Lehigh University
Lehigh Preserve

Theses and Dissertations

2013

A Performance Analysis of Vision-Based Robot Localization System

Yufei Qi
Lehigh University

Follow this and additional works at: http://preserve.lehigh.edu/etd

Recommended Citation

Qi, Yufei, "A Performance Analysis of Vision-Based Robot Localization System" (2013). *Theses and Dissertations.* Paper 1091.

This Thesis is brought to you for free and open access by Lehigh Preserve. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Lehigh Preserve. For more information, please contact preserve@lehigh.edu.

A Performance Analysis of Vision-Based Robot Localization System

by

Yufei Qi

A Thesis

Presented to the Graduate and Research Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Mechanical Engineering

Lehigh University

December, 2012

This thesis is accepted and approved in partial fulfillment of the requirements for the Master of Science.

Date

Thesis Advisor
Eugenio Schuster

Department Advisor
Eugenio Schuster

Chairperson of Department
D. Gary Harlo

Acknowledgements

I would like to thank my advisor, Dr. Eugenio Schuster of the Lehigh University Department of Mechanical Engineering, for his input throughout this work. I would also like thank my colleague, Anthony Dzaba of Lehigh University Department of Mechanical Engineering for his contributions and continued feedback.

Table of Contents

Acknowledgements iii

Table of Contents iv

List of Figures vi

List of Tables viii

List of Symbols and Nomenclature ix

1. Abstract 1

1. Introduction 2

- 1.1. Background 3
- 1.2. Literature Review 4
  - 1.2.1. Vision-Based Robot Navigation 4
  - 1.2.2. Image Processing Theory 6
  - 1.2.3. Motion Estimation 7

2. Background Theory 9

- 2.1. The Hough Transform 9
- 2.2. Digital Image Correlation 12
- 2.3. Feature-Based Correspondences Detection 14

3. Materials and Apparatus 16

- 3.1. Vision-Based Local Positioning System (LPS) 16
- 3.2. Camera 17
- 3.3. Microsoft Kinect 18
- 3.4. The Test Bench and Membrane Potentiometer Positioning Sensor 20

4. Evaluation of LPS Attitude Estimation with Cylinder Marker 22

- 4.1. Cylindrical LED Marker 23
- 4.2. The Hough Line Detector for Cylinder Marker 25
- 4.3. The Cylinder Marker Projection Model 30

5. LPS Position Estimation Performance Analysis 32

- 5.1. Accuracy and Precision 32
- 5.2. Measurement Resolution 37

6. Visual Odometry for MAV Positioning 38

- 6.1. Monocular Visual Odometry 39
  - 6.1.1. Visual Odometry Imaging Model 40
  - 6.1.2. Monocular Visual Odometry with DIC 41
  - 6.1.3. Monocular Visual Odometry with SURF 43

- 6.2. Stereo Visual Odometry ... 45
  - 6.2.1. Stereo Camera ... 46
  - 6.2.2. Kinect Visual Odometry ... 48
  - 6.2.3. 3D-to-3D Motion Estimation ... 52
- 6.3. Experiment and Results ... 53
  - 6.3.1. Linear Positioning Sensor ... 53
  - 6.3.2. Experiments and Results ... 57

7. Conclusion ... 62

Literature Cited ... 65

Vita ... 69

List of Figures

Figure 1: Block diagram showing main components of VO system 6

Figure 2: Polar coordinates form for line representation 10

Figure 3: The Hough Transform 11

Figure 4: The USB CMOS Monochrome Board Camera [33] 17

Figure 5: Microsoft Kinect [34] 18

Figure 6: Microsoft Kinect structure [35] 18

Figure 7: IR image and depth image acquired by Microsoft Kinect........................... 19

Figure 8: SoftPot structure and functionalize demonstration [36] 20

Figure 9: SoftPot membrane potentiometer sensor [36] 20

Figure 10: Membrane potentiometer sensor attached to the test bench...................... 21

Figure 11: SoftPot calibration curve 21

Figure 12: Cylinder marker ray intersection 23

Figure 13: The marker placement geometry model 25

Figure 14: The marker detection.. 25

Figure 15: The line extraction for cylinder markers 26

Figure 16: Hough Line Detector 27

Figure 17: The line sharp tracking model 29

Figure 18: The cylinder projection model 31

Figure 19: The demonstration sample of a point measured 34

Figure 20: Monocular VO Positioning model 41

Figure 21: Image correlation for VO 42

Figure 22: Feature based Correspondence Establishing Demonstration 44

Figure 23: Block diagram of Feature based Monocular VO 45

Figure 24: Block Diagram of stereo VO 46

Figure 25: The stereo feature matching 48

Figure 26: Depth map aligned with the color map 49

Figure 27: Block diagram of Kinect VO 50

Figure 28-3: Real world coordinates of detected features 51

Figure 29: Linear positioning sensor measurement standard deviation 54

Figure 30: Membrane potentiometer sensor sampling time test 55

Figure 31: One-Dimensional Digital Image Correlation 58

Figure 32: Results comparison for visual odometry with DIC 59

Figure 33: 2D Visual odometry with Feature based method 59

Figure 34: Camera Displacement comparison 60

Figure 35: Camera path tracking 61

Figure 36: A comparison of measurement error with feature number 61

Figure 37: Offline Kinect positioning result 61

List of Tables

Table 1: LPS X measurement..35

Table 2: LPS Y measurement..36

Table 3: LPS Z measurement 37

Table 4: Membrane potentiometer sensor sampling time test 55

Table 5: Membrane potentiometer linear positioning sensor calibration test.........................56

List of Symbols and Nomenclature

DoF – Dimension of Freedom

LPS – local positioning system

VO – Visual Odometry

V-SLAM – Visual Simultaneous Localization and Mapping

UGV – Unmanned Ground Vehicle

MAV – Micro Aerial Vehicle

DoG – Difference of Gaussian

SVD – Singular Value Decomposition

SD – Standard Deviation

RSD – Relative Standard Deviation

1. Abstract

A performance analysis of several vision-based robot localization systems is presented for real-time Micro Aerial Vehicle (MAV) navigation tasks in touch-free, GPS denied, and high-accuracy environments. The systems were designed and utilized during our quadrotor visual control research, consisting of a local positioning system (LPS), a simplified monocular visual odometry (VO), and a stereo visual odometry ranging system. Measurement performance is evaluated through experiments performed using a membrane potentiometer sensor as reference. Feature-based image processing algorithms and motion detection methods are implemented to generate 3D position information from 2D image data. Performance analysis gave verified data for the applications of those systems. Final measurement resolution of 1mm was obtained for the LPS, and a high positioning accuracy was demonstrated for the VO systems.

1. Introduction

Robot localization and tracking is one of the most considerable competences required for robot navigation. Vision-based positioning, as an efficient way for robot localization, is to generate the target position information or solve 6 DoF problems for a robot, by performing an incremental analysis with only the input of a single or multiple cameras. The overall goal of this research is to give an analysis of several proposed vision-based localization system, with the comments about their advantages and limitations. The result of this research gives the comments of real world applications for these proposed systems, based on the analysis of their measurement performance.

Depends on the type of task, the vision-based localization system can be divided into Local Positioning System (LPS) and the Visual Odometry (VO) system. The LPS is a camera system, containing two local-fixed cameras observing a marked target. The position and altitude of the target is estimated by triangulating the markers (usually LED markers) through their detected projections in the image frames. The VO is the process estimating the camera's motion using the consecutive frames captured with a sufficient scene overlap. Depending on the cameras that been used, the VO can be further divided into Monocular VO and Stereo VO. As measuring systems, all these vision-based systems can be utilized for the tasks that require low-cost, touch-free measurement techniques, especially in robotics areas. The optimal application of each proposed system is determined by the concentrations of these vision-based localization systems, which is discussed in this research.

1.1. Background

Prior research has demonstrated various applications of vision-based positioning techniques [1]-[10], including obstacle avoidance [1]-[2], and mobile robot navigation [3]-[5], aerial and underwater vehicle visual control [6]-[10]. In each of these approaches, vision-based positioning techniques are utilized for either tracking or detecting objects, or localizing the position or pose of an agent (e.g. vehicle, aircraft, or robot). Depending on the task, the visual system can be of stereo cameras [4], [7], single camera [1], [6], or a combination of camera with other sensors [2], [9], [7] (i.e. IMU, laser sensors, and ultrasonic sensors).

In general, the goal of a visual positioning system is to produce continuous, accurate and repeatable real-world position information about a target under surveillance, or a moving agent with camera attached, while minimizing the cost, computation and complexity, under desired environment. The visual positioning system is well studied over past several years. Yet with the improvement in camera and sensor techniques, and image processing methods, new approaches that employed visual positioning were developed continuously, representing high performance in real-time tasks [6]-[8].

In this work, the introduction and analysis of several visual positioning systems developed by our lab over the past two years is presented. The primary contribution for this work is to give the measurement performance of the proposed visual positioning systems, and test the real-time VO implements which were developed for quadrotor tasks. The experiments were done with a test bench, providing reliable true values for the results, by simulating the vehicle movement.

1.2. Literature Review

The computer vision techniques have been employed into robot localization and navigation since the early 1980s. Additionally, the image processing and motion estimation techniques were developed for more accurate and stable real-time applications. In this section, an overview of prior research is drawn. The literature review is divided in three areas: The vision-based robot navigation, the image processing techniques, and the motion estimation with visual inputs.

1.2.1. Vision-Based Robot Navigation

In robot navigation, the process of determining the position and orientation of an agent by analyzing the input of the camera system attached to it is called Visual odometry (VO) [11]. It focuses on estimating the 3D camera displacement and poses changing over the last several frames and generating the motion of the camera. It has wide applications in target tracking and localization. VO is not interested in global consistency of the path. Instead, it concentrates on the analysis of the last several frames, without keeping track of all the pervious history. As an early VO application, Moravec’s cart [1] introduced a sliced-stereo technique as a navigation method estimating mobile robots’ egomotion based on visual input. Limited by technology of the day, the robot moved in a stop-and-go fashion, digitizing and analyzing images at each stop. As a consequence of thirty years’ research in sensors and algorithms, the increased positioning accuracy, speed operation, and functional flexibility in computer vision techniques, we are now capable of implementing an efficient system that provides good results with high accuracy, reliability and real-time performance, while minimizing cost, complexity and computation. As two independent lines of VO research, the monocular camera system and the stereo camera system are discussed in the remainder of this section.

The monocular VO is to estimate the position with only one camera, providing bearing information. Since the magnitude of the direction vector is unknown, the motion is usually resolved to a relative scale by setting the movement of the first two images to one. To settle this problem and determine the absolute scale, the monocular camera can be combined with other sensors to produce distance measurements.

In comparison with monocular VO, stereo VO computes the relative motion by using the 3D point generated through triangulating the stereo pairs. The image features in two image sets are detected or tracked with image feature detection and matching theories. By triangulating the corresponding features in a stereo pair, the 3D correspondences are established. With the detected 3D points, the relative motion is computed with 3D-to-3D or 3D-to-2D motion estimation methods, and refined by an outlier rejection scheme.

Nister et al. [5] have presented a system that can operate in real-time and estimate the motion for robot navigation. They involved the feature based image processing, and estimated the relative pose using 5-point algorithm [12] based on the tracked features over several frame, and utilized RANSAC [13] to eliminate the outliers.

JPL has presented several stereo-vision-based terrain mapping studies and the capacity of stereo techniques to perform UGV navigation in [4]. Various stereo techniques such as multi-resolution, thermal infrared cameras, and multi-baseline stereo camera were included for off-road approaches.

An overview of the VO system is given in [11]. The general procedure of a VO system is demonstrated in the following block diagram.

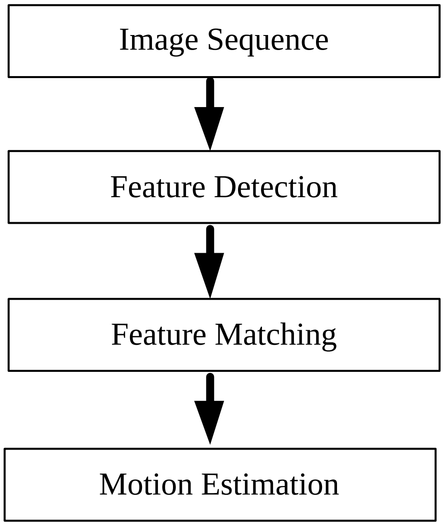


Figure 1: Block diagram showing main components of VO system

1.2.2. Image Processing Theory

The VO system performance is highly related to the image processing procedure. This stage most heavily influences the execution speed and real-time performance. As illustrated in figure 1, the first three steps, called the image correspondences problem, which detects and marches (or tracks) the 2D feature of the same 3D feature cross consecutive images.

The motion estimation is established upon good correspondences detection over consecutive images. The procedure of establishing correspondence can be divided into two main categories, representing two main classes of algorithms.

- Appearance-based methods: Based on correlating pixel intensities of the images or their sub-regions. For instance, Digital image correlation.
- Feature-based methods: Based on salient and repeatable features that are tracked over several frames. For instance, SIFT and SURF algorithms.

The early researches concentrated on the former approach for small-scale environment or small viewpoints changing. The later works focused on the large-scale environment with images taken as far apart as possible from each other. In feature-based methods, features are specified to be the image structures or patterns that can be easily identified and repeatable across a set of images. The ideal feature should have following the properties: accuracy, repeatability, efficiency, robustness, distinctiveness, and invariance [14].

The prior works gave various approaches in feature detector and descriptor. For early researches, the intersections of edges (known as corner points) were interested. Harris corner detector [15] gave a fast way to detect and track distinctive corner points through images with various applications. Concentrating on the invariance after large viewpoint and scale changes, blob detector gave a better result than the corner detector. SIFT [17] is an algorithm developed by Lowe, which extracts distinctive invariant features from images to perform reliable matching between consecutive images. In terms of real-time performance, Herbert et al. [20] had built a Speed-Up Robust Features (SURF) method to present a high-speed approach in feature detection and description. The review and comparison of the existing algorithms can be found in [14].

1.2.3. Motion Estimation

After detecting and matching the correspondences, the next stage in a VO system is motion estimation. In this step, the purpose is to get the transformation $\mathrm{T_k}$ between the current image $\mathrm{I_k}$ and the previous image $\mathrm{I_{k-1}}$ through analysis two sets of correspondence $\mathrm{f_k, f_{k-1}}$. And then the trajectory of the camera or the agent to which it is rigidly mounted can be further determined by concatenation of all these movements. This stage can be divided into three different methods based on the dimension in which the corresponding features are specialized [11].

- 2D-to-2D: $\mathrm{f_k}, \mathrm{f_{k-1}}$ are specified in 2D image frame.
- 3D-to-3D: Both features points are triangulated into 3D coordinates at each iteration.
- 3D-to-2D: Feature set $\mathrm{f_{k-1}}$ are triangulated into 3D coordinates and then matched with it 2D corresponding feature $\mathrm{f_k}$.

The geometric relations between two image frames of a camera are described by the essential matrix (with intrinsic and extrinsic parameter) or fundamental matrix (with extrinsic parameter only). With 2D feature correspondences, the essential matrix can be computed by using the epipolar constraint. It involves a minimal five-point solution [5]. With eight or more non-collinear points, the eight-point algorithm [30] gives a solution for both calibrated and uncalibrated cameras. For an over-determined system, which having more than eight points, the singular value decomposition is used to estimate the result.

In the stereo vision case, the camera motion is computed by determining the transformation of two 3D corresponding feature sets. The evaluation of motion estimation method for stereo VO can be found in [31].

