Image processing: A quick reference sheet for the exam

by Tom T. Doodle

October 30, 2014

Contents

1 Introduction

This document is done in order to describe the general concepts that has been covered in the course, and also provide answers to some common exam questions. The content is shamelessly ripped from various sources, which is indicated at the start of a section or paragraph.

Guide:

[**W**] Wikipedia

[**L**] Lecture notes

[**E**] Exam

[**B**] Course Book

[**P03**] Paper: Romeo et al. (2003)

[**P04**] Paper: Romeo et al. (2004)

[**O**] Other

None Missing or anecdotal

2 Noise

2.1 General concepts

2.1.1 What is noise?

[W] Image noise is random (not present in the object imaged) variation of brightness or color information in images, and is usually an aspect of electronic noise. It can be produced by the sensor and circuitry of a scanner or digital camera. Image noise can also originate in film grain and in the unavoidable shot noise of an ideal photon detector. Image noise is an undesirable by-product of image capture that adds spurious and extraneous information.

The original meaning of "noise" was and remains "unwanted signal"; unwanted electrical fluctuations in signals received by AM radios caused audible acoustic noise ("static"). By analogy unwanted electrical fluctuations themselves came to be known as "noise". Image noise is, of course, inaudible.

The magnitude of image noise can range from almost imperceptible specks on a digital photograph taken in good light, to optical and radioastronomical images that are almost entirely noise, from which a small amount of information can be derived by sophisticated processing (a noise level that would be totally unacceptable in a photograph since it would be impossible to determine even what the subject was).

2.2 Various types of noise

2.2.1 Salt and pepper noise

[W] Salt-and-pepper noise is a form of noise sometimes seen on images. It presents itself as sparsely occurring white and black pixels. An effective noise reduction method for this type of noise is a **median filter** or a morphological filter. For reducing either salt noise or pepper noise, but not both, a contraharmonic mean filter can be effective.

2.2.2 Gaussian noise

[W] Gaussian noise is statistical noise having a probability density function (PDF) equal to that of the normal distribution, which is also known as the Gaussian distribution. In other words, the values that the noise can take on are Gaussian-distributed.

2.2.3 Poissonian noise

TODO

2.2.4 Additive

Additive because it is added to any noise that might be intrinsic to the information system.

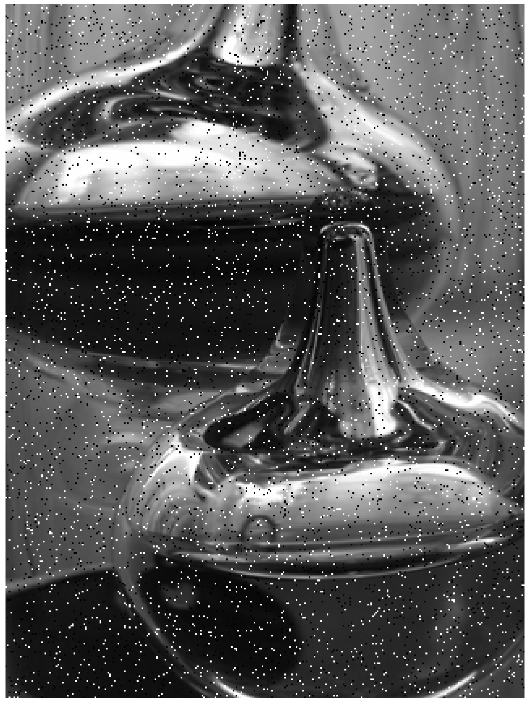

Figure 1: Example of salt and pepper noise

2.2.5 Multiplicative

In signal processing, the term multiplicative noise refers to an unwanted random signal that gets multiplied into some relevant signal during capture, transmission, or other processing.

2.2.6 Exponential

TODO

2.3 Colors of noise

2.3.1 White noise

[W] White noise is a signal (or process), named by analogy to white light, with a flat frequency spectrum when plotted as a linear function of frequency (e.g., in Hz). In other words, the signal has equal power in any band of a given bandwidth (power spectral density) when the bandwidth is measured in Hz.

