Sampling and Quantisation
Sampling & quantisation

- 1. Discretization of continuous signals
- 2. Signal representation in the spatial and frequency domain
- 3. Effects of sampling and quantisation
- 4. More on sampling
- 5. More on quantisation
Learning objectives: what can you do after today?

- Describe how images are discretized
- Describe effects of choices involved in the discretization process
- Describe the mathematics behind sampling
- Describe convolution, Fourier transform of images and convolution theorem
- Given the properties of an image decide whether a loyal reconstruction is possible from samples
- Describe basic quantization
- Recognize quantization artifacts
Discretisation

Computer to process an image:

1. sampling → “pixels”

2. quantisation → “grey levels”
Sampling & quantization
Sampling schemes

regular, image covering tessellation

11 with regular polygons ▶ 3 if equal

rectangular (square) most popular

hexagonal has advantages (more isotropic, no connectivity ambiguities, …) + similar structure in retina
Example of sampling:

384 x 288 pixels

192 x 144 pixels

92 x 72 pixels

48 x 36 pixels
Example of quantisation:

- 2 levels - binary
- 4 levels
- 8 levels
- 256 levels – 1 byte
Image distortion through sampling
Image distortion through quantisation
Remarks

- 1. Importance of binary images

- 2. Non-uniform sampling and/or quantisation
  - a. fine sampling for details
  - b. fine quantisation for homogeneous regions
A model for sampling

1. Integrate brightness over cell window

   Image degradations

2. Read out values only at the pixel centers

   Aliasing
   Leakage
STEP 1: integrating over a pixel cell

\[
o(x', y') = \int \int i(x, y) p(x - x', y - y') dx dy
\]
STEP 2: local probing of functions

Distributions as extension of functions: the Dirac pulse

\[ \delta(x - x_0) = 0 \quad x \neq x_0 \]

\[ \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} \delta(x - x_0) \, dx = 1 \]

Function probing (in 1D)

\[ \int_{-\infty}^{\infty} \delta(x) f(x) \, dx = f(0) \]

\[ \int_{-\infty}^{\infty} \delta(x - x_0) f(x) \, dx = f(x_0) \]
Spatial domain characterization

Decomposition of a function into individual pulses

\[ f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta) \delta(x - \alpha, y - \beta) d\alpha d\beta \]

Signals described in a linear space through decomposition in an orthonormal basis
A system view on image/signal processing

Input: $f(x,y)$

System: $\emptyset$

Output: $g(x,y)$
Linear, shift-invariant operators

1. Linear :

- $f_1 \rightarrow g_1$
- $f_2 \rightarrow g_2$
- $af_1 + bf_2 \rightarrow ag_1 + bg_2$

2. Shift-invariant :

- $f(x,y) \rightarrow g(x,y)$
- $f(x - a, y - b) \rightarrow g(x - a, y - b)$
Characterization of LSI systems through spatial domain pulses

\[ g(x, y) = O[f(x, y)] \]
\[ = O \left[ \int \int f(\alpha, \beta)\delta(x - \alpha, y - \beta) d\alpha d\beta \right] \]
\[ = \int \int f(\alpha, \beta) O [\delta(x - \alpha, y - \beta)] d\alpha d\beta \]
\[ = \int \int f(\alpha, \beta) r(x - \alpha, y - \beta) d\alpha d\beta \]

*convolution* of \( f \) and \( r \)

\[ g(x, y) = f(x, y) \ast r(x, y) \]
Convolution

$$o(i,j) = c_{11} f(i-1,j-1) + c_{12} f(i-1,j) + c_{13} f(i-1,j+1) +$$

$$c_{21} f(i,j-1) + c_{22} f(i,j) + c_{23} f(i,j+1) +$$

$$c_{31} f(i+1,j-1) + c_{32} f(i+1,j) + c_{33} f(i+1,j+1)$$
An example of convolution

\[ o(x', y') = \int \int i(x, y) p(x - x', y - y') dx dy \]

This is a convolution: \( i(x, y) * p(-x, -y) \)
Characteristics of convolution

\[ f \ast g = g \ast f \]

\[ (f \ast g) \ast h = f \ast (g \ast h) \]

\[ k = h \ast f \]

\[ = (h_1 \ast h_2) \ast f \]

\[ = h_1 \ast (h_2 \ast f) \]
Alternative characterization of functions: The frequency domain

