A fast algorithm for constructing image identification

Guoping Qiu

University of Derby, School of Computing & Mathematics
Derby DE22 1GB, United Kingdom

ABSTRACT

In the transmission of remote images, a typical scenario may be one in which a user is searching through an image database which contains a large number of different images, only a small number of which are of interest. In order to select the desired image, the user may have to download (from a remote site) the complete image itself. Instead of downloading the complete image, it would save time and network bandwidth by first sending a much compressed version of the image (its identification image) that gives sufficient information to allow the user visually classify the image and then transmit the complete version of the desired image. In this work, we present a computationally very simple approach to create image identification for viewing classification purpose. The new scheme compresses and transmits only the information in and around the edges of the image and a simple interpolation technique is used to construct the ID image from those compressed features at the user's computer. The technique described herein may be useful in tele-browsing image databases.

Keywords: Tele-browsing image database, Image ID, Image coding, Image interpolation

1. INTRODUCTION

In the transmission of remote images, a typical scenario may be one in which a user is searching through an image database which contains a large number of different images, only a small number of which are of interest. In most situations the user will not be able to tell from its file name whether an image is the desired one. In order to select the desired image, the user may have to download (from a remote site) the complete image itself. One of the major problems in image delivery is the lengthy transmission time. For example, transmitting a picture of 512 x 512 pixels at 8 bits/pixel over a 2400-baud line takes 15 minutes.

Instead of downloading the complete images, it would save time and network bandwidth by first sending a much compressed version of the images (their identification image) that give sufficient visual information to allow the user to classify those desired images and then transmit the complete version of them. In this work, we present a computationally very fast approach to create image identification for viewing classification purpose. The new technique is based on a recently developed image coding algorithm¹ and image interpolation. The new scheme compresses and transmits only the information in and around the edges of the image and a simple interpolation technique is used to construct the ID image from those compressed features at the user's computer.

Computer simulations have been performed and results are presented in this paper. It is shown that the identification images take up about 1.6 % to 4.1% of the original images size, depending on the particular image. Because grey scale image is constructed, it makes visual classification task easier. One of the most important merits of the present technique is that it is computationally very simple. The computational complexity of the new scheme is a fraction of that of those existing techniques³⁻⁵ whilst achieving the same amount of compression; although it should be pointed out that at a similar bit rate, the image quality obtained by the current system is worse than that achievable by those techniques outlined in references 3 - 5. It is expected that further improvement is possible, we will leave this topic for future research. Even though the visual quality achieved by the present technique may not as good as that obtained by the techniques in references 3 - 5, it serves our present purpose, i.e. the ID image allows the user recognise the original image and because it is computationally much simpler, the whole process may be implemented in real time by using simple technologies.

The organisation of the paper is as follow: In section 2, we briefly outline a recently developed fast image coding algorithm termed visual vector quantization (VVQ). In section 3, we combine VVQ and a simple interpolation scheme to construct image identifications. In section 4, we present simulation results.

2. VISUAL VECTOR QUANTIZATION OF IMAGE

Traditional vector quantization⁴ (VQ) for image coding operates in 'pixel space'. The image to be coded is divided into small blocks, and the pixel values within this block form a multi-dimensional vector. It is well known that the encoding process for traditional VQ based image compression schemes is computationally intensive.

Recently, a computationally efficient new image coding system, the visual vector quantization (VVQ)¹ image coding technique has been developed. In VVQ, perceptually important features of the block (horizontal and vertical edge information) are extracted and used to form a 'visual vector'. An image block is coded by quantizing each of these features individually. Thus coding consists of two sets of one-dimensional tests, reducing the computational load significantly. The system codebook is created by training a neural network with the visual vectors as input, and the residual training blocks as the desired output. Block reconstruction is achieved by substituting the appropriate image block from the codebook. In the following, we outline this technique, readers are referred to reference 1 for details.

The codebook is created in two stages, each using data from training image(s). Firstly, the training data is segmented into $m \times n$ blocks. The horizontal and vertical derivatives of the block, D_h and D_v , are then found. This results in a pair of values for each block, i.e. a data set in the two-dimensional 'visual vector' space. A competitive learning algorithm⁶ is used to cluster the D_h values for the training blocks into several classes. For the results presented in this paper, three classes were used corresponding to the directional derivative being negative large (NL), small (S) or positive large (PL). This process was repeated for the vertical derivative D_v . Since three classes were defined in each direction, an image block may fall into one of nine different classes. The three co-ordinates for each direction are then stored and used during the coding stage to classify a block into one of the nine classes.

