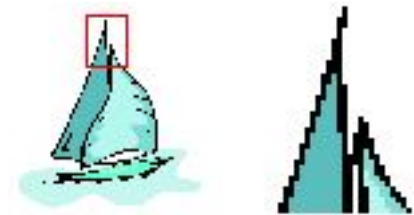


# Image Formats and Files

Jung-Ming Wang  
jmgwang@gmail.com

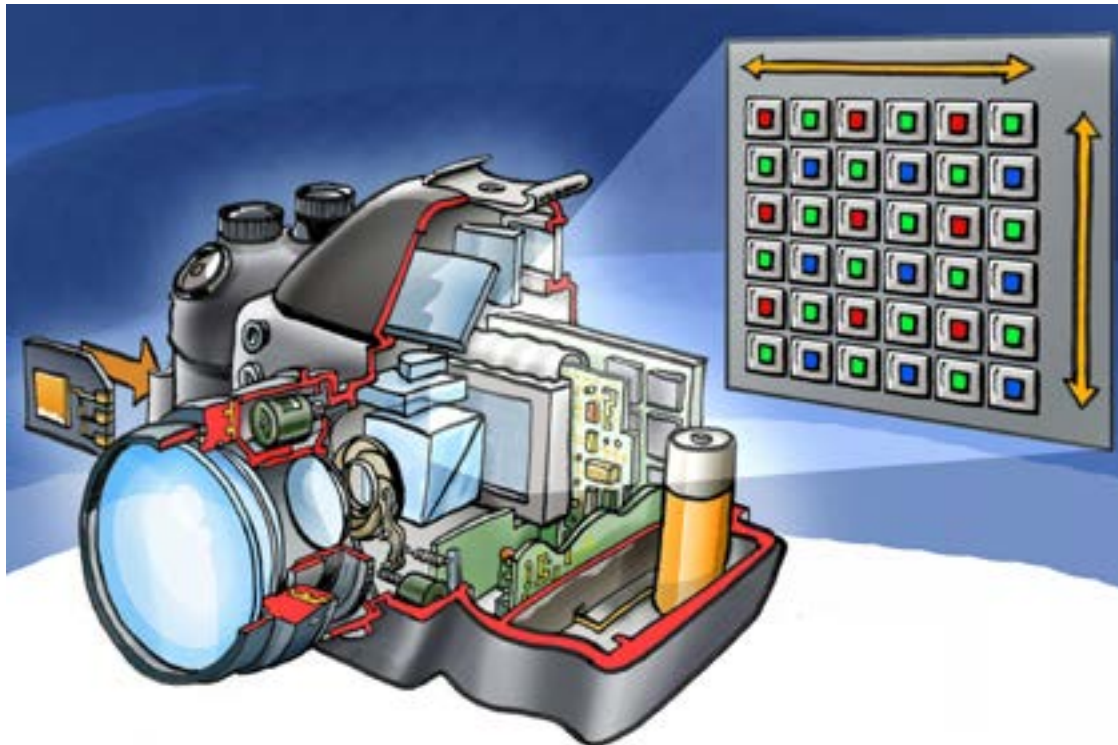
# Vector vs. Raster Images

- Vector images
  - Preserving sharpness after rescaling
  - Not good for natural scenes
  - Hard to construct
- Raster images
  - Contrastively



# Raster Image

- Formed by a digital camera



<http://zh.wikipedia.org/zh-tw/數位影像>

# Pixel Resolution

- N pixels high by M pixels wide  $\rightarrow M \times N$



<http://zh.wikipedia.org/zh-tw/灰階圖像>

# Color Depth

- Number of bits for a single pixel
  - 1-bit color: black and white
  - 8-bit color: gray-scale
  - 24-bit color: true color



# File Format

- Header: characteristics of the image
  - Size, color map, compression method, .....
- Data value:
  - Pixel values, index values
- Common formats
  - BMP, GIF, PNG, JPEG, TIFF .....

