Image Representation

SS 2008 – Image Processing

Multimedia Computing, Universität Augsburg
Rainer.Lienhart@informatik.uni-augsburg.de
www.multimedia-computing.{de,org}
Reference


Chapter on Image Representation
(chapter 2 in 4th edition)

Figures are mostly taken from that book
Image

• Definition of Image
  – (phys.) := spatial distribution of irradiance at a plane
  – (math.) := continuous function of two spatial variables
    \[ I(x_1, x_2) = I(x) \]

• Computers generally cannot handle continuous images → 2D array of digital numbers

• Pixel
  – := a point on the 2D grid (abbr. of ‘picture element’)
  – Represents a rectangular elementary region (‘cell’)
  – Its value = avg. irradiance in the cell
**Pixel and Voxel**

- In algos: \( I(x,y) \)
- In C/C++: \( I[y][x] \)

- In algos: \( I(x,y,z) \)
- In C/C++: \( I[z][y][x] \)

**Figure 2.1:** Representation of digital images by arrays of discrete points on a rectangular grid: a 2-D image, b 3-D image.
Video

• Definition of Video
  – (phys.) := spatio-temporal distribution of irradiance at a plane
  – (math.) := continuous function of one temporal and two spatial variables
    \[ I(x_1, x_2, t) = I(x, t) \]

• Computers generally cannot handle continuous videos \(\rightarrow\) 3D array of digital numbers

• Pixel
  – := a point on the 3D grid (abbr. of ‘picture element’)
  – Represents a rectangular elementary region (‘cell’)
  – Its value = avg. irradiance in the cell
How Many Pixels Are Sufficient?

• There is no general answer

• For a given task the pixel size should be smaller than the finest scales of the object that we want to study

• More is not always better

*Figure 2.2: Digital images consist of pixels. On a square grid, each pixel represents a square region of the image. The figure shows the same image with a $3 \times 4$, $b 12 \times 16$, $c 48 \times 64$, and $d 192 \times 256$ pixels. If the image contains sufficient pixels, it appears to be continuous (exercise 2.1).*
Alternative Sampling Grids

Figure 2.3: The three possible regular grids in 2-D: a triangular grid, b square grid, c hexagonal grid.
Generally used to define connected regions

**Figure 2.4:** Neighborhoods on a rectangular grid: **a** 4-neighborhood and **b** 8-neighborhood. **c** The black region counts as one object (connected region) in an 8-neighborhood but as two objects in a 4-neighborhood.
Two choices:
(a) Have at least one common edge
(b) Have at least one common corner
Figure 2.5: The three types of neighborhoods on a 3-D cubic grid.  

a 6-neighborhood: voxels with joint faces; b 18-neighborhood: voxels with joint edges; c 26-neighborhood: voxels with joint corners.
Distance Measures on Grid

General:
Given vectors $x$ and $y$

$$L_p(x, y) = \left[ \sum_{i=1}^{n} (x_i - y_i)^p \right]^{1/p}$$

Euclidean distance:
$$L_2(x, y) = \left[ \sum_{i=1}^{n} (x_i - y_i)^2 \right]^{1/2}$$

City block distance:
$$L_1(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

Chess board distance:
$$L_\infty(x, y) = \max_{1 \leq i \leq n} (|x_i - y_i|)$$
Figure 2.6: A discrete line is only well-defined in the directions of axes and diagonals. In all other directions, a line appears as a staircase-like jagged pixel sequence.

