A Fast Index Assignment Method for Robust Vector Quantisation of Image Data

N. L. C. Talbot and G. C. Cawley

November 30, 2017

Contents

Vector quantisation is a widely used technique in low-bit rate coding of speech and image data, but is highly sensitive to noise in the transmission channel. If the reference vector recalled by a corrupted index differs greatly from the intended reference vector, image quality can be degraded quite dramatically. The index assignment (IA) process attempts to re-order the code book so as to minimise the effects of errors introduced in the transmission channel, by assigning indices with similar binary patterns to similar reference vectors, usually at considerable computational expense. This paper describes a fast, novel index assignment algorithm based on Hall's solution to the quadratic assignment problem.

Chapter 1

Introduction

Given p-dimensional unlabelled data $Y = \{\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_m\} \subset \Re^p$ representative of a data manifold $V \subseteq \Re^p$, the process of vector quantisation attempts to partition V into a number of sub-regions V_i using a finite set of reference or *code book* vectors, $W = \{\vec{w}_1, \vec{w}_2, \vec{w}_3, \ldots, \vec{w}_n\} \subset \Re^p$. The values of the reference vectors are chosen so as to minimise the quantisation error, measured according to a distance metric, between a training vector $\vec{y}_i$ and the best matching reference vector $\vec{w}_{(y_i)}$ over all vectors in Y. An incoming vector $\vec{v} \in V_i$ is mapped onto the most similar reference vector $\vec{w}_i$. The incoming vector can then be represented by the binary index i, resulting in a considerable reduction in the required bit rate. Vector quantisation is however very sensitive to errors in the transmitted codeword, due to noise in the transmission channel, as the reference vectors retrieved by the intended and corrupted indices might be very different. The index assignment (IA) process attempts to re-order the code book, such that similar reference vectors are recalled by indices with similar binary patterns, minimising the effects of an error in the transmitted index.

This paper presents the results obtained using a novel index assignment method, based on Hall's solution to the quadratic assignment problem, for robust gain-shape vector quantisation of image data.

Chapter 2

Quadratic index assignment

The aim of index assignment can be viewed as placing the $n = 2^r$ reference vectors in an r dimensional binary space, representing the transmitted code words or indices, such that similar reference vectors are placed in similar locations in this r dimensional binary space. In the approach described here, Hall's solution to the quadratic assignment problem is used to obtain a suitable placement for each of the reference vectors in an r dimensional *continuous* space, which is then quantised using a simple recursive partitioning algorithm to produce a near optimal reordering of the code book.

2.1 Quadratic assignment

Many practical problems, for instance VLSI cell placement, can be considered as special cases of the quadratic assignment problem. The quadratic assignment problem can be simply stated as follows: If x_{ik} represents the x_k co-ordinate of node i, and c_{ij} represents the 'connection strength' between nodes i and j, find the optimal placement for each node so as to minimise

$$\frac{1}{2} \sum_i \sum_j \sum_k (x_{ik} - x_{jk})^2 c_{ij}$$

Hall presents a solution to this problem by constructing the 'disconnection' matrix, B, where

$$b_{ij} = \begin{cases} \sum_k c_{ik} & i = j \\ -c_{ij} & i \neq j \end{cases}$$

The matrix B is semi-positive definite with eigenvalues $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_n$. The solution vectors, $\vec{x}_k$, are then the eigenvectors of B corresponding to the r smallest non-zero eigenvalues.

2.2 Index assignment

The following procedure is used to reorder the code book using Hall's quadratic assignment algorithm: given $n = 2^r$ reference vectors, construct the disconnection matrix such that the connection strength c_{ij} is large if $\vec{w}_i$ and $\vec{w}_j$ are

similar, and small if they are dis-similar. For initial experiments, the connection strength is given by:

$$c_{ij} = \begin{cases} \frac{1}{\|\vec{w}_i - \vec{w}_j\|} & i \neq j \\ 0 & i = j \end{cases}$$

A recursive partitioning algorithm is then used to convert the continuous valued solution to a set of binary indices used to re-order the code book, as follows:

1. Construct the $(n \times (r + p))$ augmented matrix

$$\vec{Z} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1r} & \vec{w}_1^T \\ x_{21} & x_{22} & \cdots & x_{2r} & \vec{w}_2^T \\ \vdots & & \ddots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nr} & \vec{w}_n^T \end{pmatrix}$$

2. For each $i = 1, \ldots, r$
   set $N = 2^{r-i+1}$
   3. For each $j = 1, \ldots, 2^{i-1}$
      sort rows $(j-1) \times N + 1$ to $j \times N$ of $\vec{Z}$
      so that column i is in piecewise ascending
      order
4. Remove the first r columns of the matrix $\vec{Z}$

$\vec{Z}$ now represents the sorted code book.

Chapter 3

Method

For evaluation, a codebook was created from images taken from the ORL face database. Each image was first reduced to 90×111 pixels, by deleting the first and last columns and the last row, and then divided into 3×3 pixel blocks. A gain-shape vector quantisation scheme was implemented, and so the resulting training vectors were normalised to have unit length before generating a code book of 256 reference vectors using the Linde-Buzo-Gray (LBG) algorithm. The codebook was then re-ordered using the proposed method based on Hall's solution to the quadratic assignment problem and also, for comparison, using a simple pair-wise interchange algorithm with simulated annealing.

Chapter 4

Conclusions

A novel index assignment algorithm is presented, based on Hall's solution to the quadratic assignment problem. The resulting code book ordering is of comparable quality to that generated using a conventional simulated annealing algorithm, but is obtained at a greatly reduced computational expense.