Particle Filter-Based Robust Visual Servoing for UCF-MANUS-An Intelligent Assistive Robotic Manipulator

N. Paperno, Z. Wang, D. J. Kim and A. Behal*

Department of EECS, University of Central Florida, Orlando, FL 32826, USA

Abstract: A particle filter based tracking scheme is proposed to robustify visual servoing of objects in the UCF-MANUS camera-in-hand vision setup. Instead of simply fusing global and local information, a concatenation of the two sources of information is proposed here which enables the combination of the two independent measurements with a synergistic collaboration between them. A novel overlap metric to encode the degree and quality of overlap between two arbitrarily shaped Regions of Interest (ROIs) is defined to facilitate the prior and posterior pdfs in the particle filter setup. A sub-ROI is defined and utilized in the observation step to facilitate the global target detection. Based on extensive experimental results under a variety of scenarios obtained by using the UCF-MANUS assistive robotic testbed, it is seen that the proposed particle filter based fusion approach is superior to other non-fused global detection or local tracking approaches. The efficacy of the proposed approach has also been verified using standard data sets. Finally, robustification of a hybrid visual servoing technique is shown by implementing the proposed particle-filter based tracker during closed-loop operation in real-time.

Keywords: Particle-filter, visual servoing.

1. INTRODUCTION

In a visual tracking application, a target is detected and tracked over time to perform a given task. There is 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 *et al.* [1] 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 [2], an Extended Kalman Filter (EKF) framework was used to estimate the position and orientation 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 [3] using fuzzy logic. Lee *et al.* [4] proposed a fusion of local odometry and global magnetic compass information to control an omnidirectional mobile robot. Moore *et al.* [5] proposed a local frame based robotic navigation to overcome disadvantages of global frame and local body frame representations such as increase in uncertainty, multi-modal noise, *etc.* Stephen *et al.* [6] 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) [7] features and backward correction algorithm were efficiently used to deal with uncertainty. In [8], Rodriguez-Losada *et al.* 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 *et al.* [9] 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 [10] and Monte Carlo (MC) localization [11] 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

*Address correspondence to this author at theDepartment of EECS, University of Central Florida, Orlando, FL 32826, USA; Tel: 407-882-2820; Fax: 407-882-2819; E-mail: abehal@ucf.edu

E-ISSN: 2409-9694/15 © 2015 Avanti Publishers

predecessors such as the EKF. Examples of comparisons between the particle filter and EKF can be found in [12, 13]. 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 [14]. In [15], Wang *et al.* propose a novel tracking method by incorporating the efficiency of the mean-shift algorithm with the multi-hypothesis characteristics of particle filtering in an adaptive manner. In [16], an offline-boosted detector was used to amend the proposal distribution of the particle filters for multi-target tracking. In [17], 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 [18]. In [19, 20], a novel multi-person tracking method was proposed in a particle filter framework, where both detectors and classifiers were used. In [21], a particle filter approach was extended with depth estimation of the target for tracking multiple targets with possibility of overlapping. Recently in [22], 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 [23-27].

In this work, the use of the term ‘fusion’ of local and global information is different from conventional categories of fusion terms including data-level fusion, feature-level fusion, and decision-level fusion. Instead of merging local and global information at the same level, this paper aims to make a systematic chain *via* a synergistic concatenation of global and local sources of information. Here, synergistic concatenation means that local and global information is used in an interconnected sequence wherein the local information informs the global measurement and the global measurement in turn updates the prior distribution. *Via* this concatenation, a robust Region of Interest (ROI) for a tracked object is successfully propagated using a particle filter framework. The algorithm proposed in this paper is novel in that it merges information extracted from disparate sets of features by utilizing a novel measure of overlap between minimal enclosing polygons - this overlap metric encodes the quality and degree of the overlap and is utilized in the definition of the observer posterior distribution. Furthermore, the measurement itself is robustified *via* the use of a pertinent sub-image defined from underlying particles. Specifically, the term synergistic is utilized to denote the fact that during complex scenarios (such as target tracking problem in a low frame rate video with multiple identical targets and large initial offset), both the underlying primary measurements fail when used exclusively but are synergistically able to support one another such that the proposed approach is highly successful.

The remainder of this paper is organized as follows. In Section II, the problem is motivated *via* illustration of cases of non-robust tracking gleaned from experiments in the Assistive Robotics Laboratory at UCF [28, 29]. Next, in Section III, a particle filter based fusion framework is presented along with the various steps of implementation. Section IV presents experimental results and discussion; the first part of this section presents tracking results with a camera mounted in an eye-in-hand configuration on the UCF-MANUS assistive robotic testbed while the second part of this section presents the real-time implementation of a hybrid visual servoing technique that is shown to be robustified by the inclusion of the proposed particle-filter based tracker. We conclude with some remarks in Section V.

2. PROBLEM MOTIVATION

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 end-effector. To formulate the problem, consider the dynamics of a pixel (or feature point) $p = \begin{bmatrix} p_x & p_y \end{bmatrix}^T$ on the target object as captured by the end-effector mounted camera as follows [30];

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

Where J_p is known as the image Jacobian and defined as follows;

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

In the above equation, Z enotes the Euclidean distance along the z-direction of the camera frame, while $V = \left[\begin{array}{cc} v_c & \omega_c \end{array}\right]^T$ 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 (dyn) and (Jacobian) 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 (see [28, 29] for more details on this assistive robot) is a Wheelchair-Mounted Robotic Arm (WMRA) that 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 frame-rate 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.

3. PARTICLE FILTER BASED TRACKING FRAMEWORK

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 $\mathbf{g}_t$ are extracted by a global detector. Using a known template model with a feature point set $\mathbf{G}_d$, one can find another set of feature points $\mathbf{g}_d \subset \mathbf{G}_d$ which shows a one-to-one correspondence with $\mathbf{g}_t$. We can also define a global detector Region of Interest (ROI) $\mathbf{y}_t$ which is a minimal polygon enclosing all feature points in $\mathbf{g}_t$. We will use this global detector ROI $\mathbf{y}_t$ to indicate the identified target on the frame grabbed at time t. Furthermore, the local tracker can also track a set of feature points $\mathbf{L}_t$ at time t matching with the feature point set $\mathbf{L}_{t-1}$ in the frame grabbed at time $t-1$. Note that the feature points in $\mathbf{g}_t$ and $\mathbf{L}_t$ are independent and obtained through global detector and local tracker, respectively. Next, we define the i^{th} particle as follows

$$\boldsymbol{\pi}_t^i = \left\{\mathbf{x}_t^i, w_t^i\right\} \qquad i = 1,\ldots,N_p \quad (3)$$