Only alpha mod 45 == 0 lines well-defined

Lines at all other angles are jagged, staircase-like
Quantization := mapping the irradiance at the image plane onto a limited number of $Q$ discrete gray values

Commonly $Q=256$ (1 byte = 8 bits) for grayscale images

However, dynamic is still a problem (e.g., for movies) → needs more bits

Exercise 2.3
#include <iostream>
#include <cv.h>
#include <highgui.h>
#include <math.h>

typedef unsigned char Ipp8u;

int main(int argc, char** argv)
{
    std::cout << argv[0] << " <filename of color image> <# of significant bits>" << std::endl;

    if (argc != 3) return 1;

    // Create source image
    char* imageFileName = argv[1];
    int noOfSigBits = atoi(argv[2]);
    Ipp8u bitmask=0xff; bitmask <<= (8 - noOfSigBits);
    Ipp8u bucketSize=1; bucketSize <<= (8 - noOfSigBits);

    IplImage* srcGray = cvLoadImage(imageFileName, 0 /* force grayscale */);
    IplImage* srcColor = cvLoadImage(imageFileName);
// Show source images
cvNamedWindow("Source Grayscale Image", CV_WINDOW_AUTOSIZE);
cvNamedWindow("Source Color Image", CV_WINDOW_AUTOSIZE);
cvShowImage("Source Grayscale Image", srcGray);
cvShowImage("Source Color Image", srcColor);

// Pause until any key is hit
cvWaitKey(0);

cvAndS(srcGray, cvScalar(bitmask), srcGray);
cvAddS(srcGray, cvScalar(bucketSize/2), srcGray);
cvAndS(srcColor, cvScalar(bitmask,bitmask,bitmask), srcColor);
cvAddS(srcColor, cvScalar(bucketSize/2,bucketSize/2,bucketSize/2), srcColor);

// Show result images
cvShowImage("Source Grayscale Image", srcGray);
cvShowImage("Source Color Image", srcColor);

cvWaitKey(0);
return 0;
Signed Images

• Normally brightness (irradiance) is considered to be a positive quantity
  → suggests unsigned pixel representation

• BUT: Problem with arithmetic operations on images
  → suggests signed pixel representation

Conversion operations for 8u ↔ 8s images:

\[
q' = (q - 128) \mod 256 \quad 0 \leq q < 256 \\
q = (q' + 128) \mod 256 \quad -128 \leq q < 127
\]

Abbreviations:

- 8u, 8s
- 16u, 16s
- 32u, 32s
- 32f, 64f
- 8u_C1, 8u_C3, 8u_C4
Human visual system shows a logarithmic response (and not a linear one)

→ 256 values cannot capture this dynamic range!

→ Linear sensor; either the bright parts are overexposed or the dark parts are underexposed

→ Possible solution:

\[ f(x,y) = l(x,y)^\gamma, \, \gamma = 0.4 \]
Spatial Representation (1)

Image composed of individual pixels

- Basis image \( m,n \mathbf{P}(x,y) = \text{image with “1” at row } m \text{ and column } n \text{ and 0 everywhere else:} \)

\[
m,n \mathbf{P}(x, y) = \begin{cases} 
1 & \text{if } (m == y) \& (n == x) \\
0 & \text{otherwise}
\end{cases}
\]

- Arbitrary image \( I(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(x, y) \cdot m,n \mathbf{P}(x, y) \)

- \( m,n \mathbf{P}(x, y) \) forms an orthonormal basis (why?)

“A basis is a minimum set of vectors that, when combined, can address every vector in a given space. More precisely, a basis of a vector space \( V \) is usually defined as a subset \( B \) of \( V \) if it satisfies one of the four equivalent conditions:

- \( B \) is both a set of linearly independent vectors and a generating set of \( V \).
- \( B \) is a minimal generating set of \( V \), i.e. it is a generating set but no proper subset of \( B \) is.
- \( B \) is a maximal set of linearly independent vectors, i.e. it is a linearly independent set but no proper superset is.
- every vector in \( V \) can be expressed as a linear combination of vectors in \( B \) in a unique way.”

