Joint Image Classification and Compression Using Hierarchical Table-Lookup Vector Quantization

Navin Chadda, Keren Perlmuter and Robert M. Gray
Information Systems Laboratory
Stanford University CA-934305

Presented by – Pramod J. Nathan

Motivation

- Image Compression: Efficient Storage and Transmission
- Image Classification: Provides Classes
  - Classes used for: Assisting human observers
    - Different types if images/anomalies in similar images
  - Help compression by using different compression algorithms/tables etc for different classes
    - E.g. using different DCT for different images

Introduction

- In this paper low level compression and classification is desired for:
  - C&C of a digital image to highlight anomalies
  - Archival and categorization of different terrain types
- Use Hierarchical Table-Lookup VQ to increase speed and make table sizes manageable for large dimension VQ

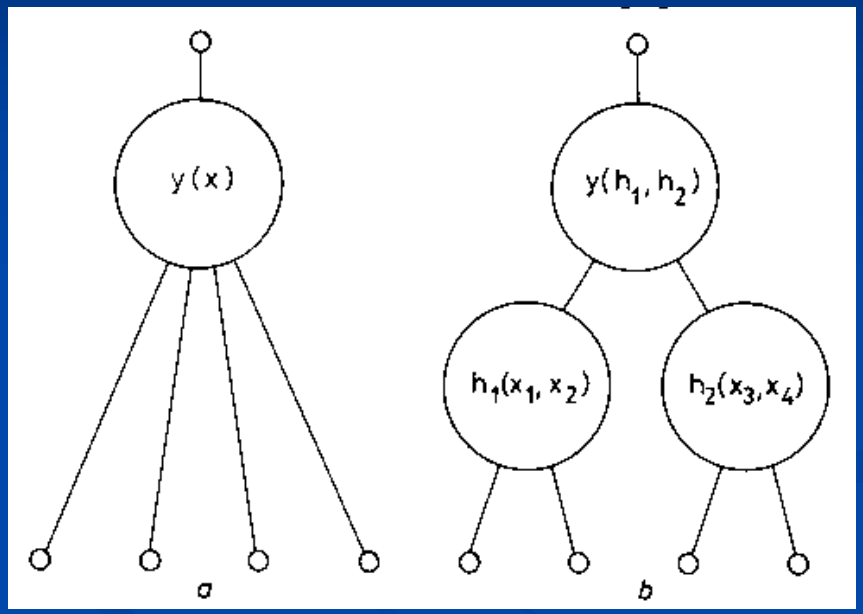
Introduction

Compression technique: Hierarchical Lookup-Table Vector Quantization

- Explore different techniques for image classification and compression:
  - Sequential classfier/quantizer
  - modified Learning Vector Quantizer (LVQ)
  - Sequential quantizer/classifier
  - Bayes VQ with posterior estimation

Hierarchical Table-Lookup Vector Quantization

- VQs use an exhaustive search to encode
- HVQs encode Vectors using only Table-Lookups
  - Tradeoff – accuracy for speed
- Replaces the full search VQ with hierarchical table lookups. This gives 1 table lookup per input symbol for encoding.
- Hierarchical structures successively quantize the vector in stages



"The most important difference between Fig. 1a and Fig. 1b is the input dimensionality of the various encoding operations. In Fig. 1a it is 4, whereas in Fig. 1b it is 2. If we were to consider a look-up table implementation of the whole encoding operation, then the hierarchical scheme in Fig. 1b is very much cheaper (in terms of table size) to implement. This is the crucial advantage of hierarchical vector quantisation which we shall make use of in this paper." – "*Hierarchical Vector Quantization*" ***S. P. Luttrell***

HVQ: Example

- Single Frame of B/W video. Single frame = array of pixels with 8 bit value for intensity with Vector Dimension V = 2
  - A lookup table will have 16 address bits
  - We need this and N codewords to encode each 2-D vector to its nearest codeword to encode into a **single lookup table**
  - BUT! For large V: the table size can get very large
    - 8X8 block of pixel => V = 64 ; Lookup table = 64*8 = 512 value
- This complexity can be reduced by using HVQ
- V = 8 dimensional vector with precision 8 bits is encoded to R = 8 bits in 3 stages
- Compression ratio = 8:1; Computational Complexity = 1 look up per input symbol

