Laboratory Experiments

Signal Processing and Analysis

Sheet of light laser

2-D image sensor

Scanning in X-dimension
Outline

1. Demodulation and Sound Processing
2. Image Processing
3. Image Analysis and Classification
4. 3D Imaging
5. Camera Calibration
1. Demodulation and Sound Processing

1.1. Introduction
This first experiment is all about getting practical experience in filter design. Filters are one of the most crucial components for a radio receiver and that is why the superheterodyne radio receiver is selected as our design case. Figure 1 depicts a schematic sketch of the principle for such a receiver. The data files that you have got for this project contains the antenna input signal sampled with a frequency of 6 MHz. Figure 2 shows an example of the amplitude spectrum of the antenna input signal. In this example, there are obviously 3 radio stations on the air. The maximum bandwidth for the received output Low Frequency signal is 15 kHz. The receiver is tuned to the different stations by adjusting the frequency of the local oscillator.

![Figure 1. A superheterodyne radio receiver](image)

The radio stations shown as 3 peaks in Figure 2 are amplitude modulated. The principles for amplitude modulation are shown in Figure 3. The spectra of the input LF signal B is shifted
and added to the spectra of the carrier signal $A$. The resulting spectrum for the amplitude modulation is shown in $C$. This shift in frequency can easily be explained by the simple trigonometric relation also shown in Figure 3.

\[
A(t) = \cos(\omega_C t)
\]

\[
C(t) = A(t) + kA(t)B(t)
\]

\[
\cos A \cos B = \frac{1}{2}(\cos(A + B) + \cos(A - B))
\]

**Figure 3. Principles for amplitude modulation.**

A plot in the time domain of an amplitude modulated signal is shown in Figure 4A where the envelope of the carrier clearly reveals the shape of the LF signal. Figure 4B shows the first step of the demodulation process where the signal from $A$ has been rectified by calculating the absolute values. Figure 4C shows signal from $B$ after low pass filtering.

**Figure 4. Amplitude demodulation.**

**Task 1**
- Investigate the supplied script for an IIR filter in MatLab. It is a direct Signal Flow Graph type II described in file `filterIIR2t.m`. This model is a function with an input data vector and the coefficients as function parameters.
- Calculate the filter coefficients for an IIR filter specification of your own choice.
- Write a “testbench” in Matlab that verifies the specified amplitude- and phase transfer functions of your calculated IIR filter. Make sure that the output plots are nice, have titles and names and units on all axis!

**Task 2**
The following steps are a suggestion for a suitable workflow for modelling of the radio receiver depicted in Figure 1.
- Create a Matlab script that reads and analyses the RF spectrum of the antenna signal stored in the simulation input data files available on course Homepage. You should start with the smaller file.
radioA.mat for antenna signal. The sampling frequency is always 6 MHz. This is the start of your Matlab radio receiver model.

- Design a band pass filter for the IF signal of the receiver. Use the test bench from previous task for testing your filter design to work according to your specification. A suitable IF signal is approx. 450 kHz.
- Include the IF filter into the receiver model. Write the code for generation of a local oscillator and the mixing (multiplication) of the local oscillator with the antenna RF frequency. Analyse the spectrum of the IF signal. Make sure that by selecting different frequencies of the local oscillator, you should be able to tune and select one out of three radio stations.
- Design a low pass filter for the amplitude demodulator. The maximum bandwidth for the received output Low Frequency signal is 15 kHz.
- Use the test bench from previous lab for testing your LF filter design to work according to your specification.
- Include your LF filter design into the radio receiver model together with the rectification (absolute value) of the IF signal to form the amplitude detector.
- You should now be able to run your receiver model and depending on the selected frequency of the local oscillator be able to see three different LF waveforms (time domain) of one out of three radio stations. Those three different waveforms can be viewed using the smaller sized data file.
- When you have successfully been able to plot and view those three different waveforms, then you can continue with using the larger simulation data input file, radioB.mat. You’ll find both simulation files on course Homepage. The larger simulation input file will need several minutes (5 minutes or more depending on computer) to be processed.
  - Let’s say that vector lf contains low frequency data (sound) sampled with 6 MHz. Use the following commands to play this data:
    - `sf = resample(lf,44100,Fs)';`
    - `P = audioplayer(sf, 44100);`
    - `play(P);`
  - The first instruction down samples data from sampling frequency Fs to 44100 Hz. Depending on how you set the frequency for the local oscillator, you will be able to listen to three different sound clips with noisy music. You can replay the same clip by simply retyping `play(P)` as many times as you want to enjoy the music.
2. Image Processing