Where $\mathbf{x}_t^i$ 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 $\mathbf{l}_t^i \subset \mathbf{L}_t \quad \forall i = 1,\ldots,N_p$, we define a particle filter ROI $\mathbf{x}_t^i$ on the frame grabbed at time t as a minimally enclosing polygon of the residual image features in the set $\mathbf{l}_t^i$ as $\mathbf{x}_t^i = \mathbf{r}\left(\mathbf{l}_t^i\right)$. This polygon ROI $\mathbf{x}_t^i$ for $\mathbf{l}_t^i$ can be explicitly given as

$$\mathbf{x}_t^i = \mathbf{r}\left(\mathbf{l}_t^i\right) = \left(\begin{array}{cccc} p_{t,1}, & p_{t,2}, & \cdots & p_{t,N_x} \end{array}\right)$$

Where $p_{t,i} \in \mathbf{l}_t^i$ and N_x denotes the number of points on the boundary of this polygon. Note that $\mathbf{x}_t^i$ can be considered as a polygon representation of the associated local feature set $\mathbf{l}_t^i$ in the image space. These particles are used to obtain particle filter output ROI $\hat{y}_t^p$ based on particle ROIs $\mathbf{x}_t^i$ and their associated weight w_t^i. The global detector ROI $\mathbf{y}_t$ 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 $\mathbf{y}_{1\,:\,t}$ up to time t. A posterior pdf can be described by a set of N_p random samples (*i.e.,* particles) as follows [40].

$$p(\mathbf{x}_{0\,:\,t} \mid \mathbf{y}_{1\,:\,t}) \approx \sum_{i=1}^{N_p} w_t^i \cdot \delta(\mathbf{x}_{0\,:\,t} - \mathbf{x}_{0\,:\,t}^i) \tag{4}$$

Where $\mathbf{x}_t^i$, w_t^i, and $\mathbf{y}_t$ have been previously defined. Figure **6** shows an example of particles ROIs x_t^i and output ROI $\hat{y}_t^p$ 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.

3.2. Particle Filter Framework

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

$$r(\mathbf{a},\mathbf{b}) \triangleq \frac{\mathbf{A}_{overlap}}{\mathbf{A}_{\mathbf{a}}} \cdot \frac{\mathbf{A}_{overlap}}{\mathbf{A}_{\mathbf{b}}} \cdot \exp\left(-kf(\left|d\right|)\right). \tag{5}$$

Here, $\mathbf{A}_{\mathbf{a}}$ and $\mathbf{A}_{\mathbf{b}}$ denote, respectively, the area enclosed by the ROIs $\mathbf{a}$ and $\mathbf{b}$ while $\mathbf{A}_{overlap}$ denotes the area of overlap between the two ROIs. Furthermore, the function $f\left(\cdot\right)$ 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 $r(\mathbf{a},\mathbf{b})$ capture the degree of overlap while the last factor encodes the quality of the overlap, *e.g.,* between two particles, say $\mathbf{a}_1$ and $\mathbf{a}_2$ with identical ROI sizes and similar overlap area with $\mathbf{b}$, we deem the one located more centrally with respect to $\mathbf{b}$ as having a better overlap than the other whose location is more peripheral. Note that by definition, $r(\mathbf{a},\mathbf{b}) \in \left(0,1\right]$. In this paper, the empirical determination of the pdfs was obtained using the following choices: $k = 0.02$ and $f(\left|d\right|) = \left|d\right|$. Details of calculations for the underlying polygons considered in the definition of $r(\mathbf{a},\mathbf{b})$ 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 [7] and SURF [37] are commonly used detectors, in this paper, we turn toward a real-time implementable global detection-based tracker known as *ferns* [32].

3.2.1. Initialization Step

Given an initial specification of a target ROI $\mathbf{R}_1$ in the first frame as illustrated in the left part of Figure **2**, one can randomly generate particle ROIs $\mathbf{x}_1^i$ around

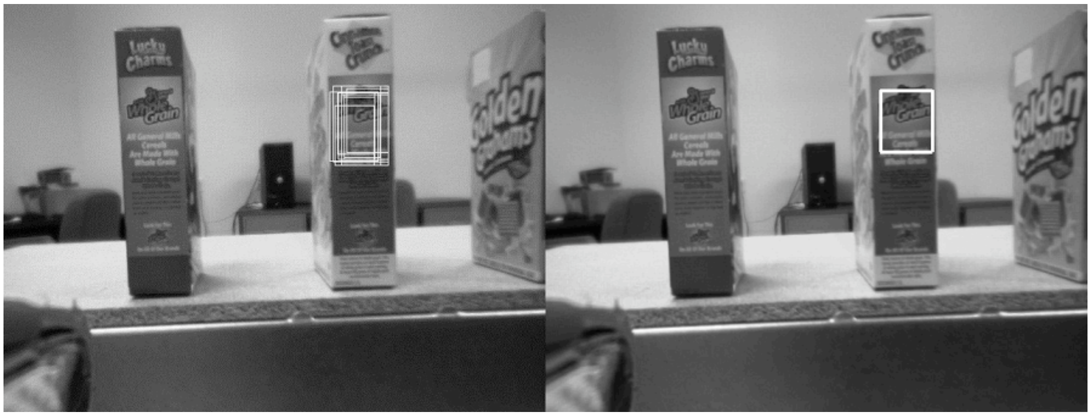


Figure 1: Particles ROIs xi t (left) and output ROI ^yp t (right) in frame 42 of a cereal-box video sequence recorded in the UCF Assistive Robotics Laboratory.

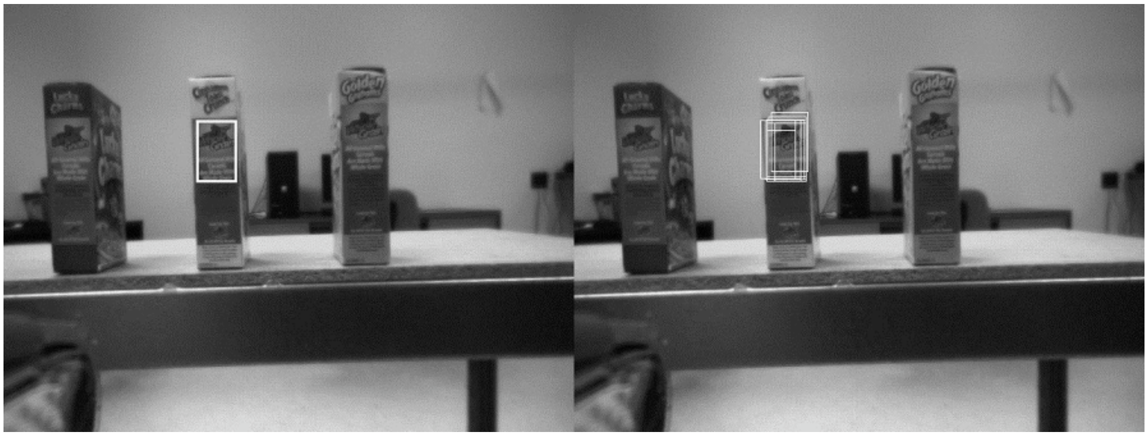


Figure 2: Initially specified ROI (left) and initial particle ROIs distribution (right).

the specified region as illustrated in the right part of Figure **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 $\mathbf{x}_1^i$ 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.