Fast JPEG Encoding for Color Fax Using HVQ

*Ricardo L. de Queiroz*1 *and Patrick Fleckenstein*2

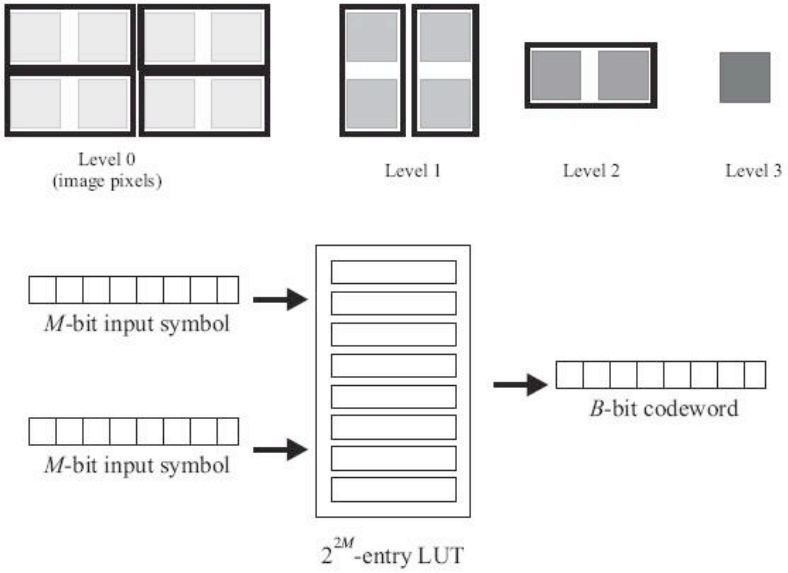
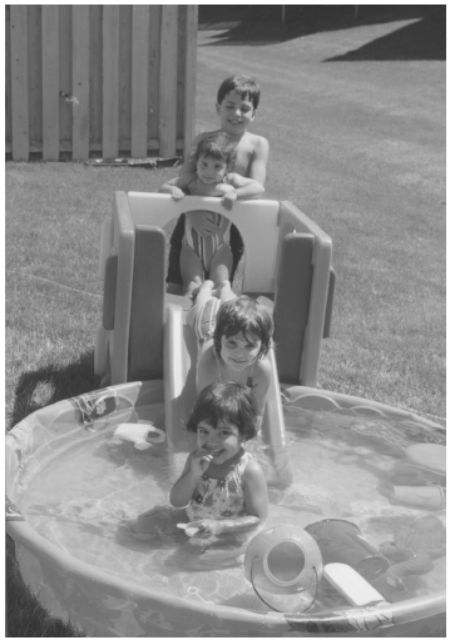


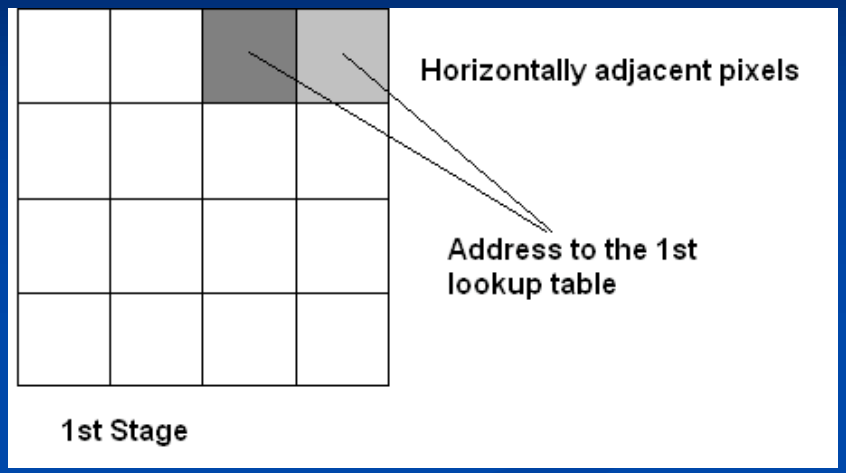
Figure 3. HVQ look-up tables and a 3-level encoding hierarchy.

