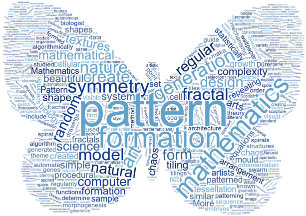
Advanced Mathematical Perspectives 1

Lecture 12: Diffusion and Smoothing



Matthew Roughan

<matthew.roughan@adelaide.edu.au>

www.maths.adelaide.edu.au/matthew.roughan/notes/AMP1/

School of Mathematical Sciences,
University of Adelaide

THE UNIVERSITY of ADELAIDE

SUB CRUCE LUMEN

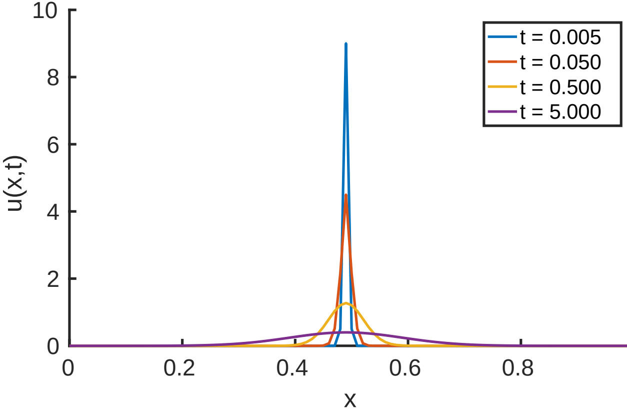
ACEMS

AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

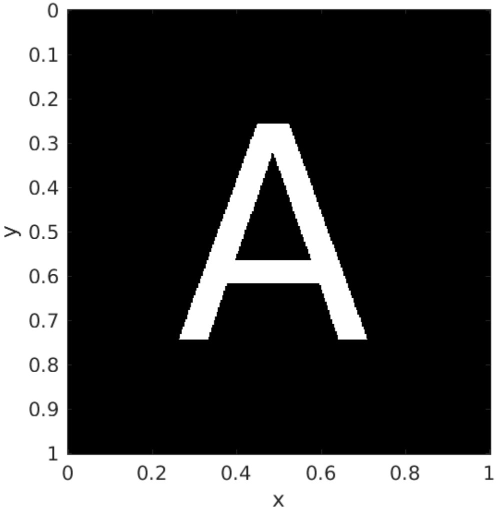
Section 1

Diffusion as Smoothing

Example

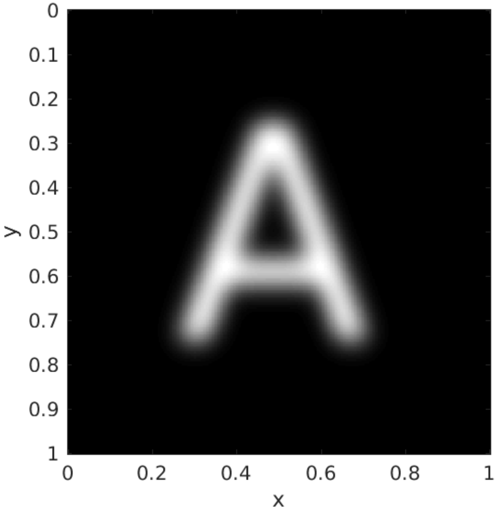


http://www.maths.adelaide.edu.au/matthew.roughan/notes/AMP1/files/diffusion1.gif



Original

www.maths.adelaide.edu.au/matthew.roughan/notes/AMP1/files/diffusion2.gif



At $t = 0.04$

www.maths.adelaide.edu.au/matthew.roughan/notes/AMP1/files/diffusion2.gif

Diffusion as Smoothing

- We can think of diffusion as a "smoothing out" or spreading
  - notice the Gaussian (Bell curve) shape
- Underlying model is often Brownian motion of molecules
  - molecules bounce around at random, slowly diffusing outwards, or spreading kinetic energy (heat)

Section 2

Smoothing out noise – pattern recognition

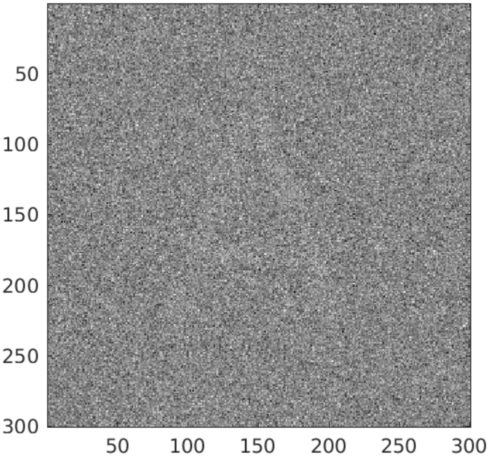
Denoising

- We saw that diffusion "smoothed" out a pattern
  - why would we want to do that?

