Computer assisted image analysis I

Linus Falk

November 8, 2022

Contents

›

Lecture 2: Image arithmetic

tuesday 01 nov 10:15

1 Image enhancement

Enhance part of an image in some way. Transform an image into a new image. Create a better. restore information, reduce noise. Enhance certain details, edges etc. Just look better. We don't increase the information. Can be performed in spatial domain. Point process : pixel base. Filter works with neighborhood Another way is in frequency domain. The chain of image analysis process. The first part today. Pre processing enhancement. We must start by understanding the problem otherwise it hard to make decisions along the chain/pipeline

- image arithmetic
- intensity transfer function
- histogram and histogram equalization

image arithmetic

we do arithmetic with images. Position in matrix/image operator position in image.

- standard operation + - / *
- logic operator AND OR XOR

Pitfalls could be add and divide could be outside range of 0-255 for example. Need to normalize but needs to be done before we do the operation. otherwise we might have destroyed information. Bit depth important.

Useful way is to truncate the image.

We can subtract images. Leaf example and chessboard example. Binary or grey scale images.

Arithmetic useful when parts of images should be excluded for example.

Logical operator in binary images, pixel example in slides. Nothing strange. But good method to add or remove certain objects in binary images.

noise reduction

using mean or median, useful in microscopy and night pictures/astronomy

$$I = \frac{1}{N} \sum_{n=1...n} I_k \tag{1}$$

Reduction of noise by using the mean of the pixels by using images of the same scene. Good in microscopy and astronomy were the scene doesn't change.

application

- Image arithmetic useful in medication/diagnosis. Subtracting picture before contrast fluid with the picture after to get en enhancement/ better picture of the blood vessels.
- change or motion in a scene. Persons in scene before and after example.
- illumination correction by subtraction background image. Max or median of the pixel intensities.

2 intensity transfer function

$$g(x,y) = Tf(x,y) \tag{2}$$

- linear (neutral negative, contrast, brightness)
- smooth, gamma log
- arbitrary

old value on x axis new on y axis.

The negative transformation

the inverse

$$g(x,y) = max - f(x,y) \tag{3}$$

An example

$$\begin{bmatrix} 255 & 254 & 253 \\ 125 & 130 & 110 \\ 4 & 3 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 1 & 2 \\ 130 & 125 & 145 \\ 251 & 252 & 255 \end{bmatrix} \tag{4}$$

Useful in medical image processing. Retina example. Easier to distinguish brighter lines/object. Sometimes the opposite.

Brightness

If we add a constant to the image it becomes brighter. Subtracting will make it darker. C positive integer or

$$g(x,y) = f(x,y) + C \tag{5}$$

Contrast

By multiplying the image we spread out the information and increases the contrast

$$g(x,y) = f(x,y) \times C, C > 1 \tag{6}$$

if C $<$ 1 reduce the contrast.

Gamma transformation

$$g(x, y) = C \times f(x, y)^{\gamma} \tag{7}$$

Computer monitors $\gamma \approx 2.2$

- Computer monitors $\gamma \approx 2.2$
- eyes ≈ 0.45
- microscopes ≈ 1

microscopes should have 1. 1 to 1 ratio. Lower gamma brighter image. Gamma high, darker.

Log transformation

Used to visualize dark regions of an image. To display the Fourier spectrum. Enhance the brighter regions.

$$g(x, y) = Clog(1 + f(x, y)) \tag{8}$$

arbitrary

only one output per input. Possibly not continuous.

3 Intensity histogram and histogram equalization

Gray scale histogram show how many pixels at each intensity level.

Normalized histogram: normalized by the total number of pixels in the image. Histogram show intensity distribution. How many pixels of certain intensity.

Intensity histogram doesn't say anything about the spatial distribution of pixel intensities. Images with the same pixels histogram can be totally different.

What do we use them for?

