Introduction to Image Processing

Lecturer:
Dr. mat. nat. Christian Kehl

Week 1.2 – Basics and Practice

November 13, 2024

Fundamentals of Images

Signal- & Information Theory – Fundamentals

Major subtopics:

1. Mathematical representation
2. Digitization / Analog-Digital conversion (ADC)
   1. Sampling & grid resolution
   2. Quantization & value resolution

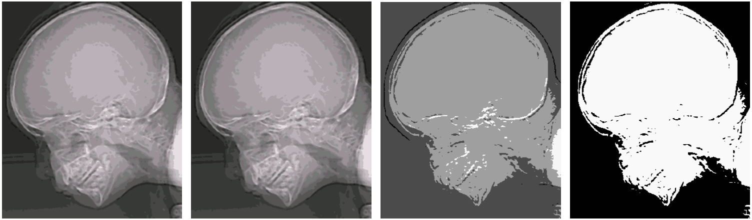
Signal- & Information Theory – Fundamentals

Major subtopics:

1. Mathmematical representation
2. Digitization / Analog-Digital conversion (ADC)
   1. Sampling & grid resolution
   2. **Quantization & value resolution**

Fundamentals: Information Theory & Quantization

Adapting quantization

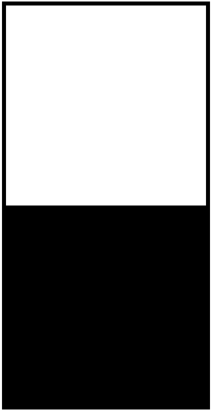


(a) 16 levels (b) 8 levels (c) 4 levels (d) 2 levels

Q: *How many bits do you need to store the information without visible quality loss ?*

Values & information

(a) Binary image



(b) no. of value states: 2

1-Bit case: each pixel has 1 of 2 **states** – black or white

Values & information

Stochstic perspective:

- each state: *event x*
- (uniform) probability Pr of event x: $Pr(x) = 0.5$
- digital information are stored as *binary numbers*
  $\rightarrow$ *base b* to express event x: 2

Values & information

Stochstic perspective:

- each state: *event x*
- (uniform) probability Pr of event x: $Pr(x) = 0.5$
- digital information are stored as *binary numbers*
  $\rightarrow$ *base b* to express event x: 2

Shannon information: $I(x) = -log_2[Pr(x)]$
$\rightarrow$ *information content of a pixel*

Values & information

(a) 2 levels: $I(x) = ?$ [bit]

(b) 4 levels: $I(x) = ?$ [bit]

(c) 8 levels: $I(x) = ?$ [bit]

Values & information

(a) 2 levels: $I(x) = 1$ [bit]

(b) 4 levels: $I(x) = 2$ [bit]

(c) 8 levels: $I(x) = 3$ [bit]

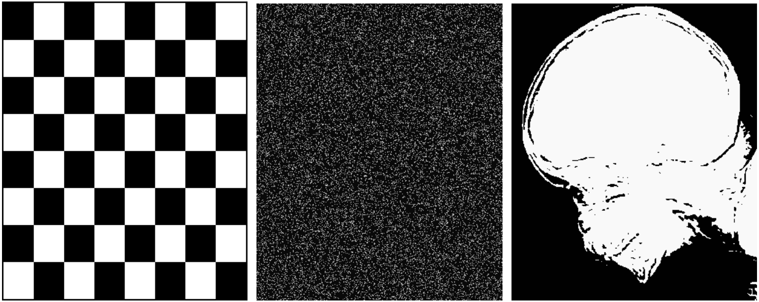
Values & information

Properties of information:

- An *certain* event x with just 1 state (thus, $Pr = 1.0 = 100\%$) yields no information
- The less probable an event is, the more surprising it is and the more information it yields.
- If two independent events x_1 and x_2 are measured separately, their total information content $I(X)$ is the sum of information of both individual events.
- The total probability $Pr(X)$ of all correlated events $x \in X$ is $1.0(100\%)$.

