Image processing

Errors of measurements

Many source of errors affect observations, like CCD images:

- Diffraction and seeing blur the image.
- The imaging optics causes geometric distortions.
- The imaging hardware itself produces some noise. The amount of noise is given by the signal to noise ratio. The ratio can be improved (increased) by repeating the obsevation several times and taking the average of the observations.
- Discretisation of the signal, depending on the number of bits of the A/D converter, causes some error. This is a random error that does not affect the average of several measurements. Statistical methods can be applied to control the effect.

- Sensitivity of different cameras at different wavelengths is different. This is a systematic error that can be detected only after comparing the data with observations made eith other instruments. Observations made with the same camera can be systematically wrong but still compatible with each others. When comparing observations made with different instruments, each instrument must be calibrated separately.
- The image may have reflections of bright objects (like the Moon). The problem is that different observations are incompatible, due to changing conditions. If a variable star is observed at different altitudes, the atmospheric perturbations are different. Observing comparison stars this error can be removed at least partly.
- The object may be so bright that the image is saturated and no measurement value is obtained. It is only known that the brightness exceeds some limit.

- Other perturbations (cosmic rays, disturbances due to poor electric isolation etc.).
- The data may also contain single deviating values (outliers). They can be erroneous observations or due to some external source. The reason for such deviations should be found out in order not to throw out exceptional but real observations that might indicate some interesting phenomenon.
- Many observing methods involve their own specific reduction techniques to convert the raw data into actual values that can be analysed. An example in image processing is the removal of geometric distortions caused by the imaging system.

The purpose of image processing is to deal with these problems. It includes e.g.

- data acquisition and storage
    - digitisation of an analogous signal
    - data compression
    - data representation
- image restoration
    - geometric transforms
    - noise reduction
    - deconvolution
    - detail enhancing
- image analysis
    - edge detection
    - object recognition
    - classification

Image representations

An image is usually stored as a matrix, each element of which correponds to one pixel. Another possibility would be to store the coeddicients of a Fourier series. Both are special cases of a general representation.

Let f be quadratically integrable function defined in some region S. We want to express the function in terms of some basis functions ϕ_{mn}, $m, n = 0, 1, \ldots$. The basis functions are orthonormal, if

$$\int\int_S \phi_{mn}(x,y)\phi^*_{rs}(x,y)\,dx\,dy = \delta_{mr}\delta_{ns}.$$

We are looking for an expansion

$$\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} a_{mn}\phi_{mn}(x,y)$$

that will minimise the sampling error e^2:

$$e^2 = \int\int_S \left| f(x,y) - \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} a_{mn}\phi_{mn}(x,y) \right|^2 dx\, dy,$$

This is satisfied when

$$a_{mn} = \int\int_S f(x,y)\phi^*_{mn}(x,y)\, dx\, dy.$$

The set of basis functions is complete, if for each quadratically integrable function the error $e \to 0$, when the number of terms is increased.

Standard representations: Basis functions are rectangles:

$$\phi_{mn}(x,y) = \begin{cases} \sqrt{\frac{MN}{AB}}, & \text{jos } x \in [\frac{mA}{M}, \frac{(m+1)A}{M}) \text{ ja } y \in [\frac{nB}{N}, \frac{(n+1)B}{N}), \\ 0 & \text{muuten.} \end{cases}$$

The coefficient a_{mn} is simply the mean value of f in the rectangle where $\phi_{mn} \neq 0$.

Fourier representasion: Basis functions are

$$\phi_{mn}(x,y) = \frac{1}{\sqrt{AB}} e^{2\pi i(mx/A + ny/B)},$$

where A and B are the width and height of the image, respectively.

Optimal representation: Is there a basis that will make the error e^2 as small as possible.

Minimising the error will lead to the eigenvalue problem

$$\int\int R_{ff}(x,x',y,y')\phi_{m,n}(x',y')\,dx'\,dy' = \gamma_{mn}\phi_{mn}(x,y).$$

The optimal basis functions, known as Karhunen and Loève functions, are obtained as the solution of this equation.

