

---

# VC 11/12 – T12

## Video Compression

Mestrado em Ciência de Computadores  
Mestrado Integrado em Engenharia de Redes e  
Sistemas Informáticos

*Miguel Tavares Coimbra*

---

# Outline

---

- The need for compression
- Types of redundancy
- Image compression
- Video compression

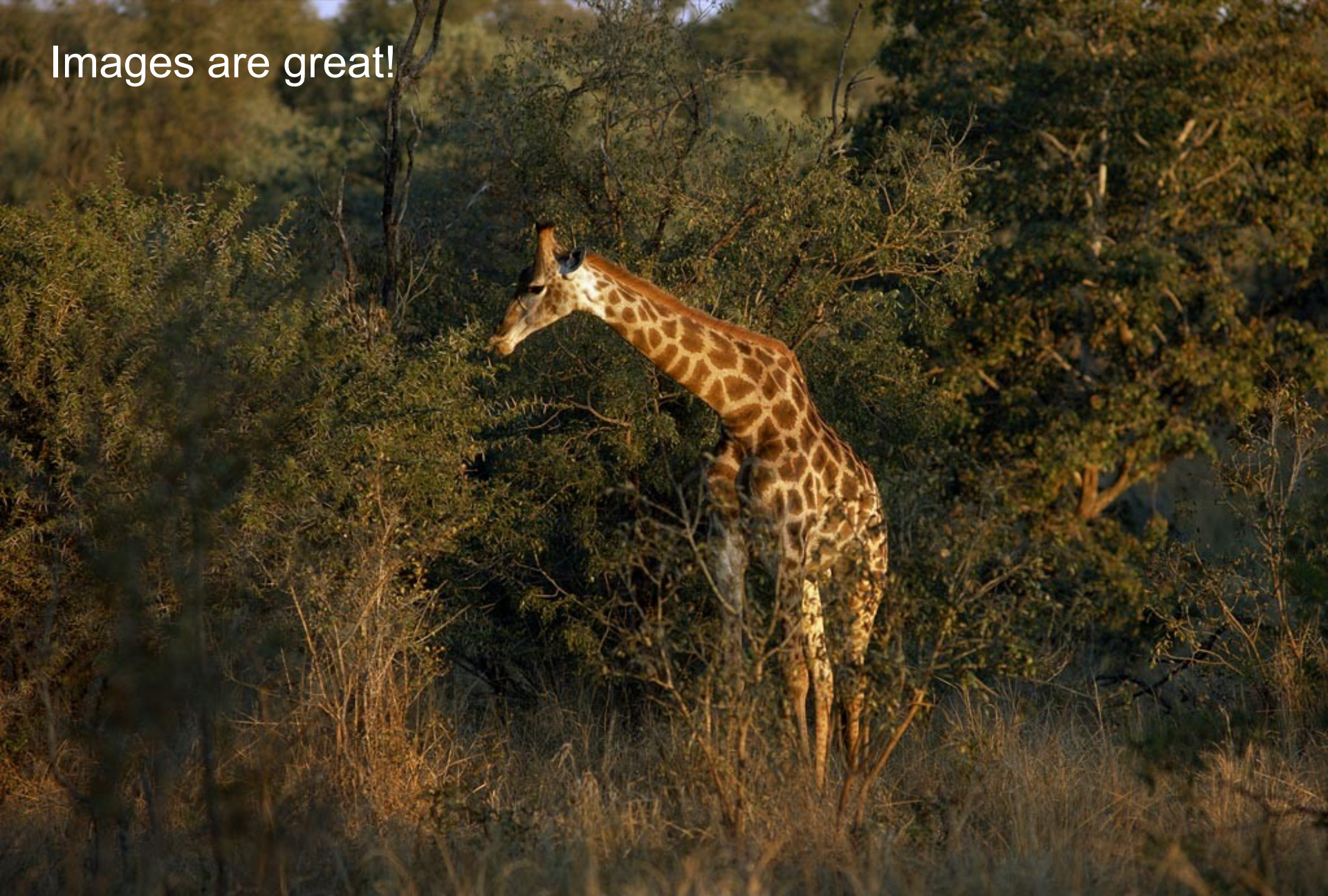
---

# Topic: The need for compression

---

- The need for compression
- Types of redundancy
- Image compression
- Video compression

Images are great!



Giraffe in Mala Mala Game Reserve, South Africa  
Photograph by James P. Blair



But...

Images need storage space...

A lot of space!

Size: 1024 x 768 pixels  
RGB colour space  
8 bits per color  
= 2,6 MBytes

---

# What about video?

---

- VGA: 640x480, 3 bytes per pixel -> 920KB per image.
- Each second of video: 23 MB
- Each hour of video: 83 GB



The death of Digital Video

---

# What if... ?

---

- We exploit redundancy to compress image and video information?
  - Image Compression Standards
  - Video Compression Standards
- “Explosion” of Digital Image & Video
  - Internet media
  - DVDs
  - Digital TV
  - ...

---

# Compression

---

- **Data compression**
  - Reduce the quantity of **data** needed to store the same **information**.
  - In computer terms: Use fewer **bits**.
- **How is this done?**
  - Exploit data **redundancy**.
- **But don't we lose information?**
  - Only if you want to...



---

# Types of Compression

---

- **Lossy**

- We do not obtain an exact copy of our compressed data after decompression.
- Very high compression rates.
- Increased degradation with successive compression / decompression.

- **Lossless**

- We obtain an exact copy of our compressed data after decompression.
- Lower compression rates.
- Freely compress / decompress images.

It all depends on what we need...

---

# Topic: Types of redundancy

---

- The need for compression
- **Types of redundancy**
- Image compression
- Video compression

---

# Coding Redundancy

---

- **Information Theory**
  - The most common values should be encoded with fewer bits.
- **Huffman coding**
  - Smallest possible number of code symbols per source symbols.
  - Lossless.
- **LZW coding**
  - Creates additional values for common sequences of values (e.g. sequence of black pixels).
  - GIF, TIFF, PDF.
  - Exploits the spatial redundancy of images!

