Outline

- Multi-Dimensional Fourier Transform
- Sampling and Quantization
- Decimation and Interpolation
- Relationships Between Pixels
- Distance Measures
- Linear vs Non-linear Operations
- Image Enhancement in the Spatial Domain
  - Gray level transformations
  - Histogram processing
  - Enhancement with Arithmetic Operations

Multi-dimensional Fourier Transform

- The multi-dimensional Fourier Transform (FT) is a straightforward generalization of the ordinary FT

$$
\begin{aligned}
F(u_1, u_2, \ldots, u_n) &= \int f(x_1, x_2, \ldots, x_n) \\
&\quad \times \exp[-j2\pi(u_1 x_1 + u_2 x_2 + \ldots u_n x_n)] \\
&\quad \times \mathrm{d}x_1 \mathrm{d}x_2 \ldots \mathrm{d}x_n
\end{aligned}
$$

- The multi-dimensional inverse FT is

$$
\begin{aligned}
f(x_1, x_2, \ldots, x_n) &= \int F(u_1, u_2, \ldots, u_n) \\
&\quad \times \exp[j2\pi(u_1 x_1 + u_2 x_2 + \ldots u_n x_n)] \\
&\quad \times \mathrm{d}u_1 \mathrm{d}u_2 \ldots \mathrm{d}u_n
\end{aligned}
$$

- The FT gives us the information about the frequency content of the image (as in 1-D FT)

Multi-dimensional Fourier Transform (Cont.)

- The dicrete versions can similarly be obtained

$$
\begin{aligned}
F[u_1, u_2, \ldots, u_n] &= \tfrac{1}{M_1 M_2 \ldots M_n} \sum f[x_1, x_2, \ldots, x_n] \\
&\quad \times \exp[-j2\pi(u_1 x_1/M_1 + u_2 x_2/M_2 + \ldots u_n x_n/M_n)]
\end{aligned}
$$

- The multi-dimensional inverse FT is

$$
\begin{aligned}
f[x_1, x_2, \ldots, x_n] &= \sum F[u_1, u_2, \ldots, u_n] \\
&\quad \times \exp[j2\pi(u_1 x_1/M_1 + u_2 x_2/M_2 + \ldots u_n x_n/M_n)]
\end{aligned}
$$

- The multi-dimensional image can have different pixels along different dimensions
- Fast algorithms (FFT) can be applied to obtain multi-dimensional DFT's

Shifting the FT

- Now let us consider 2-D images of size $M \times N$ with spatial coordinates x, y and frequency variables u, v
- Remember most signals (also images) that we encounter have most of its energy in the low-pass region
- Then the FT will have high values around (0,0), but we would like to see this content in the center area of the spectrum
- We can do that by first multiplying the image by $(-1)^{x+y}$ and then take the FT

Shifting the FT

- We have

$$
\begin{aligned}
F(u,v) &= \frac{1}{MN} \sum \sum f(x,y)(-1)^{x+y} \exp[-j2\pi(ux/M + vy/N) \\
&= \sum \sum f(x,y) \exp[j\pi(x+y)] \exp[-j2\pi(ux/M + vy/N)] \\
&= \sum \sum f(x,y) \exp[-j2\pi((u - M/2)x/M + (v - N/2)y/N)]
\end{aligned}
$$

- Then the frequency content will appear at the center of the frequency rectangle
- Just for visualisation purposes, no mathematical difference

Sampling

- Sampling and quantization are two steps required to obtain a digital image
- A two-D continuous image is then transformed into a 2-D matrix after sampling
- Sampling results in

$$f[x, y] = f(x', y')g(x', y')$$

where (x', y') are the continuous variables and (x, y) are discrete variables, f the sampled image and g the function representing sampling grid

Sampling (Cont.)

- An example of sampling grid would be a uniform one

$$g(x', y') = \begin{cases} 1, x' = nN \ \text{ and } \ y' = mN \\ 0, \text{otherwise} \end{cases}$$

  with n, m being integers and N sampling periods along two dimensions

- We often use non-uniform sampling in several applications depending on how the image is acquired. One example where non-uniform sampling would be an image where we know certain parts have much more details (requiring denser sampling) and other parts are relatively smooth (requiring more sparse sampling)

Quantization

- Quantization discretizes the amplitude of the samples obtained
- using b bits result in 2^b gray levels usually used as $0, 1, \ldots, 2^b - 1$
- Number of bits to be used certainly depends on the application, 8 and 16 are common choices
- Increased number of samples and increased number of gray levels certainly increase the overall quality of the image. However, there is a saturation point beyond which no visible improvement is observed
- The required number of pixels and number of gray levels also depends on the image content
- For an image with high frequency content (a lot of detail) we would require a high number of samples
- For an image with low contrast (closer gray levels) we would require a high number of gray levels (not to loose information)

