University of Southern Denmark

Hand-eye calibration
for 3D modelling

Participants	Frederik Hagelskjær, M.Sc. student, Robot Systems Engineering The Maersk Mc-Kinney Moller Institute
	Kent Stark Olsen, M.Sc. student, Robot Systems Engineering The Maersk Mc-Kinney Moller Institute
	Leon Bonde Larsen, M.Sc. student, Robot Systems Engineering The Maersk Mc-Kinney Moller Institute
	Rudi Hansen, M.Sc. student, Robot Systems Engineering The Maersk Mc-Kinney Moller Institute
Primary supervisor	Kjeld Jensen, Ph.D. student, Field Robotics The Maersk Mc-Kinney Moller Institute
Secondary supervisor	Lars-Peter Ellekilde, Associate professor The Maersk Mc-Kinney Moller Institute
ECTS	10
Period	Spring 2014

Faculty of Engineering
University of Southern Denmark
Niels Bohrs Allé 1
5230 Odense M
Denmark

www.sdu.dk/tek
+45 6550 7303
tek@tek.sdu.dk

Abstract

Preface

Preface my ass...

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

Models are necessary in most applications of robotics. Often the models are build in advance and used by the robot as a priori knowledge, but this approach limits the application of robots in dynamic environments. Furthermore modelling complex environments can be a tedious and time consuming task, making automatic modelling desirable.

One way of recording such 3D models is by mounting a stereo camera on the end effector of a robot arm and let the robot move the camera to the views needed to generate the model. The precision of such systems is dependent on robust calibration with respect to both camera and kinematics.

The camera can be calibrated using a marker plate in different poses thus calculating intrinsic parameters and camera disparity [**Zhang1999**, 7]. The process of generating a 3D model involves calculating a disparity image from the stereo camera input. From the disparity a point cloud can be generated and point clouds from different views can be stitched together. The combined point cloud can then be filtered and used for surface- or volume reconstruction. The stitching process is dependent on the robot pose, since each point cloud must be transformed to the same frame.

The robot pose can be calibrated in a process called hand-eye calibration solving for the unknown spatial relationships in the kinematic chain. This calibration thus influences the quality of the combined point cloud.

It is hypothesised that a hand-eye calibrated system can generate significantly more precise models than the same system without calibration.

Evaluation of the 3D model is based on a known object, where the combined point cloud can be compared to the 3D model of the object. This introduces a pose estimation problem, since the model must be aligned to the point cloud.

Calibration of robot systems has received considerable attention and continues to be an active field of research. A solution for the unknown transforms from camera to end effector and from maker to robot base can be obtained relatively easy by solving a homogeneous transform [4]. Several algorithms for more or less autonomous calibration has been proposed [5, 6] often simultaneously calibrating camera and hand-eye [**Jordt2009**, 2, 8].

This work is a manipulative study investigating the effect of hand-eye calibration measured on the quality of the produced point cloud. The novelty of the study is the practical implementations of hand-eye calibration and model evaluation as well as calibration routines for the robot kinematics.

Chapter 2

Analysis

To test the hypothesis a closed loop system based on the Good Old Fashion Artificial Intelligence (GOFAI) environment interaction model [**Pfeifer2007**] was developed (Figure 2.1). In the following each block will be described with respect to functionality and interfaces.