The codebook entry for each block class is found by training a backpropagation neural network⁷ to reproduce the residual block ($m \times n$ pixel values) at its output, when the D_h and D_v co-ordinates for the appropriate class are presented at its input. After the network has been trained to reproduce all nine classes of block, the network outputs for each of the nine classes are taken as the codebook entries. Thus the codebook needed to reconstruct the blocks consists of nine entries, each of which comprises $m \times n$ pixel values. The VVQ image coding system operates as indicated in Fig. 1.

Fig. 1 Block diagram of VVQ image coding system

The directional derivatives of the block, D_h and D_v , are found and used to classify the block into one of the nine classes. This classification is simple since only six comparisons are required (three for D_h and three for D_v). A class index denoting the class of the block, and the quantized mean value of the block, constitute the compressed data for the block. Reconstruction of the block is achieved by substituting the appropriate image block from the codebook, which resides at the reconstruction stage of the system, and adding the quantized block mean to each pixel value. Note that the codebook is not used during the image coding stage.

3. IMAGE ID CONSTRUCTION BASED ON VVQ AND INTERPOLATION

From section 2, it is seen that image coding by VVQ is based on classifying the directional derivatives of the image blocks. The magnitude of the directional derivatives give indications of whether or not the block contains edges. A large magnitude value of the directional derivative indicates the block contains edge information, conversely, a small value of the directional derivative indicates the block contains no significant edge. Fig 2 below shows a typical nine codewords VVQ codebook and its corresponding directional derivative classes. It can be seen that when both D_h and D_v are small, the codewords pattern (class 5) contains no significant edge.

Fig. 2 Directional derivative classes and their corresponding codewords

It is well known that edges, although consist of a small part of the image, are the most important visual information contents². In this study, our main purpose is to construct visually recognisable images with as little information as possible. This has lead to the development of the following image coding technique: If a block contains no significant edges, i.e., both D_h and D_v are small (it belongs to class 5 in Fig. 2), the block is not transmitted. If a block contains edges (it belongs to one of the eight classes, excluding class 5 in Fig. 2), the block is coded using VVQ and transmitted to the user's computer. In the user's computer, those image pixels in the positions where the original pixels have not been transmitted are generated using a simple interpolation scheme. The interpolation scheme we developed is explained as follow. The process is performed in three steps. Firstly, a horizontal interpolation is performed line by line. Let pixels $g(i, j)$ and $g(i, j + K)$ be transmitted pixel values and the K pixels between these two pixels have not been transmitted. In determining the pixel values of these K pixels, the following four cases will happen. (a) If $j < 0$ and $j + K > C_{\max}$, where $C_{\max}$ is the maximum column number of the image, this means the entire line of pixels have not been transmitted, then do nothing. (b) If $j < 0$ and $j + K < C_{\max}$, this means the first K pixels in this line have not been transmitted, then the K untransmitted pixels take the value of $g(i, j + K)$. (c) If $j > 0$ and $j + K < C_{\max}$, then the K untransmitted pixels take the average of $g(i, j)$ and $g(i, j + K)$. (d) If $j > 0$ and $j + K > C_{\max}$, this means the last K pixels in this line have not been transmitted, then the K untransmitted pixels take the value of $g(i, j)$. When the process is completed for all untransmitted pixels, this will create an interpolated image $fh(x, y)$. Secondly, the same process is applied vertically. Let pixels $g(m, n)$ and $g(m + L, n)$ be transmitted pixel values and the L pixels between these two pixels have not been transmitted. In determining the pixel values of these L pixels, again the following four cases will happen. (a) If $n < 0$ and $m + L > R_{\max}$, where $R_{\max}$ is the maximum row number of the image, this means the entire column of the pixels have not been transmitted, then do nothing. (b) If $n < 0$ and $m + L < R_{\max}$, this means the first L pixels have not been transmitted, then the L untransmitted pixels take the value of $g(m + L, n)$. (c) If $n > 0$ and $m + L < R_{\max}$, then the L untransmitted pixels take the average of $g(m, n)$ and $g(m + L, n)$. (d) If $n > 0$ and $m + L > R_{\max}$, this means the last L pixels in this column have not been transmitted, then the L untransmitted pixels take the value of $g(m, n)$. When the process is completed for all untransmitted pixels, this will create an interpolated image $fv(x, y)$. The

final ID image $f(x, y)$ is formed by combining these two vertically and horizontally interpolated images as $f(x, y) = \text{average}[fh(x, y), fv(x, y)]$.

