Administrivia

- Extended office hours tomorrow (Wednesday afternoon). If you want to reserve a slot, drop me an E-mail.

- Assignment 3 updated with a hint for 4(b).

- I am hoping for lots of mail tonight regarding projects (does not need to be long, but should demonstrate that you have been thinking about it). It is best to get feedback early on this kind of endeavor!

- Looking ahead: There will **not** be a fourth assignment this week so that you can push your projects forward enough to give nice presentations.

- Midterm: Just before spring break or a bit after?

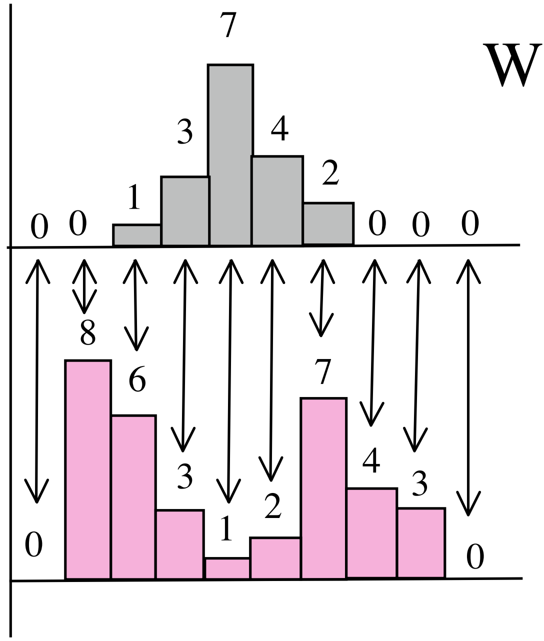
Syllabus Notes

- Filters is next. We will do §7.1, in less detail §7.1-7.4, in more detail §7.5 and §7.6, and mention some of §7.7.

- Then onto edge detection. We will touch on much of §8.

Linear Filters (§7)

- General process:
  - Form new image whose pixels are a **weighted sum** of original pixel values, using the same set of weights at each point.
- Much like a 2D version of the sensor response computation (sensitivities are like weights), but now compute a similar weighted sum for each point



Weights (kernel)

Multiply lined up
pairs of numbers and then sum up

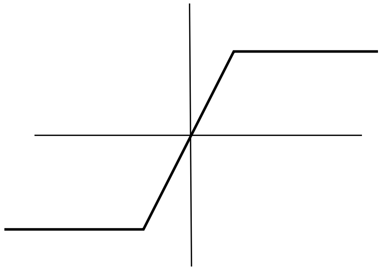
Signal

Linear Filters (§7)

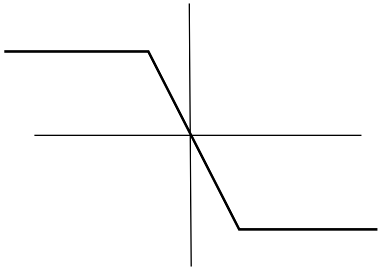
- Example: smoothing by averaging
  - form the average of pixels in a neighbourhood (weights are equal)
- Example: smoothing with a Gaussian
  - form a weighted average of pixels in a neighbourhood (weights follow a Gaussian function)
- Example: finding a derivative
  - negative weights on one side, positive ones on the other

Linear Filters (§7)

- Properties
  - Output is a **linear** function of the input
- Terminology
  - Array of weights is referred to as the kernel (H)
- Be aware of two forms
  - Correlation (more natural, often what we visualize)
  - Convolution (more commonly referred to, has some mathematical properties which are useful)
  - Convolution is correlation by a flipped kernel (if kernel is symmetric, then no difference)



versus



Convolution

- Denote by (will see others symbols!).
- Represent weights as an image, H (the kernel)
- Result is
- $$R_{ij} = \sum_{u,v} H_{i-u,j-v} F_{uv}$$
- Notice weird order of indices (includes the flips)

Properties

- Linear
- Output is a s**hift-invariant** function of the input (i.e. shift the input image two pixels to the left, the output is shifted two pixels to the left)
- Commutative
- **Associative** (Can save CPU time!)

$$(A \quad B) \quad C = A \quad (B \quad C)$$

- Converse of above is true: If a system is linear and shift invariant, then it is a convolution.