Q: *Is the information content of an image the sum of information for each pixel event ?*

Pixel- and image information



(a) no. white pixels: 65600 (50%)

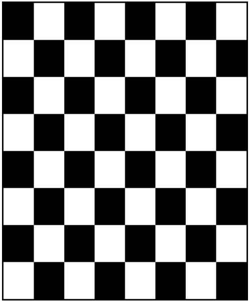
(b) no. white pixels: 20105 (15%)

(c) no. white pixels: 78631 (59%)

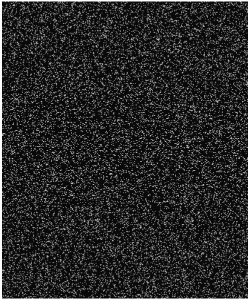
Q: *Is the information content an image the sum of information for each pixel event ?*

A: Apparently not. Why ?

Pixel- and image information



(a) no. white pixels: 65600 (50%)



(b) no. white pixels: 20105 (15%)

(c) no. white pixels: 78631 (59%)

Q: *Is the information content an image the sum of information for each pixel event ?*

A: Apparently not. Why ? $Pr(x) \neq 0.5$ (probability depends on content)

Pixel- and image information

Skull example - event set X with discrete, random events $x \in X$:

- $I(x = 0) = -log_2[0.41] = 1.2863$
- $I(x = 1) = -log_2[0.59] = 0.7612$

This is the self-information $I(x \in X)$ for each individual event.

Pixel- and image information

Skull example - event set X with discrete, random events $x \in X$:

- $I(x = 0) = -log_2[0.41] = 1.2863$
- $I(x = 1) = -log_2[0.59] = 0.7612$

This is the self-information $I(x \in X)$ for each individual event.

The average information $H(X)$ (also called **entropy**) is the content-related information.
Shannon entropy: $H(X) = -\sum Pr(x) log_2[Pr(x)] \; \forall x \in X$

Image information - application

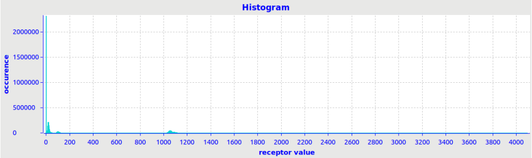
Skull example:

- image source: computed tomography
- typical storage format: DICOM - 16-bit for 1 channel
- information expressed as *Houndsfield units (HU)*
- $HU = 0..2000 \rightarrow 2000$ levels
- $I(x) = -log_2[Pr(x)] = 10.96$
- self-information $I(x) < 16$ bit (short-integer)

Image information - application

Skull example - content:

(a) Skull-CT



(b) Associated histogram

Image information - application

Skull example - content:

- full-head CT; $pdf(HU)$, see histogram
- $I(x) = 10.96$
- for more than 50% of $x \in X$: $Pr(x) = 0$
- $Pr(x_0 = 0) = 0.6$
- $Pr(x_1) = \mathcal{N}(10, 20)$
- $Pr(x_2) = \mathcal{N}(100, 10)$
- $Pr(x_3) = \mathcal{N}(1050, 30)$
- $H(X) = 5.727 < I(x)$

Information & Quantization - wrap-up

Take-away message 1: no. bits to store a given content $\geqslant I(x)$

Information & Quantization - wrap-up

Take-away message 1: no. bits to store a given content $\geqslant I(x)$
Take-away message 2: $I(x) \neq H(X|X = \{0..x_{max}\})$
$\rightarrow$ compression (later)

Information & Quantization - wrap-up

Take-away message 1: no. bits to store a given content $\geqslant I(x)$
Take-away message 2: $I(x) \neq H(X|X = \{0..x_{max}\})$
$\rightarrow$ compression (later)
Take-away message 3: **data** $\neq$ **information**
sizeof(X) $\neq I(x) \neq H(X)$

