International Workshop on Systems, Signals and Image Processing
Poznań, POLAND
September 13-15, 2004

IWSSIP'04
ambient multimedia

A Novel Image Coding Algorithm using Ant Colony System Vector Quantization

Nasir Rajpoot

Department of Computer Science
University of Warwick
United Kingdom

email: nasir@dcs.warwick.ac.uk

Arshad Hussain, Usman Ali, Kashif Saleem, Mahmud Qureshi

Faculty of Computer Engineering
GIK Institute of Engineering Sciences & Technology
Pakistan

emails: ahchanna@yahoo.com
usman_gikian_abbott@hotmail.com,
kashif_gikian@hotmail.com, mahmudaq@hotmail.com

Abstract – Ant colony system (ACS) is a combinatorial optimization method motivated by the behaviour of real ants. In this paper, we present a novel image coding method based on ACS vector quantization of groups of wavelet coefficients. The generation of codebook using ACS is facilitated by representing the coefficient vectors in a bidirectional graph, followed by defining a suitable mechanism of depositing pheromone on the edges of graph. Experimental results show that the quantization of zerotree vectors using ACS outperforms, in most cases, its traditionally used Linde-Buzo-Gray (LBG) counterpart.

Keywords: Image coding, wavelet transform, vector quantization, ant colony systems.

1. INTRODUCTION

In recent years, wavelets have been used extensively in image coding schemes mainly due to their efficient energy compaction. A range of quantization methods have been employed in order to quantize the wavelet coefficients. Most of the methods can be classified as belonging to either of the two broad range of quantization schemes: scalar quantization (SQ) and vector quantization (VQ). In case of VQ, the quantizer encodes a number of image pixels or transform coefficients – the so-called *vector* – using a single code symbol. Thus an obvious advantage of VQ over SQ is its coding efficiency in terms of the bit rate. However, there is a little caveat: the coder (decoder) requires to have a codebook to refer to at the time of coding (decoding) vectors into (from) a code symbol, an index to the codebook. The generalized Lloyd algorithm (GLA) or LBG algorithm [1] is a clustering technique which has traditionally been used for codebook generation from known probability distributions. A major disadvantage of this algorithm is that it can get stuck in local minima in high-dimensional vector spaces. In this paper, we propose to use ant colony system (ACS) for the vector quantizer part of our wavelet image coding algorithm.

The ACS is a metaheuristic motivated by the behaviour observed in colonies of real ants for finding the shortest path from a food source to their nest. Ants can find the shortest path because they deposit pheromones on paths they visit and they follow paths with higher pheromone trails. In [2], the ACS was proposed to solve the traveling salesman problem (TSP) by generating successively shorter feasible tours using information accumulated in the form of a pheromone trail deposited on the edges of the TSP graph.

It can be argued that the quantizer block of image coding algorithms based on Shapiro's zerotree idea, such as [3] and [4], is a particular form of VQ with zerotrees of wavelet coefficients forming the vectors to be quantized. The success of wavelet zerotree image coding methods is due to the fact that at the root of such zerotrees are coefficients from coarse resolution subbands. This can potentially lead to only a few bits required to encode all the coefficients in a zerotree at the cost of relatively little distortion. Motivated by this observation, we propose to use a zerotree vector which replaces the vector made from naive grouping of wavelet coefficients (such as taking 4×4 neighbouring coefficients etc).

The remainder of this paper is organised as follows. In the next section, a brief introduction to the ACS is provided for the sake of completeness. The application of ACS metaheuristic to VQ is described in Section 3. Experimental results are presented and discussed in Section 4. The paper ends with some concluding remarks and future directions.

2. ANT COLONY SYSTEM (ACS)

Real ants are capable of finding the shortest path from a food source to the nest without using visual cues. Also, they are capable of adapting to changes in the environment, for example finding a new shortest path once the old one is no longer feasible due to a new obstacle. It is interesting to note that the ants who choose, even by chance, a shorter path around the obstacle will more rapidly reconstitute the interrupted pheromone trail as compared to the ones who choose the longer path. Thus, the shorter path will receive a greater amount of pheromone per time unit and in turn a larger number of ants will choose the shorter path. Due to this positive feedback (autocatalytic) process, all the ants will rapidly choose the shorter path.