Denoising

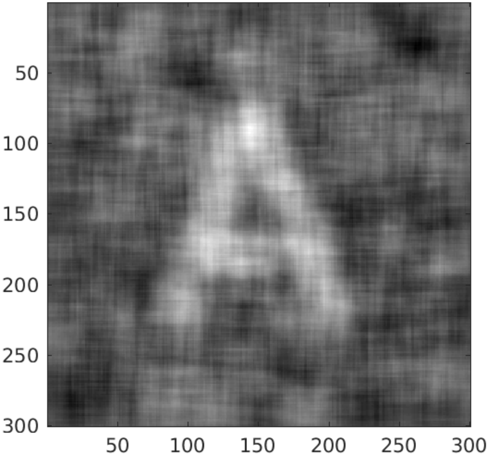
- We saw that diffusion "smoothed" out a pattern
  - why would we want to do that?
- Sometimes there is more than one "pattern" present, and we can *filter* out one to see the other one better
- *e.g.*, a common case is where a pattern is obscured by noise
  - we call this *denoising*

Denoising example



Original

Denoising example



(somewhat) denoised

Diffusion patterns in “dithering”

Another example of diffusion ideas in image processing

- Some devices only have two colours (black and white), or a small set of colours (yellow, cyan, magenta), and can’t mix them.
- So we build up an image from smaller dots
  - sometimes called half-toning
- But if the pattern of the dots is too regular, we start to see artefacts, so often a “diffusion” pattern[1] is used to randomise the dots, to avoid artefacts

[1] We will talk a little more about this randomisation in the next lecture.

Diffusion patterns in “dithering”

- Newsprint is the classic example

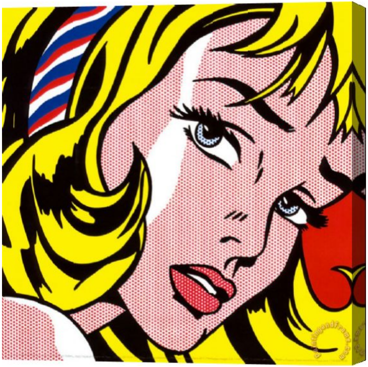
from the Australian, Dec 11th, 2005

Why does dithering work?

- Remember I said that we (humans) are designed to find patterns
- Our eyes (and brain) are really good at denoising
- So we see the pattern (the image) instead of the dithering
  - best dithering patterns have some randomness so that we don't see the wrong pattern
  - one of the most famous uses "error diffusion" to spread out errors: see Floyd-Steinberg error diffusion algorithm

Colour dithering

- The same all works in colour as well
- The idea has been exploited in Art: Roy Lichtenstein (1923-97) played with it in his art, exaggerating the dithering patterns



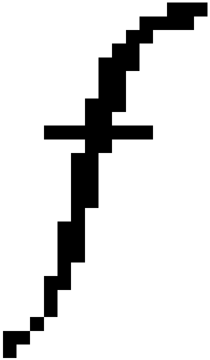
Aliasing in images

Aliasing has technical roots in frequency analysis, but in images we can see it visually, particularly in computer generated images or video

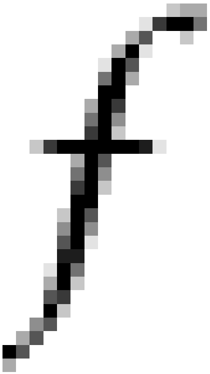
- "jaggies" in images and fonts
- Moire patterns
- marching ants

Anti-aliasing usually involves some form of smoothing.

Anti-aliased fonts



aliased



anti-aliased

Takeaways

- Diffusion is one of the underlying models for many physical processes (often ones that build patterns)
- It results in "smoothing" of an initial signal, and this can be used in filtering and denoising patterns
- We have implicit filtering going on in our heads!
- We will come back to use diffusion again as part of a larger pattern formation process, but next we will look at another model for diffusion