Comparing images - Mutual Information

Joint (2D) histogram: **information** shared between two images
$P(\mathrm{I}_1, \mathrm{I}_2) = P(\mathrm{I}_1 = k | \mathrm{I}_2 = l) \qquad \forall \{k, l\} \in \{0..255\}$

Comparing images - Mutual Information

Joint (2D) histogram: **information** shared between two images
$P(\mathrm{I}_1, \mathrm{I}_2) = P(\mathrm{I}_1 = k | \mathrm{I}_2 = l) \qquad \forall \{k, l\} \in \{0..255\}$
Shannon information: $I(x) = -log_2[Pr(x)]$

Comparing images - Mutual Information

Joint (2D) histogram: **information** shared between two images
$P(\mathrm{I}_1, \mathrm{I}_2) = P(\mathrm{I}_1 = k | \mathrm{I}_2 = l) \qquad \forall \{k, l\} \in \{0..255\}$
Shannon information: $I(x) = -log_2[Pr(x)]$
Shannon entropy: $H(X) = -\sum_{x=0}^{|X|} P(x) log_2[P(x)]$

Comparing images - Mutual Information

Joint (2D) histogram: **information** shared between two images

$P(\mathrm{I}_1, \mathrm{I}_2) = P(\mathrm{I}_1 = k | \mathrm{I}_2 = l) \qquad \forall \{k, l\} \in \{0..255\}$

Shannon information: $I(x) = -log_2[Pr(x)]$

Shannon entropy: $H(X) = -\sum_{x=0}^{|X|} P(x) log_2[P(x)]$

Shannon entropy: $H(X, Y) = -\sum_{x=0}^{|X|} \sum_{y=0}^{|Y|} P(x, y) log_2[P(x, y)]$

Comparing images - Mutual Information

Joint (2D) histogram: **information** shared between two images

$P(\mathrm{I}_1, \mathrm{I}_2) = P(\mathrm{I}_1 = k | \mathrm{I}_2 = l) \qquad \forall \{k, l\} \in \{0..255\}$

Shannon information: $I(x) = -log_2[Pr(x)]$

Shannon entropy: $H(X) = -\sum_{x=0}^{|X|} P(x) log_2[P(x)]$

Shannon entropy: $H(X, Y) = -\sum_{x=0}^{|X|} \sum_{y=0}^{|Y|} P(x, y) log_2[P(x, y)]$

Mutual information: $MI(\mathrm{I}_1, \mathrm{I}_2) = H(\mathrm{I}_1) + H(\mathrm{I}_2) - H(\mathrm{I}_1, \mathrm{I}_2)$

Comparing images - Mutual Information

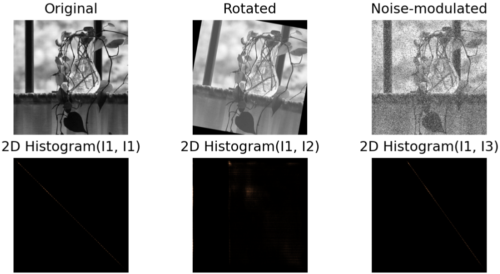
Joint (2D) histogram: **information** shared between two images

$P(\mathrm{I}_1, \mathrm{I}_2) = P(k \in \mathrm{I}_1 | l \in \mathrm{I}_2)$

Shannon entropy: $H(X) = -\sum_{x=0}^{|X|} P(x) log_2[P(x)]$

Shannon entropy: $H(X, Y) = -\sum_{x=0}^{|X|} \sum_{y=0}^{|Y|} P(x, y) log_2[P(x, y)]$

Mutual information: $MI(\mathrm{I}_1, \mathrm{I}_2) = H(\mathrm{I}_1) + H(\mathrm{I}_2) - H(\mathrm{I}_1, \mathrm{I}_2)$



$MI(\mathrm{I}_1, \mathrm{I}_1) = 7.23; MI(\mathrm{I}_1, \mathrm{I}_2) = 0.50;\ MI(\mathrm{I}_1, \mathrm{I}_3) = 5.92$

Practice Sampling & Discretization

Adapting single-channel quantization

Task: subdivide or reallocate value space → quantization

Adapting single-channel quantization

Task: subdivide or reallocate value space $\rightarrow$ quantization
how to determine the new quantization levels ?

