Image Compression with Wavlets

Benjamin Ong

Abstract— **This report outlines ideas used in a simple image compression algorithm involving wavelets.**

Keywords— **Wavelets, Image Compression, Filter.**

I. Introduction

The simplist way to understand wavelets is to realize that we are trying to represent a signal (i.e. a sequence of numbers) by a different set of numbers that yields more information about the original signal.

Most people are likely familiar with taking fourier transforms of a signal. In taking a fourier transform, we take a signal, $y(x)$,and represent it by a sequence of coefficients that depend on wave numbers, $y(\omega)$.

$$y(\omega) = \sum y(x) e^{i\omega x}$$

Here, wave numbers give information about "periodicity" of a signal. A wavelet transform on the other hand gives information about the overall shape of a function and the local shape of the function on a finer scale.

II. Representation of a Signal by Haar Wavelets

We first discuss a one-dimensional wavelet transform of a signal using `Haar wavelets` (The simplest form of a wavelet). This is best illustrated by an example.

Example

Consider a simple signal, $x = [1, 2, 4, 8]$. By averaging the signal pairwise

$$\frac{x(2)+x(1)}{2} = \frac{2+1}{2} = 1.5 \text{ and } \frac{x(4)+x(3)}{2} = \frac{8+4}{2} = 6$$

and by computing the differences between the same pairs

$$\frac{x(2)-x(1)}{2} = \frac{2-1}{2} = 0.5 \text{ and } \frac{x(4)-x(3)}{2} = \frac{8-4}{2} = 2$$

We can represent the original signal $x = [1, 2, 4, 8]$ by the averages and differences as $\tilde{x} = [1.5, 6, 0.5, 2]$. The skeptic may question whether we can recover the original signal from our new representation.

$$\begin{aligned} x(1) &= \tilde{x}(1) - \tilde{x}(3) = 1.5 - 0.5 = 1 \\ x(2) &= \tilde{x}(1) + \tilde{x}(3) = 1.5 + 0.5 = 2 \\ x(3) &= \tilde{x}(2) - \tilde{x}(4) = 6 - 2 = 4 \\ x(4) &= \tilde{x}(2) + \tilde{x}(4) = 6 + 2 = 8 \end{aligned}$$

Notice that no new information has been gained or lost with this process: x has 4 coefficients, and the wavelet transform $\tilde{x}$ has 4 coefficients.

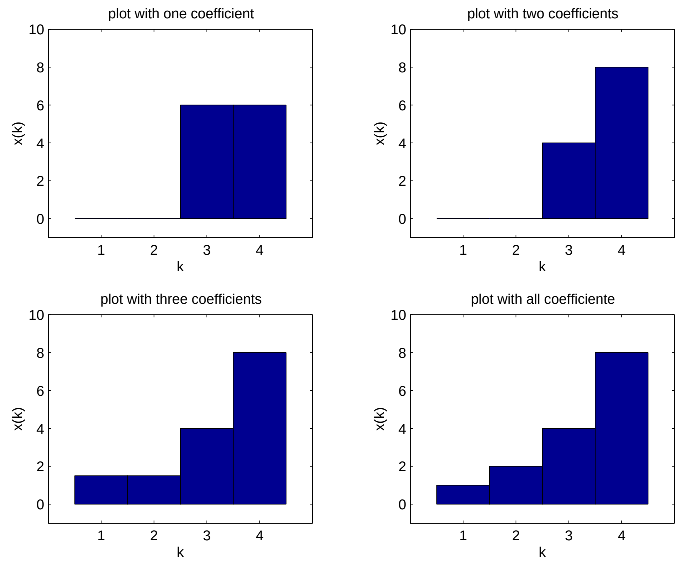

Fig. 1. Reconstructing the image using subsets of the given coefficients. Notice that using all the coefficients results in perfect reconstruction

One advantage of using wavelet transforms in image storage, is that a large number of the wavelet coefficients tend to be very small numbers. Removing these small numbers often produce small errors in the actual representation of the image, allowing for good image compression. (fig 1)

How do we vectorize this "averaging" and "detailing" operation? Consider what happens when the following operator H_0 acts on a signal.

$$\begin{aligned} H_0\,\vec{x} &= \frac{1}{2}\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & . \\ 0 & 1 & 1 & 0 & 0 & . \\ 0 & 0 & 1 & 1 & 0 & . \\ 0 & 0 & 0 & 1 & 1 & . \\ . & . & . & . & . & . \end{bmatrix} \begin{bmatrix} x(1) \\ x(2) \\ x(3) \\ x(4) \\ \ldots \end{bmatrix} \quad (1) \\ &= \frac{1}{2}\begin{bmatrix} x(1)+x(2) \\ x(2)+x(3) \\ x(3)+x(4) \\ \ldots \end{bmatrix} \end{aligned}$$

Notice that every other row gives the desired average for individual pairs, so defining a down-sampling operator D

$$\begin{aligned} D\vec{u} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & . \\ 0 & 0 & 1 & 0 & 0 & . \\ 0 & 0 & 0 & 0 & 1 & . \\ . & . & . & . & . & . \end{bmatrix} \begin{bmatrix} u(1) \\ u(2) \\ u(3) \\ u(4) \\ u(5) \\ \ldots \end{bmatrix} \quad (2) \\ &= [u(1), u(3), u(5), \ldots]^T \end{aligned}$$

The operation $(DH_0)x$ gives the averaging coefficients for a signal x which has an even number of terms.

Defining the operator H_1 as

$$H_1\,\vec{x} = \frac{1}{2}\begin{bmatrix} -1 & 1 & 0 & 0 & . \\ 0 & -1 & 1 & 0 & . \\ 0 & 0 & -1 & 1 & . \\ . & . & . & . & . \end{bmatrix}\begin{bmatrix} x(1) \\ x(2) \\ x(3) \\ \cdots \end{bmatrix} \quad (3)$$

$$= \frac{1}{2}\begin{bmatrix} x(2)-x(1) \\ x(3)-x(2) \\ x(4)-x(3) \\ \cdots \end{bmatrix}$$

The operation $(DH_1)x$ gives the detail coefficients for a signal x. Schematically,