2015

Modified System Design and Implementation of an Intelligent Assistive Robotic Manipulator

Nicholas Paperno
University of Central Florida

MODIFIED SYSTEM DESIGN AND IMPLEMENTATION OF AN INTELLIGENT ASSISTIVE ROBOTIC MANIPULATOR

by

NICHOLAS ALEXANDER PAPERNO
B.S. University of Central Florida, 2011

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2015

Major Professor: Aman Behal

ABSTRACT

This thesis presents three improvements to the current UCF MANUS systems. The first improvement modifies the existing fine motion controller into PI controller that has been optimized to prevent the object from leaving the view of the cameras used for visual servoing. This is achieved by adding a weight matrix to the proportional part of the controller that is constrained by an artificial ROI. When the feature points being used are approaching the boundaries of the ROI, the optimized controller weights are calculated using quadratic programming and added to the nominal proportional gain portion of the controller. The second improvement was a compensatory gross motion method designed to ensure that the desired object can be identified. If the object cannot be identified after the initial gross motion, the end-effector will then be moved to one of three different locations around the object until the object is identified or all possible positions are checked. This framework combines the Kanade-Lucase-Tomasi local tracking method with the ferns global detector/tracker to create a method that utilizes the strengths of both systems to overcome their inherent weaknesses. The last improvement is a particle-filter based tracking algorithm that robustifies the visual servoing function of fine motion. This method performs better than the current global detector/tracker that was being implemented by allowing the tracker to successfully track the object in complex environments with non-ideal conditions.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Aman Behal as well as Zhao Wang and Dae-Jin Kim for all their help in completing this thesis. Without them, this would not have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 UCF-MANUS System

The MANUS arm developed by Exact Dynamics is an assistive robotic manipulator designed to help those confined to a wheelchair in their day-to-day activities. The arm is meant to be mounted to the side of a wheelchair and controlled via keypad or joystick. While this initial interface does allow for complete control of the MANUS and provides some pre-defined task functions built in, it is not very intuitive in design and takes a while to learn. The UCF MANUS has taken the initial interface and improved upon it by adding various interface options, along with a general automated function for grabbing and retrieving objects [Kim 2014].

The automated function and interface options are designed to allow those with varying levels of disability perform everyday tasks as much more easily than the default interface. This is done through a motion segmentation method [Kim 2012] that first identifies the location of the desired object for an initial gross motion. Once the gross motion has been performed the UCF MANUS will then identify the object to ensure that it has approached the desired object and then utilizing a fine motion algorithm it will align itself with the object in a suitable grasping position. If set to, the UCF MANUS will then proceed to approach, grab, and retrieve the object in question.

1.2 Problem Motivation

While the current system that is in place for the UCF MANUS works well in most situations, it is susceptible to various problems that can be inherent to the methods that are being used for the different parts that make up the segmented motion. The first issue arises during the gross motion step. The method for gross motion involves a stereo analysis of 3-D points that have been identified as the desired object. The depth of the object away from the robot's center is determined to see if the object is within reach of the manipulator. If the object is within range, the final position of the end-effector will be determined and the MANUS will move to the desired location placing the end-effector approximately 10cm from the object. Full details of the gross motion are given in [Kim 2012]. The issues that arise involve the pose of the object in relation to the end-effector. Since there is no object identification taking place, there is no way to determine beforehand if the desired object is the one being approached or if the face of the object being approached is identifiable, therefore grabbable according to the UCF MANUS' setup.

The next two issues happen during the fine motion stage of motion. The current control scheme designed for the fine motion is a proportional controller that utilizes the visual servoing dynamics that are expressed in [Kim 2012]. If there is a significant pose error in both the translational and rotational positions, the end-effector can lose sight of the object while the alignment with the object is being made. The second issue that arises during fine motion is due to the ability of the ferns global detector [Ozuysal 2007] that is being utilized to successfully track the desired object. Global detectors suffer from a lack of robustness when encountering multiple

identical objects, low frame rate, large initial offset, and motion discontinuity. Template based methods also can suffer from pose/appearances changes.

1.3 Goal

The goal of this thesis is to provide improvements to the current UCF MANUS system that allow it to perform robustly in unknown situations by addressing the issues mentioned in the previous section.

1.4 Organization of Thesis

The remainder of this thesis is organized into five different chapters. Chapter 2 will discuss the theory behind the different aspects of the segmented motion and go into the current ones being used with the UCF MANUS. Chapter 3 will discuss the modifications to the nominal and optimal fine motion control system. Chapter 4 will discuss the compensatory gross motion method implemented to ensure positive object identification or rejection. Chapter 5 will present a particle-filter based visual servoing method that will act as the tracker for the fine motion. Chapter 6 summarizes the improvements to the UCF MANUS system.

1.5 References

Dae-Jin Kim; Zhao Wang; Paperno, N.; Behal, A., "System Design and Implementation of UCF MANUS—An Intelligent Assistive Robotic Manipulator," *Mechatronics, IEEE/ASME Transactions on*, vol.19, no.1, pp.225,237, Feb. 2014 doi: 10.1109/TMECH.2012.2226597

Dae-Jin Kim; Zhao Wang; Behal, A., "Motion Segmentation and Control Design for UCF-MANUS—An Intelligent Assistive Robotic Manipulator," *Mechatronics, IEEE/ASME Transactions on*, vol.17, no.5, pp.936, 948, Oct. 2012 doi: 10.1109/TMECH.2011.2149730

M. Ozuysal, P. Fua, and V. Lepetit, “Fast Keypoint Recognition in Ten Lines of Code,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, pp.1-8, 2007.

CHAPTER 2: BACKROUND AND RELATED WORK

2.1 Introduction

This chapter introduces the various methods that are implemented in the different portions of the motion control system of the UCF MANUS. Each section looks at one aspect of the design and the methods used to implement that particular task. The first part that is discussed is fine motion, followed by gross motion and compensatory gross motion, and ending with object tracking algorithms used for visual servoing.

2.2 Chapter Objectives

- Review of Fine Motion control algorithms

- Review of Compensatory Gross Motion algorithms

- Review of global, local, and particle filter tracking algorithms

2.3 Fine Motion Control Algorithms

The purpose of fine motion is to align the end-effector with the desired object so that a grasping motion can then be performed. To accomplish this, visual servoing controllers are implemented to guide the end-effector to the desired position based on feedback from the images being gathered. This can be performed using an image-based (IBVS), position-based (PBVS), or a hybrid approach which incorporates elements of both the IBVS and PBVS methods.

IBVS is based on the idea of minimizing the error between the desired feature positions and current ones on the image plane. This is done by selecting a set of feature points that are

attributed to the object and giving them a desired position on the image space. The desired feature locations are usually acquired offline using a model of the object in question that will align the robot in an appropriate position to the object for the next task.

PVBS is based on the 3-D recreation of the relationship between the object, camera, and end-effector. The vision system being implemented is used to create a 3-D model that gives a desired position of the camera to the object that is to be reached from the current position. This takes away any use of the image in the control design. The issues that arise when using either IBVS or PBVS can be found in [Chaumette 1998]. While these systems can work well, they do not take into account the full system being used which can lead to, issues especially when significant errors are present. This is where hybrid methods have come into play.

One of these methods is the 2- ½ D visual servoing method in [Malis 1999]. The name comes from the fact that the input to the control system is expressed in both the 3-D Cartesian space of the robot and the 2-D image space given by the camera. The image plane is converted into a normalized plane based on the feature points' z value in the camera coordinate system which allows for the desired camera position from the object to be acquired without utilizing a prior 3-D model. This eliminates any necessity to develop a 3-D model of the scene and allows for the vision system to contribute to the control feedback. This in turn ensures that the object will remain within view of the camera and that the end-effector will arrive at the desired pose.

The visual servoing system that is implemented in the UCF MANUS is based off the 2- ½ D method designed by Malis. The MANUS utilizes the ferns global detector [] to create a template of the desired object that contains the necessary feature points that will then be used to determine the final pose of the end-effector at the end of the fine motion. Homography obtained from feature points is then decomposed to obtain the pose errors that are then fed into a proportional controller. The full details of the fine motion control system of the UCF MANUS can be found in [Kim 2012] and [Wang 2012].

2.4 Compensatory Gross Motion Control Algorithms

The purpose of gross motion is to take a sub-region of the current field of view and then have the end-effector move toward the target area to gather more detailed information than what can be discerned at the current position. Since visual identification is employed by the UCF MANUS, a close enough proximity to an object is required to adequately gather enough feature points to make a positive identification. The details on the gross motion algorithm can be found in []. Since the pose of the object is not taken into account when performing gross motion, not enough feature points may be found to make a positive identification or multiple surfaces of the object may be present that prevent a positive identification due to the spread of feature points across the various surfaces. This has led to the development of compensatory motion.

The first of these motions is a realignment scheme that generates orientation set points that aligns the camera optical axis with the least significant direction of variation of the feature point cloud that is created when attempting to identify an object. This method maintains the set distance from

the object that is obtained through the initial gross motion while re-orientating the camera's x-y plane. The second motion scheme is referred to as "Get Another View". This scheme will reposition the end-effector around the object to see if a more desirable set of feature points can be obtained for object identification. It is this method that will be expanded upon for the compensatory gross motion section of this thesis.

2.5 Visual Tracking Techniques[1]

In a visual tracking application, a target is detected and tracked over time to perform a given task. There are a large number of advanced processing algorithms available to track one or multiple targets in different problems. Generally, the target can be found using a global (or model-based) detector while a small portion of the object can be tracked using a local tracker. Global matching process can accurately locate a pattern which is most similar with a built-in model, however, they do not take into account the spatiotemporal constraints associated with a target. On the other hand, local matching processes can track a small set of image features with efficient usage of resources but it is impossible to determine whether the target (even partially) is correctly found or not. In order to overcome these drawbacks, fusion techniques have emerged in the last decade. Fusion of local and global information has been of interest for many robotics-related researchers. [Ishiguro 1990] proposed an incremental build-up process of global map using omnidirectional stereo analysis in the vicinity of a mobile robot; a Kalman Filter (KF) was used to reduce the effect of uncertainty from noisy image measurements. In [Roumeliotis 1999], an Extended Kalman Filter (EKF) framework was used to estimate the position and orientation

[1] This section is comprised of sections of the manuscript "Particle Filter-Based Robust Visual Servoing for UCF-MANUS – An Intelligent Assistive Robotic Manipulator" by N. Paperno, Z. Wang, D.-J. Kim, and A. Behal which is currently pending submission.

of the mobile robot using local odometry information and global sun position information. Compared with local odometry information, global sun position information is relatively infrequent but effective to compensate the estimation error using EKF framework. Similarly, fusion of Global Positioning System (GPS)-driven Genetic Algorithm (GA)-based global information and Inertial Navigation System (INS)-driven feature-based local information was done by [Fu 2006] using fuzzy logic. [Lee 2008] proposed a fusion of local odometry and global magnetic compass information to control an omnidirectional mobile robot. [Moore 2009] proposed a local frame based robotic navigation to overcome disadvantages of global frame and local body frame representations such as increase in uncertainty, multimodal noise, *etc.* [Se 2005] proposed a vision-based global localization and mapping technique using fusion of local submaps and globally matched map information; the use of distinctive visual Scale-Invariant Feature Transform (SIFT) [Lowe 2004] features and backward correction algorithm were efficiently used to deal with uncertainty. In [Rodriguez- Losada 2006], Rodriguez- Losada proposed a local map fusion technique with novel analysis on Simultaneous Localization And Mapping (SLAM)-EKF framework in consideration of SLAM-EKF inconsistency and shape constraints. Recently, [Persson 2008] noted that rule-based fusion of global aerial imagery and locally generated geographical information using mobile robot was effective to build an improved semantic mapping. In RoboCup applications, Bayesian fusion [Pinheiro 2004] and Monte Carlo (MC) localization [Ferrein 2005] were adopted to estimate the ball position and to build a world model around player robots. The particle filter has become a well-established method that has proven to be more effective than its predecessors such as the EKF. Examples of comparisons between the particle filter and EKF can be found in [Kotecha 2003] and [Aydogmus

2012]. During the last decade, particle filter methods have proved to be an effective and powerful approach for single/multiple target tracking, due to their simplicity and flexible treatment of nonlinearity in the system dynamics and non-normality in the sources of uncertainty. A review of particle filter approach and its applications in various fields can be found in [Cappe 2007]. In [Wang 2009], Wang *et al.* propose a novel tracking method by incorporating the efficiency of the mean-shift algorithm with the multihypothesis characteristics of particle filtering in an adaptive manner. In [Okuma 2004], an offline-boosted detector was used to amend the proposal distribution of the particle filters for multi-target tracking. In [L 2008], a cascade particle filter was designed for target tracking in a low frame rate video, where an integration of conventional tracking and detection was used. In order to solve the dynamic view planning problem, an improved particle filter with the largest effective sampling size was applied to accomplish 3-D tracking task in [Chen 2009]. In [Breitenstein 2009] and [Breitenstein a.t.b.p.], a novel multi-person tracking method was proposed in a particle filter framework, where both detectors and classifiers were used. In [Huang 2011], a particle filter approach was extended with depth estimation of the target for tracking multiple targets with possibility of overlapping. Recently in [Wang 2011], Wang *et al.* proposed an adaptive appearance modeling technique to handle various challenges in the tracking task; a third-order tensor was used to represent the target while the particle filter technique was used in the target state estimation. Similar applications of particle filter in 3-D human body tracking, position and orientation estimation, motion tracking, fuzzy spatial information based tracking can be found in [Peurum 2010], [Won 2010],[Kristan 2010], [Rincon 2011], and [Widynski 2011].