Quantization

Intensities are continuous real values, which must be mapped to a finite number of gray levels. Let the limiting intensities be z_k, $k = 1, 2, \ldots, K+1$. If the intensity is in the range $[z_k, z_{k+1})$, the pixel will be given the value q_k. If $p(z)$ is the probability that the intensity is z, the quantization error is

$$\delta_q^2 = \sum_{k=1}^{K} \int_{z_k}^{z_{k+1}} (z - q_k)^2 p(z)\, dz.$$

Minimising this we get

$$\frac{\partial \delta_q^2}{\partial z_k} = (z_k - q_{k-1})^2 p(z_k) - (z_k - q_k)^2 p(z_k) = 0, k = 2, 3, \ldots, K,$$
$$\frac{\partial \delta_q^2}{\partial q_k} = -2 \int_{z_k}^{z_{k+1}} (z - q_k) p(z)\, dz = 0, k = 1, 2, \ldots, K.$$

or

$$z_k = \frac{q_{k-1} + q_k}{2},$$
$$q_k = \frac{\int_{z_k}^{z_{k+1}} z p(z)\, dz}{\int_{z_k}^{z_{k+1}} p(z)\, dz.}$$

If $p(z)$ is constant, the limits will be equally spaced.

In practice quantization is carried out directly by the hardware and the limits are equally spaced even if the result may not be optimal.

Compression

Compression is needed to reduce storage space and also to speed up data transfer (important e.g. in communication with a space probe).

Often neighbouring pixels are correlated, and space is wasted by storing data with low information content.

There are many methods for compression:

- Transformation methods express the image as a combination of suitable basis functions.
- Coding of gray levels can be changed to adapt to the nonrandomness of the distribution (shorter codes are used for the most common values).
- Use some fractal process to represent the image.

Noiseless methods keep all information of the original image; the image can be reconstructed exactly.

Noisy methods keep only the essential features; usually the smallest details are lost.

If the probability of gray level q_i is p_i, the information content of the image is

$$H = -\sum_{i=1}^{K} p_i \log_2 p_i.$$

This gives a lower limit for the number of bits per pixel that are needed to reconstruct the original image exactly. (Some methods may claim to give a very high compression ratio, but the amount of compression is always limited by the information content.)

White noise (pixels not correlated at all) has the highest information content and cannot be compressed.

Transformation compression: Express the image in terms of basis functions and keep only the most important terms. Choosing the best base requires some information about the images.

Karhunen and Loève transform (KL transform): If the image is expressed as a combination of the KL functions, the coefficients are independet. Finding the basis functions is a laborious eigenvalue problem. If the autocorrelation function is assumed to be known, expressions of the basis functions can be derived.

Fourier transform: Easy to calculate, but usually to obtain the same image quality much more terms are needed than using the KL transform.

Hadamard transform: A very efficient transform; basis matrices contain only numbers $+1$ and -1 (and a normalization constant). The basis matrices for an $N \times N$ image, where N is a power of two, are

$$\phi^{(u,v)}(m,n) = \frac{1}{N}(-1)^{b(u,v,m,n)},$$

where

$$b(u,v,m,n) = \sum_{k=0}^{\log_2 N-1} (b_k(u)b_k(v) + b_k(m)b_k(n)),$$

and $b_k(x)$ is the k^{th} bit of the number x.

Run length coding: Pixels are replaced by pairs (v,n), where v is the value of the pixel and n the number of successive pixels having the same value. If the image contains large areas with the same colour, a lot of space is saved.

Huffman coding: Common values are represented by short codes and rare ones by longer codes.

Fractal coding: Natural objects rarely have simple geometric shapes, but have often a selfsimilar structure (a magnified detail looks similar to the whole). Fractal comprssion is based on iterated function systems (IFS, iterated function systems, ks. esim. Barnsley: *Fractals everywhere*, Academic Press 1988.). IFS is a random walk at each point of which one of a given set of transforms is selected with a certain probability. Computationally heavy but can greatly compress complex images.

Wavelets

Wavelets can be used as a local analysis method. In a way wavelets split the data into wave packets of different sizes.

Wavelets can be used in data compression e.g. in image processing.

Wavelets are functions of one variable and two indices:

$$g_x = \frac{1}{\sqrt{|a|}} g\left(\frac{x-b}{a}\right), \quad a \neq 0,$$

where g satisfies

$$C = 2\pi \int_{-\infty}^{\infty} |\xi|^{-1} |F(g)| d\xi < \infty,$$

where $F(g)$ is the Fourier transform of g.

