Coding of Image Residuals with Tailbiting Convolutional Codes and BCJR Decoding

Mirek Novak

Dept. of Information Technology, Lund University, S-221 00 Lund, Sweden, mirek@it.lth.se

Abstract

Image residuals have been source encoded using modulated codewords from a tailbiting convolutional code. The source encoder represents each residual with the closest codeword, using the BCJR decoding algorithm.

INTRODUCTION

This work uses the duality of source encoding/channel decoding for still image coding. The BCJR decoding scheme for tailbiting convolutional codes, [1], is used to find close representatives for observed data.

An image, $u$, is divided into blocks, $u_i$, of $n$ pels. Each block is the sum of a model block, $\mu_i$, and a residual, $r_i$, see, e.g., [2]. If the model order is $m$, then $r_i$, is in a $k = n - m$ dimensional subspace of $\mathbb{R}^n$. We here concentrate on coding the difficult part; the residual.

The residual, $r_i$, is transformed using a nonunitary, modified KL-transform, $G$, into a $k$-dimensional transformed residual vector, $s_i$, of uncorrelated components with unit variance and zero mean. Hence $s_i = Gr_i$. The probability density of the transformed residual vectors is spherically symmetric. For transformed residuals obtained from images, the distribution of its components is close to Laplacian.

In Euclidean space, $\mathbb{R}^k$, a set of vectors, $\{Y_i\}_{i=1}^{2^kR}$ is generated from the codewords, $\{c_i\}_{i=1}^{2^kR}$, of a length $q \cdot k$, rate $1/c$, tailbiting convolutional code, [3], by mapping $q$ consecutive bits from $c_i$ to the corresponding component of $Y_i$, i.e., a modulation. Each $Y_i$ can then be represented by $qk/c$ bits, or $q/c$ bits per component.

The encoder works as follows: after observing $s_i$, it is “decoded”, using the tailbiting BCJR. The scheme produces a $qk/c$ bit long sequence, $d_i^*$, which defines the codeword $c_i^*$. This codeword in turn defines $Y_i^*$, which is a good approximation of $s_i$. The approximation of the transformed residual is then $\hat{s}_i = Y_i^{**}$.

The decoder obtains $d_i^*$. Using the convolutional encoder it produces the codeword $c_i^*$ and from this it generates $Y_i^{**}$ which is used as the approximation $\hat{s}_i$. The image residual approximation is computed using the inverse transform as $\hat{r}_i = G^{-1}\hat{s}_i$.

Finally, the reconstructed image block is computed as $\hat{u}_i = \mu_i + \hat{r}_i$.

The scheme has been used with rate 1/4 and 1/8 codes (0.5 bpp and 0.25 bpp) for coding of the residuals of the standard 512 $\times$ 512 test images lenna and baboon.

REFERENCES


This work has been supported by the Swedish Foundation for Strategic Research – Personal Computing and Communication.