2. Background Theory

In the visual localization task, images acquired with cameras are analyzed; the correspondences are detected and matched over a continuous frame stream. The image features are defined as distinctive patterns that have high repeatability over different scenes. The efficiency and effectiveness of the correspondences recognition stage influence the overall execution speed and accuracy of a localization system. For the LPS, the image features are easily identified as bright LED markers under low illumination. The noise and interference from other objects in the field of view are eliminated or minimized with low exposure condition. In the way, the marker can be tracked over a frame stream without complex feature detection and object identification. During the design of a VO applied for positioning and path tracking, the complex background features were involved as land beacons to produce abundant location information.

This section gives the theories that involved in this research, and is organized as follow. For the accuracy of LPS altitude measuring, the cylinder markers were involved and identified with the Hough line detector introduced in section 2.1. The standard image processing algorithms were involved in the VO design, which are introduced in section 2.2. Section 2.3 introduces the digital image correlation theory.

2.1. The Hough Transform

The Hough edge detector was employed to extract the straight lines in binary images. The principle of this technique is to group the noisy extracted edge features into object candidates by a voting procedure in a parameter space. Each edge pixel (x, y)

corresponds to a family of straight lines $L(x, y) = \{l_1, \dots, l_n\}$ in 2D image frame parameterized by two variables,

$$y = a_i x + b_i \tag{2-1}$$

Considering $a, b \in [-\infty, +\infty]$ (e.g. vertical line), it is better to use polar coordinate form for line representation, as shown in figure 2,

$$r_i = x \cdot \cos\theta_i + y \cdot \sin\theta_i \tag{2-2}$$

Here, r_i corresponds to the normal distance to the line and θ_i corresponds to the polar angle, with a range of $r_i \in [0, \sqrt{im_v^2 + im_h^2}]$, $\theta_i \in [0, 2\pi]$. Here, im_v and im_h are the vertical and horizontal dimensions of image.

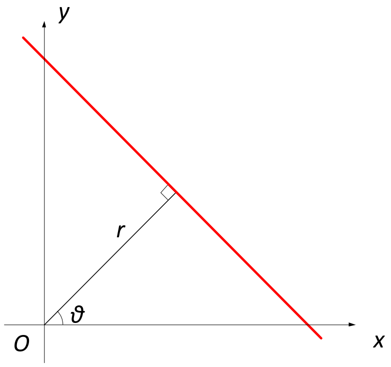


Figure 2: Polar coordinates form for line representation

Theoretically, each pixel (x, y) votes for the parameters (r_i, θ_i) of each line $l_i \in L(x, y)$, and straight lines have more edge pixels on it will receive more votes than the ones that are forming a line in the image. And then, the line feature detection problem is transformed into a peak detection algorithm finding the most voted lines, corresponding to the greatest intensity in the image frame. Since the parameters (r_i, θ_i) are bounded, it is easy to define a resolution dr and $d\theta$ for an angular scan. The angular interval $\theta_i \in$

$[0,2\pi]$ is dispersed by $d\theta$ into $n = 2\pi/d\theta$ discrete points, and for each of them the related radial distance r_i is calculated through equation (2-2) and rounded to the nearest dr value. Furthermore, the corresponding element in a so-called accumulator array $H(r,\theta)$ will be accumulated once. Therefore, an accumulator space denoted by a $m \times n$ matrix H is constructed respectively by accumulating the votes to obtain local maxima. Here $m = \sqrt{im_v^2 + im_h^2}/dr$ and $n = 2\pi/d\theta$ correspond the discrete radial and angular interval. As illustrated in figure 3, each edge pixel (x, y) corresponds to a family of straight lines $L(x, y) = \{l_1, \dots, l_n\}$. Edge pixels that form a line will each place one vote for the same (r_i, θ_i). Lines have more edge pixels on it will receive more votes than ones that are forming a line in the image.

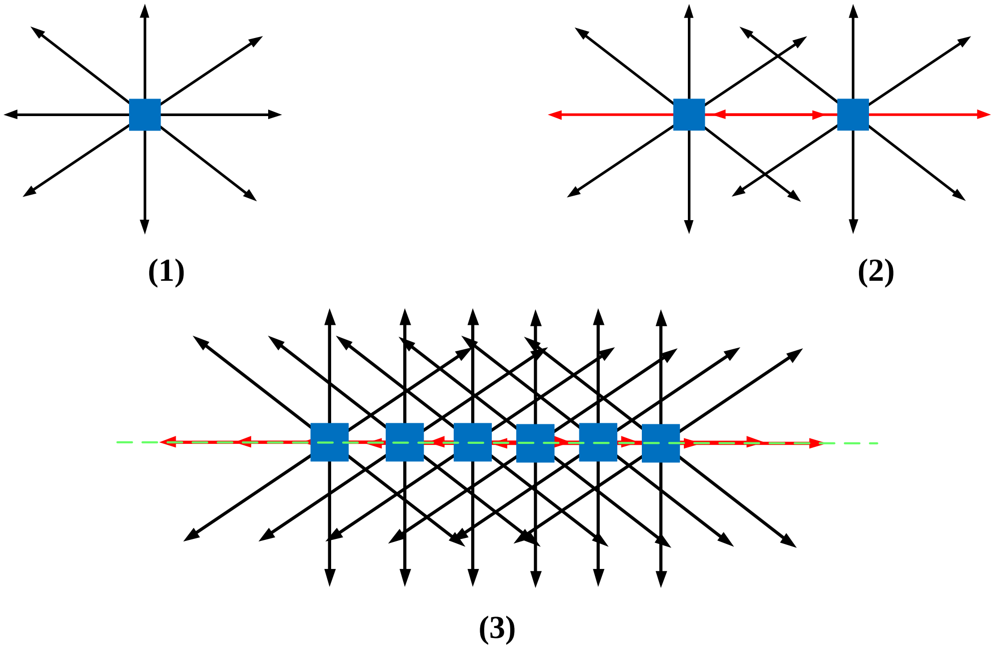


Figure 3: The Hough Transform

2.2. Digital Image Correlation

DIC was developed in the recent 30 years and is now capable for real time task with a satisfied execution speed. As a touch free technique with high accuracy, this technique is widely used in mechanical testing, machining, and recently nano-scale chemical surface restructuring. DIC is also utilized for motion measuring; one common application is optical mouse. Inspired by this, the visual odometry system is developed based on the concept of DIC, intent to represent an alternative solution for robot positioning.

The first approach for the visual odometry system imaging processing procedure is Digital Image Correlation (DIC). This technique is an optical method predicated on the maximization of a correlation coefficient that generated by comparing sub-pixel intensity or gray scale on two or more corresponding images. Let $\mathrm{F}(\mathrm{P}_{\mathrm{ij}})$ represent the pixel intensity at point P_{ij} in the original image, and $\mathrm{G}(\widetilde{\mathrm{P}}_{\mathrm{ij}})$ is the pixel intensity at point $\widetilde{\mathrm{P}}_{\mathrm{ij}}$ in a displaced image. The correlation coefficient r_{ij} is determine by normalized cross-correlation given by,

$$\mathrm{r}_{\mathrm{ij}} = \frac{\sum_{\mathrm{i,j}\in \mathrm{S}}(\mathrm{F}(\mathrm{P}_{\mathrm{ij}})-\overline{\mathrm{F}}_{\mathrm{S}})(\mathrm{G}(\widetilde{\mathrm{P}}_{\mathrm{ij}})-\overline{\mathrm{G}}_{\mathrm{S}})}{\sqrt{\sum_{\mathrm{i,j}\in \mathrm{S}}(\mathrm{F}(\mathrm{P}_{\mathrm{ij}})-\overline{\mathrm{F}}_{\mathrm{S}})^2 \sum_{\mathrm{i,j}\in \mathrm{S}}(\mathrm{G}(\widetilde{\mathrm{P}}_{\mathrm{ij}})-\overline{\mathrm{G}}_{\mathrm{S}})^2}} \tag{2-3}$$

With S is a subset of pixels around a point of interest and $\overline{\mathrm{F}}_{\mathrm{S}}$, $\overline{\mathrm{G}}_{\mathrm{S}}$ are the mean values of Fand G in the area of S. The mapping from the corresponding image to the original image is defined by,

$$\mathrm{P}^{*} \rightarrow \mathrm{P} = \chi(\mathrm{P}) \tag{2-4}$$

By assuming that the camera only has parallel motions and always perpendicular to the optical axis of the camera, or in other word, only translations are expected and no rotations exist, the relation between P_0 and P_0^{*} is reduced to a 2D affine transformation,

$$x^* = x + u(x, y) \tag{2-5}$$

$$y^* = y + v(x, y) \tag{2-6}$$

Or take the first Taylor series expansion,

$$x^* = x + u(x_0, y_0) + \frac{\partial u}{\partial x}\Delta x + \frac{\partial u}{\partial y}\Delta y \tag{2-7}$$

$$y^* = y + v(x_0, y_0) + \frac{\partial v}{\partial x}\Delta x + \frac{\partial v}{\partial y}\Delta y \tag{2-8}$$

Here u and v are respectively the x and y direction in-plane displacements between two images, specifically, let $P_0 = (x_0, y_0)$ represents the point of interest and P_0^* is the corresponding point in the deformed image, are the translation of the point of interest (or center of the sub-image). $\Delta x = (x - x_0), \Delta y = (y - y_0)$ denotes the distance between P and P_0. Under the assumption of parallel movements of camera, tiny surface distortion between I_k and I_{k-1} will be ignored. That means the sub-images are considered as rigid, or $\frac{\partial u_i}{\partial x_i} = 0$. Equation (2-7) and (2-8) can be rewritten as,

$$x^* = x + u(x_0, y_0) \tag{2-9}$$

$$y^* = y + v(x_0, y_0) \tag{2-10}$$

2.3. Feature-Based Correspondences Detection

Feature-based methods are based on salient and repeatable features that are tracked over several frames. The following of this section introduces two blob feature detection algorithms.

Developed by Lowe [3], SIFT detects local features in a scale space generated by Gaussian filter. Images are smoothed and masked by convolved with the Gaussian kernel incrementally in a down- sampling way to produce a set of the scale space images and establish a scale space. Point with the difference of Gaussian (DoG) larger than its neighborhoods in the scale space is believed to be a local extreme. For stability and repeatability, the key points are selected as local extreme by eliminating points with low contrast and close or lying on to edges. After key points are localized, histograms of gradient direction are created, within a region around each key point. A 128D SIFT descriptor is assigned for a key point, containing its 8 dimensional orientation vector. By searching for features in other images, a 128D descriptor will find a best match and considered as correspondence.

SURF [20] algorithm is a blob feature detection technique, concentrating on the find image pattern that distinctive from it neighborhood and match them over images. It builds up on SIFT, but comes out with box filters to approximate the Gaussian, and detects interest features with high repeatability under different viewing conditions by up-scaling filter in a scale space. The advantage of box filters is to separate calculation time from filter size. It can directly apply up-scaling filter on the original without iteratively down-sampling the image to reach the upper level of scale space octaves, which improved the computational efficiency. And then by calculating Haar wavelet responses at a neighborhood of the detected point and summing up the responses in x and y direction, SURF creates a 64D descriptor utilizing integral images for each key point. In trade off some robustness to illumination and viewpoint changes, it achieves a faster speed than SIFT.

These algorithms have good scale- and rotational- invariance and stability to small view point changes. By working with local feature, the feature-based algorithms can extract feature vectors independent from scale, rotation and illumination from several images. In this research, these algorithms were utilized for image corresponding feature recognition. The VO tests were implemented in real time by applying SURF.

3. Materials and Apparatus

This approach was to design a visual inputs based camera motion estimation system for robot localization. The system contained several sensors, and each of them can work separately to provide continuous position information inputs. The purpose of each system or sensor is discussed in the following section. The combinations of the different data inputs were performed as independent systems to provide required information for camera positioning. Developing solutions for robot localization were carried out by utilizing low-cost, lab-ready materials, and this section are introduced the materials that was used in this research.

3.1. Vision-Based Local Positioning System (LPS)

The vision-based local positioning system [37] was developed to provide low-cost, scalable and real-time assistance for laboratory robotics research. It produced the real-time positioning information of a marked robot estimated through two calibrated local-fixed cameras. It was constructed using two digital cameras, white LEDs, and a computer workstation. A set of four LED markers were placed on a quadrotor and monitored by two strategically placed digital cameras. By using 3D computer vision techniques, the pose of the miniature aerial vehicle can be estimated by tracking the LED markers in image frame.

3.2. Camera

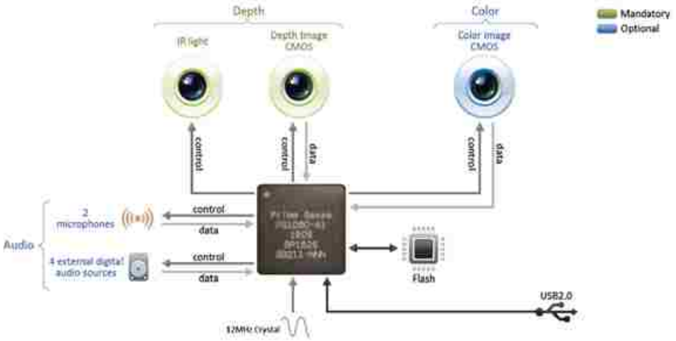
[redacted]

The camera system utilized in this study involves a USB CMOS Monochrome Board Camera from The Imaging Source (DMM 72BUC02-ML). With the tiny size ($1'' \times 1''$) and low payload (7g), this camera is capable for on-board task of various vehicles, including the quadrotor. In demonstration, it was attached to a test bench and provided with linear motions to evaluate the real-time performance of the algorithm. The picture resolution was set to a minimum 640×480 with grayscale image style to achieve an optimal operational frequency (30~50 Hz). Also, decreasing the image size can efficiently improve the algorithm execution speed because when the pixels are reduced, the image processing speed is increased. The 12 mm focal length limits the field of view to a very narrow area representing a zoomed-in image. The reason for such a long focal length is that the system acquires the displacement information through correlating a set of input image. The image correlation procedure is intended to get a so-called sub-image movement (will be discussed in Monocular VO section) by continuously comparing an input image and the previous images. That requires a fixed narrow field of view to speed up the execution with less input information and reduce the noise from complicated background environment and various objects. A zoomed-in view of the reference system allows for more details to distinguish sub-images.

3.3. Microsoft Kinect

[redacted]

The Microsoft Kinect motion sensing input device is a human pose recognition system developed on top of a 3D scanner called “light coding”. This technique involves only normal CMOS camera for 3D measurement, thus drastically reducing the cost and making the Kinect one of the primary positioning devices in robotics. The algorithm behind Kinect was published in [28]. The applications and researches involving Kinect for robot navigation and object recognition can be found in [8], [23], [24]. Studies and analysis about Kinect imaging, calibration and measuring can be found in [25], [26].



[redacted]

The Microsoft Kinect is a powerful 3D measurement system, yet the technologies and algorithms were not discussed in this paper, instead focused on the applications of Kinect as visual odometry. As illustrated in figure 6, posted by PrimeSense, the Kinect sensor contains a CMOS color camera and a depth camera formed by an IR projector and an IR CMOS camera. The IR projector sent out a fixed pattern generated from a set of diffraction gratings (see figure 7). By triangulation and correlation against those patterns, the depth at each pixel was generated. This procedure is adopted from the Open NI (open Natural Interaction) interface published by PrimeSense for natural interaction devices. The primary algorithm used to generate the depth map for each pixel in the field of view is wrapped by Hideki Shirai [27]. The depth information for each pixel in the field of view was converted from depth map into camera fixed coordinates. Kinect provides a set of range information for each pixel inside the field of view instead of an average distance approximation by using a sonar sensor. For that reason, Kinect sensor is demonstrated to be an alternative solution of VO with the depth sensor generating range data for the whole field and RGB camera running the correlation procedure.