---

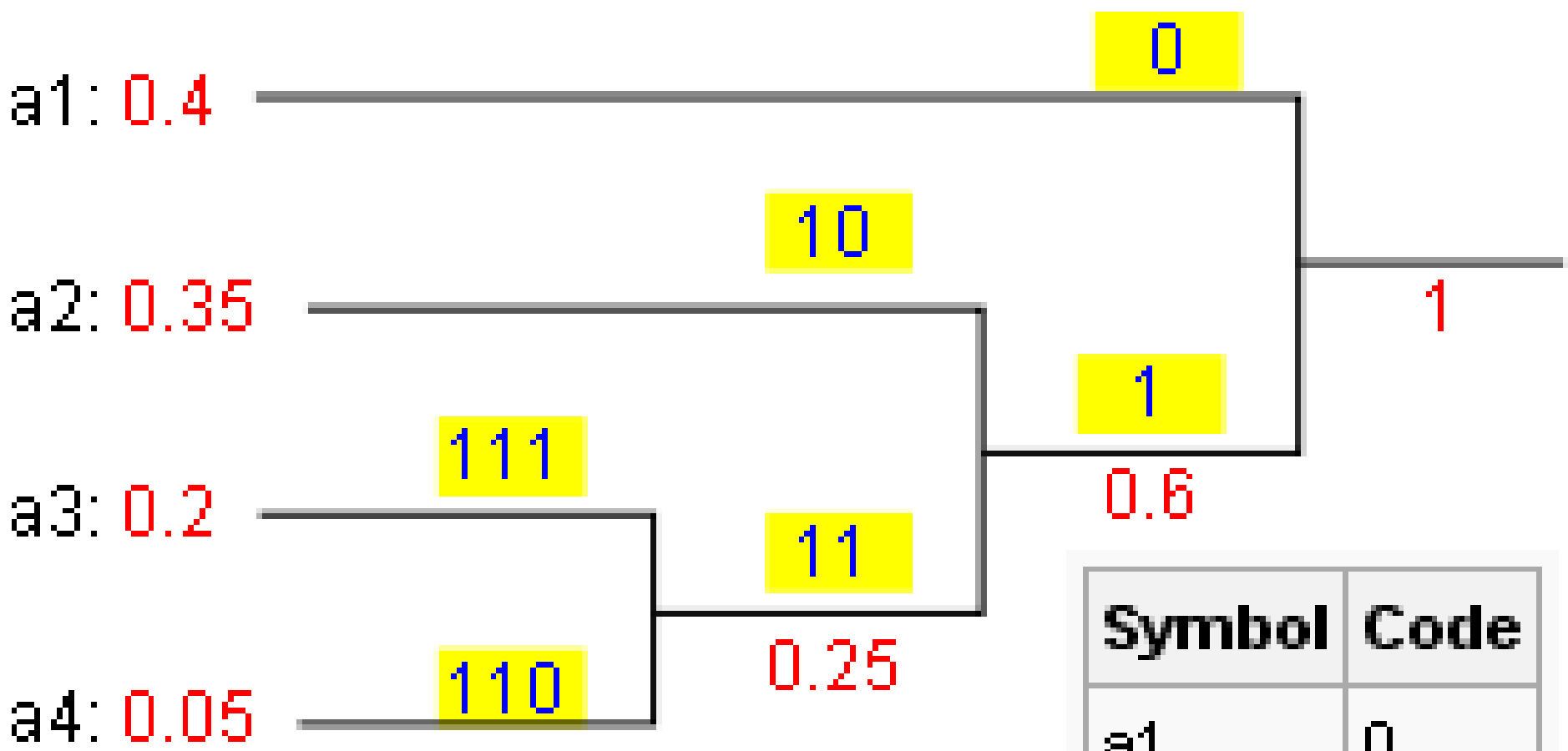
# Huffman Coding

---

- Developed by David A. Huffman while he was a Ph.D. student at MIT.
  - Variable-length code.
  - Entropy encoding algorithm.
- Optimal for a symbol-by-symbol coding.
- Lossless.

[http://en.wikipedia.org/wiki/Huffman\\_coding](http://en.wikipedia.org/wiki/Huffman_coding)





Symbol	Code
a1	0
a2	10
a3	111
a4	110

Normal representation: 2 bits/symbol  
 Entropy of the source: 1.73 bits/symbol  
 Huffman code: 1.83 bits/symbol

---

# Spatial Redundancy

---



How spatially redundant is this ... Image?



What about this one?

---

# How to exploit this?

---

- **Correlation between neighboring pixels.**
  - E.g. A white line can be coded with two numbers: [nr. Pixels; colour].
- **Mathematics:**
  - Lossless
    - LZW Coding – GIF
    - ...
  - Lossy
    - The DCT Transform – JPEG
    - ...



---

# LZW Coding (Lempel-Ziv-Welch)

---

- **In a nutshell:**
  - Uses a *string translation table*.
  - Maps *fixed length codes* to *strings*.
- **Why is this great for images?**
  - ‘Imagine’ pixels as chars.
  - Common sequences of pixels are mapped by a single code.
  - How many codes are needed to represent a white line?

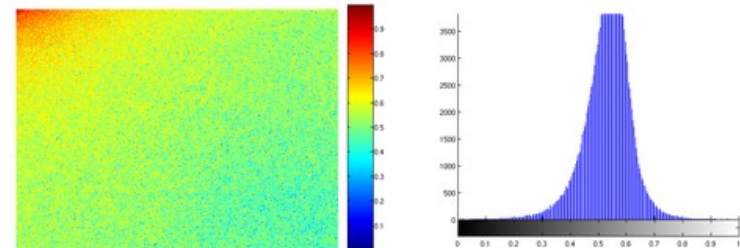
<http://en.wikipedia.org/wiki/LZW>

# Discrete Cosine Transform (DCT)

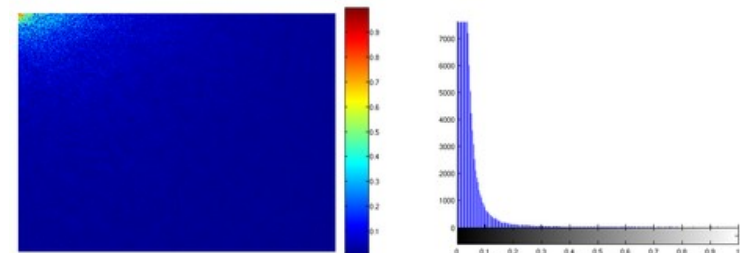
- Can be seen as a ‘cut-down’ version of the DFT:
  - Use only the ‘real’ part but...
  - Has double the resolution so...
  - It has the same number of coefficients.
- Why do we use it?