2.3.2 Pink noise

TODO

2.3.3 Blue noise

[W] In computer graphics, the term ”blue noise” is sometimes used more loosely as any noise with minimal low frequency components and no concentrated spikes in energy. This can be good noise for dithering.

3 Filters

3.1 Median filter

3.1.1 Description

[W] In signal processing, it is often desirable to be able to perform some kind of noise reduction on an image or signal. The median filter is a nonlinear digital filtering technique, often used to remove noise. Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image). Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise.

3.1.2 Algorithm description

[O] Like the mean filter, the median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings.

Instead of simply replacing the pixel value with the mean of neighboring pixel values, it replaces it with the median of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. (If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used.)

Why use median instead of mean:

- The median is a more robust average than the mean and so a single very unrepresentative pixel in a neighborhood will not affect the median value significantly.
- Since the median value must actually be the value of one of the pixels in the neighborhood, the median filter does not create new unrealistic pixel values when the filter straddles an edge. For this reason the median filter is much better at preserving sharp edges than the mean filter.

3.2 Mean/Averaging filter

[O] The idea of mean filtering is simply to replace each pixel value in an image with the mean ('average') value of its neighbors, including itself. This has the effect of eliminating pixel values which are unrepresentative of their surroundings.

Problems with mean filtering:

- A single pixel with a very unrepresentative value can significantly affect the mean value of all the pixels in its neighborhood.
- When the filter neighborhood straddles an edge, the filter will interpolate new values for pixels on the edge and so will blur that edge. This may be a problem if sharp edges are required in the output.

Both of these problems are tackled by the median filter, which is often a better filter for reducing noise than the mean filter, but it takes longer to compute.

4 Image sharpening

The Basic concept is as follows:
Sharpened image = Original image - smoothened image.
You can smoothen an image by applying a lowpass filter on it.

4.1 Edge detection

A lot of different ways. One basic approach for finding edges is by detecting changes in intensity, which can be accomplished using first or second order derivatives.

[W] There are many methods for edge detection, but most of them can be grouped into two categories, search-based and zero-crossing based. The search-based methods detect edges by first computing a measure of edge strength, usually a first-order derivative expression such as the gradient magnitude, and then searching for local directional maxima of the gradient magnitude using a computed estimate of the local orientation of the edge, usually the gradient direction. The zero-crossing based methods search for zero crossings in a second-order derivative expression computed from the image in order to find edges, usually the zero-crossings of the Laplacian or the zero-crossings of a non-linear differential expression. As a pre-processing step to edge detection, a smoothing stage, typically Gaussian smoothing, is almost always applied.

The edge detection methods that have been published mainly differ in the types of smoothing filters that are applied and the way the measures of edge strength are computed. As many edge detection methods rely on the computation of image gradients, they also differ in the types of filters used for computing gradient estimates in the x- and y-directions.

4.2 High Boosting

[B] High-boost filtering is used for sharpening an image $f(x, y)$. It is carried out in several steps, namely:

- Unsharp masking
  1. Blur the original image $\rightarrow \hat{f}(x, y)$
  2. Subtract the blurred image from the original (yielding a mask) $\rightarrow g_{\text{mask}}(x, y) = f(x, y) - \hat{f}(x, y)$
  3. Add the mask to the original $g(x, y) = f(x, y) + k * g_{\text{mask}}(x, y)$
- If $k > 1$ this will yield a high boosted image.

5 Image histograms

5.1 Histogram equalization

[W] This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values.

The method is useful in images with backgrounds and foregrounds that are both bright or both dark. In particular, the method can lead to better views of bone structure in x-ray images, and to better detail in photographs that are over or under-exposed. A key advantage of the method is that it is a fairly straightforward technique and an invertible operator. So in theory, if the histogram equalization function is known, then the original histogram can be recovered. The calculation is not computationally intensive. A disadvantage of the method is that it is indiscriminate. It may increase the contrast of background noise, while decreasing the usable signal.

