Features

Examples of Features

- Primitive features, e.g.:
  - weight, length, width, height, volume …
  - variance, moments, eigenvalues, …
  - amplitude, frequency, phase, duration, roll-off, flux …
  - beats per minute, temperature, pressure,...
  - edges, corners, lines, curvature, …
  - mean RGB colour, colour histogram, …
- Semantic features, e.g.:
  - colour layout (red, cyan, magenta,…)
  - texture descriptors (coarse, fine, rough, smooth,…)
  - shape descriptors (rectangular, circular, elliptical,…)
  - kind of day (warm, cold, sunny, rainy, …)
- Statistical features...
- Complex features...

Recall example: Image Categorization (Indoors or Outdoors)

Majid's part of SPS: *How to represent the data and then extract features from it*

Training Images

Image Features → Classifier Training → Trained Classifier

Training Labels → Classifier Training

Training

Testing

Test Image → Image Features → Trained Classifier → Prediction *Outdoor*

Quick Review: Features

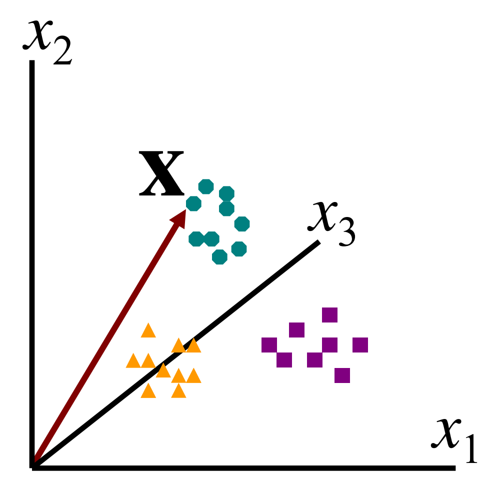
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

- Features describe characteristics of our data.
- The combination of d features is represented as a d-dimensional column vector called a *feature vector*.
- The d-dimensional space defined by the feature vector is called the *feature space*.

$\mathbf{x}$ is a point in feature space X

$$\mathbf{x} \in X$$

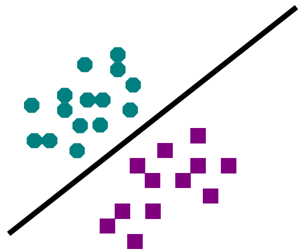
Example: 3D feature space X



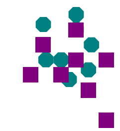
Feature Properties

what makes a good feature vector?

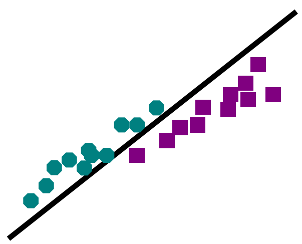
good features,
also
linearly-separable
features



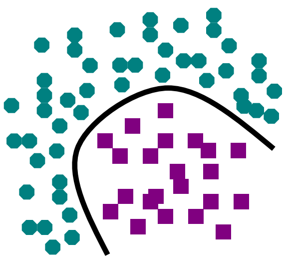
bad features



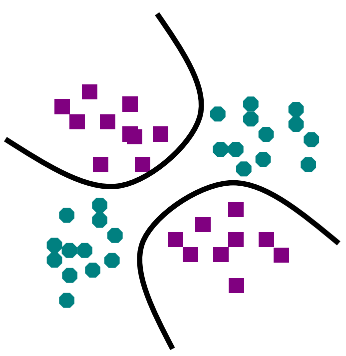
highly correlated
features



nonlinearly-separable
features



multimodal
features



Dimensionality Reduction

- Strive for compact representation of the *properties* of data.
- This compact representation removes redundancy/irrelevancy.
- The choice of features is very important as it influences:
  - accuracy of classification
  - time needed for classification
  - no. of learning examples
  - difficulty in performing classification

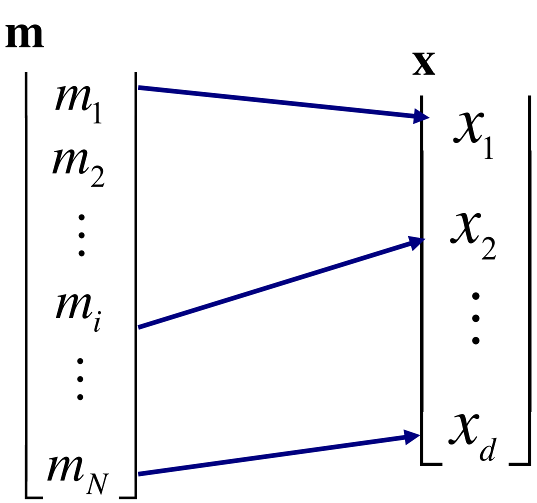
Feature Selection and Feature Extraction:

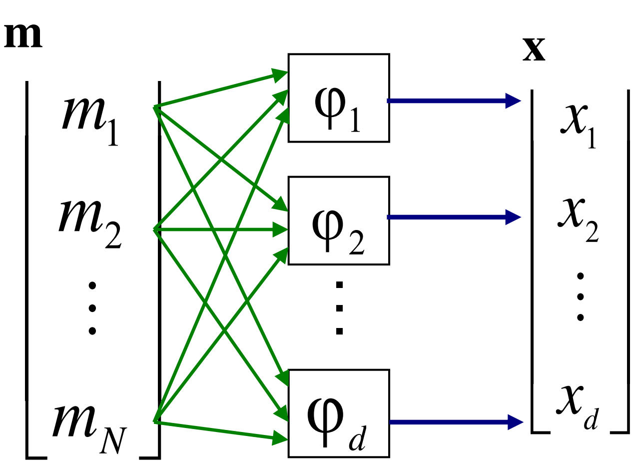
- to generate a set of characteristic attributes from data
- to allow representation of data in a *reduced dimension*

Selection or Extraction?

Two general approaches to dimensionality reduction:

- Feature Selection: Selecting a subset of the existing features without a transformation
- Feature Extraction: Transforming the existing features into a lower dimensional space





$d<N$

Implementing Feature Selection

- Feature Selection is necessary in a number of situations,

  e.g. there may be too many features or may be too expensive to obtain.

Given a feature set $\{\mathbf{x}_i\}$, $i=1,\ldots,N$, find a subset $\mathbf{X}$ of size d with $d<N$, that optimizes an objective function $J(\mathbf{X})$, e.g. P(correct classification). This function would have to be evaluated many times:

$$\frac{N!}{(N-d)!d!}$$

e.g. for 10 features out of 25 one would have to consider 3,268,760 feature sets.

- Feature Selection involves a search strategy that may explore the space of all possible combinations of features.

Heuristic Feature Selection Methods

- Assume features are independent.
- Best single features can be chosen by significance tests.
- bottom-up: build up d features incrementally, **starting with an empty set** → *step-wise feature selection*:
  - The best single-feature is picked first
  - Then next best feature conditioned to the first, ...
- top-down: **start with full set** of features and remove redundant ones successively → *step-wise feature elimination*

Feature Extraction

- Linear or non-linear transformation of the original variables to a lower dimensional feature space → also known as *feature selection in the transformed space*.
- Given a feature space R^N with feature vectors $\mathbf{m}$ find a mapping $\mathbf{x} = \varphi(\mathbf{m}) : R^N \rightarrow R^d$, $d < N$ such that the transformed feature vector $\mathbf{x} = \{x_i\} \in R^d$ preserves (most of) the information or structure in R^N.
- Ideally, we want distinguishing features that are invariant to *operations on the input data:*

Example: how would a robot recognise an object given there might be translation, rotation, scaling, projective distortion, deformation, etc.

Power Spectrum Features

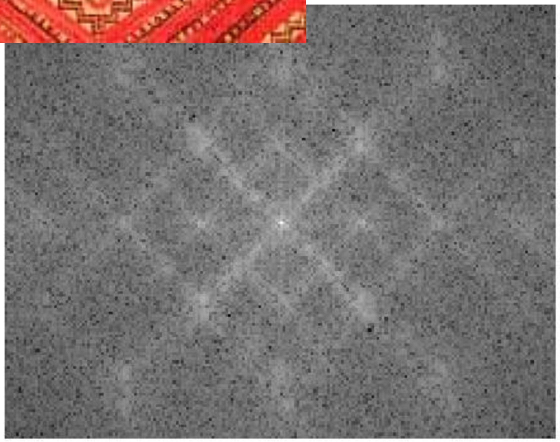
- Primary metric in the frequency domain is *power,* i.e. the square of the magnitude.

