

**ACCURATE MOTION AND POSE ESTIMATION ALGORITHM FOR
RIGID OBJECTS**

By

Mehmet Akif Alper

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering — Doctor of Philosophy

2022

ABSTRACT

Motion and Pose estimation are a widely studied research problem area in the field of computer vision. Despite major progress that has been made in pursuing this area, pose estimation is still a largely unsolved problem. Many challenging, practical, and real-world applications need to be taken into consideration when developing new estimation solutions for this research area. These challenges include low Signal to Noise Ratio (SNR), local optimal solutions, and other related practical issues. Therefore, accurate and robust pose estimation solutions are needed, especially for time-critical and sensitive applications such as medical surgeries and space robot applications.

In this thesis, we focus on one important class of pose estimation solutions that are based on fusion techniques. We focus on solutions that exploit depth cues, color information, and wearable sensor, which can be fused to enhance accuracy and robustness of the pose estimation system. Furthermore, we explore graphical inference models, such as Loopy Belief Propagation methods, that can enhance pose and motion estimation accuracy. Consequently, in this thesis, we present our findings regarding various fusion techniques and graphical inference methods to solve the pose estimation problem. We further apply these techniques in the estimation of Parkinson's Disease tremors, rigid object pose estimation, and robot localization. Additionally, we have developed a prototype by using Kinect v2, which is capable of tracking motion of Parkinson's Disease patients. The proposed system can lead to cost-effective and efficient motion tracker for medical applications.

Copyright by
MEHMET AKIF ALPER
2022

I would like to dedicate this thesis to my wife, my children, and my parents: Gozde Ummahan Alper, Huma Marziya Alper, Ali Mete Alper, Aysen Alper and Nazif Alper for supporting me endlessly towards achieving this goal.

ACKNOWLEDGMENTS

I would like to thank Professor Daniel Morris who guided many early aspects of my research that led to the completion of this thesis. I would also like to acknowledge my current advisor, Professor Hayder Radha, who has supported me during the later stages of my PhD studies. I would like to acknowledge my lab members and colleagues, Muhammad Saif Imran and Mehmet Cagri Kaymak. I would like to thank them for being a part of this long journey, by engaging me through many inspiring discussions, helping me throughout this intellectual journey, and making my PhD career a more enjoyable and fun experience.

TABLE OF CONTENTS

Chapter 1 Introduction and Motivation	1
1.1 Problem Formulation	5
1.2 Thesis Organization	7
Chapter 2 Related Study	8
2.1 Summary of the Chapter	17
Chapter 3 Accurate Tremor Detection and Quantification Algorithm	18
3.1 Introduction	18
3.2 Kinect Based Sensing	19
3.3 Method	20
3.4 Evaluation Techniques	27
3.5 Results	34
3.6 Discussion	44
3.7 Conclusion	46
3.8 Alignment Method	47
3.9 Summary of the Chapter	51
Chapter 4 Robot Localization Through Graphical Inference Method	53
4.1 Introduction	53
4.2 Summary of the Chapter	56
Chapter 5 Unstructured 3D Shape Matching Algorithm	57
5.1 Rigid Object Pose Estimation	57
5.2 Proposed Framework	62
5.3 Experiments	68
5.4 Results	70
5.5 Summary of the Chapter	75
Chapter 6 Experimental Results	78
6.1 Results And Discussion	78
6.2 Summary of the Chapter	80
Chapter 7 Conclusion and Future Work	81
7.1 Conclusion	81
7.2 Future Work	82
BIBLIOGRAPHY	84

Chapter 1

Introduction and Motivation

Computer vision has been actively evolved to abundant and modern applications for automated motion and pose estimation, human-computer interaction, augmented reality, perception, sport analysis, and other fields. Therefore, many algorithms are proposed to find solutions to applications. Abundance of real world applications require automated motion and pose estimation that helps to autonomous driving.

Motion and pose estimation refers to the task of estimating pose parameters that enable to find location of objects in real world. Pose estimation is computed to find trajectory, motion and relative pose from frames. Pose estimation is widely studied research topic and applicable to humans and rigid objects. Though many researchers studied pose estimation algorithms, it still remains a open research area because there is limited number of training data sets, generalization problems, blurring and illumination problems, occlusions, cluttering, sensor noises, incorrect pose matches etc. Therefore, research community deeply study and propose new algorithms to enhance accuracy and robustness of motion and pose estimation applications.

Parkinson's Disease (PD) is a progressive nervous system disorder that creates drawbacks for PD patients and societies, decreases life quality of patients and costs to societies due to treatment and work loss. Therefore, it is required to track and evaluate some of symptoms PD patients. It is very common that PD causes tremor symptoms on patients especially on

their arms and legs, and PD tremors can come and go. Human motion and pose estimation requires to estimate human body parts configuration from a multiple frames, so tremor motion can be tracked. As average human age increases, number of PD patients increase and severeness of PD can be increased. There are many different devices and algorithms that collect and estimate tremors on PD patients. Wearable units required charge and may not be stationary on patient. Wearable units can change and add some noise on tremor measurements. Therefore, there is a requirement for continual and remote monitoring that can provide a feedback measures for health professionals regarding stage and severeness of PD. Camera systems can be utilized for remote monitoring. Many algorithms developed for N-joint kinematic skeleton model and seek to estimate pose of the human body where each vertex represents a human joint. We have focused to estimate limb tremor motion where tremors are most explicit and severe. Tremor motion analysis can provide feedback measure for doctors to check progress of PD.

Pose estimation is an important and widely studied research topic for robot applications as well. Robots can work autonomously, and complete some tasks where humans may not be able to. Some robots work at the space duty to grasp and release objects, some robots work on micro surgeries to assist doctors where it is required to make sensitive movement on defective human skins and organs, some robots can haul heavy objects etc. In order to complete those tasks, a robot needs to work autonomously, know relative pose of the objects, and position of itself in the outside environment. Therefore, robots need to compute pose of the objects in outside environment and robot needs to find its position at the environment continuously. Robot pose estimation and position localization methods can provide relative position information of objects and location of robots which is important process of

autonomous applications.

There are several different type of motion capture devices that collects raw pose or motion data as wearable units like accelerometers, magnetometers, and gyroscopes; remote units as camera sensors, laser scanners, lidars, and radar devices etc. Raw data can be processed, motion and pose information can be computed. Though accelerometers and gyroscopes can be utilized to detect orientation of the objects, they have some drawbacks for single use. Accelerometers can give good collection of orientation for small motions, but accelerometers can include noisy measurements for rapid movements due to noise at high frequencies. Gyroscopes are effective for rapid motion and rotations, but measurements drift which causes accumulation of error. These drawbacks can be solved by applying some algorithms such as Extended Kalman Filter (EKF). Combination of accelerometer and gyroscopes can be utilized to collect acceleration and orientation measurements, so they can be processed to recover pose of the moving objects or persons. Therefore, accelerometer and gyroscope measurements can be fused for pose estimation. Though wearable units are used to collect motion directly, and don't suffer for occlusions and illuminations, wearable units generally require cable connection or battery charge which may not be preferable in some applications where continual measurement is important. Therefore, alternative pose estimation methods are also studied widely by using remote sensors. Some applications can be required remote measurement and monitoring such as biomedical and robot applications, so camera systems, radars, lidar sensors can be preferable for continuous and non-touch measurement. Image features, markers, depth cues can be extracted from camera frames and RF wave data can be collected from radars. Motion and pose information can be recovered by processing image or radar measurements.

Feature based pose estimation algorithms generally extract image features and compute pose by processing feature correspondences on model and target imagery. Point cloud based methods extract depth from model and target object, compute pose iteratively finding minimum error of the point cloud distance such as ICP algorithm. Some algorithms find pose by modeling point clouds as data points and GMM centroids and seek to find pose by fitting data points and GMM centroids which maximizes likelihood. Some researchers study artificial neural networks such as CNN's. CNN based approaches need to be trained with public datasets and CNN networks can produce accurate pose matches depending on learning. However, CNN based methods can suffer for generalization where pose of non-trained samples will be estimated.

There are many different approaches for motion and pose estimation problems. Pose can be found for rigid object that refers to all points of the object moving in same parameters as vehicles, robots. Non-rigid pose refers to pose parameters that are varied on the object points and points on the object aren't moved in same value as human cell motion or limb pose estimation. Pose estimation can be 2D or 3D depending on dimension of output parameters. Pose estimation can be 2D which is found by processing x and y motion of RGB images. Pose estimation can be found 3D which can be computed adding depth information, so 3D pose estimation seeks to predict a three dimensional spatial arrangement of human body parts for a final output.

In this thesis, we have studied human limb motion estimation algorithms and we have applied our method to PD patients to estimate PD limb tremors. RGBD camera system was placed stationary and accelerometer were placed to tremorous limbs. PD data was collected from patients that stretched their arms. We have applied our algorithm to single person

or object motion and pose estimation. We have collected PD data from MSU clinic and PD data is studied. We found motion trajectory of the moving limb that helps doctors to evaluate level of PD. We have also utilized wearable units as well for providing ground truth measurements. In extension of our research, we have studied pose estimation of rigid objects by processing point cloud information from RGBD imagery and our approach is applicable to robot pose estimation problems.

1.1 Problem Formulation

We have studied PD tremor estimation, pose from wearable units, point cloud based pose estimation of rigid objects. We have formulated motion and pose problems for rigid and non-rigid object pose estimation problems. Accelerometer was placed to patients' arm for PD tremor estimation problem. Therefore, motion of the wrist and elbow need to be estimated and arm motion can be found explicitly. Frame set can be defined as S and sample number n is termed as F^k . We need to find set of samples set, S :

$$S = \{F^1, F^2, \dots, F^n\} \quad (1.1)$$

Limb poses are specified by their 3D end-points. We termed the forearm with elbow u_e , and wrist, u_w , arm motion is somewhere between arm and wrist. Therefore, we need to find motion where accelerometer was placed. Motion of arm can be found at RGBD imagery and can be projected to camera coordinate system. Camera projection can be termed as P , elbow location at RGB image termed as x_e , and location of wrist at the RGB image termed as x_w . Arm motion at camera coordinate system can be termed, u_a , can be found by elbow

and wrist projections:

$$u_e = P(x_e) \quad (1.2)$$

$$u_w = P(x_w) \quad (1.3)$$

Then, u_a can be found through interpolation of motions: u_e and u_w .

At second part of our research problem, we have studied Loopy Belief Propagation (LBP) algorithm that can give accurate localization for mobile robot applications. LBP is a method that infers hidden information through message passing rule and inference is computed by product of set of neighbour messages. Inference probability can be formulates as,

Simultaneous Localisation and Mapping (SLAM) is the process of finding map of an environment of a mobile robot, and SLAM problem can be defined as conditional probability distribution where x defines vehicle location, m defines landmark, Z defines landmark observation, U defines control inputs.

$$p(x_k, m | Z_{0:k}, U_{0:k}, x_0) \quad (1.4)$$

At third part of our research problem, we have studied rigid object pose estimation. To estimate pose of the rigid objects, we have utilized from point cloud measurements from model and target object. Pose parameter can be termed as θ , model geometry can be termed as X and target observation can be termed as Y for problem formulation. And, Joint distribution of X and Y , $p(X, Y, \theta)$, can be defined as:

$$P(X, Y, \theta) \propto \phi_{kinematic}(X, \theta) \phi_{geometric}(X, Y) \quad (1.5)$$

Pose parameter, θ , can be estimated by maximization of likelihood function, (L):

$$L = \sum_{i=1}^M \log(\sum_{j=1}^{N+1} P(y_j) P(x_i(\theta|y_j))) \quad (1.6)$$

1.2 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, we presented literature survey of IMU and camera sensor based PD tremor motion estimation, and rigid object pose estimation algorithm and applications. In Chapter 3, we presented an efficient and accurate limb tremor estimation algorithm. Chapter 4, we presented a graphical probabilistic inference method for robot localization problems. Chapter 5, we presented a unstructured 3D shape matching method that estimates relative pose of the rigid objects. In Chapter 6, we presented Experimental Results regarding our research problems. In Chapter 7, we presented conclusions of the studies performed and discusses our future work.

Chapter 2

Related Study

Most current limb tremor monitoring is performed through wearable sensors by applying series of process on raw data and extracting amplitude and frequency characteristics of PD tremors [1, 2, 3, 4]. Wearable sensors can be used to quantify limb tremors with less than 1 cm amplitude error through sensor fusion algorithms [1], and wearable sensors provided 0.12 Hz mean frequency difference for severe PD samples [2]. These typically rely on accelerometers, gyroscopes, and magnetometers for measuring postural limb tremor frequencies in the range of 1 – 10 Hz, and provide tremor data regardless of body pose or location. Wearable sensors enable to direct PD tremor measurement which provides some advantages for measurement and quantification. However, wearable sensors have a number of drawbacks for tracking PD tremor symptoms; they can be uncomfortable due to contact to patients, have limited battery power that needs to be replaced and charged again, multiple sensors attachment to PD patient is required to be placed for each limb being monitored, and the patient may forget to wear those sensors.

A contactless sensor can be addressed problems on wearable sensors, and the Kinect 2 is potentially suitable device as it can automatically detect and track limb motion for up to six individuals in video imagery. However, while the Kinect has demonstrated robust large-motion pose estimations, Obdrzalek et al. [5] reported poor accuracy for finer scale motions of under a centimeter amplitude. This presents a problem for tremor measurement

and quantification, as PD tremors can often be only several millimeters amplitude levels [6, 7]. This low accuracy motion tracking prevents the Kinect being directly used for small amplitude tremor monitoring and analysis as biomedical applications.

We address the low accuracy of the Kinect for small limb motion measurement by proposing an algorithm to super-resolve the PD tremor tracking accuracy of the Kinect. We have proven that the proposed algorithm improves the accuracy of the Kinect’s limb tracking capacity for small-amplitude PD tremors. In so doing we make it possible to use the Kinect as a fully contactless PD tremor monitoring system.

Kinect relies primarily on its depth sensor measurements for human pose estimation [8]. While this enables robust pose estimation, it also limits the precision of its joint position estimates. Here we propose to leverage the higher angular resolution Kinect color imagery to improve the precision of small PD limb tremor estimates. The depth and color modalities in the Kinect have previously been fused, for example, to achieve real-time map building applications [9]. Our fusion goal is different than previous applications; rather than improved 3D estimation, it is to obtain improved small amplitude tremor estimation. To achieve this goal, we leverage the Kinect’s coarse depth-based motion and supplement it with a precise optical flow estimation from the color imagery. We provide quantitative comparisons between our method and the Kinect using ground truth from both a marker and from an accelerometer.

There are two primary contributions of this research.

1. We proposed a tremor measurement method that operates on human limbs in natural poses and is both contactless and automatic, without requiring wearable sensors attachment to tremulous limbs or human supervision to use.

2. While the Kinect’s motion detection accuracy is poor for tremors which are below 10 mm amplitude, the proposed method upgrades the sensitivity to achieve detection and analysis for tremor motion down to 2 mm amplitude for both healthy subjects and PD patients.

While there are other commercial and remote sensor-based trackers that have sufficient accuracy for PD measurement and monitoring, such as Vicon and OptiTrack, these devices actually require markers be worn, and so they are not truly contactless measurement. Moreover, they are expensive and require careful installation to human body and system calibration. In contrast, the Kinect is fully contactless, inexpensive and very simple to install and use. Use of proposed algorithm with the Kinect will enable accurate tremor measurement in homes and hospitals with no worn sensors, markers etc. Resulting measurements can be utilized by clinicians to monitor PD patient’s symptoms, and assist doctors in adjusting dosage levels and frequency of PD medication etc.

Wearable sensors have been investigated as a way to collect limb motions and PD tremors that makes it a potential tracker for applications. Dutta et al. [3] used a Bayesian network to estimate grasp and grip efficiency with sensor data by utilizing smart gloves. Aşuroğlu et al. [4] used shoe-attached sensors to monitor PD patients’ symptoms, and they developed a regression model from labeled PD tremor data and their algorithm estimates the severity of PD based on gait features. Niazmand et al. [10] utilized from wearable devices, and placed two accelerometers into wirelessly connected smart gloves to evaluate frequency and rigidity of tremors based on frequency levels. Szumilas et al. [11] present a vibration sensor that monitors PD tremors by measuring mechanical activities of the limb muscles and their sensor is attachable to many points of human body and they presented a qualitative signal

comparison method to verify functionality of the device by using linear correlation and Pearson's distance. Camara et al. [12] collected data through surface implanted electrodes from patients suffered from PD, and their algorithm classifies resting tremors and identifies PD tremors. They extract features from tremor signal and apply down-sampling and filtering on raw tremor data, and the classification task is done with neural networks. Salarian et al. [13] developed an ambulatory system that is comprised of multiple miniature gyroscopes that quantifies tremors and they statistically analyzed their PD tremor measurements. Ji et al. [14] used multiple wearable and wireless sensors to measure arm motion and trajectories, they modeled arm motion that presents motion from wrist to shoulder joints. Castillo et al. [15] used a wearable sensor that collects sensor measurements from patients during daily activities. Their algorithm first extracts voluntary motion by applying adaptive algorithm. Then, their algorithm subtracts voluntary motion from total motion to find PD tremors. Gallego et al. [16] measured wrist tremors with couple of gyroscopes. Their algorithm extracts tremor pattern from gyroscope measurements. Their algorithm collects voluntary motion including PD tremors from daily activities. Their algorithm applies g-h filter and Kalman Filter for estimating voluntary part and extracts tremorous motion by modeling tremor with weighted fourier linear combiners (WFLC). Then, their algorithm models tremors by using weighted linear combiners. Olivares et al. [17] used Inertial Measurement Units (IMUs) which are connected via wireless and their algorithm fuses accelerometer, gyroscope and magnetometer measurements by utilizing KF and LMS filters, estimates gait and body posture. Bakstein et al. [18] used surface implanted electrodes which are implanted to head and their algorithm extracted temporal and spectral features from brain signals. Then, neural network based algorithm identifies PD tremors. Joshi et al. [19] utilized from PD from gait signals and

applied signal transformation approach and they applied combination of discrete wavelet features and a SVM that can give efficient PD tremor classification. Schäffer et al. [20] applied a Kalman filter to fuse optical flow, IMU, and GPS information within a FPGA in a wearable sensor that give 3D pose estimation. PD tremors measured with wearable sensors [21, 22] often require manual downloading and external processing.

While our research shares the same goals of many of these studies, it differs in that we aim for remote PD tremor measurement without requiring any wearable sensors. Numerous studies have investigated the Kinect’s reliability, accuracy and suitability for healthcare applications, and have developed algorithms and integrated on the Kinect [23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38]. Kinect based tracking is sufficiently precise to enable interactive abdominal surgical guidance with 2D shape matching algorithm [29], balance development on elderly patients [30], and measuring upper limb motion for rehabilitation exercises that require large motion [31]. Nevertheless, the reliability of the Kinect in estimating pose can be poor and insufficient relative to the needs of some biomedical applications [32, 33, 34, 35, 36, 37]. Xu et al. [32] found that Kinect can be utilized for gait analysis depending of the accuracy level of the application, and the magnitude of the pose is not accurate due to substantial errors. Wochatz et al. [33] tested Kinect for lower limb extremities, and concluded that Kinect produces poor results for small amplitude motions and poor-to-fair alignment with the Vicon. The Kinect produces discrepancies on measurements of shoulder motion [34], it has a modest reliability on transitional motion as sitting, standing, and stepping measurements [35], and it produces poor results in measuring gait asymmetry [36]. Joint pose estimation accuracy is quantified in [37] and it is stated that Kinect can be used for medical applications with some limitations. Diaz-Monterosaz et al. [24] compared

Kinect with other motion capture devices and reported that Kinect has poor estimates for joint location and rotation estimates. Grooten et al. [23] investigate the Kinect for medical applications, and find poor performance for human posture estimates. Casacanditella et al. [25] evaluate the Kinect for measuring PD tremor frequency with Laser Doppler Vibrometer and find a mean frequency error of 0.31 Hz which is close to our frequency analysis. Pöhlmann et al. [27] claims that the Kinect has a potential device for healthcare imaging and applications. Pagliari et al. [39] statistically analyzed versions of Kinect’s depth measurements and find that the Kinect II has improved stability and precision over the Kinect I. Torres et al. [28], Sooklal et al. [40] and Galna et al. [41] utilized from the Kinect to measure large limb tremors (greater than 1 cm amplitude) and compared with Vicon. In contrast, our goal is to extend the Kinect’s capability to estimate PD tremor frequencies which can be sub-centimeter amplitudes.

Color camera based tracking systems have been investigated in detail, such as Ishii et al. [42], which can provide promising motion estimates, but their algorithm require manual target identification. Pressigout et al. [43] developed a hybrid tracker that outperformed classical edge based trackers, and their proposed algorithm used optical flow displacements and edge features of the object. Krupicka et al. [44] developed a 3D stereo camera based system that measures tremor frequency and amplitude characteristics, but their proposed system requires that the limb be positioned in a volume sized 40 x 40 x 30 cm³. Wei et al. [45] developed a tracker that tracks human motion automatically with depth cameras, and was implemented on a GPU to make it real-time device. Soran et al. [46] studied motion detection using motion filters and Support Vector Machines (SVM’s). Their algorithm first detects skin pixels from color imagery, extracts features from skin pixels, and detects forearm

motion. Chen et al. [47] collected tremor motion with a non-contact optical device from patients suffered for essential tremors, and they quantified tremor amplitude and frequency characteristics.