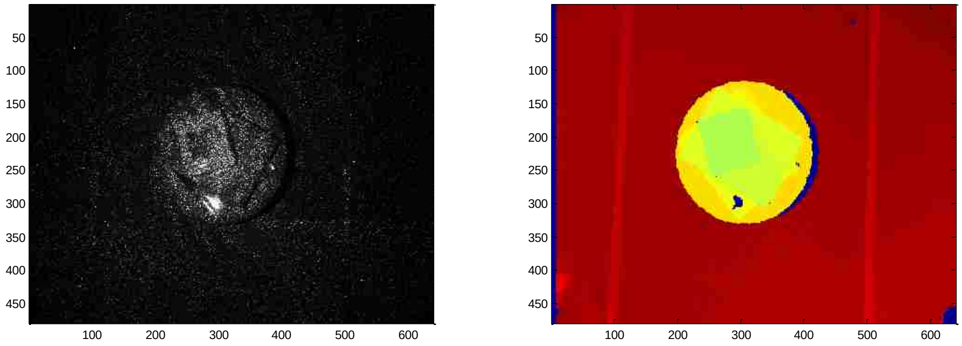
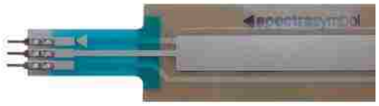


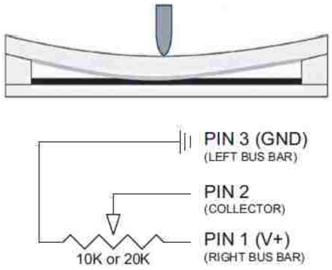
Figure 7: IR image and depth image acquired by Microsoft Kinect

3.4. The Test Bench and Membrane Potentiometer Positioning Sensor



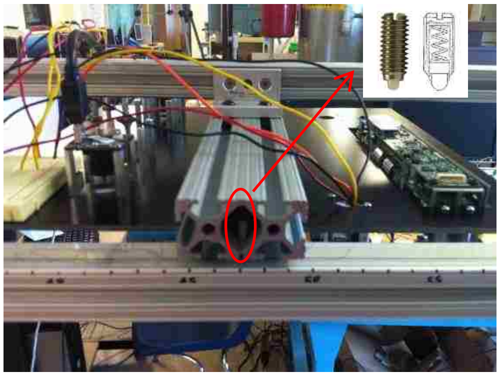
[redacted]

The test bench was built for the purpose of estimating the visual odometry system measurement performance in real time. Theoretically, the sensors will be assembled onto the test bench and operate simultaneously. The data gathered through the visual odometry systems transported to a desk computer, and the computer operated an imaging analyzing procedure, which is the core component of the VO system and determines its performance. The test bench also contains a SoftPot membrane potentiometer sensor from SpectraSymbol as a reference linear positioning sensor. As a resistive element, it contains a conductive resistor and a sealed encasement. It is functionalized by adding a pressure to the encasement to form a current loop at the specific position, as illustrated in figure 9. Pin 2 is the collector outputs analog voltage data representing the position of pressure. Voltage outputs can be read and converted into position information in real time by a microcontroller like Adriano. The reason to use this sensor as a reference measurement was that the experiments required a side positioning system that returns position data in real time, as a true value to estimate the VO system performance. This reference positioning system required a minimal resolution of 1 mm with at least 1 m range. As the specifications claim, SoftPot has a theoretically infinite analog output affected by variation of contact pressure area. To confirm the reliability of this reference system and determine the transformation between analog



[redacted]

outputs and positions, a short test was conducted to calibrate the membrane potentiometer sensor.



[redacted]

The VO system was attached onto the test bench with an aluminum bar. As illustrated in figure 10, a wiper was taped into the aluminum to apply a point pressure. The movement of the aluminum camera can be measured with the membrane potentiometer sensor and regarded as the reference position information for the VO system. To calibrate it, points distributed every 1 cm were marked and assigned as measurements within the sensor maximum range. For each point, an offline sensor data reading procedure was repeated 1000 times to get the most reliable measurement and a curve representing the relation between sensor analog outputs and positions, as shown in figure 11. An approximately linear relation can be observed and the plot of the curve can be utilized for membrane potentiometer sensor data converting. This curve is the calibration curve to convert analog to position.

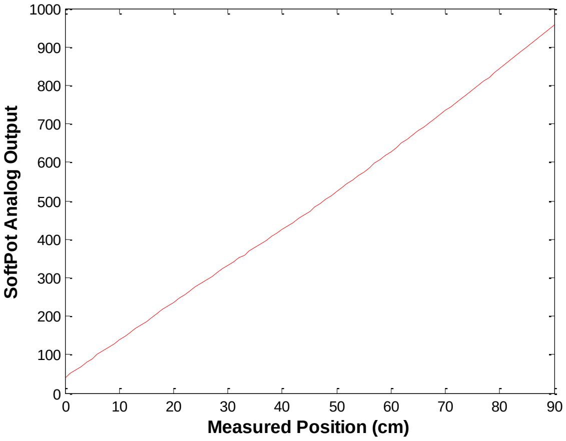


Figure 11: SoftPot calibration curve

4. Evaluation of LPS Attitude Estimation with Cylinder Marker

In the above sections, demonstrations have been made for how the Local Positioning System (LPS) functions by tracking spherical markers, and analyzed its performance as a measuring method, including its accuracy, precision, and measurement resolution in pose estimation. However, a 1~2 degree error in MAV's orientation has been observed with sphere markers, which were due to an intersection error in the triangulation phase. Considering this error in orientation would lead to even larger linear error in positioning, a new cylinder marker was introduced in pursing higher angular accuracy. This was derived from the fact that out as considering the orientation of the vehicle was divided into the 3D vector coordinates of its x-axis and y-axis (z-axis is represented as the cross product of x and y vectors), and a cylinder LED marker was regarded as continuously distributed sphere markers in one axis, which essentially averaged and canceled out orientation error.

However, as the centroid locating and triangulation method were applied, an even larger error occurred. The reason was suggested that, unlike the isotropic sphere marker, the cylinder marker was distorted by off-center observation angles, which actually exaggerated the mismatching of centroids. Thus, a shape-based detection system was created to avoid this kind of centroid matching error, and take advantage of cylinder markers in orientation estimation. This method was based on line extraction for x and y axes in image frame, and their intersection was converted into 3D points as the detected origin of the vehicle-fixed coordinates. The 2D to 3D transformation was the same as the one mentioned before. The difference lied in line extraction, which determined the overall performance and accuracy of the experiments. Mathematically, any 3D vector in the earth-fixed frame could be resolved in the camera-fixed frame, and converted through the matrix transformation into its 2D projection onto the camera's lens plane. Thus, the accuracy and precision of 2D projection image extraction directly determines the performance of the measuring system. Measures were taken to solve the problem of how to direct a feature extraction algorithm to find the axis in the image frame. Unlike the isotropic sphere marker, the cylinder marker was distorted by off-center observation angles and actually exaggerates the mismatching of centroids. Rays from camera I and II might not necessary look at the same centroid of the structure.

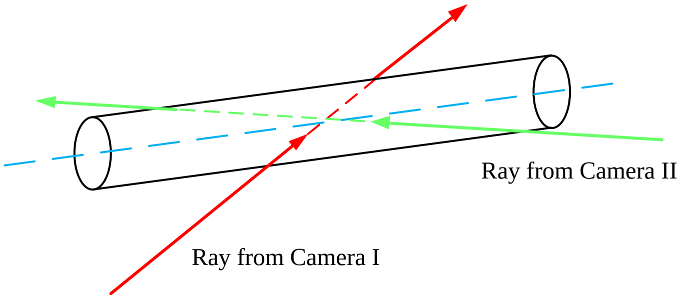


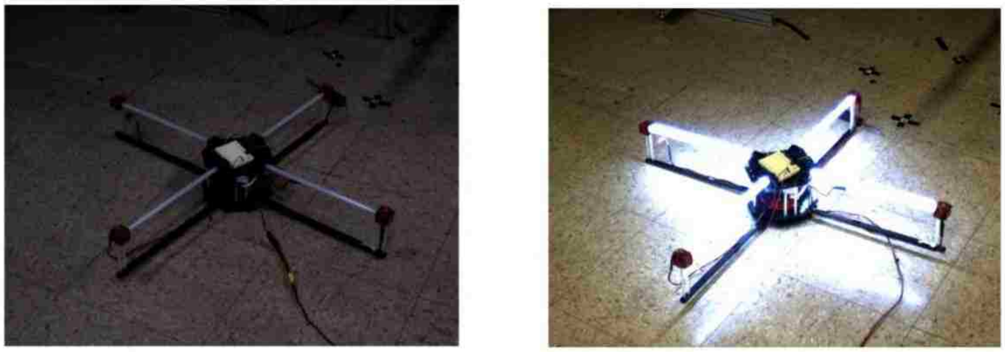
Figure 12: Cylinder marker ray intersection

As illustrated in figure 12, unlike circular cross-section projections of sphere, the marker's geometric center was no longer precisely represented by the centroid of the marker's projection image, and the perspective distortion would cause unpredictable error for any point-feature-finding methods. To account for this distortion, a shape based feature extraction method was employed in settling this problem.

4.1. Cylindrical LED Marker

The cylinder marker vision system was also designed to measure the six degree-of-freedom pose of the MAV, and attempted to achieve higher accuracy in altitude estimation. The position of the vehicle was given by the origin of the inertial and vehicle-fixed axes in the earth-fixed frame. The altitude, different from the sphere marker, was no longer determined by the origin and its permutation in vector space. Instead, the orientation of the vehicle-fixed axes x_v and y_v in the Earth-fixed frame were determined by the projection of x_v and y_v in the image frame, and the origin was regarded as the cross point of x_v and y_v. The projection of x_v and y_v, denoted by x_I and y_I, are 2D axes that were marked by the cylinder LED lights. The projection model was similar to the one introduced section 3-1, and illuminated cylinders were designed to project images to identify the axes in the image frame.

The markers were assembled as demonstrated in figure 13. Two pairs of cylinder LED light were placed vertically on the MAV, denoting x and y axes of the vehicle fixed frame. With the markers functioning, the LED lights should be bright enough to be identified itself in the dark background, and the vector coordinates were extracted after the procedures of imaging processing. Yet, as all kinds of noise such as light reflection can be observed, the measurement was noisy. Reflection and light generation from sources other than the LED markers are the most critical noises for the system and could result in the completely failure of target tracking, especially in online, continuous tracking procedures. One shortcut to address this problem in most indoor scenarios was to use low exposition grayscale images. In image processing, marker pixels were identified by setting the threshold h for pixel values ($h = 255$ in that example), and a binary image was returned for edge extraction.

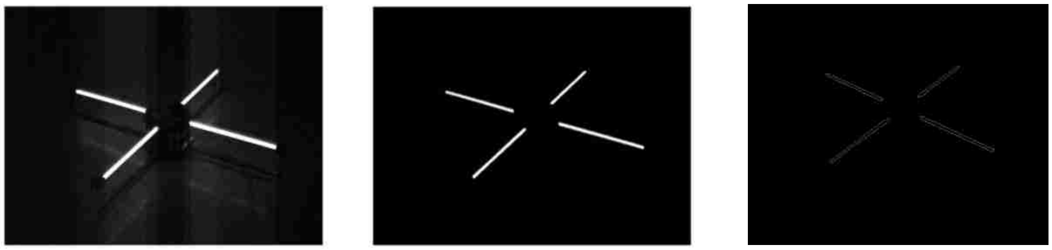


(1) **(2)**

Figure 13: The marker placement geometry model

4.2. The Hough Line Detector for Cylinder Marker

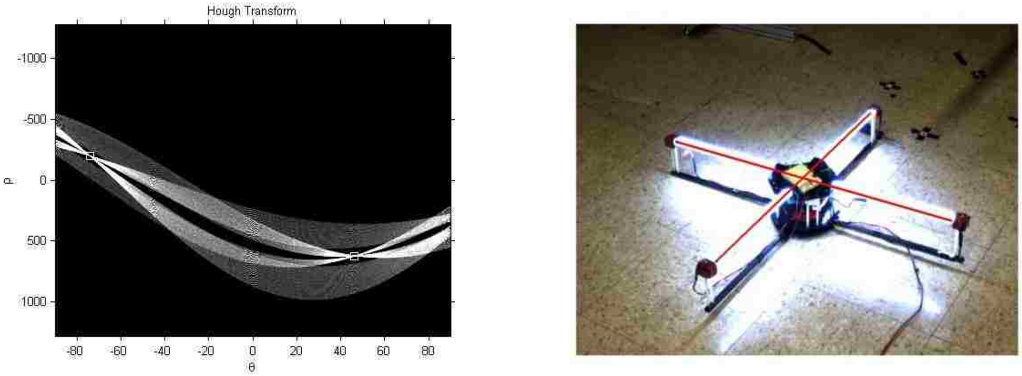
Hough transform is highly dependent on the quality of image: the imperfection errors in edge detection are usually the error in the accumulator space and a denoising stage was required. In most indoor scenario, high-powered LEDs are considered the primary light generator that can provid enough irradiation to excite the pixels of the imaging sensor with minimal exposure time. In image processing, marker pixels were identified by setting threshold h for pixel values ($h = 255$ in that example), and a binary image was returned for edge extraction.



(1) **(2)** **(3)**

Figure 14: The marker detection

To eliminate noise, a binary image was applied by setting threshold $h = 255$. Edge pixel extracted from the binary image, as illustrated in figure 14. With the edge pixel extracted, a Hough transform voting process was directed to establish the accumulator space denoted by matrix H, which was constructed respectively by accumulating the votes to obtain local maxima. More delicate angular resolution presents a higher accuracy, but at the cost to slow down the execution speed respectively. As in this case, an $d\theta = 0.1°$ angular resolution was employed to guarantee a minimal $0.5s$ execution speed. Based on the parameters (r_i, θ_i) obtained from accumulator space H, two line function was extracted to constructed the x and y axis in image frame, as the red line represented in figure 15. The origin of vehicle fixed coordinates in image frame were determined by the cross point of the extracted axes. The 2D image processing extracted the xy axes and the origin in both cameras. The triangulation procedure was applied to transform 2D vectors and points into 3D coordinates.



(1) **(2)**

Figure 15: The line extraction for cylinder markers

An accumulator space denoted by matrix H was constructed respectively by accumulating the votes to obtain local maxima. Based on the parameters (r_i, θ_i) obtained from accumulator space H, two line function were extracted to construct the x and y axis in image frame, as the red line represented.

One limitation of Hough Transform was that it was only efficient for high number voting, that means small bins could not be extracted and the votes would fall in the neighboring bins. Moreover, the accumulator space organized by discrete angular and radial interval, dispersed by angular resolution $d\theta$ and radial resolution dr. Setting resolutions is a very delicate matter. Although it appears to be that, for better measuring, the resolution should be as small as possible under certain hardware conditions. Generally, the resolution cannot be too delicate, not only for execution speed reasons, but also for threshold detection. That is especially critical for small bins.