# Hexadecimal Dumps

- Image file is often saved as a binary file.
- Binary file can be shown in hexadecimal dump.

Contents:	ASCII:
ff d8 ff e1 4c cf 45 78 69 66 00 00 49 49 2a 00	....L.Exif..II*.
08 00 00 00 0b 00 0e 01 02 00 20 00 00 00 92 00	.....
00 00 0f 01 02 00 05 00 00 00 b2 00 00 00 10 01	.....
02 00 09 00 00 00 b8 00 00 00 12 01 03 00 01 00	.....
00 00 01 00 00 00 1a 01 05 00 01 00 00 00 c2 00	.....
00 00 1b 01 05 00 01 00 00 00 ca 00 00 00 28 01	.....(.
03 00 01 00 00 00 02 00 00 00 32 01 02 00 14 00	.....2.....
00 00 d2 00 00 00 13 02 03 00 01 00 00 00 02 00	.....
00 00 69 87 04 00 01 00 00 00 02 01 00 00 a5 c4	..i.....
07 00 1c 00 00 00 e6 00 00 00 06 09 00 00 20 20	.....
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	
20 20 20 20 20 20 20 20 20 20 20 20 00 53 4f	
4e 59 00 00 44 53 43 2d 50 32 30 30 00 00 48 00	.S0
00 00 01 00 00 00 48 00 00 00 01 00 00 00 32 30	NY..DSC-P200..H.
30 36 3a 31 32 3a 32 33 20 31 38 3a 31 36 3a 35	.....H.....20
	06:12:23 18:16:5

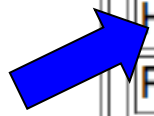
# BMP Format

- BMP header format

<b>Name</b>	<b>Size</b>	<b>Description</b>
Header	14 bytes	Windows Structure: BITMAPFILEHEADER
Signature	2 bytes	'BM'
FileSize	4 bytes	File size in bytes
reserved	4 bytes	unused (=0)
DataOffset	4 bytes	File offset to Raster Data



InfoHeader	40 bytes	Windows Structure: BITMAPINFOHEADER
Size	4 bytes	Size of InfoHeader =40
Width	4 bytes	Bitmap Width
Height	4 bytes	Bitmap Height
Planes	2 bytes	Number of Planes (=1)
BitCount	2 bytes	Bits per Pixel 1 = monochrome palette. NumColors = 1 4 = 4bit palletized. NumColors = 16 8 = 8bit palletized. NumColors = 256 16 = 16bit RGB. NumColors = 65536 (?) 24 = 24bit RGB. NumColors = 16M
Compression	4 bytes	Type of Compression 0 = BI_RGB <a href="#">no compression</a> 1 = BI_RLE8 <a href="#">8bit RLE encoding</a> 2 = BI_RLE4 <a href="#">4bit RLE encoding</a>
ImageSize	4 bytes	(compressed) Size of Image It is valid to set this =0 if Compression = 0
XpixelsPerM	4 bytes	horizontal resolution: Pixels/meter
YpixelsPerM	4 bytes	vertical resolution: Pixels/meter
ColorsUsed	4 bytes	Number of actually used colors
ColorsImportant	4 bytes	Number of important colors 0 = all



ColorTable	4 * NumColors bytes	present only if Info.BitsPerPixel <= 8 colors should be ordered by importance	
	Red	1 byte	Red intensity
	Green	1 byte	Green intensity
	Blue	1 byte	Blue intensity
	reserved	1 byte	unused (=0)
repeated NumColors times			