3.2.2. 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 $p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1\,:\,t-1})$ of the state at time $t-1$, the first step is to draw samples from the importance function as follows

$$\mathbf{x}_t^i \sim p(\mathbf{x}_t \mid \mathbf{x}_{t-1}). \tag{6}$$

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 (1) and (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 [33, 34] based on optical flow which is based on the following relationship holding under small relative motion between frames:

$$p_t \approx p_{t-1} + gd \tag{7}$$

Where $p_t, p_{t-1} \in \mathbb{R}^2$ denote a feature point in the image at time t and $t+dt$, respectively, $g \in \mathbb{R}^{2\times2}$ 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 $\mathbf{L}_{t-1}$ at frame # $t-1$ into $\mathbf{L}_t$ in the image frame captured at time t, we can find the feature point subset $\mathbf{l}_t^i \subset \mathbf{L}_t$ for $\mathbf{x}_t^i$ matching with $\mathbf{l}_{t-1}^i \subset \mathbf{L}_{t-1}$ associated with i^{th} particle ROI $\mathbf{x}_{t-1}^i$. The sample ROI for $\mathbf{x}_t^i$ can then be obtained as the minimum polygon including all the feature points in the set $\mathbf{l}_t^i$.

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 (flow) 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.

3.2.3. 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 $\mathbf{x}_t^i$ 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 $\mathbf{X}_t$ as follows

$$\mathbf{X}_t = \bigcup \mathbf{x}_t^i,\ \forall w_t^i > w_{th},\ i = 1,\dots,N_q. \qquad (8)$$

In this predicted sub-region $\mathbf{X}_t$, 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 $\mathbf{x}_t^i$ which are representative of the pdf $p(\mathbf{x}_t \mid \mathbf{y}_{0\,:\,t-1})$, 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 $\mathbf{X}_t$ of the current frame and comparing with the template frame, one can get the set of feature points $\mathbf{g}_t$ (defined previously in Section Prelim) which shows one-to-one correspondence with a template feature point set $\mathbf{g}_d \subset \mathbf{G}_d$. Then, one can find an observed global detector ROI $\mathbf{y}_t$ which encloses all feature points in the feature point set $\mathbf{g}_t$. 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

$$\bar{w}_t^i \propto p(\mathbf{y}_t \mid \mathbf{x}_t^i) w_{t-1}^i \qquad (9)$$

Where $p(\mathbf{y}_t \mid \mathbf{x}_t^i)$ is defined as the observer posterior that can be approximated based on empirical observations as follows

$$p(\mathbf{y}_t \mid \mathbf{x}_t^i) = \frac{2}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(1 - r(y_t, x_t^i))^2}{2\sigma^2} \right) \qquad (10)$$

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 Figure **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 (wpdate) are normalized using

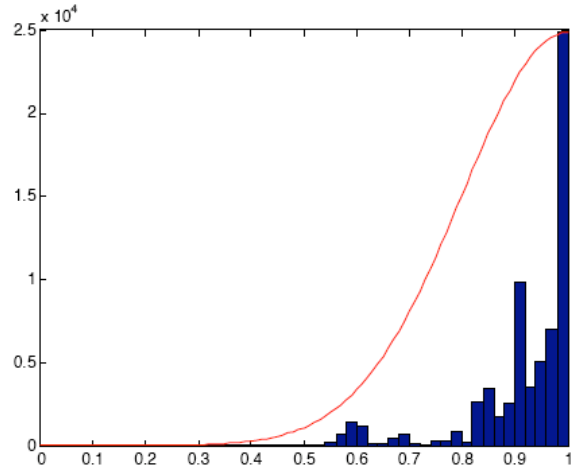


Figure 3: Posterior pdf for the particle filter overlaid over the histogram of the overlaps taken from various objects. This histogram was created by taking the overlaps of the ferns results to the ground truth of various objects.

$$w_t^i = \frac{\bar{w}_t^i}{\sum_{j=1}^{N_p} \bar{w}_t^j}. \qquad (11)$$

3.2.4. 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 [41]. 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 $0 < \epsilon < 1$.

$$\begin{array}{ll} \text{y1:} & \hat{\mathbf{y}}_t^p = \mathbf{x}_t^{(\arg\max_j w_t^j)} \\ \text{y2:} & \hat{\mathbf{y}}_t^p = \sum_{i=1}^{N_p} \mathbf{x}_t^i w_t^i \\ \text{y3:} & \hat{\mathbf{y}}_t^p = \sum_{i=1}^{N_p} \mathbf{x}_t^i w_t^i;\ \ w_t^i \geq \epsilon \end{array} \qquad (12)$$

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 $\hat{\mathbf{y}}_t^l$ 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 $\hat{\mathbf{y}}_t^g = \mathbf{y}_t$. Here, the superscripts $p, l,$ and g, respectively, refer to particle-based, local and global approaches.

3.2.5. 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 $\boldsymbol{\pi}_t^i = \left\{\mathbf{x}_t^i, w_t^i\right\}, \quad i = 1,\ldots,N_p$ needs to be replaced with a new set of particles $\tilde{\boldsymbol{\pi}}_t^i = \left\{\tilde{\mathbf{x}}_t^i, \tilde{w}_t^i\right\}, \quad i = 1,\ldots,N_p$ 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 $\tilde{\mathbf{x}}_t^i$ 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 [Liu01]

$$cv_t^2 = \frac{var(w_t^i)}{E^2(w_t^i)} = \frac{1}{N_p}\sum_{i=1}^{N_p}(N_p w_t^i - 1)^2 \tag{13}$$

$$ESS_t = \frac{N_p}{1 + cv_t^2}. \tag{14}$$

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

$$ESS_t < \eta_1 N_p,$$

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 [40]. 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 $\mathbf{x}_t^i$ then associates with the newly generated local feature points located inside it.

