

COMPSCI.373

Tutorial 9 solutions: ImageJ

If you find any issues with the solutions below, please email them to tgee862@aucklanduni.ac.nz

ARRAY

```
width = getWidth();
height = getHeight();

cloneImage = newArray(width * height);

counter = 0
for (row = 0; row < height; row++)
{
    for (column = 0; column < width; column++)
    {
        cloneImage[counter++] = getPixel(column, row);
    }
}

// Display pixel (50,50) -> do NOT use print in Moodle!
print("Original Pixel: " + getPixel(50,50));
print("Array Pixel: " + cloneImage[50 + 50 * width]);
```

HISTOGRAM THEORY

- a) To programmatically assess whether an image is over or under-exposed. To decide whether thresholding may be used to segment an object.
- b) 32
- c) 2
- d) 9

HISTOGRAM PRACTICAL

```
width = getWidth(); height = getHeight();

hist = newArray(256);

for (row = 0; row < height; row++)
{
    for (column = 0; column < width; column++)
    {
        pixelValue = getPixel(column, row);
        hist[pixelValue]++;
    }
}

// Display histogram -> do NOT use Plot in Moodle!
Plot.create("Histogram", "X", "Y", hist);
```

CUMULATIVE HISTOGRAM PRACTICAL

```
cumulative = newArray(256);
cumulative[0] = hist[0];
for (i =1; i<256; i++)
{
    cumulative[i] = cumulative[i-1] + hist[i];
}
```

IMAGE NEGATION

```
width = getWidth(); height = getHeight();

for (row = 0; row < height; row++)
{
    for (column = 0; column < width; column++)
    {
        setPixel(column, row, 255 - getPixel(column, row));
    }
}
```

CONTRAST STRETCHING THEORY

$$0 = 50a + b \text{ and } 255 = 101a + b$$

$$\text{Therefore } 51a = 255 \text{ so } a = 5$$

$$\text{This makes } b = -50a = -50 \times 5 = -250$$

CONTRAST STRETCHING PRACTICAL

```
width = getWidth(); height = getHeight();
min = 255; max = 0;
for (row = 0; row < height; row++)
{
    for (column = 0; column < width; column++)
    {
        pixelValue = getPixel(column, row);
        if (pixelValue < min) min = pixelValue;
        if (pixelValue > max) max = pixelValue;
    }
}
gain = 0; bias = 127;
if (min != max)
{
    gain = 255 / (max - min);
    bias = - min * gain;
}

for (row = 0; row < height; row++)
{
    for (column = 0; column < width; column++)
    {
        setPixel(column, row, getPixel(column, row)* gain + bias);
    }
}
```