# BMP Example

Address:	Contents:	ASCII:
00000000	42 4d 36 12 00 00 00 00 00 00 36 04 00 00 28 00	BM6.....6...(. ..=...8..... ..... ..... ..f...3..... .....f...3. ..... ..f...3.....f ...f...f..ff..3f ...f...3...3...3 ..f3..33...3.... .....f...3. ..... ..f...3..... .....f...3.
00000010	00 00 3d 00 00 00 38 00 00 00 01 00 08 00 00 00	
00000020	00 00 00 00 00 00 13 0b 00 00 13 0b 00 00 00 00	
00000030	00 00 00 00 00 00 ff ff ff 00 cc ff ff 00 99 ff	
00000040	ff 00 66 ff ff 00 33 ff ff 00 00 ff ff 00 ff cc	
00000050	ff 00 cc cc ff 00 99 cc ff 00 66 cc ff 00 33 cc	
00000060	ff 00 00 cc ff 00 ff 99 ff 00 cc 99 ff 00 99 99	
00000070	ff 00 66 99 ff 00 33 99 ff 00 00 99 ff 00 ff 66	
00000080	ff 00 cc 66 ff 00 99 66 ff 00 66 66 ff 00 33 66	
00000090	ff 00 00 66 ff 00 ff 33 ff 00 cc 33 ff 00 99 33	
000000a0	ff 00 66 33 ff 00 33 33 ff 00 00 33 ff 00 ff 00	
000000b0	ff 00 cc 00 ff 00 99 00 ff 00 66 00 ff 00 33 00	
000000c0	ff 00 00 00 ff 00 ff ff cc 00 cc ff cc 00 99 ff	
000000d0	cc 00 66 ff cc 00 33 ff cc 00 00 ff cc 00 ff cc	
000000e0	cc 00 cc cc cc 00 99 cc cc 00 66 cc cc 00 33 cc	



# Little-Endian

- More significant bit is stored in larger memory address.
  - 3d 00 00 00 → 00 00 00 3d →  $3 \times 16 + d \rightarrow 61$
- Little-endianness: Intel, AMD
- Big-endianness: Sun, Motorola
- On Holy Wars and A Plea for Peace
  - Computer, pp. 48-54, Oct 1981

# GIF Format

Name	Size	Description
Signature	6 bytes	'GIF87a' or 'GIF89a'
GlobalDescriptor	7 bytes	global descriptor, always present
Width	2 bytes	width in pixels
Height	2 bytes	height in pixels
Flags	1 byte	global descriptor flags
GlobalColorMap	bit 7	=1 if GlobalColorMap exists (should be true in almost all cases) =0 if <a href="#">default map</a> is used, or if every image has a LocalColorMap
ColorResolutionBits	bits 6-4	+1 = significant bits per color in GlobalColorMap
reserved	bit 3	=0
PixelBits	bits 2-0	+1 = ColorDepth, NumberOfGlobalColors := $2^{\text{ColorDepth}}$
BackgroundColor	1 byte	background color number (from GlobalColorMap or default map)
AspectRatio	1 byte	usually =0
GlobalColorMap	NumberOfGlobalColors * 3	global color table, present only when GlobalDescriptor.Flags.GlobalColorMap = 1
Red	1 byte	red intensity of color (not necessarily 8 significant bits)
Green	1 byte	green intensity of color (not necessarily 8 significant bits)
Blue	1 byte	blue intensity of color (not necessarily 8 significant bits)
repeated NumberOfGlobalColors times		

