

Downsampling

이진영



Image Downsampling

- Decreasing the number of pixels within an original image
- Downscaling from a high resolution image to a low resolution image
- Information loss after downsampling

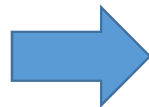


Subsampling

- Selection of a subset of an original image, for example, even rows and columns
- Simplest method for reduction of image size

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 101 | 102 | 103 | 100 | 100 | 100 | 100 | 100 |
| 104 | 50 | 103 | 100 | 100 | 100 | 100 | 100 |
| 100 | 100 | 105 | 110 | 100 | 100 | 100 | 100 |
| 100 | 110 | 100 | 100 | 100 | 100 | 120 | 125 |
| 130 | 140 | 100 | 100 | 100 | 100 | 111 | 100 |
| 150 | 130 | 255 | 255 | 100 | 100 | 109 | 107 |
| 100 | 120 | 255 | 255 | 255 | 100 | 100 | 101 |
| 110 | 100 | 220 | 255 | 255 | 255 | 255 | 100 |

÷2



| | | | |
|-----|-----|-----|-----|
| 101 | 103 | 100 | 100 |
| 100 | 105 | 100 | 100 |
| 130 | 100 | 100 | 111 |
| 100 | 255 | 255 | 100 |

| | | | |
|-----|-----|-----|-----|
| 102 | 100 | 100 | 100 |
| 100 | 110 | 100 | 100 |
| 140 | 100 | 100 | 100 |
| 120 | 255 | 100 | 101 |

| | | | |
|-----|-----|-----|-----|
| 104 | 103 | 100 | 100 |
| 100 | 100 | 100 | 120 |
| 150 | 255 | 100 | 109 |
| 110 | 220 | 255 | 255 |

| | | | |
|-----|-----|-----|-----|
| 50 | 100 | 100 | 100 |
| 110 | 100 | 100 | 125 |
| 130 | 255 | 100 | 107 |
| 100 | 255 | 255 | 100 |

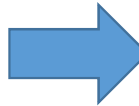


Average Downsampling

- Downsampling method that substitutes pixels within a predefined window with their average value

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 101 | 102 | 103 | 100 | 100 | 100 | 100 | 100 |
| 104 | 50 | 103 | 100 | 100 | 100 | 100 | 100 |
| 100 | 100 | 105 | 110 | 100 | 100 | 100 | 100 |
| 100 | 110 | 100 | 100 | 100 | 100 | 120 | 125 |
| 130 | 140 | 100 | 100 | 100 | 100 | 111 | 100 |
| 150 | 130 | 255 | 255 | 100 | 100 | 109 | 107 |
| 100 | 120 | 255 | 255 | 255 | 100 | 100 | 101 |
| 110 | 100 | 220 | 255 | 255 | 255 | 255 | 100 |

÷2



| | | | |
|-----|-----|--|-----|
| 89 | 102 | | |
| 103 | | | |
| | | | |
| | | | 139 |

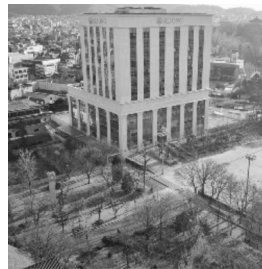


Experiment

- Change from an original resolution to a reduced resolution
- Extraction of even rows and columns in AICenterY.bmp