Dorgio and Gambardella [2] were perhaps first to suggest the use of ACS for combinatorial optimization. They used the ACS in order to solve the traveling salesman problem (TSP). In their solution, a set of cooperating agents called *ants* are positioned at a starting city. The ants then cooperate with eachother to find good solutions to region growing. They do so by using an indirect form of communication mediated by *pheromone* they deposit on the edges of the path while building solutions. The pheromone trail is modified both locally and globally [2]:

ACS Local Updating Rule: While building a solution, the ants visit edges and change their pheromone level by applying the local updating rule:

$$\tau(r,s) \longleftarrow (1-\rho)\cdot\tau(r,s) + \rho\cdot\Delta\tau(r,s) \qquad (1)$$

where $\tau(r,s)$ denotes the pheromone level of the edge between nodes r and s, and $0 < \rho < 1$ is a parameter related to the evaporation time for pheromone.

ACS Global Updating Rule: Global updating is performed after all ants have completed their tours. In this case, only the globally best ant is allowed to deposit pheromone:

$$\tau(r,s) \longleftarrow (1-\alpha)\cdot\tau(r,s) + \alpha\cdot\Delta\tau(r,s)$$

where

$$\Delta\tau(r,s) = \begin{cases} {L_{gb}}^{-1} & \text{if } (r,s) \in \text{global-best} \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

where L_{gb} is the distance between the nodes r and s, and α is a parameter related to the evaporation time for pheromone.

3. ACS VECTOR QUANTIZATION

The key to application of ACS for any problem is to represent it as a graph to be searched by many artificial ants [2]. We construct vectors $v_1, v_2, \ldots, v_n$, where n denotes the codebook size, from coefficients belonging to wavelet subbands by grouping coefficients together in a non-overlapping manner. Each of these vectors is represented by a node

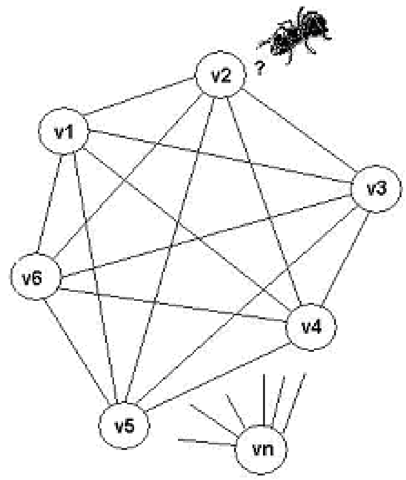


Fig. 1. Graph representation of wavelet subbands

in a fully connected bidirectional graph. The graph representation of wavelet coefficient vectors is shown in Figure 1. Note that all the nodes are connected to eachother with the edges labelled by the distortion between those vectors. Every time an edge is chosen by an ant, its amount of pheromone is changed by applying the local trail updating formula:

$$\tau(r,s) \longleftarrow (1-\alpha)\cdot\tau(r,s) + \alpha\cdot\tau_0$$

where τ_0 denotes the initial pheromone level for all edges.

Once artificial ants have completed their tours, the best ant deposits pheromone on the edges it visited. The amount of pheromone $\tau(r,s)$ deposited on each visited edge (r,s) by the best ant is inversely proportional to the distortion between the vectors joining the edge. The global trail updating formula is:

$$\phi(r,s) \longleftarrow (1-\alpha)\cdot\phi(r,s) + \alpha\cdot\Delta\phi(r,s)$$

where $\Delta\phi(r,s)$ is inverse of the distortion between vectors r and s. The shorter the distortion, the greater the amount of pheromone deposited on edges.

3.1. Region Growing

When the system has generated solution to a group of vectors, it jumps to another vector in the graph. There are a few possibilities for which vector should be selected next, as shown in Figure 2. We pick a vector randomly and all edges of the graph which have pheromone level within the range of a pre-selected threshold θ are grouped together to form one region. The centroids from all the regions form the codebook elements.

3.2. Formation of vectors

Our experiments with vectors using adjacent wavelet coefficients and zerotrees of coarse resolution wavelet coeffi-

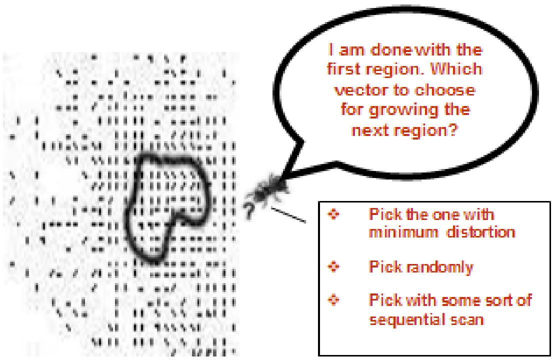


Fig. 2. Options for selecting the next starting vector.

