Visual Computing
by dcamenisch

1 Introduction

This document is a summary of the 2022 edition of the lecture *Visual Computing* at ETH Zurich. I do not guarantee correctness or completeness, nor is this document endorsed by the lecturers. If you spot any mistakes or find other improvements, feel free to open a pull request at TODO: INSERT LINK. This work is published as CC BY-NC-SA.

2 The Digital Image

An image is simply a continuous function over 2 or 3 variables (XY-coordinates and possibly time). Usually we use brightness as the value of the function, but other physical values can also be used. For a computer this is just a collection of numbers, but instead of continuous values we have discrete. Note that in real life images are never completely random and almost always contain some structure. It is important to know that **pixels are not little squares**, they are point measurements.

When taking a picture with a digital camera, we can encounter various problems, e.g.:

- Transmission Interference
- Compression Artefacts
- Spilling
- Sensor Noise

2.1 Sampling

When taking an image, we are sampling such a continuous. When trying to reconstruct the original function, we can encounter undersampling, i.e. when we loose information due to a too low amount of sampling points.
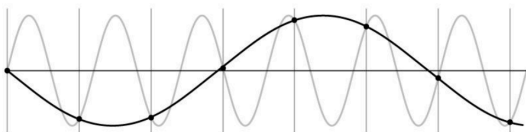

Due to undersampling, the result can not be distinguished from a lower or a higher frequency wave. Signal disguised as other frequencies is also called **aliasing**.

Nyquist-Shannon Sampling Theorem
For sine waves we have to sample at half the wave length. This corresponds to double the frequency, we also call this the **Nyquist Frequency**.

2.2 Quantization

Another problem we have to deal with is **quantization**, since the real valued function will get digital (integer) values, it is lossy. Compared to sampling which lets us reconstruct the original function. Simple quantization uses equally spaced levels with k intervals.
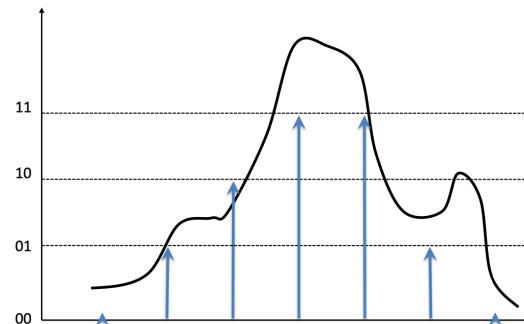
2.3 Image Properties

Image resolution is divided into two parts:

- Geometric Resolution: How many pixels per area
- Radiometric Resolution: How many bits per pixel

2.4 Noise

When taking pictures we can almost always encounter some noise. A common way to model this is additive gaussian noise:

$$I(x,y) = f(x,y) + c, \qquad c \sim \mathcal{N}(0, \sigma^2)$$

The signal to noise ratio (SNR) is an index of image quality:

$$\text{SNR} = \frac{F}{\sigma}, \qquad F = \frac{1}{XY} \sum_{x=1}^{X} \sum_{y=1}^{Y} f(x,y)$$

The usefulness for this metric can vary drastically depending on the type of image (dark images will have a higher SNR compared to bright images). Therefore we introduce peak SNR:

$$\text{PSNR} = \frac{F_{\max}}{\sigma}$$

3 Segmentation

Image segmentation is often viewed as the ultimate classification problem, once solved, computer vision is solved. A complete segmentation of an image is a finite set of disjunct regions $R_1, ..., R_n$, such that $I = \bigcup R_i$.

3.1 Thresholding

Thresholding is a simple segmentation process, that produces a binary image by labelling each pixel in or out of the region of interest. We do this by comparison of the grey level with a threshold value T. Another, better approach can be chromakeying. Hereby we measure the distance from a defined color g:

$$I_\alpha = |I - g| > T$$

One limit of thresholding is that it does not consider image context.

3.2 Segmentation Performance

If we want to choose the best performing segmentation algorithm or determine a good value for T, we need a performance metric. To use automatic analysis, one needs to know the true classification of each test, for this the test images have to be segmented by hand.