Limb tremors can also be measured remotely and accurately with other sensors as radar and lasers. Blumrosen et al. [48] studied tremor signal characterization with low power wideband radar and they analyzed reflected radar signals from human limbs, their system is able to measure small tremors accurately. However, radar systems are expensive devices, and so application to home settings is limited. Laser based systems can be potential devices to measure tremors remotely. Chang et al. [21] studied tremor estimation integrated on laser systems, and they utilized from CMOS sensor based systems for PD tremor measurement. Their system projects laser beams on to a patient's hand, CMOS sensor detects hand tremors, and their computer software measures tremor frequency accurately. Yang et al. [22] developed a tremor detection system that works based on the laser line triangulation method. The laser projector sends the laser beam to patient's tremulous hand, and CMOS image sensor detects reflected lights from the patient's hand. They tested their laser sensor-based system by measuring hand tremors on PD patients. In contrast, our approach leverages the Kinect, an inexpensive, consumer device that can easily be mounted at home or in a clinic.

Rigid object pose estimation algorithms can be divided into three main methods: Template based methods, feature based methods, and machine learning based methods. There are different approaches to solve pose estimation depends on image data and algorithm. Some algorithms utilize depth image and computes pose by searching minimum distance locally through point cloud information such as an ICP algorithms, some algorithms utilize from

RGB frames and solve pose from global optimization methods such as image correspondence matching methods.

One solution for pose estimation problem is to apply template matching methods, the image template is created by rendering 3D shape model from tracked object. Template based pose estimation algorithms such as ICP, CPD, and NDT are widely studied in the literature that can produce hi-accuracy estimations [49, 50] ICP algorithm converges to local minimum iteratively, and proposes high accuracy estimates in some cases. Upon reaching to local minimum distance, ICP computes pose estimates by determining the closest distance and computes spatial transformation for point sets. Chen et al. [51] developed a variant of pose estimation algorithm is point to plane ICP algorithm that searches point correspondences and minimizes the defined error metric. Although, the approach can yield promising estimates, performance of ICP algorithm is relatively lower for planar surfaces, non-Gaussian noisy measurements, occlusions, and erroneous point clouds. There are also variants and different fusion techniques of ICP algorithm, Iversen et al [52] proposed shape descriptors to ICP that decreased computation time of the ICP, Presnov et al. [53] fused ICP pose estimates with Inertial Measurement Unit (IMU) measurements by linearizing problem using Extended Kalman Filter (EKF), Aghili et al. [54] integrated a fusion method by combining IMU measurements to ICP algorithm by utilizing Adaptive Kalman Filter (AKF) for space craft pose problem. Myronenko [50] proposed a probabilistic registration method termed as CPD algorithm, CPD finds registration of point clouds by modeling one point cloud as a Gaussian Mixture Model (GMM) and other point cloud as data point set and finding maximum GMM posteriori probability. finds, Delavari et al. [55] utilized from mesh construction of objects and added new model parameters to CPD algorithm, their algorithm

was applied their modified CPD algorithm to medical liver data that gave enhanced registration accuracy, Liu et al. [56] implemented likelihood field model to CPD algorithm that improves the algorithm to find pose of the far away objects with sparse point clouds. Biber et al. [57] developed Normal distance transform (NDT) algorithm that transforms point clouds to probabilistic distributions and compute pose in probabilistic manner. NDT algorithm models point cloud as a set of 2D normal distribution and second scan to the NDT is defined as maximizing sum that defines score on the density for second scan. Hong et al. [58] upgraded NDT by truncating and fusing Gaussian components of point cloud that enhanced the pose estimation accuracy. Liu et al. [59] studied on NDT, they upgraded by clustering Gaussian distribution transform that adds point clustering and k-means clustering to match points. Despite upgrades on NDT algorithm, there are still problems regarding poor convergence. Opromolla et al. [60] utilized from LIDAR point clouds, they find centroid of LIDAR measurements and pose is computed from defined correlation measure of point clouds, their algorithm requires template models and finds pose for space robot applications. Picos et al. [61] applied to correlation filters that estimates locations and orientation of the target frame by iteratively finding highest correlation between model and target frames. Philips et al. [62] developed an algorithm on LIDAR sensor for excavator 6 DoF pose estimation, their algorithm utilizes from maximum evidence strategy infers pose is likely equals to most consistent LIDAR measurements.

CAD model-based methods grasp 3D environment and uses CAD model for shape matching, it presents a noiseless and ideal representation of object model that can be enhanced to pose estimation accuracy. He et al. [63] developed a template based pose estimation algorithm that extracts key points, their algorithm utilizes from CAD model and finds pose by

error minimization method. Tsai et al. [64] integrated template matching and Perspective-n-Point (PnP) pose estimation, their algorithm extracts and matches image key points, their algorithm can be used in Augmented Reality (AR) applications. Song et al. [65] developed a CAD model based pose estimation algorithm, their pose estimation algorithm filters depth images to remove noisy measurements, and random bin picking infers pose from RGB imagery.

2.1 Summary of the Chapter

In this chapter, we have presented and discussed the literature research for the PD tremor monitoring and rigid object pose estimation problem. Wearable and remote sensor based PD measurement methods are explained including advantages and disadvantages. Also, rigid object pose estimation approaches are presented in detail. In the next chapter, we will explain and discuss our PD tremor measurement algorithm.

Chapter 3

Accurate Tremor Detection and Quantification Algorithm

3.1 Introduction

Limb tremor measurements are one factor utilized to characterize and quantify the severity of neurodegenerative disorders. These tremor measurements can also provide dosage-response feedback to guide medication treatments of PD patients. Here, we propose a system to automatically measure limb tremors in home or clinic settings. The key feature of proposed method is that it is contactless; not requiring a user to wear or hold a sensing device or a marker. Our base sensor is a Kinect 2, which measures color and depth images, and estimates rough joint positions of human for indoor environments. We show that its pose accuracy is poor for small limb tremors which are below 10 mm amplitude, and so we propose an additional level of PD tremor tracking method that recovers limb motion at a higher precision down to 2 mm amplitude. We include empirical experiments and measurements showing improved PD tremor amplitude and frequency estimation using our proposed Pose and Optical Flow Fusion (POFF) algorithm.

3.2 Kinect Based Sensing

The Kinect is used in this work for three primary purposes. First, it automatically detects a patient and determines his or her pose in indoor settings with varying lighting. This is a challenging problem to automate, and there has been significant research into it including [8, 44, 46]. For this application the Kinect provides a comparatively high quality and real-time solution. Second, the Kinect provides person 3D limb pose tracking without requiring marker attachment, or manual initialization. The Kinect technology is used in many computer vision applications such as pose estimation, object recognition, object tracking, fall detection, virtual reality, and indoor 3D mapping problems etc [66, 67, 68, 69, 70]. Person tracks from the Kinect are a key input to our algorithm. And third, the Kinect is an inexpensive consumer device that can be readily interfaced with a PC. This enables us to easily build a portable system for a tracker.

The key limitation that prevents a Kinect being useful for tremor measurement is its poor accuracy in estimating small (sub-centimeter) limb motions. This accuracy limitation is due in part to the depth sensor resolution (512 x 424), and also in part due to the robust, but not so precise, 3D shape fitting algorithm by Shotton et al. [8] used to estimate human pose.

A benefit of using the Kinect's color camera is that it has higher angular resolution, 18.2 pix/degree, compared to the depth camera, 6.6 pix/degree. Also, the color and depth cameras are synchronized, and their intrinsic and relative extrinsic camera parameters are known from calibration so that depth pixels can be projected onto the color image. The result is that processing performed on the color images can be easily used to update motion estimates in the depth image.

3.3 Method

The goal of this research is to upgrade the second-generation Kinect’s rough position estimates to obtain precise limb tremor estimations. Our key insight is that for small limb motions we only need relative pose, and not absolute pose, to characterize PD tremor symptoms for short intervals. This enables us to leverage differential measurement from the color camera in addition to the absolute pose estimates from the Kinect. We focus on PD arm tremor detection and evaluation. This is because arm tremors have particularly high impact on patient’s functionality [71], and also because they exhibit relatively larger amplitude motions being distal joints [72].

This section describes steps of our Pose and Optical Flow Fusion (POFF) algorithm, summarized at Figure 3.1. We first define the notation and then explain each of our algorithm steps.

3.3.1 Notation and Parameters

Each frame, F^i , of Kinect, indexed by a superscript frame number, i includes a depth image, color image and automatically estimated body-pose skeletons for persons in the field of Kinect’s view, as illustrated in Figure 3.2 (a). Motion analysis is performed on a sequence, $S = F^1, F^2, \dots, F^N$ containing sequential Kinect’s frames collected over a short time interval, t , at 30 fps. Since we seek limb tremors in the frequency range of 1 to 10 Hz, we selected a time interval of $t = 4$ seconds for limb tremor analysis.

There are two modes of the POFF algorithm. A low computation mode will only perform optical flow estimation after a stationary or slowly-moving limb is detected, and restrict the optical flow estimation in the cropped rectangular region around the tremulous limb,

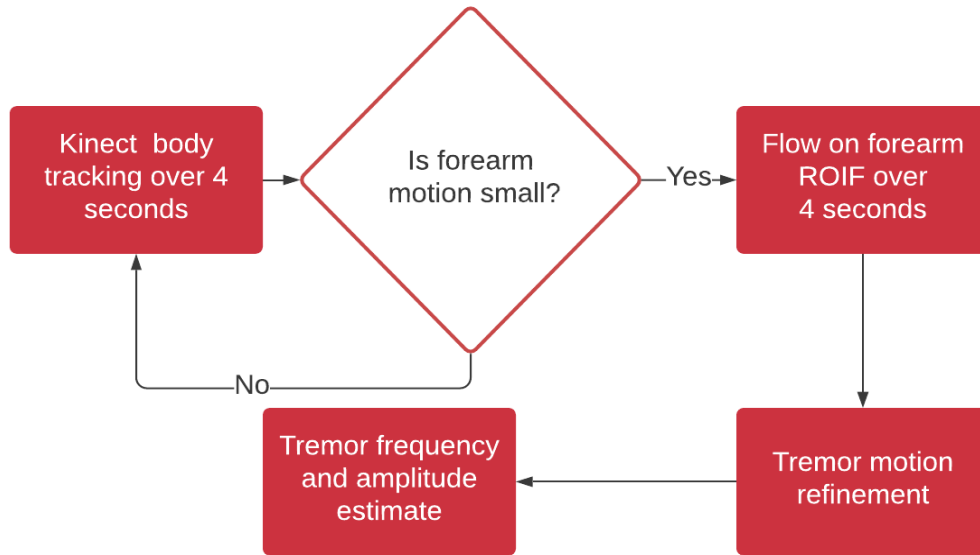


Figure 3.1: POFF seeks for small tremors. Once it detects small limb tremors, POFF starts for further processing for remote PD tremor quantification.

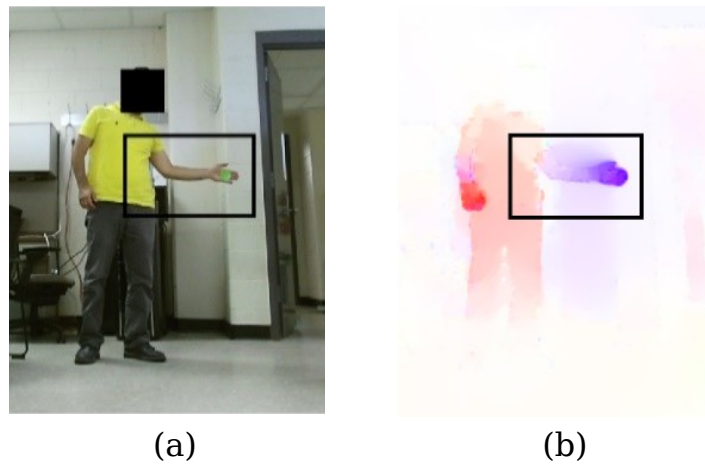


Figure 3.2: Once limb is tremulous, POFF finds flow inside ROI and finds depth correspondences of those flow measurements.

as illustrated Figure 3.2, over the buffered frames along the N . This keeps optical flow estimation computation low at the cost of a slightly delayed tremor detection. A real-time mode performs optical flow estimation continuously over the full image, and buffers the optical flow estimation with along the image sequence, S . In each case, optical flow estimation is computed between each pair of sequential frames, see Figure 3.2 (b), to obtain a dense 2D motion vector, j , for each limb pixel between frames, i and $i + 1$.

$$u_e = P(x_e) \tag{3.1}$$

$$u_w = P(x_w) \tag{3.2}$$

Limb poses are specified by their 3D limb end-points. Here, we consider the forearm with elbow, x_e , and wrist, x_w , locations estimated as 3D vectors by the Kinect. It is straightforward to extend this method to any limb detected by the Kinect. When needed, a superscript index in parentheses will indicate the frame number. Joint coordinates are projected into the color camera image space with a camera projection function, P , giving 2D pixel coordinates, see Eq 3.1 and , Eq 3.2.

3.3.2 Stationary Limb Detection

We focused on estimating limb tremors for stationary (or slowly moving) limbs. By recording forearm endpoints, x_e , and x_w , over time intervals t , it is straight forward to measure the maximum wrist motion. When the total motion is under 10 cm, the tremulous limb is determined to be stationary, and the next step tremor analysis is performed on the color

imagery.

3.3.3 Region of Interest Flow

A region inside the color image is determined that covers the target limb region throughout the interval, t . Optical flow estimation is computed inside the region of interest (ROI) over the interval using the method in [73], which is freely available online, and works well even for low-textured image regions. The ROI is illustrated in Figure 3.2 (b), bounded in the black rectangle, and it provides a vector field of pair-wise motion estimates of tremulous limb and the background.

3.3.4 Tremor Tracking

POFF algorithm tracks limb tremors in the image plane. Using Eq 3.3, the image coordinates of the wrist, $u_w^{(i)}$, can be found for any frame. However, as we show in the results section, these tracks are unreliable for sub-centimeter amplitude tremors. Instead, we use just the image coordinates in the first frame. $u_w^{(1)}$, and rely on optical flow estimation for tracking. This is done by linearly interpolating the flow field to obtain a motion vector $v_w^{(i)}$ of the wrist location and predicting its location in the subsequent frame as:

$$u_w^{(i+1)} = u_w^{(i)} + v_w^{(i)} \quad (3.3)$$

Performing this over all color frames in the sequence generates an image-track of the wrist. This 2D trajectory is converted into a 1D signal by projecting it onto the line orthogonal to the forearm in image-space, namely orthogonal to: $u_w - u_e$. In this way POFF models primarily rotational tremors around the elbow. The magnitude of the trajectory is scaled

by the depth of the wrist position, x_w , and divided by the focal length to generate a tremor measurement in millimeters.

3.3.5 Tremor Motion Modeling

We assume a rigid limb and thus expect horizontal and vertical displacements of the templates will vary linearly across limb as in Eq 3.4. Here the arm positions, y^* are explained with parameters, β_0, β_1 and noise (ε). Performing linear regression gives a fit on multiple data points. We then replace the estimated motion y^* with its linear estimate in our frequency analysis.

Larger sample sizes provide more accurate mean values, identify outliers that could skew the data in a smaller sample and provide a smaller margin of error on motion and frequency estimates. As a result, larger sample motion estimates can be tracked over arm positions because it will provide more accurate mean values and identify outliers that could cause skew on measurements in a smaller sample size and present smaller error. Therefore, those arm points can be fitted for lower error estimates that can improve frequency analysis in case measurements can be noisy.

$$y^* = \beta_0 x + \beta_1 + \varepsilon \tag{3.4}$$

3.3.6 Frequency Estimation

Typical postural tremor frequencies are in the range of 1 to 10 Hz [74]. We selected a four-second interval, as this is sufficient to estimate frequencies in this range. The POFF algorithm first filters the limb tremors in the range 1 to 10 Hz. Band-pass filtering can add

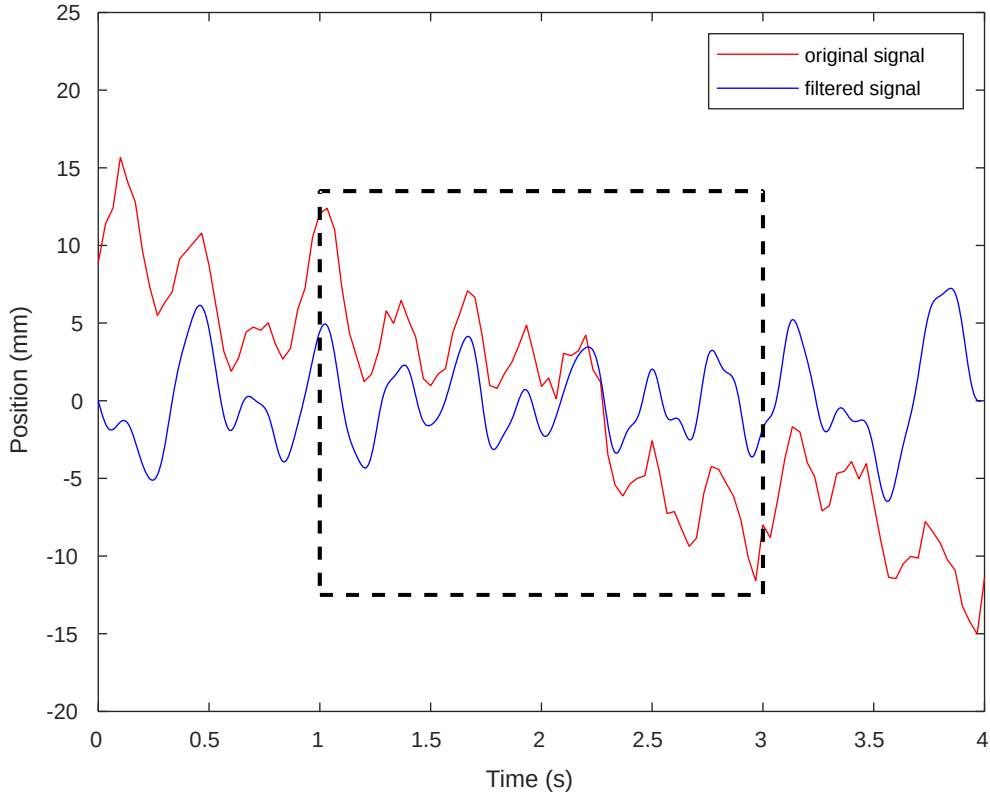


Figure 3.3: Tremulous signal is extracted for period of time and 2 seconds of interval is extracted from tremor signal.

transient effects to the start and end parts that can be amplified, as illustrated in Figure 3.3. Thus, subsequent to filtering, 1-second intervals are removed from the start and end of the 4-second interval, leaving a 2-second, filtered tremor signal.

The dominant frequency is a key measure for characterizing PD symptoms [75], and our analysis focuses on estimating dominant tremor frequencies. Following the work in [76], we use Burg's auto-regressive model to fit the tremor signal and obtain a power spectral density (PSD) estimate of the perpendicular limb motion, see Figure 3.4. We utilized the Matlab 2019 `pburg` function, developed by MathWorks. The PSD describes how power of the tremor is distributed in frequency components and can be computed through periodogram of signal segments of the time. PSD is typically used to characterize signals whose frequency

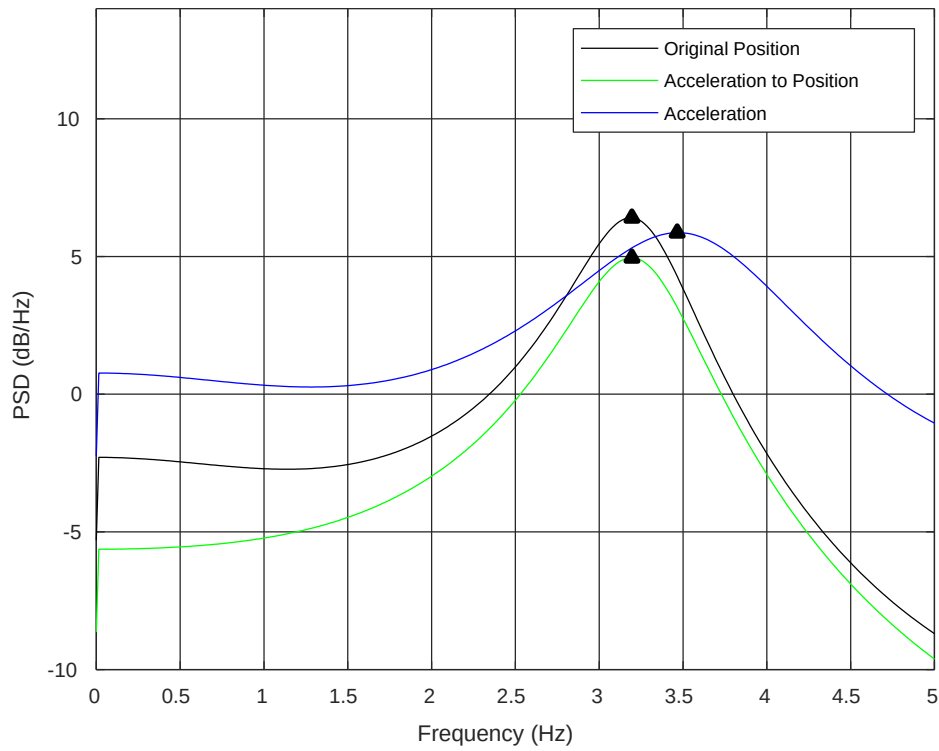


Figure 3.4: Since highest PSD is dominant frequency of tremor, so dominant frequencies are extracted. Acceleration needs to be converted to motion unit.

components can be changed over time. The dominant frequency of the tremor is estimated from the location of the peak value of the PSD.

