FINGERPRINT COMPRESSION USING A PIECEWISE-UNIFORM PYRAMID LATTICE VECTOR QUANTIZATION

Shohreh Kasaei and Mohamed Deriche

Signal Processing Research Centre (SPRC), QUT
GPO box 2434, Brisbane, Q. 4001, Australia
skasaei@lux.sprc.qut.edu.au

ABSTRACT

A new compression algorithm for fingerprint images is introduced. Using *Lattice Vector Quantization* (LVQ), a technique for determining the largest radius of the Lattice and its scaling factor is presented. The design is based on obtaining the smallest possible *Expected Total Distortion* (ETD) measure, using a given bit budget, while using the smallest codebook size. In the proposed *Piecewise-Uniform Pyramid LVQ*, the *wedge* problem encountered with the Pyramidal Lattice point shells is resolved. At very low bit rates, for the coefficients with high-frequency content, the *Positive-Negative Mean* (PNM) method is proposed to improve the resolution of the reconstructed image. The proposed algorithm results in a high compression ratio and a high reconstructed image quality with a low computational load compared to other existing algorithms.

1. INTRODUCTION

Fingerprints have been used as unique identifiers of individuals for a very long time. The increasing amount of fingerprints collected today by government agencies (such as the Police) has created an enormous problem in *transmission, storage*, and *automated analysis*. Although there are many image compression techniques currently available, there still exists a need to develop faster and robust data compression algorithms adapted to fingerprints using emerging signal analysis techniques.

Since fingerprint ridges are not necessarily continuous across the impression due to *minutiae* (ridges endings and bifurcations), the information used for matching one fingerprint to another resides in these fine details and their relationships. Consequently, the details have to be retained in *comp/decomp* algorithms. In order to achieve a high compression ratio while retaining these fine details, the *Wavelet Packet Transform* (WPT) associated with a proposed fixed decomposition structure tailored to fingerprint images is considered.

Quantization is the mapping of a large set of possible inputs into a smaller set of possible outputs, which can be implemented in either scalar or vector versions. Shannon showed that block coding of a discrete memoryless source asymptotically approaches the performance promised by the rate distortion function for a given source. Later, Zador proved that Vector Quantization (VQ) can yield even a smaller average *Mean Square Error* (MSE), per dimension, than scalar quantization. Furthermore, Gersho conjectured that LVQ is an optimal high-resolution entropy-constrained vector quantizer.

However, LVQ is a multidimensional generalization of uniform quantizers which produces minimum distortion for an input with *Uniform* distribution. Unfortunately, the distribution of the WT coefficients of natural images is found to approximate the generalized *Gaussian* law. In order to be able to take advantages of LVQ's properties and its fast implementation, while considering the i.i.d. Non-Uniform distribution of input sources, a piecewise-uniform companding approach to the LVQ is proposed in this paper. The proposed algorithm quantizes almost all of the source vectors without being projected to the outermost sell of the Lattice, while it properly maintains the smallest possible codebook size. It also overcomes the wedge-region problem, encountered with Pyramidal shape LVQ, which are two major drawbacks of the PLVQ algorithm proposed by Barlaud [7]. The proposed algorithm can also handle all kinds of Lattices, not only cubic Lattices, as opposed to the algorithms in [9] and [8]. In the proposed algorithm, no predetermined assumption on the values of the Lattice parameters is made and no training and multi-quantizing is required, as opposed to [12]. A method for determining the largest Lattice radius (while using the smallest appropriate codebook size) is presented. For each subimage, the concentric Lattices are *truncated* and *scaled* in order to achieve the best rate-distortion function, adapted to the Probability Density Function (PDF) of that subimage. At very low bit rates, for the coefficients with high-frequency content, the PNM method is proposed to improve the quality of the reconstructed image. Additionally, for the fourth WT level, for the coefficients with low-frequency content, a *lossless predictive* compression scheme is used. The overall algorithm results in a high compression ratio and a high reconstructed image quality with a low computational cost compared to other existing algorithms. The performance of the proposed algorithm is compared to that of other compression techniques.

2. WAVELET TRANSFORM DECOMPOSITION

Due to the particular nature of fingerprints and the necessity of retaining ridge details, the WT is considered in this work. The Wavelet decomposition is a powerful tool in image coding because of: its decorrelating effects on image pixels, the concentration of energy in a few coefficients, its multiresolution framework, and its flexible frequency split-

ting [5]. The resulting individual components can hence be processed using different algorithms adding flexibility to the coding process to obtain near optimal results in compression.