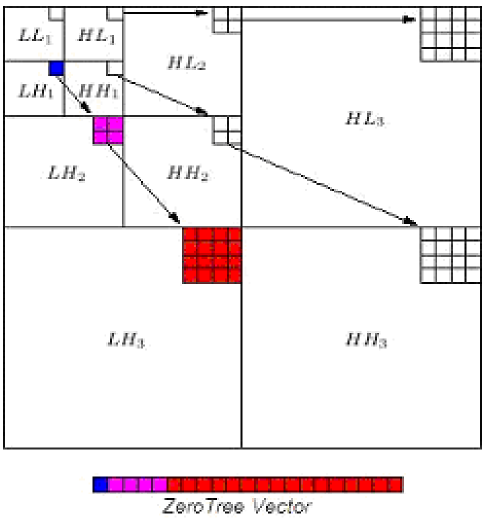


Fig. 3. Formation of a zerotree vector

cients showed that the latter outperformed the former by a wide margin. Zerotree vectors of three orientations – horizontal, vertical, and diagonal – were formed by taking coefficients from the coarsest resolution band of the orientation and following a quadtree of similar orientation coefficients, as shown in Figure 3.

4. EXPERIMENTAL RESULTS & DISCUSSION

The ACS VQ codebook design utilizes zerotree vectors from the wavelet transform of a number of images. Having designed the codebook, the job of the coder is reduced to searching the codebook for entries closest to zerotree vectors for the wavelet transform of a given image.

Experimental results, in terms of visual quality, for three standard images using 3-level Haar transform are shown in Figure 4. Blocky type of artifacts can be observed in Figure 4(d)–(i). The boundaries between pixel regions are clearly visible in the results of quantizing pixel-group vectors with no transform. The occurrence of such artifacts in the results of quantizing zerotree vectors is relatively much lower and can be attributed to the nature of Haar wavelet transform.

Experimental results, in terms of the peak signal-to-noise-ratio (PSNR), for seven images using LBG and ACSVQ are presented in Table 1. These results were obtained for a 3-level wavelet transform using both Daubechies-4 and biorthogonal 9/7-tap filters. Codebooks of two different sizes were designed using first five of the images. The results show the promise carried by the zerotree vector based ACSVQ algorithm employed by our coder. From Table 1, we note that the biorthogonal 9/7-tap filters generally perform better than Daubechies-4 filters. The ACSVQ algorithm for the two filters outperforms the LBG algorithm for most of the images used for training, since it starts from single vectors as regions and gradually increases the regions by adding the most suitable elements available. It can also be observed from the Table that the ACSVQ coder does not perform as well on the unseen images *Barbara* and *Stan*. This may be due to the fact that only five images were used to train the quantizer codebook.

5. CONCLUSIONS

In this paper, a novel image coding algorithm based on a new vector quantization (VQ) method was presented. The VQ employs ant colony system (ACS) method of combinatorial optimization for codebook design, where the vectors are made up of zerotrees of wavelet coefficients. The key to the application of ACS to VQ is to represent the vectors in terms of a graph, and then use an appropriate heuristic defining the edge label between any two nodes in the graph. Experimental results have shown the promise carried by the new VQ algorithm. Our experiments showed that increasing the number of levels of wavelet transform reduces the PSNR value for fixed codebook size, perhaps due to an increased dynamic range of the coefficients. However, gains may be expected when the codebook size is increased. Furthermore, we expect to see increased coder performance with the application of a suitable entropy coder for codebook indices.

6. REFERENCES

[1] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Comm.*, vol. 28, Jan. 1980.