Example:

- *Texture* exhibits peaks in the power spectrum (especially if it is periodic or directional) .
- Common to extract features by measuring the power in specific regions of the spectrum.

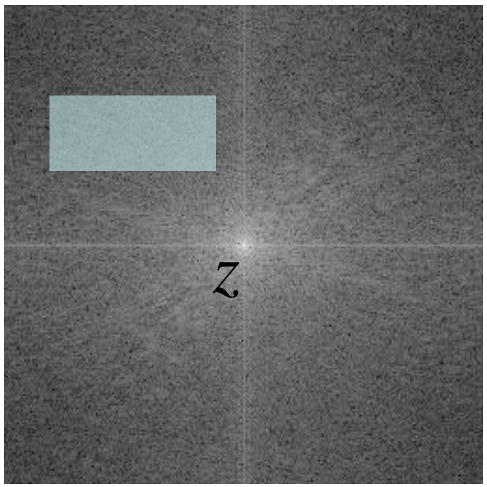
power

$$\sum_{u} \sum_{v} |F(u,v)|^2$$



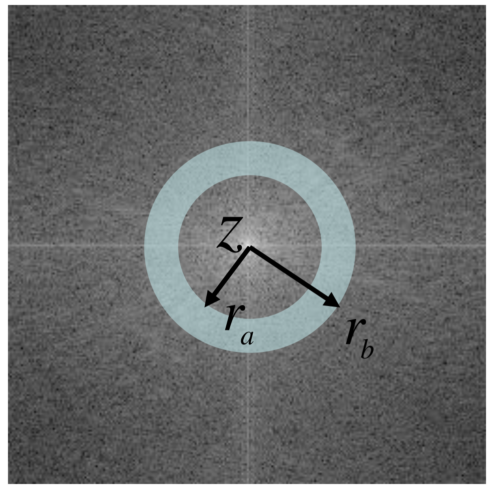
Spectral Features from Spectral Regions

- Fourier space, with origin at $z=(u=0,v=0)$.