One performance metric is the ROC curve. This curve characterizes the error trade-off in binary classification tasks, by plotting the true positive fraction against the false positive fraction. We often choose the operating point on the ROC curve, by assigning cost and values to each outcome:

- V_{TN} - value of true negative
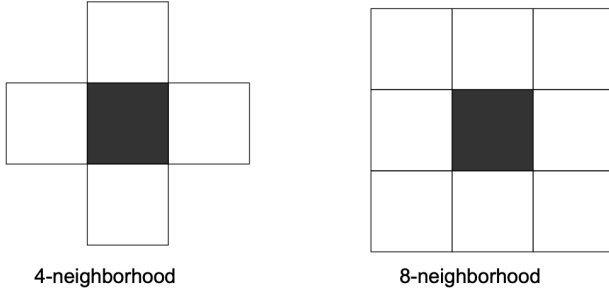- V_{TP} - value of true positive

- C_{FN} - cost of false negative

- C_{FP} - cost of false positive

We then choose the point on the ROC curve with the gradient:

$$\beta = \frac{N}{P} \cdot \frac{V_{\text{TN}} + C_{\text{FP}}}{V_{\text{TP}} + C_{\text{FN}}}$$

3.3 Pixel Connectivity

We try to define which pixels are neighbors.

4-neighborhood   8-neighborhood

A 4 (or 8) connected path between p_1, p_n is a set of pixels such that every p_i is a 4 (or 8) neighbor of p_{i+1}. Now we can defined a region as 4 (or 8) connected if it contains a 4 (or 8) connected path between any two of its pixels.

With this we can introduce **region growing**. We start from a seed point or region and add neighboring pixels that satisfy the criteria defining a region until we can include no more pixels. There are different approaches to selecting the seed and we could also use multiple seeds. For the inclusion criteria we could choose thresholding or a distribution model.

Another criteria is a snake (active contour). While each point along the contour moves away from the seed, it always has to have some smoothness constraints (minimizing energy function).
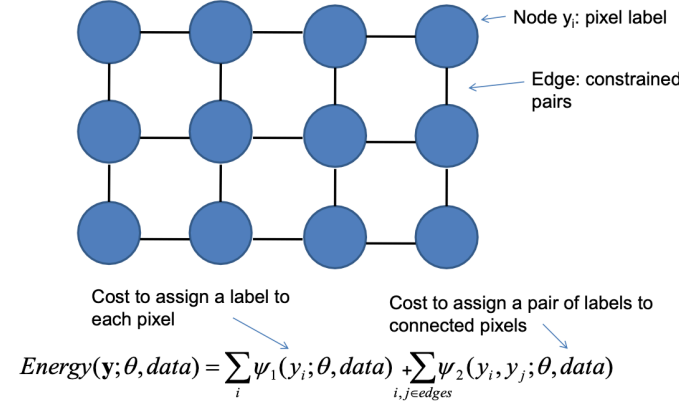
3.4 Distance Measures

Plain background substraction $I_\alpha = |I - I_{bg}| > T$, where I_{bg} is the background image, we get this by fitting a Gaussian (Mixture) model per pixel. Even better would be:

$$I_\alpha = \sqrt{(I - I_{bg})^\top \Sigma (I - I_{bg})} > T$$

Where Σ is the background pixel appearance covariance matrix.

3.5 Markov Random Fields

We can add spatial relations with Markov Random Fields (2D Markov Chains).

← Node y_i: pixel label

← Edge: constrained pairs

Cost to assign a label to each pixel

Cost to assign a pair of labels to connected pixels

$$Energy(\mathbf{y}; \theta, data) = \sum_i \psi_1(y_i; \theta, data) + \sum_{i,j \in edges} \psi_2(y_i, y_j; \theta, data)$$

Using a graph cut algorithm we can determine the optimal segementation. We can further optimize this by using iterated graph cut and k-means for learning the colour distribution (GMM) of the image.

Iterated graph cut