In addition to peak frequency estimate, we can estimate the amplitude of the tremor parallel to the image plane. This is because the depth measurement on the limbs enable image-motion to be transformed into 3D motion. These motions are illustrated in Figure 3.5.

These five steps constitute our proposed POFF algorithm. Those steps provide fully automated tremor detection and quantification algorithm for small tremors as PD.

3.4 Evaluation Techniques

We propose two simple and inexpensive ways for evaluating the POFF algorithm. While others [77] have used the Vicon for ground truth for tremors, it is both expensive and not easily transported to a clinic setting. Tremors can be tracked by utilizing markers and wearable devices. Our first method uses a green marker that observed by the Kinect’s camera and the second method uses an accelerometer that can measure tremors. A marker-based tracker is utilized as a baseline method for comparison. An accelerometer, able to measure sub-millimeter PD tremors without requiring a camera, provided the ground truth. These are described at next.

3.4.1 Marker Tracking

A direct way to measure limb motion is through a marker attached to patient’s wrist. Since the Kinect includes a color camera, we use a green marker affixed directly to the subject’s wrist. The following semi-automated procedure was used to detect the marker in color

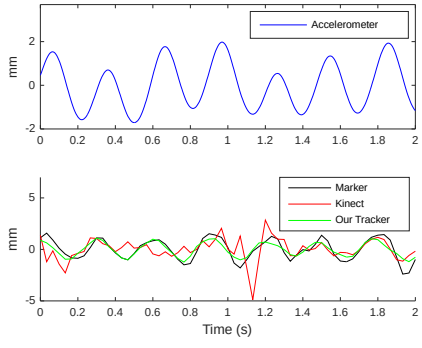
images. Marker position is tracked to measure wrist tremors, and trajectory of the marker gives a baseline method for comparing our results.

In order to track a marker motion, it is required to extract the marker pixels as a foreground and other pixels as a background. We utilized from Linear Discriminant Analysis (LDA) algorithm for classification task [78]. LDA classifies well when clusters' class conditional densities are close to Gaussian distribution which is the expected case for classification of the marker and skin pixels inside the ROI. LDA finds a discriminating hyperplane between target pixels and the background. A 4-dimensional data space is used that consisted of the image region transformed to hue-saturation-intensity (HSV) color space plus a radial distance from the approximate target center position. This approximate center is obtained manually in the first image frame and as the estimated centroid of the previous frame in subsequent frames assuming small motion. Once LDA is trained on the sample frame, it can be utilized to reliably detect target pixels on the marker on the sequence.

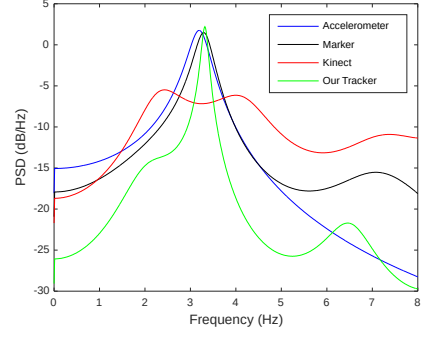
Once target image pixels are detected in a given frame, its image-centroid is computed. Then, its 3D position is obtained by projecting the centroid ray using the mean depth on the marker pixels. The result is that we obtain accurate 3D limb position tracks to compare with the POFF algorithm.

3.4.2 Accelerometer Motion

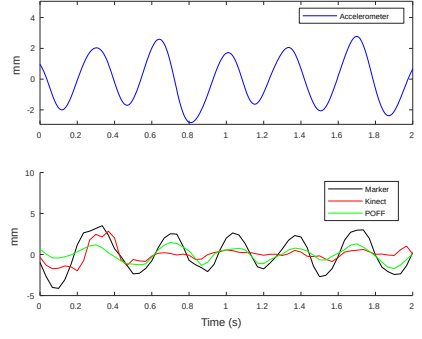
Our ground-truth source is a light-weight, inexpensive, 3-axis accelerometer called a Slam Stick X. It is general purpose for testing motion and vibration data for robotics, automobile industry etc. It is sensitive to small motions, vibration, gravity, and shock which is fabricated with high performance piezoelectric material. This was affixed behind the forearm, and



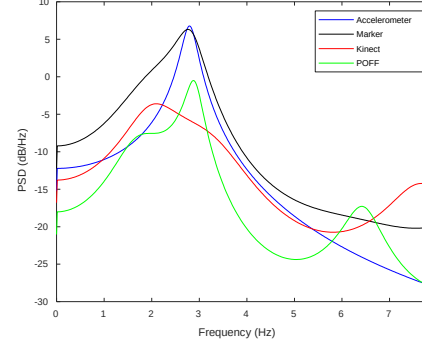
(a)



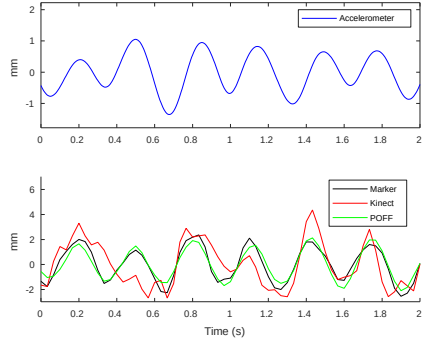
(b)



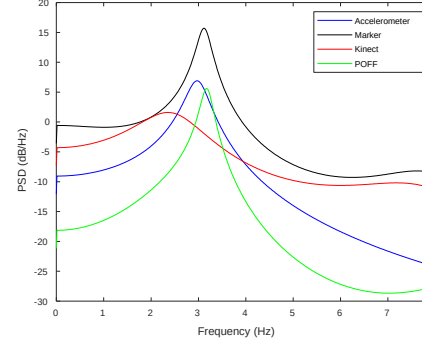
(c)



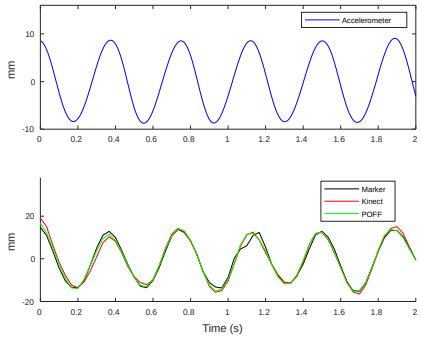
(d)



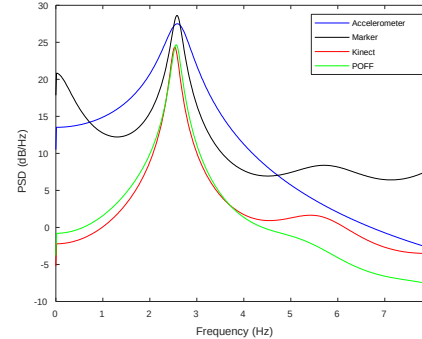
(e)



(f)



(g)



(h)

Figure 3.5: (a) PD tremor is measured with 4 methods. Person stands 2.2 m far away from Kinect; (b) PSD estimations are plotted with 4 techniques; (c) Person is standing 3.2 m far away from Kinect; (d) Kinect's PSD result diverges more than 1 Hz; (e) Person standing 4 m far away from Kinect; (f) Accelerometer, marker, and our tracker show good agreement; (g) Person is standing 3.1 m far away and amplitude is more than 1 cm; (h) PSD results show good agreement for all.

sampling frequency is 200 Hz.

Accelerometers can measure linear accelerations on 3-axis and can be utilized for a 3D motion and pose estimation device, see Eq 3.5 and Eq 3.6. Then, angular rotation can be found by using gravity measurement at 3-axis. However, Accelerometer measurements can be suffered for high frequency noise and drift. Therefore, motion and rotation estimates from accelerometer can be erroneous.

$$\begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = R_x(\phi)R_y(\theta)R_z(\psi) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.5)$$

$$\begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix} \quad (3.6)$$

In order to eliminate noise and drift problems from accelerometer, we applied series of process on accelerometer measurements. Recovering exact motion from this is not feasible due to noisy measurements. However, our goal for this is not to recover full limb motion with the accelerometer, but rather capturing close motion estimate and pattern for frequency analysis. In order to find motion estimate, we have applied series of process to accelerometer measurement. Since accelerometer placed parallel to arm and PD patients stretched their arm during data collection, accelerometer orientation is approximated as perpendicular to x-z axis $[0, 1, 0]$. Dominant tremor frequency measurements for position and acceleration are different because acceleration is numerically equal to twice differentiation of motion that can shift frequency estimations, see Figure 3.4. In order to compare frequency estimates,

accelerometer measurements are transformed to position. With this goal in mind, we developed a sequence of filtering, mean subtraction, and numerical integration steps to achieve quasi - periodic motion analysis.

There are three obstacles in the way of utilizing an accelerometer signal for PD tremor frequency analysis. The first is that the power distribution of a signal's acceleration differs from that of the raw signal. Secondly, integrating acceleration to obtain translation adds significant drift over time. Thirdly, there is in general a temporal offset between the Kinect and accelerometer which needs to be calibrated. The latter issue is addressed, and our solution to the first two concerns are described in this section.

The accelerometer's PSD is not directly useful since its peak frequency may differ from the wrist motion's peak frequency. This is due to the power of a signal's temporal derivative being a function of the frequency. A simple sinusoidal signal illustrates this: $\varphi = \sin \omega t$ with frequency ω , will have a time derivative, $\dot{\varphi} = \omega \cos \omega t$, and so the power of the derivative depends on ω . More general signals that are superpositions of multiple frequencies will have complex dependencies on the frequencies. The consequence for this work is that we need to integrate the accelerations to translations in order to obtain a ground-truth frequency.

Now integrating acceleration measurements to estimate translation leads to erroneous drift that can be accumulated over time. Fortunately, this drift is an offset that is not oscillatory, and so can be filtered without impacting our peak frequency values. Therefore, the following five-step procedure is used to transform a periodic acceleration signal to a periodic translation signal:

1. The acceleration is integrated,
2. The mean velocity is subtracted,

3. The velocity is integrated,
4. The mean position is subtracted
5. A band-pass filter is applied

This procedure removes voluntary motion (including amplified noise) as well as frequencies below that of interest for PD. Finally, to remove transient effects of the band-pass filtering, the first and last second of the signal are discarded. What remains is a translation signal containing quasi-periodic frequencies representative of the PD tremors.

3.4.3 Motion from IMUs

Motion and pose information can also be computed by utilizing IMU measurements and fusing IMU measurements with EKF. Also, IMU and camera sensor measurements can be fused to overcome limitations of IMU only pose and camera only pose estimations. KF can only predict linear functions. When estimated motion is a non-linear function, KF will produce bad predictions on estimated signal. To predict non-linear functions, EKFs can be utilized. The EKF handles non-linearity by applying as a Gaussian approximation to the joint distribution of states, x , and measurements, z , utilizing from Taylor series expansions and predicts next estimates. Therefore, PD tremors can be estimated by EKFs. IMUs can give acceleration and orientation information that can be utilized for PD tremor quantification.

The EKF can be divided into two main steps as Prediction, see Eq 3.7 and Eq 3.8 and Correction step, see Eq 3.9, Eq 3.10, Eq 3.11, Eq 3.12, and Eq 3.13.

Prediction Step:

$$\hat{x}_t = f(x_{t-1}, u_t) \quad (3.7)$$

$$\hat{P}_t = F(x_{t-1}, u_t)P_{t-1}F^T(x_{t-1}, u_t) + Q_t \quad (3.8)$$

Correction Step:

$$v_t = z_t - h(x_t) \quad (3.9)$$

$$S_t = H(x_t)\hat{P}_tH^T(x_t) + R_t \quad (3.10)$$

$$K_t = \hat{P}_tH^T(x_t)S_t^{-1} \quad (3.11)$$

$$x_t = \hat{x}_t + K_tv_t \quad (3.12)$$

$$P_t = (I_4 - K_tH(x_t))\hat{P}_t \quad (3.13)$$

where predicted state is termed as x^t , the residual of the prediction, v_t , f is the nonlinear dynamic model function, h is the nonlinear measurement model function. S_t is prediction covariance, K_t is Kalman Gain, Q is noise covariance matrix. The matrices F is state transition matrix and H is observation matrix which are the Jacobians of f and h , respectively.

3.5 Results

3.5.1 Healthy Subjects

We use a collection of tremor-motion examples to illustrate the performance of POFF, the raw Kinect tracker and our baseline marker position estimate, see Figure 3.5 (a-h). In addition, the twice-integrated, twice mean subtracted, and band-pass filtered accelerometer are shown. For each of these the Burg PSD is shown, from which we can estimate the dominant tremor frequency. In the small tremor examples, plots (a), (c) and (e), the peak frequency of the Kinect has greater error than POFF, see (b), (d), and (f), using the accelerometer, as ground truth. Results are similar regardless of range to target: 2.2 m for (a), 3.1 m for (c), and 4.1 m for (e). On the other hand, for large tremors of greater than 1 cm, shown in (g) and (h), all methods had close frequency estimates.

Next, a series of ablation studies were performed comparing the Kinect’s estimates with our method and the marker method. These studies vary the amplitude of tremors, distance from sensor, the patient pose, and clothing. In each case, the filtered accelerometer frequency is used as ground truth for frequency estimations. Results are plotted in Figure 3.6, and explained in detail. These studies are discussed individually below.

Results in Figure 3.6 (a) show a significant impact of tremor amplitude on accuracy. The distance to the person is fixed at 3.1 m, and limb tremors have amplitudes of approximately 4, 6, and 15 mm. The Kinect performs well for large tremor amplitudes (15 mm), achieving 0.031 Hz mean absolute error. However, its performance degrades by an order of magnitude for small amplitude tremors (under 6 mm). This degradation is the motivation for this work. On the other hand, the POFF and marker methods show similar accuracy as the Kinect for

large tremors, and maintain this accuracy for small tremors.

Figure 3.6 (b) compare accuracy as a function of distance from sensor, in the range of 2 to 4 m. We find little dependence on distance, with POFF and the marker method maintaining much smaller mean absolute frequency errors than the Kinect.

Results for small tremors with varying pose and clothing are shown in Figure 3.6 (c) to (f). These have larger error than the out-stretched arm for all methods, although the pattern remains in which our method significantly out-performs the Kinect. The POFF algorithm, Kinect, and marker method all provide good motion estimation for limb tremors which are above a centimeter range, and all methods share close peak frequency estimations, see Figure 3.6 (g-h).

We also analyzed limb tremor estimates for limb-touching surfaces, which are more challenging poses than a out-stretched arm pose, and clothing variation, see Figure 3.6 (c-h). Both the Kinect and POFF's performance degrade for nearby-limb touching surfaces, and wearing plain jacket because person's upper body is occluded by the forearm and the jacket, see Figure 3.6 (e-h). Nevertheless, the POFF algorithm significantly out-performs the Kinect on small-tremor experiments, see Figure 3.6 (c-h).

Our combined result for all experiments with small tremors (under 10 mm) is 0.159 Hz error where Kinect produces 0.661 Hz frequency difference. The marker-less POFF accuracy is close to the baseline marker method, while the Kinect has close to six times the mean absolute error of the POFF.

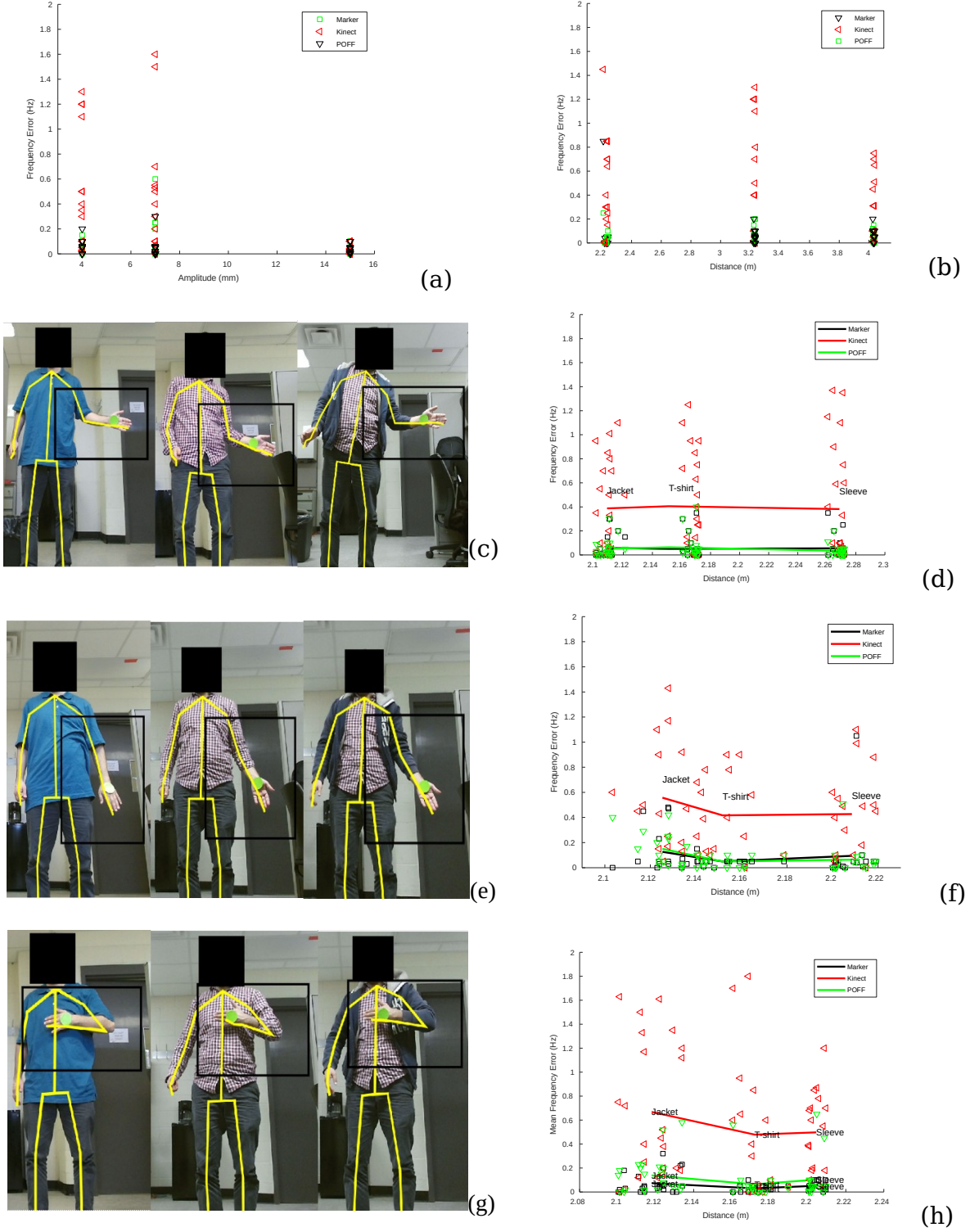


Figure 3.6: (a)Our tracker and Kinect's error is plotted; (b)Our tracker and Kinect's error is plotted for different amplitudes; (c)Person stands about 3.2 m far away and his limb is close to his body; (d)Mean frequency error (MFE) is scattered (c); person's limb is close to his body (e)person stands about 3.2 m far away; (f)MFE are scattered for 3 different methods.

3.5.2 PD Patients

We collected tremor data from nine patients previously diagnosed as PD, during their regular visit the clinic for treatment. They include six males and three females and are all seniors. Tremors varied from mild to severe (up to 5 cm). Patients stood with one arm extended laterally about 3 m from the Kinect. Postural tremor data is collected from the PD patients for 10 minutes. Data were recorded from the Kinect and from an accelerometer placed at patients' wrist for ground truth. The POFF algorithm is able to measure wrist tremors up to a 2 mm tremors, see Figure 3.7, and peak frequency is estimated coherent with the ground truth, see Figure 3.8. Both the POFF algorithm and the Kinect fail to measure tremors which are below a millimeter amplitude, see Figure 3.9, and peak frequency is estimated which deflects from the ground truth, see Figure 3.10. A summary of tremor frequency results is in Figure 3.11, and frequency error as a function of amplitude is shown in Figure 3.12. At 1 cm amplitude the Kinect performs well, but it degrades at lower amplitudes while the POFF algorithm performs well. Below about 1 mm amplitude the POFF error grows significantly. An error scatter plot is shown in Figure 3.12, and frequency estimates have larger errors below 2 mm amplitude. This is consistent with experiments on healthy subjects which showed effective tremor estimation down to 2 mm. No markers were attached to patients; only an accelerometer to measure ground truth. Data statistics are plotted in Figure 3.13, showing significant reliability improvement of POFF over the Kinect.