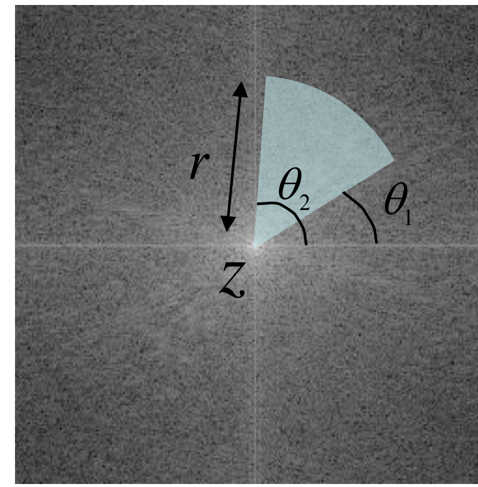
$$a \le u \le b$$
$$c \le v \le d$$

box



$$-r_b \le u \le r_b$$
$$\pm\sqrt{r_a^2 - u^2} \le v \le \pm\sqrt{r_b^2 - u^2}$$

ring



$$u^2 + v^2 = r^2$$
$$\theta_1 \le \tan^{-1}\frac{v}{u} \le \theta_2$$

sector

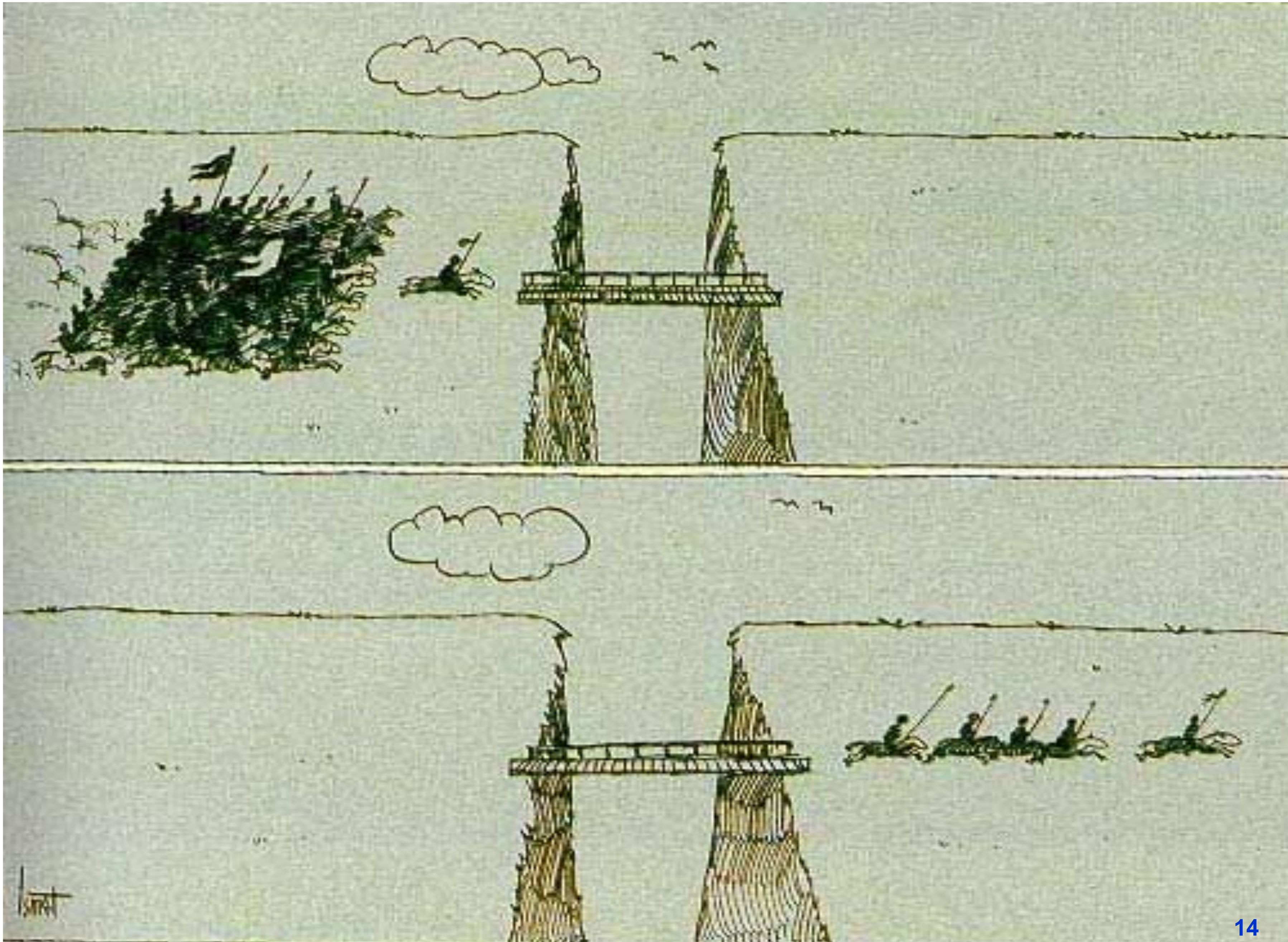
Sum the power for $u, v \in \Re$

Spatial Filtering

We can filter signals and symbols in the spatial/time domain:

- introduce some form of enhancement
  - remove noise/outliers
  - smoothing/averaging out detail
  - sharpening/highlighting detail
- prepare for next stage of processing
  - feature extraction

Filters are also referred to as *kernels* or *masks*.



Spatial Filtering

Many spatial filters are implemented with ***convolution*** masks.

To do convolution we need to know about ***neighbourhoods***.

Symbolic Data

ATAGACATGGC .....

neighbours of G?

1D signal data

3	2	4	4	2	6	.....

2D signal data

3	2	4	4	2	6
3	4	5	4	3	6
4	2	4	4	3	3
3	2	4	4	2	6
3	2	4	5	2	6

Convolution mask is applied to each signal sample and its neighbourhood.