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 $\mathbf{x}_1^i, \quad i = 1,\ldots,N_p$ randomly around $\mathbf{R}_1$; Generate local features $\mathbf{L}_1$; Associate subset $\mathbf{l}_1^i \subset \mathbf{L}_1$ with $\mathbf{x}_1^i$ and set $w_1^i = 1 / N_p$; Hence, the i^{th} particle can be described as follows
2. $$\boldsymbol{\pi}_t^i = \left\{\mathbf{x}_t^i, w_t^i\right\} \qquad i = 1,\ldots,N_p. \tag{15}$$
3. 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 $\mathbf{l}_{t-1}^i \to \mathbf{l}_t^i$. Then calculate $\mathbf{x}_t^i$ which is the minimal particle ROI enclosing feature points in the set $\mathbf{l}_t^i$.
4. Define sub-region $\bar{\mathbf{y}}_t$ according to the predicted particle ROI $\mathbf{x}_t^i > w_{th}$ as in (subregion).
5. Measurement $\mathbf{y}_t$ is obtained by global detector applied in the sub-region $\bar{\mathbf{y}}_t$, and then update weight w_{t-1}^i using $p(\mathbf{y}_t \mid \mathbf{x}_t^i)$ as in (wpdate).

6. Normalize weight according to (wormalize), and then calculate the output $\hat{\mathbf{y}}_t^p$ using (r1).

7. 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 $\tilde{\boldsymbol{\pi}}_t^i$.

8. Regenerate lost feature points for local tracker; update feature point set $\mathbf{l}_t^i$ for each particles.

9. Go to step 2 unless last frame has been reached.

4. EXPERIMENTAL RESULTS AND DISCUSSION

4.1. Tracking Results

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 [28]. The input images are grabbed through a Dragonfly 2 firewire camera with 640×480 pixel-sized, 8bit image. In this experiment, a *ferns* based detector/tracker [32] is adopted as the global tracker while a Kanade-Lucas-Tomasi (KLT) feature tracker [33, 34] 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 [40]. 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 (ovd) 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 $\tilde{\mathbf{y}}_t$ and estimated ROI $\hat{\mathbf{y}}_t^p$, $\hat{\mathbf{y}}_t^l$, and $\hat{\mathbf{y}}_t^g$ at time t as follows

$$E_t \triangleq 1 - r(\tilde{\mathbf{y}}_t, \hat{\mathbf{y}}_t^x) \qquad (16)$$

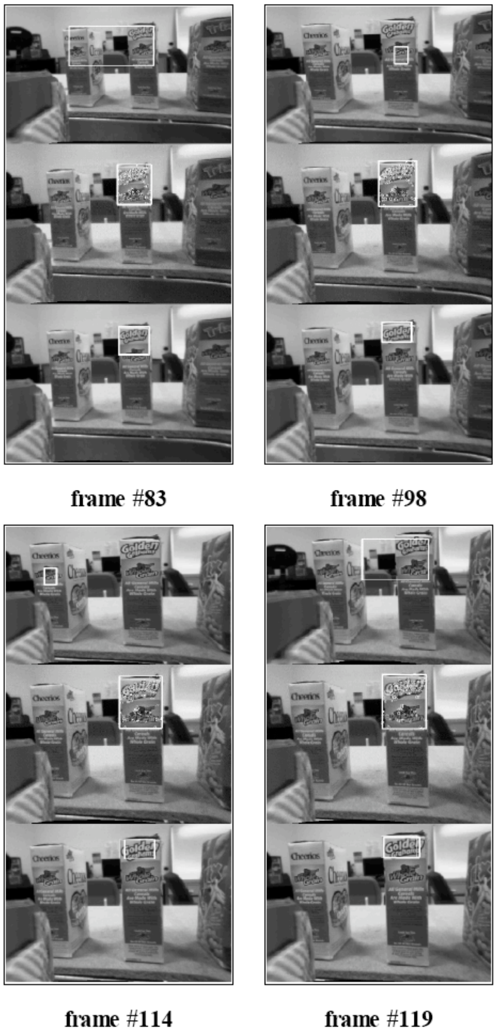
where $r(\cdot,\cdot)$ has been previously defined in (ovd) while $\hat{\mathbf{y}}_t^x$ stands for the output ROI obtained from the particle filter based, local, and global approaches, respectively, when x is p, l, and g. In Tables **1**, **2**, and **3**, the overall performance of these three methods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

$$RMSE = \sqrt{\frac{1}{N_t}\sum_{t=1}^{N_t} E_t^2}.$$

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

4.1.1. Target Tracking with Multiple Identical Objects

Figure **4** shows snapshots of observed global tracker target ROI $\hat{\mathbf{y}}_t^g$, local tracker ROI $\hat{\mathbf{y}}_t^l$, and



frame #83 frame #98

frame #114 frame #119

Figure 4: Sample results from three different ROI estimators during target tracking with multiple identical objects: global (top), local (middle), and particle-filter based (bottom) estimates.

particle filter output ROI $\hat{\mathbf{y}}_t^p$ 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 Figure **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 **1** as well as the first and second rows in each of the annotated frames shown in Figure **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 Figure **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 **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 **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.

4.1.2. 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 Figure **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. Figure **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

Table 1: Mean and Variance of Errors for all Used Estimators

	E_t in #77	E_t in #81	E_t in #86	E_t in #114	RMSE	variance of E_t
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

