The proposed index mapping can also be applied to many hybrid VQ systems such as mean-predictive VQ (MPVQ) and hierarchical VQ (HVQ) [1]. The index mapping process is similar to that in ordinary VQ except that the cost in eqn. 2 is modified as

$$C_{ij}(f) = d_{ij}[\delta(k_{i,j})] + \delta(m_{k_j} - h_{j}) + \delta(f)$$

where $m_{k_j}$ is the mean of the decoded $(k,j)$th block and $l$ is a vector of length $M$ with each element equal to 1. In MPVQ, the block mean was predicted from four previously decoded blocks and a codebook with size 128 was used to quantise the residual vector. In HVQ, a $2 \times 2$ block-means were first vector quantised with a mean codebook of size 64, and then each mean-removed block (of size $4 \times 4$) was vector quantised with a residual codebook of size 128. Our index mapping scheme was used to encode the residual codewords in MPVQ and both the residual codewords and mean codewords in HVQ. The simulation results are shown in Tables 2 and 3.

Table 2: Bit rate reduction performance in MPVQ

<table>
<thead>
<tr>
<th>Image</th>
<th>Original</th>
<th>Proposed</th>
<th>Gain [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>0.3199</td>
<td>0.2670</td>
<td>16.5</td>
</tr>
<tr>
<td>Bank</td>
<td>0.3424</td>
<td>0.2672</td>
<td>21.9</td>
</tr>
</tbody>
</table>

Table 3: Bit-rate reduction performance in HVQ

<table>
<thead>
<tr>
<th>Image</th>
<th>Original</th>
<th>Proposed</th>
<th>Gain [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>0.0839</td>
<td>0.4189</td>
<td>78.8</td>
</tr>
<tr>
<td>Bank</td>
<td>0.0877</td>
<td>0.3525</td>
<td>59.2</td>
</tr>
</tbody>
</table>

Conclusion: We proposed a simple dynamic index mapping which considerably reduces the index entropy by exploiting interblock correlation. It was shown that this mapping scheme can provide a considerable amount of bit-rate reduction not only in ordinary VQ but also in many hybrid VQ systems. Since the proposed index encoding scheme requires only a small amount of additional computation and does not change the quantiser structure, it is expected to be easily applied to other VQ systems.

Fast LBG codebook generator for BTC image compression

Kuang-Shyr Wu and Ja-Chen Lin

Indexing terms: Image processing, Data compression, Vector quantization, Block codes

When the high-mean and low-mean generated by the BTC image compression technique are to be quantised using VQ, the computation time required to search for the nearest centroid can be reduced significantly using the proposed method. Experiments comparing the full-search and EHPM algorithms demonstrate this.

Introduction: The image compression technique BTC uses two values $a$ and $b$, called the high-mean and low-mean, to replace all values in a block. To increase the compression ratio, the high/low mean pairs are often quantised further through VQ [1, 2]. We concentrate here on the task of LBG [3] codebook generation for the (high/low) mean pairs. A typical LBG cycle contains the steps:

1. For each data point $q$, assign to the $j$th cluster if $|q-c_j| = \min_{c_{i\in C}} |q-c_{i}|$

2. Replace $\{c_{i}\}_{i=1}^{N}$ by the new centroids of the $N$ clusters just formed.

Here, $C = \{c_{i} = (x_{c_{i}}, y_{c_{i}})\}_{i=1}^{N}$ is the codebook in question. We present the new method for reducing the time needed in step (i). First, evaluate to obtain $C = \{c_{i} = (x_{c_{i}}, y_{c_{i}})\}_{i=1}^{N}$ and $C = \{c_{i} = (x_{c_{i}}, y_{c_{i}})\}_{i=1}^{N}$, which are the sum-projection and subtraction-projection of the codebook to the straight lines $x = x$ and $y = y$, respectively. Assume that the $\{c_{i}\}_{i=1}^{N}$ has been sorted. For each data point $q$ we proceed as follows, to obtain the nearest code-word $d(q)$ always denotes a 2-D Euclidean norm $\|q\|$, whereas $|q|$ denotes the 1-D absolute value:

Algorithm:

Step 1: Calculate $\tilde{q} = (qx, qy)$ and $\tilde{q} = (qx, qy)$. Use one-dimensional binary searching to find the $c_{i}$, that satisfies $|q-c_{i}| = \min_{c_{i}\in C} |q-c_{i}|$. The corresponding $c_{i}$ is then used as an initial guess for the current nearest codeword (CNC) of $q$.

Step 2: Calculate $d_{min} = |q-c_{i}|$. Construct the remaining set (RS) by collecting those $c_{i}$ whose $c_{i}$ satisfy $|q-c_{i}| < (d_{min}/\sqrt{2})$.

Step 3: Delete $c_{i}$ from RS, because $c_{i}$ has been checked.

