Context-Aware Tracking Robotic System Based on Context Toolkit

Kun Wang
Dept. of Systems and Computer Engineering, Carleton University
Contributor: Peter X. Liu

Introduction

Context-aware technology offers new opportunities for developers in software applications and end users in mobile distribution computing system. In addition, the interests of context-aware research also increasingly fall into the area of robotic control. This idea of context-aware robotic system conforms to the trend of intelligent robot, with the ability of sensor detection, context reasoning and decision making. For context-aware system design, several frameworks have evolved in previous research. The Context Toolkit, originally released from Georgia Tech., provides a framework to facilitate the building of context-aware applications using the Java and XML programming language. It has been popularly used to make prototype context-aware applications. In this poster, layered and centralized system architecture for the context-aware robotic system is presented based on the Context Toolkit. Specifically, a mobile robot, with local & remote sensors and human direct input, could sense the necessary contexts from surrounding environment and target. Thus the surrounding information could be monitored, proper decision could be inferred and corresponding task could be executed by the mobile robot. Visual contexts experiment of human tracking is performed based on the system architecture. The proposed system architecture could be further developed to fulfill more advanced context reasoning & inference and task execution in the assistive healthcare-related applications.

Interaction of Applications and Components in Context Toolkit

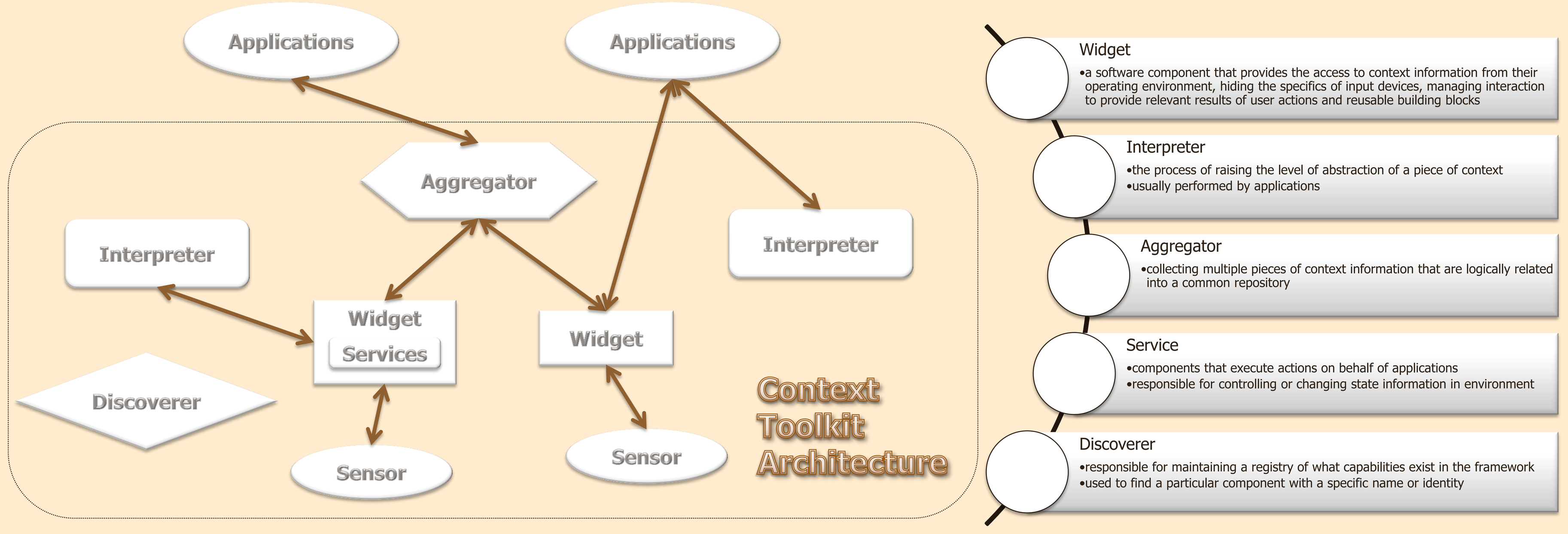

Fig. Components and interaction of Context Toolkit

Context-Aware Tracking Robotic System

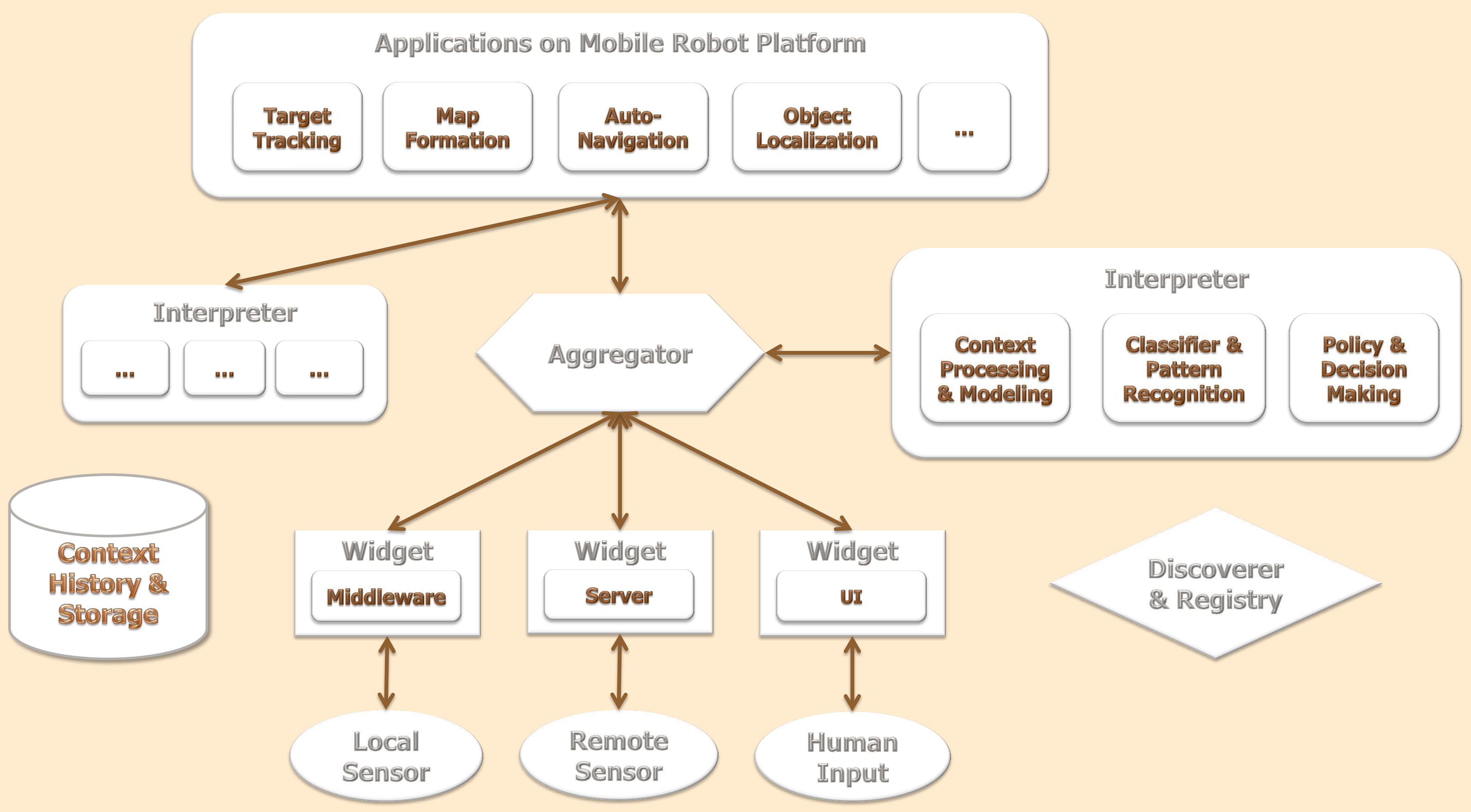

Fig. Context-aware robotic system structure block

Particle Filtering Harr-like Feature Detection Tracking Methods

❖ Harr-like feature detection algorithm is a rapid object detection algorithm using a boosted simple features to construct a cascade style of classifiers based on the AdaBoost learning approach.

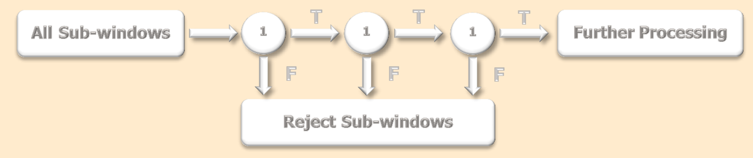

Fig. Depiction of classifiers in cascade style

Fig. Human face detection with Haar-like feature detection algorithm

❖ Particle Filtering (PF) method is implemented to fulfill more reliable tracking performance. Particle Filtering is a non-parametric technique with no functional form, using a set of weighted samples, or particles, to estimate the posterior distribution of target state.