Adapting single-channel quantization

Task: subdivide or reallocate value space $\rightarrow$ quantization

how to determine the new quantization levels ? use histogram

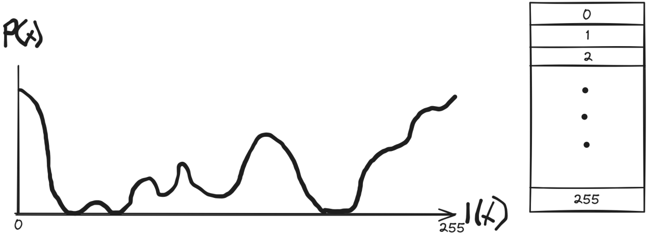
how to re-distribute quantization levels ?

Adapting single-channel quantization

Task: subdivide or reallocate value space $\longrightarrow$ quantization
how to determine the new quantization levels ? use histogram
how to re-distribute quantization levels ? use *look-up table*

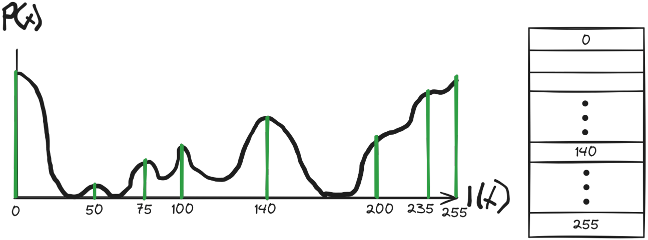
Histogram & look-up table

Look-up table: re-mapping of levels, i.e. quantization



Histogram & look-up table

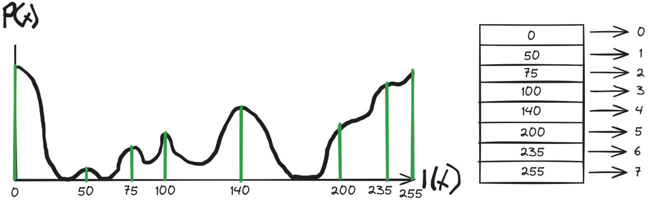
Look-up table: re-mapping of levels, i.e. quantization



Identify quantization via m peaks in $P(x)$

Histogram & look-up table

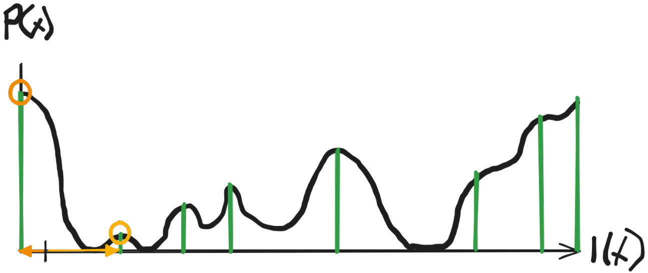
Look-up table: re-mapping of levels, i.e. quantization



Memory map of gray levels to new bit- or level patterns

Histogram & look-up table

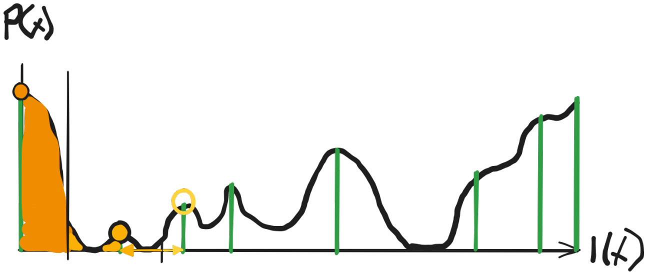
Look-up table: re-mapping of levels, i.e. quantization