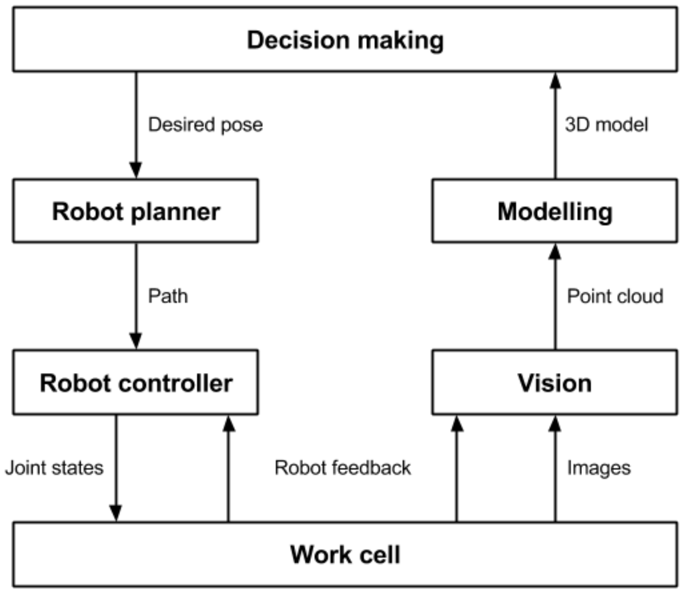

Figure 2.1: Block diagram of the eye-in-hand 3D reconstruction system.

2.0.1 Decision maker

The desired pose is 'hard-coded' into the decision maker to capture the object from for example 24 discrete locations on a sphere. The captions are uniformly distributed to cover the entire object. When the robot is at rest in the desired pose a signal is sent to the vision block to capture the current view.

2.0.2 Robot planner

The robot planner executes the desired pose based on the PRM algorithm subject to the constraint that the camera always points to the center of the object being modelled. The robot planner is based on the Open Motion Planning Library (OMPL) which is integrated into the moveIt stack in ROS and collision checking is based on the a priori model of the work cell.

2.0.3 Robot controller

The robot controller further processes the path adding kinematics and time tessellation to meet velocity and acceleration constraints. The controller interpolates the joint states and handles closed loop control with feedback from the workcell. The robot controller is based on the control stack in ROS.

2.0.4 Work cell

The workcell contains a six degrees of freedom RX60 robot arm with a stereo camera mounted on the end effector. The work cell interface is based on a ROS node communicating with the robot and a node broadcasting data from the camera.

2.0.5 Vision

The vision module takes input from the stereo camera mounted on the end effector of the robot. The images are undistorted and rectified using calibration parameters obtained independently using the ROS calibration node. A disparity image is generated using the semiglobal block matching [1], and by using the obtained projective parameters the point cloud is obtained. The vision part will be implemented with a combination of built-in ROS nodes and homemade ROS nodes using the OpenCV library for image processing.

2.0.6 Modelling

The point clouds from the vision block are transformed to a common frame based on the robot pose and are then cropped, filtered and combined into one point cloud representing the sum of information about the object. Methods from Point Cloud Library (PCL) will be used for point cloud cropping, filtering and the assembly process. A wavelet based algorithm will be implemented for surface reconstruction according to [3].

Chapter 3

Requirements

RoVi is the best course ever .... blablabla

Chapter 4

Decision making

RoVi is the best course ever .... blablabla

Chapter 5

Robotics

RoVi is the best course ever .... blablabla

CHAPTER 6

Vision

RoVi is the best course ever .... blablabla

CHAPTER 7

Modelling

The task of reconstructing an unknown surface from a point cloud is not a trivial task. Especially not when this point cloud is obtained through a camera mounted as the tool on a robot arm. To be able to reconstruct the full surface several views of the object are required, which mean that several point clouds are required to be aligned with each other and stitched together. The modelling component of this system is divided into two sub-components, the first component described in this chapter is filtering and the second component described is the reconstruction component.

7.1 Filtering

Filtering of raw point clouds is required because of several different reasons. The number of points delivered to the modelling component from the raw point clouds is huge, so filtering non-interesting points away creating a region-of-interest (ROI) in the raw point clouds. Lowering the number points in each cloud delivered to the modelling component is required such the workload can be kept within an acceptable range. The number of points can be further reduced by down-sampling the points left in the ROI by the cut-off filter. A voxel-grid filter utilised for down-sampling also creates the advantage of equal sampling density, but the disadvantage is that the down-sampling means loss of information, and therefore there is a trade off between speed and level of detail of the reconstruction process.

7.1.1 Point cloud library

The Point Cloud Library (PCL) utilised in this project is a library which provide functionality for working on 3D point clouds. PCL delivers a variety of functionality such as filters, segmentation, surface reconstruction, kd- and oc-trees, visualisation, etc. PCL can be found at http://www.pointclouds.org/, along with documentation and tutorials.