Consider the basic problem of tracking a point on a target object using images captured by an eye-in-hand configured camera as the robot undergoes translational and rotational motion at its end-effector. Given the special requirements for wheelchair mounted robotic arms (WMRAs), namely, that of being lightweight (for longer battery life) and having low center of gravity (for balance), these robots necessarily include extensive transmission and gearing. It is well known that transmission and gearing introduce kinematic uncertainties in the robot which may be modeled as additive noise in the commanded translational and rotational velocities at the endeffector. To formulate the problem, consider the dynamics of a pixel (or feature point) $p = [p_x \quad p_y]^T$ on the target object as captured by the end-effector mounted camera as follows [Huchinson 1997]

$$\dot{p} = J_p V \tag{2-1}$$

where J_p is known as the image Jacobian and defined as follows

$$J_p = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{p_x}{Z} & p_x p_y & -(1+p_x^2) & p_y \\ 0 & -\frac{1}{Z} & \frac{p_y}{Z} & 1+p_y^2 & -p_x p_y & -p_x \end{bmatrix} \tag{2-2}$$

In the above equation, Z denotes the Euclidean distance along the z-direction of the camera frame, while $V = [v_c \quad \omega_c]^T$ denotes the composite camera velocity vector comprising of its translational velocity v_c and rotational velocity ω_c. It is clear to see from the structure of (2-1) and (2-2) that any *additive uncertainty* in the composite velocity vector V will manifest itself *nonlinearly* in the pixel dynamics. Thus, the nonlinear image dynamics and the non-additive (and non- Gaussian as we will show in the sequel) nature of the "process noise" make this system not

amenable to Kalman-filter type of schemes – therefore, we contend that any estimation problem is best dealt with inside a particle filter (PF) type of framework.

The UCF-MANUS is capable of utilizing computer vision (among other sensing modalities) for target tracking and manipulation in unstructured environments. Tests in the Assistive Robotics laboratory at UCF using the UCF-MANUS setup have shown that typically utilized global detection and local tracking based algorithms, when used exclusively, fall prey to typical problems such as multiple identical targets, lack of robustness when dealing with a large initial offset, motion discontinuity, *etc.* Specifically, global detectors are robust to low framerate video, motion discontinuity *etc.* while local trackers can easily handle multiplicity of the target in the same frame. However, their weaknesses and strengths are complementary to one another. This has motivated us to design and implement a fusion-based target-template matching algorithm in order to obtain robust and sustained target tracking under a variety of scenarios. By taking advantage of the aforementioned redeeming qualities of the global and local methods, the algorithm systematically prescribes empirically-validated choices for sub-image and/or feature points to consider at each step. As explained above, the system nonlinearity and non-additive nature of the process noise hinder us from getting a closed form solution; therefore, a probabilistic PF-based framework is developed in this paper; specifically, we propose a systematic way to improve the performance of a global matching function by augmenting it with a local matching function. The work presented here is novel as this paper does not simply 'fuse' global and local information in the traditional sense of the word; instead, a synergistic concatenation of the two sources of information is proposed in a probabilistic setup implemented via particle filters to find a better method of tracking objects.

2.6 Discussion

This chapter as looked at the various techniques used in the different components of the motion control used by the UCF MANUS. Typical schemes for fine motion involve using an image-based or position-based visual servoing design. These methods generally produce satisfactory results, but falter when presented with large pose errors and only take one aspect of the system into account. To overcome these deficiencies, hybrid methods have been implored to create more robust systems. The UCF MANUS uses one of these methods as the basis for its fine motion control design.

Gross motion is implemented to maneuver the end-effector to an appropriate distance to the object so that identification can occur. This does not always lead to an optimal relationship between the object and end-effector that allows for a positive identification to be made. To correct this, compensatory motion schemes are introduced to yield a successful position from which the object can be identified from.

Local and global tracking algorithms commonly used in visual servoing lack any robustness when encountering complex situations. Several algorithms have been developed to fuse the methods together to use their individual strengths to overcome their inherent weaknesses. Particle filters have been recent developments in that provide a suitable framework for such a fusion to take place.

2.7 References

Dae-Jin Kim; Zhao Wang; Behal, A., "Motion Segmentation and Control Design for UCF-MANUS—An Intelligent Assistive Robotic Manipulator," *Mechatronics, IEEE/ASME Transactions on*, vol.17, no.5, pp.936,948, Oct. 2012 doi: 10.1109/TMECH.2011.2149730

Zhao Wang; Dae-Jin Kim; Behal, A., "Design of Stable Visual Servoing Under Sensor and Actuator Constraints via a Lyapunov-Based Approach," *Control Systems Technology, IEEE Transactions on* , vol.20, no.6, pp.1575,1582, Nov. 2012 doi: 10.1109/TCST.2011.2168958

E. Malis, F. Chaumette, and S. Boudet, "2 ½ D visual servoing," *IEEE Trans. Robot. Autom.,* vol. 15, No. 2, pp. 238-250, Apr. 1999

F. Chaumette, "Potential problems of stability and convergence in imagebased and position-based visual servoing," in *The Confluence of Vision and Control*, vol. 237, D. Kriegman, G . Hager, and A. S. Morse, Eds. New York: Springer, 1998, pp. 66–78.

H. Ishiguro, M. Yamamoto, and S. Tsuji, "Omni-directional Stereo for Making Global Map," in *Proc. IEEE International Conference on Computer Vision*, Osaka, Japan, pp. 540-547, 1990.

S. I. Roumeliotis and G. A. Bekey, "An Extended Kalman Filter for frequent local and infrequent global sensor data fusion," in *Proc. SPIE Conference on Sensor Fusion and Decentralized Control in Autonomous Robotic Systems*, Pittsburgh, PA, pp. 11-22, 1999.

Y. Fu, H. Xu, H. Li, S. Wang, and H. Xu, "A Navigation Strategy based on Global Geographical Planning and Local Feature Positioning for Mobile Robot in Large Unknown Environment," in *Proc. International Symposium on Systems and Control in Aerospace and Astronautics*, Harbin, China, pp.1189-1193, 2006.

J.-H. Lee and S. Jung, "Global Position Tracking Control of an Omnidirectional Mobile Robot Using Fusion of a Magnetic Compass and Encoders," in *Proc. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Seoul, Korea, pp. 246-251, 2008.

D. C. Moore, A. S. Huang, M. Walter, E. Olson, L. Fletcher, J. Leonard, and S. Teller, "Simultaneous Local and Global State Estimation for Robotic Navigation," in *Proc. IEEE International Conference on Robotics and Automation*, pp. 3794-3799, Kobe, Japan, 2009.

S. Se, D. G. Lowe, and J. J. Little, "Vision-Based Global Localization and Mapping for Mobile Robots," *IEEE Trans. on Robotics,* vol. 21**,** no. 3, pp. 364-375, 2005.

D. Lowe, “Distinctive Image Features from Scale-invariant Keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, 2004, pp. 91-110.

D. Rodriguez-Losada, F. Matia, A. Jimenez, and R. Galan, “Local Map Fusion for Real-time Indoor Simultaneous Localization and Mapping,” *Journal of Field Robotics,* vol. 23, no. 5, pp. 291-309, 2006.

M. Persson, T. Duckett, and A.J.Lilienthal, “Fusion of aerial images and sensor data from a ground vehicle for improved semantic mapping,” *Robotics and Autonomous Systems,* vol. 56, pp. 483-492, 2008.

P. Pinheiro and P. Lima, “Bayesian Sensor Fusion for Cooperative Object Localization and World Modeling,” in *Proc. the 8th Conference on Intelligent Autonomous Systems*, Amsterdam, Netherlands, 2004.

A. Ferrein, L. Hermanns, and G. Lakemeyer, “Comparing Sensor Fusion Techniques for Ball Position Estimation,” *RoboCup 2005, Lecture Notes in Computer Science*, pp. 154-165, 2005.

Kotecha, Jayesh H.; Djuric, P.M., "Gaussian particle filtering," *Signal Processing, IEEE Transactions on* , vol.51, no.10, pp.2592,2601, Oct. 2003

Aydogmus, O.; Talu, M.F., "Comparison of Extended-Kalman- and Particle-Filter-Based Sensorless Speed Control," *Instrumentation and Measurement, IEEE Transactions on* , vol.61, no.2, pp.402,410, Feb. 2012

O. Cappe, S. J. Godsill, E. Moulines,“An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo,” *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899-924, 2007.

J. Wang and Y. Yagi, “Adaptive Mean-Shift Tracking With Auxiliary Particles,” *IEEE Trans. on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 39, no. 6, pp. 1578-1589, 2009.

K. Okuma, A. Taleghani, D. Freitas, J. J. Little, and D. G. Lowe, “A Boosted Particle Filter: Multitarget Detection and Tracking,” in *Proc. European Conf. Computer Vision*, 2004.

Y. L, H. Ai, T. Yamashita, S. Lao, and M. Kawade, “Tracking in Low Frame Rate Video: A Cascade Particle Filter with Discriminative Observers of Different Life Spans,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1728-1740, 2008.

H. Chen and Y. Li, “Dynamic View Planning by Effective Particles for Three-Dimensional Tracking,” *IEEE Trans. on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 39, no. 1, pp. 242-253, 2009.

M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Gool, "Robust Tracking-by-Detection using a Detector Confidence Particle Filter," In *Proc. IEEE International Conference on Computer Vision*, pp. 1515-1522, Kyoto, Japan, 2009.

M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Gool, "Online Multi-Person Tracking-by-Detection from a Single, Uncalibrated Camera," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, accepted, to be publised.

C. Huang and L. Fu, "Multitarget Visual Tracking Based Effective Surveillance With Cooperation of Multiple Active Cameras," *IEEE Trans. on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 41, no. 1, pp. 234-247, 2011.

Q. Wang, F. Chen, and W. Xu, "Tracking by Third-Order Tensor Representation," *IEEE Trans. on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 41, no. 2, pp. 385-396, 2011.

P. Peurum, S. Venkatesh, and G. West, "A study on smoothing for particle-filtered 3-D human body tracking," *Int. J. Comput. Vision,* vol. 87, nos. 1–2, pp. 53–74, 2010.

S. P. Won, W. W. Melek, and F. Golnaraghi, "A Kalman/Particle Filter- Based Position and Orientation Estiamtion Method Using a Position Sensor/Inertial Measurement Unit Hybrid System," *IEEE Trans. Industrial Electronics*, vol. 57, no. 5, pp. 1787-1798, 2010.

M. Kristan, S. Kovacic, A. Leonardis, and J. Pers, "A Two-Stage Dynamic Model for Visual Tracking," *IEEE Trans. on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 40, no. 6, pp. 1505-1520, 2010.

J. M. del Rincon, D. Makris, C. O. Urunuela, and J.-C. Nebel , "Tracking Human Position and Lower Body Parts Using Kalman and Particle Filters Constrained by Human Biomechanics," *IEEE Trans. on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 41, no. 1, pp. 26-37, 2011.

N. Widynski, S. Dubuisson, I. Bloch, "Integration of Fuzzy Spatial Information in Tracking Based on Particle Filtering," *IEEE Trans. On Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 41, no. 3, pp. 635-649, 2011.

S. Hutchinson, G.Hager, and P. Corke, "A Tutorial on Visual Servo Control," *IEEE Trans. Robot. Automat.,* vol. 13, pp. 582-595, Aug. 1997.

CHAPTER 3: FINE MOTION OPTIMIZATION

3.1 Introduction

The purpose of fine motion is to precisely align the gripper with the desired object so that the robotic manipulator can grab the object. After gross motion has been performed and the desired object identified, the end-effector will then need to be aligned in such a way that the object can be grabbed without any problem. This is typically done using a visual servoing system that utilizes an object tracker of some kind. The existing nominal controller being implemented by the UCF MANUS is a proportion controller. For most cases, this controller will perform relatively well when aligning the end-effector to the object. Issues begin to arise when there are significant pose errors between the desired position for alignment and the current orientation with the object. To help overcome these difficulties, an optimized controller was developed to ensure that these issues did not hinder the robots performance.

3.2 Chapter Objectives

-Summarize control design

- Go over initial implementation of method

- State revisions to method needed for successful implementation using PI controller

3.3 Lyapunov-Based Optimization

3.3.1 Lyapunov-Based Control Design

The controller that will be discussed can be found in [Wang, 2012]. What will be presented here is a brief summary of the theory that comprises the control design. The UCF MANUS is implemented using a proportional controller in the form of

$$\tau_c = L_\tau^i K_\tau e \qquad (3\text{-}3)$$

where L_τ^i contains the pixel and camera coordinate information (for more information on the technical development and definition, see [Wang, 2012] and [Malis, 1999]), K_τ is the gain matrix, and $e \triangleq [e_v^T \; e_\omega^T]^T$ is the error signals. This controller works for the majority of cases without the need of any assistance, but is not optimal when dealing with large rotational errors. To help with this, the controller was modified to

$$\tau_c = L_\tau^i K_\tau W_c e \qquad (3\text{-}4)$$

where W_c is a weight matrix designed to optimize the controller to optimize the output. This ensures that the object will remain within view of the camera so that the necessary data can be obtained to perform the Fine Motion required of the robot to align the end-effector with the object. A detailed explanation of the theory behind the development of the optimization and the corresponding stability analysis can be found in [Wang, 2012].

3.3.2 Implementation of Optimized Controller

For the initial implementation of this controller, the gain matrix will be set as follows: $K_\tau = diag(25,25,25,0.12,0.12,0.15)$. These are the normal gain values used in the controller if there is no need to optimize. The gains for the optimization portion will be set as follows: $K_\tau =$

$diag(24,25,24,0.06,0.06,0.06)$. The gains for the optimization have been lowered as a safe guard in the off chance that the quadratic programming function deems the current solution as infeasible. This in turn slows down the movement speed of the robot so when a feasible solution is found it can be implemented. While for the initial paper the quadratic programming was being handled by Matlab using the "quadprog" function, for the final implementation here the quadratic programming will be taken care of using the CGAL library function "quad_comp". While not as fast as the Matlab version, it does allow for all computation to take place within the same program as opposed to running a separate Matlab script that then needs to be communicated with.