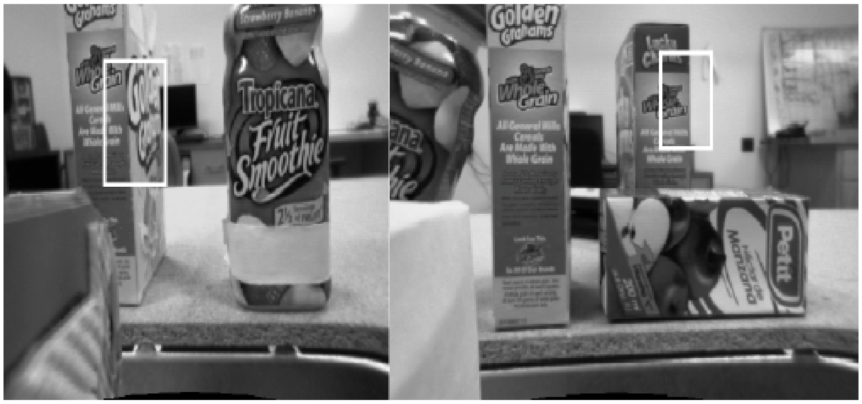
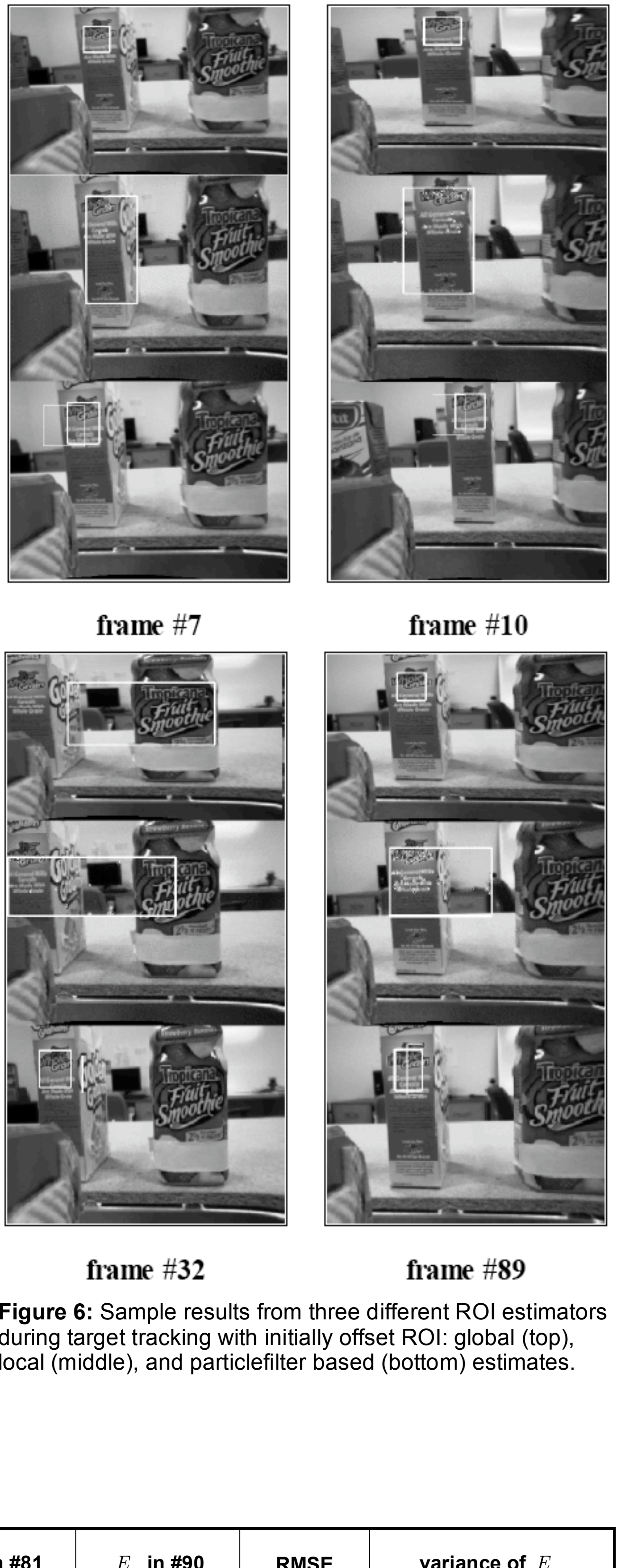


Figure 5: User initial target ROI selection with offset. Left: selection for the experiment in Section IV-A2. Right: selection for the experiment in Section IV-A3.

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 Figure **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 **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.

4.1.3. 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 Figure **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. Figure **7** shows the target tracking results for each of



frame #7

frame #10

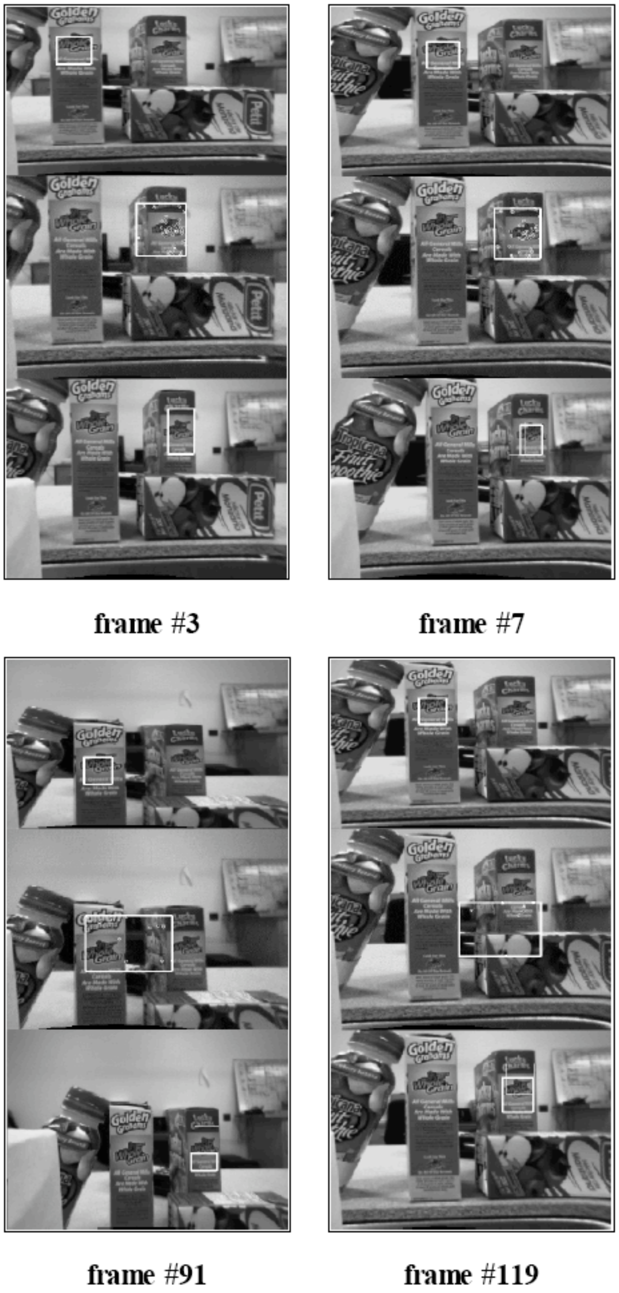
frame #32

frame #89

Figure 6: Sample results from three different ROI estimators during target tracking with initially offset ROI: global (top), local (middle), and particlefilter based (bottom) estimates.