Convolution/Correlation

Convolution Filter/Kernel *h(x)*

1	2	1

Signal *f(x)*

1	1	2	2	1	1	0	1	2	2

Filter Response *g(x)*

	$\frac{5}{4}$								

Convolution/Correlation

Convolution Filter/Kernel $h(x)$

1	2	1

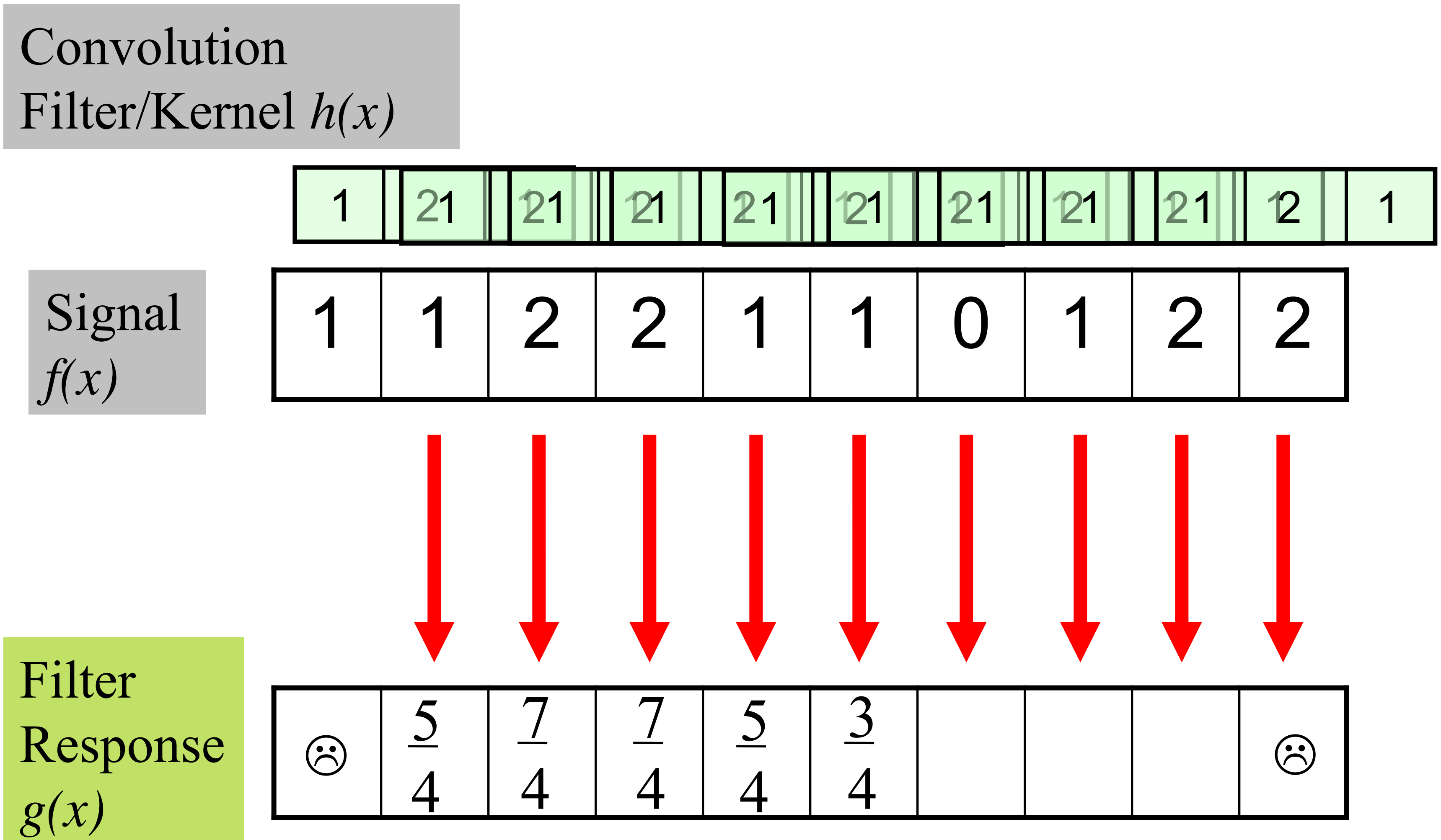
Signal $f(x)$

1	1	2	2	1	1	0	1	2	2

Filter Response $g(x)$

	$\frac{5}{4}$	$\frac{7}{4}$							

Convolution/Correlation



Convolution

- f is the signal, h is the convolution filter
  - h has an origin: e.g. $\frac{1}{4}$ | 1 | **2** | 1 |
  - Normalization factor, e.g. $\frac{1}{4}$, is also part of the filter!