It follows from the definition that at the origin $F(g)$ is zero and the mean value must be

$$\int_{-\infty}^{\infty} g(x)dx = 0.$$

Hence g must change sign somewhere and $g(x)$ approaches zero when x approaches plus or minus infinity.

The mother function $\psi(t)$ can be used to generate a family of wavelets that can be scaled and translated:

$$\psi_{s,l}(t) = \frac{1}{2^{s/2}} \psi \left(\frac{t}{2^s} - l \right)$$

Here s is the scale index and l position index.

The width of the wavelet depends only on the quantity

$$S = 2^s,$$

which can in a way considered analogous to the frequence in the Fourier space.

The position index l has no corresponding quantity in the Fourier space. $\psi_{s,l}(t)$ can be chosen as an orthonormal family of functions. These properties make wavelets efficient tools.

In the following we discuss only discrete wavelets and equally spaced data points. Assume that there are 2^N data points X_i, where N is an integer.

Now a wavelet transform can be defined as

$$W^x_{s,l} = \sum_{i=1}^{N} \psi_{s,l}(i) X_i,$$

$W^x_{s,l}$ is a measure of variations in the scale s at the location l. The index $s = 0$ can refer either to the largest or smallest scale. If it refers to the smallest scale larger scales are indicated by positive indices.

If the wavelets form an orthonormal set, also the inverse operation is available:

$$X_i = \sum_{s,l} \psi_{s,l}(i) W^x_{s,l}.$$

Due to the locality of wavelets they can effectively detect sharp discontinuities and singularities (Haar wavelets) or oscillations lasting a short time (Morelet wavelets).

Haar wavelet

The mother function is

$$\psi(t) = \begin{cases} 1, & 0 \leq t < \frac{1}{2}t_0 \\ -1, & \frac{1}{2}t_0 \leq t < t_0 \\ 0, & \text{elsewhere.} \end{cases}$$

In the smallest scale the wavelet is

$$h_n = \begin{cases} h_0 & n = 1 \\ -h_0 & n = 2 \\ 0 & \text{elsewhere.} \end{cases}$$

The factor $h_0 = \frac{1}{\sqrt{2}}$ is a normalization constant. The wavelet of the next scale is twice as long and the amplitude is $\frac{1}{\sqrt{2}}$ -fold. Thus the "power" in each wavelet is the same.

The Haar wavelets form an orthonormal set.

Daubechies wavelets

The wavelet D_4 is

$$\psi_{s,l}(x) = \frac{1}{2^{s/2}}\psi\left(\frac{x}{2^s} - l\right),$$

wher $\psi(x)$ is the mother of D_4:

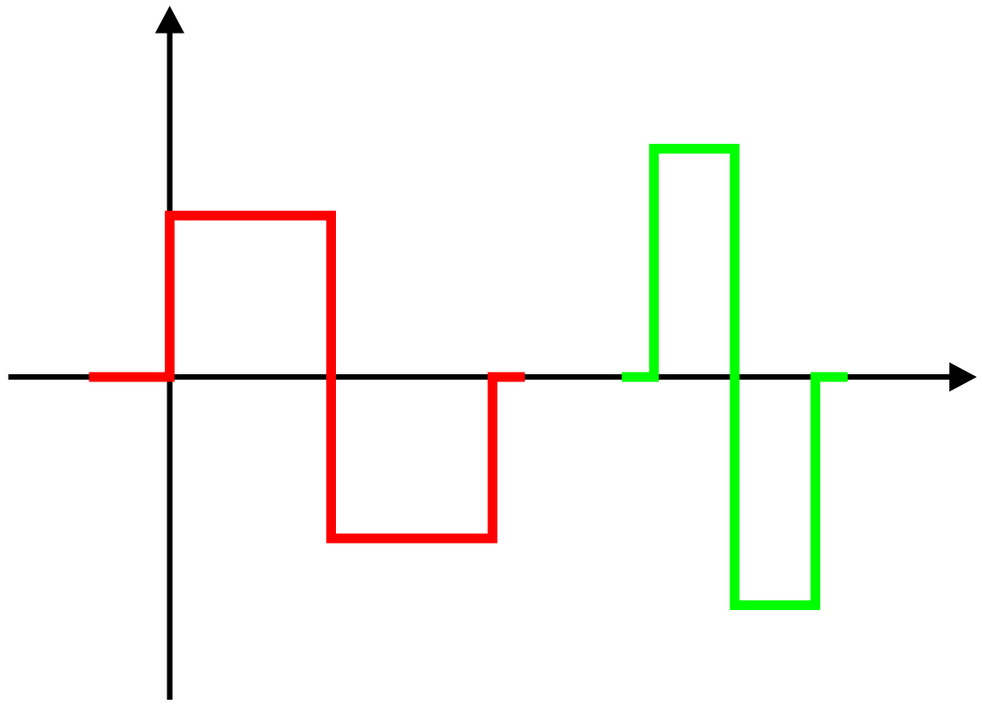
$$\psi(x) = c_1\phi(2x) - c_0\phi(2x-1) - c_2\phi(2x+1) - c_3\phi(2x+2),$$