Hypothesis can be tested with statistical tests. Using the Student t-test on the PD results, we tested and rejected the hypothesis that the mean absolute errors of the Kinect and POFF are equal with p-value 0.0014. Next, we hypothesized that the mean absolute frequency error of the Kinect is at least 0.25 Hz higher error than POFF, and found a p-value

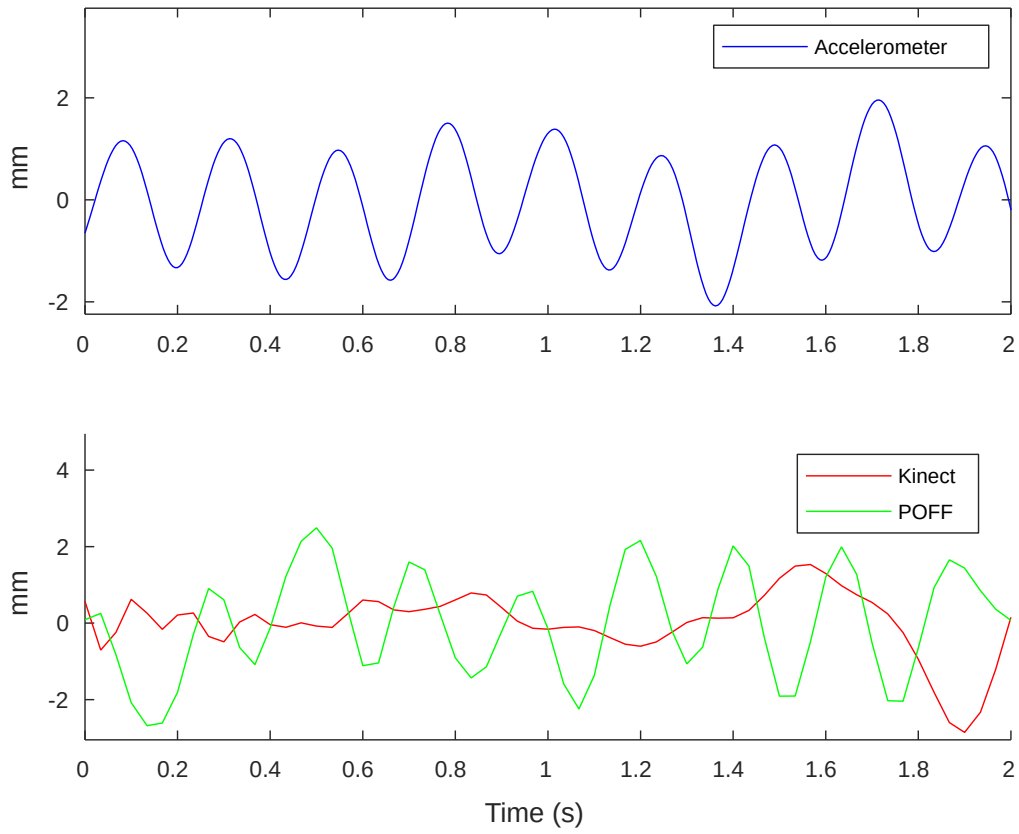


Figure 3.7: Limb tremors roughly 3 mm amplitude are recovered well with POFF algorithm. In this case, Kinect's motion measurements are poor.

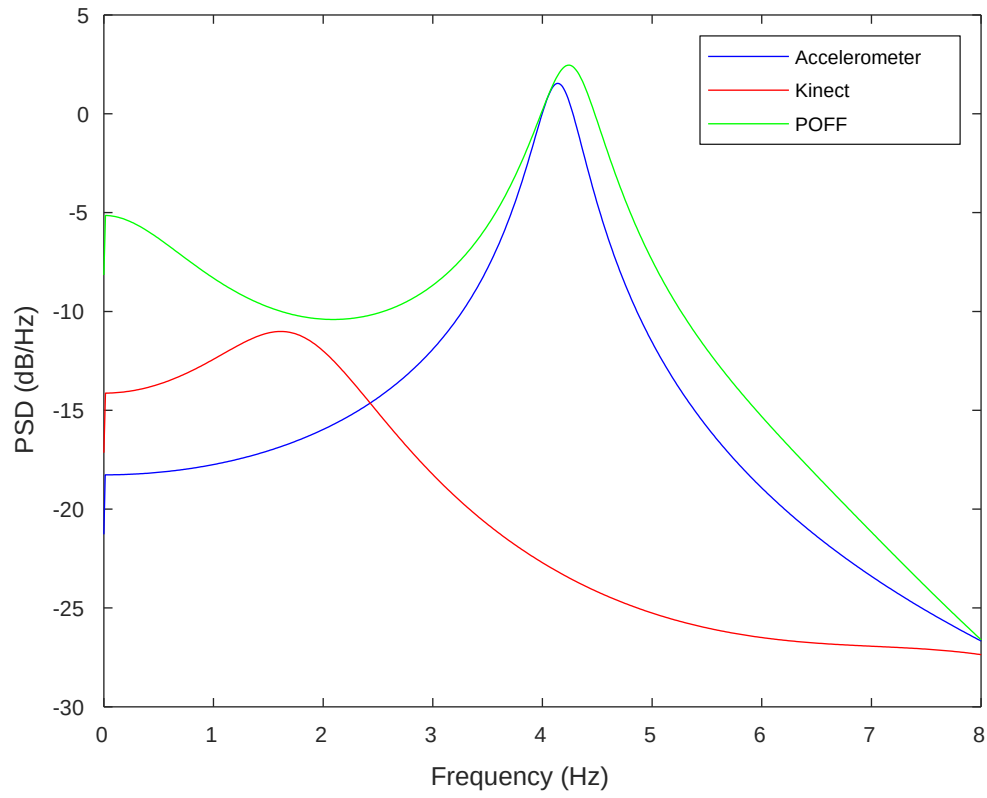


Figure 3.8: PSD computed for 3 different methods for PD tremor at Fig. 3.7. POFF and the accelerometer share close peak frequency, but the Kinect's diverges from the accelerometer.

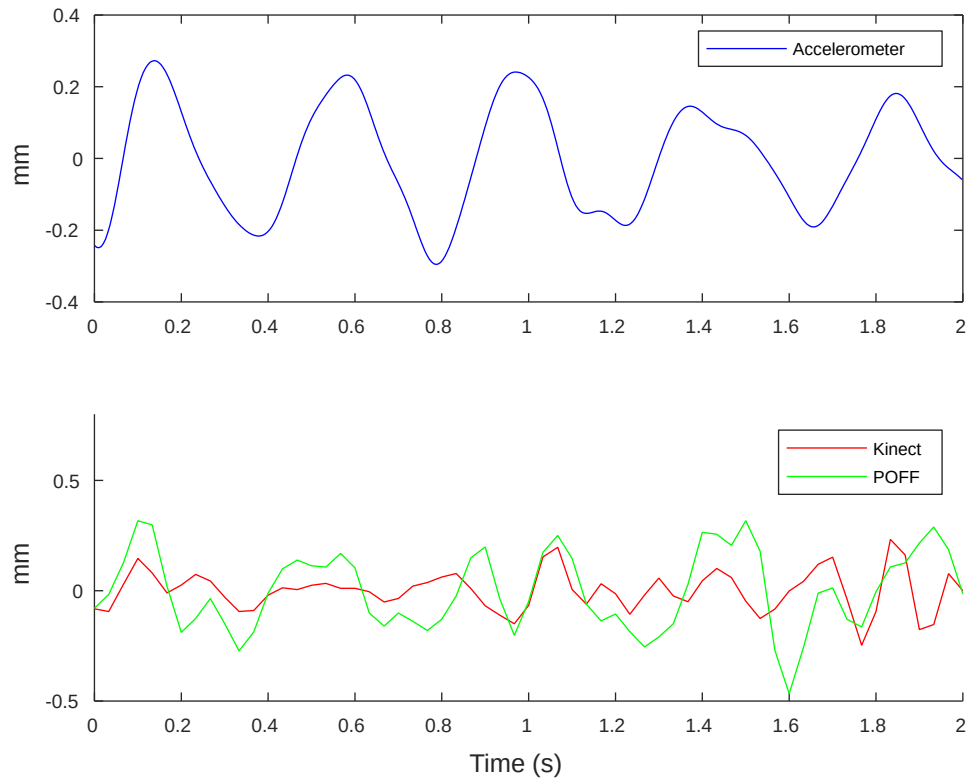


Figure 3.9: For submillimeter tremors, neither the POFF nor the Kinect can recover the tremors well, see Fig. 3.10 PSD estimation is plotted for motion measurements at Fig. 3.9.

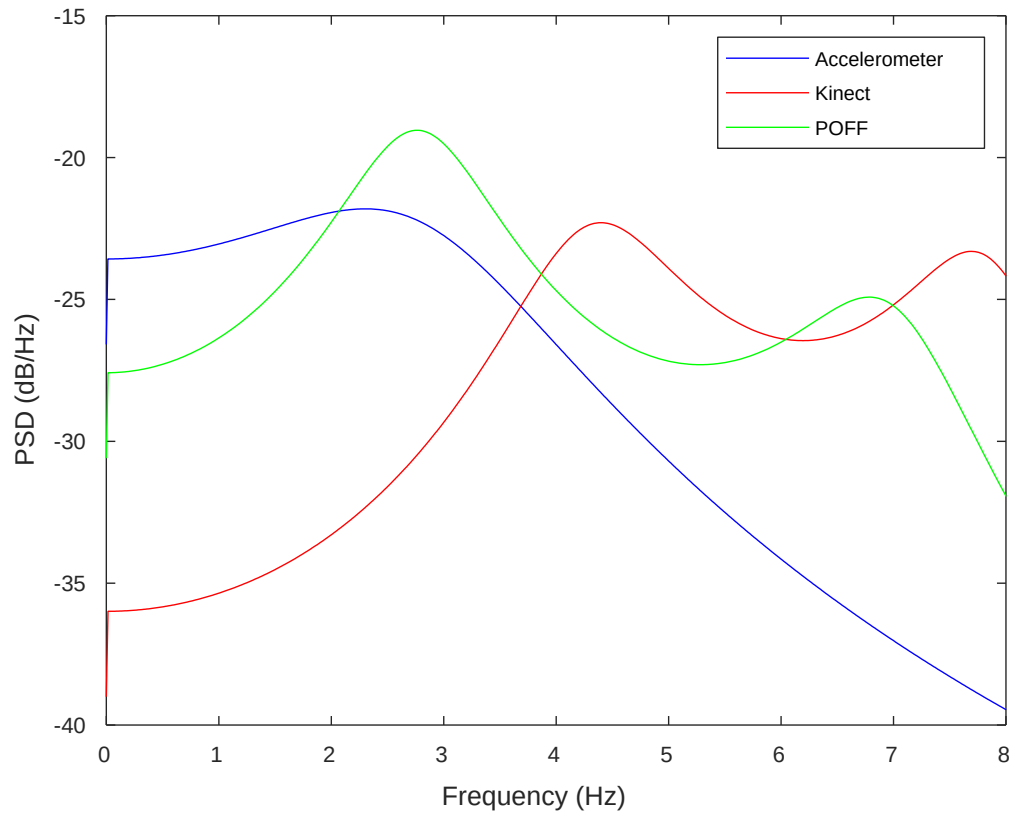


Figure 3.10: PSD estimation is plotted for motion measurements at Fig. 3.9. In this case, amplitude of motion is sub-mm level, both POFF algorithm and Kinect fail to capture tremor frequency.

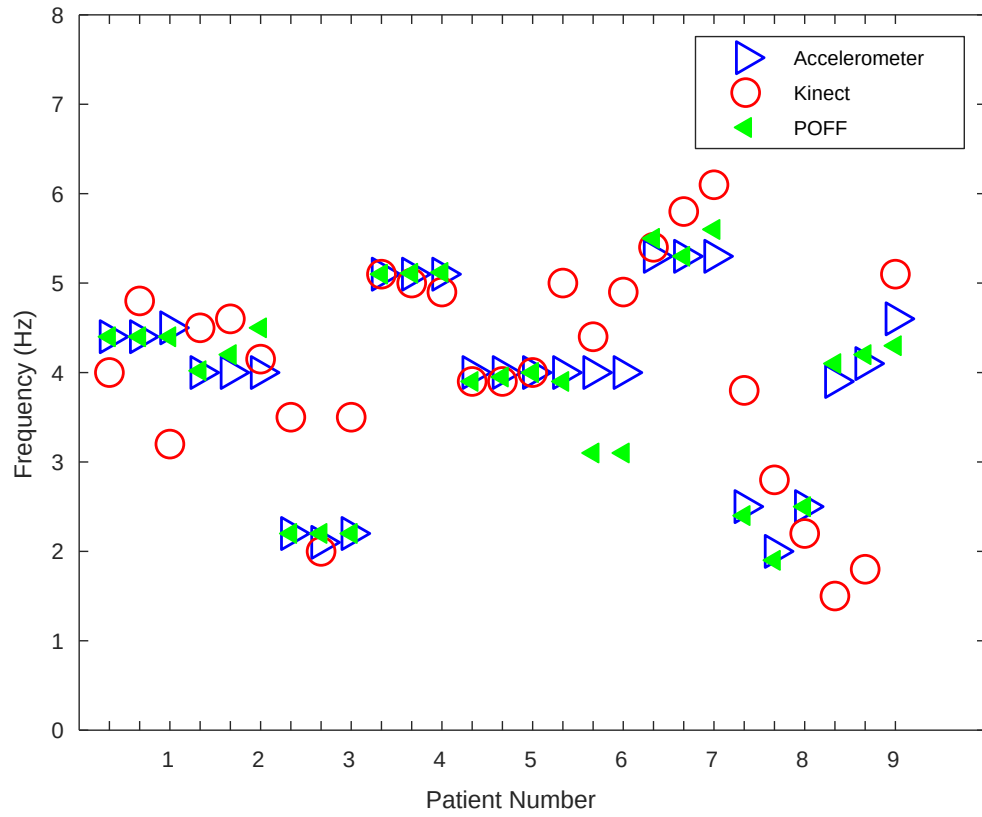


Figure 3.11: Frequencies regarding PD patients are plotted. Three set of sample is measured for each PD patients. Patient's wrist motion is measured with three different methods.

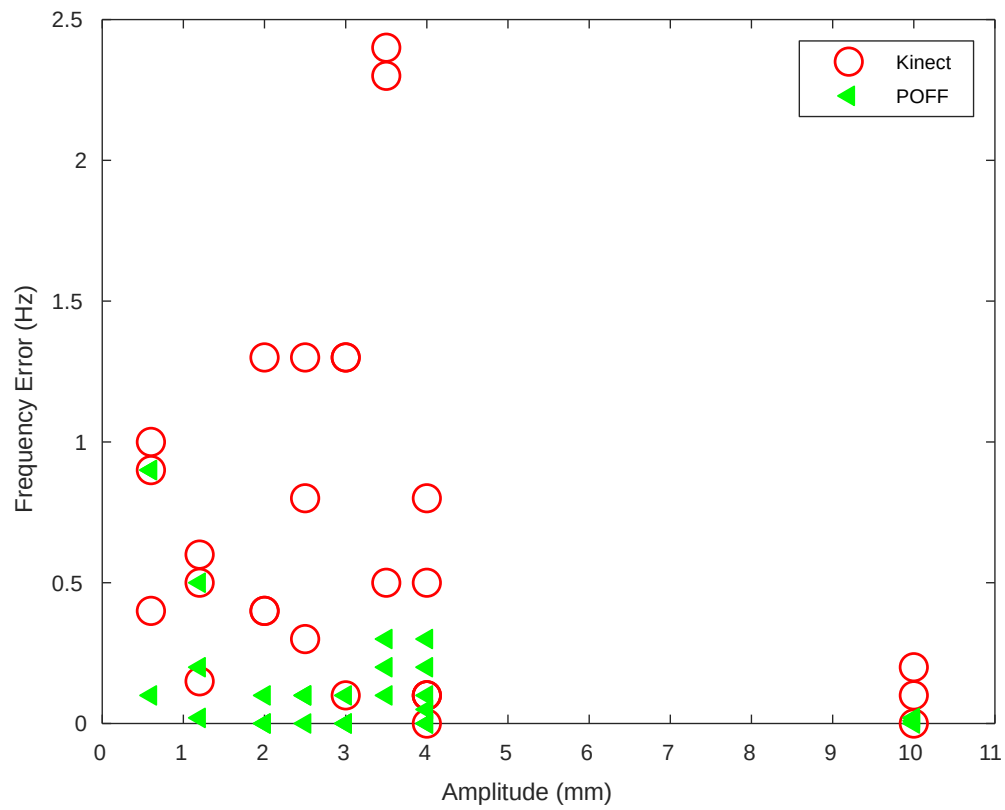


Figure 3.12: Frequency error comparison between the Kinect and POFF for PD patients.

of 0.96. Finally, we confirmed that POFF has a mean absolute error less than 0.25 Hz with p-value 0.936. This gives us confidence in using POFF for small-tremor estimation.

3.6 Discussion

We have presented our findings from results and statistical analysis of small PD tremor detection and quantification problem. The following are six findings from our experiments:

1. The POFF algorithm obtained accurate dominant frequency estimates (error $\leq$ 0.25 Hz) for small, outstretched-arm, PD-patient tremors under 10 mm amplitude levels.
2. In contrast, the Kinect's pose tracking resulted in at least 0.25 Hz larger frequency estimate errors than POFF for the same PD tremors.
3. The accuracies of both POFF and the Kinect are insensitive to distance from 2 to 4 m.
4. Accuracy is highest for all methods for outstretched forearm pose, and has some degradation for other poses.
5. For small PD tremors, POFF significantly out-performs the Kinect and achieves accuracy very close to a baseline method which relies on a marker.
6. The PD patients' tremors have a correlation coefficient with the ground truth of 0.90 for the Kinect and 0.97 for POFF algorithm.

We have interpreted our findings on PD tremor detection and quantification:

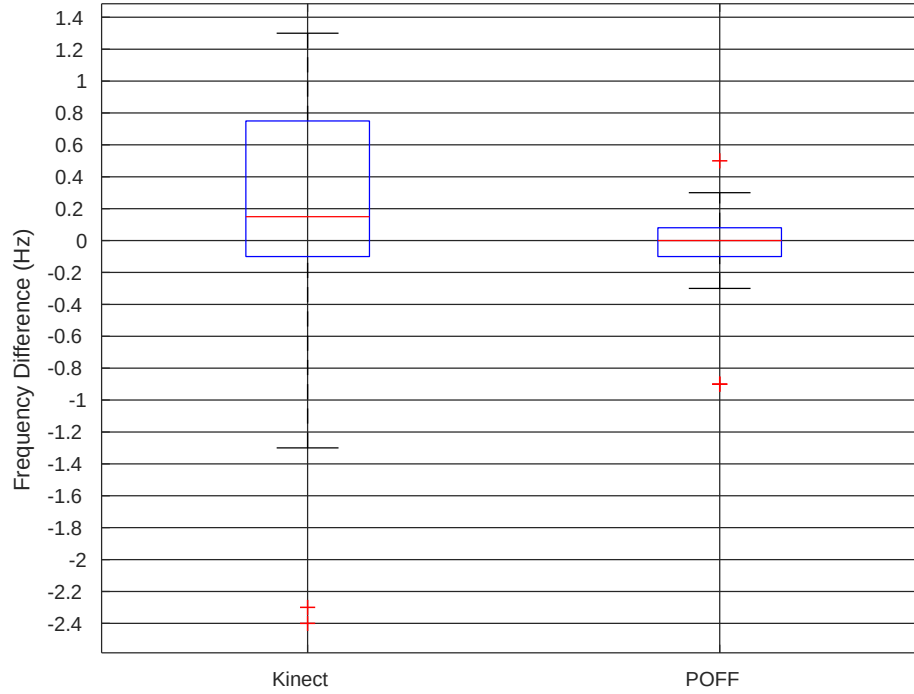


Figure 3.13: Distribution of frequency estimates relative to accelerometer ground truth. The boxes span the distribution between the 25 th and 75 th percentiles, with maximum and minimum lines shown (black) that exclude outliers (red crosses). The median values, (red lines), are closer to zero for POFF indicating lower bias than the Kinect. Also, the much tighter spread of POFF between the 25 th and 75 th percentiles reflects its improved reliability for small tremors.

1. Findings (1) and (2) are the primary reasons showing the need for, and utility of, the POFF algorithm.
2. Findings (3) and (4) agree with previous work on Kinect-only tracking. Gonzales-Jorge et al. [79] found constant pose accuracy as a function of range, and [80] found higher accuracy for standing-stretched arm poses than sitting stretched arm poses.
3. Finding (5) give us confidence that our contactless PD tremor measurement method is comparable to a wearable, marker-based method.

3.7 Conclusion

While the Kinect 2 has robust person detection, and human body joint tracking capabilities, the results presented show that the pose estimates are too coarse for mm level amplitude limb tremor measurements. POFF algorithm uses the Kinect’s initial pose estimates, and upgrades them to obtain mm-level precision needed for tracking small PD limb tremors. This is achieved by combining optical flow in the color images with initial pose estimates from the depth images.

As part of this work, we developed a baseline method by utilizing from a marker, and inexpensive ground truth frequency values by integrating accelerometer measurements. These will be useful for further refinements to PD tremor estimates. Marker motion and the POFF algorithm are compared to obtain additional mean frequency errors.

Our goal is to measure PD patients’ tremors accurately to evaluate any limb tremors because tremor is a health disorder that can be an indicator of serious diseases and degrade patients’ quality of life. We collected data from health subjects at laboratory and PD patients from the clinic. Then, we tested the POFF algorithm on both datasets. For both test results, the POFF algorithm is able to capture limb tremors down to 2 mm amplitude.

Obtaining accurate tremor measurements with a passive, remote monitoring device, as is obtained with the Kinect using the POFF algorithm, will enable improved, continual health monitoring and treatments. This has particular potential for tracking progress of neurodegenerative disease symptoms to provide feedback for treatments. Limb tremor amplitude and frequency characteristics are related to their severity [81], and continually monitoring them can aid in tailoring treatments to the PD patients [82].

The POFF method has the following limitations on small tremor measurement. So far

we have demonstrated accurate frequency estimation for small tremors of at least 2 mm amplitude. POFF can not measure finger tremors, and operates only in well-lit indoor environments without exposed to direct sunlight due to limitations of the Kinect. Also, the POFF's operation distance is limited to between 2 and 4 m. In future work, we intend to extend our POFF algorithm to measure other limbs and fusing POFF with wearable devices. Finger tremors would require a higher resolution camera systems or wearable devices.