Section 3

Extras

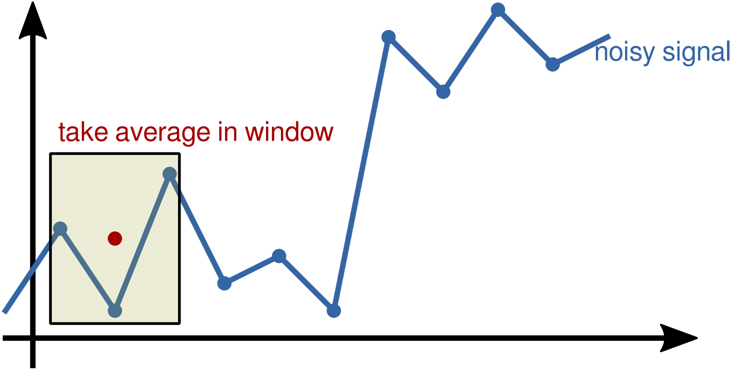
Convolutions and Moving Average Filters

- Solving a set of difference equations to approximate diffusion isn't the best (computational) way to solve our problem
- What we want to do is use the idea of smoothing to build a *filter* that is a bit more direct
- We can do this with *convolutional* filters
- I'm not going to go into much more detail, except that
  - we often call these a Moving Average (MA) filter because for each data point, we take an average of a window around the point, and we move this window onto the next point
  - we can implement these easily in MATLAB using

```
conv    % for 1D signals
conv2   % for 2D signals, such as images
```

Convolutions and Moving Averages

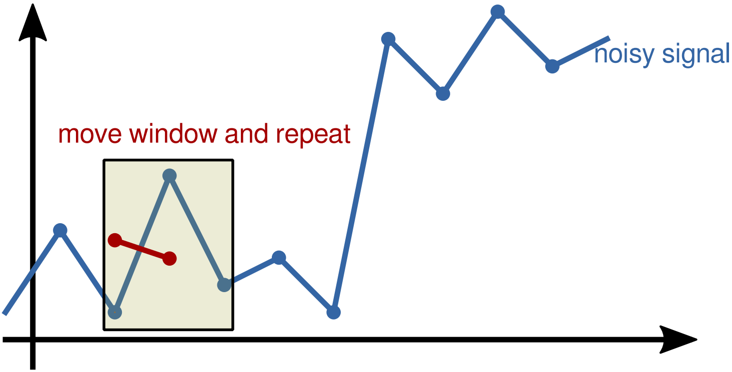
- Take an average in a moving window



- Equivalent to doing a local least-squares regression at each point (see statistics for notes on regression).

Convolutions and Moving Averages

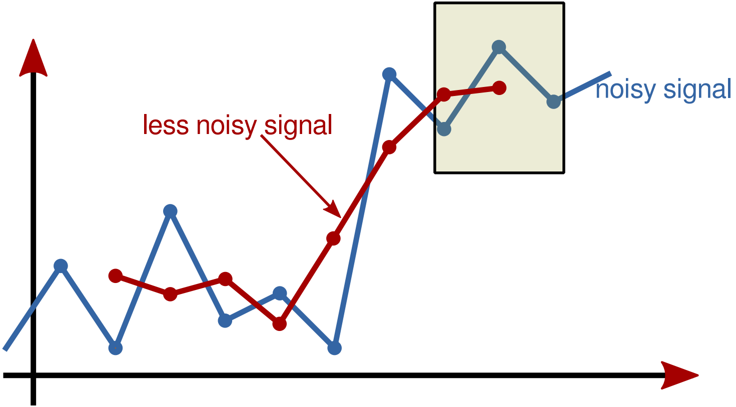
- Take an average in a moving window



- Equivalent to doing a local least-squares regression at each point (see statistics for notes on regression).

Convolutions and Moving Averages

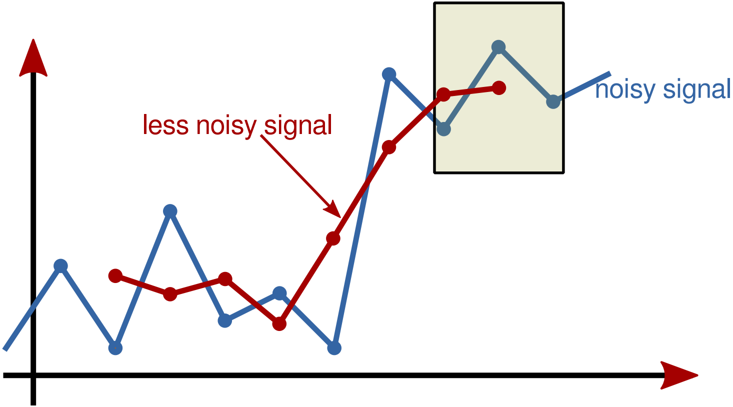
- Take an average in a moving window



- Equivalent to doing a local least-squares regression at each point (see statistics for notes on regression).

Convolutions and Moving Averages

- Take an average in a moving window



- Equivalent to doing a local least-squares regression at each point (see statistics for notes on regression).
- In the image example I used conv2 with a 31×31 pixel window

Further reading I