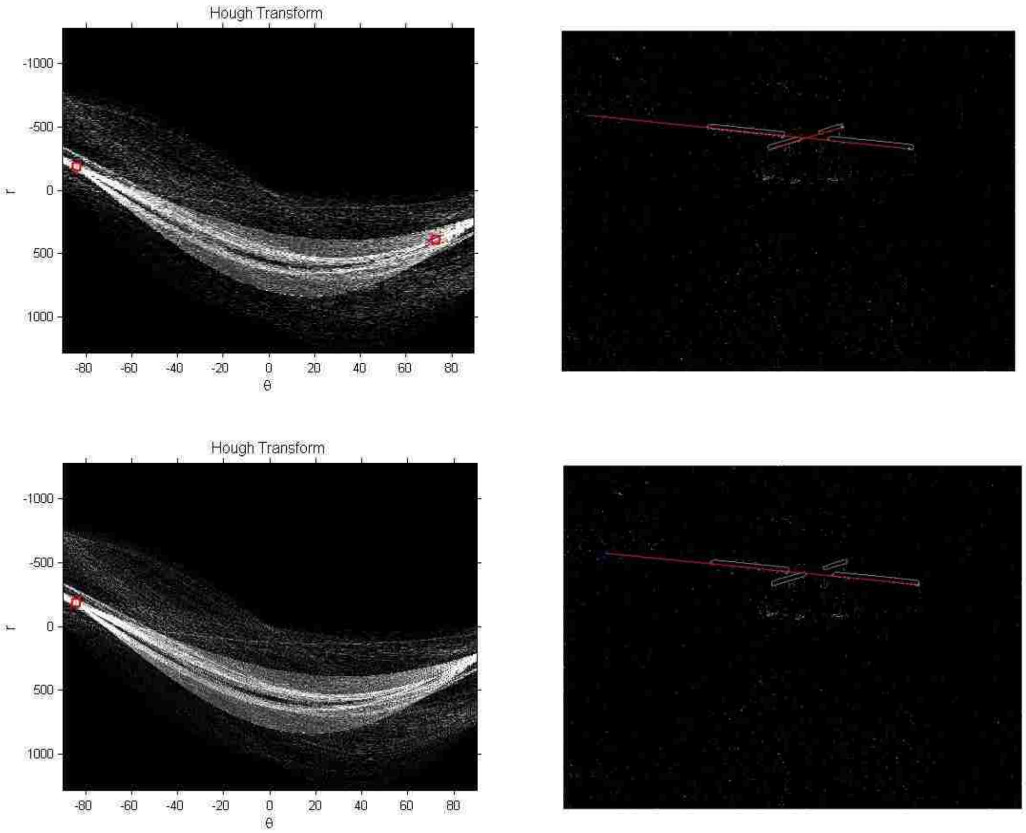


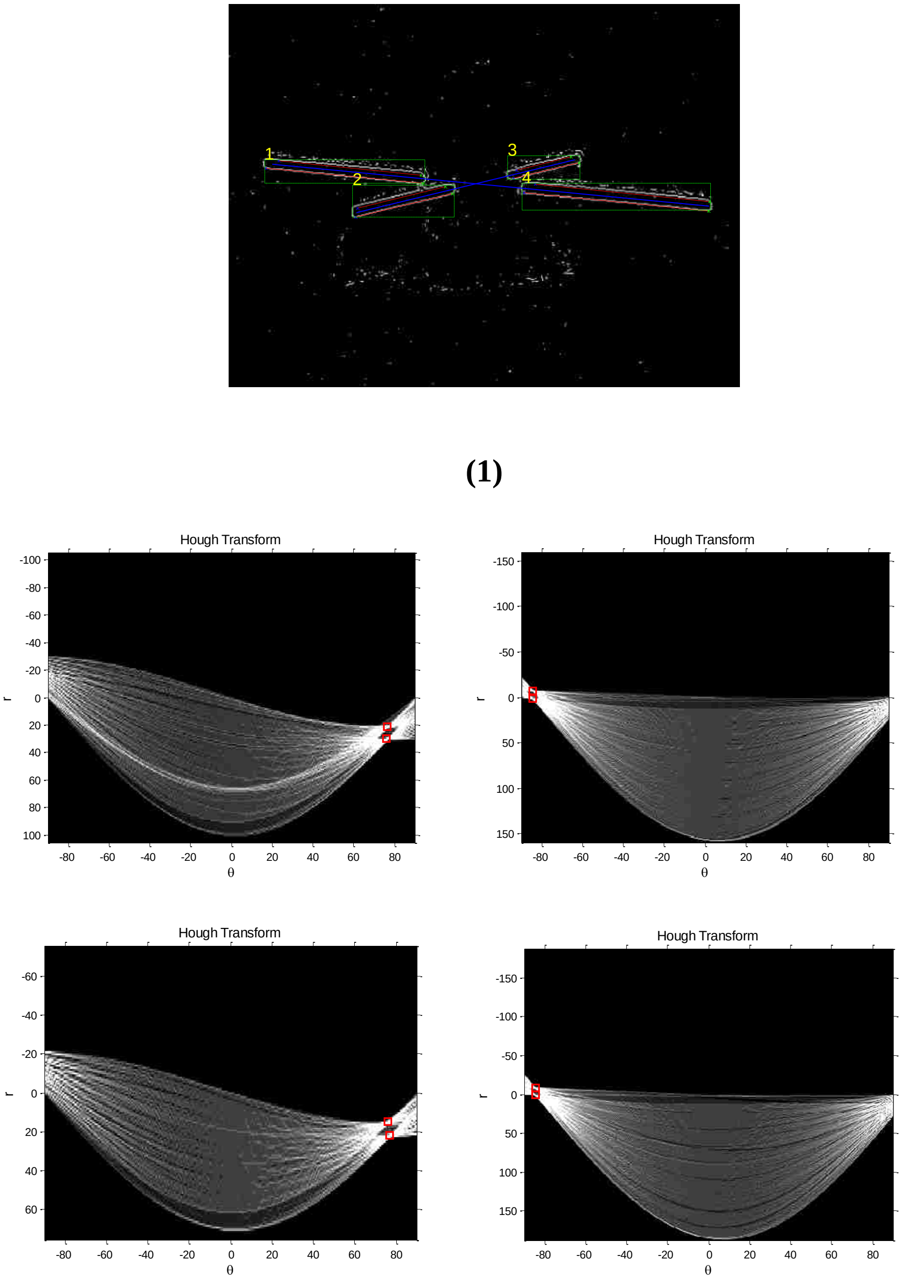
Figure 16: Hough Line Detector

As illustrated in figure 16, from top to bottom, setting radial resolution as a constant 1 pixel, and angular resolution $d\theta = 1 \; and \; 0.1$. As the angular resolution got more delicate, the peak region of the short bin in the accumulator space became smooth, that led to the peak value standing for the short bin to be ignored by the same peak detection algorithm. This kind of failure could vary with different input images.

The vehicle was expected to appear in any pose, random distance to cameras, various altitudes and orientations, and even the bins would be overlapped. To address this problem, at the offline calibration stage, the connected region was first to be identified. As illustrated in figure 17, each marker was bounded by a window representing its region and number. A similar line detector was directed with each window. The image corresponding x and y axes (blue lines) was given by averaging their related parallel red lines. The cross section of blue lines was the most reliable image reference representing vehicle origin.

The vehicle motion was estimated by the pose information gathered over time step $T_0, T_1 \dots T_k$, and the pose in T_{k+1} is predictable. The prediction model was a 3D-to-2D procedure (mapping 3D feature to its 2D projection onto the camera's image plane), and returned the predicted window. This is also a solution to the mismatch problem mentioned before. The execution speed is very critical in that case. For our system, the execution speed was 2~4 Hz, with radial resolution $dr = 1 \; Pixel$ and angular resolution $d\theta = 0.1°$. However, the fast dynamic of the vehicle would be expected, thus high frequency computation and superior processor is desired for real-time task. The markers were bounded and labeled in figure 17. The tracking procedures were operated within each sub image. The extracted lines were denoted by red. Blue lines represented x and y

axes image references, given by average their related parallel red lines. The accumulator space of each marker was given in figure 17-2. The peaks were easy identified with each window.



(1)

(2)

Figure 17: The line sharp tracking model

4.3. The Cylinder Marker Projection Model

The illuminated projections of the cylinder LEDs were identified by the line extractor. Considering foreshortening effect, a line-to-line projection model was involved instead of the point-to-point projection model, which would introduce error in that case. For each camera, consider the plane in earth-fixed frame formed by any extracted line of the project image and the camera focal points. The respective cylinder marker was obviously contained in that plane, which could be further determined by its cross section with the correspondent plane extended from another camera.

The object's projection onto the camera image plane formed a related feature representing the projector in the 2D image called correspondence. With two or more cameras, the correspondence features are distinct via different view angles. Detecting them and matching the set of correspondences related to the same object was the basis of the stereo system. With the information gathered from different cameras, the object can be triangulated and located. Our system was much simpler, only two features needed to be detected and matched. They were easily identified through the prediction model, which were discussed later. Now, assuming we have separated the x_v related plane from the y_v related plane, the detected vehicle axes, x_v and y_v, were defined by the cross section of each pair. Note that the normal vectors were resolved in the separated camera frames, in which they were measured. We first transformed them into the earth-fixed frame.

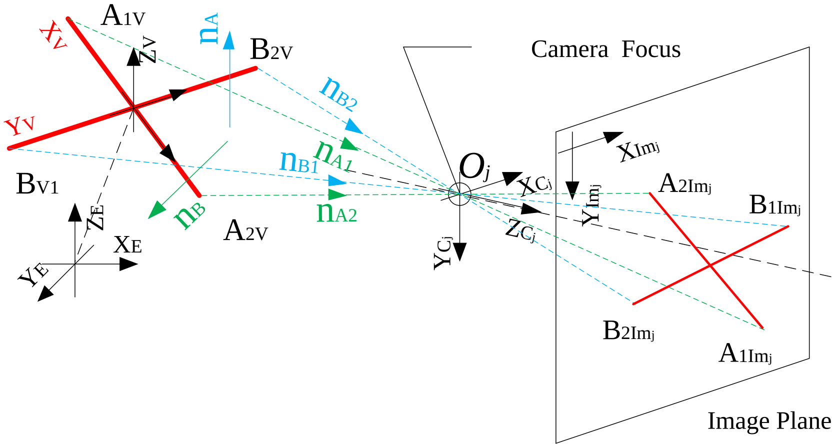


Figure 18: The cylinder projection model

As illustrated in figure 18, rays of light traveled from the LED markers through the focus of the camera at point O. The vehicle axes x_v and y_v, together with O defined two planes. For each plane, its cross section with the image plane (the object's image projection) was extracted through feature detection techniques and used to relocate the target plane. A_1, A_2 and B_1, B_2 were the random points extracted from the line equations, and it was not necessary to locate the end points in that practice.

5. LPS Position Estimation Performance Analysis

The following experiments were aimed to present a holistic analysis of this measurement, including accuracy, precision, and to give the measurement resolution. In static states, we located the vehicle at several fixed positions. For each fixed position, we took measurement for 1000 times for each position and recorded the measured marker centroids in image frame and vehicle positions in earth-fixed frame. These measured vehicle positions are illustrated in figure 19, in comparison with the actual position of vehicle at this fixed point. The x, y, and z coordinate of the recorded positions were demonstrated. From the recorded data, accuracy and precision of these measurements were evaluated.

5.1. Accuracy and Precision

Quantitative measures of accuracy:

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} \quad (5\text{-}1)$$

$$Absolute\ Error\ (E) = Actual\ Value - \bar{x} \quad (5\text{-}2)$$

$$Relative\ Error\ (RE) = \frac{E}{Actual\ Value} \times 100\% \quad (5\text{-}3)$$

Quantitative measures of precision:

$$SD = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \bar{x})^2}{N-1}} \quad (5\text{-}4)$$

$$RSD = \left(\frac{s}{\bar{x}}\right) \times 1000\, parts\ per\ thousand\ (ppt)$$

(5-5)

As consequence, results were listed in table 1. From table 1 and figure 19, we can see this measuring system has a good accuracy and a very high precision. For all 1000 times of measurement, the system has less than 1 mm standard deviation ($\pm 0.1 \sim 0.5\%$ as RSD) and less than $\pm 1.5\ cm$ absolute error ($\pm 1 \sim 5\%$ as RE) in measuring target with largest dimension of one meter.

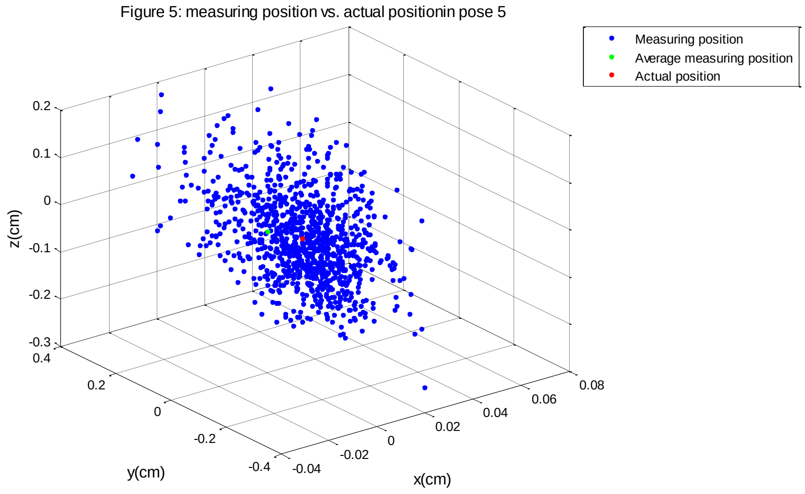


Figure 5: measuring position vs. actual positionin pose 5

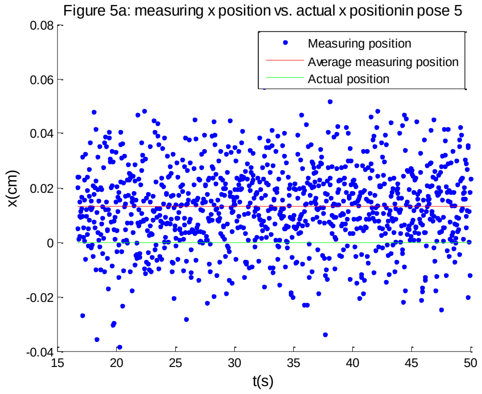


Figure 5a: measuring x position vs. actual x positionin pose 5

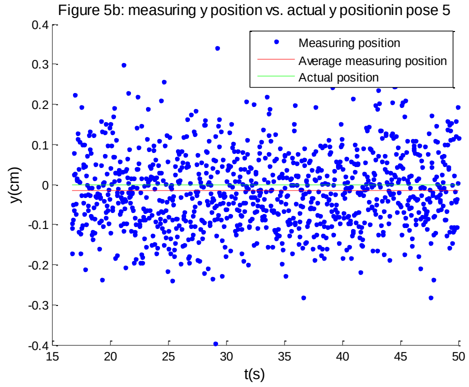


Figure 5b: measuring y position vs. actual y positionin pose 5

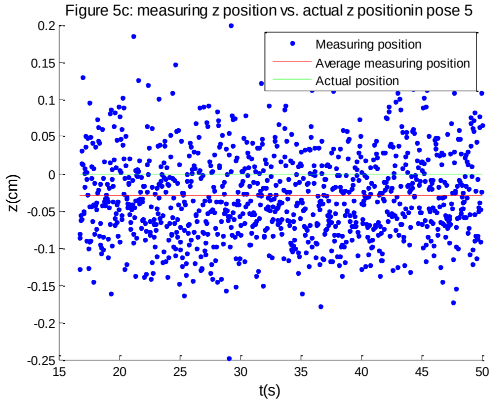


Figure 19: The demonstration sample of a point measured

Node	Position(cm)	x	E(cm)	RE(%)	SD(cm)	RSD(ppt)
1	(-30.48,30.48,0)	-31.475	0.99496	-3.26431	0.019421	-0.63719
2	(0,-30.48,0)	-0.29586	0.295864	-	0.014847	-
3	(30.48,-30.48,0)	29.61005	0.869951	2.854169	0.021657	0.710527
4	(-30.48,0,0)	-31.232	0.75205	-2.46735	0.014566	-0.47789
5	(0,0,0)	0.013005	-0.013	-	0.015019	-
6	(30.48,0,0)	29.67141	0.808586	2.652842	0.021248	0.697114
7	(-30.48,-30.48,0)	-30.6082	0.128229	-0.4207	0.026534	-0.87055
8	(0,-30.48,0)	-0.40153	0.401529	-	0.020782	-
9	(30.48,-30.48,0)	29.78432	0.695681	2.282417	0.027817	0.912641
1h	(-30.48,30.48,13.81)	-31.1727	0.692686	-2.27259	0.017459	-0.5728
2h	(0,-30.48,13.81)	-0.42676	0.426762	-	0.013246	-
3h	(30.48,-30.48,13.81)	29.14828	1.331719	4.369158	0.026435	0.867286
4h	(-30.48,0,13.81)	-31.5306	1.050562	-3.44673	0.015229	-0.49963
5h	(0,0,13.81)	-0.04199	0.04199	-	0.016146	-
6h	(30.48,0,13.81)	29.59713	0.882867	2.896545	0.016822	0.551906