512×512



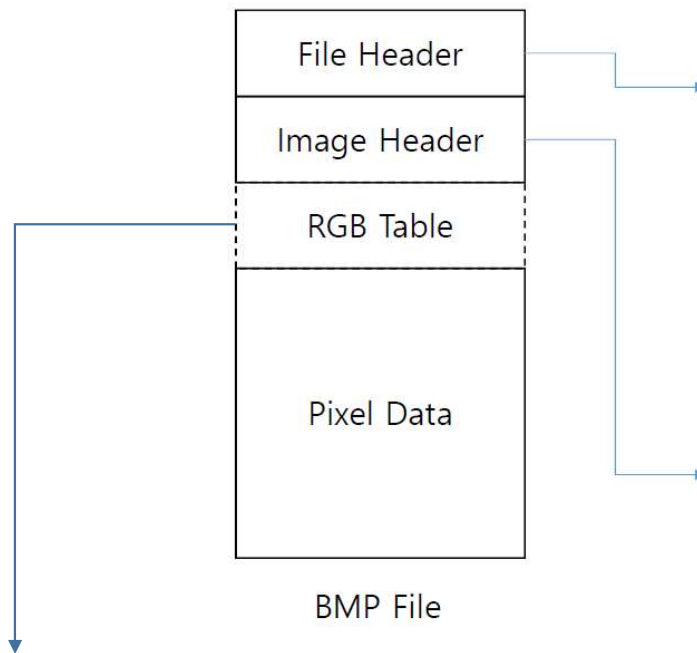
256×256



128×128



BMP File Format



RGB Table: Not used in our experiments

```
typedef unsigned long    DWORD;
typedef int              BOOL;
typedef unsigned char    BYTE;
typedef unsigned short   WORD;
typedef float            FLOAT;
```

```
typedef struct tagBITMAPFILEHEADER {
    WORD    bfType;
    DWORD   bfSize;      → Total File Size
    WORD    bfReserved1;
    WORD    bfReserved2;
    DWORD   bfOffBits;
} BITMAPFILEHEADER, FAR *LPBITMAPFILEHEADER, *PBITMAPFILEHEADER;
```

```
typedef struct tagBITMAPINFOHEADER {
    DWORD   biSize;
    LONG    biWidth;      → Image Width/Height
    LONG    biHeight;
    WORD    biPlanes;
    WORD    biBitCount;   → The number of bits
    DWORD   biCompression;
    DWORD   biSizeImage;  → Image Size
    LONG    biXPelsPerMeter;
    LONG    biYPelsPerMeter;
    DWORD   biClrUsed;
    DWORD   biClrImportant;
} BITMAPINFOHEADER, FAR *LPBITMAPINFOHEADER, *PBITMAPINFOHEADER;
```

Bit-depth = 24 in our experiments
(RGB × 8 bit-depth = 24 bits)



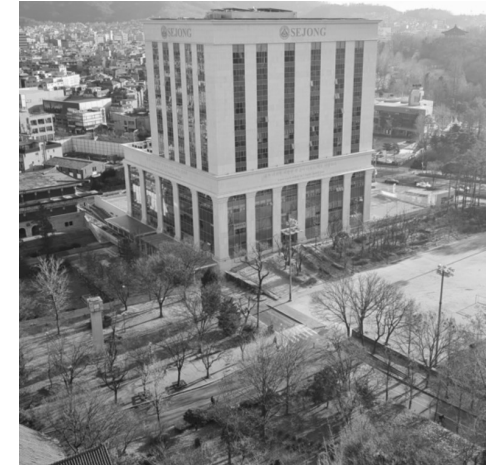
Implementation

```
int ratio = 2;
int width2 = bmpInfo.biWidth >> ratio;
int height2 = bmpInfo.biHeight >> ratio;
int stride2 = (((bitCnt / 8) * width2) + 3) / 4 * 4;
int size2 = stride2 * height2;
```

```
unsigned char *Y2 = (unsigned char *)calloc(width2 * height2, sizeof(unsigned char));
for (int j = 0; j < height2; j++)
{
    for (int i = 0; i < width2; i++)
    {
        Y2[j * width2 + i] = Y1[(j << ratio) * width + (i << ratio)];
    }
}
```

```
FILE *outputFile = fopen("Output.bmp", "wb");
bmpInfo.biWidth = width2;
bmpInfo.biHeight = height2;
bmpInfo.biSizeImage = size2;
bmpFile.bfSize = sizeof(BITMAPFILEHEADER) + sizeof(BITMAPINFOHEADER) + size2;
fwrite(&bmpFile, sizeof(BITMAPFILEHEADER), 1, outputFile);
fwrite(&bmpInfo, sizeof(BITMAPINFOHEADER), 1, outputFile);
fwrite(outputImg, sizeof(unsigned char), size2, outputFile);
```

ratio=2
(/4)



Y1



Y2



Chroma Subsampling

- Less resolution for chroma components (Cb, Cr), compared to a luma component (Y)
- Lower perception for color differences in human visual system
- Y:Cb:Cr = 4:4:4, 4:2:2, 4:2:0...