When the Fine Motion control begins, the current points are bounded within an artificial region of interest within the camera frame as seen in Fig. 3-1. This region of interest is what is used when determining whether or not the object is heading out of view of the camera. The farthest points in each direction are used monitored to determine the objects position in relation to the artificial ROI. If one of these points approaches or exits the ROI, then the optimization controller is implemented in place of the nominal one. As noted earlier, there is a possibility of the quadratic programming stating that the solution to the particular circumstances is infeasible. If this happens the weights are set in such a way that the robot will perform a backward motion to distance itself from the object as a way to keep it within the boundary. As stated earlier, the gains used for the optimal controller are slightly different from the nominal controller to help resolve these situations. Given that the time it takes for the quadratic programming to compute a solution

is greater than the time it takes for it to conclude that the solution is infeasible, it was necessary to slow the movement speed of the robot down long enough for a feasible solution to be found or have the robot retreat enough so that the object returns to within the bounding ROI.

Figure 3-1: Initialization step for Fine Motion artificial ROI (black) around the object

3.3.3 Conversion of Standard Fine Motion Controller to PI Controller

While the new control design was successful in its initial performances, there were still some issues with the computational time being taken to complete the quadratic programming algorithm. In Matlab, this was not an issue given that the platform is optimized for mathematical computation. Since the programs that control the robot are done in Visual C++, the entirety of the program would need to be done on the same platform. This prevents the need to develop a separate program to handle the Matlab portion that would also need to be active along with the

current six programs required to run our interface and the need to create additional space for the Matlab libraries that would be required for the program. There are already defined functions in CGAL that can handle the necessary calculations. The consequence of this shift is that computational time is slightly increased given the way that the CGAL quadratic programming function works. This in turn prevents the modified control system from acting in time to correct the trajectory of the end-effector so that the object is within its view.

To correct this, the controller was modified into a PI controller so that the gains could be lowered while maintaining a steady state error that was within our necessary threshold for a successful alignment. The new controller is in the form of

$$\tau_c = L_\tau^i K_\tau W_c e + I_\tau \int e \qquad (3\text{-}5)$$

where $I_\tau = diag(0,0,0,1.7e-4,1e-5)$ are the integral gains for the system and the other values have been previously defined. As you will notice, only the rotational errors have integral gains while the translational errors have been zeroed out. This is due to the fact that the translational errors did not need to be altered because they continually converged below the threshold despite the changes to their proportional gain values. The main issue for this modification arose due to the speed at which the rotational errors were corrected. The necessary calculations could not be computed fast enough for them to be implemented. For this implementation, the proportional gain values for the rotational errors have been adjusted to 0.11, 0.11 and 0.13 respectively. This is a slight decrease compared to their initial values, but gives the robot enough time to compute and implement the necessary steps for the optimization process.

The implementation of the PI controller is to ensure that the rotational errors converge below the necessary threshold. As seen in Fig. 3-2 (top), the roll error for the proportional controller converges and settles to -0.12. This does not hold true for the PI controller as the error seen quickly converges toward zero.

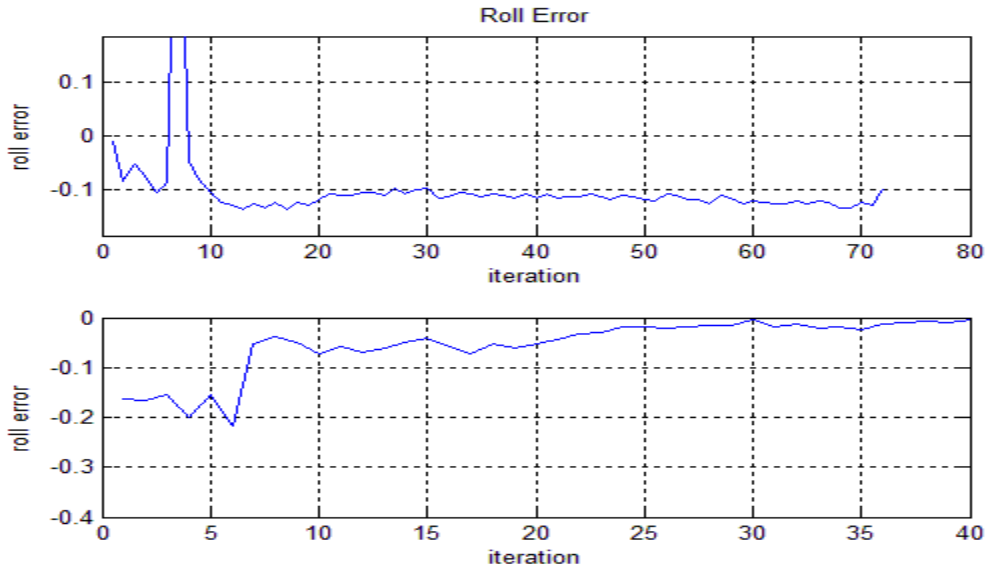

Figure 3-2: Roll error for proportional controller (top) and PI controller (bottom)

3.3.4 Roll Bounding

Another minor issue that appears when doing fine motion is when the end-effector is trying to correct the roll error that is present. If the error is greater than |90|, the motion of the end-effector can lead to the connections between the sensors and cameras being strained or possible broken. This could also lead to the end-effector restricting its movement by being tangled within the wires. To correct this, a different desired position needs to be obtained. Since the objects that are grabbable by the UCF-MANUS are symmetric in shape along their grabbable surface, the new pose error can be found by setting the roll error to $180 - |e_r|$, where e_r is the roll error. To

implement this, the image that is being received to calculate the error is also flipped 180° so that proper error values can be determined. To relate the error values obtained from the new image to the actual position, y, pitch, and yaw errors are reset to their negative and depending on the position of the end-effector, x or z error is also reset to its negative. This ensures that while the end-effector is moving it maintains vision on the identifiable part of the object. In Fig. 3-3 below is an example using a Cheerios box. The initial image is taken as is and the following images are reversed for the remainder of the fine motion.

Figure 3-3: Initial image for fine motion (top) and flipped image (bottom).

3.4 Discussion

This chapter discusses the use of a Lyapunov-based control system designed to optimize the existing UCF MANUS nominal control system. The optimal design constrains the system based on the artificial ROI in order to keep the object within the sight of the camera. The initial

implementation was modified to take into account the processing time and speed of the robotic manipulator. This led to the proportional controller being used being modified to a PI controller to ensure that the steady-state error converges to below the necessary threshold for a completed task.

3.5 References

Zhao Wang; Dae-Jin Kim; Behal, A., "Design of Stable Visual Servoing Under Sensor and Actuator Constraints via a Lyapunov-Based Approach," *Control Systems Technology, IEEE Transactions on* , vol.20, no.6, pp.1575,1582, Nov. 2012 doi: 10.1109/TCST.2011.2168958

E. Malis, F. Chaumette, and S. Boudet, “2 ½ D visual servoing,” *IEEE Trans. Robot. Autom.,* vol. 15, No. 2, pp. 238-250, Apr. 1999

CHAPTER 4: COMPENSATORY GROSS MOTION

4.1 Introduction

The purpose of gross motion is to line up the end-effector with an object so that enough feature points can be gathered to make a positive identification of the desired object. This is not always the case. Identification algorithms perform better when there is very little error in pose between the displayed object and the templates that are being used. The face of the object that is being observed could be one that is not currently in the database or the object cannot be grasp from that particular side. The object could also be orient in such a way that the feature points being used are spread across more than one face of the object, leading to no positive identifications being found. To ensure that the desired object is identified, compensatory gross motion needs to be applied to handle situations that arise that lead to no object identification.

4.2 Chapter Objectives

-Lay out method to compensate gross motion

-Discussion on "Get Another View" method and improvements

4.3 "Get Another View" Approach

The "Get Another View Approach" mentioned in [Kim, 2012] was developed to allow the robot to find a more advantageous view of the object that it can identify. This is to be used if no positive identification of the object can be found after the initial approach. If the object cannot be identified, the end-effector will be moved to a point that lies on a circle with radius r and center c that is $\Delta\theta$ degrees from its current position. For this approach, r = d where d is the distance from the object to the end-effector and $c = o_c$ where o_c is the assumed center of the object. An

illustration of this can be seen in Fig. 4-1 below. This method works relatively well for the first instance of no identification, but is not particularly suitable if there is need for the robot to find additional views to try and identify the chosen object. If there are not enough points for a well-defined point cloud, then the two significant directions that would be used to define the plane of the circle that the end-effector would move upon cannot be formed. The initially proposed method also does not take into account that the end-effector may be moving to any previous positions that have been used for the particular instance of object identification.

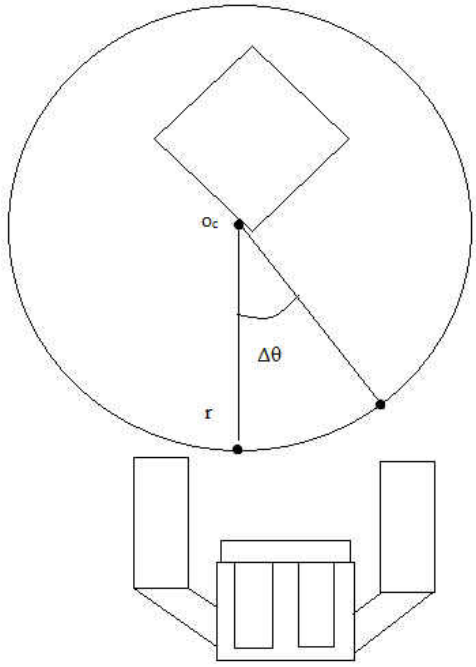

Figure 4-1: Position of end-effector relative to object and constructed circle for determining next position.

The implementation of this method has been modified to take into account these issues. First, the robot will only take into account the front facing and top side. This will prevent the robot from trying to position itself outside of its reach or in a precarious position. This also limits the number of areas that have to be searched to three making the other modifications simpler. The second modification implements a default move in the instance not enough points remain to form a reliable point cloud. Since we know the offset distance that the robot will place the end-effector from the object with its initial gross motion, we can construct the circle that the end-effector can move about. From this we can estimate the final position based on the $\Delta\theta$ that we are using. For our purposes, the circle will lie parallel to the z-plane, placing the end position above the current one. This done due to the fact that all objects used by the UCF MANUS are grabbable from their top surfaces. If there is no reliable point cloud found at the new location the process is terminated, given the assumption that there is no identifiable object in view. This remains true for any given point in the process.

The last modification prevents the robot from checking the same area twice. This is done by logging the previous positions. Since there are only areas being considered, this is a very simple process. To be efficient in checking the necessary areas, the end-effector will be moved to the nearest orthogonal plane that has not been logged as a checked position. A flow chart of the entire method can be seen in Fig. 4-2.

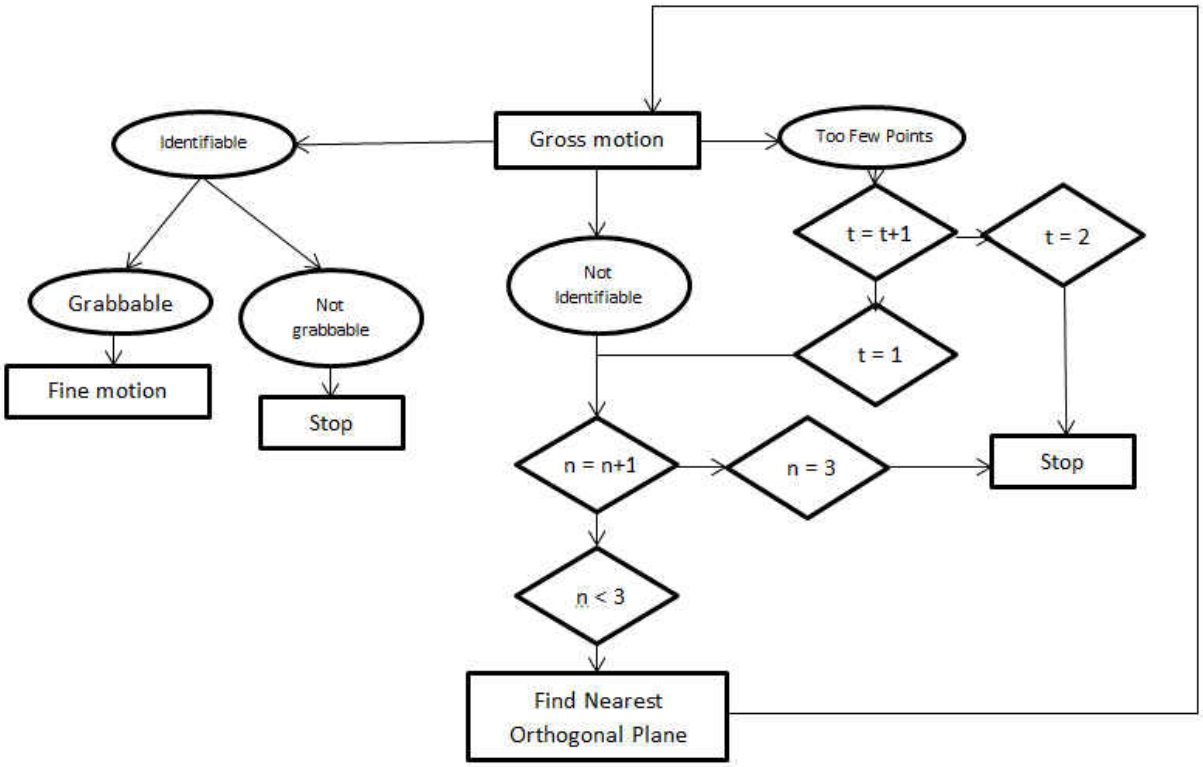

Figure 4-2: Flow chart of Compensatory Gross Motion Process. Variables t and n are counters that keep track of how many times each process has been performed. Once those actions have been performed a certain amount of times, it is assumed that there is no identifiable object being viewed.

