# Image and video processing

**Cousera Course:**

Image and video processing: From Mars to Hollywood with a stop at the hospital, Pof. Guillermo Sapiro, Duke University, 2015

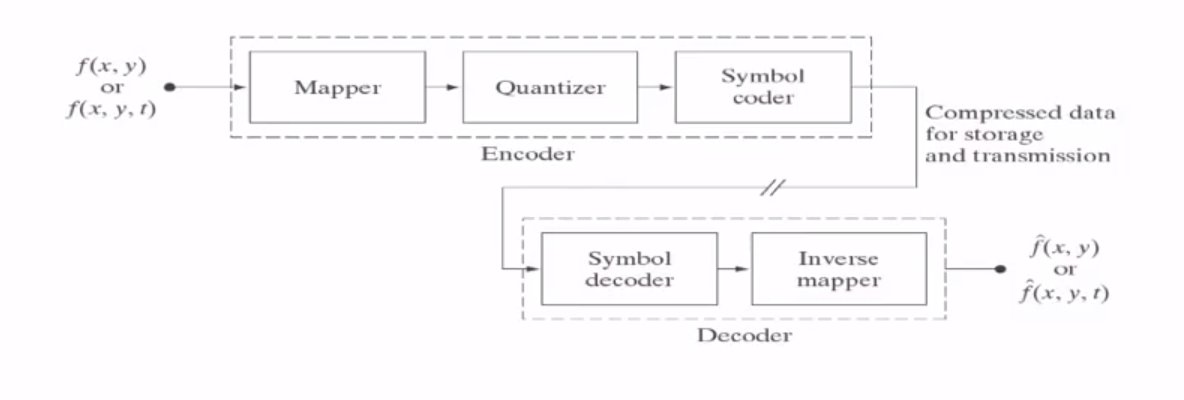
**Ch1 Image and Video compression**

1. **Image and video compression overview:**

Image: JPEG,JPEG-LS

Video: MPEG

1. **Image Compression: JPEG**
   1. Flow Chart of JPEG

****

* 1. Mapper:

一般使用DCT當作當做mapper的工具，其中有幾個好處

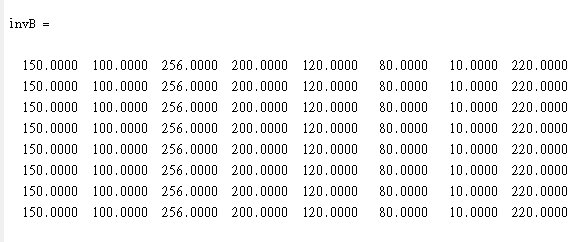
1. 正反轉換函數相同:

\begin{displaymath}
F(u,v) = \left(\frac{2}{N}\right)^{\frac{1}{2}}
\left(\frac{...
 ...}(2i+1)
\right]cos\left[ \frac{\pi.v}{2.M}(2j+1) \right].f(i,j)\end{displaymath}

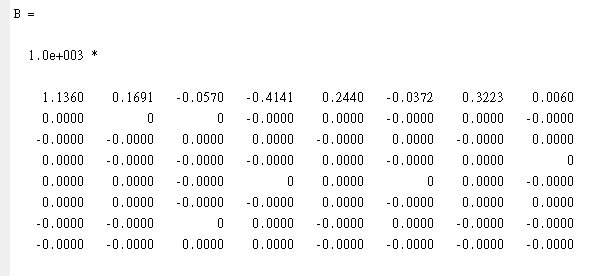
\begin{displaymath}
\Lambda(\xi) = \left\{ \begin{array}
{ll} \frac{1}{\sqrt{2}} & {\rm
for}
\xi = 0 \ 1 & {\rm otherwise}\end{array} \right.\end{displaymath}