1) System Transition Model:

$$\begin{bmatrix} r_k \\ s_k \\ r'_k \\ s'_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{k-1} \\ s_{k-1} \\ r'_{k-1} \\ s'_{k-1} \end{bmatrix} + \begin{bmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} v_k + \begin{bmatrix} 0 \\ 0 \\ r'_{est_{k-1}} \\ s'_{est_{k-1}} \end{bmatrix}$$

2) Observation Model:

$$z_k = \sqrt{r_k^2 + s_k^2} + w_k$$

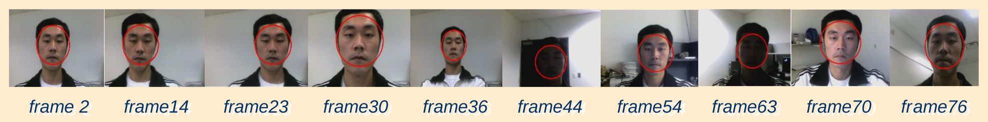

Fig. Human face detection with Particle Filtering enhanced Haar-like feature detection algorithm

Experimental Platform

The designed architecture and proposed tracking solution is implemented on the mobile robot platform -- PeopleBotTM from MobileRobots Inc.

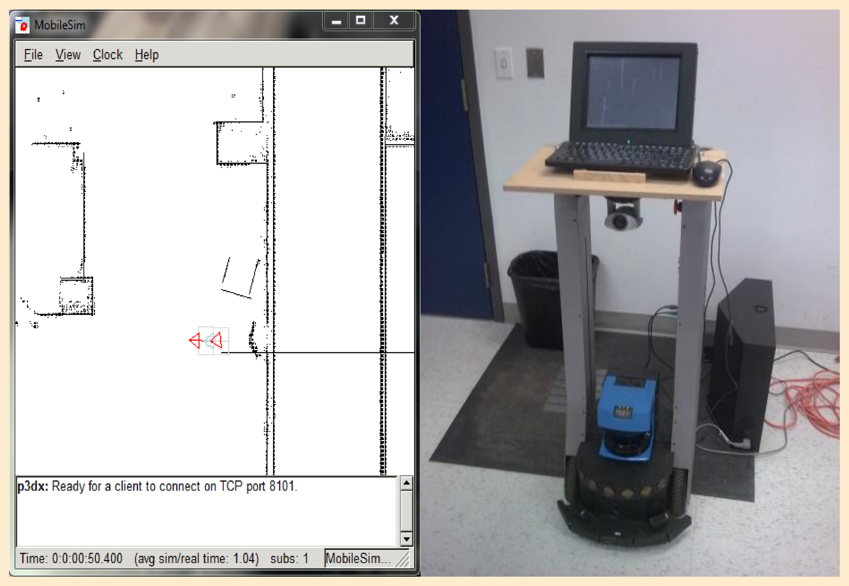

Fig. Experimental platform of PeopleBot™

Results

The proposed Particle Filtering enhanced Harr-like feature detection tracking solution is implemented on the mobile robot to perform simulation and two types of real-time on-line applications: straight line tracking and circle tracking.