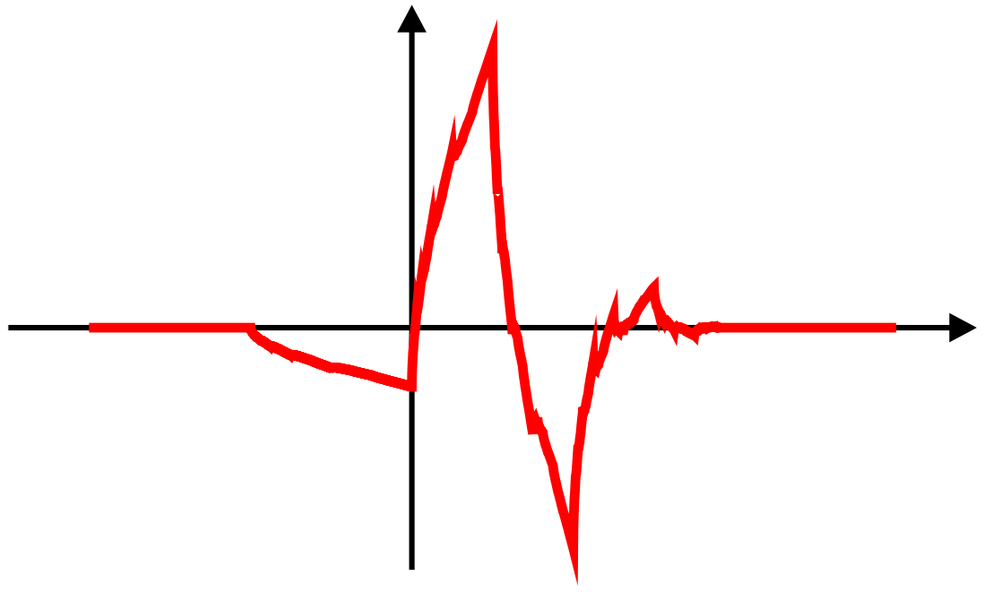
and $\phi(x)$, teh father od D_4 is defined as

$$\phi(x) = c_0\phi(2x) + c_1\phi(2x-1) + c_2\phi(2x-2) + c_3\phi(2x-3).$$

The coefficients c_i are

$$c_0 = \frac{1}{4}(1+\sqrt{3}),\ c_1 = \frac{1}{4}(3+\sqrt{3}),\ c_2 = \frac{1}{4}(3-\sqrt{3}),\ c_3 = \frac{1}{4}(1-\sqrt{3}).$$





Morelet wavelet

$$\psi_{s,l} = \exp\left(-ic\left(\frac{x-l}{s}\right) - \frac{1}{2}\left(\frac{x-l}{s}\right)^2\right).$$

If c is small the wave packet is wide but less sensitive to noise. A compromise could be $c = 2\pi$. If we denote $s = 1/\nu$, we can see a similarity to the Fourier transform:

$$\psi_{s,l} = e^{-i2\pi\nu(t-t_0)} \cdot e^{-\frac{1}{2}\nu(t-t_0)^2}$$

Mexican hat

$$\psi(x) = (1-x^2)e^{-\frac{1}{2}x^2}.$$

These do not form an orthonormal set and they do not have a proper mother wavelet. They have been used widely because of their mathematical simplicity.

