Repetition and extension of some known concepts
Today’s lecture

Repetition and refinement of some basic concepts:

- What is an image?
- Convolution
- The Fourier Domain
- Sampling
- Aliasing
- Interpolation
- Point operations
- Thresholding
Image definition
Image definition

- **Continuous image**
  - Bounded domain
  - Each point is assigned a scalar value

- **Multi-valued image**
  - Each point is assigned multiple values

- **Discrete image**
  - Sampled version of continuous image
  - Sample values are still assigned real values!

- **Digital image**
  - Sample values also discretized (quantized)
Digitization

- Sampling rate – spatial resolution
- Quantization - grey level resolution
Samples and pixels

- **Sample**: the value \( f(x) \) of a function at a point \( x \)
  - 1-D signals: “samples”
  - 2-D images: “pixels” - picture elements
  - 3-D images: “voxels” – volume elements
  - generally: imels – image elements, spels – spatial elements

- Sampling points
  - geometric arrangement (sampling grid)
  - sampling period (resolution)

How many sample points to use and how to arrange them?
Samples and pixels

Integer grid – samples taken at points with integer coordinates.

Pixels are Voronoi regions of the sample points.

Tiling of a plane (no gaps, no overlaps)

Integer grid – square (cubic) pixels (voxels).

Regular grid - all pixels are regular polygons (tiles). In 2D: squares, equilateral triangles, and hexagons.

Semi-regular – repeated combination of shapes.
Alternative sampling grids

- Do we always need to use rectangular grids?
- Good alternative in 2D: hexagonal grid
  - All direct neighbours at same distance
  - All direct neighbours share an edge (no "connectivity paradox")
Alternative sampling grids

Hexagonal grid – more efficient sampling (13.4% fewer samples needed to recover a signal).
Alternative sampling grids

- Alternatives in 3-D (generalizations of hexagonal grid):
  - **Face centered cubic** (FCC)
    - Voronoi tessellation (voxel) is rhombic dodecahedron
    - Tightest possible packing density (Kepler)
  - **Body centered cubic** (BCC)
    - Voronoi tessellation is truncated octahedron
    - Like hexagonal grid, all direct neighbors share faces
Resolution

Reflects the image quality.

Quality grows with increase of spatial, spectral, radiometric and time resolution.

How is this related to sampling density/pixel size? And how is this related to number of pixels?
Image Acquisition: Sampling

- **Oversampled**: Wastes computer memory
- **Correctly sampled**
- **Undersampled**: Looses information
Image Acquisition: Sampling

How to select correct sampling scheme?

- **Oversampled**: Wastes computer memory
- **Correctly sampled**:
- **Undersampled**: Looses information
Convolution

Brief repetition
Convolution

\[ x(\tau) \]

\[ h(\cdot, \cdot) \]

\[ h(t_1 - \tau) \]

\[ h(2t_1 - \tau) \]

\[ h(3t_1 - \tau) \]

\[ h(-\cdot) \]

\[ h(-t_1 - \tau) \]

\[ h(4t_1 - \tau) \]

\[ h(5t_1 - \tau) \]
Convolution

For **continuous** 1D functions $f$ and $h$

$$(f * h)(t) = \int_{-\infty}^{\infty} f(\xi) h(t-\xi) \, d\xi$$

For **discrete** 1D functions over a finite domain

(sum instead of an integral)

$$(f * h)(n) = \sum_{k=0}^{N-1} f(k) \, h(n-k) = \sum_{k=0}^{N-1} f(n-k) \, h(k)$$
Convolution properties

- **Linear:**
  - **Scaling invariant:** 
    \[(af) * h = a(f * h) = f * (ah)\]
  - **Distributive:** 
    \[f * (g + h) = f * g + f * h\]

- **Translation Invariant:**
  \[T_x(f * g) = T_x(f) * g = f * T_x(g)\]
  \[T_x(f(t)) = f(t - x)\]

- **Commutative:** 
  \[f * h = h * f\]

- **Associative:** 
  \[f * (g * h) = (f * g) * h\]
Linear shift-invariant image filtering

Convolution of the input signal $x$ and the impulse response (kernel) $h$

$$y(m, n) = \sum_{i, j = -\infty}^{\infty} h(m - i, n - j) \cdot x(i, j)$$

$$y[1, 1] = \sum_{j = -\infty}^{\infty} \sum_{i = -\infty}^{\infty} x[i, j] \cdot h[1 - i, 1 - j]$$

$$= x[0, 0] \cdot h[1, 1] + x[0, 1] \cdot h[0, 1] + x[2, 0] \cdot h[-1, 1] + x[0, 1] \cdot h[1, 0] + x[1, 1] \cdot h[0, 0] + x[2, 1] \cdot h[-1, 0] + x[0, 2] \cdot h[1, -1] + x[1, 2] \cdot h[0, -1] + x[2, 2] \cdot h[-1, -1]$$
Convolution at the image edge
Convolution at the image edge