7h	(-30.48,-30.48,13.81)	-31.8028	1.322798	-4.33989	0.021972	-0.72086
8h	(0,-30.48,13.81)	-0.58009	0.580087	-	0.01952	-
9h	(30.48,-30.48,13.81)	29.66912	0.810882	2.660373	0.021533	0.706458

Table 1: LPS X measurement

Node	Position(cm)	x	E(cm)	RE(%)	SD(cm)	RSD(ppt)
1	(-30.48,30.48,0)	31.03961	-0.55961	-1.83598	0.083579	2.742098
2	(0,-30.48,0)	30.83063	-0.35063	-1.15035	0.081127	2.66166
3	(30.48,-30.48,0)	31.19738	-0.71738	-2.35361	0.086528	2.838829
4	(-30.48,0,0)	0.101259	-0.10126	-	0.089429	-
5	(0,0,0)	-0.01565	0.015655	-	0.097412	-
6	(30.48,0,0)	-0.08044	0.080439	-	0.117014	-
7	(-30.48,-30.48,0)	-30.6673	0.187345	-0.61465	0.132685	-4.3532
8	(0,-30.48,0)	-30.5881	0.108085	-0.35461	0.127369	-4.17878
9	(30.48,-30.48,0)	-30.6399	0.159891	-0.52458	0.133414	-4.3771
1h	(-30.48,30.48,13.81)	31.08473	-0.60473	-1.98402	0.062416	2.047761
2h	(0,-30.48,13.81)	30.94134	-0.46134	-1.51358	0.069803	2.290125
3h	(30.48,-30.48,13.81)	31.89073	-1.41073	-4.62838	0.089591	2.939352
4h	(-30.48,0,13.81)	-0.087	0.087004	-	0.09376	-
5h	(0,0,13.81)	0.021666	-0.02167	-	0.08776	-
6h	(30.48,0,13.81)	0.247144	-0.24714	-	0.075463	-

7h	(-30.48,-30.48,13.81)	-30.7609	0.280881	-0.92153	0.110049	-3.61052
8h	(0,-30.48,13.81)	-30.8415	0.36149	-1.18599	0.128355	-4.21112
9h	(30.48,-30.48,13.81)	-31.2251	0.745084	-2.4445	0.122559	-4.02095

Table 2: LPS Y measurement

N	Position(cm)	x	E(cm)	RE(%)	SD(cm)	RSD(ppt)
1	(-30.48,30.48,0)	0.38258	-0.38258	-	0.055872	-
2	(0,-30.48,0)	0.866878	-0.86688	-	0.057341	-
3	(30.48,-30.48,0)	1.272128	-1.27213	-	0.060285	-
4	(-30.48,0,0)	-0.26769	0.267689	-	0.056752	-
5	(0,0,0)	-0.02898	0.028976	-	0.061154	-
6	(30.48,0,0)	0.272253	-0.27225	-	0.071956	-
7	(-30.48,-30.48,0)	-1.07823	1.07823	-	0.072232	-
8	(0,-30.48,0)	-0.55138	0.551375	-	0.070238	-
9	(30.48,-30.48,0)	-0.32909	0.329089	-	0.074573	-
1h	(-30.48,30.48,13.81)	13.8939	-0.08268	-0.59865	0.042462	3.0745
2h	(0,-30.48,13.81)	14.5875	-0.7762	-5.62006	0.046279	3.3508
3h	(30.48,-30.48,13.81)	15.4288	-1.61753	-11.7117	0.058732	4.2525
4h	(-30.48,0,13.81)	13.507	0.304223	2.202719	0.052317	3.788
5h	(0,0,13.81)	13.78	0.031277	0.226457	0.050758	3.6751
6h	(30.48,0,13.81)	14.0848	-0.27351	-1.98037	0.041304	2.9906

7h	(-30.48,-30.48,13.81)	12.6569	1.154385	8.358296	0.056881	4.1184
8h	(0,-30.48,13.81)	13.1454	0.6659	4.82143	0.06458	4.6759
9h	(30.48,-30.48,13.81)	13.2059	0.605388	4.383296	0.062222	4.5051

Table 3: LPS Z measurement

Comparing table 1, 2, 3, we can see x direction measurements had high precision than y and z direction measurements (about one decimal less in RSD), which indicates the system has higher precision in measuring horizontal direction movements. An approximately $-1\ cm$ is observed for almost all of the measurements. That is considered to be the error caused by instrument limitation of the reference measurements. This error might also have contributed to other kind of uncertainty and reduced accuracy.

5.2. Measurement Resolution

Resolution defines the ability to distinguish one reading from another. For this measurement system, the measurement resolution was determined by its accuracy for one measurement, and its repeatability for multi-measurements. The precision of LPS recorded in the experiments was about $\pm 1.5\ cm$ and accuracy less than $1\ mm$, thus this system would identify millimeters movements with errors at centimeters level. Yet, the errors were possibly caused by limitation of human measuring. The measurements resolution was at least $1\ cm$ and at best $1\ mm$.

In this case study, less than one centimeter error in measuring meters object was observed for all 18 nodes, and $1\ mm$ stand deviation for all 1000 times measurement was calculated, which indicates Local Posing System (LPS) has very precision and a good accuracy. The measurement resolution of this system is at least 1 cm.

6. Visual Odometry for MAV Positioning

In this section, the visual odometry systems that were designed for our quadrotor visual positioning will be discussed. Originally, the system was developed under the assumption that the camera attached to the aircraft had the orientation vertically downside, and can be maintained in that pose during operation through control methods. The downside camera constantly produces a video stream about the ground features, which provides the most abundant information for MAV localization. In that case, the visual odometry was only assigned with in-plane movements, and the complexity of a VO was reduced with a concentration in translation only. This application was aimed to measure the local position changes for MAV by observing the ground facts, in a touch-free, GSP denied, and accuracy desired tasks. In comparison with LPS, this VO operated as an on-board camera system, without field of view limitations. Moreover, the VO utilized ground information by taking images under normal exposure condition, which avoid the complexity in marker design and exposure limitation. By considering the vehicle as a rigid body, this VO focused on the location information in the real-world coordinates, but produced the results with higher accuracy in comparison with a GSP system. The VO system included:

- Monocular Visual Odometry: Correspondences are specified in 2D image coordinates and the camera motion is estimated with 2D-to-2D motion methods.
- Stereo Visual Odometry: Correspondences are specified in 3D world coordinates and the camera motion is estimated with 3D-to-3D motion methods.

This section is organized as follows. Section 6.1 and 6.2 present the monocular visual odometry and stereo visual odometry, respectively. Section 6.3 gives the experiments and results. The geometric model for this approach is introduced in section 6.1.1. The first approach presented in section 6.3 was based on DIC. Furthermore, in seeking of a solution with the properties of scale- and rotation- invariance, real-time performance, a feature-based approach was presented in presented in section 6.4.

6.1. Monocular Visual Odometry

The 2D Visual Odometry is based on the Monocular camera. The correspondences are detected with either appearance-based methods or feature-based methods and utilized to calculate a relative camera motion, due to scale ambiguity in monocular methods. In this section, the geometric model for this approach is introduced in section 6.1.1. The first approach presented in section 6.1.2 was based on DIC. Furthermore, in seeking of a solution with the properties of scale- and rotation- invariance, real-time performance, a feature-based approach was presented in presented in section 6.1.3.

6.1.1. Visual Odometry Imaging Model

As shown in figure 20, the ground is setting as a reference plane and assuming camera's lens plane always parallel to the ground, i.e. the camera always looks vertically down. Under the assumption that only translation exist and vertical motion are measured separately, the in-plane camera movements can be estimated by analyzing the concatenation between image pair I_k and I_{k-1}. Let P_1 and P_2 be the correspondences of a ground feature P in frame I_k and I_{k-1}, respectively. The projection of an ideal interest point P together with pixels nearby should form an image feature identifiable enough to determine the ground feature point in the image frame. The corresponding features detected between I_k and I_{k-1} gave a sub-pixel movement, which can be converted into the camera motion by the camera geometric model,

$$\Delta D = (P_k Z_\mathrm{k} - P_{k-1} Z_{k-1})/f_c \tag{6-1}$$

Here $P_k = (x_k, y_k)$ is the coordinate of protection point P_k in the kth frame, and Z_k is the related vehicle altitude. ΔD is the camera horizontal displacement.

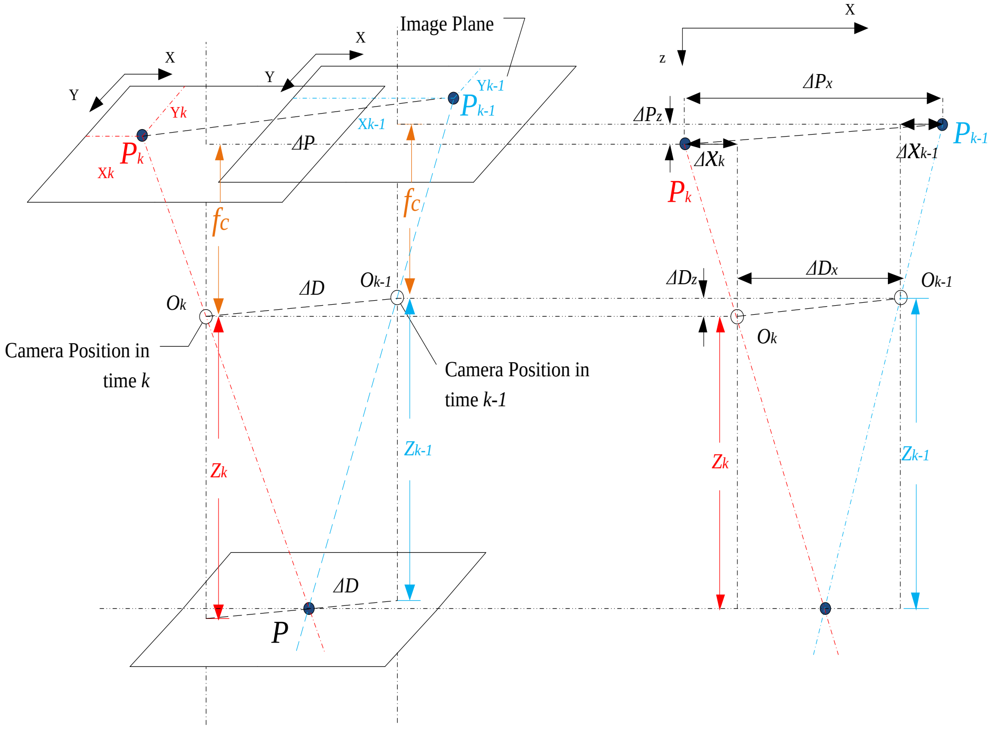
The camera x direction movement is given by,

$$\Delta D_x = (x_k Z_k - x_{k-1} Z_{k-1})/f_c \tag{6-2}$$

Similarly,

$$\Delta D_y = (y_k Z_k - y_{k-1} Z_{k-1})/f_c \tag{6-3}$$

Projections of the ground feature point P are extracted as correspondences in frame I_k and I_{k-1}. By analyzing the pixel movement of correspondences, a simple geometric relationship can be utilized to estimate camera motion for the sub-pixel movement. **(2)** is a y direction side view: Demonstrating the 1D geometric relationship that convert Δx_k into ΔD_x.

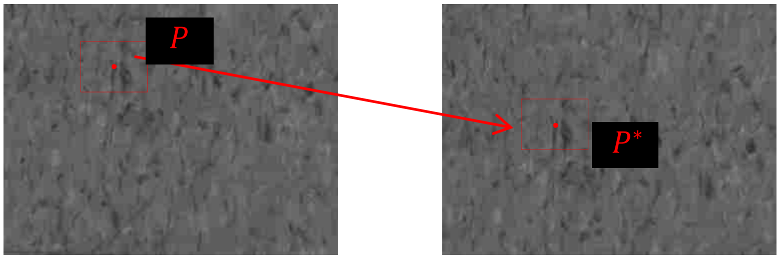


(1) **(2)**

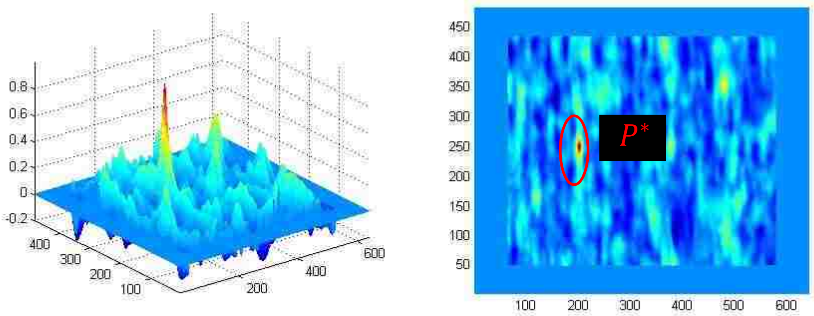
Figure 20: Monocular VO Positioning model

6.1.2. Monocular Visual Odometry with DIC

In this section, the sub-image motion estimation in is utilized to analyze the camera motion and can be measured by the movement of sub-image centroid. In equation 2-4, if $F(P) = G(\tilde{P})$ or $\tilde{P} = P^*$ representing a correspondence of P in image I_k, correlation coefficient r should equal to 1. Practically, due to uncertainties like imaging noise and exposure time difference, correlation coefficient r can hardly reach 1. Thus, the maximum value of r was selected to estimate the position of correspondence P^* in the displaced image. Figure 21 gives an example of image correlation. With the pair of input image I_k and I_{k-1}, a sub-image are formed with the pixels inside of window. Results showed the correlated sub-image in I_k corresponding to I_{k-1}. The correlation coefficient is denoted by the intensity of red. Large intensity of red represents a correlation coefficient close to 1. The correlation coefficient is increased rapidly near point P^* to the peak value. The red color near the correspondence makes it easy to be identified (shown as the red dot in figure 21-4). The demonstration of DIC in finding image correspondences represents a high accurate result. DIC could eventually be utilized by the VO system as a technique for image identification.



[redacted]



[redacted]

Figure 21: Image correlation for VO

The success of image correlation depends on the uniqueness of the sub-image. As shown in figure 21, Structures occurs frequently in the search area not distinctive enough to be located. Red color stands for a correlation coefficient close to 1 representing similar image intensity as the marked region in figure 21. The sub-region occurs frequently and is not distinctive enough to locate through peak value.

Camera motion can be estimated by analyzing subpixel movements which is generated by matching two corresponding features in two or more images. The general idea of DIC feature matching is searching for maximum correlation coefficient between two image pixel value matrixes. Pixel intensity is a number between 1~255 representing the grayscale value of an image and corresponding features tend to have the similarly pixel intensity distribution between two images. Yet, DIC feature matching, as a matrix correlation procedure, is computational expensive. The rest of this section gives a demonstration of the DIC approach done with a one-dimensional camera motion case.

6.1.3. Monocular Visual Odometry with SURF

DIC works satisfactorily with translation of distinctive scenes. Corresponding regions in images with only small camera motion can be matched appropriately by utilizing this algorithm. Yet, it is very sensitive to rotation, scale and affine variations. Take apart its time performance, DIC is still not a suggested descriptor for feature analysis in most real-world and complicated task. Thus more powerful descriptors with good invariance to scale, rotation and illumination changes are desired.

