
UNIVERSITY OF SPLIT
A SHORT INTRODUCTION TO IMAGE PROCESSING

Department of Electronics & Computing
Faculty of Eletrical Engineering, Mechanical Engineering & Naval Architecture; FESB

UNIVERSITY OF SPLIT

A SHORT INTRODUCTION TO IMAGE PROCESSING

A BRIEF INTRODUCTION IMAGE PROCESSING AND HOW COMPUTERS STORE DIGITAL IMAGES.

Presented by Joško Križanović

April 3, 2023

OUTLINE

UNIVERSITY OF SPLIT

1 Introduction

What is Image Processing?

Subcategories of Image Processing.

2 Reading Digital Images

How Windows Bitmap Files Are Formatted.

What Information Do The Headers Store?

How Are Images Stored In Windows Bitmaps?

Loading a Windows Bitmap to memory.

3 Manipulating Digital Images

Grayscaling and interpolation.

Kernels.

Basic filters using kernels.

IMAGE PROCESSING

UNIVERSITY OF SPLIT

… is the processing of a digital image to heighten/manipulate its visual attributes to our liking and/or to analyze the contents contained within the digital image.

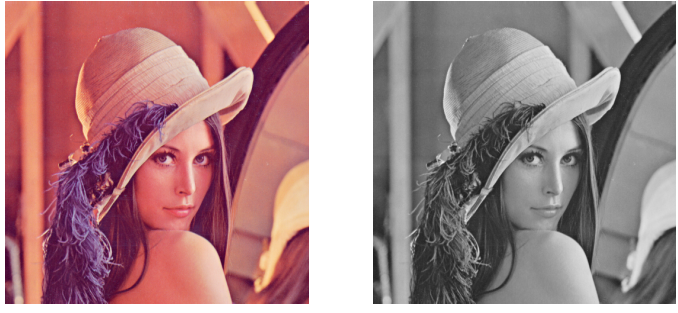


Figure (1): Standard test image “Lena” converted from color to gray scale via. image processing.

MAIN SUBCATEGORIES OF IMAGE PROCESSING

UNIVERSITY OF SPLIT

These aren't so much subcategories as they are image transformations that image processing covers:

- Visualization
- Image sharpening/restoration
- Object detection
- Pattern recognition

Figure (2): On the left a example of image restoration from “salt and pepper” distortion, on the right is a object detection example.

OUTLINE

UNIVERSITY OF SPLIT

1 Introduction

What is Image Processing?

Subcategories of Image Processing.

2 Reading Digital Images

How Windows Bitmap Files Are Formatted.

What Information Do The Headers Store?

How Are Images Stored In Windows Bitmaps?

Loading a Windows Bitmap to memory.

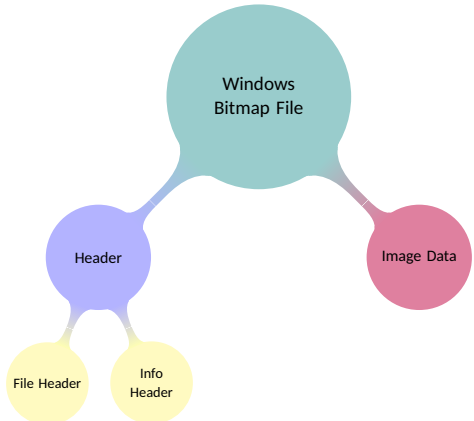
3 Manipulating Digital Images

Grayscaling and interpolation.

Kernels.

Basic filters using kernels.

HOW ARE ".BMP" FILES FORMATTED?



WHAT INFORMATION DO THE HEADERS STORE?

UNIVERSITY OF SPLIT

File Header:

- Signature “*BM*”
- File size
- Image data offset

Info Header:

- Info header type
- Width
- Height
- Bits per pixel
- etc.

HOW ARE IMAGES STORED IN WINDOWS BITMAPS?

Windows Bitmaps store the images as a sequence of bytes that depending on the bits per pixel stores a part of a pixel or multiple pixels in a single byte.
Instead of envisioning it as a sequence of bytes, most if not all image processing books handle images as a matrix of pixels like the following:

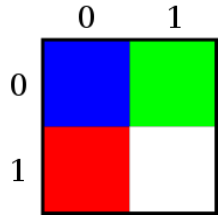
$$p = [r, g, b]$$

$$w = image\ width$$

$$h = image\ height$$

$$Image = \begin{bmatrix} p_{11} & \cdots & p_{1w} \\ \vdots & \ddots & \vdots \\ p_{h1} & \cdots & p_{hw} \end{bmatrix}$$

We can store the following image in a matrix as such:



$$Image = \begin{bmatrix} [0,0,255] & [0,255,0] \\ [255,0,0] & [255,255,255] \end{bmatrix}$$