Mean padding

Zero order hold
Convolution at the image edge

Periodic boundary condition

Symmetric boundary condition
Sampling theory

Fourier transform
Aliasing
Discretization

Continuous image 

(if band-limited!)

sampling

Discrete image

Continuous LTI

Continuous image

reconstruction

Continuous image

Discrete LTI

Discrete image

Continuous image
Signal as a sum of periodic functions
Fourier transform in 1D

\[ F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} \, dt \]

\[ f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} \, d\omega \]

\[ F(k) = \sum_{n=0}^{N-1} f(n) e^{-i\frac{2\pi}{N}kn} \]

\[ f(n) = \frac{1}{N} \sum_{k=0}^{N-1} F(k) e^{i\frac{2\pi}{N}kn} \]
Fourier transform pairs

$F(-\omega) = F^*(\omega)$

**Spatial**
- impulse
- cosine
- sine
- box
- sinc
- Gaussian

**Frequency**
- 1 impulse
- 2 impulses
- 2 impulses
- sinc
- box
- Gaussian

Notice the symmetry!
Properties of the Fourier transform

- **Linearity**
  \[ F \{ cA + dB \} = cF \{ A \} + dF \{ B \} \]

- **Convolution**
  \[ F \{ A \ast B \} = F \{ A \} \cdot F \{ B \} \]

- **Spatial scaling**

- **Spatial shift**

- **Symmetry**
  - real, even
  - real, odd
  - imaginary, odd
Fourier transform in 2D, 3D, etc.

- FT is separable:
  - Perform transform along x-axis (row-wise)
  - Perform transform along y-axis of result (column-wise)
  - Perform transform along z-axis of result, (etc.)
- Properties are same as for 1-D Fourier Transform
Aliasing

Sampling distance is large (sampling rate is low)

The reconstructed signal is not the desired one
Aliasing
Revisiting sampling

Spatial domain:
- Continuous function
- Sampling function
- Sampled function

Frequency domain:
- Continuous function
- Sampling function
- Sampled function
Avoiding aliasing
Avoiding aliasing

Low-pass filtered image does not contain high frequencies
Reconstruction

Interpolation kernels
Reconstruction

spatial domain

frequency domain

sampled function

ideal interpolator

continuous function
Multiplying a spectrum by a box filter is equivalent to convolving a (sampled) function with the box filter's inverse Fourier transform.

The **filter kernel** (Fourier filter pair element from a spatial domain) of a box filter is a sinc function, $\text{sinc}(x) = \frac{\sin(x)}{x}$. 
Ideal reconstruction
Interpolation by a sinc function

Sinc function is zero at all other grid points when its origin matches a grid point. Its amplitude can be scaled so that it equals the sample value at the origin.
Ideal reconstruction

Normalized sinc function:
\[ \frac{\sin(\pi x)}{\pi x} \]
Zero-values at integer points.

This function has infinite support!
Alternative interpolation kernels
Nearest neighbour interpolation

uses 1 data point
Alternative interpolation kernels
Linear interpolation

Linear interpolation uses 2 data points.
Linear interpolation

The triangle function and its spectrum.
Alternative interpolation kernels
Cubic spline interpolation

4 cubic polynomials pieced together uses 4 data points
Alternative interpolation kernels

a) Nearest neighbour, c) Cubic spline (a=-1)
b) Linear, d) Cubic spline  a=0.5)

\[
f(x) = (a + 2)x^3 - (a + 3)x^2 + 1 \quad (0, 1)
\]

\[
f(x) = ax^3 - 5ax^2 + 8ax - 4a \quad (1, 2).
\]
Alternative interpolation kernels

\[ L(x) = \begin{cases} \sin(x) \sin(x/a) & \text{if } -a < x < a \\ 0 & \text{otherwise} \end{cases} \]

lanczos filter (a=2)

\[ \sin(\pi x) / (\pi x) \times \sin(\pi x/2) / (\pi x/2) \]

uses 4 data points
Alternative interpolation kernels

\[
\frac{\sin(\pi x)}{\pi x} \times \frac{\sin(\pi x/4)}{\pi x/4}
\]

Uses 8 data points

Lanczos filter (\(a=4\))
What is trilinear interpolation?

