Master Thesis

- Implementation of stereo vision engine -

Project Report

Group 1072

Aalborg University
Electronics and IT

Here you can write something about which tools and software you have used for typesetting the document, running simulations and creating figures. If you do not know what to write, either leave this page blank or have a look at the colophon in some of your books.??

AALBORG UNIVERSITY
STUDENT REPORT

Electronics and IT
Aalborg University
http://www.aau.dk

Title:
Stereo vision implementation??

Theme:
Master Thesis??

Project Period:
Spring Semester 2016

Project Group:
1072

Participant(s):
Tomas Brandt Trillingsgaard

Supervisor(s):
Peter Koch

Copies: 4

Page Numbers: 27

Date of Completion:
June 8, 2016

Abstract:

Here is the abstract

Contents

Todo list

Preface

Here is the preface. You should put your signatures at the end of the preface.

Aalborg University, June 8, 2016

Tomas Brandt Trillingsgaard
<ttrill10@student.aau.dk>

Chapter 1

Introduction

In this chapter, the project is introduced and motivated. Furthermore, a brief description is presented for stereo vision and the use for it at HSA systems . Lastly, this chapter also describes a delimitation of the project and report. Måske anden formulering

1.1 Stereo vision introduction

In 280 A.D the greek mathematician Euclid described the perception [2]
Human has the incredible ability of depth perception. This is due to our two eyes which are separated a bit from each other. Since the eyes are separated they each receive different images. These images are combined in the brain and enable us to perceive depth. This is shown on figure 1.1.
This concept can be used in computer system and enable a system to perceive depth and hence distinguish between different objects.

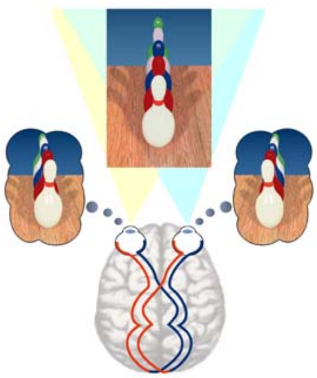

Figure 1.1: Example of human stereo vision [1]

Use of stereo vision:
Giving the ability of distinguishing between objects to a computer system gives the system the ability to perform more task. These task includes counting number of people entering pass through a secure door, enables a robot arm to interact with different objects.

HSA systems wish to keep an eye on packages going through their system. A strategically placed stereo vision camera will enable them to know how many and where these objects are in the system.

1.2 Motivation

Stereo vision algorithms usually are very heavy computational wise. A high resolution real-time stereo vision can be hard to acquire.

1.3 Problem Introduction

HSA systems wish to keep an eye on packages going through their system. A strategically placed stereo vision camera will enable them to know how many and where these objects are in the system. The primary objectives of this is to:

- Analyze obstacles within stereo vision
- Analyze different stereo algorithms
- Design and optimize an architecture for executing stereo vision

1.4 Delimitation

This project is mainly concerned with the design and implementation of a hardware design for a FPGA. This project will not focus on developing a new stereo vision algorithm. Obstacles and issue with stereo algorithms will not be

1.5 Report Structure and Design Process

The A3 methodology is a way to handle a system design. Figure 1 shows a diagram of the A3 method. As seen it consist of 3 spaces: Application, Algorithm, and Architecture. The report will follow this structure where chapter 2: Application Analysis will explore the Application space and chapter 3: Requirements will contain a specification for moving into the Algorithm space. Chapter 4: Algorithm Analysis will explore the algorithm space with the requirements as constraints. The chapter will conclude in the choice of an algorithm to be implemented on the

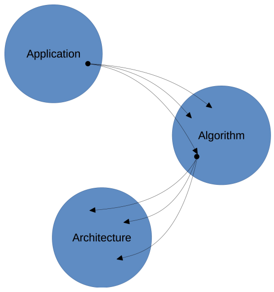

Figure 1.2: A3 model

hardware. Chapter 6: Design methodology will describe different methods which can be used to move from the algorithm space to the architecture space. Chapter 7: Architecture Design will explore the architecture space based on the chosen algorithm. The chapter will result in a implementation of the design on the hardware platform.