**DFT**

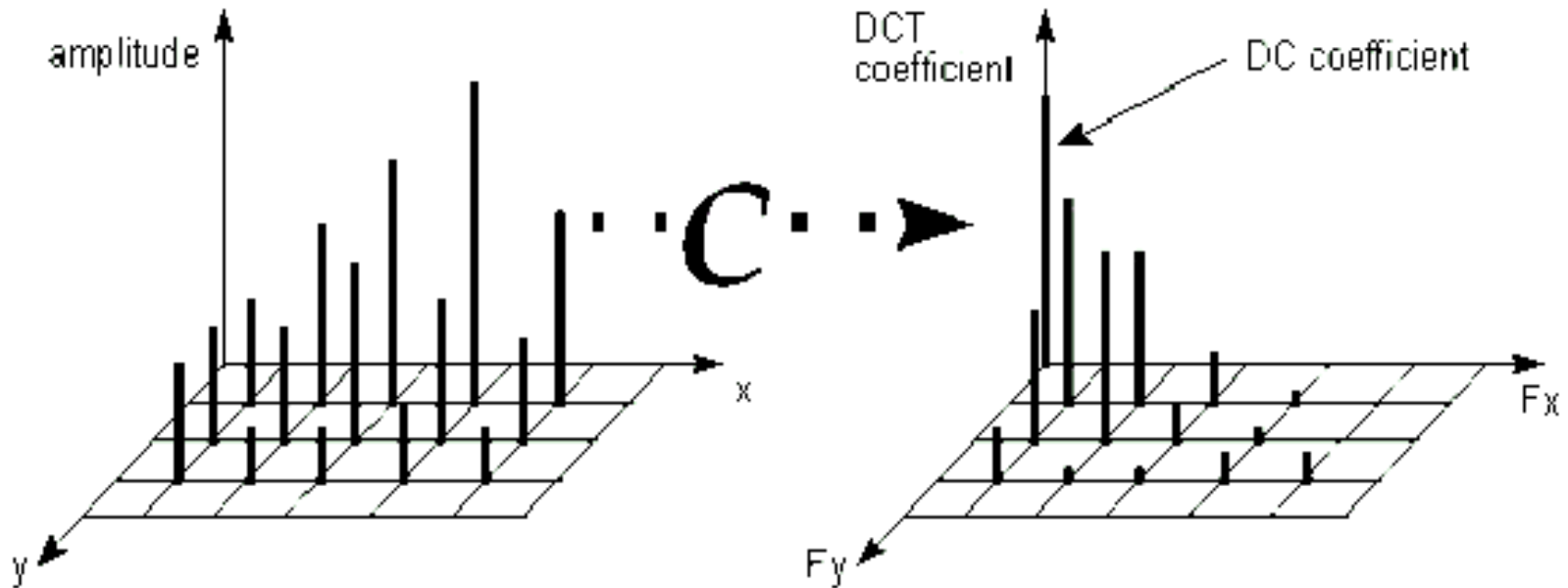


**DCT**

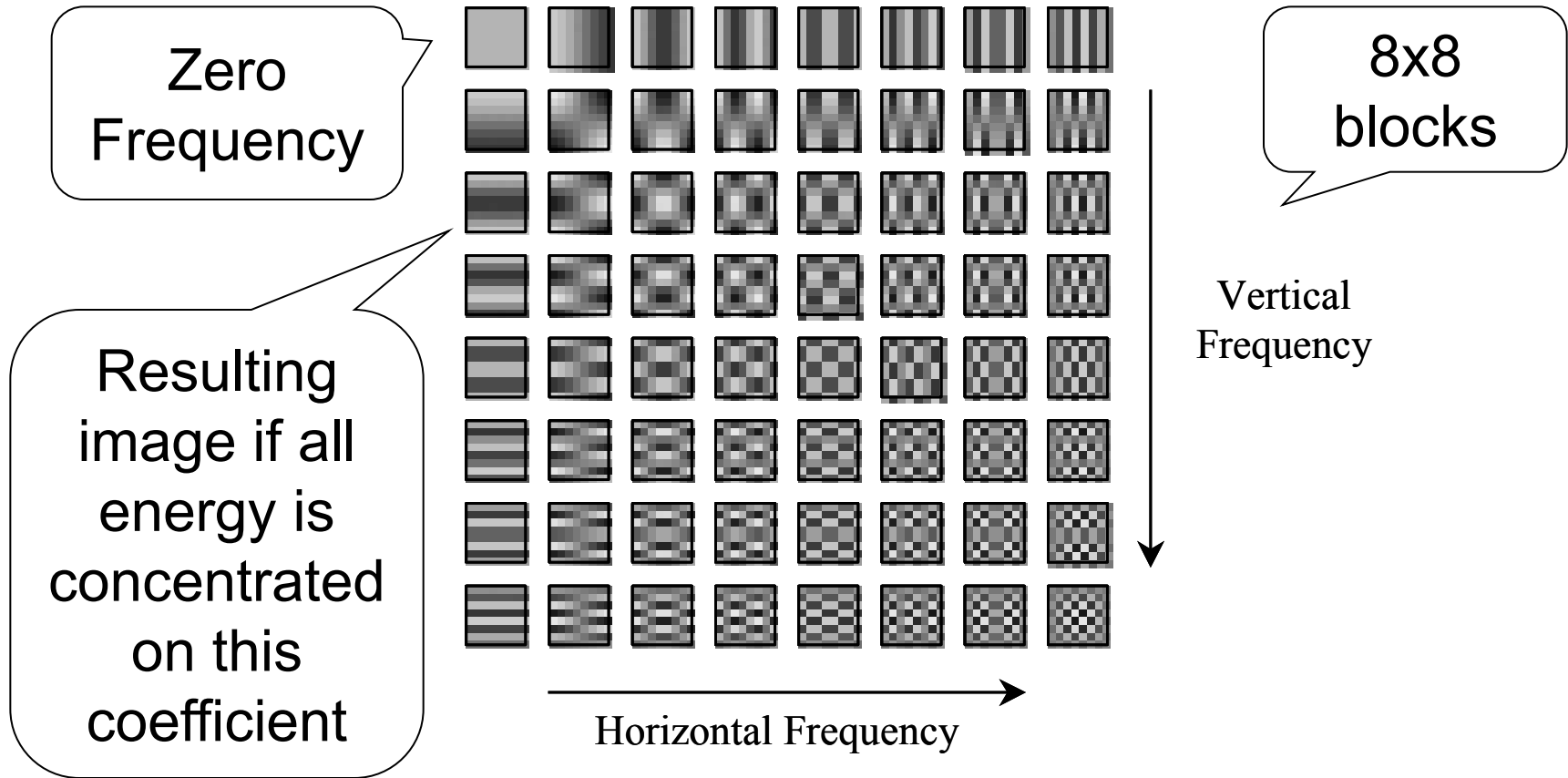


# Why DCT?

- Energy 'compacting potential' superior to DFT.