Iterate through original gray levels, map to represenative peak by distance $d(f(x_1), f(x_2)) = |f(x_2) - f(x_1)|$

Histogram & look-up table

Look-up table: re-mapping of levels, i.e. quantization



Distance compare always as $f(P_{max_k}(x)) < f(x) < f(P_{max_{k+1}}(x))$, until $f(x) = f_{max}(x)$

Histogram & look-up table

```
Initialize nlevels[0..k-1] via histogram peak levels
Initialize array LUT[0..p-1], p = # original levels
thispeak = nlevels[0] and nextpeak = nlevels[1]
thispeak_index = 0
For i in levels
  dist_current = abs(i-thispeak)
  dist_next = abs(i-nextpeak)
  If dist_next < dist_current
    LUT[i] = thispeak_index + 1
  Else
    LUT[i] = thispeak_index
  If i == nextpeak and i < max(levels)
    thispeak = nextpeak; thispeak_index += 1
    nextpeak = nlevels[thispeak_index + 1]
For x,y in image
  f(x,y) = LUT[f(x,y)]
```

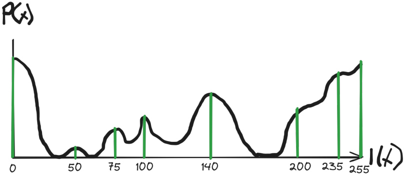
Adapting multi-channel quantization

For single-channel: subdivide or reallocate value space $\rightarrow$ quantization

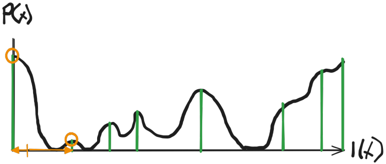
For multi-channel: value spaces of trichromatic images are *connected*

$\rightarrow$ plain subdivision insufficient

Colour palette



0	→ 0
50	→ 1
75	→ 2
100	→ 3
140	→ 4
200	→ 5
235	→ 6
255	→ 7



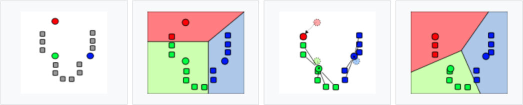
R	G	B	
0	0	0	→ 0
155	155	155	→ 1
148	92	0	→ 2
196	133	31	→ 3
250	186	84	→ 4
159	255	247	→ 5
172	208	205	→ 6
0	45	147	→ 7

Quantization using k-means

Reduce # channel levels in an image to k values via k-means:

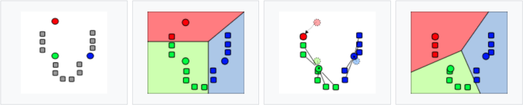
Quantization using k-means

Reduce # channel levels in an image to k values via k-means:



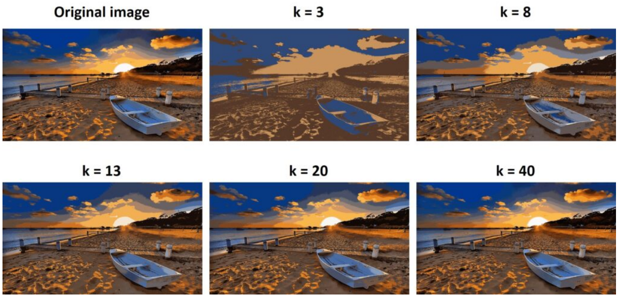
Quantization using k-means

Reduce # channel levels in an image to k values via k-means:



```
Initialize mean[1..k] with random (different) values
For a given number of iterations:
  set cluster[1..k] to empty sets (clusters)
  For each item x do
    find i such that distance(x,mean[i]) is minimal
    Add x to cluster[i]
  End for
  For each cluster[i] do
    mean[i] = mean of cluster[i]
  End for
End for
```

Adapting multi-channel quantization



Adapting image resolution

3 key procedures:

1. Interpolation
2. Downsampling
3. Upsampling

Image interpolation - nearest

What is the image value S_x in an **arbitrary** point x (with $x \in \mathcal{R}$) ?