Response as sum of basis function (§7.2)

- Another interpretation (explains the flips): The result is a linear combination of shifted versions of the kernel.
- The shifted versions of the kernels form a basis over which the result image is constructed
- The weights are the values of the function being convolved

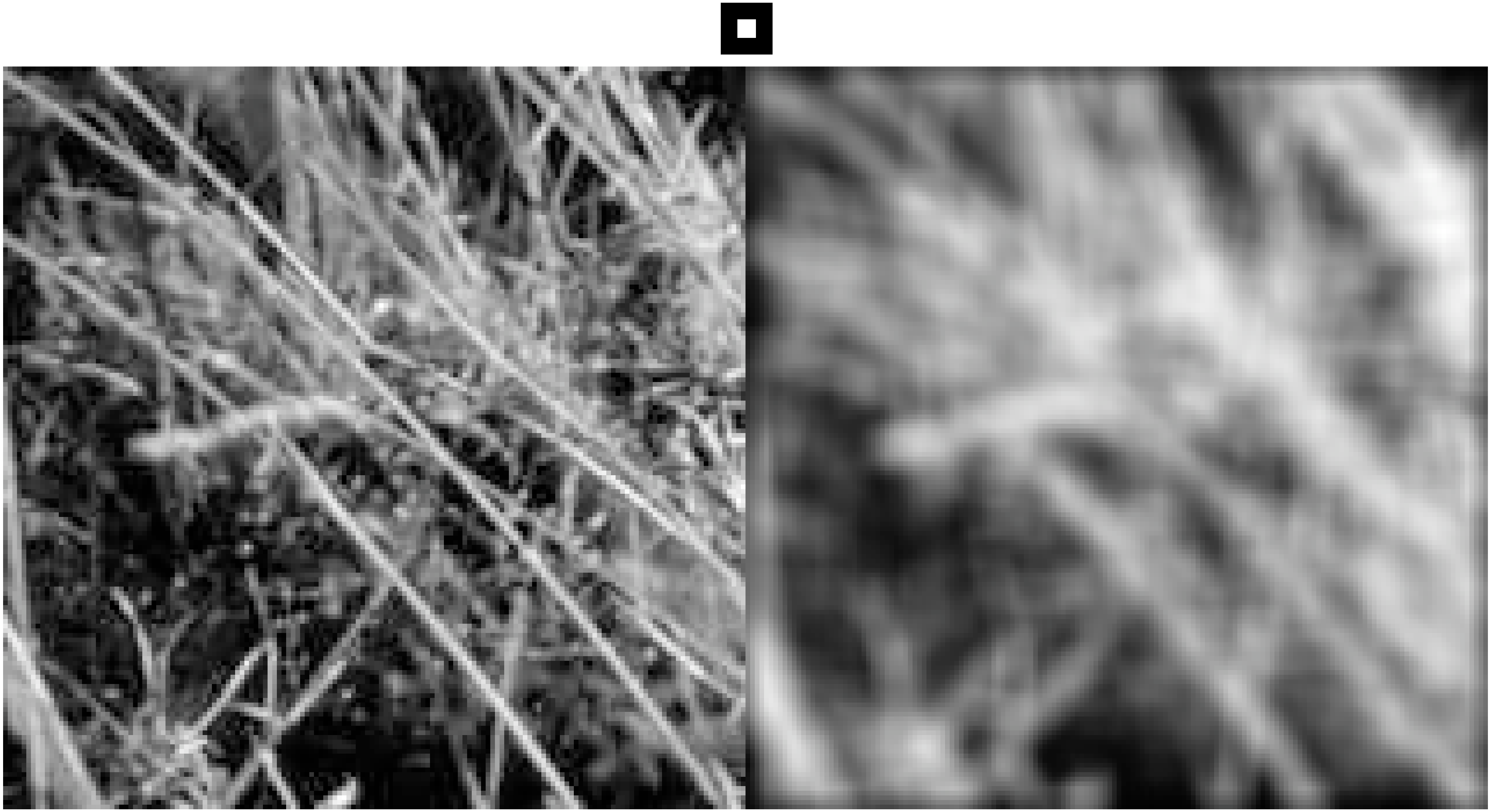
$$R_{ij} = \sum_{u,v} H_{i-u,j-v} F_{uv}$$

- This interpretation **requires** flips if the kernel is not symmetric (try the kernel -1 in (-1,0) and 1 in (0,1)).
- Thinking of an image as a weighted sum over a basis is a generally useful idea--we will come back to it when we do Fourier transforms.