A cereal box will be used as an example. In Fig. 4-3 (left) below, the end-effector has performed its initial gross motion and is unable to identify the object due to its pose. The object has two of its sides within view of the camera, one of which is not identifiable, but has enough texture on it to attract the attention of the global detector. There are three positions that the end-effector can now move to. The desirable position is located at position C. To ensure that the positions are searched in an optimal fashion, the end-effector will move to either position C or D. In this instance, the end-effector moved to position C and was able to get enough feature points to make a positive identification.

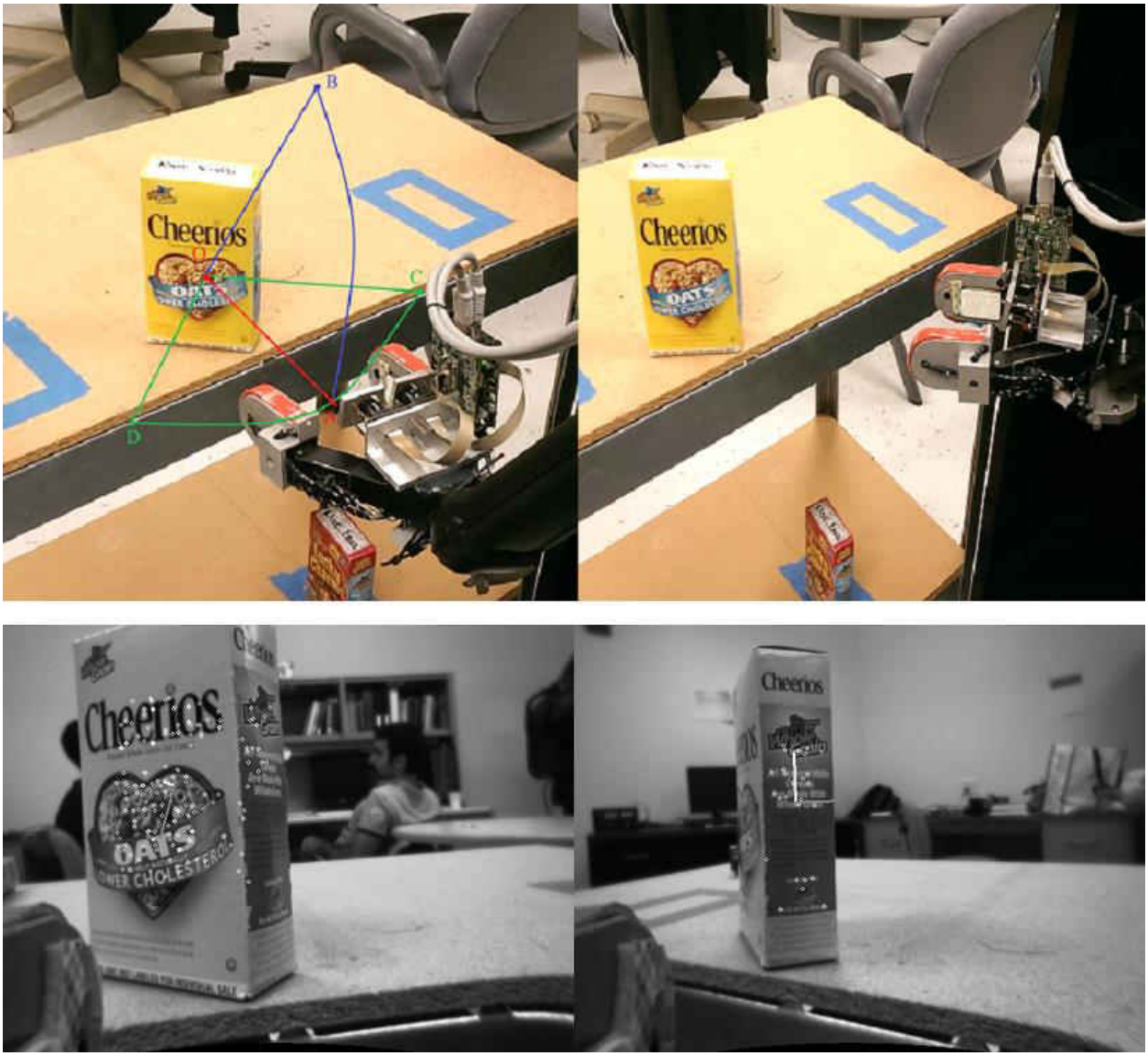

Figure 4-3: "Get Another View" approach tested on Cheerios box. The top pictures are of the position of the end-effector in relation to the object and the bottom pictures are of the feature points found by Ferns for identification.

4.4 Discussion

This chapter presents a method that will be used to compensate any errors in the initial gross motion. The method presented expands upon the idea mentioned in [Kim, 2012] on altering the

position of the end-effect relative to the object in order to obtain a more satisfactory view. The method described above will search three alternative locations if necessary to find a position in which it can identify the desired object. Scenarios in which there are too few points to make identification have also been taken into account.

4.5 References

Dae-Jin Kim; Zhao Wang; Behal, A., "Motion Segmentation and Control Design for UCF-MANUS—An Intelligent Assistive Robotic Manipulator," *Mechatronics, IEEE/ASME Transactions on*, vol.17, no.5, pp.936,948, Oct. 2012 doi: 10.1109/TMECH.2011.2149730

CHAPTER 5: PARTICLE FILTER-BASED VISUAL SERVOING[2]

5.1 Introduction

Visual servoing is a necessary application for creating a reliable automated system that can operate in a complex environment. For our application it allows the MANUS to find and retrieve desired objects for its user. The most common methods for this usually involve a global detector to identify the object and a local tracker to track a portion of the object while the robot is aligning itself. While these processes are easy to implement and can work rather effectively, they do possess some drawbacks that hinder them when it comes to operating in complex environments or with less than ideal conditions such as low frame rate. Since global detectors are model-based, they can be confused relatively easily by pose changes or a variety of other spatiotemporal constraints. Also if any similar objects are present in the frame, the detector can then identify the other object and instead proceed to align itself and grab the similar object as opposed to the desired one. To solve these problems, a novel solution using a particle filter that utilizes both a local and global tracking method has been developed and implemented with the UCF MANUS system.

5.2 Chapter Objectives

- Lay out Particle Filter framework.

[2] This section is comprised of sections of the manuscript "Particle Filter-Based Robust Visual Servoing for UCF-MANUS – An Intelligent Assistive Robotic Manipulator" by N. Paperno, Z. Wang, D.-J. Kim, and A. Behal which is currently pending submission.

- Demonstrate ability of Particle Filter as tracking method on its own.

- Demonstrate ability of Particle Filter as visual servoing method in loop.

- Discussion on Particle Filter algorithm as an alternative to the current method.

5.3 Particle Filter Framework

5.3.1. Preliminaries

Let us assume that a frame of image has been grabbed from the camera at time t. From this image, features points denoted as g_t are extracted by a global detector. Using a known template model with a feature point set G_d, one can find another set of feature points $g_d \subset G_d$ which shows a one-to-one correspondence with g_t. We can also define a global detector Region of Interest (ROI) y_t which is a minimal polygon enclosing all feature points in g_t. We will use this global detector ROI y_t to indicate the identified target on the frame grabbed at time t. Furthermore, the local tracker can also track a set of feature points L_t at time t matching with the feature point set L_{t-1} in the frame grabbed at time $t-1$. Note that the feature points in g_t and L_t are independent and obtained through global detector and local tracker, respectively. Next, we define the i^{th} particle as follows

$$\pi_t^i = \{x_t^i, w_t^i\} \qquad i = 1, \dots, N_p \tag{5-1}$$

where x_t^i denotes the ROI encoded in the i^{th} particle, w_t^i is the weight associated with the particle, while N_p denotes the number of particles. Then, given the feature point sets $l_t^i \subset$

$L_t\ \forall i\ =\ 1, \ldots, Np$, we define a particle filter ROI x_t^i on the frame grabbed at time t as a minimally enclosing polygon of the residual image features in the set l_t^i as $x_t^i = r\left(l_t^i\right)$. This polygon ROI x_t^i for l_t^i can be explicitly given as

$$x_t^i = r\left(l_t^i\right) = (\ p_{t,1}, p_{t,2}, \ldots, p_{t,N_x}) \tag{5-2}$$

where $p_{t,i} \in l_t^i$ and N_x denotes the number of points on the boundary of this polygon. Note that x_t^i can be considered as a polygon representation of the associated local feature set l_t^i in the image space. These particles are used to obtain particle filter output ROI $\widehat{y_t^p}$ based on particle ROIs x_t^i and their associated weight w_t^i. The global detector ROI y_t is used as an observation at time t. Under the particle filter setup, our goal is to robustly identify the target from the currently grabbed image at time t and the observations $y_{1:t}$ up to time t. A posterior pdf can be described by a set of N_p random samples (*i.e.,* particles) as follows [Arulampalam 2001]

$$p(x_{0:t}|y_{1:t}) \approx \sum_{i=1}^{N_p} w_t^i \cdot \delta(x_{0:t} - x_{0:t}^i) \tag{5-3}$$

where x_t^i, w_t^i, and y_t have been previously defined. Fig. 5-1 shows an example of particles ROIs x_t^i and output ROI y_t^p for a cereal box.

Remark 1: We note here that the enclosing polygon and not the features themselves are utilized as particles because in general, global and local tracking algorithms may not necessarily obtain similar or even overlapping sets of features on an object.

Figure 5-1: Particles ROIs (left) and output ROI (right) in frame 42 of a cereal-box video sequence recorded in the UCF Assistive Robotics Laboratory.

5.3.2 Particle Filter Framework

We begin by defining a novel measure of the degree and quality of overlap between two particle ROIs **a** and **b** in the form of an overlap ratio r(**a**,**b**) as follows

$$r(\boldsymbol{a},\boldsymbol{b}) \triangleq \frac{A_{overlap}}{A_a} \cdot \frac{A_{overlap}}{A_b} \cdot exp(-kf(|d|)) \tag{5-4}$$

Here, A_a and A_b denote, respectively, the area enclosed by the ROIs **a** and **b** while $A_{overlap}$ denotes the area of overlap between the two ROIs. Furthermore, the function f (·) denotes a monotonically increasing function of its argument, d represents the distance between the geometric centers of the two ROIs while k is an empirical rate constant. To motivate the selection of this overlap ratio, we note that the first two factors in the definition of $r(\boldsymbol{a},\boldsymbol{b})$ capture the degree of overlap while the last factor encodes the quality of the overlap, *e.g.,* between two particles, say a_1 and a_2 with identical ROI sizes and similar overlap area with **b**, we deem the one located more centrally with respect to b as having a better overlap than the other whose location is more peripheral. Note that by definition, $r(\boldsymbol{a},\boldsymbol{b}) \in (0,1]$. In this paper, the

empirical determination of the pdfs was obtained using the following choices: $k = 0.02$ and $f(|d|) = |d|$. Details of calculations for the underlying polygons considered in the definition of $r(\boldsymbol{a}, \boldsymbol{b})$ can be found in the Appendix. As for the measurement process which enables the pdf update in the PF framework, one can choose from a bevy of available global template-based detectors; while SIFT [Lowe 2004] and SURF [Bay 2008] are commonly used detectors, in this paper, we turn toward a real-time implementable global detection-based tracker known as *ferns* [Ozuysal 2007].

1) *Initialization Step:* Given an initial specification of a target ROI R1 in the first frame as illustrated in the left part of Fig. 5-2, one can randomly generate particle ROIs x_l^i around the specified region as illustrated in the right part of Fig. 5-2. This is done by creating a set of offset values that have a uniform distribution and applying them to the borders of the initially specified ROI. Then, the local tracker will select good feature points for future tracking and each randomly generated particle ROI x_l^i will be associated with all feature points located inside it. We note that the initially specified target ROI may not exactly match with the ideal target ROI. By choosing a large number of particles randomly around the initial ROI, we can easily increase the probability of including the ideal ROI into the union of all particle ROIs. One can adjust the parameter of the uniform distribution used in the particle initialization step according to the quality of the initial selection, *i.e.*, the generated particles could be spread in a large region if we expect the initial selection to be inaccurate.

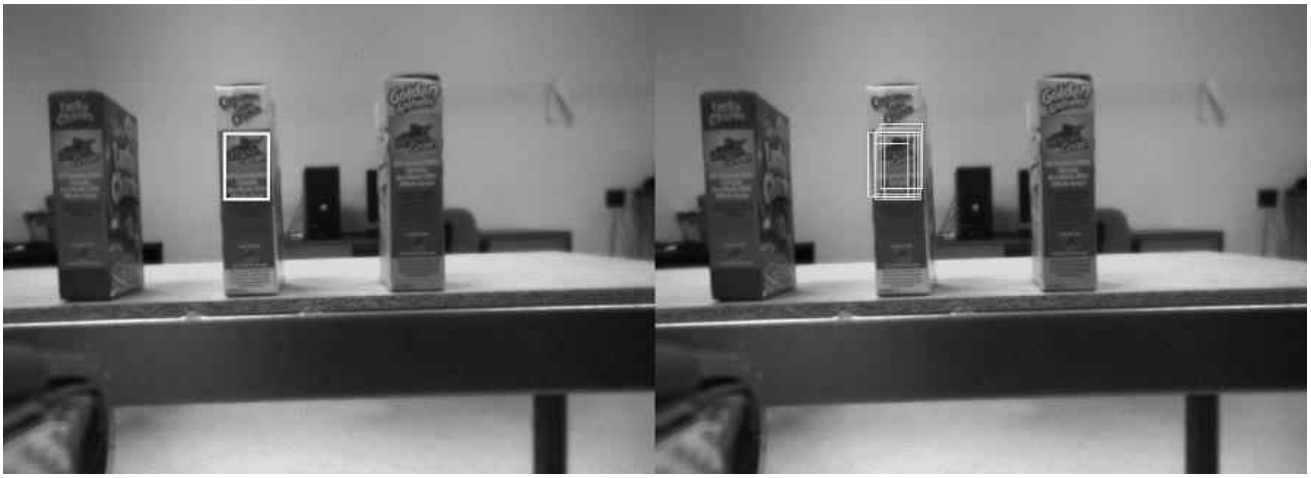

Figure 5-2: Initialization of particles (right) using initial ROI (right).