For real time application, one dimensional VO is limited. An algorithm that concentrates on detecting the corresponding features over images is required for real time applications. Considering intensity based method is computational expensive and variant to various facts, a feature based method are suggested for real-time image processing. For this approach, the SURF detector and descriptor were utilized for image analyzing. The corresponding features were matched by computing the correspondence metric matrix S for the detected feature sets $f_{1,m\times 64}$ and $f_{2,n\times 64}$ using sum-of-absolute differences (SAD).

$S(i,j) = \sum_{k}^{64}(norm(f_1(i,k)) - norm(f_2(j,k))^2)$

(6-4)

Where $norm(f_1)$ and $norm(f_2)$ are the normalized feature vectors in f_1 and f_2 respectively. S is a $m \times n$ matrix with its column and row representing the feature vectors in $f_{1,m\times 64}$ and $f_{2,n\times 64}$, respectively. Low values in the scoring matrix indicate similar feature pairs while the high values indicate features with large difference. For computational cost, the features were matched using SAD by searching for the SAD score with the minimal value in both its related column and row in a SAD matrix. This indicates that $f_{1,a}$ is coincident with the corresponding feature vector $f_{1,a}{}'$ of $f_{2,b}$, while its corresponding feature vector $f_{1,b}{}'$ in the other image is coincident with $f_{2,b}$. Therefore, $f_{1,a}$ and $f_{2,b}$ are declared as a match. The outliers in matched features were first removed with epipolar constrain. An example was shown in figure 22. As illustrated, the sub-pixel movements were presented by the displacement between tow corresponding features.

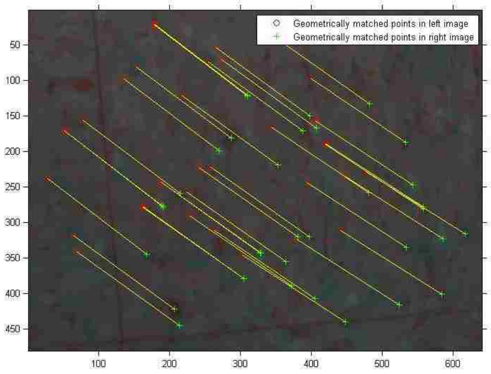


Figure 22: Feature based Correspondence Establishing Demonstration

The block diagram of this system was figure 23, a Monocular camera provided continuous frame stream from. The correspondences between the current image I_k and the pervious image I_{k-1} were detected with SURF feature based method and utilized for a 2D-to-2D motion estimation to determine the cameras trajectory. With the detected features the camera motion is estimate by the monocular VO positioning model.

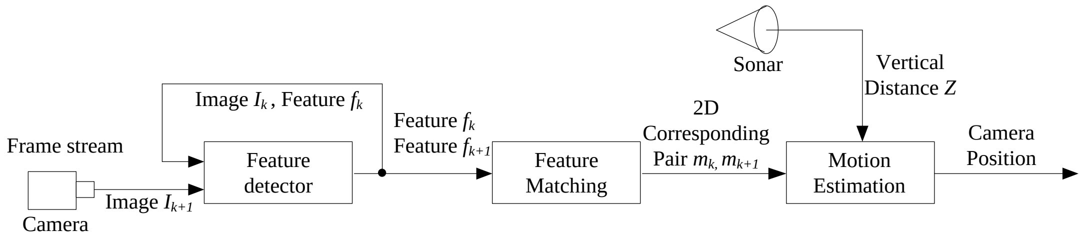


Figure 23: Block diagram of Feature based Monocular VO

6.2. Stereo Visual Odometry

For the 3D Visual Odometry, the correspondences are specified in world coordination. To determine the 3D coordinates of a feature detected in a 2D image, an extra view is provided by a camera with a baseline between them, it forms a Stereo Visual Odometry. As an alternative solution, the 3D position of correspondence can be estimated with depth information from Microsoft Kinect motion capture system which forms the Kinect Visual Odometry.

One efficient solution for scale ambiguity in monocular methods is to give an additional view with large baseline between them. Features are extracted in two views and corresponded for triangulation to get 3D features from 2D correspondences. The on-board stereo computing the camera motion (same as the MAV motion) by determining the aligning transformation of two 3-D feature sets. As introduced in triangulation section, this stereo visual odometry can give the absolute scale from at least three correspondent points in both view, but with errors from interesting rays intersection problem, an approximation solution is provided.

6.2.1. **Stereo Camera**

In this section, the general ideal of stereo camera is discussed. In the stereo camera model that two cameras assembled in arbitrary position and orientation, the block diagram is illustrated in figure 24.

Two cameras provide two sets of continuous frame inputs. The detected SURF features in the left image $I_{l,t+1}$, where the subscript denoted the time this image acquired, are matched with left image $I_{l,t}$. The selected feature vectors in $I_{l,t+1}$ are considered as interest features and match with the right image at current time step $I_{r,t+1}$. The correspondences detected over three frames are illustrated in figure 24, and a triangulation stage is applied to the pair of corresponding components at the current time step to get a set of 3D features, while the related previous corresponding points are triangulate at the previous time step. Therefore, two sets of 3D corresponding features are generated. The execution speed is improved without involving entire depth which is not interested in this case.

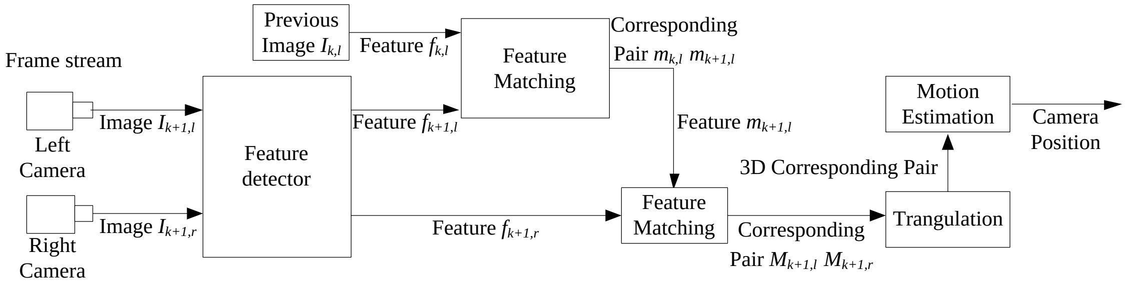
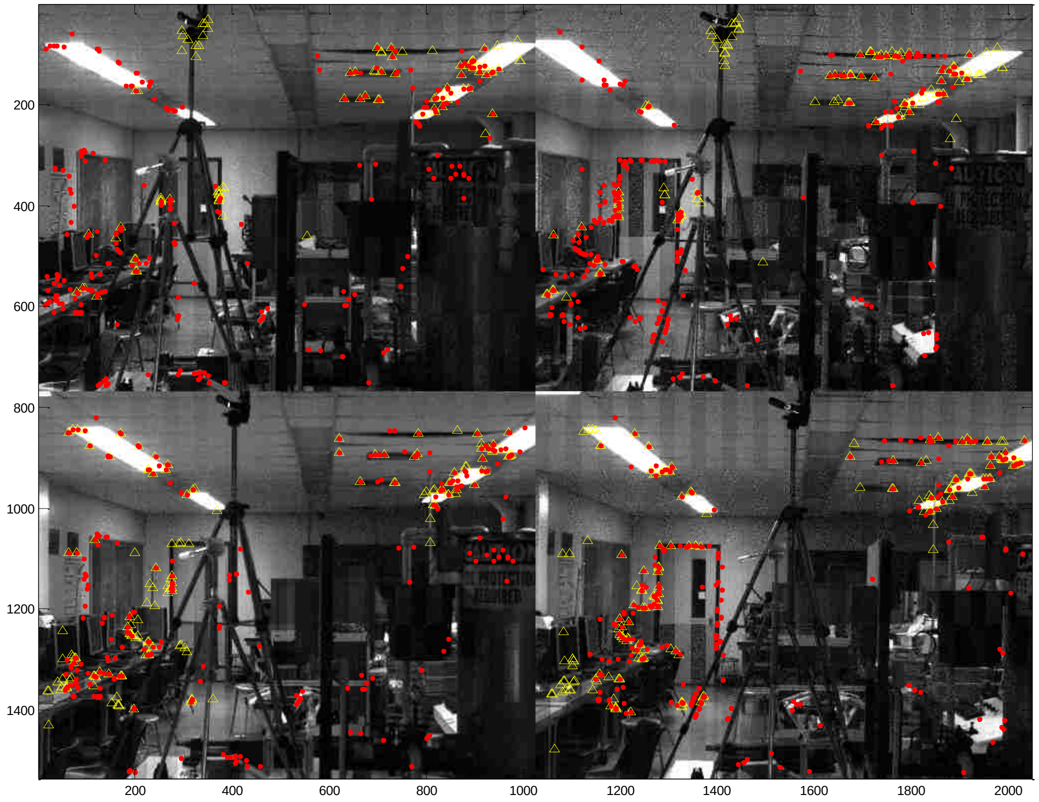


Figure 24: Block Diagram of stereo VO

As illustrated in figure 25, the features detected with SURF were matched over left and right scenes in t_k and t_{k+1}. The rows presented the left and right images, and the columns showed the stereo pair over time in t_k and t_{k+1}. The red dots denoted the matched feature in time domain, and the yellow triangle denoted the matched features in the left and right images. Generally, the features detected over the left and right scenes in t_k and t_{k+1} would be matched to produce the 3D correspondence positions by triangulating the left and right scenes, and generate camera motion by corresponding pair recognition over t_k and t_{k+1}. Yet, a feature matching over four scenes is complex and produces only limited features. As illustrated in figure 25, the inconsistence between the detected features denoted red and yellow could be observed. In consequently, in image feature matching over multiple scenes consequently decrease the number of 2-D correspondences detected.

6.2.2. Kinect Visual Odometry

In this section, an alternative solution for stereo VO would be discussed. The Microsoft Kinect introduced a low-cost, stable and high speed solution for robot positioning applications such as indoor mapping and navigation, real world coordinates measurement, target tracking and motion capturing. The principle of Kinect depth sensor, different from binocular camera stereo triangulation, is more similar to sonar sensors. The light project from IR light, called structured light, is attenuated by distance and reflects signals with multiple intensities onto the IR depth camera to generate objects depth



[redacted]

information.

This technique demonstrates a remarkable advance in computer vision and has various robotics applications. As an alternative solution for stereo vision, the Kinect sensor providing a frame rate of about 30fps in capturing depth and color images and a more than 10Hz frequency in generating real world coordinates data for each pixel from depth information at MATLAB environment, as illustrated in figure 26.

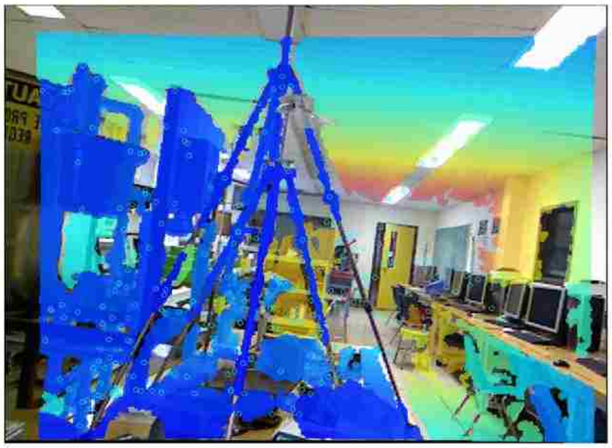


Figure 26: Depth map aligned with the color map

The block diagram of Kinect VO is demonstrated in figure 27. With the color map stream, a feature based image processing is operated to detect and match the 2D image feature between I_k and I_{k-1} in the grayscale image. The 3D position (x, y, z) of most pixel in the field of view is generated with the depth information from Kinect IR projector. By aligning the color image and the depth map, the real world coordinates of 2D corresponding features with respect to camera are produced. The feature matching only involves two frames at each time. The idea of image processing of a Kinect VO is demonstrated in figure 28.

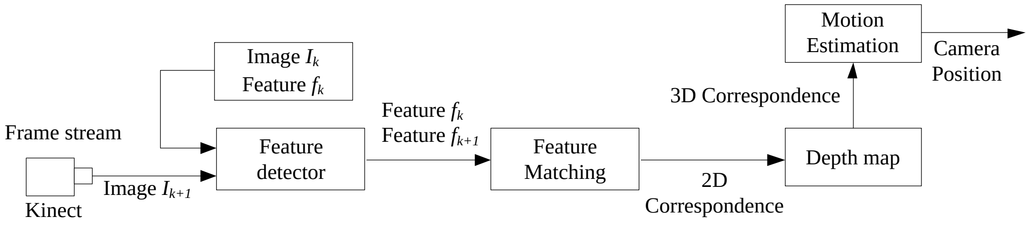
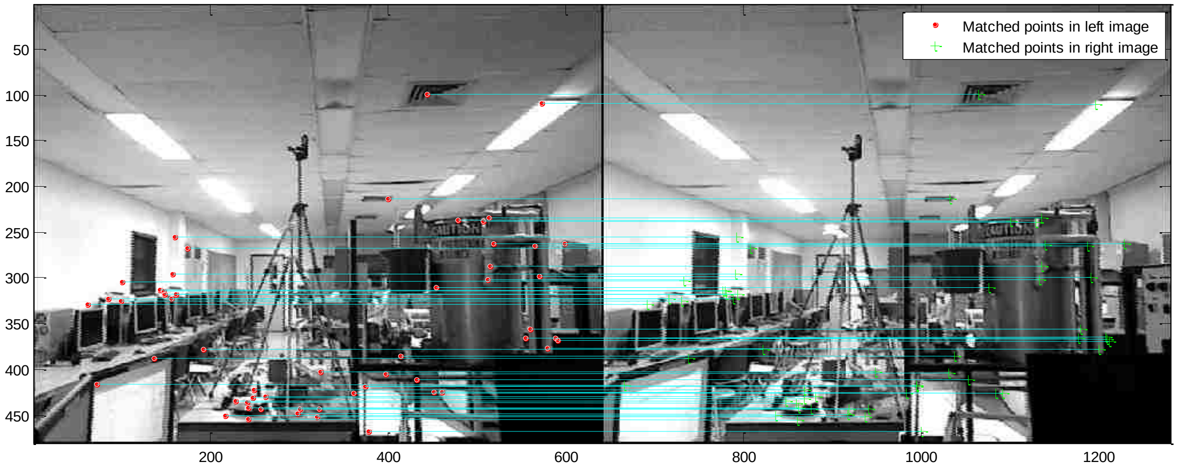
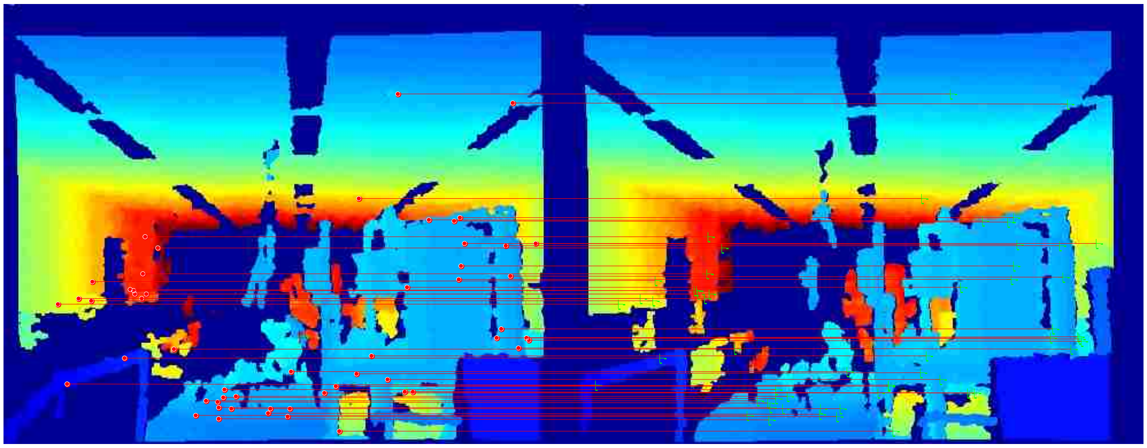


