

## Lecture 31: Change of basis; image compression

We've learned that computations can be made easier by an appropriate choice of basis. One application of this principle is to image compression. Lecture videos, music, and other data sources contain a lot of information; that information can be efficiently stored and transmitted only after we change the basis used to record it.

### Compression of images

Suppose one frame of our lecture video is 512 by 512 pixels and that the video is recorded in black and white. The camera records a brightness level for each of the  $(512)^2$  pixels; in this sense, each frame of video is a vector in a  $(512)^2$  dimensional vector space.

The standard basis for this space has a vector for each pixel. Transmitting the values of all  $(512)^2$  components of each frame using the standard basis would require far too much bandwidth, but if we change our basis according to the JPEG image compression standard we can transmit a fairly good copy the video very efficiently.

For example, if we're reporting light levels pixel by pixel, there's no efficient way to transmit the information "the entire frame is black". However, if one of our basis vectors corresponds to all pixels having the same light level (say 1), we can very efficiently transmit a recording of a blank blackboard.

Along with a vector of all 1's, we might choose a basis vector that alternates 1's and  $-1$ 's, or one that's half 1's and half  $-1$ 's corresponding to an image that's bright on the left and dark on the right. Our choice of basis will directly affect how much data we need to download to watch a video, and the best choice of basis for algebra lectures might differ from the best choice for action movies!

### Fourier basis vectors

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ \omega \\ \omega^2 \\ \omega^3 \\ \omega^4 \\ \omega^5 \\ \omega^6 \\ \omega^7 \end{bmatrix}, \begin{bmatrix} 1 \\ \omega^2 \\ \omega^4 \\ \omega^6 \\ \omega^8 \\ \omega^{10} \\ \omega^{12} \\ \omega^{14} \end{bmatrix}, \dots, \begin{bmatrix} 1 \\ \omega^7 \\ \omega^{14} \\ \omega^{21} \\ \omega^{28} \\ \omega^{35} \\ \omega^{42} \\ \omega^{49} \end{bmatrix}$$

The best known basis is the Fourier basis, which is closely related to the Fourier matrices we studied earlier. The basis used by JPEG is made up of cosines – the real parts of  $\omega^{jk}$ .

This method breaks the 512 by 512 rectangle of pixels into blocks that are 8 pixels on a side, each block containing 64 pixels total. The brightness information for those pixels is then compressed, possibly by eliminating all coefficients

below some threshold chosen so that we can hardly see the difference once they're gone.

$$\text{signal } \mathbf{x} \xrightarrow{\text{lossless}} 64 \text{ coefficients } c \xrightarrow{\text{lossy compression}} \hat{c} \text{ (many zeros)} \longrightarrow \hat{\mathbf{x}} = \sum \hat{c}_i \mathbf{v}_i$$

In video, not only should we consider compressing each frame, we can also consider compressing sequences of frames. There's very little difference between one frame and the next. If we do it right, we only need to encode and compress the differences between frames, not every frame in its entirety.

### The Haar wavelet basis

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \dots, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}$$

The closest competitor to the JPEG encoding method uses a wavelet basis. (JPEG2000 improves on the Haar wavelets above.) In Haar's wavelet basis for  $\mathbb{R}^8$ , the non-zero entries are half 1's and half  $-1$ 's (except for the vector of all 1's). However, half or even three quarters of a basis vector's entries may be 0. These vectors are chosen to be orthogonal and can be adjusted to be orthonormal.

### Compression and matrices

Linear algebra is used to find the coefficients  $c_i$  in the change of basis from the standard basis (light levels for each pixel) to the Fourier or wavelet basis. For example, we might want to write:

$$\mathbf{x} = c_1 \mathbf{w}_1 + \dots + c_8 \mathbf{w}_8.$$

But this is just a linear combination of the wavelet basis vectors. If  $W$  is the matrix whose columns are the wavelet vectors, then our task is simply to solve for  $\mathbf{c}$ :

$$\mathbf{x} = W \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_8 \end{bmatrix}.$$

So  $\mathbf{c} = W^{-1}\mathbf{x}$ .

Our calculations will be faster and easier if we don't have to spend a lot of time inverting a matrix (e.g. if  $W^{-1} = W^T$ ) or multiplying by the inverse. So in the field of image compression, the criteria for a good basis are:

- Multiplication by the basis matrix and its inverse is fast (as in the FFT or in the wavelet basis).
- Good compression – the image can be approximated using only a few basis vectors. Most components  $c_i$  are small – safely set to zero.

## Change of basis

### Vectors

Let the columns of matrix  $W$  be the basis vectors of the new basis. Then if  $\mathbf{x}$  is a vector in the old basis, we can convert it to a vector  $\mathbf{c}$  in the new basis using the relation:

$$\mathbf{x} = W\mathbf{c}.$$

### Transformation matrices

Suppose we have a linear transformation  $T$ . If  $T$  has the matrix  $A$  when working with the basis  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_8$  and  $T$  has the matrix  $B$  when working with the basis  $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_8$ , it turns out that  $A$  and  $B$  must be similar matrices. In other words,  $B = M^{-1}AM$  for some change of basis matrix  $M$ .

Reminder: If we have a basis  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_8$  and we know  $T(\mathbf{v}_i)$  for each  $i$ , then we can use the fact that  $T$  is a linear transformation to find  $T(\mathbf{v}) = c_1T(\mathbf{v}_1) + c_2T(\mathbf{v}_2) + \dots + c_8T(\mathbf{v}_8)$  for any vector  $\mathbf{v} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \dots + c_8\mathbf{v}_8$  in the space. The entries of column  $i$  of the matrix  $A$  are the coefficients of the output vector  $T(\mathbf{v}_i)$ .

If our basis consists of eigenvectors of our transformation, i.e. if  $T(\mathbf{v}_i) = \lambda_i\mathbf{v}_i$ , then  $A = \Lambda$ , the (diagonal) matrix of eigenvalues. It would be wonderful to use a basis of eigenvectors for image processing, but finding such a basis requires far more computation than simply using a Fourier or wavelet basis.

### Summary

When we change bases, the coefficients of our vectors change according to the rule  $\mathbf{x} = W\mathbf{c}$ . Matrix entries change according to a rule  $B = M^{-1}AM$ .