2) Prediction Step: As is well known, the standard PF process entails two broad steps, *viz*, a prediction based on the prior (pre-measurement) pdf and an update based on the posterior (post-measurement) pdf. Specifically, assuming that we have a posterior pdf $p(x_{t-1}|y_{1:t-1})$ of the state at time $t-1$, the first step is to draw samples from the importance function as follows

$$x_t^i \sim p(x_t|x_{t-1}) \tag{5-5}$$

Normally, the prior is utilized as the importance function from which it is easier to draw samples. However, in the proposed method, this is difficult to implement using system dynamics. From (2-1) and (2-2), one can clearly see the difficulty in implementing the prediction step without explicit knowledge, assumption, or estimation of the depth of the various feature points that constitute an object of interest. Furthermore, knowledge of the nominal camera motion is required. A more practicable model-free (*i.e.*, we do not need to know the camera velocity, camera parameters, target depth, *etc*.) predictor can be implemented by utilizing the seminal Kanade-Lucas-Tomasi (*KLT*) feature tracker [Shi 1994][Birchfield 2006] based on optical flow

which is based on the following relationship holding under small relative motion between frames:

$$p_t \approx p_{t-1} + gd \tag{5-6}$$

where $p_t, p_{t-1} \in \mathbb{R}^2$ denote a feature point in the image at time t and $t + dt$, respectively, $g \in \mathbb{R}^{2\times 2}$ denotes the image gradient, while $d \in \mathbb{R}^2$ denotes the distance vector between the feature at times $t-1$ and t. Here, d is computed as the distance that minimizes the intensity difference between a suitably chosen small window of pixels around the feature point in the images taken at time $t-1$ and t. By thus utilizing KLT to track all image features in set L_{t-1} at frame # $t-1$ into L_t in the image frame captured at time t, we can find the feature point subset $l_t^i \subset L_t$ for x_t^i matching with $l_{t-1}^i \subset L_{t-1}$associated with ith particle ROI x_{t-1}^i. The sample ROI for x_t^i can then be obtained as the minimum polygon including all the feature points in the set l_t^i.

Remark 2: As noted earlier, we use KLT as our model-free predictor. To make a connection with the standard particle filter approach, the approximation inherent in the relation expressed by (5-6) is the source for the process noise. Thus, we do not need to implement an explicit definition of an importance function for our particle filter unlike in the standard approach which requires use of a model and addition of process noise to the particles as drawn from the importance function.

3) Update Step: In this step, the predicted prior pdf from the prediction step is corrected via observations by using a global detector. In order to robustify and speed up the performance of the global detector, the insight here is to apply the global detector in a sub-region based on the union of particle ROIs x_t^i whose corresponding weights are larger than an appropriately certain

threshold – this is in lieu of simply detecting the feature points from the entire image frame as is commonly done. Specifically, we define a region X_t as follows

$$X_t = \cup\, x_t^i, \forall w_t^i > w_{th}, i = 1, \ldots, N_q \tag{5-7}$$

In this predicted sub-region X_t, the global detector is more likely to locate the intended target without showing jumps between identical or similar targets in the same image frame. From a probabilistic perspective, these sub-regions x_t^i which are representative of the pdf $p(x_t|y_{0:t-1})$, can give us a potential region containing the target object with a much higher probability than any other region in the grabbed frame. During the implementation, one may apply the global detector based on a higher w_{th} at first. If the global detector is unable to detect the target, one can lower the threshold value and enlarge the predicted sub-region for the global detector in order to decrease the occurrence of false negatives. By applying the global detector on region X_t of the current frame and comparing with the template frame, one can get the set of feature points g_t (defined previously in Section III-A) which shows one-to-one correspondence with a template feature point set $g_t \subset G_t$. Then, one can find an observed global detector ROI y_t which encloses all feature points in the feature point set g_t. Based on the measurement produced by the global detector inside the aforementioned sub-region, the associated weights of the particles in the prior distribution are updated based on the current observation using

$$\overline{w_t^i} \propto p\left(y_t|x_t^i\right) w_{t-1}^i \tag{5-8}$$

where $p\left(y_t|x_t^i\right)$ is defined as the observer posterior that can be approximated based on empirical observations as follows

$$p\left(y_t|x_t^i\right) = \frac{2}{\sqrt{2\pi\sigma^2}} exp\left(\frac{-\left(1-r\left(y_t,x_t^i\right)\right)^2}{2\sigma^2}\right) \quad (5\text{-}9)$$

where r has been previously defined and $\sigma^2 = 0.04$. We calculated the empirical distribution of $r(\cdot,\cdot)$ by comparing a ground truth ROI to the measured ROIs given by the ferns detector over multiple image captures and then averaging it over multiple objects in our laboratory – this average distribution can be seen in the histogram shown in Fig. 5-3. We note here that the histogram may be further smoothed out by using data from more image captures and objects. While many different approximation functions can be utilized to capture this posterior distribution, we chose a simple 1-sided Gaussian distribution acting as an envelope for the empirical data. As will be seen in subsequent experimental results, even this simple choice leads to excellent tracking performance in a variety of scenarios. Finally, the associated weights of the updated particles computed using (5-9) are normalized using

$$w_t^i = \frac{\overline{w_t^i}}{\sum_{j=1}^{N_p} \overline{w_t^j}} \quad (5\text{-}10)$$

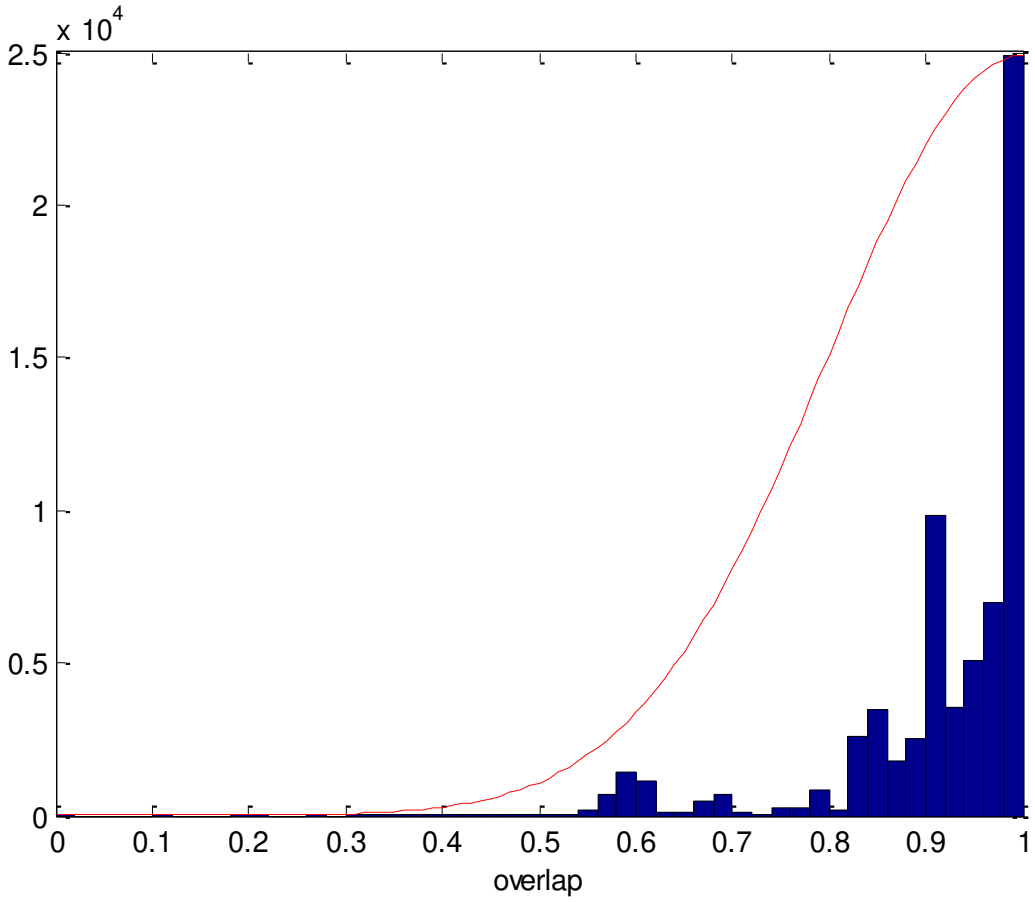

Figure 5-3: Posterior pdf for the particle filter overlaid over the histogram of the overlaps taken from various objects. This histogram was created by taking the overlaps of the ferns results to the ground truth of various objects.

4) Output Step: Finally, the best estimate of the object ROI using the particle-filter based approach can be obtained by utilizing any of a number of different methods [Rekleitis 2004]. The following three methods are commonly employed: 1) utilize the particle with the maximum weight, 2) utilize a weighted sum of all particles, and 3) utilize a constrained weighted sum of all selected particles with weight higher than a user-defined threshold $0 < \varepsilon < 1$

$$
\begin{aligned}
&y1: \quad \widehat{y_t^p} = x_t^{(arg\,max_j w_t^j)} \\
&y2: \quad \widehat{y_t^p} = \sum_{i=1}^{N_p} x_t^i w_t^i \\
&y3: \quad \widehat{y_t^p} = \sum_{i=1}^{N_p} x_t^i w_t^i \,;\; w_t^i \geq \epsilon
\end{aligned}
\tag{5-11}
$$

In the implementation scheme detailed in the following section, of the three output methods defined above, we have utilized the method of weighted sum with threshold. Since the results from the particle-filter based approach will be compared with results from the exclusive use of local or global methods, it is necessary to also define the object ROI estimate for those cases. When the local tracking algorithm is used solely (*i.e.,* outside the particle filter framework), the estimated output ROI $\widehat{y_t^l}$ is defined as the minimum polygon enclosing all local tracker feature points in the current frame tracked from the previous frame. On the other hand, the output of the global detector outside the particle filter framework is defined as $\widehat{y_t^g} = y_t$. Here, the superscripts p, l, and g, respectively, refer to particle-based, local and global approaches.

5) Particle Resampling and Feature Replacement: In this proposed particle filter based synergistic approach, the particle set will be resampled based on the following criterion. In order to overcome the depletion of particle population after a few iterations, particles with insignificant weights need to be replaced (or resampled) according to a resampling policy, *i.e.,* the current set of particles $\pi_t^i = \{x_t^i, w_t^i\}, i = 1, \dots, N_p$ needs to be replaced with a new set of particles $\overline{\pi_t^i} = \{\overline{x_t^i}, \overline{w_t^i}\}, i = 1, \dots, N_p$ such that the ones with small weights will be eliminated (probabilistically) while the ones with higher weights will be duplicated. Note that the weights for the newly

sampled particle $\overline{x_t^i}$ will be assigned as $1/N_p$ uniformly. Two different quantities have been shown to estimate the number of insignificant (*near-zero-weight*) particles, namely, the coefficient of variation cv_t^2 and the effective sample size ESS_t which are defined as follows [Liu 2001]

$$cv_t^2 = \frac{var(w_t^i)}{E^2(w_t^i)} = \frac{1}{N_p}\sum_{i=1}^{N_p}\left(N_p w_p^i - 1\right)^2 \quad (5\text{-}12)$$

$$ESS_t = \frac{N_p}{1+cv_t^2} \quad (5\text{-}13)$$

In this paper, we chose the second quantity as a decision criterion for resampling process. When the effective sample size drops below a certain threshold (usually below a percentage of the number of particles N_p), as follows

$$ESS_t < \eta_1 N_p, \quad (5\text{-}14)$$

then the particle population is resampled according to the weights of the particles as previously stated. In this paper, among different methods of resampling, we have applied the 'Sequential Importance Sampling (SIS) with Resampling' approach [Arulampalam 2001]. Further implementation details can be found in the proceeding section. Due to pattern occlusion or computational failures, local trackers show a tendency to lose features during tracking between two consecutive frames. Since the size of the particle ROI depends on the position of feature points of the local tracker, one may expect the particle ROI to shrink when the associated feature points on the ROI boundary are lost. In order to maintain the size of the particle ROI against unwanted shrinkage due to loss of feature points, we replace lost feature points via regeneration when the number of valid feature points is lower than a threshold. After regenerating local

tracker feature points inside the union set of the updated particles, each particle x_t^i then associates with the newly generated local feature points located inside it.

5.3.3 Overall Algorithm

The overall algorithm proceeds according to the steps given below.

1) Set $t = 1$; Grab a frame $\mathbf{I}_1$; According to the given initial ROI $\mathbf{R}_1$ located near the ideal target ROI, we generate particles $x_l^i, i = 1, \dots, N_p$ randomly around $\mathbf{R}_1$; Generate local features L_1; Associate subset $l_1^i \subset L_1$ with xi1 and set wi1= 1=Np; Hence, the ith particle can be described as follows

$$\pi_t^i = \{x_t^i, w_t^i\}, i = 1, \dots, N_p \quad (5\text{-}15)$$

2) Increase$t = t + 1$; Grab a frame $\mathbf{I}_t$; Compute the position of feature points by using a local tracker which is denoted by the mapping $l_{t-1}^i \rightarrow l_t^i$. Then calculate x_t^i which is the minimal particle ROI enclosing feature point in the set l_t^i.