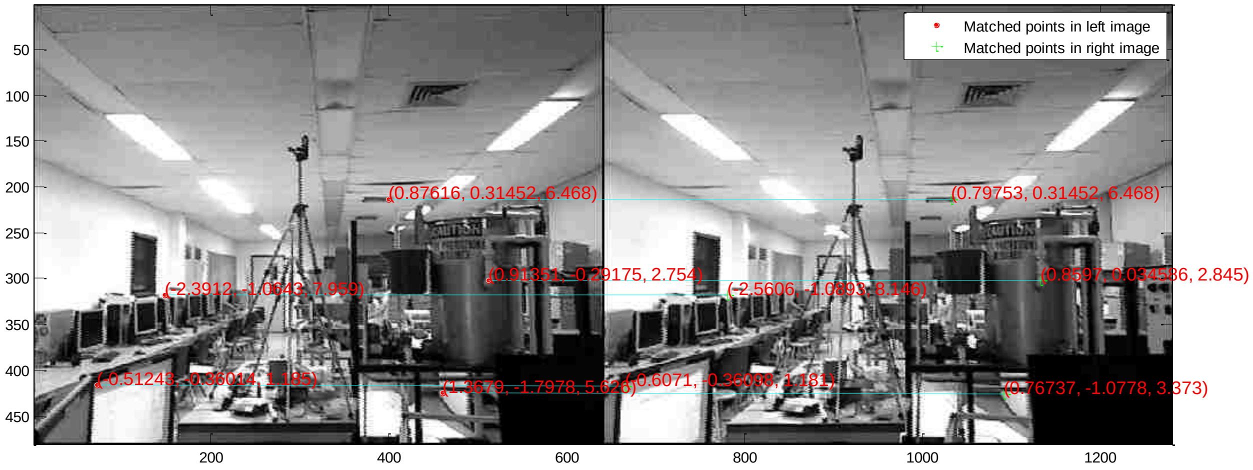
Figure 27: Block diagram of Kinect VO



[redacted]



[redacted]



[redacted]

6.2.3. 3D-to-3D Motion Estimation

In the stereo vision case the camera transformation M_k is determined by two 3D feature point sets X_t and X_{t-1} generated through triangulation at two time steps. X_t and X_{t-1}can be considered as beacons with their coordinates marked in camera frame (set left camera as the default coordinates) detected at time t and $t-1$. The transformation between X_t and X_{t-1} is given by,

$$X_t = M_k X_{t-1} + \sigma \tag{2-33}$$

Or

$$\mathrm{X_t = R_k X_{t-1} + t_k + \sigma} \tag{2-34}$$

By minimize the noise term σ, the transformation $\mathrm{M_k}$ can therefore be determined with the constraint,

$$\mathrm{M_K = argmin \sum_i \left\| \widetilde{X}_t^i - M_k \widetilde{X}_{t-1}^i \right\|^2} \tag{2-35}$$

Where i denotes the ith feature. The minimal case solution requires at least three noncollinear correspondences. For $\mathrm{n} \geq 3$ correspondences case, the possible solution is to compute the translation part $\mathrm{t_k}$ of $\mathrm{M_k}$ by decoupling the parameters by centering 3D feature sets about their centroids and solve for the rotation part $\mathrm{R_k}$ that best aligns the point sets with SVD, which demonstrated to be the best solution [24],

$$\mathrm{t_k = \overline{X}_k - R\overline{X}_{k-1}} \tag{2-36}$$

$$\mathrm{R_k = V^T diag(1,1, det(U)\, det(V)\, U} \tag{2-37}$$

Where $\mathrm{\overline{X}_k}$ stands for the geometric centroid of point set $\mathrm{X_k}$, and $\mathrm{U, S, V}$are the SVD given by,

$$\mathrm{USV^T = \frac{1}{N}(X_{k-1} - \overline{X}_{k-1})(X_k - \overline{X}_k)^T} \tag{2-38}$$

6.3. Experiment and Results

In this section, the VO was attached to the test bench and evaluated with the membrane potentiometer sensor providing reference measurements. Section 6.3.1 presents the membrane potentiometer sensor measurement performance analysis and the reference measurements were verified. In section 6.3.2, the VO experiments and results were presented.

6.3.1. Linear Positioning Sensor

In this section, a measurement performance analysis was presented to verify the reliability of the membrane potentiometer sensor as a reference measurement system. Several random, discrete points were assigned and measured by both sensor measurement and human measurement. The analog data generated by this potentiometer sensor was read with an Arduino Uno microcontroller board and transferred into MATLAB. The acquired analog data was converted into position information, measured in mm, using the calibration curve computed in section 3.3.

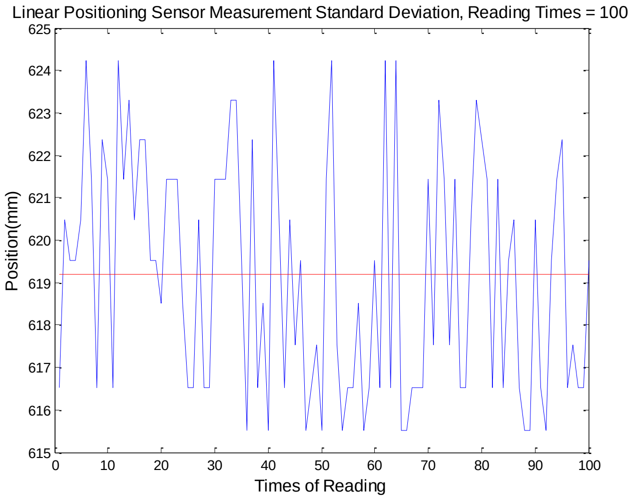


Figure 29: Linear positioning sensor measurement standard deviation

As illustrated in figure 29, about ±5 mm fluctuation can be observed. To minimize this noise and produce a reliable measuring, several measurements based on various times of analog reading were taken, and the average values were considered as the result for each measurement. The results were illustrated in figure 30 and table 4. The execution time, standard deviation, and absolute error were quantified in comparison with the data acquired with five times of reading. As shown in this figure, heavier computation burden was observed with an increasing in reading times, while the accuracy and precious were not improved significantly. Figure 30 and table 4 gives the 0.1s sampling time with accuracy less than 1mm as a suggested sampling time for this membrane potentiometer sensor.

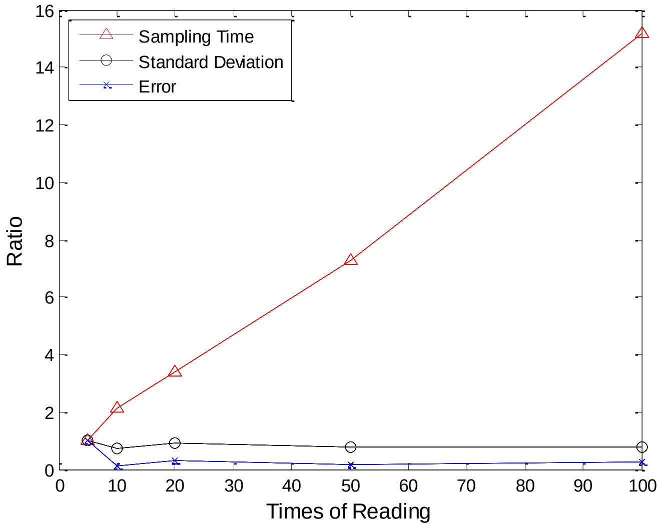


Figure 30: Membrane potentiometer sensor sampling time test

	1	2	3	4	5
Standard Deviation (mm)	3.45	2.46	3.19	2.60	2.60
Absolut Error (mm)	1.88	0.20	0.53	0.32	0.48
Sampling Time (s)	0.047	0.100	0.159	0.342	0.713

Table 4: Membrane potentiometer sensor sampling time test

With the 0.1s sampling time, several measurements were taken to evaluate the sensor measurement performance. The results were listed in table 4. At each point, the analog voltage was converted into a position measurement via the calibration curve shown in figure 30. Table 4 gives the accuracy of ten measurements. From the result, a less than 1mm resolution for this calibrated system was observed. And this system can provide a satisfactory accuracy as a reference measurement system for the VO experiment.

Position (cm)	Analog	Position(cm)	SD(mm)	Error (mm)	Error (%)
2.7	66.617	2.7051	2.4163	0.051	0.188889

12.8	164.868	12.7285	2.3725	-0.715	-0.55859
24.1	276.616	24.0771	2.2077	-0.229	-0.09502
33.9	368.802	33.9396	2.5520	0.396	0.116814
41.5	439.607	41.4922	2.4764	-0.078	-0.0188
51.7	541.56	51.6618	2.4698	-0.382	-0.07389
58.6	612.935	58.5662	2.6843	-0.338	-0.05768
71.8	753.455	71.7865	2.5783	-0.135	-0.0188
80.3	846.428	80.2507	2.3818	-0.493	-0.06139
86.6	919.026	86.6066	2.4265	0.066	0.007621

Table 5: Membrane potentiometer linear positioning sensor calibration test

6.3.2. Experiments and Results

In this section, the VO approaches are experimented in comparison with membrane potentiometer linear positioning sensor. The cameras were attached onto the test bench and provided with a linear motion, which can be measured accurately by the calibrated membrane potentiometer sensor.

The monocular VO operated as demonstrated in figure 20 and the block diagram in 23. The altitude of the camera was measured with sonar sensor. And the x and y direction motion of the camera are detected with the sub-pixel movements. For the DIC approach, the on-dimensional experiment was directed. In one-dimensional case, camera only has the x-direction motion. The DIC procedure can be reduced significantly with one-dimensional row by row scan. Image $\mathrm{I}_{\mathrm{k}-1}$ and I_{k}were taken at $\mathrm{t}_{\mathrm{k}-1}$ and t_{k}. The peak value of the correlation coefficient was determined by searching I_{k} for the corresponding sub-image of $\mathrm{I}_{\mathrm{k}-1}$. This stage will be repeated each iteration for new images to estimate the displacement of camera during t_{k}. In order to evaluate it real-time performance, the camera was assembled to the test bench and move it linearly. The position information converted from both correlation results and membrane potentiometer sensor are recorded each iteration. Figure 31 gives the demonstration of on-line one-dimensional digital image correlation.

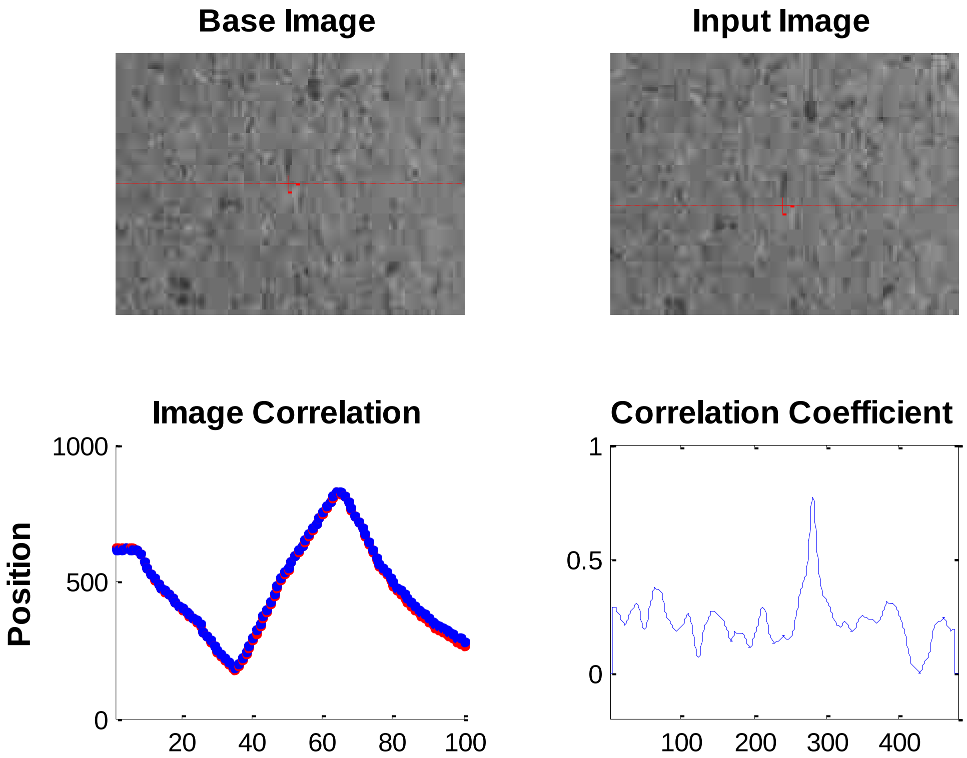


Figure 31: One-Dimensional Digital Image Correlation

The recorded results from visual odometry and linear position sensor are illustrated in figure 32. It can be seen that a good match between the VO curve (red line) and the true value which representing a good accuracy of the visual odometry system. Yet, a tendency of separation at the later part of the two curves can also be observed which was caused by error accumulation. Note that, the system realized camera positioning through visual information by analyzing two images taken with a time interval. In other words, the displacements of camera during a time interval ware measured and accumulated to get the camera motion.

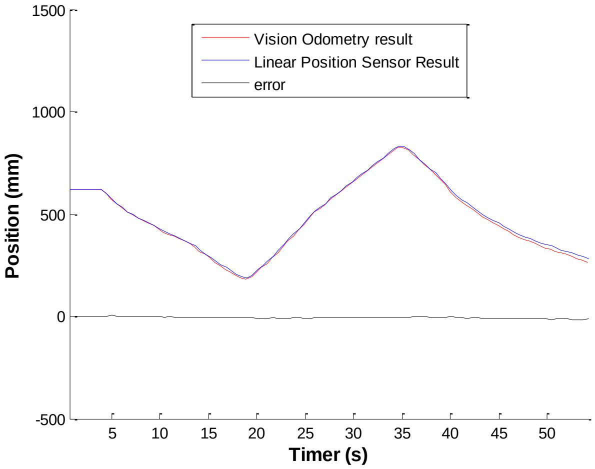


Figure 32: Results comparison for visual odometry with DIC

The feature-based VO is built on SURF algorithm and operates as the camera model in figure 20. The camera was provided with x and y direction movements which were recorded by linear sensor as the blue curves in figure 33. The red curves are the results measured with monocular VO. In comparison with the linear sensor measurement, the VO measuring presents a good estimation of the camera motion.