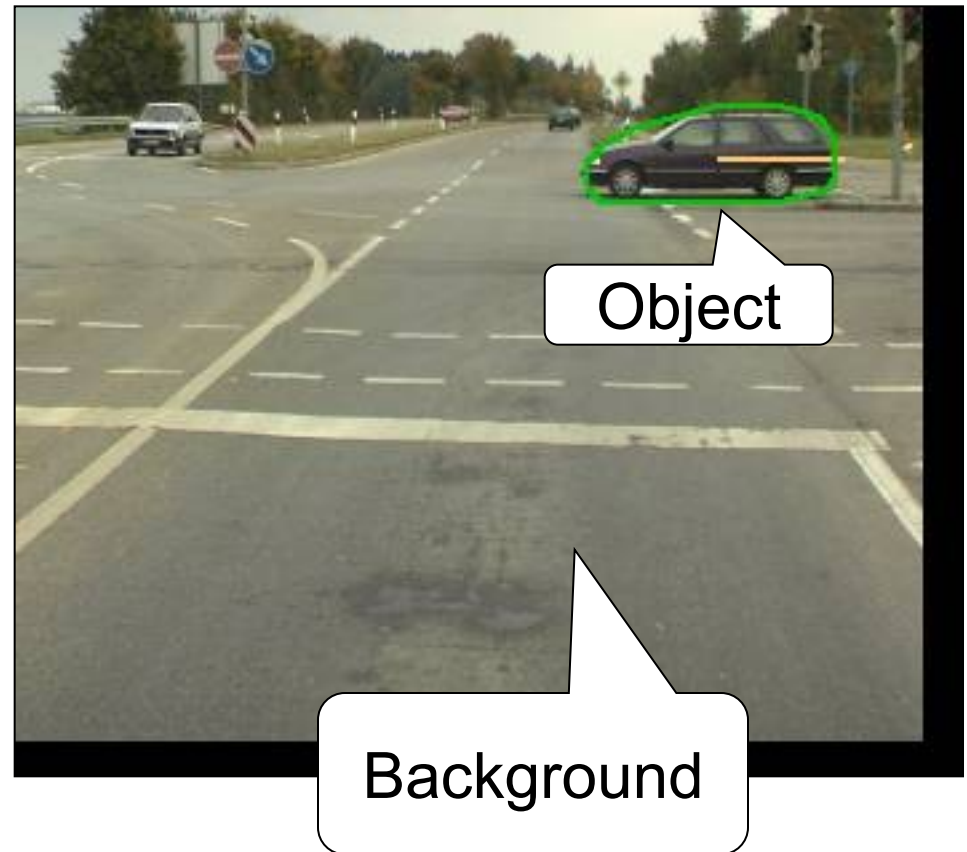
# Visual significance of coefficients





# Temporal redundancy

- Consecutive images of a video stream do not vary much.
  - Some areas don't change at all (background).
  - Others only change their spatial location (moving objects).



---

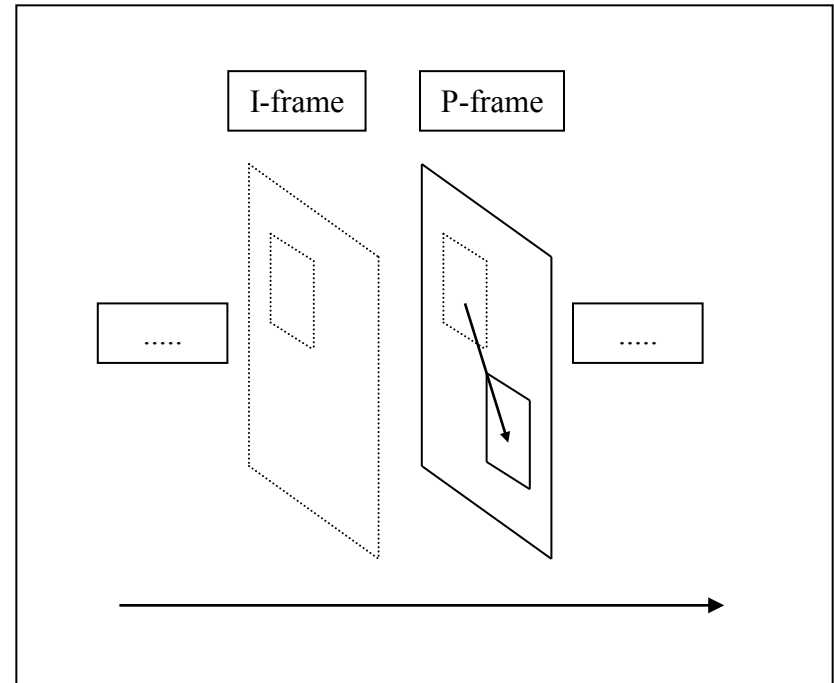
# How do we exploit this?

---

- **Send image differences**
  - Consecutive images are very similar.
  - **Difference images** are spatially much more redundant than real images.
  - Exploit spatial redundancy of difference images!
- **Motion vectors**
  - What if the camera moves?
  - What if objects move?
  - Use **motion estimation** before calculating the difference image!

# Motion estimation

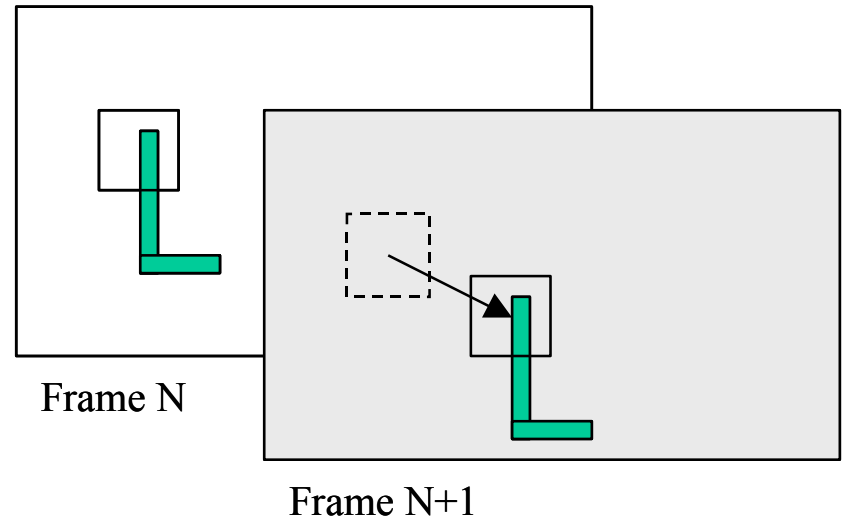
- Tries to find where an area of the image was in a previous image.
- Objective:
  - Minimize the difference between these two blocks.
- In fact:
  - We don't really care whether this is the same object or not...



Obtains **Motion Vectors**

# Block Matching

- Search for a similar block in a neighboring region
  - Full search is too expensive. Variations: 3SS [Koga81], LogS [JJ81], N3SS [Li94], 4SS [PM96],...
  - Various cost functions used: MAD, MSD, CCF, PDC,...
- Noisy approximation to optical flow.
- Aperture and 'blank wall' problems.
- Confidence measures?



$$MAD(dx, dy) = \frac{1}{mn} \sum_{i=-n/2}^{n/2} \sum_{j=-m/2}^{m/2} |F(i, j) - G(i + dx, j + dy)|$$

$$MSD(dx, dy) = \frac{1}{mn} \sum_{i=-n/2}^{n/2} \sum_{j=-m/2}^{m/2} [F(i, j) - G(i + dx, j + dy)]^2$$