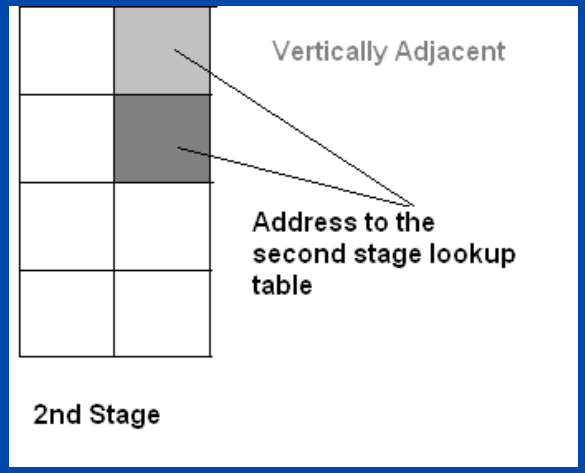
Figure 7. Left: reconstructed image "kids" after 50:1 JPEG compression; right: same after 50:1 HVQ-JPEG compression.



Designing the JCCHVQ

- Encoder: Consists of M stages
  - Each stage is a lookup table
  - Odd stages = Rows
  - Even stages = Columns





Each stage >=2, less than Final stage: the table is constructed using 2 adjacent outputs of the previous stage

Last stage produces the encoding index U

1. **U represents an approximation to the input vector**
2. **U gives classification information**
3. **U is sent to the decoder**

The lookup tables are build in the same incremental way using mean squared distortion to match to the closest index. i.e. the index if the codeword with the least squared error is put in the output table for that particular input combination

Design of JCCHVQ (cont)

- Last stage: All classification is done here
  - Sequential TSVQ/Classifier
  - Centroid based LVQ
  - Bayes TSVQ with class probability tree
  - Bayes TSVQ with posterior estimation TSVQ

*

TSVQ = Tree stuctured vector quantization

LVQ = Learning Vector Quantization (Like Kohnen map)

Design of Joint Classifier and Quantizer using HVQ

- Sequential quantizer/classifier:
  - Full search compression using GLA to minimize mean square error.
  - Bayes classifier used to classify the quantizer outputs
- Sequential classifier/quantizer
  - Class information used to classify information used to direct each vector to a particular quantizer
- Centroid based LVQ
  - Codewords are modified by each vector such that the codeword shares the same class as the vector
  - Algorithm does not compress; hence the codebook generated by LVQ is used for classification alone
  - A modified version of this codebook is used to produce the final vector
  - Encoder codewords used by LVQ are replaced by centeroids of the training vector; These centroids are used for the final stage compression

- Bayes Risk weighted quantization
  - Encoder: selects nearest neighbor minimizing the modified distortion measure* to select the best codeword
  - Trees are grown by using the node that givens the largest decrease in average distortion to increase in bit rate.
  - Trees are pruned to obtain optimal subtrees
  - Posteriors can be estimated based on the learning set:
    - P(X has class a) = number of times X occurs in a/number of times X occurs in the training set
  - Two subsystems are tested
    - 1) **Posterior Estimate TSVQ:**
      - MSE is used to determine the path till the terminal node is reached
      - Posterior probability = frequency of the class labels within a node
    - 2) **Class probability tree**
      - If x features are extracted from the vectors, the trees are used to select from a set of candidate features with which to split the data

Simulation

- Compare the performance of the different algorithms using CT scan images and Aerial images:
  - Aerial images: Man-made portions - White
  - CT scan images: Tumor regions - White
- Compare the results of the using a HVQ to VQ (remember: HVQ is less accurate but 3-4 times as fast for this simulation)

Simulation Results

- Aerial:

Method	Rate bpp	Classification error for VQ techniques	Classification error for HVQ techniques
Sequential TSVQ/Classifier	0.5	29.56%	32.24%
Sequential Classifier/TSVQ	0.5	22.41%	21.44%
Centroid Based LVQ	0.5	19.58%	19.77%
Bayes TSVQ with class probability tree	0.5	19.12%	19.46%
Bayes TSVQ with post est. TSVQ	0.5	20.89%	20.91%

Simulation Results

- CT (1.75 bpp) :

Method	Sensitivity (VQ)	Specificity (VQ)	Sensitivity (HVQ)	Specificity (HVQ)
Sequential VQ/Classifier	81.06%	96.56%	80.30%	96.51%
Sequential Classifier/TSVQ	85.61%	96.91%	82.58%	96.99%
Bayes full search with post est. TSVQ	85.61%	96.91%	82.58%	96.99%
Bayes TSVQ with post est. TSVQ	85.61%	96.99%	82.58%	97.04%

Simulation Results (Images)

Conclusions

- Both classifier/encoder are based on table lookups
  - Do not require arithmetic computations
- Faster than using VQ