As a contactless tremor measurement system, the proposed POFF algorithm with the Kinect and a computer can be utilized both in home settings to passively assess PD patients' limb tremor symptoms, and in clinical settings where PD patients could be assessed prior to a doctor visits and ambulatory device for medication. The proposed algorithm avoids the problems that come with wearable devices including discomfort, non-continual assessment, charging, loss, and forgetting to wear them.

3.8 Alignment Method

3.8.1 Heterogeneous Sensor Alignment

The Kinect and the accelerometer operate asynchronously, with different sampling frequencies, and produce different measurement units. Therefore, we need to align these sensors temporally in order to utilize the accelerometer as a ground truth device for our experiments, see Figure 3.14. This section describes proposed method to align accelerometer and the Kinect's tremor measurements temporally.

The initial steps are a pre-processing of PD tremors into the same motion units. The Kinect's position is transformed into approximate acceleration measurements by differenti-

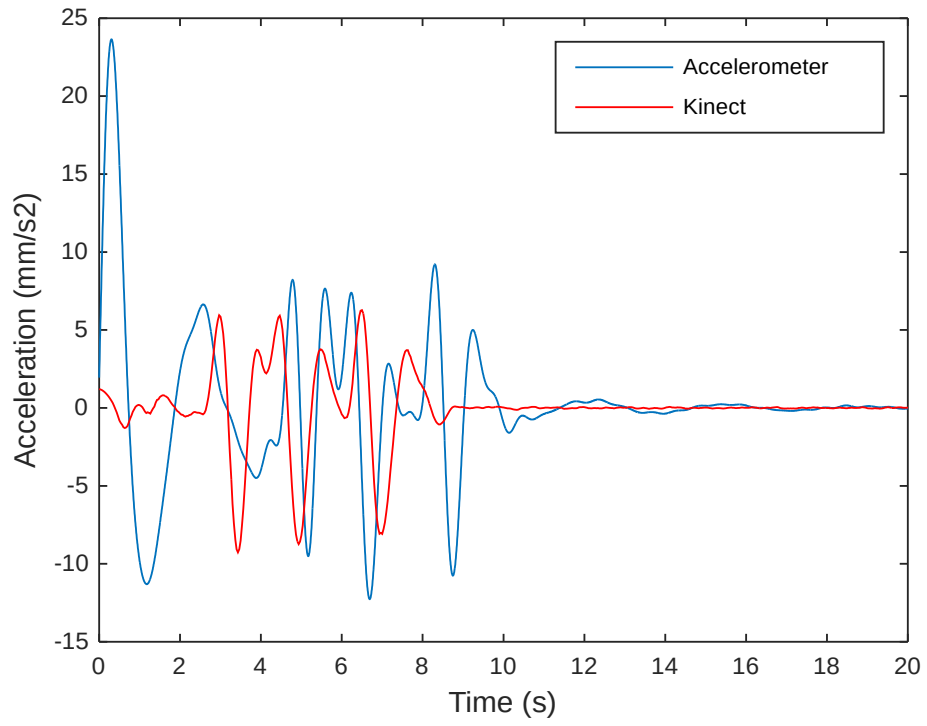


Figure 3.14: (a) Kinect's and accelerometer's measurements are not aligned temporally.

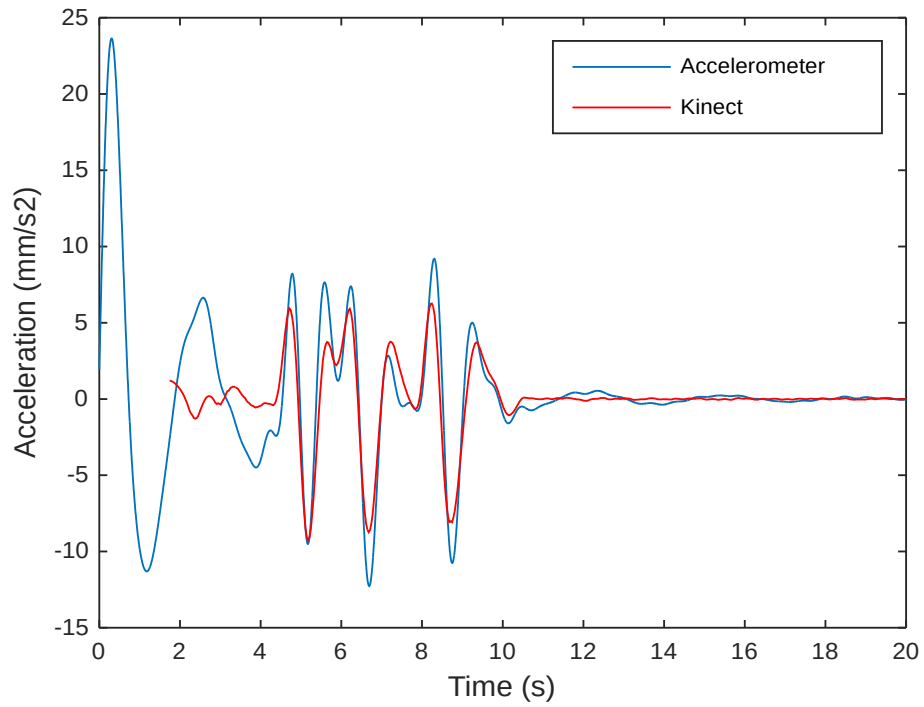


Figure 3.15: Sensors are aligned by computing NCC for the Kinect and accelerometer measurements, and temporal shifting accelerometer.

ating position estimates for twice, and then it is interpolated to the accelerometer’s sampling frequency of 200 Hz. This process can be produced some noise on acceleration, but noise can be filtered. As a result, we gathered acceleration measurements for the Kinect and the accelerometer with same sampling frequency. The differentiation removes zero frequency element which is a DC component of the Kinect’s measurements. The twice-differentiated Kinect’s measurements are smoothed and band-pass filtered to remove the gravity, drift effect, and frequency components above 10 Hz.

In order to measure temporal difference between the Kinect and the accelerometer measurements, PD patients created an artificial oscillation by voluntarily shaking their tremulous wrists for several times which created a relatively higher amplitude oscillations than limb tremors. The voluntary oscillations created a unique trace in tremor measurements which were utilized for calibration using Normalized Cross Correlation (NCC), see Eq 3.14. CC and NCC give high similarity measure when two signal are similar each other and this is expected when samples are aligned on time.

NCC is a reliable way of matching two patterns and frequently used in template based image matching algorithms for finding position of a pattern and time series data etc [83]. CC can compare two time series sample set. Similarly, NCC compare two time series sample set using a different scoring result and NCC can vary between -1 to 1 where 1 equals to maximum similarity of template and target image location. NCC multiplies mean subtracted sample sets and normalizes sample sets with square root of multiplication of variances.

We utilized NCC to align 1D position estimations temporally for the Kinect and accelerometer. We tested NCC to find temporal difference, and it produced higher precision for alignment of sensors with than Cross-Correlation (CC) method. NCC is computed for the

Kinect's and accelerometer's measurements that gives highest peak at the location of sample difference, so sample difference is computed. Then, temporal difference found by dividing sample difference to the sampling frequency of the accelerometer (200 Hz). Then, Kinect and Vicon can be aligned in time. After sensor measurements are temporally aligned, tremor amplitude and frequency measurements of symptoms are comparable for the Kinect, proposed POFF algorithm, and the marker. Temporal alignment is computed semi-automatically since the accelerometer is connected to the computer manually, and temporal difference computed automatically.

Let denote $a(x)$ is acceleration obtained through twice differentiating the Kinect's rough position measurements, then $\sum_x a_u(x) - \bar{a}_u$ is the mean-normalized motion measurements at location, u , where $\bar{a}_u$ is the mean acceleration value. Similarly, let $\tilde{A}(x)$ is the template to match Kinect's measurements, and $\tilde{A}(x)$ indicates the accelerometer measurements, normalized by the mean measurement value over the whole accelerometer measurement interval. This gives the following formula for NCC at the sample location, u .

$$\phi(a, A) = \frac{[\sum_x a_u(x) - \bar{a}_u] \cdot [\tilde{A}(x - u)]}{((\sum_x a_u(x) - \bar{a}_u)^2 \cdot (\sum_x \tilde{A}(x - u))^2)^{.5}} \quad (3.14)$$

The estimated sample difference, u^* , is obtained by maximizing the normalized cross correlation, namely: $u^* = \operatorname{argmax}_u \phi_u(a, A)$. The temporal difference is computed by dividing sample difference by the sampling frequency. Then, signal shifted and the result is a pair of heterogeneous sensors aligned temporally; the Kinect and accelerometer, see Figure 3.15. Then, heterogeneous sensors are ready for further processing steps.

3.8.2 Verification of the Accelerometer

We used the accelerometer as a simple and inexpensive ground truth of peak frequency measurements of PD tremors. After integrating and low pass filtering accelerometer to find position measurements, we are able to estimate tremor frequency accurately. Vicon motion capture device provides accurate trajectory profile of target objects by capturing non-symmetrically attached markers on human body and Vicon provides a gold standard motion and pose estimates in 3D. Also Vicon provides tunable frequency that enables to easy alignment with proposed trackers. Therefore, we verified the accelerometer’s frequency estimates by comparing with the Vicon which measures motion in sub-mm accuracy. We placed a green marker to the subject’s wrist and the accelerometer. We collected limb tremor symptoms from PD patients’ wrist for twenty seconds of duration. Vicon and the accelerometer measurements are aligned temporally, and two seconds of tremor symptoms utilized for computing frequency estimations for the Vicon and the accelerometer’s measurements. After systematically evaluating POFF algorithm, we concluded that the accelerometer’s peak frequency estimates are very close to that of the Vicon.

3.9 Summary of the Chapter

In this chapter, we have discussed fusion of depth and RGB images by utilizing depth and optical flow measurements. We have utilized from the Kinect as a remote sensor and utilized from accerometers for finding ground-truth motion for PD tremors. Additionally, we utilized from a green marker to create a baseline method for PD tremor measurement. We have tested our proposed tremor quantification algorithm on both PD tremors and healthy

samples that created small limb motions. In the next chapter, we will discuss graphical inference methods.

Chapter 4

Robot Localization Through Graphical Inference Method

4.1 Introduction

Inference problems are widely studied research field for finding route for robots, audio signal processing, image recognition, navigation, indoor positioning etc. Probabilistic graphical methods can be utilized for inference. Due to abundance and importance of robot applications, accuracy and robustness of the inference algorithms are important for applications. Markov Random Field (MRF) is a probabilistic model that typically modeled with graphs. Bayesian inference is a classic probabilistic inference approach similar to MRF and can be a powerful inference method with known conditional and prior probabilities of samples. However, Bayesian inference can be failed for problems with independent assumptions, complex problems, and inferring hidden variables where proper prior probabilities are not known properly. In this cases, probabilistic graphical inference models can improve prediction accuracy.

Probabilistic graphical models defines probability distribution of samples are dependent to neighbor samples in the network. Information can be defined on a matrix as a probabilistic graphical model and joint probability of samples can define information of a model.

Probabilistic graphical model can be defined with M variables $X = (x_1, x_2, x_3, \dots, x_m)$ where m equals to number of image pixels. Joint probability of graphical model samples can convey hidden variable information. It is required to be collected some observations from graphical model and hidden variable can be estimated. Therefore, given $P(y)$ prior probabilities, graphical models seeks to estimate conditional probability $P(x_i|y)$. $P(x_i|y)$ can be found by marginalization, and $P(x_i|y)$ can define depth of unobserved pixel for image inference problems, channel estimation of communication system etc.

We explain basic principles of graphical network and mechanism. We have studied Belief Propagation (BP) which can be effectively solve hidden values for network problems. Images can be defined as a network that comprised of nodes, states, and edges. Image pixels can be defined as a node and every node has a state value. If two nodes are dependent, those nodes are connected to each other with edges. If nodes are statistically independent, those nodes are not connected with edges. The network information can be spread to neighbor nodes with message passing mechanism, shown in Figure 4.1. Messages can flow through edges that connects nodes and message passing mechanism can recursively spread information through updating local computations at the nodes. Then, information at the nodes are updated through message passing.

4.1.1 Message Passing Rule

Messages convey state value of the nodes that passes to neighbor nodes from node j to node i . Message from node j to i can be calculated with sum of products and there is a rules to be completed for message passing, see Eq 4.1. It is need to be multiplied all messages coming to node j , multiply with compatibility function $\psi_{ij}(x_i, x_j)$, and marginalize over the variable x_j .

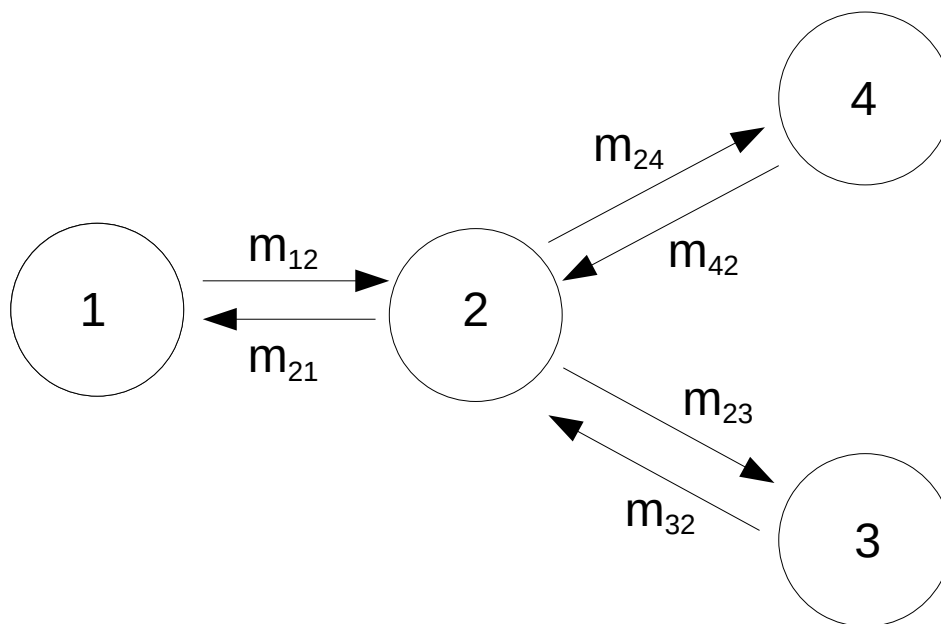


Figure 4.1: Messages passing from one node to neighbours, so information flows to neighbor nodes.

denote neighbors of node j . Probability of a node is equal to product of all coming messages to the node at Eq 4.2. Node send message to its all neighbors. Then, messages flow from edges to all directions and state of the nodes are computed and updated for inference. BP can be run through loops, so it is termed as Loopy Belief Propagation (LBP). LBP can be run until it converges and stopped where there is no significant change on nodes.

$$m_{ji}(x_i) = \sum_{x_j} \psi_{ij}(x_i, x_j) \prod_{k \in n(j) \setminus i} m_{kj}(x_j) \quad (4.1)$$

$$P_i(x_i) = \prod_{j \in n(j) \setminus i} m_{ji}(x_i) \quad (4.2)$$

4.2 Summary of the Chapter

In this chapter we have presented (LBP). LBP is a dynamic programming approach and idea behind LBP is to run LBP on a graph containing loops until convergence. Convergence is not guaranteed and LBP can oscillate between states at nodes when network is singly connected. LBP algorithms can be utilized for simultaneous localization and mapping (SLAM) problems which can be enhance prediction performance.

Chapter 5

Unstructured 3D Shape Matching Algorithm

5.1 Rigid Object Pose Estimation

An important component of scene understanding involves detecting objects and estimating their poses, namely their positions and orientations. Now in dynamic scenes, where objects are displaced or move on their own, estimating change in pose of objects becomes critical to enable humans and robots to safely operate in an environment and to interact with objects. This is evidenced by numerous applications involving active pose estimation including autonomous vehicles, robot navigation, biomedical robots, human-machine interaction, action recognition [84, 85, 86, 87, 88, 89, 90, 91, 92] etc. Pose estimation has been, and continues to be, an important focus of computer vision research as is challenging for a number of reasons. Instances of the same category of objects can vary greatly in appearance making pose comparisons difficult. With a single sensor only a portion of an object is visible to an observer and occlusions can reduce this further, limiting data available to infer pose. And most importantly, ambiguities and indeterminacies due to camera projection limit pose accuracy; these include the scale-depth ambiguity, and the similar effect of small rotations and translations. Consequently accurate pose estimation is not yet solved and remains an

important research topic.

Relative pose estimation can be distinguished from absolute pose estimation. The latter involves a prior model of an object that defines a fixed origin and orientation. A good example is a car where the coordinates at the center of its 3D bounding box can specify its absolute pose. Acquiring models from which absolute pose can be determined may be data intensive, as in 3D vehicle detection [93], and may be ambiguous or impractical to define for objects with variable shapes such as tables, chairs and utensils. On the other hand, relative pose captures the change in pose of an object from one time point to another. Estimating relative pose does not require a prior shape model or even object recognition. At the same time relative pose estimation can be used as an input to object tracking and for applications such as collision avoidance, robotic grasping, remote motion estimation with accuracy being particularly important. This research focuses on achieving high accuracy relative pose estimation. Therefore, accurate and partial object model based relative pose estimation is of great importance for space craft robots, medical surgery, and autonomous vehicles.

One solution for pose estimation problem is 3D template matching method, the image template is created by rendering a 3D shape model of an object. Template based pose estimation algorithms such as CPD are widely studied in literature [94], [95]. CPD models model point cloud as a mixture of gaussians and models target as data points and seeks maximum likelihood. Some algorithms estimate pose from point cloud information, ICP utilizes point clouds and converges to local minimum distance of point clouds iteratively, and proposes high accuracy estimates in some cases. Upon reaching the local minimum distance, ICP computes pose estimates by determining the closest distance and computes

spatial transformation for point sets. Chen et al. [96] developed a variant of pose algorithm is point to plane ICP algorithm that searches point correspondences and minimizes the defined error metric. Although the approach can yield promising estimates, performance of ICP algorithm is lower for planar surfaces, non-Gaussian noisy measurements, occlusions, and outliers on points.

Feature based pose estimation methods have a wide range of applications. The general idea is to estimate feature matches and descriptors from model and target frames which is expected to be robust to image deformations in an object then estimate pose measures of the object by error minimization, voting scheme etc. Feature based pose estimation methods can be divided to local and global methods. To capture accurate poses, image frames are required to have sufficiently enough texture on model and target object of interest. Chen et al. [47] utilized optical flow measurements that help to find large displacements and their algorithm finds poses by combining template warping and using Scale Invariant Feature Transform (SIFT) feature correspondences. Liu et al. [56] proposes a novel feature called P2P-TL and their algorithm models target appearance that reduces computation time and increases accuracy of pose estimation. Teng et al. [97] developed an algorithm for finding poses of aircrafts, their algorithm extracts line features and pose parameters are computed by processing line correspondences. Quan et al. [98] proposed a novel voxel based binary descriptor that makes 3D binary characterization on object geometry and their algorithm computes pose estimation and fine registers by matching features of point cloud. Liu et al. [99] developed an algorithm and finds pose by matching edge features, their algorithm's estimation accuracy is highly dependent to representative edge features and object geometric shape. Contour based methods are also widely studied in pose estimation algorithms, contours can present

accurate edge information on a model object. Leng et al. [100] proposed a pose estimation algorithm extracts model and target contours from gray image, and iteratively searches for matching until convergence. Schlobohm et al. [101] utilized contours and proposed projected features that increased accuracy of the pose estimation, their algorithm finds pose by global optimization method. Zhang et al. [102] proposed an algorithm that utilizes shape and image contour. Their algorithm finds inliers, rejects outlier points intensively and finds the pose of the object. Similarly, Wang et al. [103] also utilized from image contours and edge features, their algorithm applies particle filter searches for improved matches. In doing so, their algorithm produces robust pose estimations in cluttered conditions.

Recently, machine learning based pose estimation algorithms have been proposed extensively, those methods need pre-training, present automatic segmentation, and pose estimation. Machine learning based methods aim to learn feature descriptors or find pose of the object with CNN's. Zeng et al. [104] developed a Convolutional Neural Networks (CNN) based pose estimation algorithm for robotic manipulators, the algorithm is implemented to robot that able to pick and place tasks automatically. Le et al. [105] proposed a CNN network that segments object and applied pose estimation tasks to robotic applications. Brachmann et al. [106] developed a pose estimation algorithm method by utilizing a random forest algorithm for pixel classification of RGBD frames. Kendall et al. [107] fine-tuned pre-trained network (GoogLeNet) and proposed a CNN network computes camera pose through color imagery. Hua et al. [89] developed a human pose estimation algorithm utilizing color imagery and hourglass networks, they added a residual attention module to the network as a residual connections structure. Giefer et al. [108] developed an algorithm comprised of two cascade CNNs for one localization of object and second network is to compute pose of

the object. Though learning based pose estimation methods have a high potential, they can be limited for learning different geometric poses, invariances, and computational time.

In this study, we have tested our proposed algorithm (Flow Filter) on BigBird public dataset that includes large scale 3D database for multiple object’s pose information that can be freely accessible online [109]. Rigid objects are placed on a circular and rotational desk that rotated from 0 to 180 degree with respect to center of rotational desk. The dataset provides a ground truth pose measurements, five RGBD camera frames (Carmin sensor), meshes, and camera parameters. Additionally, BigBird dataset also provides high resolution RGB camera (Canon) frames for object poses. To test Flow Filter algorithm on a more challenging task, we have tested proposed algorithm on low resolution RGBD images that provided from Carmin sensor.