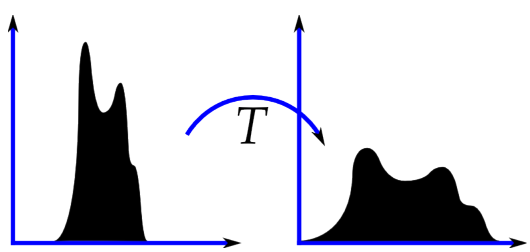

Figure 2: Example of histogram equalization

5.1.1 The idea behind histogram equalization

[L] Our eyes (brain) can see more details if the image histogram is wide and approximately flat, rather than narrow and peaked.

5.1.2 Example: How to equalize a histogram in practice

[E] Assume you have a 2-bit image like this:

$$\begin{bmatrix} 1 & 1 & 2 & 2 \\ 2 & 1 & 3 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 2 & 2 & 1 \end{bmatrix} \tag{1}$$

How do you equalize its histogram?

Answer: Create a table like table 1. Then simply create the new equalized image by replacing the original values with the new values:

Orig. Value	Probability	Cum. prob.	Rounded	New Value
0	1/16	1/16	0/3	0
1	9/16	10/16	2/3	2
2	5/16	15/16	3/3	3
3	1/16	16/16	3/3	3

Table 1: Histogram equalization table

$$\begin{bmatrix} 2 & 2 & 3 & 3 \\ 3 & 2 & 3 & 2 \\ 2 & 2 & 0 & 2 \\ 2 & 3 & 3 & 2 \end{bmatrix} \tag{2}$$

This is the new equalized image. This histogram is poorly equalized due to small number of both bits and pixels.

5.1.3 Histogram equalization of a one-color image

Assume you equalize a histogram of an image of just one colour. As the probability will reach maximum at that gray value, the cumulative and rounded value will as well. Thus all original values will be replaced with the maximum gray value (white).

5.2 Histogram matching

[W] Histogram matching is a method in image processing of color adjustment of two images using the image histograms.

It is possible to use histogram matching to balance detector responses as a relative detector calibration technique. It can be used to normalize two images, when the images were acquired at the same local illumination (such as shadows) over the same location, but by different sensors, atmospheric conditions or global illumination.

5.2.1 Algorithm

[O] The algorithm is as follows. The cumulative histogram is computed for each dataset, see figure **??**. For any particular value (x_i) in the data to be adjusted has a cumulative histogram value given by $G(x_i)$. This in turn is the cumulative distribution value in the reference dataset, namely $H(x_j)$. The input data value x_i is replaced by x_j.

In practice for discrete valued data one does not step through data values but rather creates a mapping to the output state for each possible input state. In the case of an image this would be a mapping for each of the 256 different states.

6 Wavelets

6.1 The fundamental Propertis of Wavelets

Wavelets are a multiscale method that can overcome some the difficulties with the Fourier Transform. Their fundamental property is to provide an adaptive time/space-frequency resolution to the the frequency itself [constant relative bandwidth, in the language of data processing. In other words, this means that small-scale features of the data are analysed with fine resolution in time/space and coarse resolution in frequency, as is natural, and vice versa for large-scale features.

6.2 The Wavelet Transform

TODO Maybe

6.3 Basics of Wavelet Applications

Wavelets can be used for alot of things. Some common use-areas are:

6.3.1 Waveler properties

[P04] *Size of support* The support of a wavelet is the interval where the wavelet is non-zero. Its size determines not only the time/space localization of the wavelet, but also the speed of the transform.

Symmetry also influences the quality of time/space localization. For example, an asymmetric wavelet can be regarded as giving a location with asymmetric error bars.

Number of vanishing moments A wavelet $\psi(x)$ has n vanishing moments when

$$\int_{-\infty}^{\infty} x^{v}\psi(x)dx = 0, \text{for } \upsilon = 0, 1, \ldots, n1 \tag{3}$$

where x denotes time or space. In particular, all normal wavelets have zero mean (n = 1) since, under rather general assumptions, this is related to the admissibility condition for the existence of the inverse transform. The number of vanishing moments affects the frequency localization. In fact, the Fourier transform of a wavelet with n vanishing moments peaks at a characteristic frequency and decays as kn towards the origin, where k denotes frequency.
Regularity also affects the frequency localization. In fact, the Fourier transform of a wavelet that is continuous together with its first n 1 derivatives decays as k(n+1) towards infinity.
(Bi-)Orthogonality The orthogonality property concerns the set of wavelets defining the transform, that is the set of scaled and trans- lated versions of the basic wavelet. This means that such wavelets form an orthogonal basis. The alternative bi-orthogonality property means that the decomposition and

reconstruction wavelets form two distinct bases, which are mutually orthogonal. Note that (bi-)orthogonality is intimately related to the non-redundancy of the transform.