7.1.2 Point cloud transformation

Messages received from the vision layer needs to be processed before filtering. This is because the messages delivered to the modelling component contain a point cloud and a pose of the current camera view. The coordinates of the individual points in the cloud are related to the camera frame, but this frame is moving around the object so a transformation of points is needed such they can be related a common static frame.

7.1.3 Cut-off filter

The cut-off filter utilised is the implementation from PCL, `pcl::PassThrough< ...>`. A cut-off filter is utilised to create a ROI in the transformed point cloud, partly because the

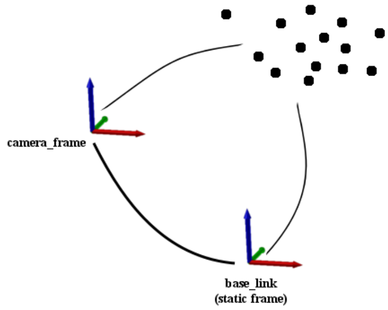

Figure 7.1: Illustrates points in different frames.

number of points needs to be reduced with respect to processing time and then because the region in which the object resides is fairly small to the region which is recorded by the camera.

7.1.4 Voxel-grid filter

The voxel-grid down-samples the left overs from the cut-off filtering. The down-sampling causes loss of information which mean that surface details are lost in the process, so the amount of down-sampling should be chosen with respect to processing time versus level of detail to be reconstructed. The PCL library luckily have such functionality (`pcl::VoxelGrid<...>`) which is utilised in the filtering sub-component.

The voxel-grid filter works by splitting down the ROI into smaller regions (voxels) of certain resolution in which each of the voxels are analysed. Figure 7.2 show the principle of the voxel-grid. A new point is approximated for each voxel, the new point is approximated by the points centroid which is contained in the voxel. This method is a little slower compared to just placing the new point in the center of the voxel, but it helps save some more detailed information about the surface curvature.

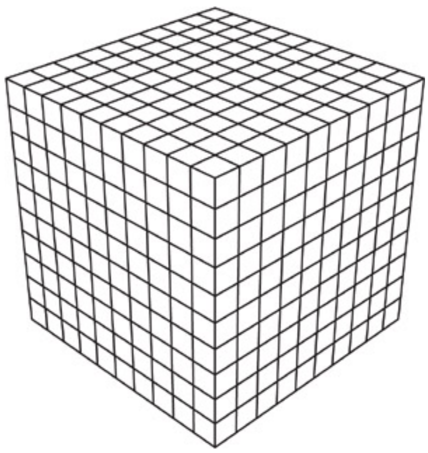

Figure 7.2: Illustrates a cube consisting of 10x10x10 cubes which resembles the voxel-grid filter.

7.1.5 Point cloud stitching

7.2 Surface reconstruction

This is fun....

7.2.1 Related work

Surface reconstruction from a large set of unstructured points obtained through laser scanning techniques or stereopsis is not a trivial task but nonetheless it is useful in many applications. For example a surface reconstruction could aid a robots end effector grasping some unknown object[1]. Many methods for surface reconstruction exist in different domains of science, some algorithms are based on neural network approaches [2], others are based on sculpting or region growing algorithms (e.g [3] and [4]) used in computational geometry, some utilise an implicit method framework to represent incoming data as a surface[5] and [6].

In 1999 F. Bernardini et al. proposed in a fairly simple algorithm (Ball Pivoting Algorithm (BPA)) for reconstruction of surfaces from point clouds sampled over smooth objects[3]. BPA is pivoting a sphere of a certain diameter around an edge of a seed triangle. Pivoting the sphere around all the edges is connecting three points to form a new triangle and so on. BPA is a part of the region growing algorithms as this algorithm uses a seed triangle builds the surface around this and outward. The algorithm is fairly easy to work with as it has one parameter which decides the radius of the sphere and that is it. N. Amenta et al. proposed in 2001 [4] the power crust algorithm, which essentially is a three dimensional Voronoi approach[7]. The power crust algorithm is well defined and proven which makes it one of the most known algorithm regarding surface reconstruction. This algorithm is a sculpting algorithm in computational geometry. Common for those algorithms mentioned is that these are explicit methods which requires neighboring information which leads to high computational time consumption. Therefore methods mentioned above are not well suited for reconstruction of large data sets.

