

CFCS1: Practical 4

Matlab Image Processing

Peter Bell

February 12, 2009

1 Introduction

In this practical you will use Matlab's matrix operations to manipulate this greyscale image of a cat:



To complete the practical you will need to refer to the notes for lectures 14 and 15.

First, save the file `cat.jpg` to your working directory. You can load the image as a Matlab matrix using the commands:

```
c = imread('cat.jpg');  
c = double(c);
```

The image is then represented by the matrix \mathbf{C} : each element of the matrix gives the shade of the corresponding pixel of the image, with a value of 0 indicating a black pixel, and a value of 255 (the maximum) indicating a white pixel. You can display the matrix as an image in Matlab using the command

```
imshow(uint8(c));
```

Try it out!

2 Manipulating the shades

Try out the following, using `imshow` to display the result. Start with the original image each time.

1. Darken the image by dividing each matrix element by 2. Experiment with other divisors too.
2. Brighten the image by multiplying each matrix element by 2 (also try other numbers) – ensure the result is capped at 255.
3. Dull the image by shifting every element closer to 128 (a dull grey), using the formula

$$C \leftarrow (C - 128)/k + 128$$

for a range of values of k .

4. Invert the shades of grey so that 0 becomes 255, 255 becomes 0, etc.
5. Make the image purely black and white by setting every element less than 128 to 0, and every element greater than 128 to 255.

3 Other manipulation

Perform the following, again using the original image each time.

6. Reflect the image along the line of the leading diagonal
7. Crop the image, leaving a central square with side lengths half that of the original,
8. Make the image symmetric about the leading diagonal, resulting in this mutant cat:



4 Manipulation using convolution

4.1 Blurring

As you saw in lectures, an image can be blurred by convolving it with a constant matrix kernel, which has the effect of replacing each element with the mean value of its neighbours. To blur with all the neighbours a distance n away, the constant matrix should be of size $2n + 1$, and the sum of all elements should be 1.

9. Write a function to blur the image, given n . Your function should create the constant matrix kernel, then use Matlab's `conv2` function to convolve it with the image. Experiment with varying n from 1 upwards.

4.2 Creating a pencil drawing (optional)

The kernels given by

$$K1 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad K2 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

can be used to detect vertical and horizontal edges in image, respectively. To create a line-drawing version of the image:

- Create two new matrices by convolving the original with K1 and K2.
- Create an image with both vertical and horizontal edges highlighted by summing the **absolute values** of the resulting two matrices
- Finally, create an cat in the style of a pencil drawing by inverting the shades on this image.