Rigid object pose defines finding best alignment of model and target object and transformation of rigid object can be quantified in terms of a R and a T . Our proposed algorithm computes relative pose from RGBD frames (model and target), doesn’t require pre-defined CAD model, markers, wearable units, cable connection, and object training. Partial or complete object frame is enough to compute relative pose estimation. Our Flow Filter algorithm utilizes from extracted point cloud and optical flow information, doesn’t utilize from image features. Our proposed algorithm integrated on to FilterReg pose estimation algorithm that produces fast and robust matching [110], and Flow Filter upgrades FilterReg to higher accuracy pose estimation method.

In the remainder of this Chapter, we have explained related publications in Section 1, we addressed problem definition and details of Flow Filter algorithm at Section 2. We explained experiments at Section 3 and summarized our findings and future development

and contributions at Section 4.

5.2 Proposed Framework

We proposed a relative pose estimation algorithm (Flow Filter) that fuses flow measurements and 3D point cloud information and integrated on FilterReg pose estimation algorithm. Pose estimation can be computed without priori information, flow and point coordinate information can be utilized to refine 3D pose. Flow Filter algorithm can be depicted, see Figure 5.1 and the algorithm can be shown series of steps, see Figure 5.2. Flow Filter is applied to compute relative pose estimation for rigid objects. We have tested our algorithm on Big-Bird public dataset, and it has shown that our proposed algorithm produces enhanced pose estimation accuracy with respect to related pose estimation algorithms. Flow Filter algorithm computes relative pose through two main steps. Details of the Flow Filter algorithm is explained in the remainder of this section.

We can define initial pose (model pose) of the object (R_1, T_1) , define secondary pose (target pose) of the object (R_2, T_2) , and relative pose (R_1^2, T_1^2) can be found from model and target pose, see Eq 5.1 and Eq 5.2. Model point cloud data (P_m) and the target point cloud data (P_t) can be defined as point clouds and we can transform model to target point cloud by using relative Rotation (R_1^2) and translation (T_1^2) , see Eq 5.3. R matrix can be defined function of angles and can be termed as Eq 5.4 and T can be defined with displacements in 3D, see Eq 5.5. Accuracy of our algorithm can be tested and quantified with ground truth pose estimates. T_1^2 can be defined as a vector and can be termed as $T_1^2(t_x, t_y, t_z)$. In our case, rigid objects are positioned on the center of rotational desk, R_1^2 will change in single axis and T_1^2 equals to zero with respect to origin of rotational desk. R_1^2 can be defined

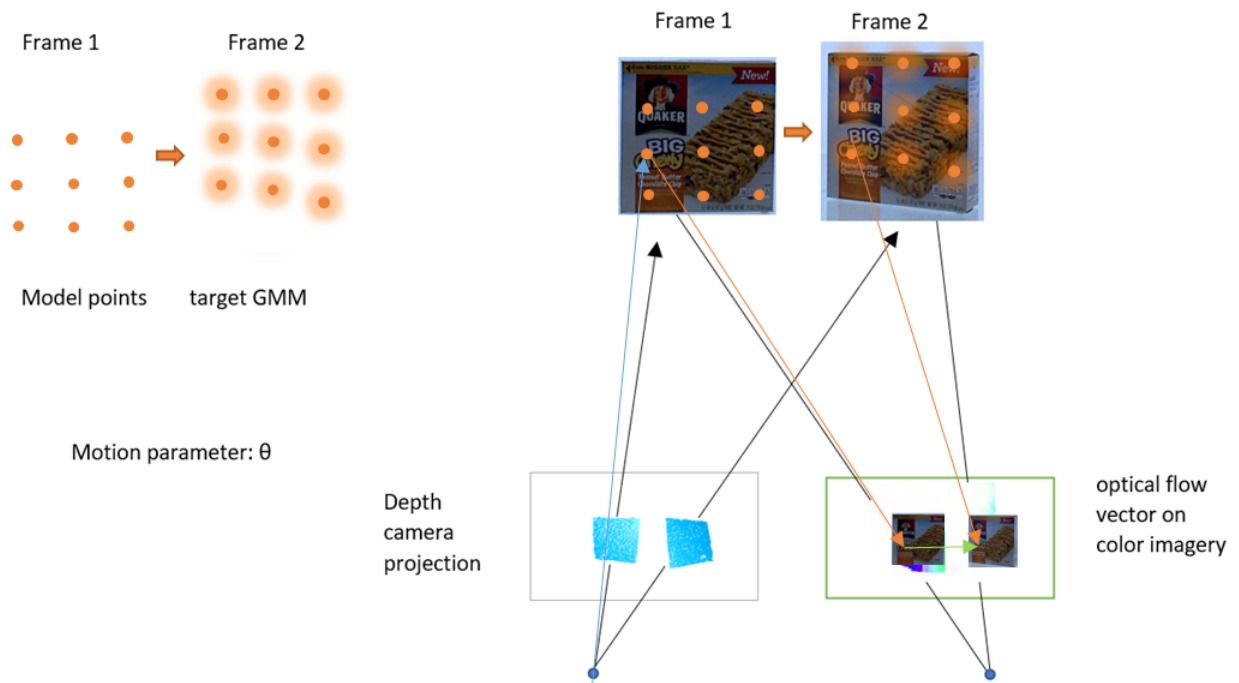


Figure 5.1: Relative pose is refined through FilterReg that treats target points as a GMM. Maximum likelihood is estimated from point cloud registration on GMM, point clouds are created and projected on color imagery. Optical flow measurements are fused with point cloud

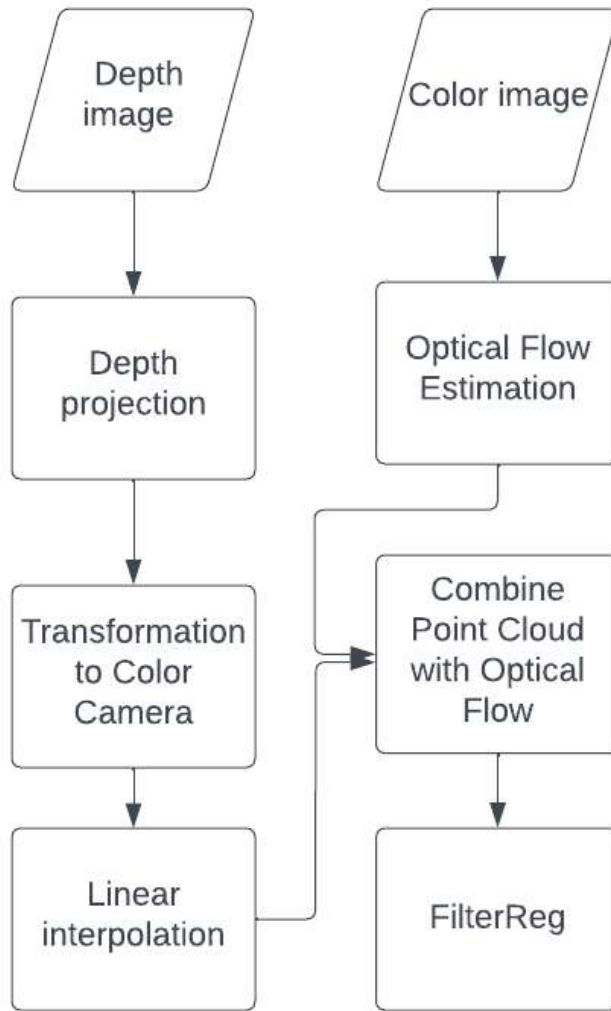


Figure 5.2: Proposed algorithm utilizes from RGBD imagery, fuses depth and optical flow information, refines relative pose.

with a single angle and can be quantified as axis-angle value, so we have transformed R_1^2 to axis-angle values.

$$R_1^2 = R_2 \cdot R_1^{-1} \quad (5.1)$$

$$T_1^2 = T_2 - T_1 \quad (5.2)$$

$$P_t = R_1^2 \cdot P_m + T_1^2 \quad (5.3)$$

$$R_1^2 = \begin{bmatrix} \cos \beta \cos \gamma & \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma & \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma \\ \cos \beta \sin \gamma & \sin \alpha \sin \beta \sin \gamma - \cos \alpha \cos \gamma & \cos \alpha \sin \beta \sin \gamma + \sin \alpha \cos \gamma \\ -\sin \beta & \sin \alpha \cos \beta & \cos \alpha \cos \beta \end{bmatrix} \quad (5.4)$$

$$T_1^2 = [t_x, t_y, t_z]^T \quad (5.5)$$

5.2.1 Depth Pixel Transformation and Projection

Flow Filter algorithm needs to know depth camera points and corresponding RGB coordinates for model and target. Since we have camera intrinsic and extrinsic parameters for depth and color cameras, the algorithm projects depth points to temporary coordinates defined somewhere in our depth camera. Then, projected depth point cloud in temporary coordinate system is projected on color camera coordinate system. Then, points which are

defined on color camera system projected on to color imagery required to be inside the model object boundary. The algorithm transforms and projects depth image pixels on color image that gives depth pixels and corresponding color image pixels. Point projection can create noise on object boundaries which may lead to incorrect pose matches, so our proposed algorithm filters noisy and far away points from model and target object. Target depth image can include some sparse points projected color images. To find all depth values inside object boundary, our proposed algorithm applies linear interpolation and masking on target imagery, Figure 5.3. Then, Flow Filter finds 3D point cloud on color imagery that presents the rigid object in model and target frames.

5.2.2 Pose Estimation

Model and target object depth point projection on color imagery have been completed at prior step and we have no priori pose estimation. Proposed algorithm also required to know optical flow measurements. Optical flow is computed from color image frame with the method that works based on image warping and CNN networks [111], and their algorithm produces promising estimations for low-textured objects and small motions. Our proposed algorithm uses optical flow from model to target frame that gives 2D color image correspondences. Flow Filter masks flow points inside of the object boundary and fuses optical flow with corresponding 3D point cloud information. Finally, our proposed algorithm is integrated on FilterReg algorithm [110], and computes relative pose by modeling and registering point clouds as a probabilistic point set method. We termed point cloud information as (X, Y, X) which is gathered from depth projection. (X_f, Y_f, Z_f) denotes the combination of optical flow point cloud correspondences of projected and interpolated depth map.

$$(x, y) = P(X, Y, Z) \quad (5.6)$$

$$(x_f, y_f) = f(x, y) \quad (5.7)$$

Update Point Cloud:

$$(X_f, Y_f, Z_f) = P^{-1}(x_f, y_f) \quad (5.8)$$

$$(X, Y, Z) = (X_f, Y_f, Z_f) \quad (5.9)$$

E step:

$$M_{x_i}^0 = \sum_{y_k} N(x_i(\theta^{old}); y_k, \sum_{xyz}) \quad (5.10)$$

$$M_{x_i}^1 = \sum_{y_k} N(x_i(\theta^{old}); y_k, \sum_{xyz}) y_k \quad (5.11)$$

M step:

$$\sum_{x_i} \frac{M_{x_i}^0}{M_{x_i}^0 + c} (x_i(\theta) - \frac{M_{x_i}^1}{M_{x_i}^0})^T \sum_{xyz}^{-1} (x_i(\theta) - \frac{M_{x_i}^1}{M_{x_i}^0}) \quad (5.12)$$

where θ defines motion parameters, X defines model point cloud in 3D and Y defines target point cloud, P defines projection of 3D camera coordinates to image pixel coordinates see Eq 5.6, f defines optical flow pixels in 2D and used to find displacements Eq 5.7 and can be projected to camera coordinates, see Eq 5.9. $M_{x_i}^0$ and $M_{x_i}^1$, are computed at E-step see

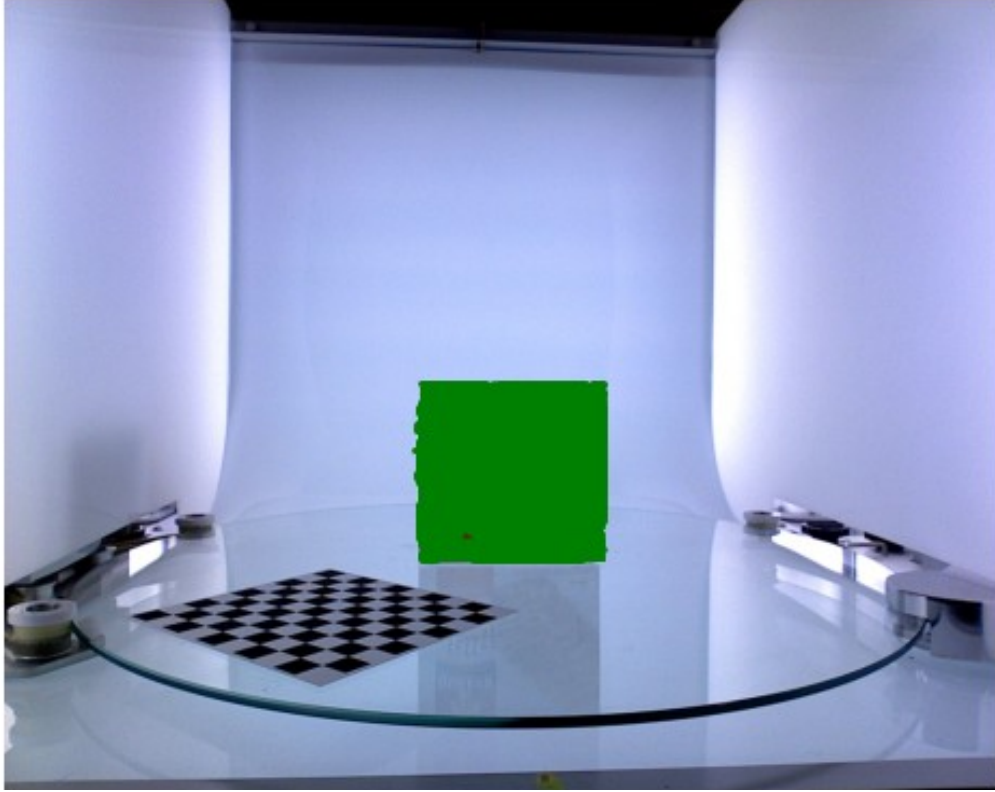


Figure 5.3: Depth points are masked, so object depth measurements and color image coordinates are known.

Eq 5.10, Eq 5.11 and M step minimizes the objective function, see Eq 5.12.

5.3 Experiments

We have tested our Flow Filter algorithm on Bigbird object pose dataset and test results have been compared with CPD and FilterReg pose estimation algorithms. We have completed tests on 12 objects and test objects have varied sizes, shape, color, and texture properties, see Figure 5.4. Test objects are rotated from 3 to 30 degrees. Since objects are not translated, rotated on single axes, we have reported results in terms of angle error values of test objects. Rigid objects are placed on a circular and rotational desk that rotated from 0 to 180 degree with respect to center of rotational desk. The dataset provides ground truth

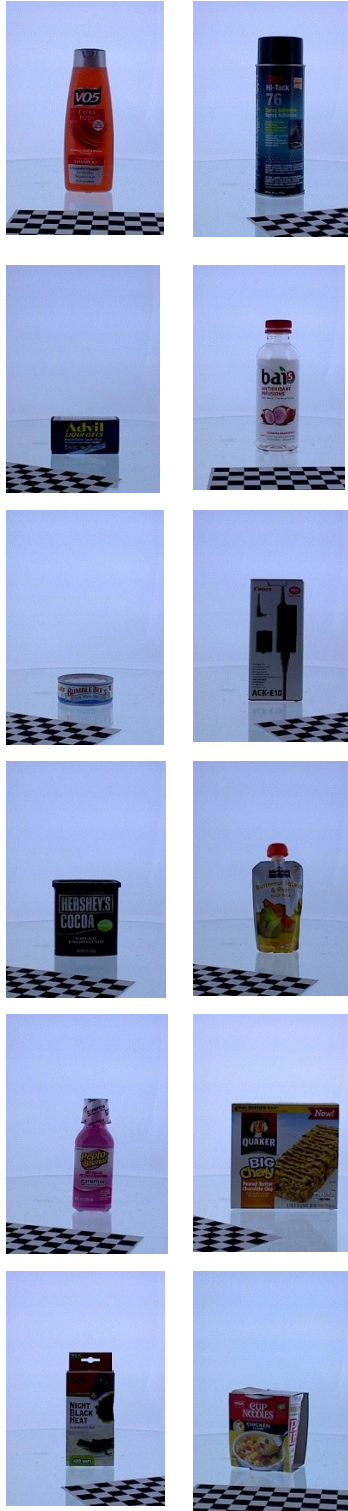


Figure 5.4: Relative Pose of rigid objects which are different shaped and textured are computed with our algorithm, CPD, and FilterReg. Pose estimations are compared with ground truth.

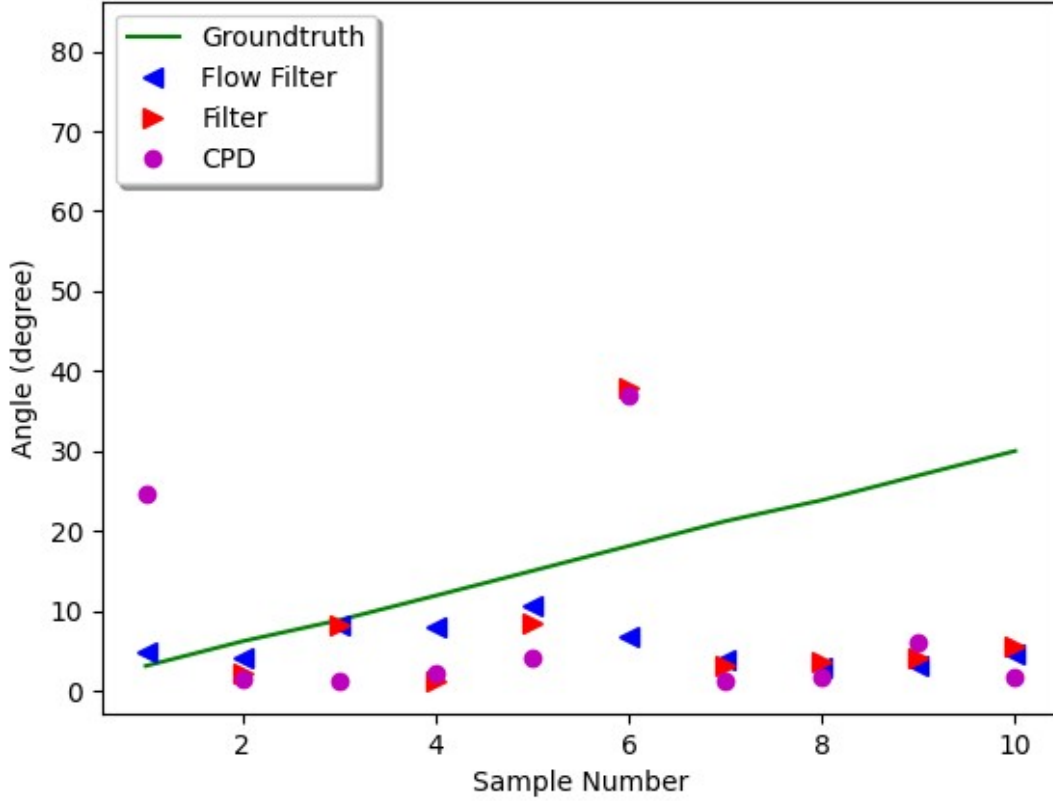


Figure 5.5: Axis angle of R is plotted for Ground truth, proposed algorithm, FilterReg, and CPD for canned fish.

pose measurements, five RGBD camera frames (Carmine sensor), meshes, and camera parameters. Additionally, BigBird dataset also provides high resolution RGB camera (Canon) frames for object poses. To test Flow Filter on a more challenging task, we have tested the algorithm on low resolution RGBD images that are provided from Carmine sensor.

5.4 Results

We have tested our Flow Filter algorithm to evaluate rotation estimations by comparing ground truth poses. Mean absolute error results have been quantified for Flow Filter, FilterReg, and CPD by utilizing ground truth measurements. The model object is rotated and

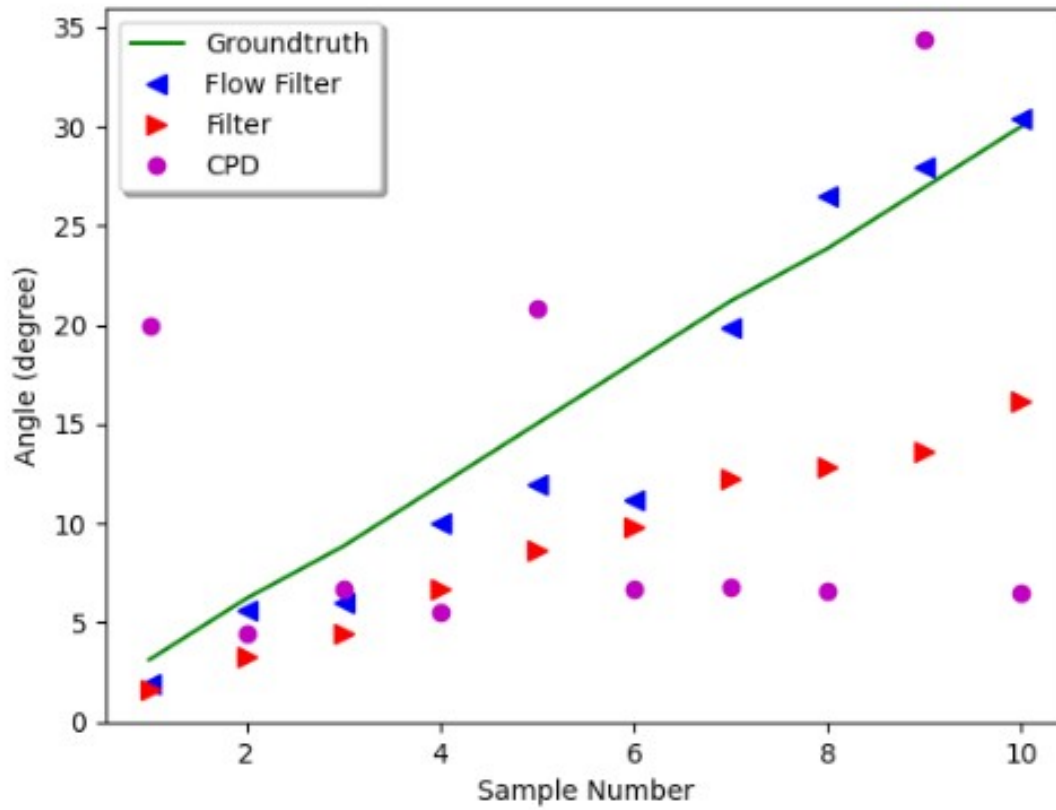


Figure 5.6: Axis angle of R is plotted for pepto liquid bottle, proposed algorithm includes decreased error with respect to FilterReg and CPD.