In *WT* fields, *"Wavelet Packets"* are functions that give rise to an *orthonormal basis* which can be used to improve the time-frequency localisation of signals. Here, the WPT associated with a special subband decomposition structure, tailored to fingerprints, is applied to the mean-removed input source image. The "essential" coefficients of each subimage are normalized to have *zero mean* and *unit variance*; this leads to PDFs with *narrower* main lobe. As a result, the quantization procedure becomes more manageable, requiring less bits and a *smaller* Lattice codebook size. The coefficients within each column in the subimages, are selected from non-neighbouring pixels. Experiments showed that with these vectors, the best LVQ result is obtained. The bit allocation and the corresponding distortion of each subimage is determined, taking into account properties of the *Human Visual System* (HVS) and the normalized i.i.d. of generalized Gaussian law [8], [6]. For more details on wavelet decomposition structure, wavelet family selection, main coefficients selection, optimal bit allocation, and vector composition, see [11] and [10].

3. PLVQ AND PREVIOUS WORK

Quantization is the mapping of a large set of possible inputs into a smaller set of possible outputs. To avoid using the well-known *LBG* method, which is computationally expensive and results in blur artifacts at low bit rates, we use the Lattice algorithm. The codebook size $|c| = 2^{nR}$, for n-D vectors with R bpp, is not achievable using *LBG*, but is easily achievable using LVQ.

A Lattice Λ_n in $\Re_n$, which is a multidimensional gereralization of uniform quantization, is composed of all integral combinations of a set of linearly independent vectors that span the space. Conway and Sloane determined the best known Lattices for several dimensions, as well as fast quantization and indexing algorithms [2], [3]. Using their fast algorithms, one does not need to design and transmit the codebook.

Two critically important issues with *LVQ* concern the *truncation* and *scaling* of the Lattice. A truncation region is the subset of the Lattice that will actually be coded. For each cell in the region of support, the reproduction vector is taken to be either the midpoint of the cell, or the centroid of the training vector lying in that cell. The optimal encoder, for a given codebook Y selects the vector Y_i if: $d(X, Y_i) \leq d(X, Y_j)$, for all j. The optimal encoder, thus, operates on a nearest-neighbour or minimum-distortion basis. Using WT, and hence having i.i.d. gereralized Gaussian source vectors, the Lattice point shells will have a Pyramidal shape [9]. By truncation, the largest pyramid radius, r, is chosen to determine the number of concentric pyramidal shells of Lattice points within the pyramidal volume and, hence, the codebook size $|c|$. The choice of support region can have an important effect on the encoding speed and quality. Since the structure of the Lattice points is always the same for a particular LVQ, regardless of the input to the quantizer, the *Voronoi* cell of every element in the region has, therefore, an identical shape and size. As a result, it is also desirable to scale the Lattice. By scaling, the density of Lattice points can be changed and the ETD, can hence be minimized. For the scaled Lattice, a scaling factor $s < 1$, squeezes the Lattice, increases the density of Lattice points, and reduces the distortion. However, the volume enclosed by the truncated surface and the probability that a source vector falls within that region decreases. An n-D vector will fall into a truncated (by r-radius) and scaled (by s) Lattice, if its ℓ_1-norm is less than a predetermined value E_{max}:

$$\sum_{i=1}^{n} |x_i| \leq E_{max} = sr. \quad (1)$$

With the pyramidal Lattice points, the wedge problem arises when the input vectors, falling into each wedge region, are all projected to Lattice points with at least one coordinate being zero. These vectors will be reconstructed with 1 to $n-1$ degrees of freedom and as n increases, the number of wedge regions increases too. Since wedge regions contain high-energy edge information, the wedge problem has an obvious distortion effect, blurness, on the reconstructed image.