- *nearest neighbor interpolation:*

  value of the *closest* pixel

Image interpolation - nearest

What is the image value S_x in an **arbitrary** point x (with $x \in \mathcal{R}$) ?

- *nearest neighbor interpolation:*

  $I(x=0.5, y=1) = I(\ int(x), int(y)\ ) = I(x=0, y=1)$

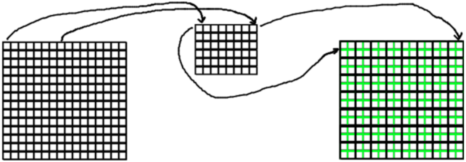


Image interpolation - nearest

What is the image value S_x in an **arbitrary** point x (with $x \in \mathcal{R}$) ?

- *nearest neighbor interpolation:*

$x_0 = int(x);\ y_0 = int(y);\ x_1 = x_0 + 1;\ y_1 = y_0 + 1$

$I(x{=}0.5,\ y{=}1) = \boxed{I(x_0,\ y_0)}\,\boxed{w_{11}} + I(x_0,\ y_1)\ w_{12} + I(x_1,\ y_0)\ w_{21} + I(x_1,\ y_1)\ w_{22}$

$\boxed{w_{11} = (1.0 - \alpha)(1.0 - ß);}\quad w_{12} = (1.0 - \alpha)\ ß$

$w_{21} = \alpha\ (1.0 - ß);\quad w_{22} = \alpha\ ß$

For 'NN': α, ß = 0

$I(\ x{=}0.5,\ y{=}1\ ) = \boxed{I(x_0,\ y_0)}\,\boxed{w_{11}} =$
$\boxed{I(0,\ 1)} * \boxed{(1 * 1)}$

Image interpolation - linear

What is the image value S_x in an **arbitrary** point x (with $x \in \mathcal{R}$) ?

- nearest neighbor interpolation
- *bilinear interpolation:*

  weighted average of 4 neighboring pixels

Image interpolation - linear

What is the image value S_x in an **arbitrary** point x (with $x \in \mathcal{R}$) ?

- nearest neighbor interpolation
- *bilinear interpolation:*

$x_0 = int(x); y_0 = int(y); x_1 = x_0 + 1; y_1 = y_0 + 1$
$I(x=0.5, y=1) = I(x_0, y_0)\, w_{11} + I(x_0, y_1)\, w_{12} + I(x_1, y_0)\, w_{21} + I(x_1, y_1)\, w_{22}$
$w_{11} = (1.0 - \alpha)(1.0 - ß); \quad w_{12} = (1.0 - \alpha)\, ß$
$w_{21} = \alpha\, (1.0 - ß); \quad w_{22} = \alpha\, ß$

For:
$\alpha = x - int(x)$, $ß = y - int(y)$
$I(x=0.5, y=1) =$
$I(0, 1) * (0.5 * 1) + I(0, 2) * (0.5 * 0) +$
$I(1, 0) * (0.5 * 1) + I(1, 2) * (0.5 * 0)$

Image interpolation - linear

What is the image value S_x in an **arbitrary** point x (with $x \in \mathcal{R}$) ?

- nearest neighbor interpolation
- *bilinear interpolation:*

$$x_0 = int(x);\ y_0 = int(y);\ x_1 = x_0 + 1;\ y_1 = y_0 + 1$$
$$I(x{=}0.5, y{=}1) = a_{00} + a_{10}\,\alpha + a_{01}\,ß + a_{11}\,\alpha\,ß$$
$$a_{00} = I(x_0, y_0); \quad a_{10} = I(x_1, y_0) - I(x_0, y_0)$$
$$a_{10} = I(x_0, y_1) - I(x_0, y_0)$$
$$a_{11} = I(x_1, y_1) - I(x_1, y_0) - I(x_0, y_1) + I(x_0, y_0)$$

$$\alpha = x - int(x),\ ß = y - int(y)$$
$$I(\ x{=}0.5, y{=}1\ ) =$$
$$I(0,1) + (I(1,1) - I(0,1)) * 0.5 + (I(0,2) - I(0,1)) * 0$$
$$+ \quad (I(1,2) - I(1,1) - I(0,2) + I(0,1)) * 0.5 * 0$$

Image interpolation - cubic

What is the image value S_x in an **arbitrary** point x (with $x \in \mathcal{R}$) ?