# GIF Example



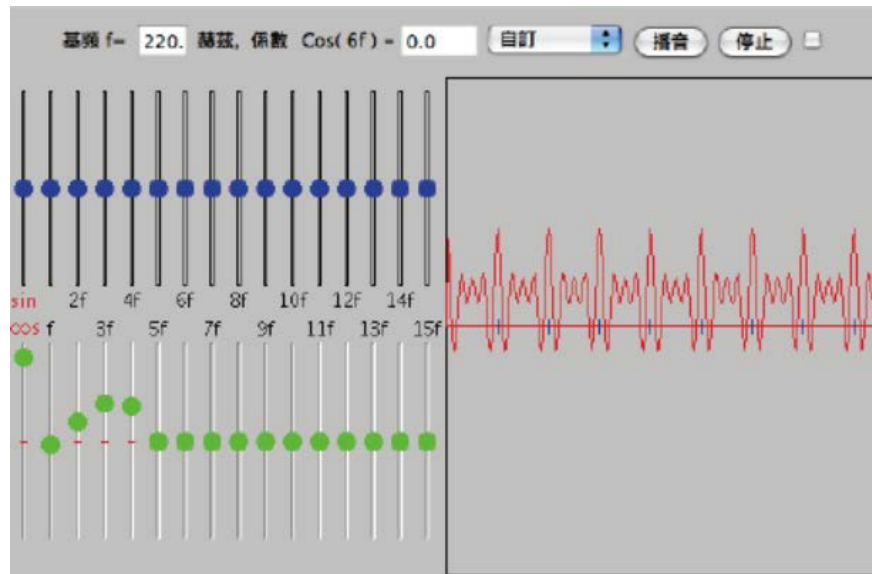
Address:	Contents:	ASCII:
00000000	47 49 46 38 39 61 3d 00 38 00 77 00 31 21 ff 0b	GIF89a=.8.w.1!..
00000010	4d 53 4f 46 46 49 43 45 39 2e 30 0d 00 00 00 01	MSOFFICE9.0.....
00000020	73 52 47 42 00 ae ce 1c e9 00 21 ff 0b 4d 53 4f	sRGB.....!.MSO
00000030	46 46 49 43 45 39 2e 30 17 00 00 00 0b 6d 73 4f	FFICE9.0.....ms0
00000040	50 4d 53 4f 46 46 49 43 45 39 2e 30 42 3c a4 f5	PMSOFFICE9.0B<..
00000050	00 21 ff 0b 4d 53 4f 46 46 49 43 45 39 2e 30 15	.!.MSOFFICE9.0.
00000060	00 00 00 09 70 48 59 73 00 00 0e 68 00 00 0e 69	....pHYs...h...i
00000070	01 e0 5c 1d 2a 00 2c 00 00 00 00 3d 00 38 00 87	..\.*,.,....=.8..
00000080	ff d4 26 ff e3 71 fd b1 48 ff de 59 ec 54 34 59	..&..q..H..Y.T4Y
00000090	85 32 a8 8b 07 ff f4 c7 70 8c 31 2a 2a 2a ff d2	.2.....p.1***..
000000a0	1b ff e9 92 ff 23 1f 9d 7b 0b 4c 7b 31 ff f1 bb	.....#{.L{1... ..X..9../.) ....
000000b0	d8 d2 58 fd cb 39 ff d4 2f f0 29 20 ff e7 85 be	.`...i.G.....C..
000000c0	c4 60 ff fb e6 69 92 47 ff f8 d8 fe da 43 fd ca	.8p...TM.?..J...
000000d0	00 38 70 2e ab b0 54 4d 8a 3f f8 87 4a aa aa aa	..=.P.....2.%..
000000e0	fe d9 3d 95 a6 50 ff ec a1 e9 bd 32 e0 25 14 ff	

# PNG Format

- LZW (Lempel-Ziv-Welch) algorithm (1978)
- Patents of Sperry, Unisys, and IBM (1983)
- LZW published by Welch (1984)
- GIF announced by CompuServe (1987 & 1989)
- Unisys' patent enforcement (1994)
- “Burn All GIFs” campaign (1999)
- Patents expired (2003-2006)

# JPEG Format

- Encoding using DCT (Discrete Cosine Transform)
- A Fourier-related transform
- Fourier transform simulator
  - <http://www.phy.ntnu.edu.tw/moodle/mod/resource/view.php?id=136>