1. 有效壓縮資料：

以8X8的subimage為例：

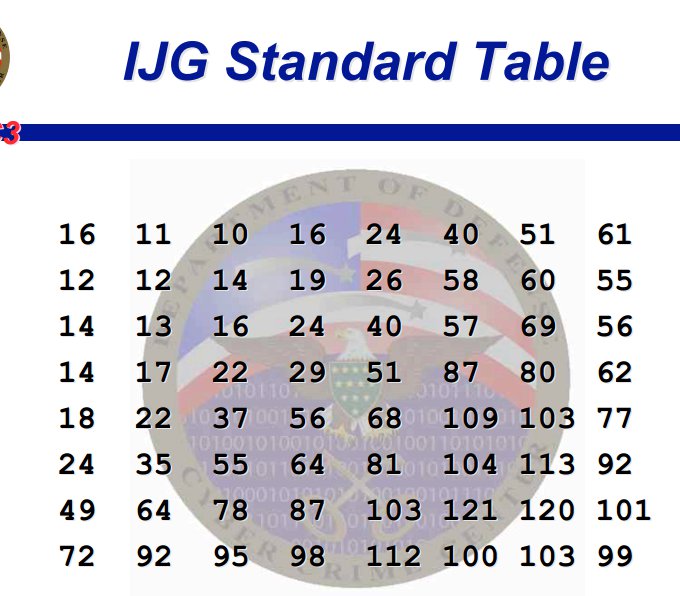


B = DCT(invB)

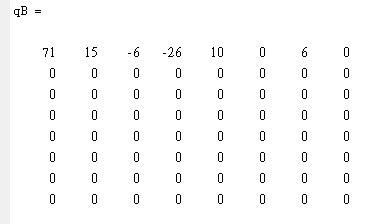


可以看出經過DCT轉換後大部分的elements皆為零，能夠壓縮資料量。

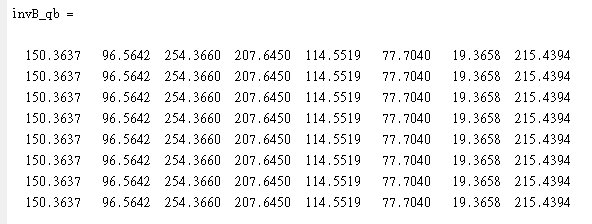
* 1. Quantizer:



將B的值除以**Quantizer Table上**相對應的值，可以得到最後的結果，然後取整數，能夠有效的將壓縮數據，降低數據變化性，方便執行接下來的Symbol code，解碼時則是乘上table上的值。最後在如下所示



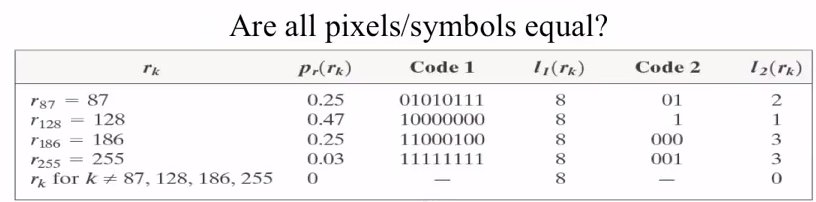
比較qB與B的值可以發現，值會出現誤差，JPEG壓縮主要的誤差就是來自這一步。若將qB解碼可得到：



可以看出來與原本的invB不太一樣，但是誤差並不會太大。

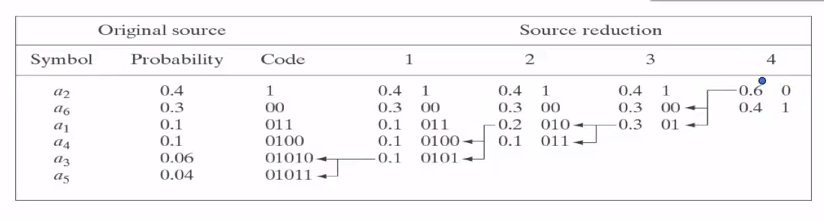
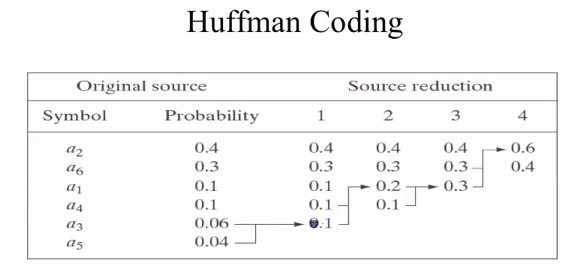
* 1. Symbol code: Huffman Coding
     1. Intuition:

將整張image做quantizer的結果做Histogram可以得到

Histogram，與每個intensity(或是轉換過後數值的結果)的機率分布，如下：

可以發現有些數值數量非常少，有些非常多，但是都必須佔8bit的空間，因此Huffman Coding的構想即是讓分布較多的數值用較少的bit表示，分布較少的數值用較大的bit表示，而且基本上都小於8bit。

* + 1. Method:



* + 1. 估測Huffman coding的節省記憶體的程度:

Entropy