- nearest neighbor interpolation
- bilinear interpolation
- *higher-order interpolation:*
  (Example: Bicubic interpolation[1])
  - idea behind bilinear polynom:
    – span a squared patch over pixels
    – pixel levels $f(x, y)$ represent patch height
    – *bi*-affix $\rightarrow$ 2-dim. interpolation
  - bi-*cubic*: 2D interpolation of a cubic (i.e. 3rd-order polynomial) curve over pixels
  - result: more polynomial constituents
    bilinear: 4; bicubic: 16
  - advantage: smoother gradient across pixels

[1] following Wikipedia - `https://en.wikipedia.org/wiki/Bicubic_interpolation`

Image interpolation - cubic

What is the image value S_x in an **arbitrary** point x (with $x \in \mathcal{R}$) ?

- *higher-order interpolation:*
  (Example: Bicubic interpolation[2])

The interpolation problem consists of determining the 16 coefficients a_{ij}. Matching $p(x,y)$ with the function values yields four equations:

1. $f(0,0) = p(0,0) = a_{00}$,
2. $f(1,0) = p(1,0) = a_{00} + a_{10} + a_{20} + a_{30}$,
3. $f(0,1) = p(0,1) = a_{00} + a_{01} + a_{02} + a_{03}$,
4. $f(1,1) = p(1,1) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij}$.

Likewise, eight equations for the derivatives in the x and the y directions:

1. $f_x(0,0) = p_x(0,0) = a_{10}$,
2. $f_x(1,0) = p_x(1,0) = a_{10} + 2a_{20} + 3a_{30}$,
3. $f_x(0,1) = p_x(0,1) = a_{10} + a_{11} + a_{12} + a_{13}$,
4. $f_x(1,1) = p_x(1,1) = \sum_{i=1}^{3} \sum_{j=0}^{3} a_{ij} i$,
5. $f_y(0,0) = p_y(0,0) = a_{01}$,
6. $f_y(1,0) = p_y(1,0) = a_{01} + a_{11} + a_{21} + a_{31}$,
7. $f_y(0,1) = p_y(0,1) = a_{01} + 2a_{02} + 3a_{03}$,
8. $f_y(1,1) = p_y(1,1) = \sum_{i=0}^{3} \sum_{j=1}^{3} a_{ij} j$.

And four equations for the xy mixed partial derivative:

1. $f_{xy}(0,0) = p_{xy}(0,0) = a_{11}$,
2. $f_{xy}(1,0) = p_{xy}(1,0) = a_{11} + 2a_{21} + 3a_{31}$,
3. $f_{xy}(0,1) = p_{xy}(0,1) = a_{11} + 2a_{12} + 3a_{13}$,
4. $f_{xy}(1,1) = p_{xy}(1,1) = \sum_{i=1}^{3} \sum_{j=1}^{3} a_{ij} ij$.

---

[2] following Wikipedia - `https://en.wikipedia.org/wiki/Bicubic_interpolation`

Image interpolation - cubic

What is the image value S_x in an **arbitrary** point x (with $x \in \mathcal{R}$) ?