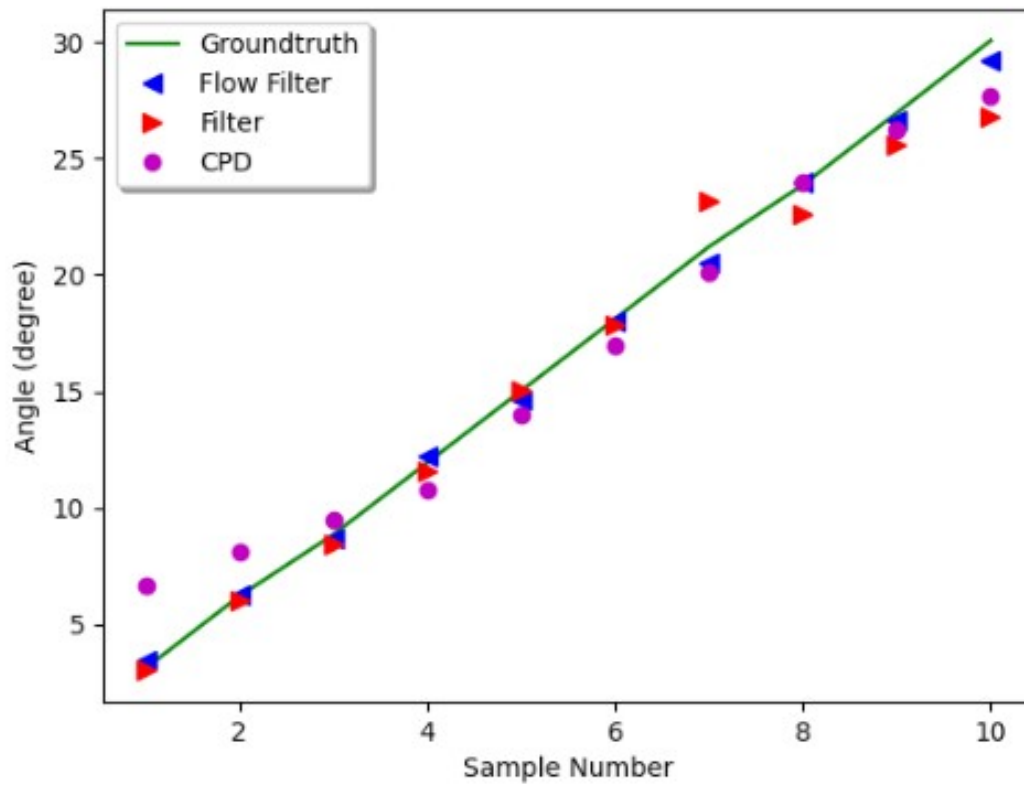


Figure 5.7: Pose estimation gives good matches and low error in terms of axis angle for proposed algorithm, FilterReg, and CPD for Granola box.

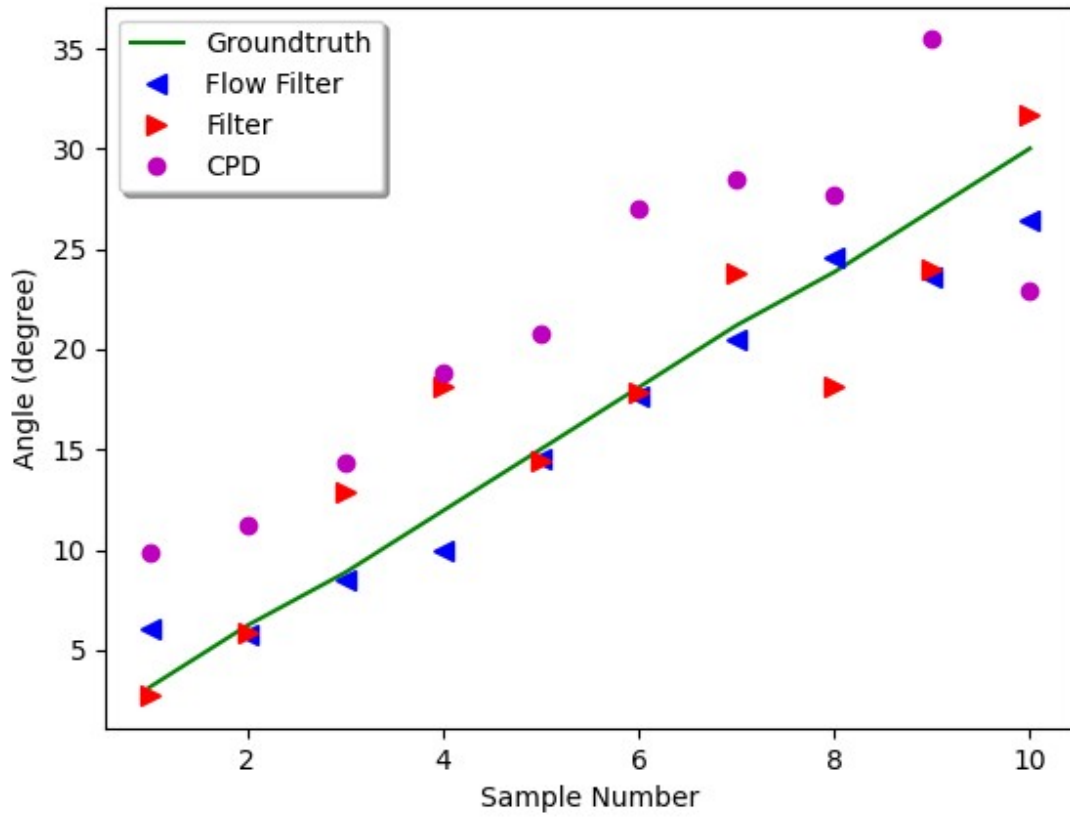


Figure 5.8: Pose estimation gives enhanced estimation accuracy in terms of axis angle with respect to FilterReg and CPD for Chicken noodle box.

not translated. It is quantified that Flow Filter algorithm computes pose of the rigid object in enhanced accuracy than FilterReg and CPD algorithm. Instead of using plain depth measurements of model and target frames, Flow Filter enables higher accuracy pose estimations by fusion of depth measurements and corresponding flow displacements. Rotational error is quantified in terms of angle of R , so we have computed angle of error by finding axis-angle values of R . Mean absolute errors are computed from axis-angle values and results have been reported.

Advil box is alike canned fish in terms of object size which causes limited number of data points. Angle error is quantified as 2.95, 8.69, and 5.8 for proposed algorithm, FilterReg, and CPD respectively. Though our proposed algorithm estimation is less than 3 for advil box and produces higher accuracy with respect to FilterReg and CPD algorithms. Object shape information can be impacted estimation accuracy. Canned fish is a small sized object as advil box, canned fish is a cylindrical shaped object that can be hard to track due to shape and limited number of points on the object. Therefore, pose results have shown that all pose estimation algorithms suffer for relative pose estimation, see Figure 5.5. Angle error is quantified as 11.12, 20.62, and 16.43 for Flow Filter, FilterReg, and CPD respectively. Optical flow can be suffered as angle of R increases that increases noise on object data points. Flow Filter produces 2.19 angle error for Pepto liquid bottle, see Figure 5.6. FilterReg and CPD produces less than 1 angle error for few samples, but amplitude of error enhances as angle of R enhances because partial target object shape differs from target frame. FilterReg produced 7.57 angle error and CPD produced 10.69 angle error. Our proposed algorithm, FilterReg and CPD produces promising results for Quaker granola box because the object shaped as a rectangular prism and textured surface which can be relatively easier to track

pose because object has a simple and regular shape that point cloud sets can converge to minimum, see Figure 5.7. Angle errors are quantified as 0.32, 0.91, and 1.37 for proposed algorithm, FilterReg, and CPD respectively. Chicken noodle box has some texture on the package, and we have tested pose estimation algorithms, see Figure 5.8. Flow Filter produces 1.51 angle error for chicken noodle box. FilterReg produces much higher error than proposed algorithm, quantified as 8.0 mean angle error. Similarly, CPD algorithm produces false alignment for chicken noodle box, produces 9.33 angle error. Overall mean angle error values are equal to 3.03, 8.0, 9.33 for our proposed algorithm, FilterReg, and CPD respectively. As it can be seen results on variety of test objects, it is clearly seen that amplitude of angle error can be changed depending on object size, shape, and texture properties of test objects.

5.5 Summary of the Chapter

Depth cameras are generally lower resolution than color cameras, so pose estimation from depth only measurements can be suffered from accuracy and robustness that can be problem for error sensitive applications. RGB images cameras are generally higher resolution and cheaper than depth cameras. Therefore, RGB cameras can be combined to depth cameras which enhances pose estimation accuracy. Low resolution cameras give a limited number of data points on tracked objects that can decrease estimation accuracy. Similarly, object shape and texture information impacts pose estimation accuracy. To present cheap and accurate pose estimation, Flow Filter utilizes pre-calibrated cameras and RGBD images. Flow Filter fuses color and depth information and rejects outliers that produce robust and enhanced pose accuracy which can be a solution for error critical applications. Flow Filter algorithm can be implemented on RGBD sensors as Kinect that enables cheap and efficient pose estimation

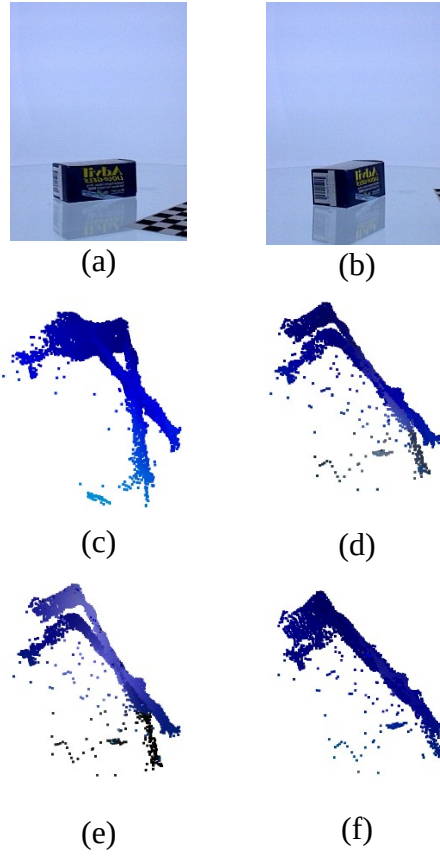


Figure 5.9: (a) model RGB image (b) target RGB image (c) model and target image clouds (d) alignment from CPD includes pose shift (e) alignment with FilterReg includes pose shift (f) alignment with proposed algorithm matches.

for indoor environments. We have tested Flow Filter algorithm for low resolution depth and RGB images and Flow Filter algorithm provided decreased pose error than CPD and FilterReg algorithms. Other algorithms such as CPD can produce false matches for R. Flow Filter utilizes optical flow and our algorithm's accuracy decreases for large rotations due to problems for optical flow. Flow Filter can be applied to real time applications and relative pose estimation applications.

Chapter 6

Experimental Results

In this chapter, we present the experimental results of the proposed frameworks. We compared our results with ground truth and comparable algorithms. We have tested our algorithms for series of experiments that can be affected estimation accuracy. We have tested our proposed algorithms for data gathered from laboratory and public datasets.

6.1 Results And Discussion

Firstly, We presented our results on PD tremors which are collected from MSU clinic and results are given, see Table. Secondly, we presented our results for rigid object relative pose estimation problem and our proposed frameworks has been tested on BigBird pose dataset.

6.1.1 PD Tremor Estimation Problem

We apply our POFF algorithm to capture PD tremors which can come and go, and can be small or high amplitude. Amplitude of motion, pose of patients, distance, and clothing can be affected motion estimation accuracy. Therefore, we have tested our POFF algorithm for series of detailed experiments, see Table 6.1. To track small tremors, our method utilizes from RGB imagery and depth projections that gives higher resolution than commercial depth cameras. We have applied our POFF algorithm and our POFF produced promising results

on PD tremors, see Table 6.2.

Table 6.1: Results are reported for tremor estimation. Amplitude, distance, pose, clothing have been changed and tested for detailed experiments.

		Kinect	POFF	Marker
Amplitude	< 5 mm	0.51	0.057	0.061
	5-10 mm	0.415	0.095	0.037
	> 10 mm	0.032	0.031	0.045
Distance	2 m	0.485	0.051	0.087
	3 m	0.461	0.068	0.049
	4 m	0.313	0.056	0.069
Pose	stretched arm	0.405	0.063	0.05
	Vertical arm	0.557	0.15	0.126
	Limb touching	0.668	0.138	0.073
Clothing	t-shirt	0.485	0.051	0.087
	Sleeve	0.461	0.068	0.049
	Jacket	0.313	0.056	0.069
Combined	Amplitude < 10 mm	0.4612	0.0775	0.0688

Table 6.2: Frequency errors are given for stretched arm, frequency difference from ground truth has been found as an error.

	Kinect	POFF
Stretched arm	0.661	0.159

6.1.2 Rigid Object Pose Estimation Problem

First, we apply our proposed methodologies to rigid object pose dataset and our proposed algorithm produced higher accuracy than CPD and FilterReg pose estimation algorithms, our pose matches fit with ground truth pose, see Table 6.3.

Table 6.3: Errors found by finding angle difference of algorithms and ground truth angles. Our algorithm produces decreased error than comparable algorithms.

	Ours	FilterReg	CPD
Mean Error (angle)	3.032	8.01	9.33

6.2 Summary of the Chapter

In this chapter we have presented detailed experiments on test the problems using all the frameworks presented in this thesis. We have also presented results on proposed motion and pose estimation algorithms. Our POFF algorithm enhanced Kinect’s motion estimation accuracy. POFF algorithm able to recover PD tremors up to 2 mm amplitude. However, Kinect’s estimations are noisy and incorrect under 1 cm amplitude tremors.

Our proposed rigid object pose estimation algorithm extracts object camera coordinates and fuses 3D camera coordinates with optical flow. As a result, our proposed algorithm estimates relative pose in higher accuracy.

Chapter 7

Conclusion and Future Work

In this chapter, we state concluding remarks of this research and we present some future research plans for solving motion and pose estimation problems including PD limbs tremors and rigid objects.

7.1 Conclusion

In this thesis, we found some real world problems regarding motion and pose estimation. Therefore, we have studied fusion methods for motion-pose estimation problems and robot localization problems by utilizing from graphical inference methods.

To be particular, we have studied PD motion and pose estimation methods which provided enhanced pose estimation results on data-sets. We proposed a data fusion technique that helps to estimate small PD tremors and we have fused accelerometer and RGBD camera motion estimates with Extended Kalman Filter that applicable to homogeneous sensor readings. Our algorithm provided a method that able to track mm-level tremor estimates. Additionally, we presented temporal calibration method that can be applicable to align heterogeneous sensor measurements.

We have fused depth and color measurements that gave accurate pose estimation results on BigBird data-set. We have tested our algorithm for 12 different objects and we have

concluded that our algorithm estimates include lower error than CPD, FilterReg. we can achieve better results in terms of accuracy and robustness for pose estimation problems using RGBD camera.

Multi sensor fusion techniques can be studied and that can enhance motion and pose estimations. EKF, Particle Filters, Extended Kalman Particle Filter algorithms are popular and can be efficiently applicable to wearable and visual sensor fusion problems.

7.2 Future Work

In future, we would like to extend our work in the following way:

- In the literature, we have seen most of the researchers utilized from EKF to fuse accelerometer and gyroscope readings which can enhance pose or motion estimation accuracy. EKF, Particle Filters can be utilized to fused sensors. We are planning to use EKF to fuse gyroscope and acceleromater readings from a wearable sensor. In addition to this, Particle Filters can be applicable to fuse wearable unit motion and camera based pose estimations which can give robust, accurate, and efficient results.
- Laser sensors can be applicable PD tremor estimate problems. Laser scans can be utilized to extract contour feature extraction of limbs and background information can be filtered. Then, point cloud based pose estimation methods can be utilized to find contour association of model and target templates that can provide accurate limb tremor estimates.
- We are studying on Robot Localization methods and we have studied0 graphical inference methods such as LBP. We plan to apply LBP algorithm to robot localization

problems that can give accurate results because LBP can give good inference for image based problems.

- An interesting application of pose estimation is for tracking human motion for interactive gaming applications. That can help developing interactive control of robots and machines etc.
- Deep neural network (DNN) can be used to automatic pose estimation and dataset for heat maps of human motion can be utilized to train DNN's. Therefore, DNN can be studied in future and applicable to PD tremor estimations and robot pose estimation. Also, DNN networks can be trained for multi-person PD tremor estimation problem.

BIBLIOGRAPHY

- [1] R. Zhu and Z. Zhou, “A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package,” *IEEE Transactions on Neural systems and rehabilitation engineering*, vol. 12, no. 2, pp. 295–302, 2004.
- [2] H. Dai, P. Zhang, and T. C. Lueth, “Quantitative assessment of parkinsonian tremor based on an inertial measurement unit,” *Sensors*, vol. 15, no. 10, pp. 25 055–25 071, 2015.
- [3] D. Dutta, S. Modak, A. Kumar, J. Roychowdhury, and S. Mandal, “Bayesian network aided grasp and grip efficiency estimation using a smart data glove for post-stroke diagnosis,” *Biocybernetics and Biomedical Engineering*, vol. 37, no. 1, pp. 44–58, 2017.
- [4] T. Aşuroğlu, K. Açıcı, Ç. B. Erdaş, M. K. Toprak, H. Erdem, and H. Oğul, “Parkinson’s disease monitoring from gait analysis via foot-worn sensors,” *Biocybernetics and Biomedical Engineering*, vol. 38, no. 3, pp. 760–772, 2018.
- [5] Š. Obdržálek, G. Kurillo, F. Ofli, R. Bajcsy, E. Seto, H. Jimison, and M. Pavel, “Accuracy and robustness of kinect pose estimation in the context of coaching of elderly population,” in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2012, pp. 1188–1193.
- [6] A. Beuter, M. Titcombe, F. Richer, C. Gross, and D. Guehl, “Effect of deep brain stimulation on amplitude and frequency characteristics of rest tremor in parkinson’s disease,” *Thalamus & Related Systems*, vol. 1, no. 3, pp. 203–211, 2001.
- [7] R. P. Meshack and K. E. Norman, “A randomized controlled trial of the effects of weights on amplitude and frequency of postural hand tremor in people with parkinson’s disease,” *Clinical rehabilitation*, vol. 16, no. 5, pp. 481–492, 2002.
- [8] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, “Real-time human pose recognition in parts from single depth images,” in *CVPR 2011*. Ieee, 2011, pp. 1297–1304.
- [9] W. Williem, Y.-W. Tai, and I. K. Park, “Accurate and real-time depth video acquisition using kinect–stereo camera fusion,” *Optical Engineering*, vol. 53, no. 4, p. 043110, 2014.
- [10] K. Niazmand, K. Tonn, A. Kalaras, U. M. Fietzek, J.-H. Mehrkens, and T. C. Lueth, “Quantitative evaluation of parkinson’s disease using sensor based smart glove,” in *2011 24th International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE, 2011, pp. 1–8.