4. EXAMPLES

To demonstrate the usefulness of the technique, we present some example images. The block size used in the experiment is 4 x 4 pixels. A nine codewords VVQ codebook was designed using a 512 x 512, 8-bit per pixel grey scale image (F16). Therefore, when a block is coded and transmitted, a 4 bit codeword index is needed (one bit is used to indicate the block is transmitted and 3 bits are used to identify the appropriate codeword for the block). The block means were coded using a 6-bit scalar quantizer. Hence a total of 10 bits are needed to code a transmitted block. When a block is not coded and transmitted, only 1 bit is needed to indicate its statue. Table 1 lists the results of twelve 512 x 512, 8-bit per pixel grey images. It is seen that compression ratios ranging from 23.8 : 1 to 61.1 : 1 have been achieved. Fig. 3 shows the original image, the coded and transmitted version and the ID images of the 12 images list in table 1. It is seen that in all cases, the ID images provide sufficient visual information that allow the user easily recognise the original images.

TABLE 1. IMAGE ID RESULTS

Image	No of transmitted blocks	No of untransmitted blocks	Total number of bits	Compression ratio
F16	3551	12833	48343	43.4 : 1
Yacht	5118	11266	62446	33.6 : 1
Peppers	2609	13755	39845	52.6 : 1
Lena	3334	13050	46390	45.2 : 1
Sailboat	4888	11496	60376	34.7 : 1
Girl	1191	14393	34303	61.1 : 1
Kids	4214	12170	54310	38.6 : 1
Presenter	2527	13857	39127	53.6:1
Cablecar	4303	12081	55111	38.0 : 1
Soccer	6766	9618	77278	27.1 : 1
Snooker	2237	14147	36517	57.4 : 1
Baboon	7964	8420	88060	23.8 : 1

Original F16 image

Original Yacht image

Encoded F16 image

Encoded Yacht image

F16 ID image

Yacht ID image

Fig. 3 Simulation results for image ID construction (continued)

Original Peppers image

Original Lena image

Encoded Peppers image

Encoded Lena image

Peppers ID image

Lena ID image

Fig. 3 Simulation results for image ID construction (continued)

Original Sailboat image

Original Snooker image

Encode Sailboat image

Encoded Snooker image

Sailboat ID image

Snooker ID image

Fig. 3 Simulation results for image ID construction (continued)

Original Girl image

Original Kids image

Encode girl image

Encoded Kids image

Girl ID image

Kids ID image

Fig. 3 Simulation results for image ID construction (continued)

Original Presenter image

Original Cablecar image

Encoded Presenter image

Encoded Cablecar image

Presenter ID image

Cablecar ID image

Fig. 3 Simulation results for image ID construction (continued)

Original Soccer image

Original Baboon image

Encoded Soccer image

Encoded Baboon image

Soccer ID image

Baboon ID image

Fig. 3 Simulation results for image ID construction

5. CONCLUDING REMARKS

In this paper, a computationally efficient approach to the construction of image identification for viewing classification purpose has been presented. This techniques may be useful in tele-browsing image databases. The merits of the new technique are its simplicity and effectiveness. Experimental results have been presented to demonstrate the effectiveness of the scheme. It is also seen, that because the interpolation scheme used in this paper is a very simple one, the quality the ID images generated by such a scheme exhibits obvious visually annoying effects. It is envisaged that if more sophisticated reconstruction techniques are employed, the visual quality of the ID image can be significantly improved, the author is currently investigating this subject.

6. REFERENCES

1. G. Qiu, M. R. Varley and T. J. Terrell, "Image coding based on visual vector quantization", to appear in IEE 5th International Conference on Image Processing and Its applications, Heriot-Watt University, Edinburgh, UK: 3-6 July 1995
2. A. N. Netravali and J. O. Limb, "Picture coding: A review", Proceedings of IEEE, Vol. 68, No. 3, pp. 366-406, 1980
3. R. J. Clarke, **Transform coding of images**, Academic Press , Inc., 1990
4. A. Gersho and R. M. Gray, **Vector Quantization and Signal Compression**, Boston: Kluwer, 1992
5. M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image coding using wavelet transform", IEEE Trans. on Image Processing, April 1992
6. T. Kohonen, **Self-organization and associative memory**, Second Edition, Springer-Verlag, Berlin 1988
7. D. E. Rumelhart, G. E.. Hinton and R. J. Williams, "Learning internal representations by error propagation", in Parallel Distributed Processing, Vol. 1, Chapter 8, D E. Rumelhart and J. L. McClelland (Eds.), Cambridge, MA, MIT Press, 1986