One of the best image compression techniques based on LVQ was proposed by Barlaud et al. [7], however the technique had two major drawbacks. The entropy measure used, is not achievable because the codebook size can be orders of magnitude greater than the number of quantizer source vectors. Furthermore, their algorithm does not consider the wedge region problem encountered with pyramidal shells. To overcome these difficulties, in [12], a concentric double-density PLVQ was discussed in which three preassumptions are made. Additionally, training and multiquantizing procedures still need to be performed. In [9], with the lack of the relation between the radius of the pyramid and the codebook size (the relation was later introduced in [7]), the vectors on the outermost shell are considered as codewords. This algorithm which covers just cubic Lattices, uses constant PDFs in different regions which makes the restriction of having only a small probability of containing input vectors within each region. Using several-density Lattice regions, the algorithm ends up with experimentally determined values for the two main parameters (scale factor ratio, s, and scale factor, c_0), using *Monte Carlo* simulations.

4. PROPOSED PIECEWISE-UNIFORM PLVQ ALGORITHM

In the proposed algorithm, the WPT associated with the 73-subband decomposition structure proposed in [11], is applied to the mean-removed input source image. Having the WPT coefficients, a *"hard thresholding"* scheme is applied where the part of coefficients having energy less than a predetermined threshold level are set to zero. For each subimage, the source vectors (with mean-removed normalized PDF), the allocated bit budget $R_{m,d}$, and its corresponding distortion $D_{m,d}$, are obtained.

In the proposed Piecewise-Uniform PLVQ algorithm, three Lattice densities are considered which are separated by two surfaces of constant probabilities. For any arbitrary PDF, the corresponding density of Lattice points is proportional to its PDF. Having a highly Non-Uniform and sharp

PDF, the inner most Lattice is considered to be the densest and the outer most Lattice is empty. The second Lattice with sparse density s_2, is designed to cover the *less probable* high-energy vectors which might fall into wedge regions. Consequently, the edge details will be quantized properly. Note that using a truncated sparse Lattice will not cost the codebook size to increase significantly. The ETD is defined as:

$$ETD = \ell D_{m,d} = \rho_\Lambda (P_1 s_1^2 + P_2 s_2^2), \quad (2)$$

where ℓ is a variable with $\ell \ll 1$, P_i is the probability of the input vectors lying within the i-th Lattice, s_i is the scaling factor and ρ_Λ is the upper MSE bound [?].

The vectors of the third Lattice are projected to the outermost shell of the second Lattice r_2, and the third Lattice becomes empty, then:

$$P_2 = 1 - P_1. \quad (3)$$

The ratio of scaling factors is:

$$k = \frac{s_1}{s_2}, \quad (4)$$

where $k < 1$. Using (3) and (4) in (2), we get:

$$s_1 = \sqrt{\frac{k^2 \ell D_{m,d}}{\rho_\Lambda (P_1 (k^2 - 1) + 1)}}. \quad (5)$$

The total codebook size becomes:

$$|c| = \nu(r_1) + [\nu(r_2) - \nu(\frac{s_1}{s_2} r_1)], \quad (6)$$

where the truncation levels are defined as:

$$r_1 = \frac{E_{max_1}}{s_1}, \qquad r_2 = \frac{k E_{max_2}}{s_1}. \quad (7)$$

The total bit requirement will be:

$$R = P_1 R_1 + P_2 R_2 \qquad bpp$$

or:

$$R = P_1 \frac{log_2(\nu(r_1))}{n_{m,d}} + (1 - P1) \frac{log_2(\nu(r_2) - \nu(\frac{s_1}{s_2} r_1))}{n_{m,d}}. \quad (8)$$

Where R_i and $n_{m,d}$, are the bit requirement for the i-th concentric Lattice and the dimension of the Lattices in the (m,d)-th subimage respectively. The computation of the truncation level and scaling factor of Lattices is based on obtaining the smallest possible ETD, using a given bit budget. The proposed algorithm starts with the best possible condition under which all of the input vectors are contained inside two Lattices which have the smallest possible scaling factors. The condition will then be examined using (8). If $R \leq R_{m,d}$ the condition is met. If the bit budget can not afford such a bit rate, first the k increases leading to a smaller difference between to scales. If this is not enough, the E_{max_1} increases leading to higher scaling factors. If the bit budget can not afford the desired ETD, by decreasing the E_{max_2}, the algorithm allows a small number of vectors to be projected on the outermost shell of the second Lattice r_2, and examines that. In this situation, since we have imposed the Lattices to be as dense as possible, the ETD will have the smallest possible value. If the current condition can not be met, the algorithm goes for a slightly higher ETD measure, and the iteration continues.