- Thresholding, intensity threshold. Decide intensity all above or under is background. Works with bi-modal histogram
- analyze the brightness and contrast
- histogram equalization

Analyze the brightness. See the transformation "chopping" the histogram. Could see that information might be missing. Low contrast = compressed histogram. When increasing we stretch the histogram. Transfer function slope.

histogram equalization

create an histogram with evenly distribution grey levels. for visual contrast enhancement. The goal is to flatten the histogram, produce the most even histogram.

Cumulative histogram

sum the number of pixels along the x axis intensity. Steep slope. Intensely populated parts of the histogram. Flat slope: sparsely populated parts of the histogram. Strive to a even slope.

example CDF

Multiplying the CDF value with number of gray levels -1 gives the intensity transfer function. We can the map the new gray level values into the number of pixels. it possible that two bins will be mapped to the same new position.

look at this again

local histogram equalization

useful when only parts of image need to be enhanced

Conclusion Image arithmetic

- useful when histogram narrow
- drawback, amplifies noise, can produce unrealistic transformation
- information can be lost. no new information gained.
- Not invertible, usually destructive.

Usefulness depends on the amount of different intensities.

Histogram matching

Want to mimic histogram of another image. Compute the histograms and CDF for each image. For each gray level G_1 $\begin{bmatrix}0 & 255\end{bmatrix}$ find graylevel G_2 so $F_1(G_1) = F_2(G_2)$ The matching function: $M(G_1) = G_2$. Not always the best solution either.

4 Summary Image arithmetic

- Many common tasks can be described by image arithmetic
- histogram eq useful for visualization
- watch out for information leaks

to think about

- relation between arithmetic and linear transfer function
- what can we know of an image from the histogram
- 8-bit image A, how will it look like B = 255*(A+1)
- conclusion if first last column really high?
- better resolution combining multiple images of same sample?

›

Lecture 3: Filtering part I

tuesday 08 nov 15:15

This lecture covers filtering and pre-processing, smoothing filters and edge enhancing. the second art covers filtering in Fourier domain and linear vs non-linear filters.

5 Image pre-processing/enhancement

We want to create a better image in some sense. In visual inspection:

- fir Visual inspection
- change contrast brightness
- subjective improvements

Important image information doesn't increase but can be better visualized In automated analysis

- restore an image, reduce noise
- enhance certain object, what we look for, dots, edges etc

Difference between point wise operations and filtering is that we use information from more neighbor pixels.

- local neighborhood
  - linear filter + filtering in freq domain
  - Non linear filter
- linear and non linear filter is basis for conv.nn

Spatial filtering

Make some transformation based on the neighborhood of x and y. Typically move the filter row by row from top to bottom.

$$g(x, y) = T(f()x, y) \tag{9}$$

Neighborhood, filter kernel, windowing function: the same thing. Inside we find weights.

Mean filter

Smoothing of sharp variation in intensity of the image. We start top left and move pixel by pixel through the row with the window function with the weight $\frac{1}{N}$ with N is the number of pixels in the window function. What do we do with the edges of the image? MATALB set everything outside the image is set to zero. So the edges becomes darker. Alternative is to reduce the window. Another alternative is to mirror the image but is computational heavy and introduces "false" information.

The window

The local neighborhood. The window is often square or disc shaped when becoming bigger. Increasing the size of the window makes for a smoother image. So only the **low frequency** variations in the image are kept, while **high frequency** variations is removed. The filter allays sums to one.

Gauss filtering

the Gaussian distribution got the area 1 under the curve and we can use this for filtering.

- Smoothing, reduces noise
- Less smoothing then mean but blur details less
- original intensity will be kept in a uniform part of the image

Gaussian filtering can be used for shading correction or remove/ decrease background variation. Take an input image, use Gaussian filtering, take the filtered image and subtract or divide from the original.

Edge enhancing filters

Enhances variations/edges, an edge can be seen as the same as gradient.