(from wikipedia.org)
Other Representations

• Imagine $M \times N$ image as a point in the $M \times N$ dimensional vector space

• If we change the coordinate system, the image remains the same, but representation changes
  – Look at same image from a different perspective
  – All representations are complete and equivalent
  – All representations can be losslessly converted into each other
1D Discrete Fourier Transform (DFT):
Maps an ordered $N$-tuple $g$ of complex numbers
onto another complex vector $\hat{g}$ of same dimension

$$g = [g_0, g_1, \ldots, g_{N-1}]^T$$

$$\hat{g}_v = \frac{1}{N} \sum_{n=0}^{N-1} g_n e^{-\frac{2\pi i n v}{N}}, 0 \leq v < N$$

Kernel of DFT

Back transform:

$$g_n = \sum_{v=0}^{N-1} \hat{g}_v e^{\frac{2\pi i n v}{N}}, 0 \leq n < N$$

$\exp(xi) = \cos(x) + is\sin(x)$
Written as scalar product

\[ \hat{g}_v = \langle \mathbf{g}, \mathbf{b}_v^* \rangle, \quad 0 \leq v \leq N \]
\[ g_v = \langle \hat{g}, \mathbf{b}_v \rangle, \quad 0 \leq v \leq N \]

where

\[ \mathbf{b}_v = \left( 1, W_N^{1v}, W_N^{2v}, \ldots, W_N^{(N-1)v} \right)^T, \quad W_N = e^{\frac{2\pi i}{N}} \]
\[ \mathbf{b}_v^* = \frac{1}{N} \left( 1, W_N^{-1v}, W_N^{-2v}, \ldots, W_N^{-(N-1)v} \right)^T = \mathbf{b}_{-v} \]

The basis vectors are (ignoring factor 1/N) orthonormal, i.e.,

\[ \frac{1}{N} \langle \mathbf{b}_v, \mathbf{b}_{v'} \rangle = \delta_{v-v'} = \begin{cases} 1 & \text{if } v = v' \\ 0 & \text{otherwise} \end{cases} \]

→ Each vector can be expressed as a linear combination of the basis vectors of the Fourier space.
First 9 Basis Functions

Example N=16

Index denotes how often the wavelength fits into the interval [0,N-1]

*Figure 2.12: The first 9 basis functions of the DFT for N = 16; a real part (cosine function), b imaginary part (sine function).*
**Table 2.1:** Comparison of the continuous Fourier transform (FT), the Fourier series (FS), and the discrete Fourier transform (DFT) in one dimension.

<table>
<thead>
<tr>
<th></th>
<th>Forward transform</th>
<th>Backward transform</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FT</strong></td>
<td>$\hat{g}(k) = \frac{1}{2\pi} \int_{-\infty}^{\infty} g(x) \exp(-ikx) , dx$</td>
<td>$g(x) = \int_{-\infty}^{\infty} \hat{g}(k) \exp(ikx) , dk$</td>
</tr>
<tr>
<td><strong>FS</strong></td>
<td>$\hat{g}<em>u = \frac{1}{2\pi} \int</em>{0}^{2\pi} g(x) \exp(-2\pi ux) , dx$</td>
<td>$g(x) = \sum_{u=-\infty}^{\infty} \hat{g}_u \exp(2\pi ux)$</td>
</tr>
<tr>
<td><strong>DFT</strong></td>
<td>$\hat{g}<em>u = \frac{1}{M} \sum</em>{m=0}^{M-1} g_m \exp\left(-\frac{2\pi i mu}{M}\right)$</td>
<td>$g_m = \sum_{u=0}^{M-1} \hat{g}_u \exp\left(\frac{2\pi i mu}{M}\right)$</td>
</tr>
</tbody>
</table>
2D DFT

Maps complex $M \times N$ matrix onto complex matrix of same size:

$$
\hat{G}_{u,v} = \frac{1}{MN} \sum_{m=0}^{M} \sum_{n=0}^{N} G_{m,n} e^{\frac{-2\pi i u}{N}} e^{\frac{-2\pi i v}{M}} = \langle G, B_{-u,-v} \rangle = \langle G, B_{u,v}^* \rangle
$$

$$
= \frac{1}{MN} \sum_{m=0}^{M} \left( \sum_{n=0}^{N} G_{m,n} W_N^{-nv} \right) W_M^{-mu}
$$