2.1. Introduction
Image processing is a large area of science and also an active area of research. We can only get minor insight in these technologies by a single lab experiment. We are going to focus on the pre-processing and segmentation of coins depicted in front of a light background. The last task also includes post-processing of segmented image using morphological filtering/analysis.

2.2. Summary of provided images
- L3S1T1.jpg is a picture of a test object for analysing resolution
- L3S1T2.jpg is a noisy image of coins
- L3S1T3.jpg is another noisy image of coins

2.3. List of useful Matlab commands
The following list of commands for Matlab is a hint on how to solve listed tasks.
- imfilter
- imread
- imshow
- rgb2gray
- graythresh
- im2bw
- imhist
- edge
- imopen
- imclose
- bwlabeln
- regionprops

2.4. List of tasks

Task 1
- Pre-process the image 'L3S1T1.jpg' with a filter that computes the mean value of a 3-by-3 pixel window function (2D convolution). What happens with image resolution?
- Compute and visualize the filter’s 2D amplitude transfer function using fft analysis. For visualisation, use Matlab function mesh. What kind of filter, lowpass-, highpass-, bandpass- or bandstop filter? Explain and analyse.

Task 2
- Pre-process the image 'L3S1T2.jpg' for reduction of image noise using your filter. Compute and visualise global grayscale histograms before and after filtering of image.
- Analyse histograms and experiment on finding a gray-level threshold that generates accurate regions for the coins. Do these experiments both for the filtered and non filtered images?
- Repeat thresholding on both filtered and non filtered images but now with automatic selection of threshold levels using graythresh.
- Which combination of pre-processing and thresholding do you think generates the most accurate result?

Task 3
- The best method from previous Task 2, try it also on image L3S1T3.jpg. Explain the problems that you are now facing.
- Use Matlab function edge to find as accurate as possible the edges of coins. You should use and compare different methods for computing image gradients.
- Which method for edge detection gives the most accurate description of coin edges and at the same time least amount of spurious edges?
**Task 4**

- Apply morphological postprocessing for the removal of spurious image components. Use the best result either from Task2 or Task3 as input for these morphological operations.
3. Image Analysis and Classification

3.1. Introduction
Image analysis include the task of recognising objects in images. Imaged objects are described by mathematical descriptors computed such that they capture information about the object’s descriptive features. Such features include: shape, colour, texture and more… The application we are going to investigate in this lab is OCR which is explained in Lecture 7, Image Analysis III. Lecture 8 explains different methods of supervised learning and classification.

3.2. Summary of provided images and scripts
- 26 images of hand written letters, letters/class1.bmp to letters/class26.bmp
- Two images of hand written letters for testing of OCR, test_a_o.bmp and test_o_z.bmp.
- 26 images of machine printed letters, machine/class1.bmp to machine/class26.bmp
- Matlab script for feature data computation, extractFeatures.m
- Other supporting scripts, calc_local_hist.m, mySobel.m

3.3. List of useful Matlab commands
The following list of commands for Matlab is a hint on how to solve listed tasks.
- pca
- plot3
- load
- knnclassify
- patternnet
- train
- net

3.4. List of tasks

Task 1
- Read images and compute feature data of hand written letters. Use supplied Matlab script extractFeatures.m which will create a data file featureData.mat.
- Create another script, analysePCA.m that reads featureData.mat into the Matlab workspace. Make sure that two variables, FD and CL are created in workspace. The matrix FD contains a feature vector of length 21 for each analysed letter. CL contains class labels from 1 to 26 and numbered in accordance with image file names, one image for each class of letters. How those 21 features are computed is described in Lecture 7 about OCR.
- Compute PCA on all feature data FD and create a score plot using three components. Make one plot showing all letters in one single plot and make another plot for three different classes of letters.