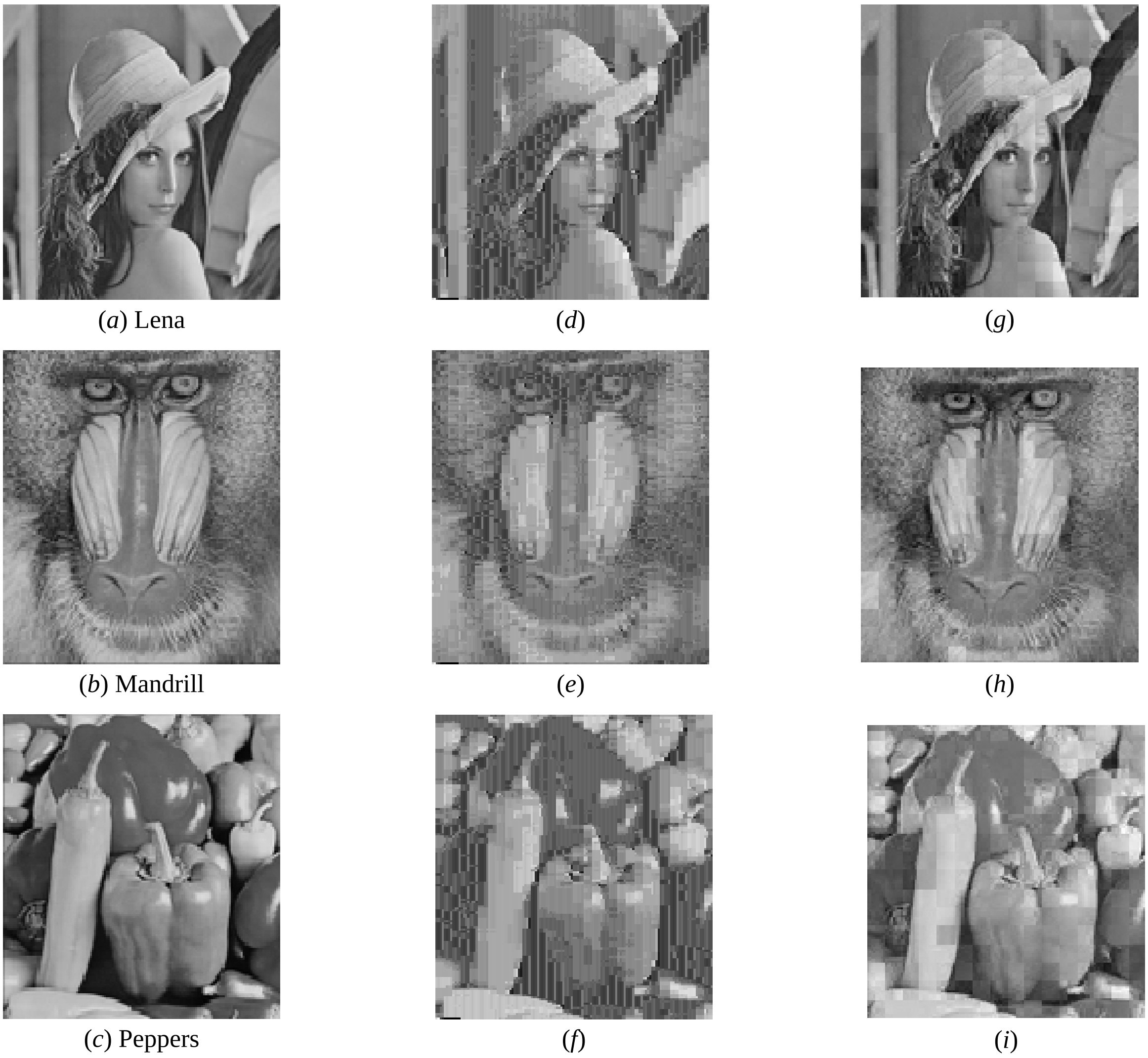
[2] M. Dorgio and L. Gambardella, "Ant colonies for the traveling salesman problem," *BioSystems*, 1997.

[3] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Sig. Proc.*, vol. 41, Dec. 1993.

[4] N.M. Rajpoot, R.G. Wilson, F.G. Meyer, and R.R. Coifman, "Adaptive wavelet packet basis selection for zerotree image coding," *IEEE Trans. on Im. Proc.*, vol. 12, Dec. 2003.

Image	LBG				ACSVQ			
	Daubechies-4		Biorthogonal-9/7		Daubechies-4		Biorthogonal-9/7	
	$n = 2^8$	$n = 2^{10}$	$n = 2^8$	$n = 2^{10}$	$n = 2^8$	$n = 2^{10}$	$n = 2^8$	$n = 2^{10}$
Lena	24.7	28.2	26.6	28.9	24.6	29.2	25.0	**30.5**
Mandrill	26.0	28.4	26.5	**28.6**	23.5	27.4	24.2	28.3
F-16	23.2	25.9	25.2	27.4	24.6	30.0	25.6	**30.8**
Peppers	19.0	25.3	19.7	22.1	22.0	21.3	21.8	**27.5**
Lake	22.0	25.4	23.6	26.3	22.1	29.1	22.3	**30.0**
Barbara	23.3	24.1	26.8	**27.3**	22.3	24.8	24.3	26.2
Stan	19.4	19.3	21.1	19.2	18.5	20.1	19.4	**21.3**

Table 1. PSNR (in dB) results for the zerotree coders using 3-level wavelet transform; n denotes the codebook size.



(*a*) Lena (*d*) (*g*)

(*b*) Mandrill (*e*) (*h*)

(*c*) Peppers (*f*) (*i*)

Fig. 4. ACSVQ coding results using 3-level Haar transform with $n = 2^{11}$.
(*a*)–(*c*) Original images; (*d*)–(*f*) ACSVQ with pixel-group vectors; (*g*)–(*i*) ACSVQ with zerotree vectors.