6.3.2 Difference of orthogonal, (bi-Orthongonal and quasi orthogonal

TLDR: quasi-orthogonal ¿ orthogonal
more vanishing moments, larger support, higher regularity and perfect symmetry.

[P04] The wavelets haar and daub 4 belong to the family of Daubechies wavelets, and are simple (see Daubechies 1992; here we use the same names as in the code). These wavelets are orthogonal; and they have few vanishing moments, small support, low regu- larity and no symmetry (haar is an exception). The wavelet pairs bior 4.4 and rbio 6.8 belong to the family of bi-orthogonal spline wavelets and to its reverse, respectively, and are more advanced (see Daubechies 1992; here we use the same names as in the code). Such wavelet pairs are not only bi-orthogonal but also quasi-orthogonal: for each pair the decomposition and reconstruction wavelets are dis- tinct but similar. In addition, they have more vanishing moments, larger support, higher regularity and perfect symmetry.

6.3.3 Data Compression

The adaptive time/space-frequency resolution and the non-redundancy of the fast wavelet transform have an iumportant implication: given regular data, most information present in the gets concentrated into a few large wavelet coefficients. In practice, this means that we can set all the other coefficients to zero and get back data almost identical to the original ones. This is also the idea behind image compression. The most important requirement for good compression ability is that the decomposition wavelet should have many vanishing moments, and the basic reason is that a wavelet with n vanishing moments is insensitive to polynomials of degree n-1.

6.4 De-noising

The compression ability of the fast wavelet transform has a further importan implication: given noisy data, the underlying regular parts gets mostly mappen into many small wavelet coefficients, whereas noise is mostly mappen into many small wavelet coefficients. In practice, this means that, if we identify a correct threshold. then we can set all small coefficients to zero and get back data almost decontaminated from noise. This is the idea behind data denoising.

7 Image compression

Compression ratio $C = b/L$, where b and L carries the same information, and L is the compressed data.

Relative data redundancy $1 - 1/C$

Shannons entropy theorem (maximum compression rate) $-\sum w_i \log_2 w_i$

7.1 Redundancy types

[B]

7.1.1 Coding redundancy

A *code* is a system of symbols (letters, number, bits and the like) used to represent a body of information or set of events. Each piece of information or event is assigned a sequence of *code symbols*, called a *code word*. The number of symbols in each code word is its *length*. The 8-bit codes that are used to represent the intensities in most 2-D intensity arrays contain more bits than are needed to represent the intensities.

7.1.2 Spatial and temporal redundancy

Because of the pixels of most 2-D intensity arrays are correlated spatially (i.e., each pixel is similar to or dependent on neighboring pixels), information is unnecessarily replicated in the represintations of the correlated pixels. In an video sequence, temporally correlated pixels (i.e., those similar to or dependent on pixels in nearby frames) also duplicate information.

7.1.3 Irrelevant information

Most 2-D intensity arrays contain information that is ignored by the human visual system and/or extreneous to the intended use of the image. It is redundant in the sense that it is not used.

7.2 Huffman coding

7.3 Run-length coding

Run-length encoding (RLE) is a lossless compression technique. The RLE is a simple form of data compression in which runs of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run. This is most useful on data that contains many such runs.

Example:
The black and white image (displayed here as horizontal raster scanned):

```
WWWWWWWWWWWWBWWWWWWWWWWWWBBBWWWWWWWWWWWWWWWWWWWWWWWWBWWWWWWWWWWWWWW
```

can be expressed as:

```
12W1B12W3B24W1B14W
```

This is to be interpreted as twelve Ws, one B, twelve Ws, three Bs, etc. The run-length code represents the original 67 bit-values in only 18.

