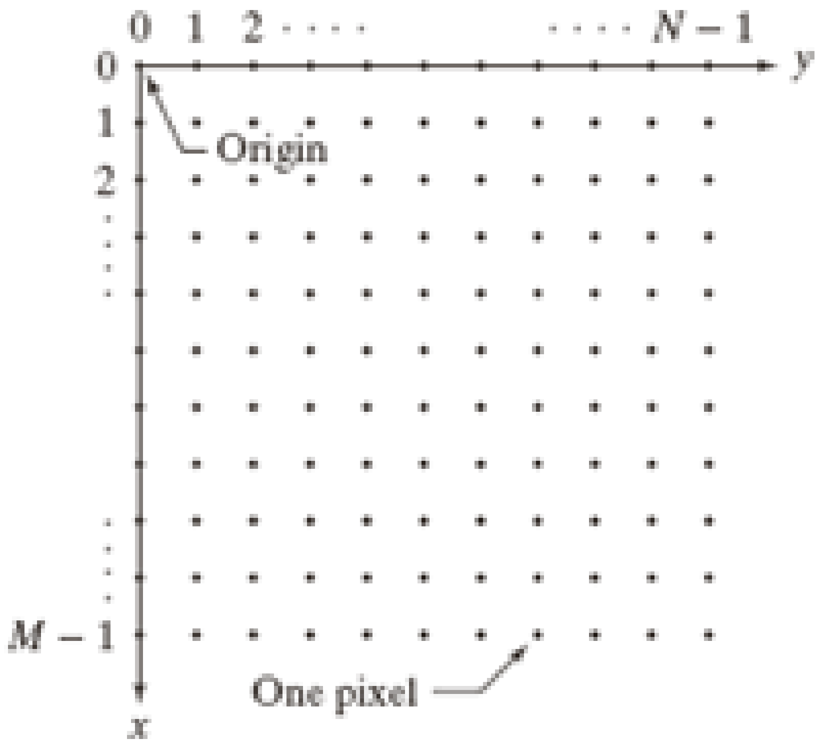
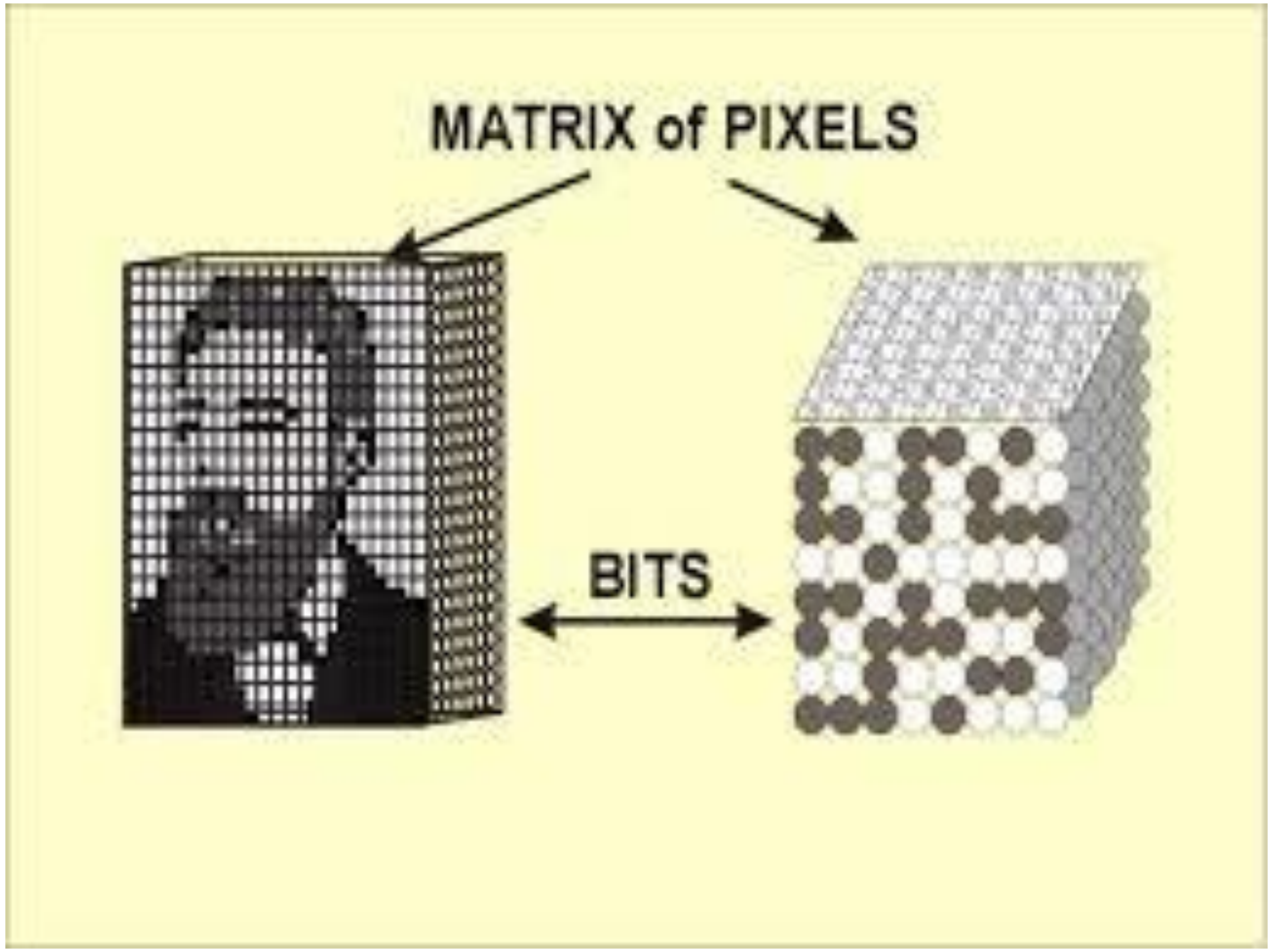
Multimedia-Lecture-One