Correlation

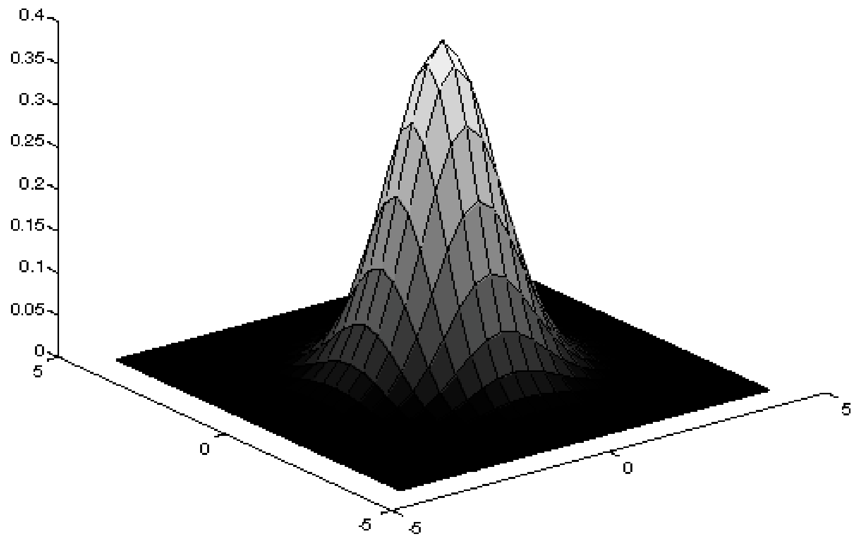
- Similar to convolution (no flips)
- Implements convolution (if a flip is used) or vice versa
- Finds things in images that “look like” the kernel
- The kernel is also referred to as a “mask”, especially in application oriented discussion (both in convolution and correlation).

Example: Smoothing by Averaging



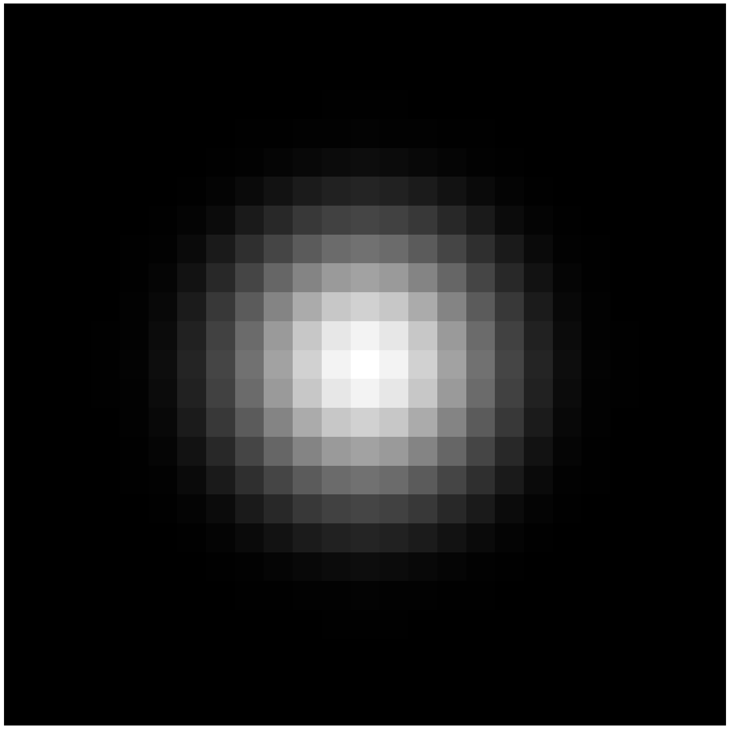
Smoothing with a Gaussian

- Smoothing with an average actually doesn’t really make sense because points close to the center should count more.
- Also, it does not compare at all well with a defocused lens
    - Most obvious difference is that a single point of light viewed in a defocused lens looks like a fuzzy blob; but the averaging process would give a little square.