Chapter 2

Application Analysis

This chapter starts by describes the basic principles of stereo vision then different aspects such as color versus gray scale etc are analyzed. ikke færdig

2.1 basic principal of stereo vision

A stereo vision setup normally consists of two cameras placed horizontally a bit from each other. An example of this is on figure 2.1

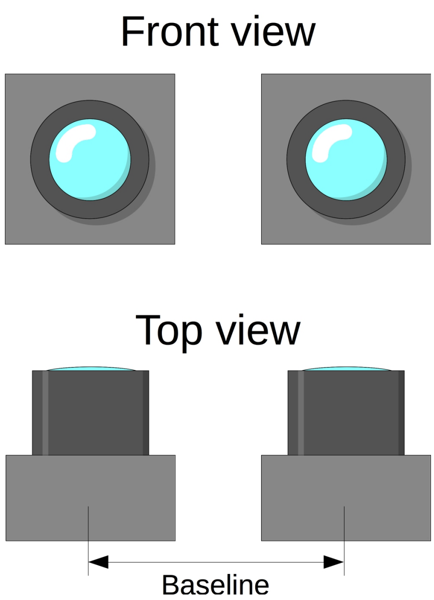

Figure 2.1: Illustration of two cameras

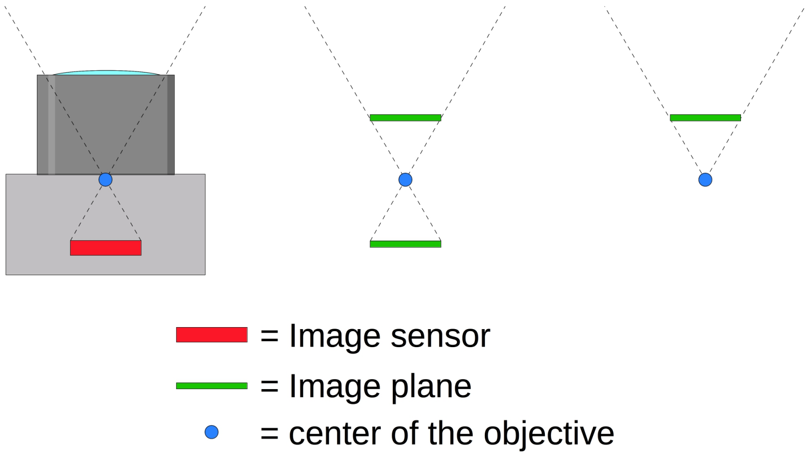

Figure 2.2: Illustration of going from camera to image plane

Missing figure

Figur af punkt ude i 'scenen'

Missing figure

Figur af beregning af disparitet

2.2 Color space and gray scale

skriv noget om forskellige farve rum og grayscale og deres indflydelse på stereo algorithmen

The article **Color correlation-based matching** takes the subject of difference in result when using color and which color space is used and grayscale when performing stereo matching. It performs different methods / algorithms using 9 different colorspace including grayscale. The result from the article is that color gives a better result with a few percentage of more correct estimations but the run time is much higher (ranging from 1.9 to 3.7 higher run time than grayscale on the teddy test set). From this it is decided to not use color in case of Normalized Cross Correlation

2.3 Resolution and disparity precision

skriv noget om disparitets opløsning i forhold til billede opløsning osv.

2.3.1 Occlusion filling

skriv noget om metoder til at udfylde occlusions områder

This section will describe methods for filling the occluded areas. All these methods comes from the article: *Occlusion filling in stereo: Theory and experiments* by *Shafik Hyq*, *Andreas Koschan* and *Mongi Abidi*. All these methods assume that the stereo matching is going from left image to right image i.e. templates are taken from the left image matched onto the right image.

Neighbor's Disparity Assignment : NDA

This is the simplest method to fill occlusions. It functions by selecting an occluded point, p_L, then find then nearest non-occluded point, q_L, to the left when filling non-border occlusion. With border occlusion the nearest point to the right is found instead. It is assumed that this non-occluded point is part of same surface as the occluded point (this can be seen on figure **??**) and the disparity value from q_L can be assigned to p_L. This method have some issues. In cases of total occlusions (see figure **??**) then a wrong disparity value is given to the total occluded object since it isn't a part of the nearest surface with non-occluded points to the left. In cases with self occlusions the occluded area should have disparity values close to the disparity values of the non-occluded points to the right (This will be the area of the surface which is in view of both cameras) but using NDA will give the occluded area disparity values corresponding to the background.