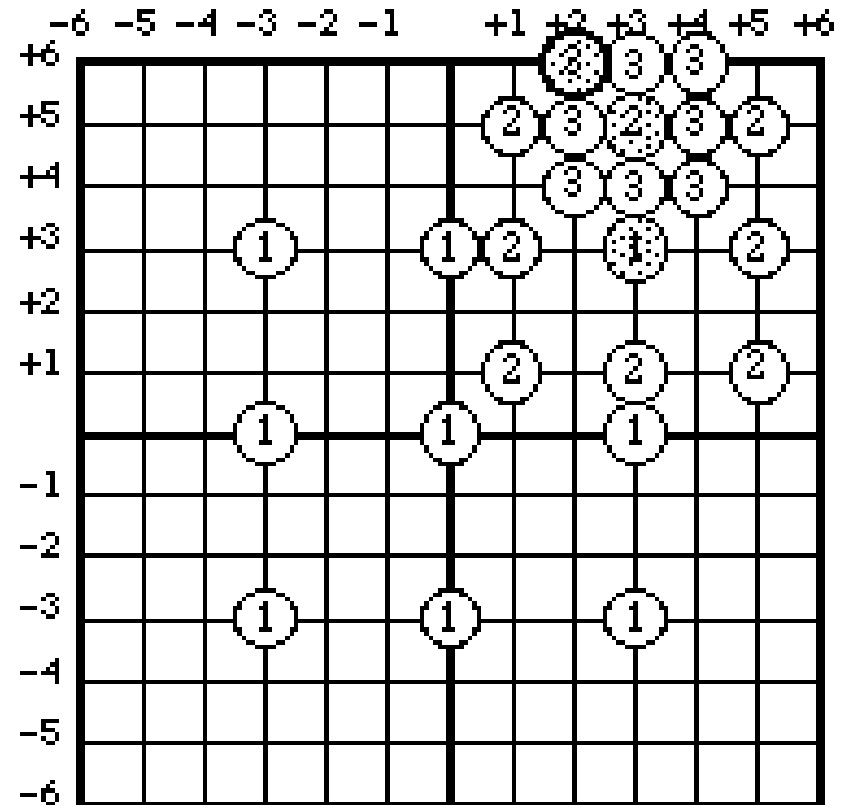
# Three-Step Search (3SS)

- **Algorithm**

- Test 8 points around the centre.
- Choose 'lowest cost'.
- Test 8 points around the new point with a lower step.
- Etc...

- **Very popular**

- Fast.
- Moderate accuracy.
- Easy to implement in hardware.



---

# Psicovisual redundancy

---

- **Human visual system**
  - Different sensitivity to different information.
- **Human processing**
  - We only see some parts of the image.
  - Our brain completes the rest.



---

# Human sensitivity

---

- We notice errors in homogenous regions.
  - Low frequencies.
- We notice errors in edges.
  - High frequencies.
- We don't notice noise in textured areas.
  - Medium frequencies.





---

# Topic: Image compression

---

- The need for compression
- Types of redundancy
- **Image compression**
- Video compression

---

# Lossless Compression

---

- Some types of images are not adequate for lossy compression.
  - Logos
  - Text
  - Medical images (??)
  - Etc.
- Our sensitivity to errors in these situations is too high.

- Some types of images are not adequate for lossy compression.
  - Logos
  - Text
  - Medical images (??)
  - Etc
- Our sensitivity to errors in these situations is too high..

---

# Graphics Interchange Format (GIF)

---

- Lossless.
- 8 bpp format.
- 256 colour palette.
- LZW data compression.
- Popular for logos, text and simple images.
- Allows animations.

[http://en.wikipedia.org/wiki/Image:Rotating\\_earth\\_%28large%29.gif](http://en.wikipedia.org/wiki/Image:Rotating_earth_%28large%29.gif)



---

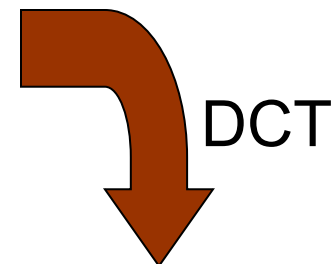
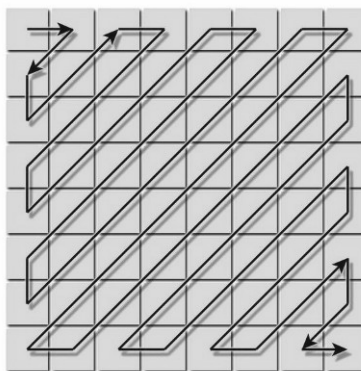
# Lossy Compression

---

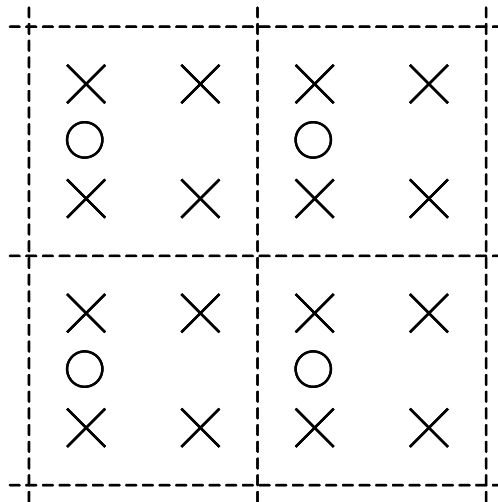
- Acceptable for most real images and situations.
- Very popular: JPEG.
- We can control the level of compression vs. Quality of the resulting image.
- How do we do this?

# Lossy Image Compression

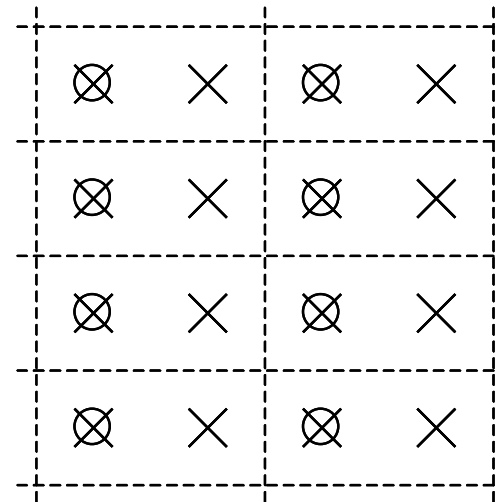
- **Most popular: JPEG**
  - Colour space: YCbCr
  - Colour less important than intensity.
  - DCT.
  - Quantization.
  - *Zig-Zag Run-Length Huffman encoding*