- *higher-order interpolation:*
  (Example: Bicubic interpolation[2])

There can be another concise matrix form for 16 coefficients:

$$\begin{bmatrix} f(0,0) & f(0,1) & f_y(0,0) & f_y(0,1) \\ f(1,0) & f(1,1) & f_y(1,0) & f_y(1,1) \\ f_x(0,0) & f_x(0,1) & f_{xy}(0,0) & f_{xy}(0,1) \\ f_x(1,0) & f_x(1,1) & f_{xy}(1,0) & f_{xy}(1,1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 3 \end{bmatrix},$$

or

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) & f_y(0,0) & f_y(0,1) \\ f(1,0) & f(1,1) & f_y(1,0) & f_y(1,1) \\ f_x(0,0) & f_x(0,1) & f_{xy}(0,0) & f_{xy}(0,1) \\ f_x(1,0) & f_x(1,1) & f_{xy}(1,0) & f_{xy}(1,1) \end{bmatrix} \begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix},$$

where

$$p(x,y) = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} 1 \\ y \\ y^2 \\ y^3 \end{bmatrix}.$$

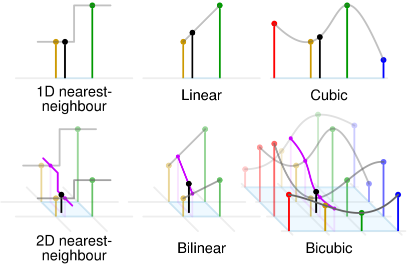
[2] following Wikipedia - `https://en.wikipedia.org/wiki/Bicubic_interpolation`

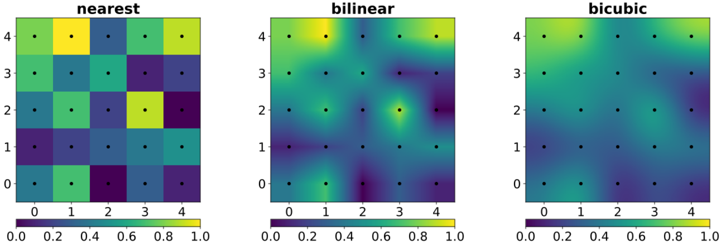
Image interpolation

What is the image value S_x in an **arbitrary** point x (with $x \in \mathcal{R}$) ?

- *nearest neighbor interpolation:*
  value of the *closest* pixel
- *bilinear interpolation:*
  weighted average of 4 neighboring pixels
- *higher-order interpolation:*
  more accurate, more computation because more neighboring pixels are used.

Comparison interpolation techniques

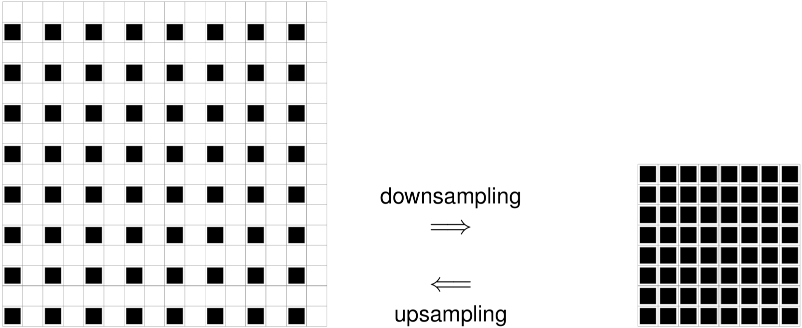




[3]

[3] image sources: Wikipedia (CC BY-SA) & Matplotlib

Downsampling and upsampling



Downsampling and upsampling (zooming) by a factor of 2 in each dimension

Zooming by pixel replication

a	b	c	d
e	f	g	f
i	j	k	l
m	n	o	p

Input image

a	a	b	b	c	c	d	d
a	a	b	b	c	c	d	d
e	e	f	f	g	g	f	f
e	e	f	f	g	g	f	f
i	i	j	j	k	k	l	l
i	i	j	j	k	k	l	l
m	m	n	n	o	o	p	p
m	m	n	n	o	o	p	p

Zoomed image

Shrink and zoom

Image (3692 × 2812 pixels) shrunk to 72 dpi (row 1) or 150 dpi (row 2) and zoomed back to original size. nearest neighbor (column 1), bilinear (column 2), and bicubic (column 3) interpolation.

That's it for this week!