Diffusion in Intensity Space : DIS

This method is inspired by diffusion. Diffusion is the movement of molecules or atoms from a high concentration region to a low concentration region.
After detecting occluded regions with cross-checking during template matching, the diffusion energy for the region is approximated. This method is depended on the stereo matching algorithm because it use the energy from the last iteration to determine initial diffusion energy for the area.
A change to the method can be made to make it independent from the stereo matching. The initial energy will be 0. Then the diffusion energy for non-border

occlusion is found by:

$$E(p_L) = \min_{l_{p_L}=\{0,\ldots,l_{max}\}} \left(\frac{1}{2|q_L \in \mathcal{N}(p_L) \wedge l_{q_L=l_{p_L}}|} \sum_{q_L \in \mathcal{N}(p_L) \wedge l_{q_L=l_{p_L}}} (|\bar{I}(p_L) - \bar{I}(q_L)| + E(q_L)) \right) \tag{2.1}$$

And the diffusion energy for border occlusions are found by by:

$$E(p_L) = \min_{l_{p_L}=\{0,\ldots,l_{p_{Lf}}-2\}} \left(\frac{1}{2|q_L \in \mathcal{N}(p_L) \wedge l_{q_L=l_{p_L}}|} \sum_{q_L \in \mathcal{N}(p_L) \wedge l_{q_L=l_{p_L}}} (|\bar{I}(p_L) - \bar{I}(q_L)| + E(q_L)) \right) \tag{2.2}$$

The diffusion energy will be calculated for each occluded point and for each point the disparity which corresponds the minimum $E(p_L)$ is set as the disparity l_{p_L} for the occluded point.

Weighted Least Squares : WLS

In this approach, WLS, all the non-occluded and filled occluded neighbors in a neighborhood around the occluded point is considered valid points and is used as control points in interpolation.
Since the neighborhood contains both foreground points and background points and the occluded point is expected to be a part of the background then the background points should have more influence than foreground points. It is assumed that the color intensity between objects is significantly different and this property can be used to distinguish between foreground points and background points.
Each error term in the aggregated residual should be weighted so the foreground don't have much influence. With this the aggregated residual is defined as:

$$\Delta = \sum_{q_L \in \mathcal{N}(p_L)} w_{q_L} (\hat{l}_{p_L}(p_L) - l_{p_L}(q_L))^2 \tag{2.3}$$

where $w_{q_L} = e^{-\mu_L |\bar{I}(p_L) - I(q_L)|}$ (the weight) is the likelihood of p_L with q_L under the assumption of an exponential distribution model of $|\bar{I}_(p_L) - I(q_L)|$. $\bar{I}(p_L)$ is the mean intensity of p_L and μ_L is the decay rate. $\hat{l}_{p_L}(p_L)$ is the estimated disparity of p_L (will be estimated during interpolation) and $l_{p_L}(q_L)$ is the disparity of q_L.
How to estimate $\bar{I}(p_L)$ and μ_L:
$\bar{I}(p_L)$ is the mean intensity of p_L which can be obtained using mean shift algorithm in a window around p_L. To estimate this value the initialize the algorithm with $\bar{I}(p_L)$ equal to the intensity of p_L then the mean shift algorithm repeatedly picks those neighbors inside the window that satisfy $|\bar{I}(p_L) - I(q_L)| \geq 3\mu^{-1}$ and the assign the average of intensities of the selected neighbors to $\bar{I}(p_L)$ until $\bar{I}(p_L)$ converges to a fixed average. $|\bar{I}(p_L) - I(q_L)|$ has decay rate μ_L which is related to

the decay rate μ of the variable $|I(p_L) - I(q_L)|$ by $\mu_L^2 = \mu$.
A matrix containing all the coordinates:

$$F = \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \ddots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \tag{2.4}$$