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$


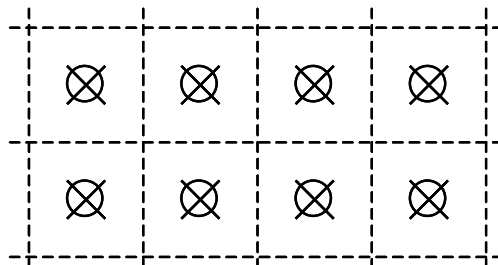
# Chroma Format



4:2:0 Chroma Format



4:2:2 Chroma Format

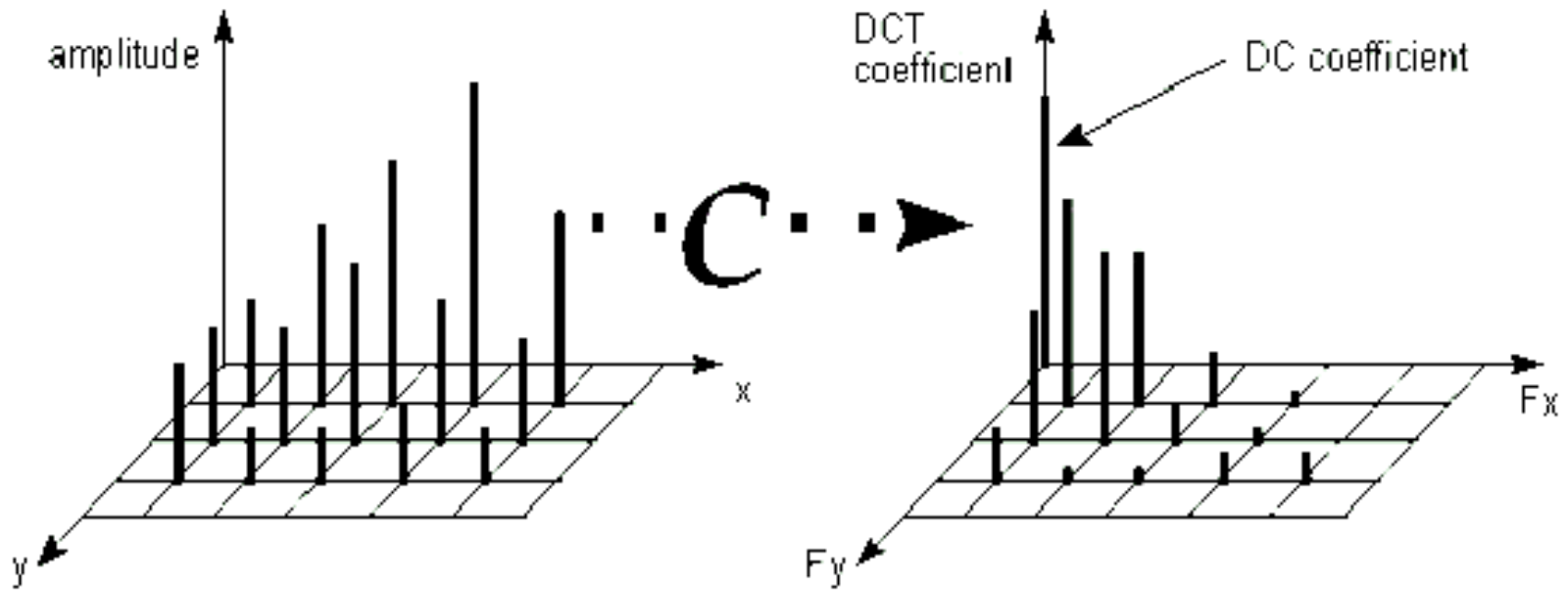


4:4:4 Chroma Format

○ Chrominance  
× Luminance

Psico-visual redundancy

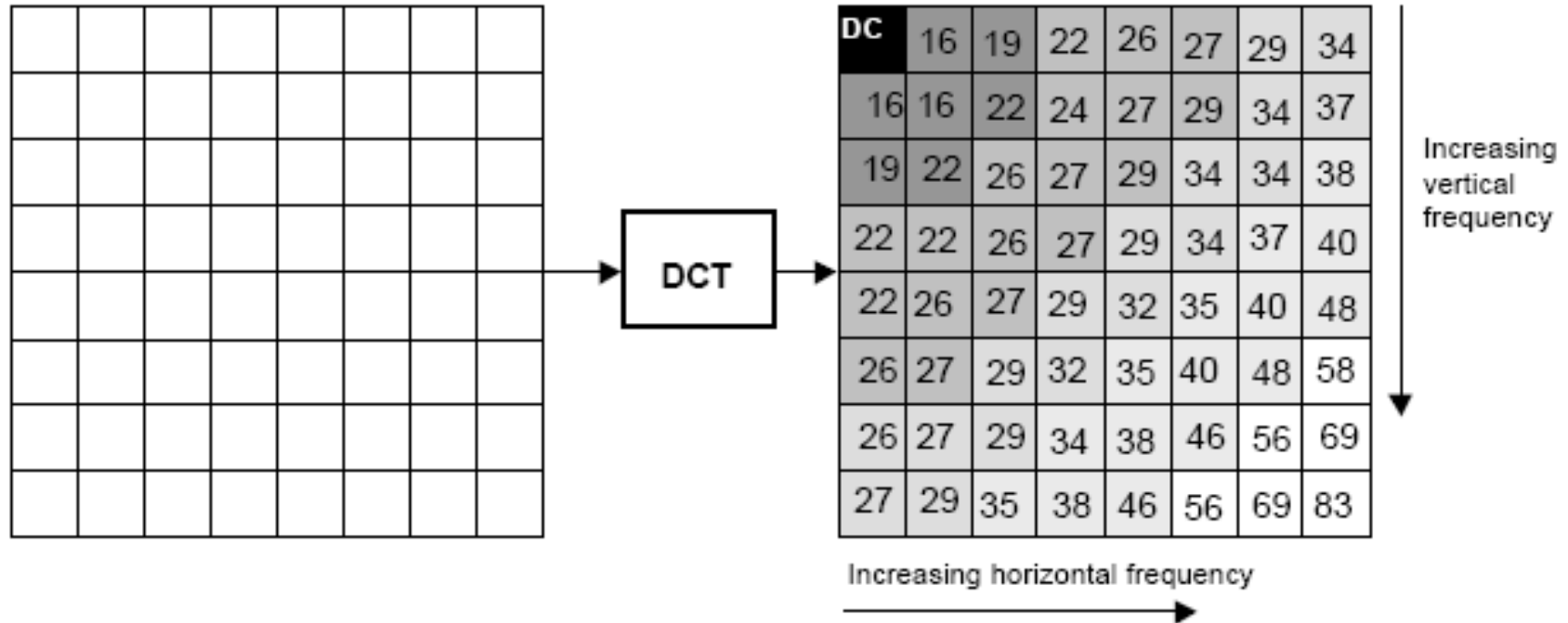
# DCT



Concentrate energy into a smaller number of coefficients



# Quantization



**Lossy Process!**

Give higher importance to low spatial frequencies



---

# Considerations

---

- We can control compression via a **quantization factor**.
- The higher the factor, the higher the number of zeros in the DCT > Better Huffman coding.
- Problem: High quantization factors produce **compression artifacts**.