The experimental results showed that with $\ell \ll 1$ which leads to a small codebook size (requiring less bits), the proposed algorithm always computes the truncation level and scaling factor which lead to a small ETD measure. In each iteration, if the value of the k or E_{max_1} tends to an unreasonable bit rate requirement or if the reduction of k tends to a higher bit rate requirement, the algorithm does not continue the loop and goes to check the next condition.

As explained above, no predetermined assumptions about the variables are made and there is no need to go through training and multi-quantizing procedures. Since the range of the desired values of the variables is very small, and the algorithm converges in few iterations only, the procedure is fast.

The *D4-PLVQ* is applied to all subimages, except for the diagonal subimage of the first level and the vertical and low-passed subimages of the fourth level. The input vectors are scaled and quantized using any of the three regions based on their energy measures. The corresponding region-code and the few projected values are saved for usage in dequantization.

The first level's diagonal subimage, is usually discarded which causes some blurness in the reconstructed image. In this work, after selecting the "essential" data, a binary image containing the mean of the positive and negative data is generated. Doing so, the algorithm will be simplified (no need of quantization and indexing part) while the quality of the reconstructed image is better preserved. For the other subimages (mostly contained in the first WT level), if the allocated bit rate $R_{m,d}$ is very small, the truncated level tends to be zero. In this case, all of the coefficients should be quantized to the Lattice centre which is a zero vector. This means that the high frequencies are going to be set to zero, which will have a blurring affect on the reconstructed image. To avoid this situation, the algorithm firstly checks the given bit rate and computes its available codebook size, using $|c| = 2^{n_{m,d} R_{m,d}}$. Referring to the corresponding ν function, if this codebook size tends to a zero Lattice radius, then the algorithm automatically implements the PNM method. The other situation where the PNM method is automatically chosen, is when (due to a very low bit rate) the scaled vectors are mostly going to be quantized to zero. Hence, the high-frequency content will still be efficiently used in dequantization. The Lattice codebooks are then indexed, using the fast encoding method [?] and entropy encoded.

The average information of the codebook is:

$$\Re_{m,d} = P_1 \Re_{1,m,d} + (1 - P_1) \Re_{2,m,d}, \quad (9)$$

where for each Lattice $\Re_{i,m,d}$ is obtained using:

$$\Re_{i,m,d} = -\frac{1}{n_{m,d}} \sum_{j=1}^{L_i} p(v_{i,j}) log_2 p(v_{i,j}), \quad (10)$$

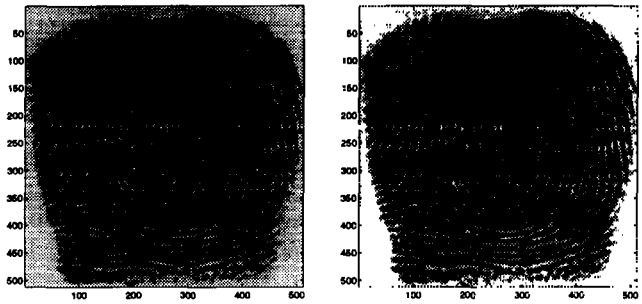
where $p(v_{i,j})$ is the probability of selecting the n-D index vector $v_{i,j}$, belonging to the obtained indices at level m and corresponding to the orientation d, during the coding of the i-th Lattice of that subimage. The total estimated entropy, $\Re_T$, is computed as:

$$\Re_T = \frac{\sum_{m=1}^{M} \sum_{d=1}^{3} \Re_{m,d} n_{m,d} y_{m,d}}{n_T^2} \qquad bpp, \quad (11)$$

where M is the depth of WT, and $n_{m,d}$ and $y_{m,d}$ are the lengths of the codewords for each $n \times y$ subimage, and n_T^2 is the size of the original image.

5. PRELIMINARY RESULTS

Using a 512 × 512 fingerprint image, the performance of the proposed algorithm was examined. The results of using different compression ratios are given in Fig. 1. Table 1 shows the obtained *PSNR*s using *JPEG*, the FBI's fingerprint coder (*WSQ*) [1] and the proposed PU-PLVQ algorithm. To show the performance of the proposed algorithm using other images, the popular "*Lena*" image, 512 × 512, was encoded. Fig. 2, shows the efficiency of the proposed algorithm compared to the *JPEG*. Using this image, the PSNRs obtained from JPEG, Barlaud's [7], and the proposed PU-PLVQ algorithm for **0.17 bpp** where 29.4 dB, 30.3 dB, **36.84 dB** respectively. The initial results using the proposed algorithm are satisfactory, both in terms of quality and computational load.