Decimation and Interpolation

- Let us assume we want to change the size of a digital image
- Increasing the size is called interpolation
- We can overlay the larger grid (the larger image size) on the original image
- Then there will be sampling points that has no available value
- How you select this missing value determines your interpolation method
- Nearest neighbourhood: select the pixel value that is closest
- Bi-linear: choose a value that lies on the line connecting the two closest pixels, usually results in better quality

Relationships Between Pixels

- Neighborhood: we can have different definitions of neighborhood
  - 4-neighbors: (x+1,y),(x,y+1),(x-1,y),(x,y-1)
  - 8-neighbors: (x+1,y),(x,y+1),(x-1,y),(x,y-1), (x+1,y+1),(x-1,y-1), (x-1,y+1),(x+1,y-1)
- Connectivity: neighborhood + some property (e.g. similar gray level)
- 4-adjacency: Two pixels have similar property and 4-neighbors
- 8-adjacency: Two pixels have similar property and 8-neighbors
- m-adjacency: Two pixels have similar property and either are 4-neighbors or 8-neighbors with no common 4-neighborhood
- A path is a set of consecutive pixels that are adjacent, it is called a closed path if the starting pixel and ending pixel are the same
- A region is a group of pixels that are connected
- A boundary is the group of pixels that are a part of a region but has neighbors that are not a part of the region

Distance Measures

- A function of two pixels is called a distance (or a metric) if
  - $D(p, q) \geq 0$, and equal to zero if and only if $p = q$
  - $D(p, q) = D(q, p)$
  - $D(p, z) \leq D(p, q) + D(q, z)$
- E.g. euclidian distance: $D(p, q) = [(p_x - q_x)^2 + (p_y - q_y)^2]^{1/2}$

Linear vs. Nonlinear Operations

- An operation is linear if and only if

$$H(af(x,y)+bg(x,y)) = aH(f(x,y)) + bH(g(x,y))$$

- There are very common non-linear operations in image processing in contrast to signal processing
- E.g. median filtering: the median of the values

$$\text{Median}\{f = 0,1,2\} = 1 \quad \text{Median}\{g = 3,7,5\} = 5$$

$$\text{Median}\{f+g = 3,8,7\} = 7$$

which is not 6

- E.g. max: the maximum value

$$\text{Max}\{f = 0,1,2,3\} = 3 \quad \text{Max}\{g = 3,2,1,0\} = 3$$

$$\text{Max}\{f+g = 3,3,3,3\} = 3$$

which is not 6

Image Enhancement in the Spatial Domain

- Image enhancement is the task of improving the quality of an image for a specific task
- Depending on the application and image content the optimum method varies considerably, no universel “good” image enhancement algorithm
- Quality of image enhancement algorithm is determined by again the specific task
- Measuring the quality of the resulting images is usually a very difficult problem, and results are mostly subjective
- E.g. in photography, the quality of the image produced can be determined by visual evaluation
- E.g. in tumor detection, the quality of the produced image is determined by the increased true detection and decreased fall detection and misses

Image Enhancement in the Spatial Domain: Gray Level Transformations

- Now we consider enhancing an image using operations in the spatial domain

  $$g(x, y) = T[f(x, y)]$$

  where $g(x, y)$ denotes the output (enhanced image), $f(x, y)$ the original image and T the spatial transformation

- Most of the time the transformation is performed using a neighborhood of the pixel, also called masks, filters, windows, templates, kernels

- Now let us have a look at some commonly known gray level transformations

Basic Gray Level Transformations

- Image negatives:

$$s = L - 1 - r$$

  where r is the original pixel value s the processed, and $L - 1$ is the maximum gray level in the image

- Useful in stressing bright spots in dark backgrounds
- Log transformations:

$$s = c \log(1 + r)$$

- We have 1 to avoid negative values
- Compresses high values, expands small values, changes dynamic range
- Power transformations

$$s = cr^{\gamma}$$

- Compresses small values, expands large values, changed dynamic range

Basic Gray Level Transformations: Thresholding

- Thresholding is basically the task of binarizing the image

$$r = \begin{cases} 0, & s < T \\ 1, & s \geq T \end{cases}$$

- There are several thresholding algorithms to select the value of T
- The optimum T really depends on the application and image content
- Widely used e.g. in character recognition