Table 2: Mean and Variance of Errors for all Used Estimators

	E_t in #7	E_t in #30	E_t in #81	E_t in #90	RMSE	variance of E_t
Global	0.941	0.784	0.805	0.975	0.9724	0.005214
Local	0.975	0.980	0.986	0.972	0.9820	0.000271
Particle Filter	0.492	0.355	0.156	0.108	0.3196	0.03086



frame #3

frame #7

frame #91

frame #119

Figure 7: Snapshots of three different ROI estimators during the tracking in complex environment: top to bottom - global, local, and particle filter based estimates are shown in the pictures.

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 Figure **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 **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 **3** and each of the frames shown in Figure **7**, the particle filter approach outperforms by far the other two cases.

Remark 3: We note that the proposed algorithm took no more than 150ms to process a 640×480 pixel-sized, 8-bit image on a PC with Intel Core(TM) i7 970 CPU. In breaking down the total time, it was seen that the local tracking algorithm needed 120ms on average to track corresponding feature points between two consecutive frames while the global detector spent around 20ms to detect the matched feature points based on the template information. The computing burden for particle related computations was in the range of 2ms. Thus, it is clear to see that the computing burden for prediction and update of particles was miniscule compared with the time needed for visual processing. According to [43], one can expect the time consumption on the local tracking algorithm to be significantly reduced if the GPU-KLT algorithm can be applied by using the parallel computing power of a GPU. Thus, the proposed algorithm is available to be utilized in real-time (as demonstrated in Section ILE below) which is critical to any robotics application.

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

	E_t **in #7**	E_t **in #13**	E_t **in #91**	E_t **in #119**	**RMSE**	**variance of** E_t
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

4.1.4. Qualitative Comparison

For the qualitative comparison, our proposed method was tested against the Incremental Learning tracker presented in Ross *et al.* [36] 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 Figure **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 [36], namely, David_Indoors and Sylvester, which produced similar results (see Figure **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.

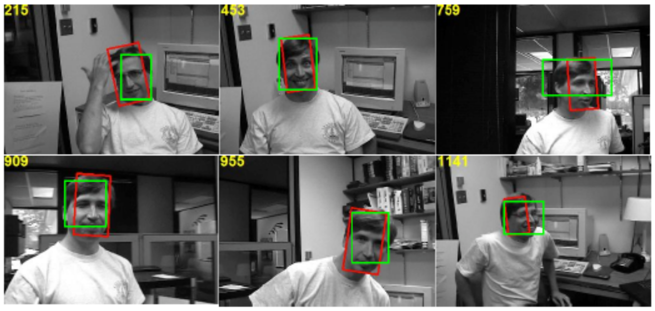


Figure 8: Sample comparison of results from IL tracker (red) to proposed method (green) on the Dudek sequence.

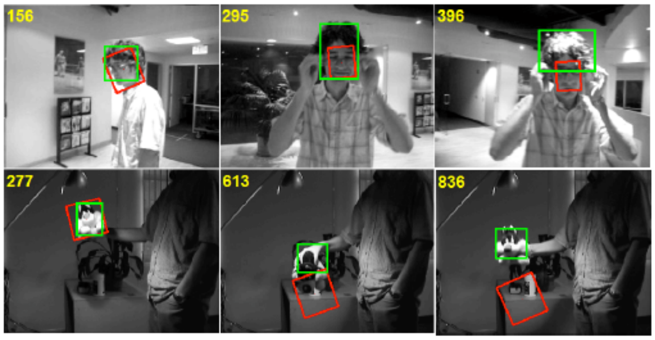


Figure 9: Sample comparison of results from IL tracker (red) and proposed method (green) on David_indoors (top row) and Sylvester (bottom row) sequences.

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 Figure **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.

The proposed method was also compared to a state-of-the-art particle filter based presented in [22], 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 Figure **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

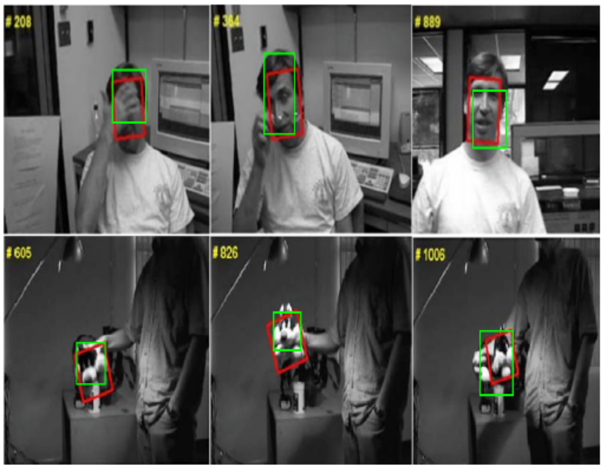


Figure 10: Sample comparison of results from TOT method (red) to proposed method (green) on the Dudek (top row) and Sylvester (bottom row) sequences.

initial tracking to collect good samples for accurate initialization of the proposed appearance model.

4.2. In-Loop Implementation

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 affects 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 [28], 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 Figure **11**, the particle filter was able to maintain its ROI on the Cheerios box and successfully line up the grippers of

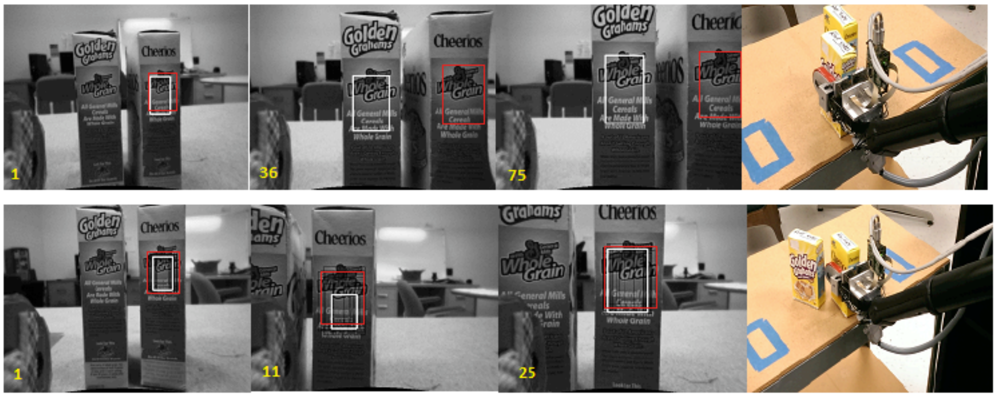


Figure 11: Comparison of servoing results using Ferns-based (top) and PF-based (bottom) tracking. Frames were taken from the start, middle, and end of the sequence. The desired object is highlighted in red and the real-time tracking ROI generated by each method is shown in white.