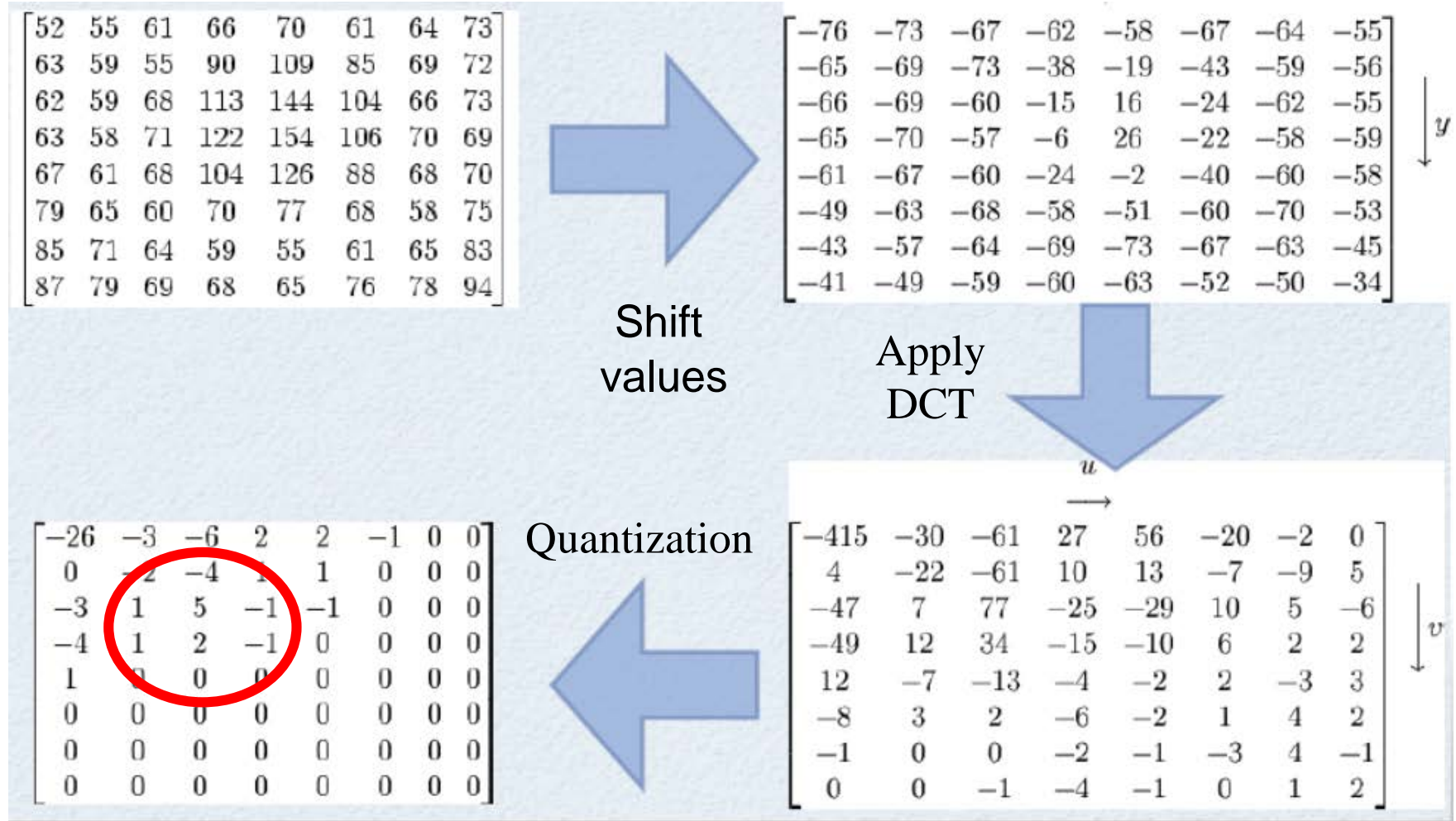




# Steps of JPEG Compression

- Splitting the image into blocks of  $8 \times 8$  pixels
- Applying DCT to each block
- Storing the magnitudes of the low-frequency components

- Example: <http://en.wikipedia.org/wiki/JPEG>



# TIFF Format

- A flexible, adaptable file format for handling images within a single file
- Color space: grayscale, RGB, YCbCr, CMYK, .....
- Compression: raw data, LZW, JPEG, .....
- Copyright holder: Aldus → Adobe

# • TIFF header

Bytes 0-1: The byte order used within the file. Legal values are:

“II” (4949.H)

“MM” (4D4D.H)

In the “II” format, byte order is always from the least significant byte to the most significant byte, for both 16-bit and 32-bit integers. This is called *little-endian* byte order. In the “MM” format, byte order is always from most significant to least significant, for both 16-bit and 32-bit integers. This is called *big-endian* byte order.