Orthonormal basis functions

$$e^{i2\pi(ux+vy)} = \cos 2\pi(ux + vy) + i \sin 2\pi(ux + vy)$$

$$\lambda = \frac{1}{\sqrt{u^2 + v^2}}$$

Eigenfunctions of LSI systems

$$\mathcal{O}[e^{i2\pi(ux+vy)}] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{i2\pi(u(x-\alpha)+v(y-\beta))} r(\alpha, \beta) d\alpha d\beta =$$

$$= e^{i2\pi(ux+vy)} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i2\pi(u\alpha+v\beta)} r(\alpha, \beta) d\alpha d\beta = Ae^{i\phi} e^{i2\pi(ux+vy)}$$
The Fourier transform

Linear decomposition of functions in the new basis
Scaling factor for basis function \((u,v)\)

\[
\mathcal{F}[f(x, y)] = F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)e^{-i2\pi(ux+vy)}dxdy
\]

→ The Fourier transform

Reconstruction of the original function in the spatial domain: weighted sum of the basis functions

\[
\mathcal{F}^{-1}[F(u, v)] = f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v)e^{i2\pi(ux+vy)}dxdy
\]

→ The inverse Fourier transform

\[
f(x, y) = \int_{-\infty}^{\infty} f(\alpha, \beta)\delta(x - \alpha, y - \beta)d\alpha d\beta
\]
Fourier coefficients

$F(u, v)$ is complex: $F_R(u, v) + iF_I(u, v)$

The magnitude

$$|F(u, v)| = \sqrt{F_R(u, v)^2 + F_I(u, v)^2}$$

The phase angle

$$\arctan \left( \frac{F_I(u, v)}{F_R(u, v)} \right)$$
Fourier decomposition of images

\[ f(x,y) = F(u,v) + F(u',v') + F(u'',v'') + \ldots \]

\[ = F(u,v) \times x + F(u',v') \times x + F(u'',v'') \times x + \ldots \]
Fourier decomposition of images
Fourier decomposition of images
Example importance of magnitude

- Image with periodic structure

\[ f(x,y) \]

\[ |F(u,v)| \]

FT has peaks at spatial frequencies of repeated texture
Example importance of magnitude

Periodic background removed

$|F(u,v)|$

remove peaks
\( f(x,y) \)

- \(|F(u,v)|\) generally decreases with higher spatial frequencies
- phase appears less informative
The importance of the phase
Some relevant properties

\[ f(x, y) \leftrightarrow F_R(u, v) + iF_I(u, v) \]

<table>
<thead>
<tr>
<th>spatial domain</th>
<th>frequency domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>real</td>
<td>real part even</td>
</tr>
<tr>
<td></td>
<td>imaginary part odd</td>
</tr>
<tr>
<td>real, even</td>
<td>real, even</td>
</tr>
<tr>
<td>real, odd</td>
<td>imaginary, odd</td>
</tr>
</tbody>
</table>
Computer Vision

Rotation:

[Diagram showing rotated images]
Scaling
Affine
The convolution theorem

\[ c(x, y) = a(x, y) * b(x, y) \]

\[ \downarrow \text{ Fourier} \]

\[ C(u, v) = \]

\[ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[ a(x, y) \ast b(x, y) \right] e^{-i2\pi(ux+vy)} dx \, dy \]

\[ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} a(x - \alpha, y - \beta) b(\alpha, \beta) d\alpha d\beta \right] e^{-i2\pi(ux+vy)} dx \, dy \]

\[ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} a(x - \alpha, y - \beta) e^{-i2\pi(ux+vy)} dx \, dy \right] b(\alpha, \beta) d\alpha d\beta \]
The convolution theorem

\[
\begin{aligned}
&\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} a(x - \alpha, y - \beta) e^{-i2\pi(ux+vy)} \, dx \, dy \right] b(\alpha, \beta) \, d\alpha \, d\beta \\
&\quad \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} a(x^*, y^*) e^{-i2\pi(u(x^*+a)+v(y^*+b))} \, dx^* \, dy^* \right] b(\alpha, \beta) \, d\alpha \, d\beta \\
&\quad \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} a(x^*, y^*) e^{-i2\pi(ux^*+vy^*)} \, dx^* \, dy^* \right] e^{-i2\pi(ua+vb)} b(\alpha, \beta) \, d\alpha \, d\beta \\
\end{aligned}
\]

