Creative Computing II

Christophe Rhodes
c.rhodes@gold.ac.uk

Autumn 2010, Wednesdays:
10:00–12:00: RHB307 & 14:00–16:00: WB316
Winter 2011, Wednesdays:
10:00–12:00: RHB307 & 14:00–16:00: WB316

Filtering

Application of systems to multimedia.

- audio:
  - mixing and EQ;
  - acoustics;
  - sound effects;
  - subtractive synthesis.
- image:
  - various effects
    - blurring;
    - edge detection;
    - sharpening;
    - ...

Filtering

Images, Matrices and Arrays

Previously:

- arbitrary-dimensional signals;
- audio signals as *Octave* vectors;
- ... so now what?

2D: Matrices.

Filtering

Images, Matrices and Arrays

Matrices:

- `[1 2; 3 4]`;
- `zeros`, `ones`, `rand`;

Filtering

Images, Matrices and Arrays

Matrices:

- `[1 2; 3 4]`;
- `zeros`, `ones`, `rand`;

Operations:

- `scalar + matrix`; `matrix - scalar`;
- `scalar * matrix`; `matrix / scalar`;
- `matrix + matrix`; `matrix - matrix`;
- `matrix .* matrix`;
- `matrix * matrix`;

Filtering

Images, Matrices and Arrays

Images as matrices:

- each channel is 2D signal;
- grayscale is directly representable;
- RGB images as 3D signal: array (size `x`×`y`×3)
- *Octave* support: `imread`, `imwrite`

Note: matrix multiplication does **not** make sense for images-as-matrices.

Filtering

Image Synthesis

Construction of matrices:

- matrix constructors:
  - `zeros`, `ones`;
  - `rand`.

Filtering

Image Synthesis

Construction of matrices:

- matrix constructors:
    - `zeros`, `ones`;
    - `rand`.
- vector multiplication:
    - if `u` is $p \times 1$ and `v` is $1 \times q$ then
    - `u*v` is $p \times q$.

Filtering

Image Synthesis

Construction of matrices:

- matrix constructors:
  - `zeros`, `ones`;
  - `rand`.
- vector multiplication:
  - if `u` is $p \times 1$ and `v` is $1 \times q$ then
  - `u*v` is $p \times q$.

Examples:

- `rand(32);`
- `rand(512);`
- `[1:512]'*ones(1,512)`
- `sin(10*2*pi/512*[1:512])' * sin(20*2*pi/512*[1:512])`

Filtering

Image Synthesis

Other techniques for image synthesis:

- cellular automata;
- visualisation;
- modelling;
- raytracing;

(beyond the scope of this course)

Filtering

Image Synthesis

Other techniques for image synthesis:

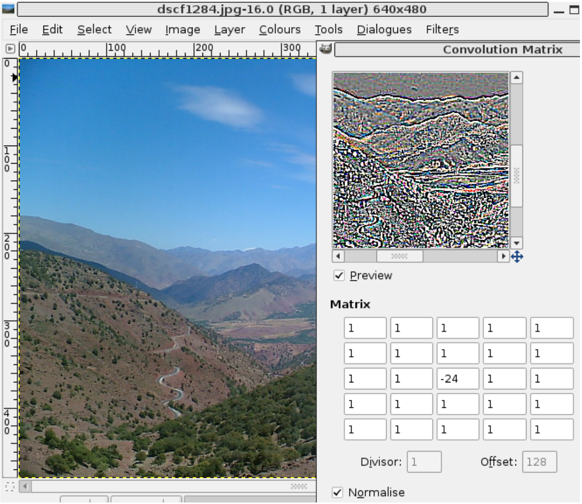
- cellular automata;
- visualisation;
- modelling;
- raytracing;

(beyond the scope of this course)
Image I/O in Octave:

- `imread`, `imwrite`
- `imshow`
- `imagesc`

Filtering

Image Filtering



☑ Preview

Matrix

1	1	1	1	1
1	1	1	1	1
1	1	-24	1	1
1	1	1	1	1
1	1	1	1	1

Divisor: 1   Offset: 128

☑ Normalise

Filtering

Image Filtering

General idea:

- express desired filter as an LTI system;
- compute kernel (IR) for the system;
- apply kernel to image.

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$$

$$y[n,m] = \sum_{k,l=-\infty}^{\infty} h[k,l]x[n-k,m-l]$$

Octave support:

- `conv2`
- `fft2` and `ifft2`
- note: **not** `fir2`

Filtering

Image Filtering

Identity system:

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

Filtering

Image Filtering

Identity system:

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

Use with `conv2(image, kernel, ’same’)`

- treats centre of `kernel` as the origin;
- returns a matrix the **same** size as `image`.

Filtering

Image Filtering

Shift:

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

Filtering

Image Filtering

Bigger shift in a different direction:

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

Filtering

Image Filtering

‘Echo’:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3 & 0 & 0 \\ 0 & 0 & 0.2 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- multiple, superposed shifts;
- scaled to add up to 1;
- not necessarily all the same weight.

Filtering

Image Filtering

Gaussian Blur:

- 1D Gaussian: `g = exp(-([0:n-1]'-(n-1)/2).^2/(2*w))/sqrt(2*pi*w);`
- $\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
- 2D Gaussian: `g * g'`

Filtering

Image Filtering

Motion Blur:

$$\frac{1}{5}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Filtering

Image Filtering

Edge Detection:

$$\begin{pmatrix} 0 & -1 & -1 & -1 & 0 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 20 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & -1 & -1 & -1 & 0 \end{pmatrix}$$

Filtering

Image Filtering

General FIR filters:

- construct an FIR filter using `fir1`;
- extend to 2D using vector multiplication;

Problem:

- resulting 2D filter is not in general circularly symmetrical;
- privileged directions along image axes.