8 Old exams

8.1 2014/2013

1a)

Sharpened image = Original image - smoothened image.
You can smoothen an image by taking a lowpass filter.

1b)

Amplify high frequency components while keeping the low frequency intact. This will enhance images and remove some blurry parts.
High boost image = Original image + Sharpened image.
By 1a).

1c)

For instance use: The Laplacian of gaussian method. Works by looking for zero-crossings after filtering the image with the Laplacian of a gaussian filter

1d)

Yes. By amplifying high frequencies we can sharpen an image and by looking at high frequencies we can find edges.

1e)

Finns i facit.

2a)

TODO

2b)

Denoising: W3 because it looks like the reconstruction wavelet, behaves more than W2 like an orthogonal and has more vannishing moments than the orthogonal.

Wavelets are good for additive white gaussian noise so yes.

If not Gaussian, preprocess and map into white-gaussian.

For salt and pepper noise you can not use wavelets, need to use a median filter.

8.2 2012/2013

1a)

A basic noise model used to mimic the effect of many random processes that ocure in nature.
Additive - Because it is added to any noise that might be intrinsic to the information system.
White - Has uniform emissions at all freq in the visible spectrum.
Gaussian - Because it has a normal distribution in the time domain with an average time domain value of zero.
Can be removed using (adaptive) median filters or linear filtering.

1b)

Poissionian Noise is a type of electrical noise originating from the discrete nature of electric charge. Gaussian noise is continious were as poissonian noise is discrete. Poissonian noise can occure in TV-streams.
To remove Poissonian noise, maybe use an averaging filter because the noise is statistic.

1c

By denoising we remove information from the image, which means we're compressing it.

Exists facit for the rest

8.3 2011/2012

1a)

It can be used to change contrast. By taking the histogram and removing peeks, and stretch the rest, we get higher contrast. (stretching the gray scale)

1b)

Can be used for color adjustment and gray scale adjustment (not just by histogram equalize).

1c)

Low-pass filter passes signals under a certain cut-off frequencies. All freq above is dampened but not necessary removed. Ideal filter completely eliminates all above cut-off freq. Can be used against wide-random noise. (maybe same thing as additive gaussian noise?)

Cons: Ideal filter will remove all high freqs, which means that all edges will be smoothened.

Pros: Good if you don't have many distinct edges, because it can remove a lot of high freqs. Not used so much in image processing but rather in aduio processing, where you want to completly remove certain freqs (above a certain freq).

1d)

Works pretty much the same way but not ideal so wont remove all freqs above cut-off freq.

Resten har facit.

8.4 2010/2011

1a)

Averaging filter works as a mean-filter.

1b)

It will work better with a median filter.

3c)

Helps to remove blur. Optimal means optimal in the sense of MMSE, minimum mean square error.

TODO

Resten har facit.

8.5 2009/2010

1a)

Trivialt ordbajs.

1b)

When enhancing the lower frequencies or dampening the high frequencies. Thereseholding is the simplest method of image segmentation. Can be used to create binary images from a gray scale image. All pixel values (A) to the left of the centered line (B) will be white, and all to the right (C), will be black.

```
                  --------------
                 |
                 |
  --------------|
    A     <     B     <     C
```

1cde)

Finns facit. Det som inte finns facit p skulle han inte ta med.

2c)

FWT (Fast Wavelet Transform)
Pros: Good ability to compress stationary signals, able to support some interesting noneucledian similarity measures.
Cons: Signals must have length: 2^{INT}. Can't support weighted distance measures.

BDCT (Block discrete cosine transform)
Better at comparisons. The difference from FWT is that it takes real cosine

functions instead of a set of harmonicaly related complex exponential functions like the DFT. Specially used for lossy data compression because of it's strong energy compaction property. Good and bad.

DFT (Discrete Fourier Transform)
Good for same things as BDCT, namely Lossy image compression.
Pros: Good ability to compress most natural signals. Fast $\mathcal{O}(n \cdot \log(n))$. Weekly abeled to support timewarped queries.
Cons: Difficult to deal with sequences of different lengths. Can't support weighted distance measures.

2008/2009

Finns facit p allt.