(a) (b)

Figure 1: (a) Original Image, 8 bpp; (b) reconstructed, *bpp* = **0.15**, *PSNR* = **29.82** dB.

$\Re_T$ (bpp)	*JPEG*	*WSQ*	*Proposed*
0.65	30.75	31.71	38.17
0.45	28.82	29.91	35.74
0.15	20.15	25.18	29.82

Table 1: *PSNR*s for fingerprint image.

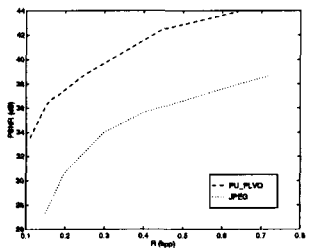


Figure 2: Performance of the proposed algorithm, using *"Lena"* image.

6. CONCLUSION

A new compression algorithm for Fingerprint images is introduced. A modified wavelet packet scheme which uses a fixed decomposition structure, designed for fingerprint images, was used. A new design method for computing the truncation level and the scaling factor of Lattices in order to obtain the smallest possible ETD, while using the smallest appropriate codebook size is presented. In the proposed Piecewise-Uniform PLVQ, the wedge problem encountered with the Pyramidal Lattice point shells is resolved. At very low bit rates, for the coefficients with high-frequency content, the PNM method was proposed which improved the quality of the reconstructed image. In addition to the proposed algorithm, the performance of other techniques was also discussed with results showing that the proposed technique outperforms JPEG and the FBI's WSQ algorithm for fingerprints.

7. REFERENCES

[1] J. N. Bradley and C. M. Brislawn. Proposed First-Generation WSQ Bit Allocation Procedure. *site:*, 1993.

[2] J. H. Conway and N. J. A. Sloane. Fast Quantization and Decoding algorithm for Lattice Quantizers and Codes. *IEEE Trans. Inform. Theory*, IT-28(2):227–232, March 1982.

[3] J. H. Conway and N. J. A. Sloane. A Fast Encoding Method for Lattice Codes and Quantizers. *IEEE Trans. Inform. Theory*, IT-29(6):820–824, Nov. 1983.

[4] J. H. Conway and N. J. A. Sloane. A Lower Bound on the Average Error of Vector Quantizers. *IEEE Trans. Inform. Theory*, IT-31(1):106–109, Jan. 1985.

[5] P. C. Cosman, R. M. Gray, and M. Vetterli. Vector Quantization of Image Subbands: A Survey. *IEEE, Trans. Image Pro.*, 5(2):202–225, Feb. 1996.

[6] M. Antonini et al. Image Coding Using Wavelet Transform. *IEEE Trans. Image Pro.*, 1(2):205–220, April 1992.

[7] M. Barlaud et al. Pyramid Lattice Vector Quantization for Multiscale Image Coding Image Coding. *IEEE Trans. Image Pro.*, 3(4):367–381, July 1994.

[8] T. R. Fisher. A Pyramid Vector Quantizer. *IEEE, Trans. Inform. Theory*, IT-32(4):568–583, July 1986.

[9] D. G. Jeong and J. D. Gibson. Uniform and Piecewise Uniform Lattice Vector Quantization for Memoryless Gaussian and Laplacian Sources. *IEEE, Trans. Infom. Theory*, 39(3):786–804, May 1993.

[10] S. Kasaei, M. Deriche, and B. Boashash. Fingerprint Compression Using a Modified Wavelet Transform and Pyramid Lattice Vector Quantization. In *TENCON'96*, pages 798–803, Perth, Western Australia, Nov. 1996.

[11] S. Kasaei, M. Deriche, and B. Boashash. Performance Analysis of Fingerprint Compression Using an Efficient Wavelet Transform Algorithm. In *ISSPA '96*, pages 433–436, Gold Coast, Australia, Aug. 1996.

[12] W. C. Powell and S. G. Wilson. Lattice Quantization in the Wavelet Domain. *SPIE, Mathematical imaging*, 2034:218–229, 1993.