3) Define sub-region $\overline{y_t}$according to the predicted particle ROI $x_t^i > w_t^i$ as in (5-7).

4) Measurement y_t is obtained by global detector applied in the sub-region$\overline{y_t}$, and then update weight w_{t-1}^i using $p(y_t|x_t^i)$ as in (5-8).

5) Normalize weight according to (5-10), and then calculate the output $\widehat{y_t^p}$using (5-11).

6) If $ESS_t < \eta_1 N_p, \eta_1 \in (0,1)$ or $N_v > \eta_2 N_p$, perform resampling to generate a new particle set $\overline{\pi_t^i}$.

7) Regenerate lost feature points for local tracker; update feature point set l_t^i for each particles.

8) Go to step 2 unless last frame has been reached.

5.4 Experimental Results

5.4.1 Open Loop Experiments

Here, the global detector, local tracker, and proposed particle filter based fusion approach have been tested and compared on the 6-DOF assistive robotic manipulator UCF-MANUS in the Assistive Robotics Laboratory at UCF [Kim 2012]. The input images are grabbed through a Dragonfly 2 firewire camera with 640 x 480 pixel-sized, 8bit image. In this experiment, a *ferns* based detector/tracker [Ozuysal 2007] is adopted as the global tracker while a Kanade-Lucas-Tomasi (KLT) feature tracker [Shi 1994][Birchfield 2006] is used for local tracking for comparison with our proposed particle filter-based fusion approach. The target object is laid down on a table with a pepper-and-salt-like surface and never moved during the motion of the robot. The camera is mounted on the robot end-effector which is moved arbitrarily and a sequence of frames is captured. Based on this setup, three different tracking approaches (*i.e.,* local, global, and particle filter based) are tested to track the target object in the sequence and compared with the ground truth ROI. As previously stated, we chose 'Sequential Importance Sampling (SIS) with Resampling' due to its simple structure and effectiveness in many applications [Arulampalam 2001]. Also, note that the number of particles has been chosen as N_p = 100 in all sets of experiments. The maximum number of feature points for the local tracker is chosen to be between 50 and 200 in the set of experiments shown below. The algorithm is implemented in C++ and tested on a PC with Intel Core(TM) i7 970 CPU and 8GB RAM. Note

that k and $f(\cdot)$ defined in (5-4) have been chosen as follows: $k = 0.02$ and $f(|d|) = |d|$. In order to measure and compare the performance between the particle filter and the local/global methods, we define the instantaneous matching error E_t between the ground truth ideal target ROI $\bar{y}_t$ and estimated ROI $\widehat{y_t^p}$, $\widehat{y_t^l}$, and $\widehat{y_t^g}$ at time t as follows

$$E_t \triangleq 1 - r(\bar{y}_t, \widehat{y_t^x}) \tag{5-16}$$

where $r(\cdot,\cdot)$ has been previously defined in (5-4) while $\widehat{y_t^x}$ stands for the output ROI obtained from the particle filter based, local, and global approaches, respectively, when x is p, l, and g. In Tables 5-1, 5-2, and 5-3, the overall performance of these three methods is compared by using RMSE, which is defined as follows

$$RMSE = \sqrt{\frac{1}{N_t}\sum_{t=1}^{N_t} E_t^2} \tag{5-17}$$

Here N_t denotes the number of frames used to compute the RMSE in each of the experiments.

1) *Target Tracking with Multiple Identical Objects:* Fig. 5-4 shows snapshots of observed global tracker target ROI $\widehat{y_t^g}$, local tracker ROI $\widehat{y_t^l}$, and particle filter output ROI $\widehat{y_t^p}$ using three different approaches in a video containing 119 frames. In the experiment setting, there are three identical cereal boxes in the same frame initially out of which we designate the box in the middle as the target object. Note that we did not show any snapshots from the experiment result before frame #83 since the global detector fails to locate an ROI between frame #1 and frame #82. When one of the three cereal boxes (on the right) starts to move out of field of view, the global tracker can locate an ROI but it is not completely located on our designated target. In the global

matching case, unstable fixation or non-detection of the target object is clearly seen (first row of frame 114 in Fig. 5-4). The experimental results show that the success rate for the global tracker is only 13.5% over the 119 frames, *i.e.,* only sixteen frames out of 119 can be identified correctly. In this experimental setting, we expected the local tracker to have a better performance than the global detector because of smooth and consistent target movement; this is confirmed by comparing the RMSE in Table 5-1 as well as the first and second rows in each of the annotated frames shown in Fig. 5-4. As compared with global and local matching processes, the synergistic chain of global and local matching processes implemented via particle filters (third row of each frame in Fig. 4 can most effectively track the object ROI in a consistent fashion; the particle filter based algorithm successfully detects the target in 115 out of 119 frames (success rate is 96.6%). As can be seen in the last row of Table 5-1, the instantaneous matching error of the particle-filter based method is consistently lower than the local and global tracker. Furthermore, it can be seen in the second to last column of Table 5-1 that the fusion approach exhibits better RMSE performance as compared with the other two approaches. The variance of the local and global trackers is lower than that of the particle filter, but this is due to the consistent failure and poor performances of those methods. This fact holds true for the rest of the experiments given below.

Table 5-1: Mean and Variance of Errors for all used Estimators

	Et in #77	Et in #81	Et in #86	Et in #114	RMSE	variance of Et
Global	1.000	1.000	0.977	0.921	0.8916	0.0242
Local	0.966	0.974	0.889	0.933	0.8288	0.0232
Particle Filter	0.511	0.436	0.468	0.385	0.6864	0.0547

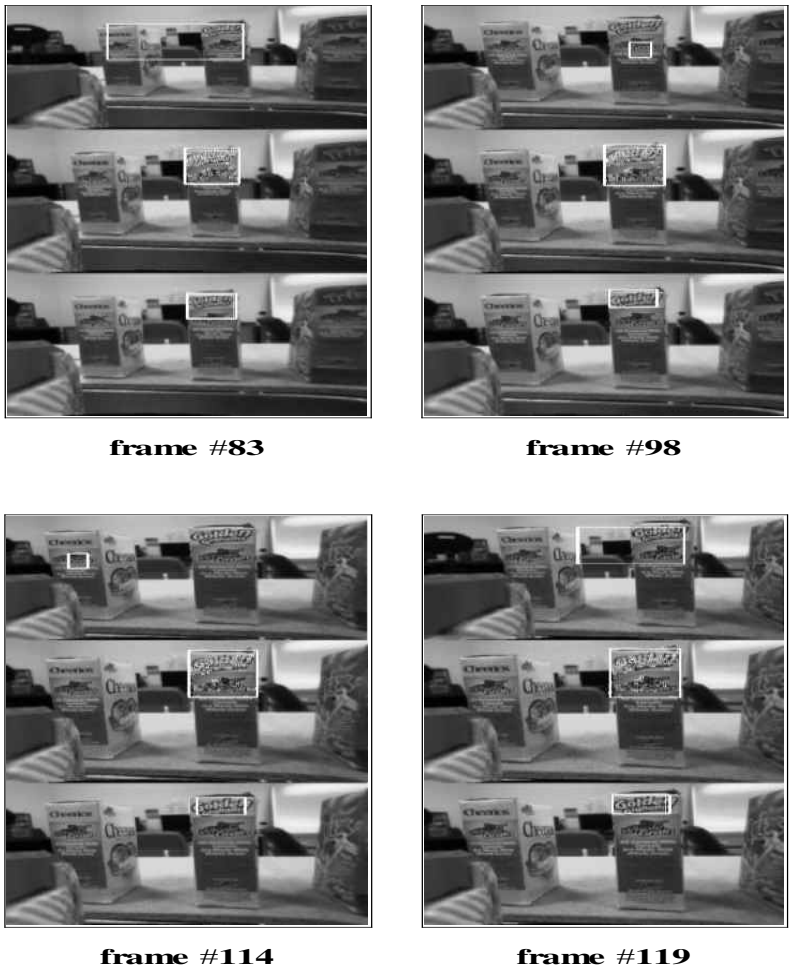

Figure 5-4: Sample results from Global (top), Local (middle), and Particle Filter (bottom) while tracking with multiple identical objects.

2) *Target Tracking with Initial Offset in Low Frame Rate Video:* In this experiment, we intend to track a single target in low frame rate video while the initial user selection of the target ROI contains an offset as can be seen from the left part of Fig. 5-5. The video is sampled at 5Hz from the camera. The average movement of target center across two consecutive frames is around 30 pixels. Since there is only one cereal box (*i.e.,* an unambiguous target) in this sequence of frame, it implies that the global tracker should work properly in this configuration. Fig. 5-6 shows snapshots of estimated object ROI using the three different approaches. It can be clearly seen in row 2 of frame #7 and frame #10 that the local tracking algorithm yields a severely biased tracking result – the identified target ROI covers nearly the entire cereal box, which is nearly five to ten times larger than the ideal target ROI. From the problem configuration, we can surmise that movement discontinuity and initial offset greatly affect the performance of the local tracking algorithm. Due to the lack of update and correction mechanism for the local tracking algorithm, the falsely enlarged target ROI will be restored only when the mismatching feature points are lost. We note that this false enlargement due to feature mismatching occurs and persists even in the presence of the affine consistency check. While the global tracker works well in general, we note that it does fail in frame #32 where the feature points on the drink bottle have been falsely matched and the observed ROI is greatly enlarged covering both the ideal target and a large part of the drink bottle. The success rate for the global detector is 94.5% *i.e.,* only five frames out of 91 cannot be identified correctly. Finally, as can be seen in the last row of all frames shown in Fig. 5-6, the proposed particle filter based fusion approach can successfully detect the target. Over

all frames, the success rate for the proposed algorithm was 99%. From the last row in Table 5-2, it is easy to see that the matching error converges to a small value in a short period of time as compared with the results from the local and global approaches.

Table 5-2: Mean and Variance of Errors for all used Estimators

	Et in #7	Et in #30	Et in #81	Et in #90	RMSE	variance of Et
Global	0.941	0.748	0.805	0.975	0.9724	0.005214
Local	0.966	0.974	0.889	0.933	0.8288	0.0232
Particle Filter	0.492	0.355	0.156	0.108	0.3196	0.03086

Figure 5-5: Initial Offset ROI for Open-loop experiments. Left: selection for experiment in 5.4.1.2. Right: selection for experiment in 5.4.1.3.

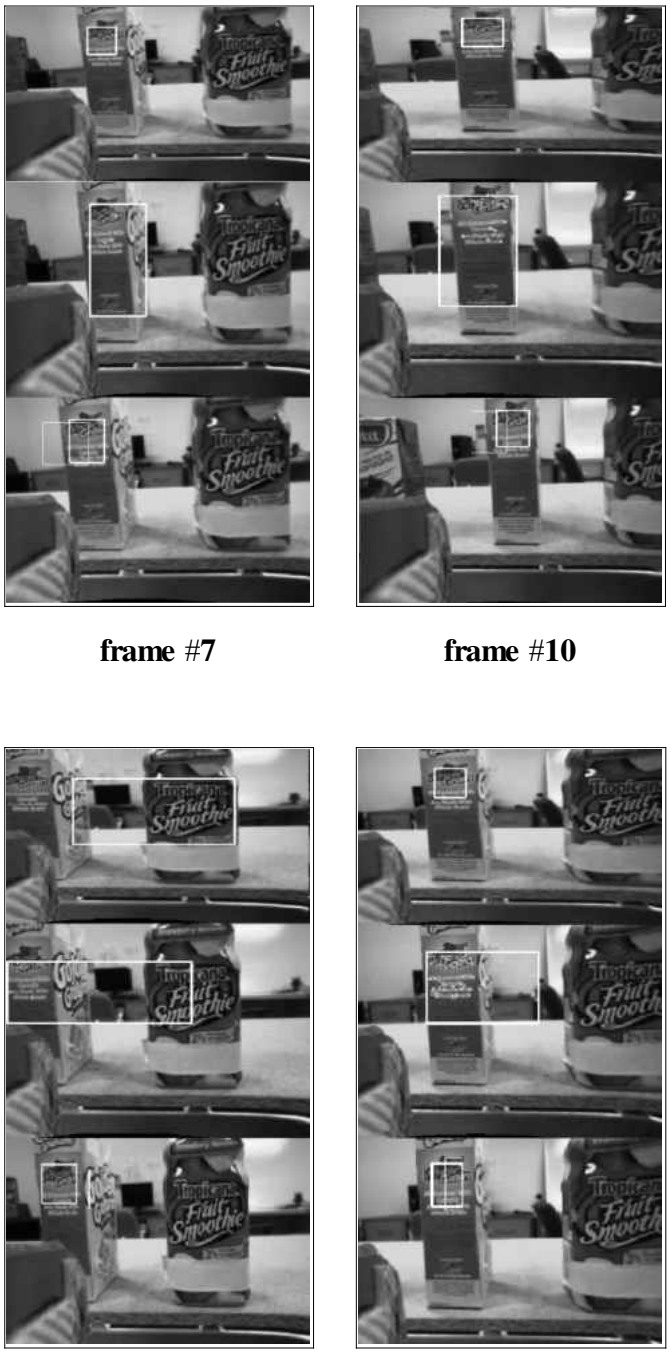

Figure 5-6: Sample results from Global (top), Local (middle), and Particle Filter (bottom) while tracking with large initial offset.