Task 2
- Create another script, analyseKNN.m that reads feature data FD and class labels CL. Write Matlab code that divides the feature vectors and class labels into two separate data sets, trainD, trainC, testData and testData.
- Include a KNN-classifier in your script using the training set for supervision of classifier and the test set for making an analysis of classification success rate. How does the K-value affect classification success rate?

Task 3
- Create another script, analyseNN.m that perform classification of letters, similar to previous task but now using a neural network. How does the number of neurons in the hidden layer affect the classification success rate?

Task 4
- Create another script that can read a sequence of hand written characters from an image and generate a corresponding string of decoded characters. Use the classifier that gave the best result in previous tasks.
4. 3D-Imaging

4.1. Introduction

You are now going to practice 3D surface acquisition using laser triangulation. A principle sketch of geometry is shown in Figure 5. An object of height \( r \) is reflecting light generated by a laser onto the focal plane array at position \( d \). An example of a Gaussian shaped light intensity profile generated by a laser source is shown in Figure 6. An elementary image processing operation is to find out the spatial position of the reflected line. Position sensing of such a light reflection can be accomplished at higher resolution than one pixel size. Center Of Gravity (COG) is an algorithm that can be used to weight position by its light intensity. Thresholding at level \( Th \) as shown in Figure 6 means that only those pixels where light intensity is greater than level \( Th \) are included into the COG computation. The reason for this thresholding is to exclude those pixels that mostly contribute with noise. The position \( d(c) \) of a laser line at image column \( c \) is defined as,

\[
d(c) = \frac{\sum_{r=\beta}^{\beta+B} r \cdot I(r,c)}{\sum_{r=\beta}^{\beta+B} I(r,c)}, \quad 1 \leq c \leq \text{NoOfColumns}
\]

This position \( d(c) \) can then be used to compute the corresponding height \( r \) of an object by means of triangulation,

\[
r = H \left(1 - \frac{(F_{2D} - D \tan(\beta)) \cdot (F_{2D} \tan(\beta) - d)}{(F_{2D} \tan(\beta) + D) \cdot (F_{2D} + d \tan(\beta))}\right)
\]

Figure 5. Principle sketch for laser triangulation.

Figure 6. Light intensity profile

In this set of laboratory exercises you are firstly going to experiment on generating a Gaussian shaped light intensity profile using Matlab and then evaluate methods for computation of COG including selection of threshold, Task 1. Secondly you are going to apply your signal processing on a sequence of real-world images generated from laser scanning the surface of a piece of toilet paper, Task 2.
4.2. Imaging system specification

The following data applies for the image acquisition system shown in *Fel! Hittar inte referenskälla.* that has been used to capture provided images.

- Used monochrome CCD camera: uEye2250 M
  - Number of pixels: 1600 x 1200 (Images are cropped to 1500 x 500)
  - Pixel size: 4.4 x 4.4 μm
- Structured linear shaped illumination sources – semiconductor near infrared pulsed laser (wavelength 0.9 μm)
- Motorized translation stages with motion controller driver and GRIB USB interface,
- Scanning step size when acquiring images is 0.5 mm.
- Height scale is 44 μm/pixel
- Focal length is 25 mm
- Entry pupil diameter is 12 mm
- Distance to sample is 250 mm
- Camera angle $\beta$ is 60 degrees
- Laser line length observed by camera is 65 mm

4.3. Summary of provided images

- 100 images of laser profiles, Cam1_2015-04-17_00000.tif to Cam1_2015-04-17_00099.tif

4.4. List of tasks

**Task 1**

- Write a Matlab function that computes COG (Center Of Gravity) in vertical direction across an imaged laser line and for all image columns along the horizontal direction. This function takes an RxC matrix (image) as input.