Small compression



Medium compression



High compression

---

# Topic: Video compression

---

- The need for compression
- Types of redundancy
- Image compression
- **Video compression**



---

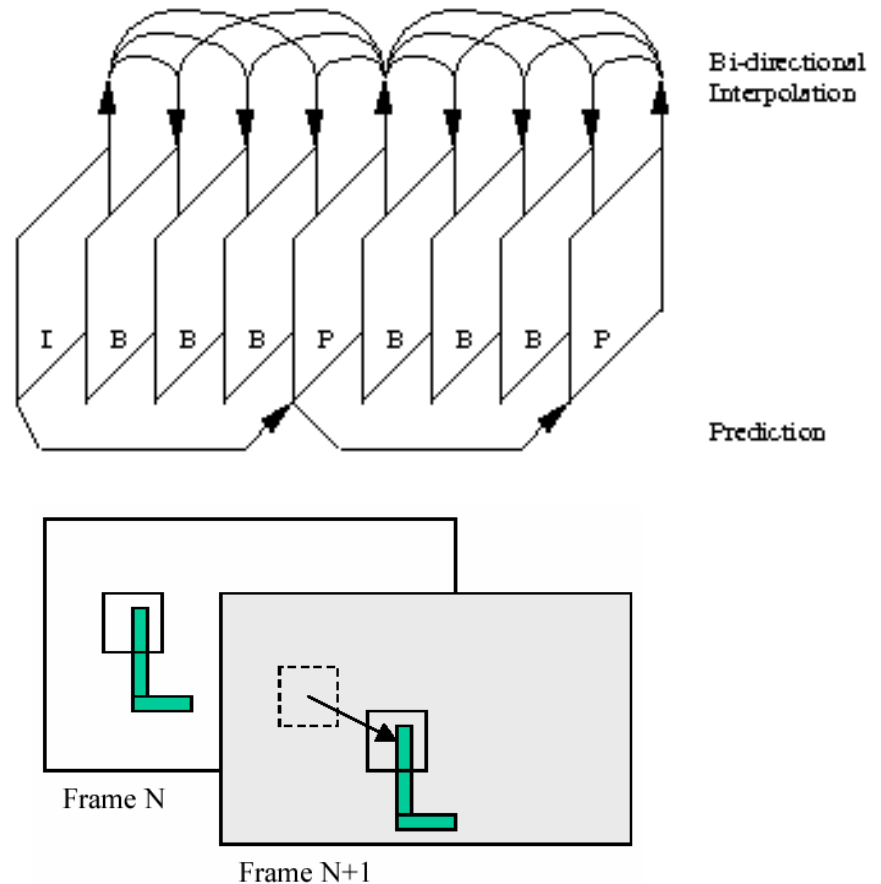
# Exploiting temporal redundancy

---

- Using all other redundancies for JPEG:
  - Compression factor - 10:1
- Exploiting temporal redundancy for MPEG-2:
  - Compression factor – 100:1
- Temporal redundancy is of **vital importance** to video compression!

# Video Compression

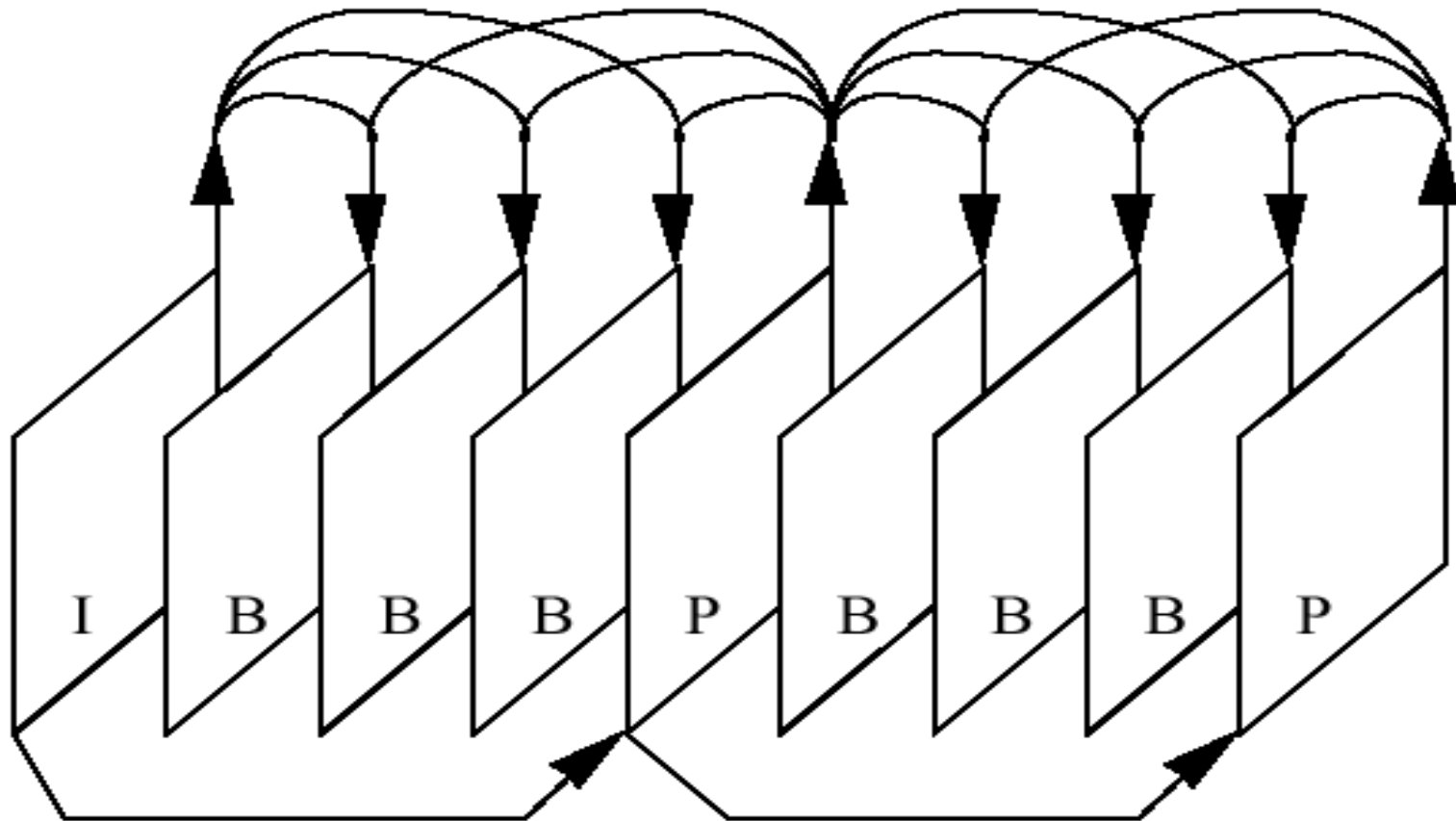
- H.261, H.263, DivX, MPEG-1, ...
- MPEG-2
  - Images compressed as JPEG.
  - Image prediction.
  - Motion estimation.
  - DVD, Digital TV, ...