INTRODUCTION TO IMAGE USING C#

Eng. Noor AL-Hakim

Digital Image Representation





Dealing With Image using C#

- Reading and Writing Image
- Reading and Writing Pixel
    - GetPixel and SetPixel Method
    - LockBits Method
- Displaying the Image

Reading and Writing Image

Bitmap represents an image as a rectangular grid of pixels and provides methods and properties for manipulating and working with images.

Reading

```
// Load the image file
string imagePath = "input.jpg";
Bitmap image = new Bitmap(imagePath);
```

Writing

```
// Save the image to a file
string outputPath = "output.jpg";
image.Save(outputPath);
```

Reading Pixel using GetPixel Method

Color GetPixel(int x, int y): Retrieves the color of the pixel at the specified coordinates(x, y) in the bitmap image

```
Bitmap image = new Bitmap("image.jpg");

Color pixelColor = image.GetPixel(100, 100);

Console.WriteLine($"Pixel Color at (100, 100):
R={pixelColor.R}, G={pixelColor.G}, B={pixelColor.B}");
```

Writing Pixel using SetPixel Method

void SetPixel(int x, int y, Color color): Sets the color of the pixel at the specified coordinates (x, y) in the bitmap image.

```
Bitmap image = new Bitmap("image.jpg");

Color newColor = Color.Red;

image.SetPixel(100, 100, newColor);

image.Save("modified_image.jpg");
```

Try it using a large image for more than one pixel, what do you notice?!