- The discrete version of convolution is defined as:

$$g(x) = \sum_{m=-s}^{s} f(x-m)h(m) \qquad \text{for} \quad s \geq 1$$

2D Convolution

- The discrete version of 2D convolution is defined as

$$g(x,y) = \sum_{m=-1}^{1} \sum_{n=-1}^{1} f(x-m, y-n)h(m,n)$$

Shorthand form:

$$g = f * h$$

Convolution symbol

h

	-1	0	1
-1	-1	0	1
0	-2	0	2
1	-1	0	1

f

		$y-1$	y	$y+1$	
$x-1$		43	12	61	
x		44	45	60	
$x+1$		43	50	61	

$$\begin{aligned}
& f(x+1, y+1)h(-1,-1) \\
&+ f(x+1, y)h(-1,0) \\
&+ f(x+1, y-1)h(-1,1) \\
&+ f(x, y+1)h(0,-1) \\
&+ f(x, y)h(0,0) \\
&+ f(x, y-1)h(0,1) \\
&+ f(x-1, y+1)h(1,-1) \\
&+ f(x-1, y)h(1,0) \\
&+ f(x-1, y-1)h(1,1)
\end{aligned}$$

-68

2D Correlation

- The discrete version of 2D correlation is defined as

$$g(x, y) = \sum_{m=-1}^{1} \sum_{n=-1}^{1} f(x+m, y+n)h(m,n)$$

h

	-1	0	1
-1	-1	0	1
0	-2	0	2
1	-1	0	1

f

		$y-1$	y	$y+1$	
$x-1$		43	12	61	
x		44	45	60	
$x+1$		43	50	61	

68

Correlation=Convolution when kernel is symmetric under 180° rotation, e.g.

1	2	1

Spatial Low/High Pass Filtering

- 1D: turning the treble/bass knob down on audio equipment!
- 2D: smooth/sharpen image

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\frac{1}{16}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Spatial/Frequency Domain Filtering

- Convolution Theorem:

Convolution in the spatial domain
is equivalent to
multiplication in the frequency domain
and vice versa

$$g = f * h \quad \text{implies} \quad G = FH$$

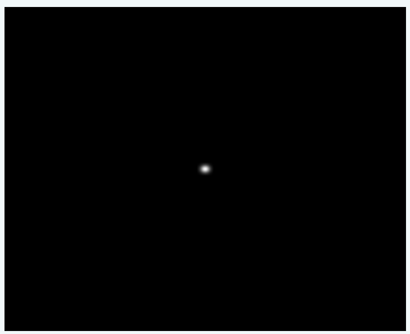
$$g = fh \quad \text{implies} \quad G = F * H$$

Example: Convolution in SD is Multiplication in FD

$f(x,y)$

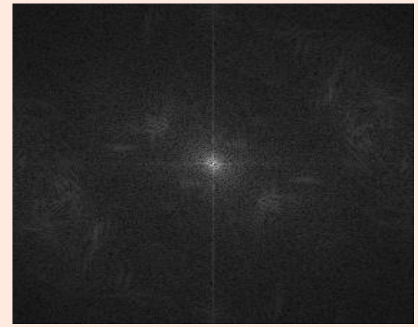
$*$

$h(x,y)$



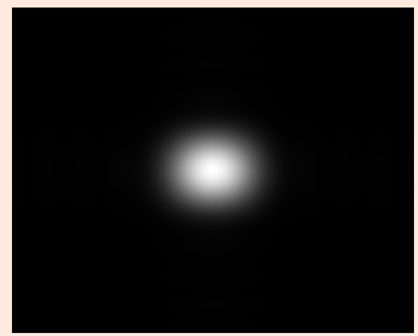
$=$

$g(x,y)$



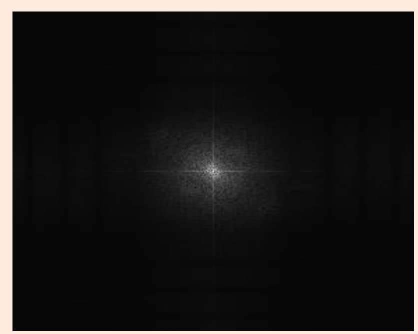
$|F(u,v)|$

$\times$



$|H(u,v)|$

$=$



$|G(u,v)|$

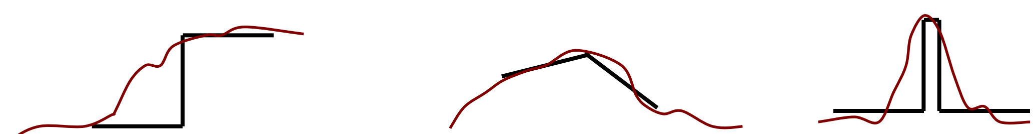
Example: Edge Features

- Edges occur in images where there is discontinuity (or change) in the intensity function.
- Biggest change → derivative has maximum magnitude.
- Gradient points in the direction of most rapid change in intensity

$$f(x, y)$$

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

Small set of example edges:



Step edge   roof edge   line edge

Edge Measures

Edge direction

$$\phi = \theta - \frac{\pi}{2}$$

radient direction

$$\tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

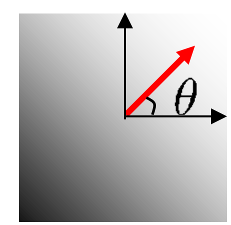
Edge magnitude

$$\left\|\nabla f\right\| = \sqrt{\left[\frac{\partial f}{\partial x}\right]^2 + \left[\frac{\partial f}{\partial y}\right.}$$

$$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$$

$$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$$

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$



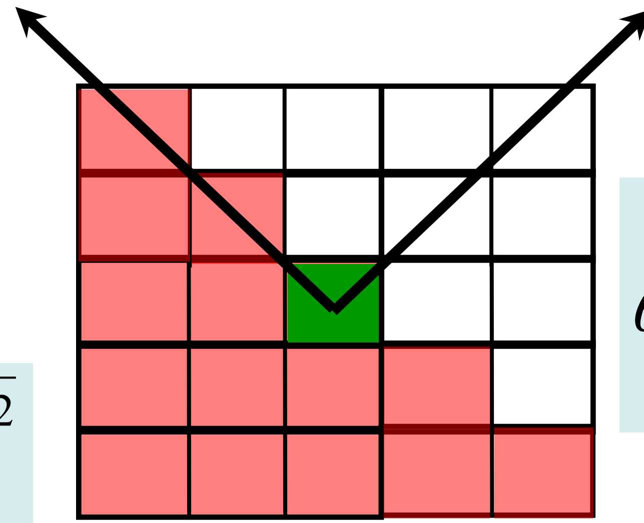
Edge Measures

Edge direction

$$\phi = \theta - \frac{\pi}{2}$$

Edge magnitude

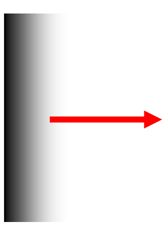
$$\|\nabla f\| = \sqrt{\left[\frac{\partial f}{\partial x}\right]^2 + \left[\frac{\partial f}{\partial y}\right]^2}$$

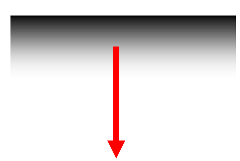


gradient direction

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

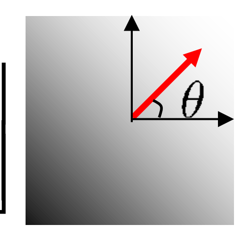
$$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$$





$$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$$

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$



Sobel Edge Detector

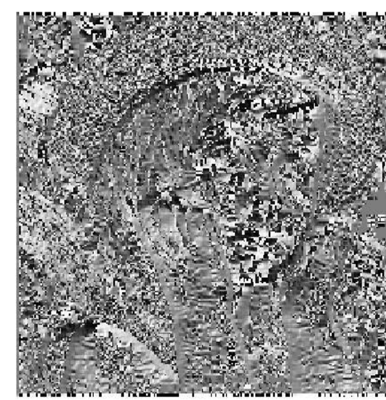
- 2D gradient measurement in two different directions.
- Uses these 3x3 convolution masks:

-1	0	+1
-2	0	+2
-1	0	+1

*

$$\frac{\partial f}{\partial x} =$$

$\|\nabla f\|$



θ

+1	+2	+1
0	0	0
-1	-2	-1

*

$$\frac{\partial f}{\partial y} =$$

Matlab: Sobel Edge Detection

```
% Sobel edge detection
A = imread('house.gif');
fx = [-1 0 1; -2 0 2; -1 0 1]
fy = [1 2 1; 0 0 0; -1 -2 -1]
gx = conv2(double(A),double(fx))/8;
gy = conv2(double(A),double(fy))/8;
mag = sqrt((gx).^2+(gy).^2);
ang = atan(gy./gx);
figure; imagesc(mag); axis off; colormap gray
figure; imagesc(ang); axis off; colormap gray
```

Histogram of Edge Gradients