In 2006 M. Kazhdan et al. proposed in [5] an algorithm which resides in the implicit method framework, this algorithm is called Poisson Surface Reconstruction (PSR). PSR is a fitting scheme that allow solving for the indicator function of the surface. It is shown in [5] this approach of fitting a surface resembles the Fast Fourier Transform (FFT). In [5] It is shown that the FFT approach requires five times as much space but is approximately double as fast while creating approximately the same number of triangles. The real advantage of the Poisson approach is that it is scalable and therefore does not rely on a uniform distribution of points and thus a higher degree of details can be reconstructed in areas where the point density is higher.

J. Manson et al. did in 2008 propose a wavelet approach of the problem of reconstructing surfaces of large sets of unstructured points [9]. The method is robust to noise in data because it is relying on implicit methods as PSR, and is therefore well-suited for reconstruction of real data which is overlayed with some random noise. The wavelet approach an advantages over PSR, the wavelet approach is able to reconstruct non-closed surfaces in opposition to PSR[8].

Using the Haar wavelet synthesise an acceptable surface if the points are well-aligned, else more smooth wavelets can be used such as Daubechies-2. Using octrees and small-support wavelets makes the synthesisation very fast, smaller support equals less computational time and space. Using Daubechies-2 compared to Haar increases computational time by a scale of four, and the space consumption is upscale by approximately four as well.

As the methods for reconstruction mentioned first does not guarantee watertight mesh and does not work very well with large point sets, mean those methods are not of interest for further studies. The PSR and the wavelet approach both does estimate an indicator function of the surface, but each handles this indicator function differently. As these method is based on implicit methods they are more robust to noise in the input and approximates the surfaces better compared to the methods mentioned first.

Chapter 8

Results and Discussion

RoVi is the best course ever .... blablabla

Chapter 9

Conclusion

RoVi is the best course ever .... blablabla

Bibliography

[1] Heiko Hirschmüller. "Stereo processing by semiglobal matching and mutual information." In: *IEEE transactions on pattern analysis and machine intelligence* 30.2 (2008), pp. 328–341. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2007.1166.

[2] Henrik Malm and Anders Heyden. "Simplified intrinsic camera calibration and hand-eye calibration for robot vision". In: *Intelligent Robots and Systems, 2003.(...* October (2003). URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1250764.

[3] J. Manson, G. Petrova, and S. Schaefer. "Streaming Surface Reconstruction Using Wavelets". In: *Computer Graphics Forum* 27.5 (2008), pp. 1411–1420. ISSN: 01677055. DOI: 10.1111/j.1467-8659.2008.01281.x. URL: http://doi.wiley.com/10.1111/j.1467-8659.2008.01281.x.

[4] YC Shiu and S Ahmad. "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX= XB". In: *Robotics and Automation, IEEE ...* 5 (1989), pp. 16–29. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=88014.

[5] RY Tsai and RK Lenz. "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration". In: *Robotics and Automation, IEEE ...* 5.3 (1989). URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=34770.

[6] RY Tsai and RK Lenz. "Real time versatile robotics hand/eye calibration using 3D machine vision". In: *Robotics and Automation, 1988. ...* (1988), pp. 554–561. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=12110.

[7] Zhengyou Zhang. "A flexible new technique for camera calibration". In: *Pattern Analysis and Machine Intelligence, IEEE ...* 1998.11 (2000), pp. 1330–1334. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=888718.

[8] Zijian Zhao and Yuncai Liu. "HAND-EYE CALIBRATION INTO ROBOT VISION". In: 2008, pp. 5721–5727. ISBN: 9781424421145.

Appendix A

Sample appendix

RoVi is the best course ever .... blablabla