Bytes 2-3 An arbitrary but carefully chosen number (42) that further identifies the file as a TIFF file.

The byte order depends on the value of Bytes 0-1.

Bytes 4-7 The offset (in bytes) of the first IFD. The directory may be at any location in the file after the header but *must begin on a word boundary*. In particular, an Image File Directory may follow the image data it describes. Readers must follow the pointers wherever they may lead.

The term *byte offset* is always used in this document to refer to a location with respect to the beginning of the TIFF file. The first byte of the file has an offset of

# Reading Image File

- C/C++ programming
- Visual programming
- Matlab

# C/C++ Programming

- **Example:** <http://www.cs.ucsd.edu/classes/sp03/cse190-b/hw1/>

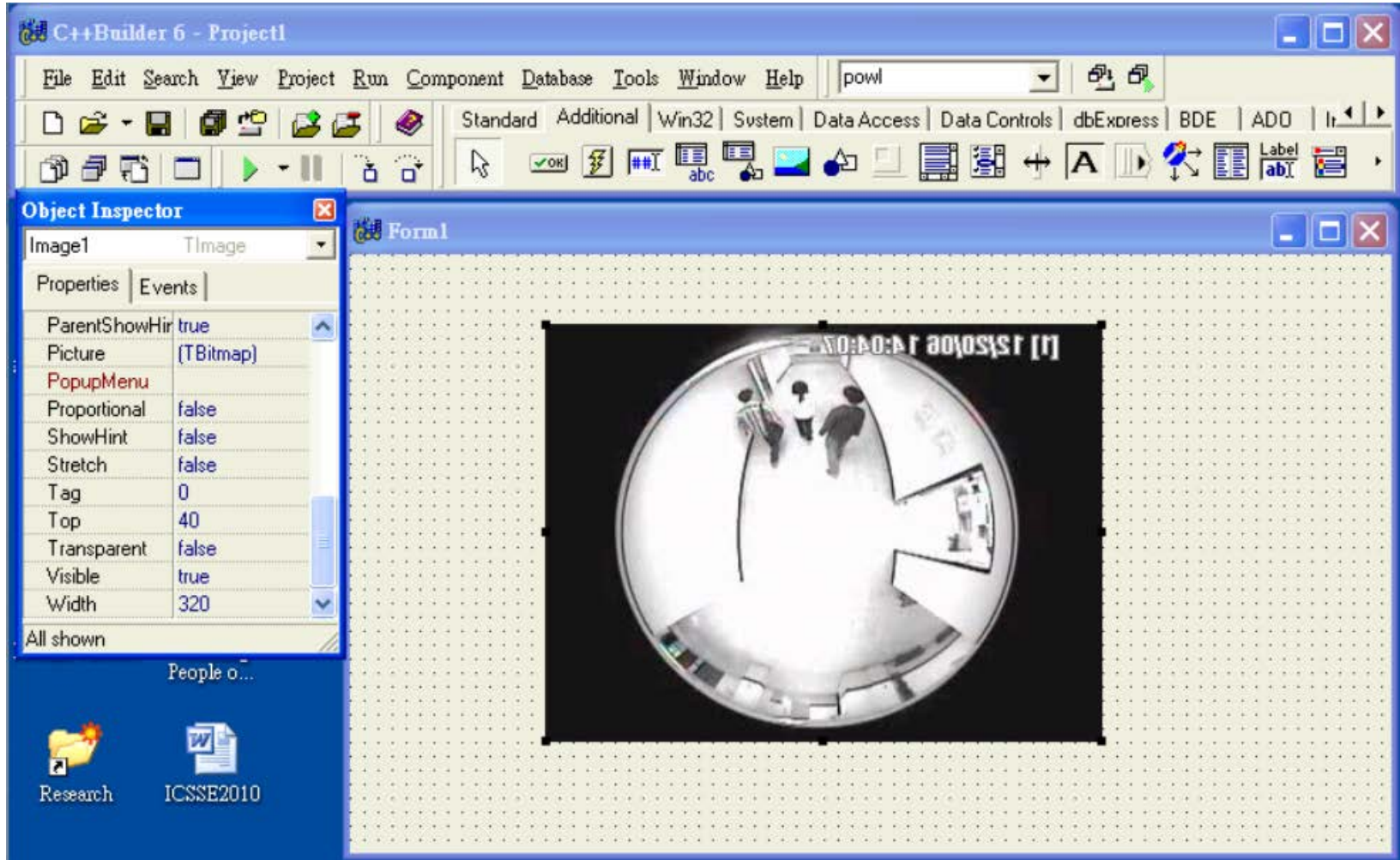
```
/*
Header, 14 bytes.
    16 bytes FileType;           Magic number: "BM",
    32 bytes FileSize;          Size of file in 32 byte integers,
    16 bytes Reserved1;         Always 0,
    16 bytes Reserved2;         Always 0,
    32 bytes BitmapOffset.      Starting position of image data, in bytes.
*/
c1 = fgetc ( filein );
if ( c1 == EOF ) {
    return ERROR;
}
c2 = fgetc ( filein );
if ( c2 == EOF ) {
    return ERROR;
}

if ( c1 != 'B' || c2 != 'M' ) {
    return ERROR;
}
```