- [11] M. Szumilas, K. Lewenstein, and E. Ślubowska, “Verification of the functionality of device for monitoring human tremor,” *Biocybernetics and Biomedical Engineering*, vol. 35, no. 4, pp. 240–246, 2015.
- [12] C. Camara, P. Isasi, K. Warwick, V. Ruiz, T. Aziz, J. Stein, and E. Bakštein, “Resting tremor classification and detection in parkinson’s disease patients,” *Biomedical Signal Processing and Control*, vol. 16, pp. 88–97, 2015.
- [13] A. Salarian, H. Russmann, C. Wider, P. R. Burkhard, F. J. Vingerhoets, and K. Aminian, “Quantification of tremor and bradykinesia in parkinson’s disease using a novel ambulatory monitoring system,” *IEEE Transactions on biomedical engineering*, vol. 54, no. 2, pp. 313–322, 2007.
- [14] L. Ji, H. Wang, T. Zheng, and X. Qi, “Motion trajectory of human arms based on the dual quaternion with motion tracker,” *Multimedia Tools and Applications*, vol. 76, no. 2, pp. 1681–1701, 2017.
- [15] J. I. Serrano, S. Lambrecht, M. D. del Castillo, J. P. Romero, J. Benito-León, and E. Rocon, “Identification of activities of daily living in tremorous patients using inertial sensors,” *Expert Systems with Applications*, vol. 83, pp. 40–48, 2017.
- [16] J. A. Gallego, E. Rocon, J. O. Roa, J. C. Moreno, and J. L. Pons, “Real-time estimation of pathological tremor parameters from gyroscope data,” *Sensors*, vol. 10, no. 3, pp. 2129–2149, 2010.
- [17] A. Olivares, G. Olivares, F. Mula, J. M. Górriz, and J. Ramírez, “Wagyromag: Wireless sensor network for monitoring and processing human body movement in healthcare applications,” *Journal of systems architecture*, vol. 57, no. 10, pp. 905–915, 2011.
- [18] E. Bakstein, J. Burgess, K. Warwick, V. Ruiz, T. Aziz, and J. Stein, “Parkinsonian tremor identification with multiple local field potential feature classification,” *Journal of neuroscience methods*, vol. 209, no. 2, pp. 320–330, 2012.
- [19] D. Joshi, A. Khajuria, and P. Joshi, “An automatic non-invasive method for parkinson’s disease classification,” *Computer methods and programs in biomedicine*, vol. 145, pp. 135–145, 2017.
- [20] L. Schäffer, Z. Kincses, and S. Pletl, “A real-time pose estimation algorithm based on fpga and sensor fusion,” in *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*. IEEE, 2018, pp. 000 149–000 154.

- [21] R.-S. Chang, J.-H. Chiu, F.-P. Chen, J.-C. Chen, and J.-L. Yang, "A parkinson's disease measurement system using laser lines and a cmos image sensor," *Sensors*, vol. 11, no. 2, pp. 1461–1475, 2011.
- [22] J.-L. Yang, R.-S. Chang, F.-P. Chen, C.-M. Chern, and J.-H. Chiu, "Detection of hand tremor in patients with parkinson's disease using a non-invasive laser line triangulation measurement method," *Measurement*, vol. 79, pp. 20–28, 2016.
- [23] W. J. A. Grooten, L. Sandberg, J. Ressaan, N. Diamantoglou, E. Johansson, and E. Rasmussen-Barr, "Reliability and validity of a novel kinect-based software program for measuring posture, balance and side-bending," *BMC Musculoskeletal Disorders*, vol. 19, no. 1, pp. 1–13, 2018.
- [24] P. R. Diaz-Monterrosas, R. Posada-Gomez, A. Martinez-Sibaja, A. A. Aguilar-Lasserre, U. Juarez-Martinez, and J. C. Trujillo-Caballero, "A brief review on the validity and reliability of microsoft kinect sensors for functional assessment applications," *Advances in Electrical and Computer Engineering*, vol. 18, no. 1, pp. 131–136, 2018.
- [25] L. Casacanditella, G. Cosoli, M. Ceravolo, and E. Tomasini, "Non-contact measurement of tremor for the characterisation of parkinsonian individuals: Comparison between kinect and laser doppler vibrometer," in *Journal of Physics: Conference Series*, vol. 882, no. 1. IOP Publishing, 2017, p. 012002.
- [26] F. Heinrich, T. Schmitz-Hübsch, T. Ellermeyer, S. Mansow-Model, and A. Lipp, "Video-based tremor analysis via kinect (r) system in comparison to accelerometric and electromyographical tremor detection," in *MOVEMENT DISORDERS*, vol. 31. WILEY-BLACKWELL 111 RIVER ST, HOBOKEN 07030-5774, NJ USA, 2016, pp. S329–S329.
- [27] S. T. Pöhlmann, E. F. Harkness, C. J. Taylor, and S. M. Astley, "Evaluation of kinect 3d sensor for healthcare imaging," *Journal of medical and biological engineering*, vol. 36, no. 6, pp. 857–870, 2016.
- [28] R. Torres, M. Huerta, R. Clotet, R. González, G. Sagbay, M. Erazo, and J. Pirrone, "Diagnosis of the corporal movement in parkinson's disease using kinect sensors." in *World Congress on Medical Physics and Biomedical Engineering, June 7-12, 2015, Toronto, Canada*. Springer, 2015, pp. 1445–1448.

- [29] D. Xiao, H. Luo, F. Jia, Y. Zhang, Y. Li, X. Guo, W. Cai, C. Fang, Y. Fan, H. Zheng *et al.*, “A kinect™ camera based navigation system for percutaneous abdominal puncture,” *Physics in Medicine & Biology*, vol. 61, no. 15, p. 5687, 2016.
- [30] K. A. Bieryla, “Xbox kinect training to improve clinical measures of balance in older adults: a pilot study,” *Aging clinical and experimental research*, vol. 28, no. 3, pp. 451–457, 2016.
- [31] B. Bonnechère, V. Sholukha, L. Omelina, S. Van Sint Jan, and B. Jansen, “3d analysis of upper limbs motion during rehabilitation exercises using the kinect™ sensor: development, laboratory validation and clinical application,” *Sensors*, vol. 18, no. 7, p. 2216, 2018.
- [32] X. Xu, R. W. McGorry, L.-S. Chou, J.-h. Lin, and C.-c. Chang, “Accuracy of the microsoft kinect™ for measuring gait parameters during treadmill walking,” *Gait & posture*, vol. 42, no. 2, pp. 145–151, 2015.
- [33] M. Wochatz, N. Tilgner, S. Mueller, S. Rabe, S. Eichler, M. John, H. Völler, and F. Mayer, “Reliability and validity of the kinect v2 for the assessment of lower extremity rehabilitation exercises,” *Gait & posture*, vol. 70, pp. 330–335, 2019.
- [34] M. Huber, A. L. Seitz, M. Leeser, and D. Sternad, “Validity and reliability of kinect skeleton for measuring shoulder joint angles: a feasibility study,” *Physiotherapy*, vol. 101, no. 4, pp. 389–393, 2015.
- [35] U. Puh, B. Hoehlein, and J. E. Deutsch, “Validity and reliability of the kinect for assessment of standardized transitional movements and balance: Systematic review and translation into practice,” *Physical Medicine and Rehabilitation Clinics*, vol. 30, no. 2, pp. 399–422, 2019.
- [36] E. Auvinet, F. Multon, V. Manning, J. Meunier, and J. Cobb, “Validity and sensitivity of the longitudinal asymmetry index to detect gait asymmetry using microsoft kinect data,” *Gait & posture*, vol. 51, pp. 162–168, 2017.
- [37] L. R. Reither, M. H. Foreman, N. Migotsky, C. Haddix, and J. R. Engsborg, “Upper extremity movement reliability and validity of the kinect version 2,” *Disability and Rehabilitation: Assistive Technology*, vol. 13, no. 1, pp. 54–59, 2018.
- [38] M. A. Alper, D. Morris, and L. Tran, “Remote detection and measurement of limb tremors,” in *2018 5th International Conference on Electrical and Electronic Engineering (ICEEE)*. IEEE, 2018, pp. 198–202.

- [39] D. Pagliari and L. Pinto, “Calibration of kinect for xbox one and comparison between the two generations of microsoft sensors,” *Sensors*, vol. 15, no. 11, pp. 27 569–27 589, 2015.
- [40] S. Sooklal, P. Mohan, and S. Teelucksingh, “Using the kinect for detecting tremors: Challenges and opportunities,” in *IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*. IEEE, 2014, pp. 768–771.
- [41] “Accuracy of the microsoft kinect sensor for measuring movement in people with parkinson’s disease,” *Gait Posture*, vol. 39, no. 4, pp. 1062–1068, 2014.
- [42] I. Ishii, T. Tatebe, Q. Gu, and T. Takaki, “Color-histogram-based tracking at 2000 fps,” *Journal of Electronic Imaging*, vol. 21, no. 1, p. 013010, 2012.
- [43] M. Pressigout, E. Marchand, and E. Mémin, “Hybrid tracking approach using optical flow and pose estimation,” in *2008 15th IEEE International Conference on Image Processing*. IEEE, 2008, pp. 2720–2723.
- [44] R. Krupicka, Z. Szabo, S. Viteckova, and E. Ruzicka, “Motion capture system for finger movement measurement in parkinson disease,” *Radioengineering*, vol. 23, no. 2, pp. 659–664, 2014.
- [45] X. Wei, P. Zhang, and J. Chai, “Accurate realtime full-body motion capture using a single depth camera,” *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, pp. 1–12, 2012.
- [46] B. Soran, J.-N. Hwang, S.-I. Lee, and L. Shapiro, “Tremor detection using motion filtering and svm,” in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, 2012, pp. 178–181.
- [47] K.-H. Chen, P.-C. Lin, Y.-J. Chen, B.-S. Yang, and C.-H. Lin, “Development of method for quantifying essential tremor using a small optical device,” *Journal of neuroscience methods*, vol. 266, pp. 78–83, 2016.
- [48] G. Blumrosen, M. Uziel, B. Rubinsky, and D. Porrat, “Noncontact tremor characterization using low-power wideband radar technology,” *IEEE transactions on biomedical engineering*, vol. 59, no. 3, pp. 674–686, 2011.
- [49] G. Blais and M. D. Levine, “Registering multiview range data to create 3d computer objects,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 820–824, 1995.

- [50] A. Myronenko, X. Song, and M. Carreira-Perpinan, “Non-rigid point set registration: Coherent point drift,” *Advances in neural information processing systems*, vol. 19, 2006.
- [51] Y. Chen and G. Medioni, “Object modelling by registration of multiple range images,” *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.
- [52] T. M. Iversen, A. G. Buch, N. Krüger, and D. Kraft, “Shape dependency of icp pose uncertainties in the context of pose estimation systems,” in *International Conference on Computer Vision Systems*. Springer, 2015, pp. 303–315.
- [53] D. Presnov, M. Lambers, and A. Kolb, “Robust range camera pose estimation for mobile online scene reconstruction,” *IEEE Sensors Journal*, vol. 18, no. 7, pp. 2903–2915, 2018.
- [54] F. Aghili and C.-Y. Su, “Robust relative navigation by integration of icp and adaptive kalman filter using laser scanner and imu,” *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 4, pp. 2015–2026, 2016.
- [55] M. Delavari, A. H. Foruzan, and Y.-W. Chen, “Accurate point correspondences using a modified coherent point drift algorithm,” *Biomedical Signal Processing and Control*, vol. 52, pp. 429–444, 2019.
- [56] K. Liu, W. Wang, R. Tharmarasa, and J. Wang, “Dynamic vehicle detection with sparse point clouds based on pe-cpd,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1964–1977, 2018.
- [57] P. Biber and W. Straßer, “The normal distributions transform: A new approach to laser scan matching,” in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 3. IEEE, 2003, pp. 2743–2748.
- [58] H. Hong, H. Yu, and B.-H. Lee, “Regeneration of normal distributions transform for target lattice based on fusion of truncated gaussian components,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 684–691, 2019.
- [59] T. Liu, J. Zheng, Z. Wang, Z. Huang, and Y. Chen, “Composite clustering normal distribution transform algorithm,” *International Journal of Advanced Robotic Systems*, vol. 17, no. 3, p. 1729881420912142, 2020.

- [60] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, “A model-based 3d template matching technique for pose acquisition of an uncooperative space object,” *Sensors*, vol. 15, no. 3, pp. 6360–6382, 2015.
- [61] K. Picos, V. H. Diaz-Ramirez, V. Kober, A. S. Montemayor, and J. J. Pantrigo, “Accurate three-dimensional pose recognition from monocular images using template matched filtering,” *Optical Engineering*, vol. 55, no. 6, p. 063102, 2016.
- [62] T. G. Phillips and P. R. McAree, “An evidence-based approach to object pose estimation from lidar measurements in challenging environments,” *Journal of Field Robotics*, vol. 35, no. 6, pp. 921–936, 2018.
- [63] Z. He, Z. Jiang, X. Zhao, S. Zhang, and C. Wu, “Sparse template-based 6-d pose estimation of metal parts using a monocular camera,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 1, pp. 390–401, 2019.
- [64] C.-Y. Tsai, K.-J. Hsu, and H. Nisar, “Efficient model-based object pose estimation based on multi-template tracking and pnp algorithms,” *Algorithms*, vol. 11, no. 8, p. 122, 2018.
- [65] K.-T. Song, C.-H. Wu, and S.-Y. Jiang, “Cad-based pose estimation design for random bin picking using a rgb-d camera,” *Journal of Intelligent & Robotic Systems*, vol. 87, no. 3, pp. 455–470, 2017.
- [66] F. A. Kondori, S. Yousefi, and H. Li, “Direct three-dimensional head pose estimation from kinect-type sensors,” *Electronics letters*, vol. 50, no. 4, pp. 268–270, 2014.
- [67] A. Hafeez, H. Arshad, A. Kamran, R. Malhi, M. A. Shah, M. Ali, and S. Malik, “Object recognition through kinect using harris transform,” in *1st MEDITERRANEAN INTERDISCIPLINARY FORUM ON SOCIAL SCIENCES AND HUMANITIES, MIFS 2014, Vol. 2*, 2014, p. 413.
- [68] J. Fabian, T. Young, J. C. P. Jones, and G. M. Clayton, “Integrating the microsoft kinect with simulink: Real-time object tracking example,” *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 1, pp. 249–257, 2012.
- [69] E. E. Stone and M. Skubic, “Fall detection in homes of older adults using the microsoft kinect,” *IEEE journal of biomedical and health informatics*, vol. 19, no. 1, pp. 290–301, 2014.

- [70] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments,” *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.
- [71] J. Kotovsky and M. J. Rosen, “A wearable tremor-suppression orthosis,” *Journal of rehabilitation research and development*, vol. 35, pp. 373–387, 1998.
- [72] G. Deuschl, J. Raethjen, M. Lindemann, and P. Krack, “The pathophysiology of tremor,” *Muscle & Nerve: Official Journal of the American Association of Electro-diagnostic Medicine*, vol. 24, no. 6, pp. 716–735, 2001.
- [73] C. Liu, W. T. Freeman, E. H. Adelson, and Y. Weiss, “Human-assisted motion annotation,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.
- [74] J. McAuley and C. Marsden, “Physiological and pathological tremors and rhythmic central motor control,” *Brain*, vol. 123, no. 8, pp. 1545–1567, 2000.
- [75] P. R. Burkhard, H. Shale, J. W. Langston, and J. W. Tetrud, “Quantification of dyskinesia in parkinson’s disease: validation of a novel instrumental method,” *Movement Disorders: Official Journal of the Movement Disorder Society*, vol. 14, no. 5, pp. 754–763, 1999.
- [76] J. P. Burg, *Maximum entropy spectral analysis*. Stanford University, 1975.
- [77] J. K. Nichols, M. P. Sena, J. L. Hu, O. M. O’Reilly, B. T. Feeley, and J. C. Lotz, “A kinect-based movement assessment system: marker position comparison to vicon,” *Computer methods in biomechanics and biomedical engineering*, vol. 20, no. 12, pp. 1289–1298, 2017.
- [78] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
- [79] H. Gonzalez-Jorge, P. Rodríguez-Gonzálvez, J. Martínez-Sánchez, D. González-Aguilera, P. Arias, M. Gesto, and L. Díaz-Vilariño, “Metrological comparison between kinect i and kinect ii sensors,” *Measurement*, vol. 70, pp. 21–26, 2015.
- [80] X. Xu and R. W. McGorry, “The validity of the first and second generation microsoft kinect™ for identifying joint center locations during static postures,” *Applied ergonomics*, vol. 49, pp. 47–54, 2015.

- [81] S. Calzetti, M. Baratti, M. Gresty, and L. Findley, “Frequency/amplitude characteristics of postural tremor of the hands in a population of patients with bilateral essential tremor: implications for the classification and mechanism of essential tremor.” *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 50, no. 5, pp. 561–567, 1987.
- [82] K. Yang, W.-X. Xiong, F.-T. Liu, Y.-M. Sun, S. Luo, Z.-T. Ding, J.-J. Wu, and J. Wang, “Objective and quantitative assessment of motor function in parkinson’s disease—from the perspective of practical applications,” *Annals of translational medicine*, vol. 4, no. 5, 2016.
- [83] M. Debell-Gilo and A. Kääb, “Sub-pixel precision image matching for measuring surface displacements on mass movements using normalized cross-correlation,” *Remote Sensing of Environment*, vol. 115, no. 1, pp. 130–142, 2011.
- [84] J. Cunha, E. Pedrosa, C. Cruz, A. J. Neves, and N. Lau, “Using a depth camera for indoor robot localization and navigation,” *DETI/IEETA-University of Aveiro, Portugal*, p. 6, 2011.
- [85] E. Murphy-Chutorian and M. M. Trivedi, “Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness,” *IEEE Transactions on intelligent transportation systems*, vol. 11, no. 2, pp. 300–311, 2010.
- [86] T. Yang, Q. Zhao, X. Wang, and Q. Zhou, “Sub-pixel chessboard corner localization for camera calibration and pose estimation,” *Applied Sciences*, vol. 8, no. 11, p. 2118, 2018.
- [87] M. Andriluka, S. Roth, and B. Schiele, “Monocular 3d pose estimation and tracking by detection,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 623–630.
- [88] A. Perez-Carrillo, “Finger-string interaction analysis in guitar playing with optical motion capture,” *Frontiers in Computer Science*, p. 8, 2019.
- [89] G. Hua, L. Li, and S. Liu, “Multipath affinity stacked—hourglass networks for human pose estimation,” *Frontiers of Computer Science*, vol. 14, no. 4, pp. 1–12, 2020.
- [90] X. Yang, T. Xue, H. Luo, and J. Guo, “Fast and accurate visual odometry from a monocular camera,” *Frontiers of Computer Science*, vol. 13, no. 6, pp. 1326–1336, 2019.

- [91] A. Ali, A. Jalil, J. Niu, X. Zhao, S. Rathore, J. Ahmed, and M. Aksam Iftikhar, “Visual object tracking—classical and contemporary approaches,” *Frontiers of Computer Science*, vol. 10, no. 1, pp. 167–188, 2016.
- [92] M. A. Alper, J. Goudreau, and M. Daniel, “Pose and optical flow fusion (poff) for accurate tremor detection and quantification,” *Biocybernetics and Biomedical Engineering*, vol. 40, no. 1, pp. 468–481, 2020.
- [93] H.-N. Hu, Q.-Z. Cai, D. Wang, J. Lin, M. Sun, P. Krähenbühl, T. Darrell, and F. Yu, “Joint monocular 3d vehicle detection and tracking,” 2019.
- [94] G. Blais and M. Levine, “Registering multiview range data to create 3d computer objects,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, pp. 820 – 824, 09 1995.
- [95] A. Myronenko and X. Song, “Point set registration: Coherent point drift,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.
- [96] Y. Chen and G. Medioni, “Object modeling by registration of multiple range images,” in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 2724–2729 vol.3.
- [97] Teng, Y.-F. Luo, L. Yun, G. Wang, and Y.-X. Zhang, “Aircraft pose estimation based on geometry structure features and line correspondences,” *Sensors*, vol. 19, p. 2165, 05 2019.
- [98] S. Quan, J. Ma, F. Hu, B. Fang, and T. Ma, “Local voxelized structure for 3d binary feature representation and robust registration of point clouds from low-cost sensors,” *Inf. Sci.*, vol. 444, pp. 153–171, 2018.
- [99] D. Liu, S. Arai, J. Miao, J. Kinugawa, Z. Wang, and K. Kosuge, “Point pair feature-based pose estimation with multiple edge appearance models (ppf-meam) for robotic bin picking,” *Sensors*, vol. 18, p. 2719, 08 2018.
- [100] D. Leng and W. Sun, “Contour-based iterative pose estimation of 3d rigid object,” *Computer Vision, IET*, vol. 5, pp. 291 – 300, 10 2011.
- [101] J. Schlobohm, A. Pösch, E. Reithmeier, and R. Bodo, “Improving contour based pose estimation for fast 3d measurement of free form objects,” *Measurement*, vol. 92, p. 79–82, Jun. 2016.

- [102] X. Zhang, Z. Jiang, H. Zhang, and Q. Wei, “Vision-based pose estimation for texture-less space objects by contour points matching,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2342–2355, 2018.
- [103] B. Wang, F. Zhong, and X. Qin, “Robust edge-based 3d object tracking with direction-based pose validation,” *Multimedia Tools and Applications*, vol. 78, pp. 12 307–12 331, 2018.
- [104] A. Zeng, K.-T. Yu, S. Song, D. Suo, J. Walker, A. Rodriguez, and J. Xiao, “Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge,” 09 2016.
- [105] T. Le, L. Hamilton, and A. Torralba, “Benchmarking convolutional neural networks for object segmentation and pose estimation,” 10 2017, pp. 1–10.
- [106] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, “Learning 6d object pose estimation using 3d object coordinates,” vol. 8690, 09 2014, pp. 536–551.
- [107] A. Kendall, M. Grimes, and R. Cipolla, “Convolutional networks for real-time 6-dof camera relocalization,” 05 2015.
- [108] L. Antoni, J. Arango, M. Mohammadzadeh Babr, and M. Freitag, “Deep learning-based pose estimation of apples for inspection in logistic centers using single-perspective imaging,” *Processes*, 07 2019.
- [109] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, “Bigbird: A large-scale 3d database of object instances,” in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 509–516.
- [110] W. Gao and R. Tedrake, “Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 095–11 104.
- [111] P. Liu, M. Lyu, I. King, and J. Xu, “Selfflow: Self-supervised learning of optical flow,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4571–4580.