3) *Target Tracking in Complex Environment:* In this experiment, we intend to present the performance of the proposed approach for target tracking in a low frame rate video with multiple identical objects. It is clear to see in this case that this is a more challenging problem than the previous two cases since both the global and local tracker are expected to face stiff hurdles due to the complexity of the environment setting. Specifically, we expect that in the low frame rate video, the performance of the local tracking method will be severely affected due to the lack of motion continuity while the global tracker will be confused when multiple identical objects are presented in one frame at the same time. Note that we also consider a large offset in the user selection of the initial ROI as shown in the right part of Fig. 5-5. The video is sampled 5Hz from the camera and down sampled to 2.5Hz. The average movement of the target center cross two consecutive frames is around 30 pixels. Fig. 5-7 shows the target tracking results for each of the three techniques over four sample frames. Over the 155 frames of the video, the global tracker never finds the desired target (success rate 0.0%), since the dummy cereal box on the left (mimicking the target cereal box in the middle of the frame) is closer to the camera compared with the desired target – the global tracker favors the cereal box on the left of the frame because more details are available due to its location in the image foreground. On the other hand, due to the lack of motion continuity and the initial ROI offset, the local tracking algorithm suffers from feature point mismatching and the target ROI is falsely enlarged. In frame #91 of Fig. 5-7, two objects have been included in the estimated target ROI by the local tracker. Later in frame #119, another object is seen included in the estimated ROI which is nearly 10 times larger than the ground truth target

ROI. As seen in the first two rows of Table 5-3, both the global and local schemes show close to maximal tracking error; obviously, the variance is low because of consistent failure to track the target over the entire length of the video. In contrast with the local and global approaches, the success rate of the proposed particle filter approach is 97.4%. As seen from the last rows of Table 5-3 and each of the frames shown in Fig. 5-7, the particle filter approach outperforms by far the other two cases.

Table 5-3: Mean and Variance of Errors for all used Estimators

	Et in #7	Et in #13	Et in #91	Et in #119	RMSE	variance of Et
Global	1.000	1.000	1.000	1.000	0.9943	0.000506
Local	0.701	0.768	1.000	0.959	0.9497	0.003964
Particle Filter	0.292	0.333	0.647	0.331	0.4183	0.04612

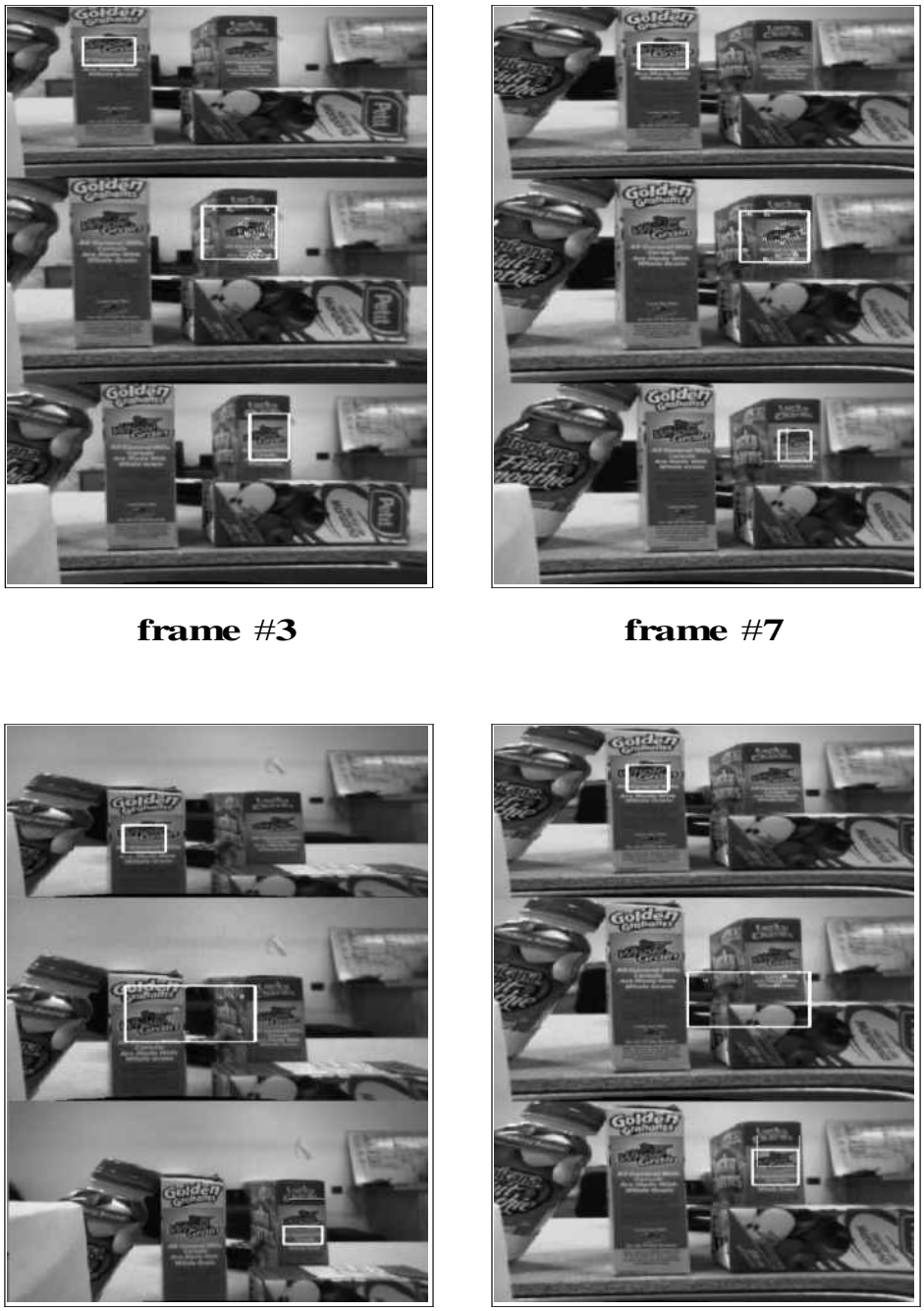

Figure 5-7: Sample results from Global (top), Local (middle), and Particle Filter (bottom) while tracking in a complex environment.

4) Qualitative Comparison: For the qualitative comparison, our proposed method was tested against the Incremental Learning tracker presented in [Ross 2008] using the Dudek sequence. This sequence provides several challenges for the trackers that include several different changes in pose, appearance, and lighting. For these tests, the sequence was half-sampled keeping only the odd numbered frames. From Fig. 5-8, it can be seen that both trackers do a good job of tracking the head in the sequence. Both trackers handle partial obscurity (frame 215) and a change in appearance similarly (frame 453). As relates to alterations in pose and lighting, there seems to be a difference in what kinds of changes affect the trackers' performances. Because of the probabilistic nature of the underlying global detector, there were a few cases in the Dudek sequence where the proposed tracker was unable to keep up with the person and lost the target object around frame 951. Both methods were also compared using two other data sets from [Ross 2008], namely, David_Indoors and Sylvester, which produced similar results (see Fig. 5-9). While the proposed method worked fairly robustly with the Sylvester sequence, the IL method consistently failed toward the end of this sequence when there was an abrupt change in the direction of motion of the target object. Since the proposed method is template-based, we note here that the most commonly occurring pose from a sequences was chosen to be its template. Each method has its strengths and weaknesses when it comes to tracking an object. The fact that the IL tracker is not template-based does give it some extra robustness when it comes to dealing with appearance and some pose changes. That being said, the template component does help the proposed method stick to the target throughout the sequences with a few exceptions. This was evident in the David_Indoors sequence where significant lighting change was also present along with pose changes. As seen in the frames in the top row of Fig. 5-9, even though the proposed

method does not fully encompass the face, it is still able to track a portion of it and was seen to remain with it throughout the sequence. This can also be seen in the Sylvester sequence where, due to a significant pose change, the IL tracker loses the target while the proposed method manages to stick with it and continues tracking the target throughout the rest of the sequence.

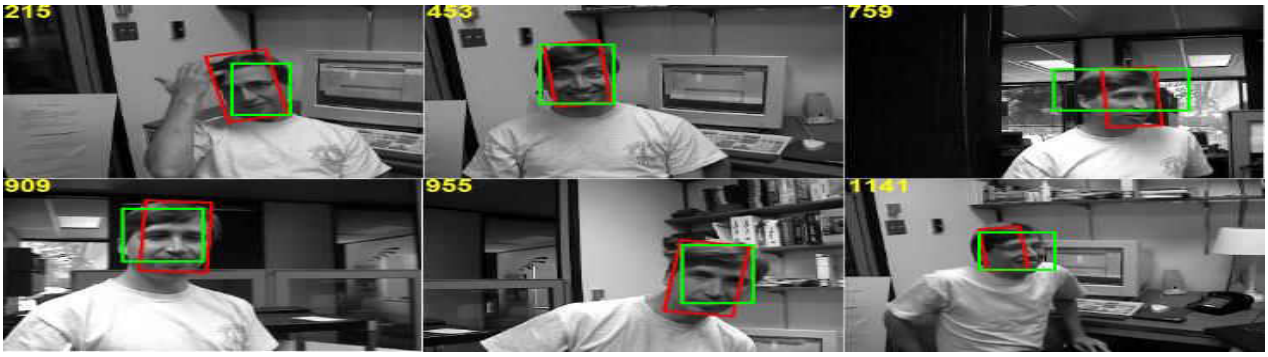

Figure 5-8: Sample comparison of results from IL tracker (red) to proposed method (green) on Dudek sequence.

Figure 5-9: Sample comparison of results from IL tracker (red) to proposed method (green) on David_indoors (top) and Sylvester (bottom) sequences.

The proposed method was also compared to a state-of-the-art particle filter based presented in [Wang 2011], which will be referred to as the TOT method (Third-Order Tensor). Since we were unable to obtain the code to test the method, we utilized the data presented in their paper and compared it with ours. Both trackers were tested using the Dudek Sequence and the Sylvester

sequence. As seen with the Dudek sequence in the top row of Fig. 5-10, the TOT method was able to handle partial occlusion better than our proposed method. Both trackers performed identically well when it came to the Sylvester sequence. However, a small limitation of the TOT method is its need for reliable initial tracking to collect good samples for accurate initialization of the proposed appearance model.

Figure 5-10: Sample comparison of results from TOT method (red) to proposed method (green) on the Dudek (top) and Sylvester (bottom) sequences.

5.4.2 Closed Loop Experiments

The previous experiments show that the proposed method can perform better than its counterparts and on par, or in some situations better, than existing methods that exist. While these tests present issues that may affect the performance of the vision system based on its surroundings, they do not however take into account the effects of operating with a control scheme in a real-time scenario. To demonstrate that our method operates effectively in a real situation, we used the two similar objects (cereal boxes) experiment from the previous section to show that the proposed method performs better in conjunction with our control system than it does with the ferns method, which is what has been used so far on the UCF-MANUS. For this

version, a Golden Grahams and Cheerios box have been placed side by side one another and, using the gross and fine motion protocols [Kim 2012], the robot has been asked to align its grippers with the Cheerios box. As stated in the previous explanation of this experiment, even though the cereal boxes look different to us, they are identical when seen by the robot. The robot managed to successfully track the Cheerios box using the proposed method as opposed to using the ferns method. As seen in Fig. 5-11, the particle filter was able to maintain its ROI on the Cheerios box and successfully line up the grippers of the robot to the cereal box. The ferns method, however, started to track the Cheerios box but eventually jumped to the Golden Grahams box and tracked those instead. The ferns method also took about three times more iterations to complete the task than the proposed method. The fact that ferns jumps between the two boxes causes the error used for the control system to become erratic as clearly seen in the two graphs in the top row of Fig. 5-12.

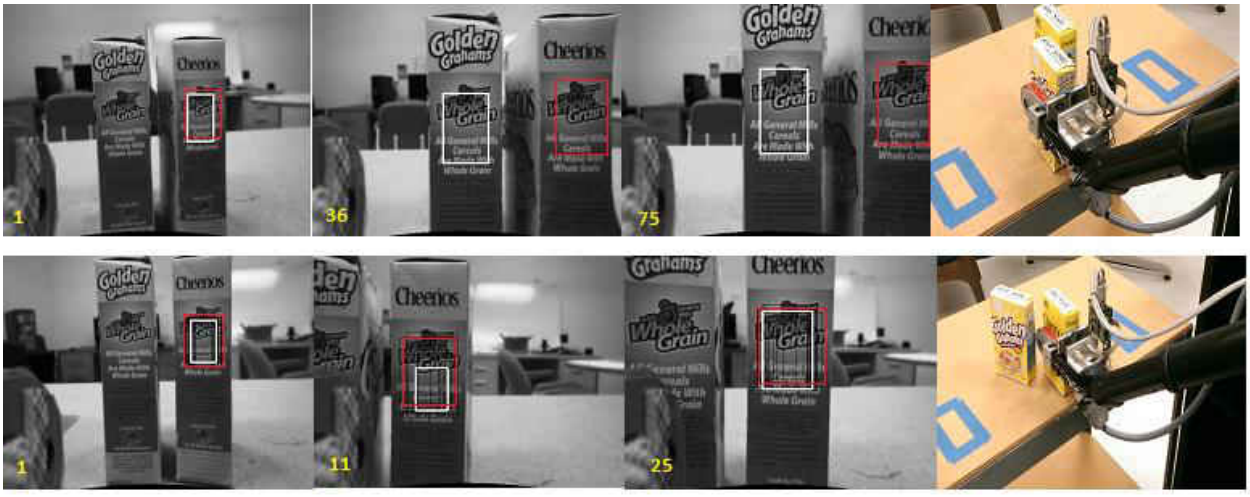

Figure 5-11: Comparison of servoing results using Ferns-based (top) and particle-filter-based (bottom) tracking. Frames were taken from the start, middle and end of the sequence. The desired object is highlighted in red and the real-time tracking ROI generated by each method is shown in white.