Vector with the corresponding labels for the coordinates in F:

$$L = [l_1 \ \cdots \ l_N] \tag{2.5}$$

Linear model:

$$l_{p_L} = a + bx(p_L) + cy(p_L) \tag{2.6}$$

Where $(x(p_L), y(p_L))$ is the coordinates of p_L and a, b and c are the model parameters.
The weights for the control points can be express in a vector as:

$$w = [w_{q_{L1}} \ w_{q_{L2}} \ \cdots \ w_{q_{LN}}]' \tag{2.7}$$

Then we compute two new matrices, F_w and L_w:

$$F_w = diag(w)F \tag{2.8}$$

$$L_w = diga(w)L \tag{2.9}$$

The model parameter vector:

$$P = [\ a\ b\ c\ ]' \tag{2.10}$$

By combining the equations above then the following equation is given:

$$P = (F_w^T F_w)^{-1} F_w^T L_w \tag{2.11}$$

With these equation the disparity of the occluded point can be estimated:

$$\hat{l}_{p_L} = [1\ x(p_L)\ y(p_L)]P \tag{2.12}$$

Segmentation-based Least Squares : SLS

Biggest difference between WLS and SLS is that SLS only uses non-occluded points as control points. The control points is a subset of the non-occluded neighboring points. The control points are segmented from the neighborhood by applying different constraints: visibility constraint, disparity gradient constraint and color similarity cues.
Sequence of operations:

- Select an occluded point

- Select control points from the neighborhood around the occluded point
- Interpolate the disparity of the occluded point from the segmented control points

$\mathcal{N}(p_L)$ is a set of non-occluded, neighboring points which will be use for control points in the interpolation. For points to be added to $\mathcal{N}$ then it needs to fulfill some constraints.

Disparity gradient constraint: In most cases the horizontal closest non-occluded point to the right, p_{Lf}, will be part of the foreground and the occluded should be a part of the background. In this cases every non-occluded point with a lower disparity than p_{Lf} will be added to $\mathcal{N}$ hence the condition for added the point, q_L, will be $l_{q_L} < l_{p_{Lf}}$. If the foreground object is narrow then all the non-occluded neighboring points might be from the background and have the same disparity. Due to this a second condition have to be added to the constraint. The horizontal closest non-occluded point to the left will be called p_{Lb} and a second condition is created: $|l_{p_{Lb}} - l_{q_L}| \leq 1$. When these conditions are combined the constraint can be defined as:

$$|l_{p_{Lb}} - l_{q_L}| \leq 1 \vee l_{q_L} < l_{p_{Lf}} \tag{2.13}$$

surface constraint: It is assumed that $\mathcal{N}(p_L)$ will contain points from maximum 2 different surfaces (due to the small neighborhood). Some cases might contain a third surface but this is expected to occur very seldom and therefore it is disregarded. The point with the lowest disparity, l_{min}, is assumed to belong to one of the surfaces and the point with the highest disparity, l_{max}, is assumed to belong to the other surfaces. If $l_{max} - l_{min} \leq 1$ then it is assumed the all the points in $\mathcal{N}$ belongs to a single surfaces otherwise the points have to be segmented into 2 groups. The first group will contain all points which satisfies $|l - max - l_{q_L}| \leq 1$ and the other group will contain all the points which satisfies $|l - min - l_{q_L}| \leq 1$.

Color constraint: The average truncated color distance from the occluded point, p_L, to each of the two groups to determine which group the point belongs to. The average truncated color distance is found by:

$$D(p_L, \mathcal{N}_i(p_L)) = \frac{1}{|\mathcal{N}_i(p_L)|} \sum_{q_L \in \mathcal{N}(p_L)} \psi(p_L, q_L) \tag{2.14}$$

Slut af med mini-konklusion på områderne / delimitation

Chapter 3

Requirements

3.1 Requirement specification

Få lavet en tabel som indeholder kravene

No.	Parameter	Value	Unit	Additional Information	Source
1	Something something	0 to 48	Mhz	• Something	1
2	Something something	0 to 48	Mhz	• Something	1
General requirements • Something something something					

3.2 Test specification

Regner med at lave en test hvor jeg med min egen python simulering trækker dataen ud lige før hvor dataen skal bruges i det jeg har fået lavet et hardware design af. så vil jeg samligne med middlebury test sets

skriv hvordan jeg vil teste de forskellige krav

beskrive middlebury test sets her? Nej beskriv dem i appendix

Chapter 4

Algorithm design

ROUGH SKETCH not done yet

In this chapter the two stereo vision algorithms, Efficient Edge Preserving Stereo Matching (EEPSM) and Fast Cost-Volume Matching (FCV), is described. Lastly, a simulation of each algorithm is created and the results of these simulations are compared and from this, an algorithm is chosen.