Step 4: If RS is empty, return CNC as the nearest codeword of $q$ and exit.

Step 5: In the RS, obtain the $c_{i}$, whose $c_{i}$ satisfies $|q-c_{i}| = \min_{c_{i}\in RS} |q-c_{i}|$.

(Unlike in step 1, there is no need to perform a binary search here. Since $\{c_{i}\}_{i=1}^{N}$ has been sorted, $|q-c_{i}|$ has been sorted after step 1; $q$ $\in$ $\{c_{i}\}_{i=1}^{N}$ has been sorted after step 2. Therefore, one of the two neighbours of the previous $c_{i}$ must be the candidate.)

Step 6: If $c_{i}$ violates any of the following three inequalities:

$$|q-c_{i}| < (d_{min}/\sqrt{2})$$

$$|q_{x}-c_{x}| < d_{x_{min}}$$

$$|q_{y}-c_{y}| < d_{y_{min}}$$

then go to step 3, else calculate $d(q) = |q-c_{i}|$.

Step 7: If $d < d_{x_{min}}$, then update $d_{x_{min}}$ and CNC with $d$ and $c_{i}$, respectively. (Also use this new $d_{x_{min}}$ to delete from the RS the $c_{i}$ whose $c_{i}$ violates eqn. 1.)
Step 6: Go to step 3.

The \( c_i \) obtained in step 1 is deleted in step 3, and hence the \( q_i \) obtained in step 5 is different. Furthermore, after the deletion of the codewords \( c_i \) in step 7, the \( q_i \) of every codeword \( c_i \) remaining in the RS must satisfy \( |q_i - c_i| < (d_{\min}/\sqrt{2}) \). Since the next \( c_i \) obtained in the next execution of step 5 is also chosen from the RS, then \( q_i \) must also satisfy

\[
|q_i - c_i| < (d_{\min}/\sqrt{2})
\]

(6)

shows the total number of vector-distance computations (tovd) required to make the LBG converge. Since there is some overhead (eqns. 1 - 5) for reducing the number of \( \| \) computations in our method, to be fair to the other two methods we also list in Table 1 the total CPU time needed in LBG (including both steps 1 and 2). Regardless which of the two criteria is used to compare the performance, the proposed method shows better results than the full-search and EHPM algorithms. All three methods obtain the same (final) codebook and require the same number of LBG cycles. The only difference is the CPU time.

Table 1: Comparison of total number of vector-distance computations (tovd) and CPU time

<table>
<thead>
<tr>
<th>Codebook sizes</th>
<th>Methods and performance</th>
<th>Lena</th>
<th>Baboon</th>
<th>Couple</th>
<th>Crowd</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>tovd</td>
<td>time(s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>FS</td>
<td>1091</td>
<td>3180</td>
<td>2277</td>
<td>2647</td>
</tr>
<tr>
<td></td>
<td>EHPM</td>
<td>4091</td>
<td>824663</td>
<td>619476</td>
<td>3606481</td>
</tr>
<tr>
<td></td>
<td>ours</td>
<td>865928</td>
<td>1270259</td>
<td>651093</td>
<td>1059592</td>
</tr>
<tr>
<td></td>
<td>tovd</td>
<td>time(s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>FS</td>
<td>3386</td>
<td>3151</td>
<td>3590</td>
<td>3362</td>
</tr>
<tr>
<td></td>
<td>EHPM</td>
<td>442462</td>
<td>6170464</td>
<td>9654594</td>
<td>3943527</td>
</tr>
<tr>
<td></td>
<td>ours</td>
<td>832203</td>
<td>954876</td>
<td>1669808</td>
<td>924708</td>
</tr>
</tbody>
</table>

Machine used is SUN-SPARC 10. FS: 'full search'

Conclusions: The proposed fast nearest codeword searching algorithm accelerated the vector quantisation of the high/low means generated in BTC compression. The method condensed the searching area, and reduced the total CPU time required for the LBG.

Acknowledgment: This study was supported by the National Science Council, Republic of China, under contract NSC83-0404-E009-081.

© IEE 1995 25 May 1995
Electronics Letters Online No: 19950978
Kuang-Shyr Wu and Ja-Chen Lin (Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 30020, Republic of China)
Ja-Chen Lin: corresponding author

References

Multicomponent heterodyne laser Doppler anemometer using chirp-modulated Nd:YAG ring laser and fibre delay lines

J.W. Czarske and H. M biller

Indexing terms: Doppler velocimetry, Anemometers, Ring lasers, Solid lasers, Optical delay lines

A two-dimensional directional laser Doppler anemometer using a frequency modulated Nd:YAG laser is presented for the first time. The magnitude and sign of the fluid velocity were determined by employing the heterodyne technique without having to use an external frequency shifter.