Reading and Writing Pixel using LockBits Method

```
// Load an image
Bitmap bitmap = new Bitmap("image.jpg");

// Lock the bitmap to access pixel data
BitmapData bitmapData = bitmap.LockBits(new Rectangle(0, 0, bitmap.Width, bitmap.Height),
ImageLockMode.ReadOnly, bitmap.PixelFormat);

// pixel manipulation (Do something)

// Unlock the bitmap
bitmap.UnlockBits(bitmapData);
```

Reading and Writing Pixel using LockBits Method

```
// pixel manipulation (Example)

// Get the address of the first line.
IntPtr ptr = bmpData.Scan0;

// Declare an array to hold the bytes of the bitmap.
int bytes = Math.Abs(bmpData.Stride) * bmp.Height;
byte[] rgbValues = new byte[bytes];

// Copy the RGB values into the array.
System.Runtime.InteropServices.Marshal.Copy(ptr, rgbValues, 0, bytes);
```

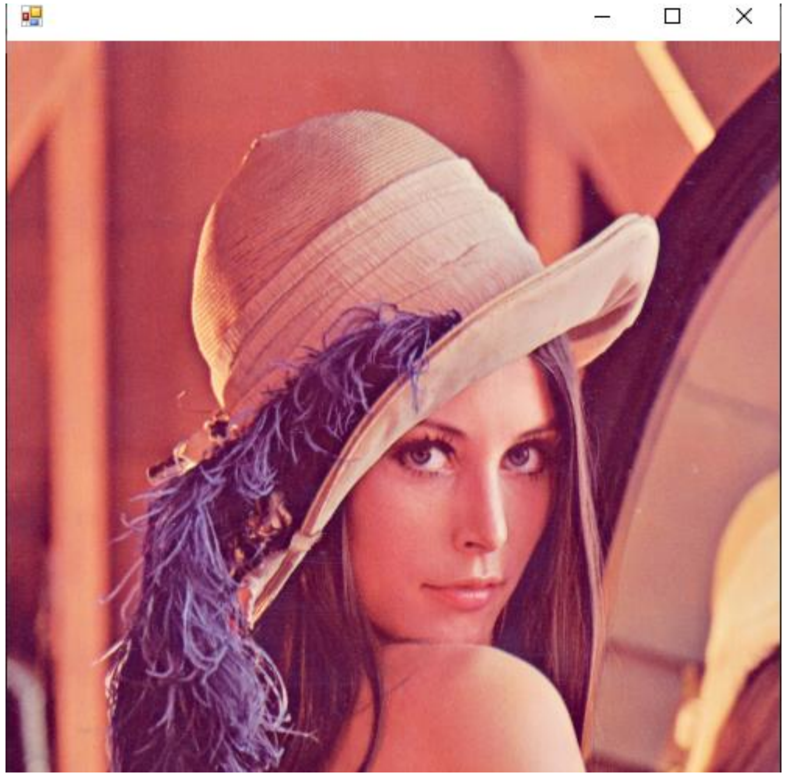
Reading and Writing Pixel using LockBits Method

```
// pixel manipulation

// Set every third value to 255. A 24bpp bitmap will look red.
for (int counter = 2; counter < rgbValues.Length; counter += 3)
         rgbValues[counter] = 255;
// Copy the RGB values back to the bitmap
System.Runtime.InteropServices.Marshal.Copy(rgbValues, 0, ptr, bytes);
```

“ Which of the two methods do you think is better, and why? "

Displaying the Image



```
Form form = new Form();
form.Size = new Size(image.Width,image.Height);

// Handle the Paint event of the form
form.Paint += (sender, e) =>
{
    // Get the graphics object
    Graphics g = e.Graphics;

    // Draw the image at position (0, 0)
    g.DrawImage(image, new Point(0, 0));
};

// Show the form
Application.Run(form);
```

Displaying the Image

```
// Specify the path to the image file
string imagePath = "image.jpg";

// Open the image file using the default image viewer
Process.Start(imagePath);
```

You Should Download !

1. Emgu CV Library to handle with image and video files.
2. NAudio to handle with audio files.
3. MathNet.Numerics to handle with Fourier transformation.

That’s All