4.1 Efficient Edge Preserving Stereo Matching:

This algorithm works in three steps. The first step is calculating a cost for each pixel and disparity. This cost is a combination of the sum of absolute differences and hamming distance of the census transform around each pixel.

$$C_d^{SAD}(x,y) = \sum_{i=1}^{3} |I_{left}(x,y,i) - I_{right}(x+d,y,i)| \tag{4.1}$$

$$C_d^{CENSUS}(x,y) = Ham(CT_{left}(x,y), CT_{right}(x+d,y)) \tag{4.2}$$

$$C_d(x,y) = \alpha \cdot C_d^{SAD}(x,y) + (1-\alpha) \cdot C_d^{CENSUS}(x,y) \tag{4.3}$$

where d is the disparity estimate, I_{left} is the left image, I_{right} is the right image, i is the color (rgb), $Ham(x_1, x_2)$ is the hamming distance between x_1 and x_2, and CT_{left} and CT_{right} is the census transform around the specified pixel
then a permeability weight is calculated. Permeability is known from biomedicine and describes the ability to transfer through a membrane. The permeability weight is inspired by this and describes how well the color transfers from one pixel to another pixel.

$$\mu(x,y) = \min(e^{\frac{-\Delta R}{\sigma}}, e^{\frac{-\Delta G}{\sigma}}, e^{\frac{-\Delta B}{\sigma}}) \tag{4.4}$$

$$\mu_{tb}(x,y) = \min(e^{\frac{-(R(x,y)-R(x,y-1))}{\sigma}}, e^{\frac{-(G(x,y)-G(x,y-1))}{\sigma}}, e^{\frac{-(B(x,y)-B(x,y-1))}{\sigma}}) \tag{4.5}$$

lastly, the cost is aggregated resulting in a combined cost for each pixel at each disparity. The cost from equation ?? is first aggregated horizontally using permeability weights from equation ??. Then the result from horizontal aggregation is aggregated vertically also using the permeability weight.

$$C_d^{lr}(x,y) = C_d(x,y) + \mu_{lr}(x,y) \cdot C_d(x-1,y) \tag{4.6}$$

$$C_d^{lr}(x,y) = C_d(x,y) + \sum_{i=1}^{x-1} \Big(C_d(x-i,y) \cdot \Pi_{j=i}^{i} \mu_{lr}(x-1,y) \Big) \tag{4.7}$$

With a cost at each pixel at each disparity estimate, the disparity map can be generated by minimization along the disparity estimates.

4.2 Fast Cost-Volume Matching:

This algorithm starts by calculating a cost for each pixel at each disparity estimate. This cost consists of the sum of absolute differences and differences in the gradient.

$$C_d^{SAD}(x,y) = \sum_{i=1}^{3} |I_{left}(x,y,i) - I_{right}(x+d,y,i)| \tag{4.8}$$

$$C_d^{Grad}(x,y) = \nabla_x I_{left}^g(x,y) - \nabla_x I_{right}^g(x,y) \tag{4.9}$$

$$C_d(x,y) = \alpha \cdot C_d^{SAD}(x,y) + (1-\alpha) \cdot C_d^{Grad}(x,y) \tag{4.10}$$

These cost values are then filtered using a Guided Image Filter. The guided image filter is a filter uses a reference image to generate the weights. The guided image filter is described further in section 4.2.1

$$C_d'(x,y) = \sum_j W_i, j(I) C_d(x,y) \tag{4.11}$$

The correct disparity for each pixel can then be found by minimizing along the disparity estimates as seen in equation 4.12.