That is,

\[
\begin{aligned}
&\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A(u, v) e^{-i2\pi(u\alpha+v\beta)} b(\alpha, \beta) \, d\alpha \, d\beta \\
&\quad = A(u, v) B(u, v)
\end{aligned}
\]

Space convolution = frequency multiplication
Modulation transfer function for LSI

\[ O(u, v) = \mathcal{F}\{o(x, y)\} \]
\[ = \mathcal{F}\{i(x, y) \ast r(x, y)\} \]
\[ = I(u, v)R(u, v) \]

\[ R(u, v) = \mathcal{F}\{r(x, y)\} \]
\[ = \mathcal{F}\{\text{point spread function}\} \]

\[ = \text{modulation transfer function} \]
The convolution theorem: reciprocity

\[ C(u, v) = A(u, v)B(u, v) \]
\[ c(x, y) = a(x, y) \ast b(x, y) \]

\[ C(u, v) = A(u, v) \ast B(u, v) \]
\[ c(x, y) = a(x, y)b(x, y) \]

Space multiplication = frequency convolution
A model for sampling

… back to STEP 1

1. Integrate brightness over cell window
   - Image degradations

2. Read out values only at the pixel centers
   - Aliasing
   - Leakage
STEP 1: integrating over a pixel cell

\[ o(x', y') = \int \int i(x, y) p(x - x', y - y') dx dy \]

This is convolution:

\[ i(x, y) * p(-x, -y) \]

\[ O(u, v) = I(u, v) P(u, v) \]
Fourier transform of window:

\[ P(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i2\pi(ux+vy)} p(x, y) \, dx \, dy \]

\[ = \int_{-w/2}^{w/2} e^{-i2\pi ux} \, dx \int_{-h/2}^{h/2} e^{-i2\pi vy} \, dy \]

\[ = \left[ \frac{e^{-i2\pi ux}}{-i2\pi u} \right]_{-w/2}^{w/2} \left[ \frac{e^{-i2\pi vy}}{-i2\pi v} \right]_{-h/2}^{h/2} \]

\[ = -\frac{1}{4\pi^2 uv} (-2i \sin(2\pi u \frac{w}{2}))(-2i \sin(2\pi v \frac{h}{2})) \]

\[ = \frac{wh}{\pi wu} \left( \frac{\sin \pi w u}{\pi w u} \right) \left( \frac{\sin \pi h v}{\pi h v} \right) \]
Fourier transform of the window function

2D sinc:

real

no phase shifts!
Illustration of the sinc
The convolution theorem: exercise

- What is the FT of $f(x, y)$?

$f(x, y)$

$\* \quad = \quad$
A model for sampling

… back to STEP 2

1. Integrate brightness over cell window
   Image degradations

2. Read out values only at the pixel centers
   Aliasing
   Leakage
STEP 2: discrete representation of functions
In the spatial and frequency domain

a) Discretizing in the spatial domain
b) Limiting the spatial extent
c) Discretizing in the frequency domain
Discretizing in the spatial domain

multiplication with 2D pulse train

\[ \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \delta(x-kw, y-lh) \]

Fourier transform:

\[ \frac{1}{wh} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \delta(x - k \frac{1}{w}, y - l \frac{1}{h}) \]

Convolution with a Dirac train: periodic repetition
Yet another duality: discrete vs. periodic
Discretizing in the spatial domain
The sampling theorem

If the Fourier transform of a function $f(x,y)$ is zero for all frequencies beyond $u_b$ and $v_b$, i.e. if the Fourier transform is band-limited, then the continuous periodic function $f(x,y)$ can be completely reconstructed from its samples as long as the sampling distances $w$ and $h$ along the x and y directions are such that $w \leq \frac{1}{2u_b}$ and $h \leq \frac{1}{2v_b}$.
Thus convolve Fourier transform with pulse train
Result is periodic Fourier transform
Select one period:
multiply window

Periods must not overlap (hence Nyquist rate)

Interpolation:
Result is an exact interpolation
Sinc is the optimal interpolation function
Perfect representation by samples

- The signal is band limited
- It is sampled at or above the Nyquist limit
### Aliasing