- Reading header information

```
/*  
The bitmap header is 40 bytes long.  
32 bytes unsigned Size;           Size of this header, in bytes.  
32 bytes Width;                   Image width, in pixels.  
32 bytes Height;                  Image height, in pixels. (Pos/Neg, origin at botto  
16 bytes Planes;                  Number of color planes (always 1).  
16 bytes BitsPerPixel;           1 to 24. 1, 4, 8 and 24 legal. 16 and 32 on Win95  
32 bytes unsigned Compression;    0, uncompressed; 1, 8 bit RLE; 2, 4 bit RLE; 3, bit  
32 bytes unsigned SizeOfBitmap;   Size of bitmap in bytes. (0 if uncompressed).  
32 bytes HorzResolution;         Pixels per meter. (Can be zero)  
32 bytes VertResolution;         Pixels per meter. (Can be zero)  
32 bytes unsigned ColorsUsed;    Number of colors in palette. (Can be zero).  
32 bytes unsigned ColorsImportant. Minimum number of important colors. (Can be zero).  
*/  
retval = read_u_long_int ( &u_long_int_val, filein );  
if ( retval == ERROR ) {  
    return ERROR;  
}  
  
retval = read_u_long_int ( &u_long_int_val, filein );  
if ( retval == ERROR ) {  
    return ERROR;  
}  
*xsize = ( int ) u_long_int_val;  
  
retval = read_u_long_int ( &u_long_int_val, filein );  
if ( retval == ERROR ) {  
    return ERROR;  
}  
*ysize = ( int ) u_long_int_val;
```

# Visual Programming





# References

- <http://www.cs.umass.edu/~verts/cs32/endianness.html>
- <http://www.phy.ntnu.edu.tw/moodle/mod/resource/view.php?id=136>
- <http://en.wikipedia.org/wiki/JPEG>

# Homework

- Design a program to input and output a color image.
  - You may download the program “bmp\_io.rar” from <http://www.cs.ucsd.edu/classes/sp03/cse190-b/hw1/>
  - Transform a color image  $C(R,G,B)$  into a grayscale image  $I$  by  $I = (R+G+B)/3$ .