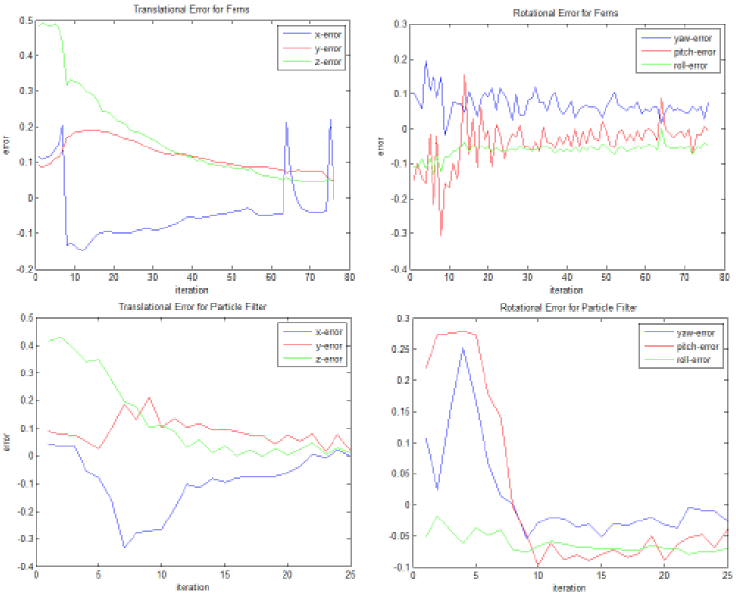


Figure 12: Robot end-effector position and orientation error profiles during a closed-loop visual servoing task using global detector-based tracking (top) vs the proposed PF-based tracking method (bottom).

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 as many 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 Figure **12**.

Remark 4: It is worth noting in this paper that we have tracked a single target in each of the experiments described above. This approach can be easily extended to solve the multiple target tracking problem by applying multiple sets of particles, where each set of particles is used to track one single target. In this extension, the local tracking algorithm only needs to be applied once for each frame while the global detector will be utilized multiple times based on the number of targets that one intends to track.

5. CONCLUSIONS

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. While this method works well enough to be implemented in conjunction with a robotic control scheme, it still needs to be refined to have wider applications. The testing on the Dudek sequence and other sequences from [36] show the efficacy of the proposed method in tracking more difficult and general objects through extended sequences. It does have problems with significant pose and lighting changes as well as very fast moving targets but those problems are common with other trackers available in literature as well. Future research will focus on improvements to alleviate these problems.

APPENDIX

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

$$\mathbf{A}_\Pi = \frac{1}{2}\sum_{j=1}^{N}\left(p_{x,j}p_{y,j+1} - p_{x,j+1}p_{y,j}\right).$$

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

$$p_x^* = \frac{b_1c_2 - b_2c_1}{a_1b_2 - a_2b_1}, \qquad \text{if } a_1b_2 - a_2b_1 \neq 0$$
$$p_y^* = \frac{a_2c_1 - a_1c_2}{a_1b_2 - a_2b_1}, \qquad \text{if } a_1b_2 - a_2b_1 \neq 0$$

where each line segments is defined as $a_ip_x + b_ip_y + c_i = 0$.

DISCLOSURE

This study was funded in part by Department of Education, NIDRR grant # H133G120275, in part by NSF grant numbers IIS-0649736 and IIS-1409823, and in part by a grant from the Florida Space Institute at UCF. However, these contents do not necessarily represent the policy of the aforementioned funding agencies, and you should not assume endorsement by the Federal Government.

REFERENCES

[1] Ishiguro H, Yamamoto M and Tsuji S. Omni-directional Stereo for Making Global Map. in Proc IEEE International Conference on Computer Vision Osaka, Japan 1990; 540-547.
http://dx.doi.org/10.1109/iccv.1990.139591

[2] Roumeliotis SI and Bekey GA. 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 1999; 11-22.

[3] Fu Y, Xu H, Li H, Wang S and Xu H. 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 2006; 1189-1193.

[4] Lee JH and Jung S. Global Position Tracking Control of an Omni-directional 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 2008; 246-251.
http://dx.doi.org/10.1109/mfi.2008.4648072

[5] Moore DC, Huang AS, Walter M, Olson E, Fletcher L, Leonard J and Teller S. Simultaneous Local and Global State Estimation for Robotic Navigation, in Proc. IEEE International Conference on Robotics and Automation, pp. 3794-3799,Kobe, Japan, 2009.
http://dx.doi.org/10.1109/robot.2009.5152763

[6] Se S, Lowe DG and Little JJ. Vision-Based Global Localization and Mapping for Mobile Robots, IEEE Trans. On Robotics 2005; 21(3): 364-375.
http://dx.doi.org/10.1109/TRO.2004.839228

[7] Lowe D. Distinctive Image Features from Scale-invariant Keypoints, International Journal of Computer Vision 2004; 60(2): 91-110.
http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

[8] Rodriguez-Losada D, Matia F, Jimenez A and Galan R. Local Map Fusion for Real-time Indoor Simultaneous Localization and Mapping, Journal of Field Robotics 2006; 23(5): 291-309.
http://dx.doi.org/10.1002/rob.20120

[9] Persson M, Duckett T and Lilienthal AJ. Fusion of aerial images and sensor data from a ground vehicle for improved semantic mapping. Robotics and Autonomous Systems 2008; 56: 483-492.
http://dx.doi.org/10.1016/j.robot.2008.03.002

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

[11] Ferrein A, Hermanns L and Lakemeyer G. Comparing Sensor Fusion Techniques for Ball Position Estimation. RoboCup 2005, Lecture Notes in Computer Science 2005; 154-165.

[12] Kotecha, Jayesh H, Djuric PM. Gaussian particle filtering. Signal Processing, IEEE Transactions on 2003; 51(10): 2592-2601.
http://dx.doi.org/10.1109/TSP.2003.816758

[13] Aydogmus O, Talu MF. Comparison of Extended-Kalmanand Particle-Filter-Based Sensorless Speed Control. Instrumentation and Measurement, IEEE Transactions on 2012; 61(2): 402, 410.