- A Gaussian gives a good model of a fuzzy blob

An Isotropic Gaussian

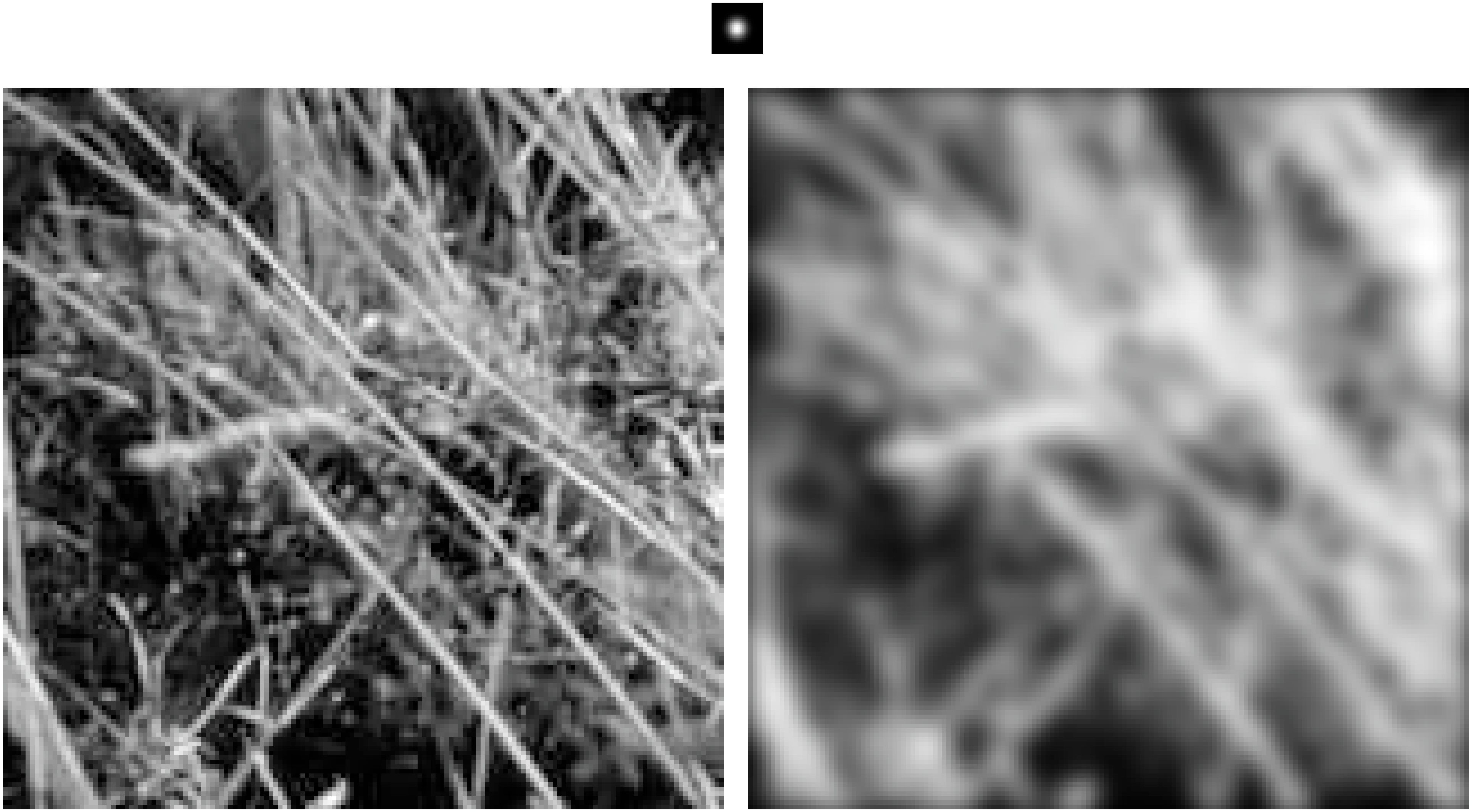


- The picture shows a smoothing kernel proportional to

$$\exp\left(-\left(\frac{x^2+y^2}{2\sigma^2}\right)\right)$$

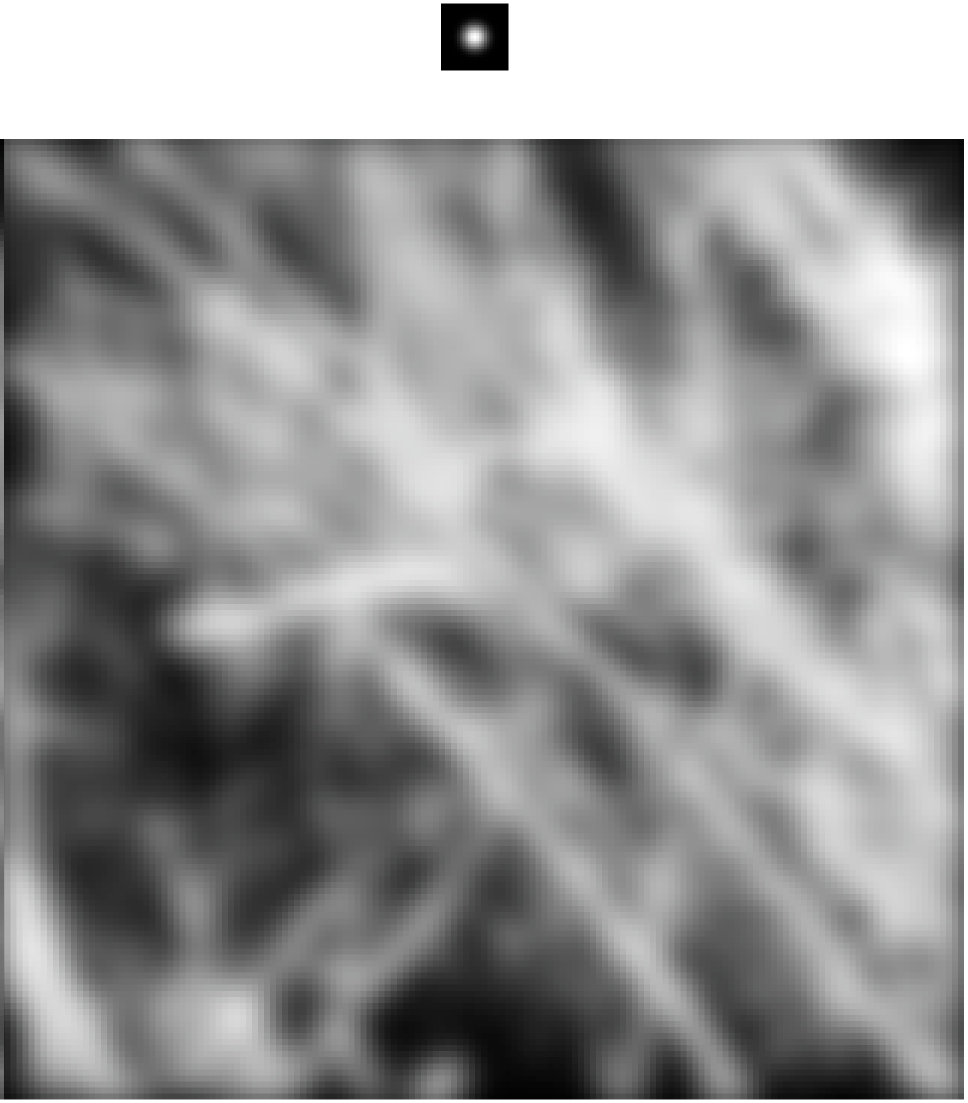
(a reasonable model of a circularly symmetric fuzzy blob)

Smoothing with a Gaussian



Block Averaging

Gaussian



Filters are templates

- Applying a filter at some point can be seen as taking a dot-product between the image and some vector
- Filtering the image is a set of dot products

- Useful intuition
  - filters look like the effects they are intended to find
  - filters find effects that look like them