Example: Laplace filter

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{10}$$

The reason is if we want to find edges we want an output were we have changes not were the image is uniform. The filter enhance changes and uniform places are set to zero. Laplace filter, the filter sums up to zero.

Laplacian operator

Linear differential operator approximating the second derivative Produces only magnitudes and no direction information. The may result in two edges if there are a thin line. It is rather noise sensitive. 2nd derivative = line detector. The crisp filter, a visually sharper image can be obtained by adding the original image and the filtered image. This can also be obtained by adding 1 to the central weight of the filter.

$$\text{Input} \begin{bmatrix} & 1 & \end{bmatrix} \text{Laplace} \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix} \text{Crisp} \begin{bmatrix} & -1 & \\ -1 & 5 & -1 \\ & -1 & \end{bmatrix} \tag{11}$$

These are linear filter so we can do it one step.

Sobel operator

Approximation of the first derivative. Finds edges (gradients) in different directions. Read more about this

Example: Sobel operators

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{12}$$

flip filter to find different edges.

DoG: Differences of Gaussian

By combining smoothing filters of different size, edges can be detected. Think the first filter remove some of the high, the new remove more of them. And if we subtract the filtered images we get those higher frequency content.

6 Frequency domain filtering

Frequency = rate of change. High freq. corresponds to sharp edges, fine detail and noise. Low freq. correspond to smoother and slower changes.

Fourier transform: Functions that are not periodic bit with finite area under a curve, such as an image can be expressed as the integral of sines and cosines of different frequencies and weights. Representation using Fourier series or transforms allow for complete recovery of the original function. Fourier transform are used for:

- To reduce periodic noise
- To smooth or low pass
- To enhance details, high pass and band pass
- To save time convolution in time domain = multiplication in frequency domain.

The 2D discrete Fourier transform, and to get back using the inverse transform.

$$\begin{aligned} F(u,v) &= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \mathrm{f}(x,y) e^{-2j\pi(ux/M+vy/N)} \\ f(x,y) &= \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \mathrm{F}(u,v) e^{j2\pi(ux/M+vy/N)} \end{aligned} \tag{13}$$

At the center of the image we got the mean value of the image. Higher frequency moving out from the center. Some lines can appear in the transformed image (Fourier spectra). These line show were lot of differences/edges changes are present. Repeating pattern can become clear to see.

Some differences to continuous FT

DFT works on finite images with M × M pixels. → Frequency smoothing DFT uses discrete sampled images i.e. pixels → aliasing DFT assumes periodic boundary conditions → Centering, edge effects. Good thing to have when capture image to have less important information in the edges.

Convolution

DFT(f*g) = DFT(f) × DFT(g) much faster with multiplication in freq. domain.

Smoothing in spatial domain is same as low pass filtering in frequency domain.

Convolution $N_1^2 N_2^2$ operations. DFT: 4 $N^2 \text{Log}_2$ N

Use convolution for small convolving functions. DFT for large convolving functions.

Noise

All images contains noise. It can be noise from sensors, transmission, storage. Spatially independent noise can be removed by smoothing. Periodic noise is better removed in frequency domain.

Relationship between convolution and correlation

Convolution is equivalent to correlation if you rotate the convolution kernel by 180 degrees. Compute the correlation of the template image with the original image by rotating the template image by 180 degrees and then using fft based convolution (multiplication).

7 Comparing linear and non-linear filters

Linear filters (mean, Gauss, Sobel) we can add this filter and get the same result. filter(f1 + f2) = filter(f1) + filter(f2). Negative values should not be set to zero if this shall work. Linear filter are also shift invariant: filter(shift(f)) = shift(filter(f)). Same behavior regardless of pixel location. Linear filter have a correspondence in frequency domain. Linear filters are often separable, can be written as a product of two or more simple filters. Typically a 2D convolution operation can be separated into two 1D operations. This reduces computational cost.

Non linear filters like median, min and max and other morphological filters **DO NOT fulfill this properties**