Image formats

eps (Encapsulated PostScript) Text, actually program code. Well suited for vector graphics, not as good for pixel images. Resolution independent and freely scalable.

tif (Tagged Image format) Pixel image, not compressed, files are big.

gif (CompuServe graphics interchange format) Compressed noisefree image.

jpeg (Joint Photographic Experts Group file interchange format) Compressed noisy image. Accuracy suffers a little, but savings in space are considerable.

mpeg (Motion Picture Experts Group file interchange format) Compression method for motion pictures. The colour of a given pixel does not usually change between successive frames. A lot of space is saved by storing only the changes.

pdf (Portable Document Format) Standard mainly for publications; much more compact than eps.

The `convert` command in Linux can convert lots of image formats to other ones.

```
 66146 Apr 15 13:40 fo\-rest.bmp ==>
524554 Apr 15 13:42 fo\-rest.tif
218378 Apr 15 13:41 fo\-rest.eps
 75115 Apr 15 13:40 fo\-rest.gif
 24881 Apr 15 13:43 fo\-rest.jpg

  826 Dec 17  1997 co\-lor.ps ==>
2908547 Apr 15 13:59 co\-lor.tif
1454166 Apr 15 14:00 co\-lor.bmp
 551223 Apr 15 14:01 co\-lor.gif
 137581 Apr 15 14:17 co\-lor.eps
 47422 Apr 15 13:59 co\-lor.jpg
```

Geometric transforms

A geometric transform maps a point (x, y) to another point (x', y'). In vector graphics points are usually expressed as column vectors with one additional element. Thus a point (x, y) on a plane is given as

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

Then transforms can be expressed as 3×3 matrices

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \mathbf{M} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

1) translation

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}.$$

2) scaling

$$\mathbf{M} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

3) rotation

$$\mathbf{M} = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

These can be used for line drawings (vector graphics), NOT for pixel images.

Geometric transforms of pixel images

1) Direct transform:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \to \mathbf{M} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}.$$

Not good, because the mapping is not one-to-one:

- some pixels may not be assigned any value
- several pixels may get mapped to the same pixel

2) Inverse transform: for each (x', y') find a pixel (x, y) that will be mapped to (x', y') and assign its value to the pixel (x', y'). This is better, because will not leave any empty pixels. Still several pixels of the original image may be mapped to the same pixel, and some to none.

3) Inverse interpolation: find (x, y) as before. Usually x and y are not integers; thus the value corresponding to the point (x, y) is interpolated using the values of the nearest neighboring pixels. This works quite well, particularly for moderatley continuous images. Sharp edges may become slightly blurred.

4) Reconstruction: express the original image in terms of some functions, and use the functions to calculate the values of the new pixels. Slow, but often gives the best results.

Deconvolution

Assume that the psf h is invarian w.r.t translations. The observed image p and the original one f are then related by

$$p(x,y) = \int\int h(x-x', y-y') f(x',y')\, dx'\, dy' + \nu(x,y),$$

In the least squares sense the optimal filter is the Wiener filter:

$$M(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_{\nu\nu}(u,v)/S_{ff}(u,v)}.$$

If the noise is white noise, $S_{\nu\nu}$ is constant and approximately

$$S_{\nu\nu}(0,0) = \int_{-\infty}^{\infty} R_{\nu\nu}(x,y)\, dx\, dy.$$

In practice, it is easiest just to experiment which value gives the most satisfactory result.

Another method is based on the concept of enthropy

$$H_f = -\sum_m \sum_n f(m,n) \ln f(m,n).$$

If the total energy

$$f_{\text{tot}} = \sum_m \sum_n f(m,n)$$

remains constant, the enthropy is the smaller the more the values $f(m,n)$ vary.

Noise can also get negative values. Add a constant C to the noise so that $\nu(m,n) + C$ is always positive:

$$H_\nu = -\sum_m \sum_n (\nu(m,n) + C) \ln(\nu(m,n) + C),$$

In the maximum enthropy method we maximize the expression

$$H = H_f + \alpha H_\nu,$$

where α is a constant describing the amount of smoothing. In addition, two constraints are needed:

1) The total energy f_{tot} must be conserved.
2) The observed image p must be of the form

$$p(q,r) = \sum_m \sum_n h(q-m, r-n) f(m,n) + \nu(m,n),$$

i.e the convolution of the original image f and the psf h + noise. By solving this we get f and ν.