 coder, however, operates most efficiently when the depth channel is not preserved on a dedicated channel in the preprocessing stage, but is operated on by the KLT in the same way as the colour channels. There is a high degree of variability from face image to face image, with different coding schemes proving optimal in different cases. In general, optimally pre-processed JPEG slightly outperforms the best case for BTPC 5 when the bitrate is restricted to less than 0.9 bpp. For bitrates over 0.9 bpp, BTPC 5 outperforms JPEG, with the creation of difference from mean images further increasing performance at bitrates of over 1.7 BPP. However, the subjective result of coding is sometimes found to be at odds with the strict MAE performance. For this reason, extensive subjective testing is required.

The test results presented above have been for images of size 250 x 168. Clearly it is possible to save data if smaller images are adequate for recognition. We have therefore conducted similar tests on smaller pictures. For 62 x 42 face images, the number of bits per pel rises for equivalent quality reconstruction. Nevertheless, it is possible to achieve reasonable reconstruction with BTPC at below 3bpp, even for such small images. This means that is it possible to encode, in a total of less than a thousand bytes, the images shown in figure 11 below.

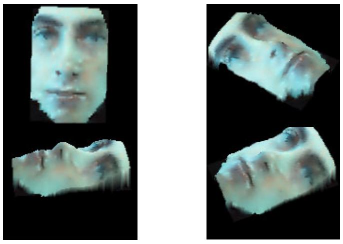

Figure 11. *Four views of a face coded in a total of less than 1kbyte.*

Acknowledgements

Thanks to Tom Heseltine of the University of York Computer Science Department for the use of the 3D data.

References

[1] V Bruce and A Young, In the Eye of the Beholder, *Oxford University Press.* 1998

[2] D E Pearson and J Robinson – "Visual communication at very low data rates", *Proceedings of the IEEE*, **73 (**4), pp 795-812, April 1985

[3] D E Pearson, E Hanna, K Martinez – "Computer-generated cartoons", in *Images and Understanding,* 1990.

[4] T Vetter – "Synthesis of novel views from a single face image", *International Journal of Computer Vision 1998*

[5] J A Robinson, "Efficient General-Purpose Image Compression with Binary Tree Predictive Coding", *IEEE Trans on Image Processing,* **6** (4), April 1997, pp 601-607.

[6] J A Robinson, "Exploiting Local Colour Dependencies in Binary Tree Predictive Coding", *Proceedings of Visual Information Engineering*, VIE 2003, June 2003.

[7] M Kirby, L Sirovich "Application of the Karhunen-Loeve Procedure for the characterization of human faces," *IEEE Trans on Pattern Analysis and Machine Intelligence,* **12** (1), pp 103-108, 1990.

[8] M Turk, A Pentland, "Eigenfaces for recognition", *Journal of Cognitive Neuroscience,* **3** (1), pp 71-86, 1991.