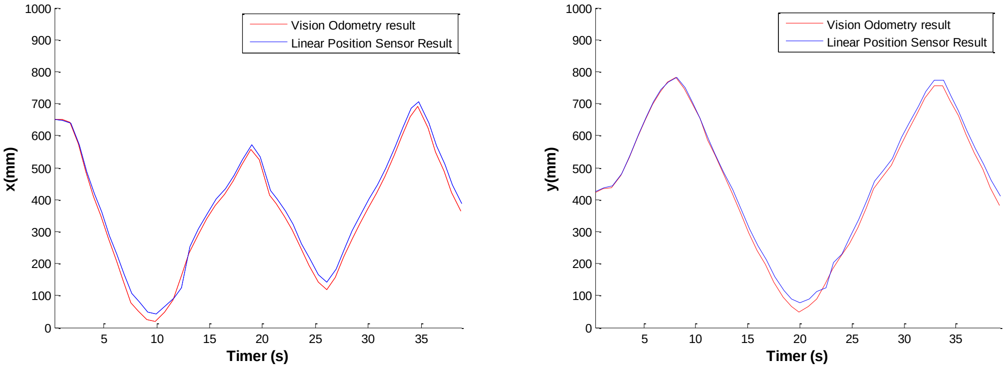


Figure 33: 2D Visual odometry with Feature based method

For the Kinect VO, the results are not as accurate as the monocular VO. As illustrated in figure 34, the displacements of camera measured with linear positioning sensor are denoted as blue, and the VO measurements are in red. The VO gives an approximately estimation of the camera movements, yet is not accurate. To study this error, the data in figure 35 was taken in an experiment. In this figure, the red and blue curves present the recorded path of the camera. The blue curve is the actual path measured with linear positioning sensor. The error accumulated over time and led to the deviation in the recorded path. This kind error may have a relation with the feature number detected and matched in the image. The camera produce image stream and extract feature for each frame automatically. The number of the feature is related to various facts and differs over images. Figure 36 gives VO error in comparison with the feature numbers. As illustrated, the large error occurred frequently when only a low number of features were involved in the motion estimation. At last, an offline approach for this system is presented. In this experiment, the feature detection stage was adjusted to produce enough features for motion estimation in trade off speed. 500~1000 features were involved for each frame. Consequently, as illustrated in figure 37, the accuracy of this measuring system was improved.

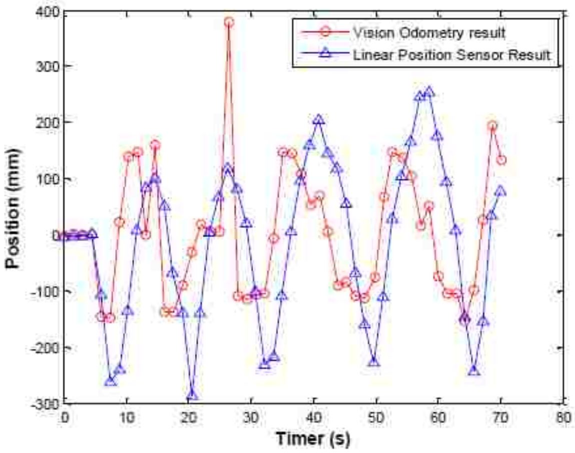


Figure 34: Camera Displacement comparison

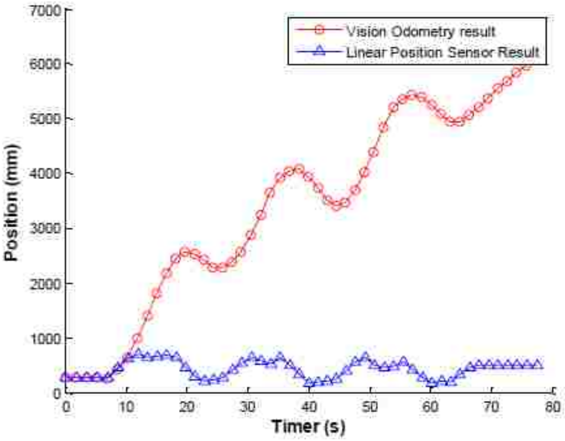


Figure 35: Camera path tracking

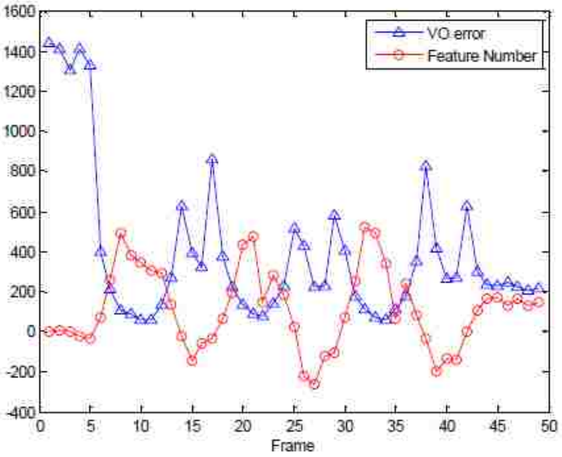


Figure 36: A comparison of measurement error with feature number

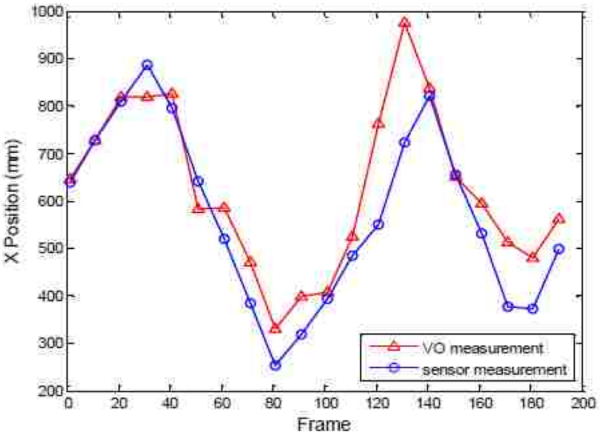


Figure 37: Offline Kinect positioning result

7. Conclusion

In this work, a description and performance analysis of three vision-based systems have been presented. Those systems, consisting of a local positioning system (LPS) and a simplified monocular visual odometry (VO) were designed for the MAV localization tasks. The LPS is aimed to provide the 6 DoF information of a quadrotor marked with LED markers, based on two local cameras. The monocular VO was designed for the MAV positioning by analyzing the in-plane motion. As a solution of 6 DoF problems, the Microsoft Kinect was utilized to generate the 3D correspondences to provide an alternative solution for stereo matching.

The LPS was utilized for the quadrotor pose measuring. The LED markers were employed for vehicle feature identification in the image frame. As a conclusion of the LPS performance analysis, this system has high accuracy and precision. Yet this system has its limitations. As a simplified feature detection stage, the LED markers are expected to be the only features that can be detected in the image. However, several drawbacks in the system have lowered this expectation. First, the image was taken under low exposure condition; hence noises from other light source and reflections would disturb the marker identification. Second, the local positioning system built up with local cameras has a limited field of view, and objects beyond this field will not be seen. This puts limitation on the scope of activities of the MAV. Last, marker overlapping (one marker blocks another) happened during the experiments. Possible suggestions to mitigate the problem are to adjust the cameras downwards, and make LPS concentrate on the local study of MAV pose information.

In the section 6, a 2D visual odometry model and a 3D visual odometry model are discussed. For the 2D visual odometry, the intensity based method and feature based method were applied. The intensity based methods establishes correspondence based on cross-correlation over a sub region. As a consequence, it is very sensitive to view angular, illumination and has low invariance to rotation, scale and affine changes. Although it has good accuracy in measure transformation, the execution speed limited its real applications. The purpose of this work is to design a vision-based odometry system supporting real-time robot applications. Thus, the feature based methods which have a good rotation and scale invariance were implemented. SURF extracts the feature with a satisfied repeatability in scale space and descripts them with 64D descriptors. In comparison with SIFT, SURF has a good real time performance.

In the monocular camera visual odometry model, the camera was adjusted to look vertically down to the ground. The ground information was utilized to estimate the camera and direction movement. This on-board camera system is presented without field of view limitations; the imaging processing algorithm has a good tolerance for imaging noise and view point changes. This system simplified the monocular camera visual odometry into the 2D motion problem, and can be utilized for MAV position by using ground feature.

The stereo visual odometry is composed of two cameras providing an additional view and gives a solution for scale ambiguity in monocular methods. The 2D feature points were converted into 3D point through triangulation. The cameras took new image pairs, and the detected correspondence between the current image pair and the pervious image pair is triangulated to generate a 3D features. Yet, feature detection has a limited feedback that only limited features can be matched over multiple images On the other hand, the Kinect system generated depth map through IR cameras. The depth information can be easily converted into 3D position for most pixels in the field of view, providing an alternative solution for stereo triangulations. This system generated 6 DoF information with a 3D-to-3D feature detection method. This system is not appropriate for high accurate measurement. Yet, as an 3D VO, it can be used as an alternative solution for stereo VO. Future developments aim to improving the motion estimation algorithms for this VO to get more accurate position and orientation.

Literature Cited

[1] Moravec, H.; , “Obstacle avoidance and navigation in the real world by a seeing robot rover,” Ph.D. dissertation, Stanford Univ., Stanford, CA, 1980

[2] Ohya, I.; Kosaka, A.; Kak, A.; , "Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing," Robotics and Automation, IEEE Transactions on , vol.14, no.6, pp.969-978, Dec 1998

[3] Se, S.; Lowe, D.G.; Little, J.J.; , "Vision-based global localization and mapping for mobile robots," Robotics, IEEE Transactions on , vol.21, no.3, pp. 364- 375, June 2005

[4] Arturo L. Rankin; Huertas, A.; Matthies, L. H.; , "Stereo vision based terrain mapping for off-road autonomous navigation", Proc. SPIE 7332, Unmanned Systems Technology XI, 733210, April 2009

[5] Nister, D.; Naroditsky, O.; Bergen, J.; , "Visual odometry," Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on , vol.1, no., pp. I-652- I-659 Vol.1, 27 June-2 July 2004

[6] da Costa Botelho, S.S.; Drews, P.; Oliveira, G.L.; da Silva Figueiredo, M.; , "Visual odometry and mapping for Underwater Autonomous Vehicles," Robotics Symposium (LARS), 2009 6th Latin American , vol., no., pp.1-6, 29-30 Oct. 2009

[7] Ahrens, S.; Levine, D.; Andrews, G.; How, J.P.; , "Vision-based guidance and control of a hovering vehicle in unknown, GPS-denied environments," Robotics and Automation, 2009. ICRA '09. IEEE International Conference on , vol., no., pp.2643-2648, 12-17 May 2009

[8] Stowers, J.; Hayes, M.; Bainbridge-Smith, A.; , "Altitude control of a quadrotor helicopter using depth map from Microsoft Kinect sensor," Mechatronics (ICM), 2011 IEEE International Conference on , vol., no., pp.358-362, 13-15 April 2011

[9] Saripalli, S.; Montgomery, J.F.; Sukhatme, G.S.; , "Visually guided landing of an unmanned aerial vehicle," Robotics and Automation, IEEE Transactions on , vol.19, no.3, pp. 371- 380, June 2003

[10] Altug, E.; Ostrowski, J.P.; Taylor, C.J.; , "Quadrotor control using dual camera visual feedback," Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on , vol.3, no., pp. 4294- 4299 vol.3, 14-19 Sept. 2003

[11] Scaramuzza, D.; Fraundorfer, F.; , "Visual Odometry [Tutorial]," Robotics & Automation Magazine, IEEE , vol.18, no.4, pp.80-92, Dec. 2011

[12] Nister, D.; , "An efficient solution to the five-point relative pose problem," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.26, no.6, pp.756-770, June 2004

[13] Nister, D.; , "Preemptive RANSAC for live structure and motion estimation," Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on , vol., no., pp.199-206 vol.1, 13-16 Oct. 2003

[14] Fraundorfer, F.; Scaramuzza, D.; , "Visual Odometry : Part II: Matching, Robustness, Optimization, and Applications," Robotics & Automation Magazine, IEEE , vol.19, no.2, pp.78-90, June 2012

[15] C. Harris and J. Pike, “3d positional integration from image sequences,” in Proc. Alvey Vision Conf., 1987, pp. 233–236.

[16] Rosten, E.; Porter, R.; Drummond, T.; , "Faster and Better: A Machine Learning Approach to Corner Detection," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.32, no.1, pp.105-119, Jan. 2010

[17] David Lowe. "Distinctive image features from scale-invariant keypoints". International Journal of Computer Vision, 60(2):91–110, 2004.

[18] Stephen Se, David Lowe, and James Little. "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks". The International Journal of Robotics Research, 21(8):735–758, 2002

[19] Mikolajczyk, K.; Schmid, C.; , "Indexing based on scale invariant interest points," Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on , vol.1, no., pp.525-531 vol.1, 2001

[20] H. Bay, T. Tuytelaars, and L.booktitle Van Gool. "Surf: Speeded up robust features". In 9th European Conference on Computer Vision, pages 404–417, 2006.

[21] Zhigang Bing; Yongxia Wang; Jinsheng Hou; Hailong Lu; Hongda Chen; , "Research of tracking robot based on SURF features," Natural Computation (ICNC), 2010 Sixth International Conference on , vol.7, no., pp.3523-3527, 10-12 Aug. 2010

[22] G. Vendroux and W. Knauss, "Submicron deformation field measurements: Part 2. Improved digital image correlation", Experimental Mechanics, vol. 38, 86-92, 1998-06-17.

[23] Rakprayoon, P.; Ruchanurucks, M.; Coundoul, A.; , "Kinect-based obstacle detection for manipulator," System Integration (SII), 2011 IEEE/SICE International Symposium on , vol., no., pp.68-73, 20-22 Dec. 2011

[24] Lu Xia; Chia-Chih Chen; Aggarwal, J.K.; , "Human detection using depth information by Kinect," Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on , vol., no., pp.15-22, 20-25 June 2011

[25] Chan-Soo Park; Sung-Wan Kim; Doik Kim; Sang-Rok Oh; , "Comparison of plane extraction performance using laser scanner and Kinect," Ubiquitous Robots and Ambient Intelligence (URAI), 2011 8th International Conference on , vol., no., pp.153-155, 23-26 Nov. 2011

[26] Smisek, J.; Jancosek, M.; Pajdla, T.; , "3D with Kinect," Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on , vol., no., pp.1154-1160, 6-13 Nov. 2011

[27] Hideki Shirai, Kinect for Matlab. Available at: http://www.mathworks.com/matlabcentral/linkexchange/links/2718-kinect-for-matlab

[28] Shotton, J.; Fitzgibbon, A.; Cook, M.; Sharp, T.; Finocchio, M.; Moore, R.; Kipman, A.; Blake, A.; , "Real-time human pose recognition in parts from single depth images," Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on , vol., no., pp.1297-1304, 20-25 June 2011

[29] D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes", Pattern Recognition, Vol.13, No.2, p.111-122, 1981

[30] H. Longuet-Higgins, “A computer algorithm for reconstructing a scene from two projections,” Nature, vol. 293, no. 10, pp. 133–135, 1981

[31] Alismail, Hatem; Browning, Brett; and Dias, M. Bernardine, "Evaluating Pose Estimation Methods for Stereo Visual Odometry on Robots" (2010), Robotics Institute. Paper 745

[32] Besl, P.J.; McKay, H.D.; , "A method for registration of 3-D shapes," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.14, no.2, pp.239-256, Feb 1992

[33] The Imaging Source DMM 22BUC03-ML. Available at: http://www.theimagingsource.com/en_US/products/oem-cameras/usb-cmos-mono/

[34] Kinect sensor as shown at the 2010 Electronic Entertainment Expo. Available at: http://zh.wikipedia.org/wiki/File:Kinect_Sensor_at_E3_2010_%28front%29.jpg

[35] Microsoft Kinect structure. Available at: http://www.neurogami.com/presentations/KinectForArtists/

[36] SoftPot membrane potentiometer sensor. Available at: http://www.spectrasymbol.com/potentiometer/softpot

[37] Dzaba, A. ; Schuster, E,; , "A Low-Cost, Scalable, Vision-Based Local Posing System for Applications in Mobile Robotics"

Vita

Yufei Qi was born in Gaoyou, Jiangsu, China, on August 29, 1987 to Yuejin and Yan Qi. As an undergraduate, Yufei pursued a B.S. degree in Engineering Mechanics at Tongji University in Shanghai, China in June of 2010. He then attended Lehigh University in Bethlehem, Pennsylvania. Yufei is currently seeking M.S. degree in Mechanical Engineering with an anticipated graduation date of Jan 2013.

Permanent Address:

XingHuaYuan, NO.40, 102

Gaoyou, Jiangsu, China

225600

E-mail:

chee.july@gmail.com