- In $n$-D images, interpolation can be done on each dimension independently:

  - “bilinear” means linear interpolation in 2-D image
  - “trilinear” means linear interpolation in 3-D image
Basic image operations

A very brief repetition
Basic image operations

Image arithmetics

Point operations
  - Function of image (pixel) values
  - Independent of spatial location

Geometric transforms (lecture on Feb 12):
  - Function applied to image coordinates
  - Independent of image values

Filtering (next 2 lectures):
  - Function that changes image values based on local neighbourhood
Point operations

- Transform each pixel value in the image, independent of pixel location
  - Increase contrast
  - Bring interesting grey-value range in view
  - Make details visible

- Commonly used operations
  - Change gamma
    \[ f(y) = y^\gamma \]
  - Contrast stretch
    \[ f(y) = ay \]
  - Logarithmic stretch
    \[ f(y) = a \log(y) \]
  - Clipping
    \[ f(y) = \begin{cases} \frac{y}{a}, & y \geq a \\ y, & otherwise \end{cases} \]
  - Histogram equalization
    \[ f(y) = \text{data-dependent} \]
  - Thresholding
    \[ f(y) = y > a \]
Thresholding
Grey level thresholding

- Simplest form of segmentation
- Associates each pixel to object or background based on the pixel's grey value
  - **Static or global threshold**: same threshold $T$ for all pixels
    \[
    g(x, y) = \begin{cases} 
    1, & f(x, y) > T \\
    0, & \text{otherwise} 
    \end{cases}
    \]
  - **Adaptive or local threshold** depends on pixel intensity, but also some other local property, (e.g. neighbourhood values)
    \[
    g(x, y) = \begin{cases} 
    1, & f(x, y) > T(x, y) \\
    0, & \text{otherwise} 
    \end{cases}
    \]
  - **Hysteresis thresholding**: combine the two approaches.
Finding a threshold level

For a global threshold, all relevant information is in the image histogram

- No neighbourhood information is used, only individual grey values.
- Many images have the same histogram!
- Object and background intensities sometimes have clearly distinct values.
- Uniform illumination is very important!
Bimodal histograms – where to threshold

Two peaks indicate the grey levels most common for the two classes. Threshold selection – minimal resulting segmentation error.

Minimum of the histogram? Possibly not unique.
Bimodal histograms

- Minimum error
- Theoretically optimal threshold
- Local minimum between peaks
Bimodal histograms – optimal thresholding

**k-means clustering** (Ridler and Calvard, 1978)
Assumes the two modes are of similar width and height.
Threshold is average of the (unknown) means (cluster centres)
Iterative method, depends on initialisation.

**Minimizing intra-class variance** (“Otsu” method, 1979)
Equivalent to maximizing variance between the classes:

\[
\sigma^2_{b}(t) = P_1(t) P_2(t) [\mu_1(t) - \mu_2(t)]^2
\]

**Minimizing error** (Kittler and Illingworth, 1986)
Assumes two normal distributions, and determines threshold that
minimizes the classification error, by minimizing the function

\[
J(t) = 1 + 2 [P_1(t) \log \sigma_1(t) + P_2(t) \log \sigma_2(t)] - 2 [P_1(t) \log P_1(t) + P_2(t) \log P_2(t)]
\]
Improving the histogram

Foreground peak much smaller than background peak

histogram of gradient magnitudes only
Unimodal histograms

Background peak

Foreground doesn't have a peak!
Unimodal histograms

T-point: Fitting two lines, sets the threshold at the intersection of two best fits (Coudray, Buessler, Urban 2010)

Chord method, a.k.a.:
- skewed bi-modality
- maximum distance to triangle
Zack, Rogers and Latt (1977)
Rosin (2001)
Finding a threshold level

Other methods used besides histogram analysis:

- Manual determination on a training set of images
  - Threshold becomes a “magic number”
  - Results useless if imaging circumstances change
- Using *a priori* knowledge:
  - Volume: if it is known that 25% of the image is foreground, choose a threshold value so that 25% of the pixels are above it
  - Shape: if round objects are expected, choose a threshold value that maximizes some roundness measure of the result
- …
Multi-channel threshold
Multi-channel threshold
Hysteresis threshold

High threshold

Low threshold

Regions in “low” that have some pixels set in “high”
Summary
Summary of today’s lecture

- Digital image is a discrete and quantized representation of a continuous image function.
- Proper sampling (spatial, intensity, temporal) makes this representation useful.
- How to sample – Fourier analysis and Nyquist theorem.
- Convolution and its properties.
- Convolution and Fourier transform: \( f(t) * h(t) \Rightarrow F(\omega) \cdot H(\omega) \)
- Convolution and interpolation
- Thresholding – some approaches.