---

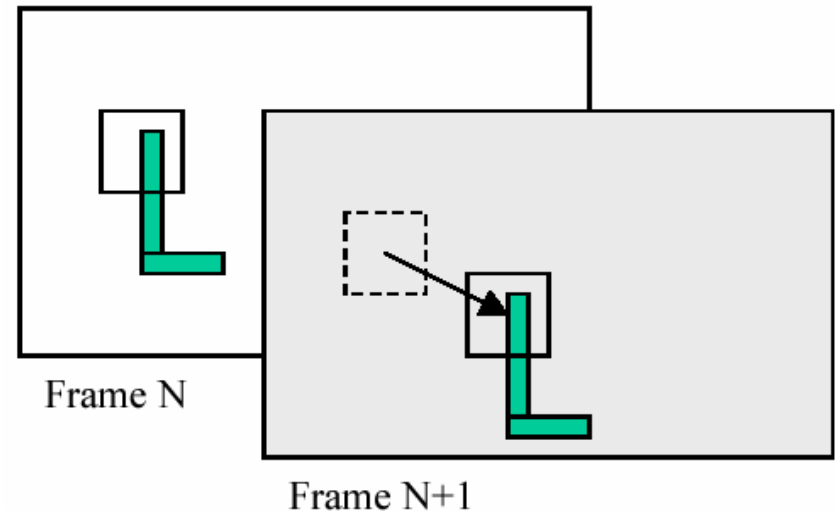
# Intra-frame and Inter-frame prediction

---

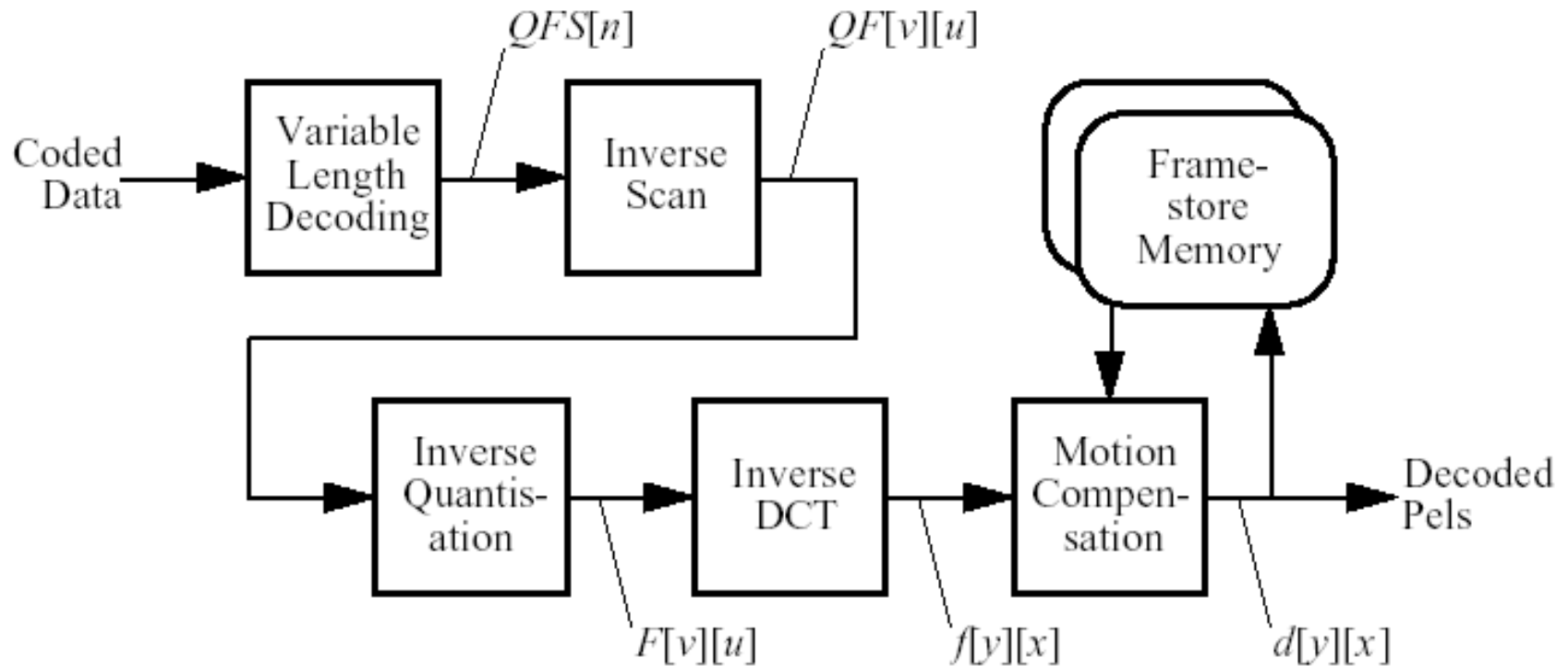


# MPEG Motion estimation

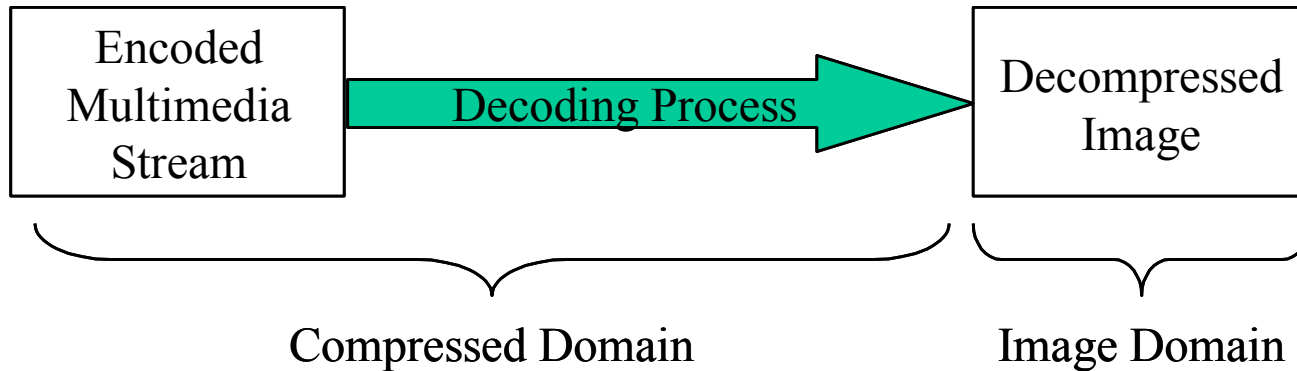
- **Motion vectors:**
  - B Images
  - P Images
- **Point to areas in:**
  - I Images
  - P Images
- **Groups Of Pictures:**
  - Consider error propagation.
  - Consider compression levels.



# Decoder Model



# Compressed Domain Processing



Can't we exploit this information?  
DC Images  
Motion Flow  
...

Decoding Function	Decoding cost (%)
Bit stream header parsing	0.44
Huffman decoding and inverse quantization	19.00
Inverse 8x8 DCT	22.10
Motion compensation	38.64
Colour transformation and display	19.82

---

# Resources

---

- Gonzalez & Woods – Chapter 6
- MPEG Compression -  
<http://mia.ece.uic.edu/~papers/WWW/MultiMediaStandards/chapter7.pdf>
- Image Coding Fundamentals –  
[http://videocodecs.blogspot.com/2007/05/image-coding-fundamentals\\_08.html](http://videocodecs.blogspot.com/2007/05/image-coding-fundamentals_08.html)