Inverse 2D DFT:

$$
G_{m,n} = \sum_{u=0}^{M} \sum_{v=0}^{N} \hat{G}_{u,v} W_N^{nv} W_M^{mu} = MN \langle \hat{G}, B_{m,n} \rangle
$$

Basis matrix (separable!!):

$$
B_{u,v} = \frac{1}{\sqrt{MN}} b_u \otimes b_v^T = \frac{1}{\sqrt{MN}} \begin{bmatrix} 1 \\ W_M^{1u} \\ W_M^{2u} \\ \vdots \\ W_M^{(M-1)u} \end{bmatrix} \begin{bmatrix} 1, W_N^{1v}, W_N^{2v}, \ldots, W_N^{(N-1)v} \end{bmatrix}
$$

Outer product
Figure 2.10: An image can be thought to be composed of basis images in which only one pixel is unequal to zero.

Exercise 2.7

Figure 2.11: The first 56 periodic patterns, the basis images of the Fourier transform, from which the image in Fig. 2.10 is composed
**DFT Properties (1)**

*Table 2.2: Summary of the properties of the two-dimensional DFT. G and H are complex-valued \( M \times N \) matrices, \( \hat{G} \) and \( \hat{H} \) their Fourier transforms, and \( a \) and \( b \) complex-valued constants.*

<table>
<thead>
<tr>
<th>Property</th>
<th>Space domain</th>
<th>Wave number domain</th>
</tr>
</thead>
</table>
| Mean                   | \[
\frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} G_{mn} \]                   | \( \hat{G}_{0,0} \)           |
| Linearity              | \( aG + bH \)                                                              | \( a\hat{G} + b\hat{H} \)     |
| Shifting               | \( G_{m-m',n-n'} \)                                                        | \( W_M^{-m'u} W_N^{-n'v} \hat{G}_{uv} \) |
| Modulation             | \( W_M^{u'} W_N^{v'} G_{m,n} \)                                            | \( \hat{G}_{u-u',v-v'} \)      |
| Finite differences     | \[
\frac{(G_{m+1,n} - G_{m-1,n})}{2}, \quad \frac{(G_{m,n+1} - G_{m,n-1})}{2}
\] | \( i \sin(2\pi u/M) \hat{G}_{uv} \) |
| Spatial stretching     | \( G_{pm,Qn} \)                                                            | \( \hat{G}_{uv} / (PQ) \)     |
| Frequency stretching   | \( G_{m,n} \)                                                              | \( \hat{G}_{Pu,Qv} \)         |
Spatial sampling: \( G_{m/P, n/Q} \)

Frequency sampling:
\[
\frac{1}{PQ} \sum_{p=0}^{P-1Q-1} \sum_{q=0}^{Q-1} G_{m+pM/P, n+qN/Q} = \hat{G}_{pu,qv}
\]

- Convolution:
\[
(H \ast G)_{mn} = \frac{1}{MN} \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} H_{m'n'} G_{m-m', n-n'} = \hat{H}_{uv} \hat{G}_{uv}
\]

- Multiplication:
\[
G_{mn} H_{mn}
\]

- Spatial correlation:
\[
(H \ast G)_{mn} = \frac{1}{MN} \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} H_{m'n'} G_{m+m', n+n'} = \hat{H}_{uv} \hat{G}^*_{uv}
\]

- Inner product:
\[
\frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} G_{mn} H^*_{mn} = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \hat{G}_{uv} \hat{H}^*_{uv}
\]

- Norm:
\[
\frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |G_{mn}|^2 = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |\hat{G}_{uv}|^2
\]
### Important FT Pairs