LOADING A WINDOWS BITMAP TO MEMORY

UNIVERSITY OF SPLIT

I wrote a small program (2000 lines) to load a Windows Bitmap to memory and to display all the header info after its loaded.

```
┌─[02:11:06]─[josko-k@fedora]─[~/nonstd/BMPh_3]
└─ ><>  gcc -lm test.c -o test && ./test

>>---+ Bitmap Contents: +---------->>

        Filename: [lena.bmp]; Header type: [BITMAPINFOHEADER];

>>---+ File Header: +--->>

        Signature       = 424d
        Filesize        = 263222
        Offset          = 1078

>>---+ Info Header: +--->>

        HeaderSize      = 40
        Width           = 512
        Height          = 512
        ColorPlanes     = 1
        BitsPerPixel    = 8
        Compression     = 0
        ImageSize       = 262144

        XRez            = 0
        YRez            = 0

        ColorUse        = 256
        ColorImp        = 256
```

OUTLINE

UNIVERSITY OF SPLIT

1 Introduction

What is Image Processing?

Subcategories of Image Processing.

2 Reading Digital Images

How Windows Bitmap Files Are Formatted.

What Information Do The Headers Store?

How Are Images Stored In Windows Bitmaps?

Loading a Windows Bitmap to memory.

3 Manipulating Digital Images

Grayscaling and interpolation.

Kernels.

Basic filters using kernels.

GRAYSCALING

... is very simple. We calculate a value using a specific formula from all the color values of the pixel. We write that value to r, g and b.

$$gs(p) = 0.299 \cdot r + 0.587 \cdot g + 0.114 \cdot b \rightarrow \mathbb{R}$$

$$Grayscale = \begin{bmatrix} gs(p_{11}) & \dots & gs(p_{1w}) \\ \vdots & \ddots & \vdots \\ gs(p_{h1}) & \dots & gs(p_{hw}) \end{bmatrix}$$

Since all the color values in the individual pixels are equal we can store only one value instead of three which is why $gs(p)$ returns only 1 number.

INTERPOLATION

UNIVERSITY OF SPLIT

... is used for upscaling images, the 2 forms of interpolation are *nearest-neighbor* and *bilinear* with bilinear being more smoother.

Figure (3): On the left is the nearest-neighbor transformed image which is comparatively more blockier than the right image which is bilinear interpolation.

KERNELS

UNIVERSITY OF SPLIT

A *kernel* also known as a digital filter is used to pass images through filters that can sharpen the image, blur the image and do many other transformations to the image. Kernels are matrices and the process of applying a kernel to a another matrix (image) is called a convolution.
Here is a example of the “identity” kernel:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

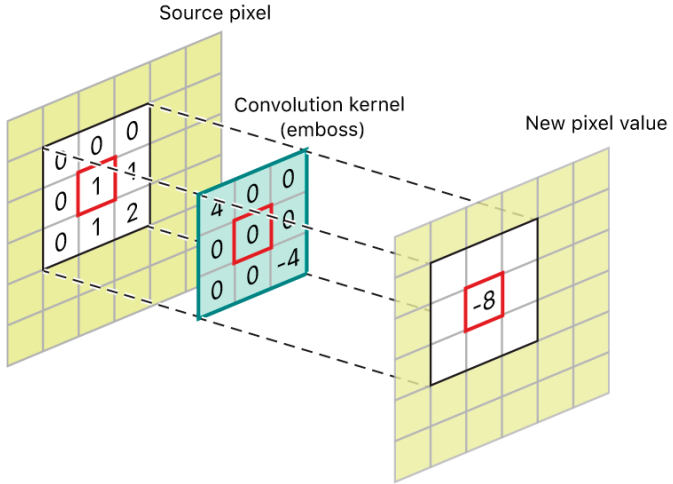


Figure (4): Example of convoluting a matrix with a kernel.

SHARPEN FILTER

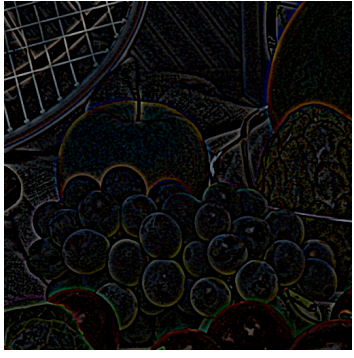
$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

BLUR FILTER

$$\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

GAUSSIAN EDGE-DETECTION FILTER

$$\frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$



BIBLIOGRAPHY

[1] Asem. *Week 4: Image Filtering and Edge Detection*.

[2] Alan C. Bovik. *Handbook of Image and Video Processing*. Academic Press, Inc., 2000.

[3] James D. Muray and William vanRyper. *Encyclopedia of Graphics File Formats, Second Edition*. Ed. by Deborah Russell. O'Relly & Associates, Inc., 1996.


Thank you for your attention!