Gray level transformations

- Sometimes they are used for improving image visibility, e.g. contrast enhancement
- Sometimes they are necessary for “correction”
- The imaging itself distorts the real object and we undo this effect by using log or power transformations
- E.g. Positron emission tomography, the data is distorted by the decay of radioactive tracer (an exponential), we need to correct this

Piecewise processing

- All these transformations can be performed globally (same transformation for the whole image), or piecewise (varying transformation for different gray levels)

Histogram Processing

- Definition of an histogram:

$$h(r_k) = n_k$$

where r_k is a bin of gray levels (e.g. values between 0 and 10), and n_k is the number of pixels with gray level values in that particular bin

- It is related to the probability of occurence of gray levels
- We usually normalize histograms resulting in total area under the histogram curve equal to unity

$$h(r_k) = n_k/N$$

where N is the total number of pixels

Histogram Processing (Cont.)

- Low contrast images have narrow histograms, and high contrast images have wider histograms
- Human vision favors high contrast images, evaluate them as high quality images
- There are methods to process histograms so that the resulting image has a wide histogram. One such operation is called histogram equalization

Histogram Equalization

- For the development let us consider continuous images with gray values $0 \leq r \leq 1$
- Let T represent the histogram equalizer then
  - T must be single valued so that inverse exists
  - T must be monotonically increasing so that the order of gray values are preserved
  - $T(r)$ must be between 0 and 1, so that the output gray values are valid

Histogram Equalization (Cont.)

- Let $p_s(s)$ denote the PDF of s and $p_r(r)$ pdf of r, then we have from probability theory that

$$p_s(s) = p_r(r)\left|\frac{\mathrm{d}r}{\mathrm{d}s}\right|$$

- Since we want to equalize the histogram we must have $p_s(s) = 1$ when $s < 0 < 1$
- Then

$$1 = p_r(r)\left|\frac{\mathrm{d}r}{\mathrm{d}s}\right|$$

- We obtain

$$\left|\frac{\mathrm{d}r}{\mathrm{d}s}\right| = p_r(r)$$

Histogram Equalization (Cont.)

- If we use the integral of $p_r(r)$ as our transformation

$$T(r) = \int_0^r p_r(r')\mathrm{d}r'$$

then we readily have

$$\frac{\mathrm{d}r}{\mathrm{d}s} = p_r(r)$$

- Then regardless of the content of an image, if we use a gray level transformation of the form

$$T(r) = \int_0^r p_r(r')\mathrm{d}r'$$

the resulting histogram is uniform

Histogram Equalization (Cont.)

- The discrete approximation of the histogram euqlizer is then

$$T(r_k) = \sum_j p_r(r_j)$$

- This transformation does not result in an exactly uniform histogram, but spreads it towards unity

Histogram Matching

- We sometimes prefer a shape other than uniform for the desired histogram
- In this case we specify a desired histogram and develop a method to obtain it
- The development is straightforward in two steps
  - Find the histogram euqalizer of the original image, T_1
  - Find the histogram euqalizer using the desired histogram T_2
  - Then the transformation $T_2^{-1}T_1$ will take the original histogram and produce the desired one
  - Difficulty arises because in practice it is not easy to obtain analytical forms for these inverses

Histogram Matching: Implementation

- Let us define original gray levels r, the equalized gray level s, the desired gray level z
- We have $s = T_1(r)$ and $s = T_2(z)$ as well as $z = T_2^{-1}T_1(r)$
- The difficulty is to computer T_2^{-1}
- Therefore instead of computing z using the inverse we perform the following

$$T_2(z) = T_1(r)$$

or

$$T_2(z) - T_1(r) = 0$$

- We can search for the vallues of z that minimizes $T_2(z) - T_1(r)$ since we can calculate T_1 and T_2

Image Enhancement with Arithmetic Operations: Subtraction

- Simply the subtraction of gray level values pixel by pixel
- Useful in identifying the differences in images
- E.g. you developed a compression algorithm, when two images (original and compressed) are put side by side differences may not be noticable
- When you subtract the image, you will see the differences (artifacts) more clearly
- We need to scale appropriately so that the difference image has all valid gray values

Image Enhancement with Arithmetic Operations: Averaging

- Simply averaging of two images pixel by pixel
- Do not confuse by low-pass filtering (averaging of a neighborhood in the same image)
- Useful for example in noise reduction
- You can obtain several images of the same object and averaga them out to reduce noise
- Another application area: evaluate quality of image registration (aligning) algorithms
- When the aligning quality is good the averaged image will preserve edges etc
- When the aligning quality is poor the averaged image will be blurred

Image Enhancement with Arithmetic Operatins: Multiplication

- Image multiplication is mostly used for masking purposes
- Input image is multiplied by a binary image to mask a certain part of the image