[14] Cappe O, Godsill SJ, Moulines E. An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo, Proceedings of the IEEE 2007; 95(5) 899-924.
http://dx.doi.org/10.1109/JPROC.2007.893250

[15] Wang J and Yagi Y. Adaptive Mean-Shift Tracking With Auxiliary Particles, IEEE Trans. on Systems, Man, and Cybernetics---Part B: Cybernetics 2009; 39(6): 1578-1589.
http://dx.doi.org/10.1109/TSMCB.2009.2021482

[16] Okuma K, Taleghani A, Freitas D, Little JJ and Lowe DG. A Boosted Particle Filter: Multitarget Detection and Tracking, in Proc. European Conf. Computer Vision, 2004.
http://dx.doi.org/10.1007/978-3-540-24670-1_3

[17] Ai YLH, Yamashita T, Lao S and Kawade M. 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 2008; 30(10): 1728-1740.
http://dx.doi.org/10.1109/TPAMI.2008.73

[18] Chen H and Li Y. Dynamic View Planning by Effective Particles for Three-Dimensional Tracking, IEEE Trans. On Systems, Man, and Cybernetics---Part B: Cybernetics 2009; 39(1): 242-253.
http://dx.doi.org/10.1109/TSMCB.2008.2005113

[19] Breitenstein M, Reichlin F, Leibe B, Koller-Meier E and Gool L. Robust Tracking-by-Detection using a Detector Confidence Particle Filter, In Proc. IEEE International Conference on Computer Vision 2009; 1515-1522. Kyoto, Japan, 2009.
http://dx.doi.org/10.1109/iccv.2009.5459278

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

[21] Huang C 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 2011; 41(1): 234-247.
http://dx.doi.org/10.1109/TSMCB.2010.2050878

[22] Wang Q, Chen F and Xu W. Tracking by Third-Order Tensor Representation, IEEE Trans. on Systems, Man, and Cybernetics---Part B: Cybernetics 2011; 41(2): 385-396.
http://dx.doi.org/10.1109/TSMCB.2010.2056366

[23] Peurum P, Venkatesh S and West G. A study on smoothing for particle-filtered 3-D human body tracking. Int J Comput Vision 2010; 87(1-2): 53-74.
http://dx.doi.org/10.1007/s11263-009-0205-5

[24] Won SP, Melek WW and Golnaraghi F. A Kalman/Particle Filter-Based Position and Orientation Estiamtion Method Using a Position Sensor/Inertial Measurement Unit Hybrid System, IEEE Trans Industrial Electronics 2010; 57(5): 1787-1798.
http://dx.doi.org/10.1109/TIE.2009.2032431

[25] Kristan M, Kovacic S, Leonardis A and Pers J. A Two-Stage Dynamic Model for Visual Tracking, IEEE Trans. on Systems, Man, and Cybernetics---Part B: Cybernetics 2010; 40(6): 1505-1520.
http://dx.doi.org/10.1109/TSMCB.2010.2041662

[26] del Rincon JM, Makris D, Urunuela CO and Nebel JC. 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 2011; 41(1): 26-37.
http://dx.doi.org/10.1109/TSMCB.2010.2044041

[27] Widynski N, Dubuisson S, Bloch I. Integration of Fuzzy Spatial Information in Tracking Based on Particle Filtering, IEEE Trans. on Systems, Man, and Cybernetics---Part B: Cybernetics 2011; 41(3): 635-649.
http://dx.doi.org/10.1109/TSMCB.2010.2064767

[28] Kim DJ, Wang Z and Behal A. Motion Segmentation and Control Design for UCF-MANUS - An Intelligent Assistive Robotic Manipulator, IEEE/ASME Transactions on Mechatronics 2012; 17(5): 936-948.
http://dx.doi.org/10.1109/TMECH.2011.2149730

[29] Kim DJ, Wang Z, Paperno N and Behal A. System Design and Implementation of UCF- MANUS - An Intelligent Assistive Robotic Manipulator, IEEE/ASME Trans. On Mechatronics 2014; 19(1): 225-237.
http://dx.doi.org/10.1109/TMECH.2012.2226597

[30] Hutchinson S, Hager G and Corke P. A Tutorial on Visual Servo Control, IEEE Trans. Robot. Automat 1997; 13: 582-595.

[31] Grabner H, Grabner M and Bischof H. Real-Time Tracking via Online Boosting, in Proc. Conf. British Machine Vision 2006; 47-56.

[32] Ozuysal M, Fua P and Lepetit V. Fast Keypoint Recognition in Ten Lines of Code, in Proc. IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis,

Minnesota 2007; 1-8. http://dx.doi.org/10.1109/cvpr.2007.383123

[33] Shi J and Tomasi C. Good Features to Track, in Proc. IEEE Conference on Computer Vision and Pattern Recognition 1994; 593-600.

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

[35] Silverman BW. Density Estimation for Statistics and Data Analysis, Vol. 26, CRC press 1986. http://dx.doi.org/10.1007/978-1-4899-3324-9

[36] Ross D, Lim J. Lin RS and Yang MH. Incremental Learning for Robust Visual Tracking, International Journal of Computer Vision 2008; 77(1): 125-141. http://dx.doi.org/10.1007/s11263-007-0075-7

[37] Bay H, Ess A, Tuytelaars T and Van Gool L. SURF: Speeded Up Robust Features, Computer Vision and Image Understanding (CVIU) 2008; 110(3): 346-359. http://dx.doi.org/10.1016/j.cviu.2007.09.014

[38] Fang Y, Behal A, Dixon WE and Dawson DM. Adaptive 2.5D visual servoing of kinematically redundant robot manipulators, Proc. of the 41 IEEE Conference on Decision and Control 2002; 3: 2860, 2865. 10-13. http://dx.doi.org/10.1109/cdc.2002.1184282

[39] Yilmaz A, Javed O and Shah M. Object tracking: A survey, ACM Comput Surv 2006; 38(4): 1-45. http://dx.doi.org/10.1145/1177352.1177355

[40] Arulampalam S, Maskell S, Gordon N and Clapp T. A tutorial on particle filter for on-line nonlinear/non-Gaussian Bayesian tracking, IEEE Trans. Signal Processing 2001; 50(2): 174-188. http://dx.doi.org/10.1109/78.97837441

[41] Rekleitis IM. A Particle Filter Tutorial for Mobile Robot Localization, Centre for Intelligent Machines, McGill University, Technical Report TR-CIM-04-02, 2004.

[42] Liu JS, Chen R and Logvinenko T. A theoretical framework for sequential importance sampling and resampling, In A. Doucet N. de Freitas, and NJ. Gordon, editors, Sequential Monte Carlo in Practice, Springer-Verlag 2001.

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

Received on 12-06-2015 Accepted on 07-07-2015 Published on 31-07-2015

DOI: http://dx.doi.org/10.15377/2409-9694.2015.02.01.4

© 2015 Paperno *et al.*; Avanti Publishers.
This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.