**Task 2**

- Create a MatLab script that reads provided 100 laser profile images for the generation of a range data using COG. Assume a linear height response of the scanner and use given calibration data.
- Visualize the generated range data as a grayscale image. It is preferable to stretch range data such that dynamic range of data is displayed with maximum contrast (without saturation). You need to rescale one dimension of range data to be able to display range as grayscale with both spatial dimensions having the same scale. Can you recognize the decorative structure shown in Figure 7?
- Visualize range data as a 3-dimensional mesh plot. Make sure that all axes have correct scales and units assigned to the plot.

![Figure 7. A piece of toilet paper with decorative structure](image)

Copyright © B. Thörnberg
Page: 14
5. Camera Calibration

5.1. Introduction
The purpose of this experiment is to give you hands-on experience on the intrinsic calibration of a camera. Intrinsic parameters include those properties that are related to the camera itself, including focal length of optics and lens distortion parameters. You will be given a set of calibration images captured using a stereo camera called DUO MLX [https://duo3d.com/](https://duo3d.com/). This camera was chosen because of its large field of view and clearly visible distortion. See Figure 8. You will use an open source toolbox for the calibration and you will evaluate and quantify distortion with and without calibration. Finally, you will write your own script for un-distortion of images.

Figure 8. Calibration target with added indication of origin and axes.

5.2. Summary of provided images and scripts
- 8 images of calibration target, duo1.bmp to duo8.bmp
- Matlab script for evaluation of calibration, reverseTest.m
- Other supporting scripts, undistortCoord.m, getDLT.m and calibration.m

5.3. List of tasks

**Task 1**
- Download the Matlab toolbox from California Institute of Technology [http://www.vision.caltech.edu/bouguetj/calib_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/)
- Copy the downloaded bundle of script files into a separate directory. Set the path to this directory in supplied script file, calibration.m
- Study the tutorial carefully on how to use this tool

**Task 2**
- Perform a calibration of the DUO MLX camera (only one of the two cameras) using images duo1.bmp to duo8.bmp. The calibration target was measured with a ruler to have rectangles of size \((dX, dY) = (48, 47) \text{ mm}\).
  The lens of the camera has a lot of distortion and you will most likely need to do the calibration in two steps: (1) Firstly, perform calibration using a smaller part of the calibration target. (2) Secondly, use the computed distortion coefficients as an “initial guess” when performing a second calibration on whole target.
- Save the coordinate data of calibration target corners into files duo1.txt to duo8.txt. Use the output format according to Willson and Heikkila.
- Save the result of calibration. A script file Calib_Results.m will be generated. Rename this file into Calib_Results_duo.m to make sure it is not accidently overwritten.

**Task 3**
- Make an evaluation of the distortion correction by analyzing residuals of back projected coordinates both with and without correction. A script, reverseTest.m has been given for this task. This script reads coordinate data from duo1.txt - duo8.txt and calibration coefficients from Calib_Results_duo.m.
Task 4
- Write your own Matlab script that reads one image and correct it for lens distortion according to the intrinsic calibration parameters given in *Calib_Results_duo.m*. The distortion model is explained in lecture 10.
6. Instructions for reports and oral presentations

You must prepare 15 minutes oral presentations (including questions) for each one of the five labs, plan for 10 minutes presentation and 5 minutes for questions. These presentations must have a disposition corresponding to a full technical report. This means that you need to give an introduction, define the problem, present methods and results, as well as also analyze the results, draw conclusions, and point out possible alternative methods. A seminar will be held at the end of the course, where all lab groups will give their oral presentations. All lab group members must participate actively in the presentations. All members must also be prepared to answer questions and defend your work and results. The oral presentations must be based on PowerPoint slides. The first slide must have the names of all lab group members written on it. PowerPoint slides for all five experiments together with MatLab code must be sent to teacher the day before the seminar. All lab groups will present one lab out of five possible. Which one of the five labs, will be decided by the teacher at the day of the seminar and at the last minute. This means that you need to be prepared to present any of the five labs. All students must attend during whole time of the seminar.

Listed authors solely own the proprietary rights to copy and distribute this document as whole or in parts. Students are not granted any right to copy figures or text from this document.