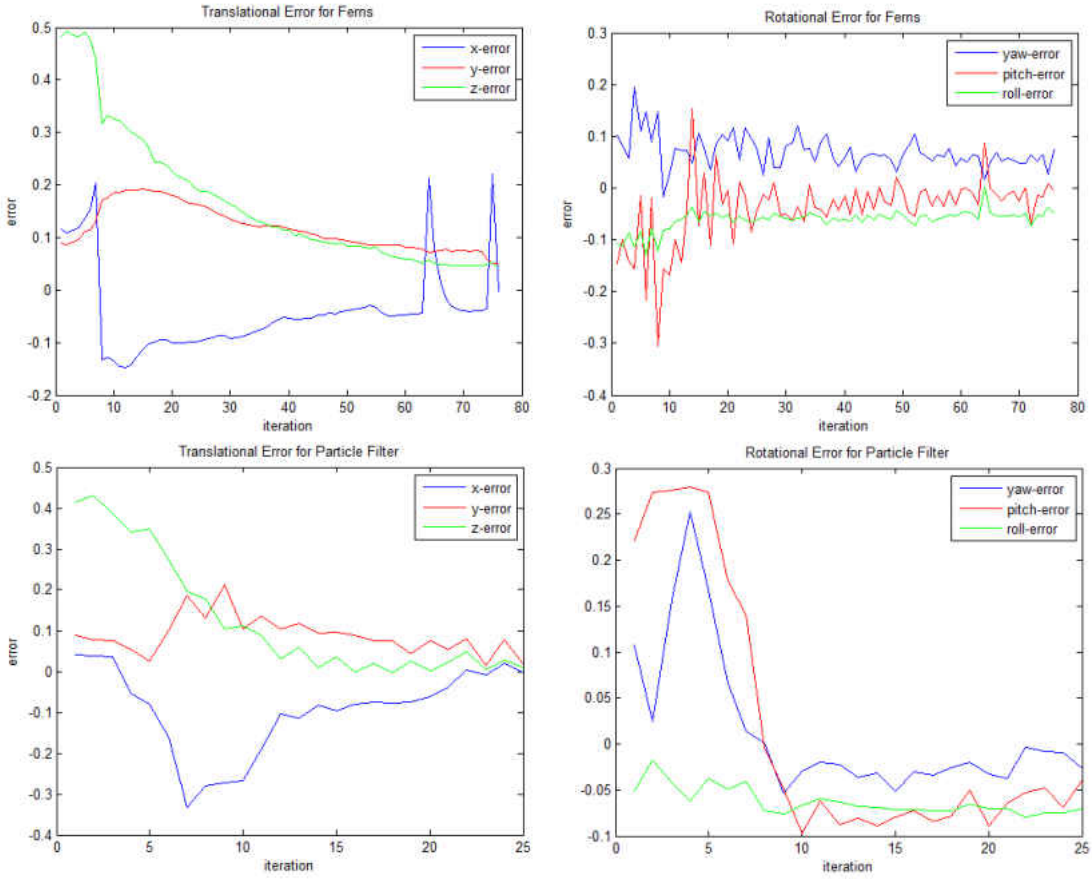

Figure 5-12: Robot end-effector position and orientation error profiles during a closed-loop visual servoing task using global detector-based tracking (top) vs. the particle-filter based tracking method (bottom).

5.5 Discussion

A particle filter (PF) based fusion framework is proposed to incorporate the global and local information for a visual tracking application relevant to assistive robotics. Iterative updating of particles' weights and a resampling process are formulated under the PF approach to deploy a fusion of global and local information. A novel metric to quantify the degree and quality of overlap between two polygonal ROIs is defined and used to evaluate the prior and posterior pdfs. Experiments with video sequences gathered from the UCF-MANUS assistive robotic testbed show that the proposed method is effective at tracking a target object without fiduciary markers

and in a natural environment. Even in the presence of perturbations such as large initial offset, multiple identical objects, and low video frame rate, the proposed approach is consistently successful at target tracking compared with exclusively local or global approaches that show poor performance and are not robust to the aforementioned perturbations. The real-time implementation of the proposed tracker inside the UCF-MANUS fine motion control scheme shows its effectiveness at robustifying visual servoing.

5.6 References

D. Lowe, "Distinctive Image Features from Scale-invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, 2004, pp. 91-110.

D.-J. Kim, Z. Wang, and A. Behal, "Motion Segmentation and Control Design for UCF-MANUS – An Intelligent Assistive Robotic Manipulator," *IEEE/ASME Transactions on Mechatronic*s, vol. 17, no. 5, pp. 936-948, October 2012.

D.-J. Kim, Z. Wang, N. Paperno, and A. Behal, "System Design and Implementation of UCF-MANUS – An Intelligent Assistive Robotic Manipulator," *IEEE/ASME Trans. on Mechatronics*, vol. 19, no. 1, pp. 225-237, February 2014.

S. Hutchinson, G.Hager, and P. Corke, "A Tutorial on Visual Servo Control," *IEEE Trans. Robot. Automat.,* vol. 13, pp. 582-595, Aug. 1997.

H. Grabner, M. Grabner, and H. Bischof, "Real-Time Tracking via Online Boosting," in *Proc. Conf. British Machine Vision,* pp. 47-56, 2006.

M. Ozuysal, P. Fua, and V. Lepetit, "Fast Keypoint Recognition in Ten Lines of Code," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, pp.1-8, 2007.

J. Shi and C. Tomasi, "Good Features to Track," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593-600, 1994.

S. Birchfield, *Source Code for the KLT Feature Tracker*, http://www.ces.clemson.edu/~stb/klt/, 2006.

B.W. Silverman, *Density Estimation for Statistics and Data Analysis, Vol. 26*, CRC press, 1986.

D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental Learning for Robust Visual Tracking," *International Journal of Computer Vision,* vol. 77, no. 1, pp.125-141, 2008

H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, 2008.

Y. Fang, A. Behal, W.E Dixon, and D.M. Dawson, "Adaptive 2.5D visual servoing of kinematically redundant robot manipulators," *Proc. of the 41*st *IEEE Conference on Decision and Control,* 2002, vol.3, no., pp.2860,2865 vol.3, 10-13 Dec. 2002.

A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, pp. 1–45, 2006.

S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filter for on-line nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 174-188, 2001.

I. M. Rekleitis, "A Particle Filter Tutorial for Mobile Robot Localization," *Centre for Intelligent Machines, McGill University, Technical Report TR-CIM-04-02*, 2004.

J. S. Liu, R. Chen, and T. Logvinenko, "A theoretical framework for sequential importance sampling and resampling," In A. Doucet, N. de Freitas, and N.J. Gordon, editors, *Sequential Monte Carlo in Practice*, Springer-Verlag, 2001.

S. Sinha, *Source Code for the GPU-KLT algorithm,* http://www.cs.unc.edu/~ssinha/Research/GPU_KLT/

CHAPTER 6: CONCLUSION AND FUTURE RESEARCH

6.1 Introduction

The purpose of these methods was to provide improvements to the UCF MANUS that would robustify its performance, allowing it to work in more complex environments and handle more difficult situations. The first improvement was a refinement of the current Fine Motion control algorithm that ensures that the end-effector will come in line with the desired object so that a grasping procedure can be performed either by the MANUS itself or the human operator. The second improvement was a compensatory gross motion procedure that improved the chances of the MANUS properly identifying the chosen object and the third improvement was a particle filter based tracking method that allows the MANUS to track the desired object in more complex environment.

6.2 Chapter Objectives

- Summarize methods used to improve fine motion control system.

-Summarize methods used for compensatory gross motion

-Summarize particle filter visual servoing method

- Discuss the future scope of research to further these improvements and optimize other MANUS systems.

6.3 Summary of Design Modification

6.3.1 Fine Motion Optimization

The optimization of the fine motion control involved adding a weight matrix to the initial scheme that restricts the movement of the feature points being tracked to inside an artificial ROI. To ensure stability, the weights are then given an upper and lower bound and solved for using quadratic programming.

During Fine Motion, if the outermost feature points are determined to be leaving the ROI the optimized controller is used in place of the nominal controller. If quadratic programming function determines that the solution for the weights is infeasible, the weights are then set to a predetermined value that initiates a backward motion of the end-effector away from the object. This allows the object to still remain in view of the camera and allows for the feature points to be transitioned back into the artificial ROI even if no feasible solutions are found. This method is described in detail in [Wang 2012] with some minor modifications to the actual implementation.

In transitioning the method from being implemented with both Matlab and Visual Studios to just being implemented in Visual Studios, the nominal controller was modified from a proportional controller into a PI controller. Since the optimal controller requires more processing time for its calculations, the end-effector velocities needed to be slowed down enough so that when the optimal controller was initiated it had time to implement itself before the object was lost. This

initially led to a higher steady state error than the required threshold needed for a successful alignment. The PI controller was implemented to reduce the steady state error allow the end-effector to complete its task.

The roll error for fine motion is also bounded by ±90°. This ensures that the optimal grasping position is achieved. When the roll error is found to be greater than|90|, it is reset to 180-$|e_r|$ and the image is flipped so that it better matches the template and more appropriate errors are obtained. The other errors are then set to their negative with the exception of whichever errors correspond to the z-direction in the camera coordinate system. This is to ensure proper alignment with the desired object.

6.3.2 Compensatory Gross Motion

The compensatory gross motion was designed to give the UCF MANUS different vantage points to identify an object. The method consists of moving the end effector along a circle to investigate the different sides of the object it has been asked to retrieve. The end-effector will be moved to the closest three sides of the object, being the right, left and top from the initial position, until the object is identified or all sides have been investigated.

If there are not enough feature points left to attempt identification, the end-effector will be moved to a pre-defined position. The position selected for this implantation has been the top view of the object since all identifiable objects for the UCF MANUS are grabbable from the top position. If there are not enough feature points to attempt identification again, then the object is

considered unidentifiable by the current system. This method is an extension of the compensatory method mentioned in [Kim 2012].

6.3.3 Particle Filter Based Visual Servoing

The particle filter based method utilized here was developed to take advantages a local and global tracking method to create a robustified visual servoing method. The method utilizes the Kanade-Lucase-Tomasi local tracker and ferns global detector/tracker as the components of the overall method. The training for the ferns method is done offline and the resulting templates are added to an object database.

The particle filter is initiated by creating the target ROI based on the features obtained using ferns to identify the chosen object. Once, the initial ROI is created, particles are formed by shifting the boundaries of the ROI by a randomly generated variable with a uniform distribution and are initially given equal weight. The KLT tracker will also be initiated within the ROI so that it can track the object throughout the frames.

As the tracker progresses, the particles weights are updated to determine whether or not they are still viable to the process. The weights are determined by the overlap metric which looks to see how much of the identified object is within the bounds of the particle. If the particle is given a weight below the threshold, it is culled from the existing particles. If several particles are culled at once, new ones are generated in a similar fashion to the initiation step using the current output

ROI for the particle filter. This method is very closely follows the framework described in [Arulampalam 2001].

6.4 Scope of Future Research

The methods developed in this thesis have addressed some issues that were present in current algorithms used for gross motion and fine motion of a robotic manipulator as well as visual servoing algorithms. This provides a more robust system that can be used in more complex environments than previously possible. The scopes of future research in improving the UCF MANUS system are:

- Optimizing the user interface to make it easier to understand and be used

- Develop a system designed to understand and compensate for user deficiencies

6.5 References

Zhao Wang; Dae-Jin Kim; Behal, A., "Design of Stable Visual Servoing Under Sensor and Actuator Constraints via a Lyapunov-Based Approach," *Control Systems Technology, IEEE Transactions on* , vol.20, no.6, pp.1575,1582, Nov. 2012 doi: 10.1109/TCST.2011.2168958

Dae-Jin Kim; Zhao Wang; Behal, A., "Motion Segmentation and Control Design for UCF-MANUS—An Intelligent Assistive Robotic Manipulator," *Mechatronics, IEEE/ASME Transactions on*, vol.17, no.5, pp.936,948, Oct. 2012 doi: 10.1109/TMECH.2011.2149730

S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filter for on-line nonlinear/non-Gaussian Bayesian tracking,” *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 174-188, 2001.

APPENDIX A: CALCULATION FOR UNDERLYING POLYGONS[3]

[3] This section is comprised of sections of the manuscript "Particle Filter-Based Robust Visual Servoing for UCF-MANUS – An Intelligent Assistive Robotic Manipulator" by N. Paperno, Z. Wang, D.-J. Kim, and A. Behal which is currently pending submission.

To compute $r(a,b)$ measurements for A_a, A_b, and $A_{overlap}$ are needed. We know that the area of an arbitrary polygon Π with N pivot points $\{p_{x,j}, p_{y,j}\}, j = 1, \dots, N$ is given as follows

$$A_{\Pi} = \frac{1}{2}\sum_{j=1}^{N}(p_{x,j}p_{y,j+1} - p_{x,j+1}p_{y,j}) \quad \text{(A-1)}$$

Thus, the area of the polygons A_a and A_b can be calculated using (17) while the area of overlap can be computed by knowing the intersection points of the two underlying polygons. An intersection point (p_x^*, p_y^*) of two polygons Π_1 and Π_2 can be found using line segments $l_1$and l_2 from each polygon as

$$p_x^* = \frac{b_1c_2 - b_2c_1}{a_1b_2 - a_2b_1}, \ \ if\ a_1b_2 - a_2b_1 \neq 0 \quad \text{(A-2)}$$

$$p_x^* = \frac{a_2c_1 - a_1c_2}{a_1b_2 - a_2b_1}, \ \ if\ a_1b_2 - a_2b_1 \neq 0 \quad \text{(A-3)}$$

where each line segment is defined as $a_i p_x + b_i p_y + c_i = 0$.