<table>
<thead>
<tr>
<th>Spatial Domain</th>
<th>Freq. Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delta $\delta(x)$</td>
<td>$\text{const.} \frac{1}{2\pi}$</td>
</tr>
<tr>
<td>$\delta \text{comb} \sum_{n=-\infty}^{\infty} \delta(x-n\Delta x)$</td>
<td>$\delta \text{comb} \sum_{u=-\infty}^{\infty} \delta(k-2\pi u/\Delta x)$</td>
</tr>
<tr>
<td>Box $\Pi(x) =$ \begin{cases} 1 &amp; x &lt; \frac{1}{2} \ 0 &amp; x \geq \frac{1}{2} \end{cases}</td>
<td>$\text{sinc} \frac{\sin(k/2)}{k/2}$</td>
</tr>
<tr>
<td>Cosine $\cos(k_0 x)$</td>
<td>$\frac{1}{2} \left[ \delta(k-k_0) + \delta(k+k_0) \right]$</td>
</tr>
<tr>
<td>Sine $\sin(k_0 x)$</td>
<td>$\frac{i}{2} \left[ \delta(k-k_0) - \delta(k+k_0) \right]$</td>
</tr>
<tr>
<td>Gauss $\frac{1}{\sigma \sqrt{2\pi}} e^{-x^2/2\sigma^2}$</td>
<td>Gauss $e^{-k^2 \sigma^2/2}$</td>
</tr>
<tr>
<td>Sign $\text{sgn}(x) =$ \begin{cases} 1 &amp; x \geq 0 \ -1 &amp; x &lt; 0 \end{cases}</td>
<td>$-\frac{2i}{k}$</td>
</tr>
<tr>
<td>Step $H(x) =$ \begin{cases} 1 &amp; x \geq 0 \ 0 &amp; x &lt; 0 \end{cases}</td>
<td>$\frac{1}{2} \delta(k) - \frac{i}{k}$</td>
</tr>
</tbody>
</table>
Periodicity

- Periodicity in the DFT kernel:
  \[ e^{\frac{2\pi i (m+kM)}{M}} = e^{\frac{2\pi i m}{M}}, \forall k \in \mathbb{Z} \]

⇒ Same periodicity for DFT/Reverse DFT

- Freq. domain: \( \hat{G}_{u+kM,v+lN} = \hat{G}_{u,v}, \forall k,l \in \mathbb{Z} \)
- Spatial domain: \( G_{m+kM,n+lN} = G_{m,n}, \forall k,l \in \mathbb{Z} \)
**Table 2.5: Symmetry properties of the continuous Fourier transform.**

<table>
<thead>
<tr>
<th>Space domain</th>
<th>Fourier domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Even, odd ( g(-x) = \pm g(x) )</td>
<td>Even, odd ( \hat{g}(-k) = \pm \hat{g}(k) )</td>
</tr>
<tr>
<td>Real ( g(x) = g^*(x) )</td>
<td>Hermitian ( \hat{g}(-k) = \hat{g}^*(k) )</td>
</tr>
<tr>
<td>Imaginary ( g(x) = -g^*(x) )</td>
<td>Antihermitian ( \hat{g}(-k) = -\hat{g}^*(k) )</td>
</tr>
<tr>
<td>Rotational symmetric ( g(</td>
<td>x</td>
</tr>
</tbody>
</table>

- **Definition of discrete symmetry:**
  
  \[
  G_{mn} = \pm G_{M-m,N-n}
  \]

  \[
  G_{mn} = G_{mn}^* \overset{d_F}{\leftrightarrow} \hat{G}_{M-u,N-v} = \hat{G}_{uv}
  \]

  Used to reduce storage and complexity requirements

  Symmetry center: \((M/2,N/2)\)
Power spectrum

log scale $|\hat{G}_{u,v}|^2$

(0,0)

256 values do not suffice,
needs float values
Phase is more important than amplitude.

If we do not know the phase of its Fourier transform, we know neither what the object looks like nor where it is located!

A shift of an object in the space domain leads to a shift of the phase in the wave number domain only. The amplitude is not changed.

**Figure 2.17**: Illustration of the importance of phase and amplitude in Fourier space for the image content: a, b two original images; c composite image using the phase from image b and the amplitude from image a; d composite image using the phase from image a and the amplitude from image b.
Phase contains the important image content.