<table>
<thead>
<tr>
<th>SPATIAL DOMAIN</th>
<th>FREQUENCY DOMAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Spatial Waveform" /></td>
<td><img src="image2" alt="Frequency Waveform" /></td>
</tr>
<tr>
<td><img src="image3" alt="Spatial Sampling" /></td>
<td><img src="image4" alt="Frequency Sampling" /></td>
</tr>
<tr>
<td><img src="image5" alt="Aliasing Error" /></td>
<td><img src="image6" alt="Aliasing Example" /></td>
</tr>
</tbody>
</table>
Aliasing: 1D example

Insufficient samples to distinguish the high and the low frequency
Example:

![Aliasing 1 Example](image-url)
Aliasing 2

oversampled example:
Aliasing 2

oversampled example:

![Image of oversampled example]
Aliasing 3

subsampled example:
Aliasing 4

schematic example:

<table>
<thead>
<tr>
<th>OVERSAMPLED</th>
<th>UNDERSAMPLED</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

The diagram illustrates the difference between oversampled and undersampled data points in a grid.
sinc interpolation

sinc has infinite support
Cannot be implemented in practice
Approximations through splines

<table>
<thead>
<tr>
<th>Order</th>
<th>Function $p_n(x)$</th>
<th>Spatial</th>
<th>MTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$p_0(x) = \frac{1}{\Delta x} \text{rect}\left(\frac{x}{\Delta x}\right)$</td>
<td>$-\Delta x/2$ to $\Delta x/2$</td>
<td>$1/\Delta x$</td>
</tr>
<tr>
<td>1</td>
<td>$p_1(x) = p_0(x) \ast p_0(x)$</td>
<td>$-\Delta x$ to $\Delta x$</td>
<td>$1/\Delta x$</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>$p_n(x) = p_0(x) \ast \ldots \ast p_0(x)$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Gaussian: $\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$
Comparison of interpolation kernels

With increasing order
- decreasing aliasing
- increasingly lowpass
1D example

$0^{th}$ order kernel: hold
Very efficient: minimal support

Triangular kernel: linear interpolation
broader support
Limiting the spatial extent
Leakage

• Caused by convolving the periodic (continuous) spectrum with $sinc$
• Mixing up frequencies over the whole spectrum
• Can be compensated for by the subsequent sampling in the frequency domain
Discretization in the frequency domain
Perfect representation by samples

- The signal is band limited
- It is sampled at or above the Nyquist limit
- The signal is periodic
- Sampling is compatible with the signal period
  no phase difference after a finite number of periods
Discretization in the frequency domain
The discrete Fourier pair

In both domains
• discrete
• periodic

Defines the discrete Fourier Transform

\[ F(k,l) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)e^{-2\pi i \left( \frac{km}{M} + \frac{ln}{N} \right)} \]

\[ f(m,n) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F(k,l)e^{2\pi i \left( \frac{mk}{M} + \frac{nl}{N} \right)} \]
Remarks on the DFT

1. Periodicity assumed in both domains, might introduce false high frequencies at image boundaries.

2. Efficient implementations exist (e.g., FFT which is $N \log_2 N$ instead of $N^2$).
Quantisation

Create K intervals in the range of possible intensities measured in bits: $\log_2(K)$

Design choices
- Decision levels
  \[ z_1, z_2, \ldots, z_{K+1} \]
- Representative value
  \[
  \text{interval } \left[ z_k, z_{k+1} \right] \rightarrow q_k
  \]
- Simplest selection
  - equal intervals
  - value is the mean
  - $\Delta$ uniform quantizer
The uniform quantizer

• simple implementation
• fine quantization needed perceptually (7-8 bits)
• can be reduced by optimal design, e.g.

\[
\minimize \delta = \sum_{k=1}^{K} \int_{z_k}^{z_{k+1}} (z - q_k)^2 p(z) \, dz := \sum_{k=1}^{K} \delta_k
\]

\(p(z)\) = prob. density function, for constant \(\Delta\) uniform
Underquantization example

256 gray level (8 bit)  11 gray level
Remarks

• Quantization:
  – Often 8 bits per pixel (monochrome), 24 bits per pixel (RGB)
  – Medical images 12 bits (4096 levels) or 16 bits (65536 levels)

• Size
  – Cameras typically few K width
  – Satellite images 2-100K width
Learning objectives: what can you do after today?

- Describe how images are discretized
- Describe effects of choices involved in the discretization process
- Describe the mathematics behind sampling
- Describe convolution, Fourier transform of images and convolution theorem
- Given the properties of an image decide whether a loyal reconstruction is possible from samples
- Describe basic quantization
- Recognize quantization artifacts