$$f(x,y) = \arg \min_{d \in [0, d_{max}]} C_d'(x,y) \tag{4.12}$$

4.2.1 Guided image filter

The guided image filter uses a image as a reference for weighting the input. The output from the filter is seen in equation

$$q_i = \sum_j W_{i,j}(I)p_j \tag{4.13}$$

$$q_i = a_k I_i + b_k \quad ,\forall i \in \omega_k \tag{4.14}$$

where:

$$a_k = \frac{\frac{1}{|\omega|}\sum_{i\in\omega_k} I_i p_i - \mu_k \bar{p}_k}{\sigma_k^2 + \epsilon} \tag{4.15}$$

$$b_k = \bar{p}_k - a_k \mu_k \tag{4.16}$$

algorithm

Input:
filtering input image: p
guidance image: I
radius: r
epsilon: ϵ
Output:
filtering output: q
Steps:

1. $\mu_I = f_{mean}(I)$
   $\mu_p = f_{mean}(p)$
   $\rho_{II} = f_{mean}(I \cdot I)$
   $\rho_{Ip} = f_{mean}(I \cdot p)$
2. $\sigma_I = \rho_{II} - \mu_I \cdot \mu_I$
   $cov_{Ip} = \rho_{Ip} - \mu_I \cdot \mu_p$
3. $a = cov_{Ip}/(\sigma_I + epsilon)$
   $b = \mu_p - a \cdot \mu_I$
4. $\mu_a = f_{mean}(a)$
   $\mu_b = f_{mean}(b)$
5. $q = \mu_a \cdot I + \mu_b$

4.3 Simulation and comparison

simulation af de 2 algorithmer og samlign resultaterne.

4.4 Choosing an algorithm

Nok en anden titel til denne sektion. Skriv hvilken algorithme jeg går videre med

Chapter 5

Platform Analysis

beskriv Zynq platformen. kom ind på hvad den indeholder

FPGA contraints ==> C = f(A, T, P, N), Lav en tabel

Chapter 6

Design methodology

læs om system design methodologies i gajski's **Embedded Systems Design - Modeling, Synthesis and Verification** og beskriv Platform Methodology

Chapter 7

Architecture design

RTL design

NOT DONE! rough sketch. De nedenstående trin er hvad jeg skal igennem: Para. Anal., Alloc., Optimizaiton, FSMD og VHDL + simulering

7.1 Parallelism Analysis

7.2 Allocating / Scheduling

7.3 Optimization

7.4 FSMD design

7.5 VHDL + Simulation

Box filter / Mean function

As seen from the guided image filter algorithm the mean function $f_{mean}(x)$ is used multiple q

this section should contain the design for my boxfilter or mean function

Finite State Machine

beskriv FSM'en jeg har lavet (se figur 6.1)

Memory

memory requirement:
$3 \cdot 8$ bits per pixel (rgb image). test image is 741×497 so for the test image 8.838.648 bits $\approx$ 9 megabit $\approx$ 1.1 megabyte.

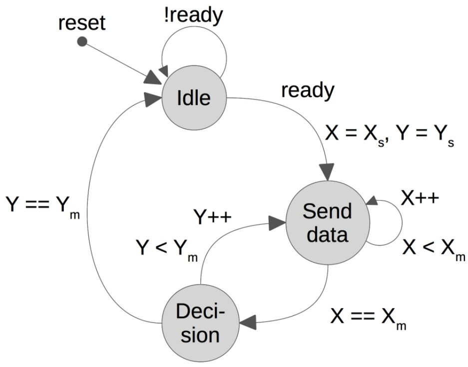

Figure 7.1: TEXT GOES HERE

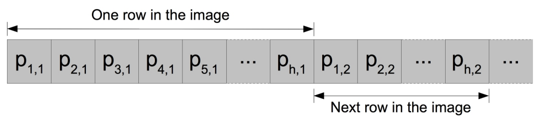

Figure 7.2: TEXT GOES HERE

VHDL/Simulation

skriv om VHDL kode og simulation af filteret

Implementation/Test

skriv om implementation på FPGA'en og gerne verificere det virker

Chapter 8

Acceptance test

udfør accept test udfra test specifikationen (brug data fra python simulering og giv det til VHDL implementationen)

Chapter 9

Conclusion

This chapter will contain the conclusion

Bibliography

[1] Optometrists network. *What is stereo vision?* http://www.vision3d.com/stereo.html. 2016.

[2] The Turing Institute. *The History of Stereo Photography*. http://www.arts.rpi.edu/~ruiz/stereo_history/text/historystereog.html. 1996.