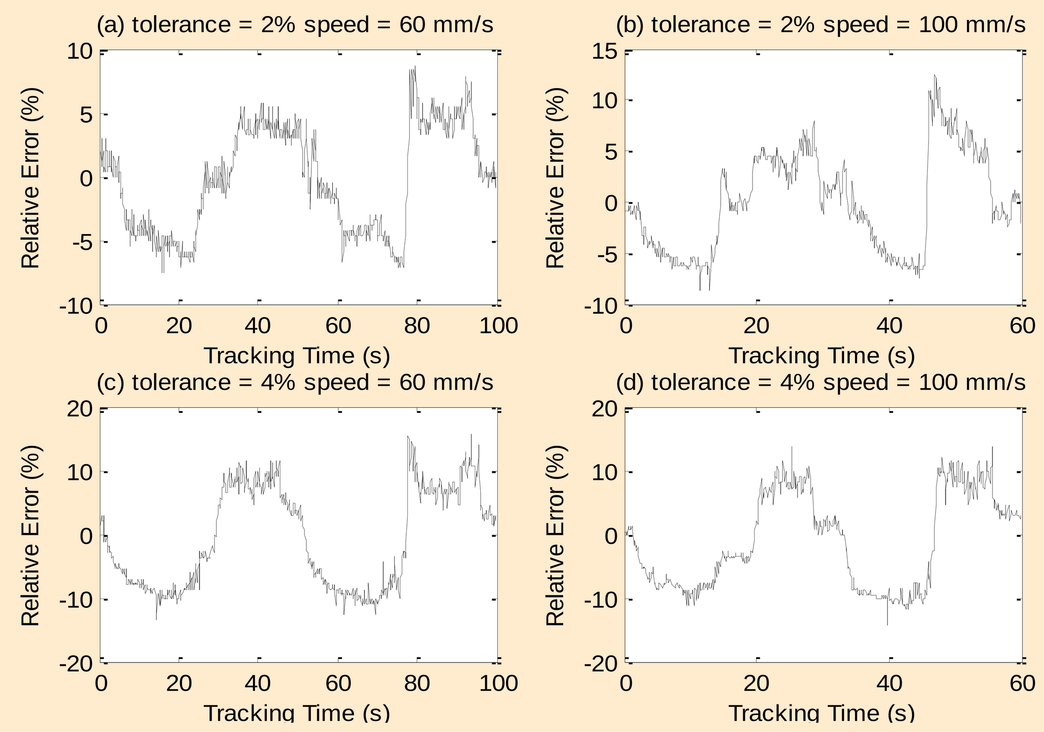

Fig. Results of straight line tracking. In this experiment, the action of person is (1) remains still (error=0, ideally), (2) moves forward (error<0), (3) stop (error=0), (4) moves backward (error>0), (5) stop (error=0), (6) moves forward (error<0), (7) abruptly moves backward (error>0), and (8) stop at last. The relative error is based on pixel.

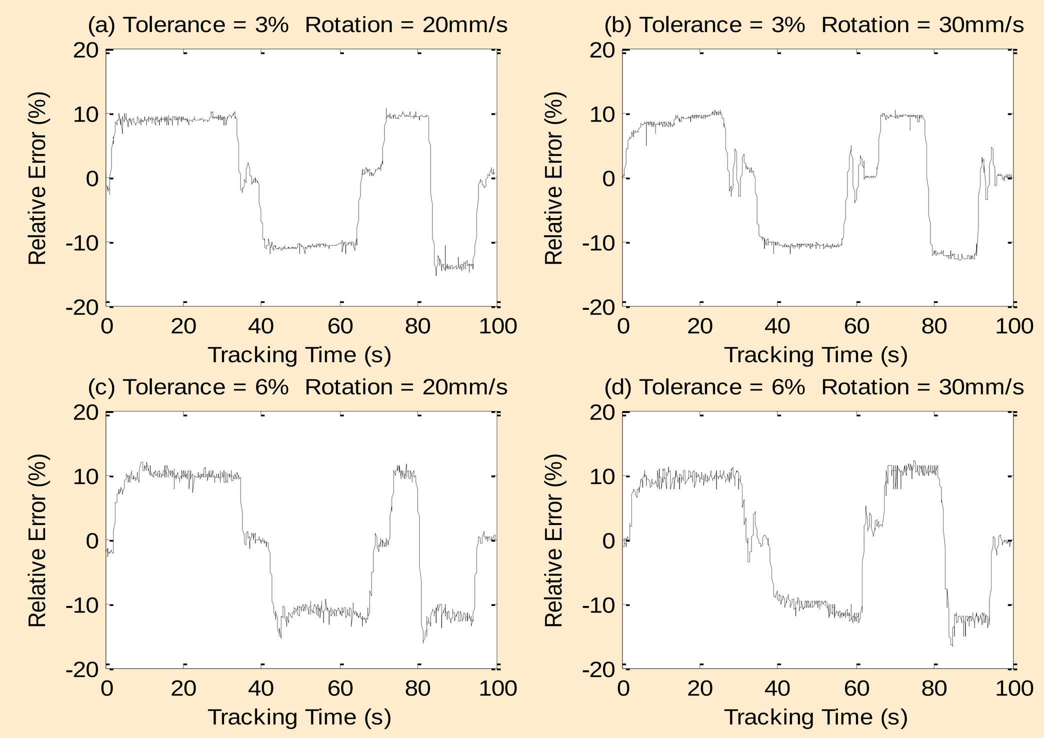

Fig. Results of circle tracking. In this experiment, the action of person is (1) remains still (error=0, ideally), (2) moves counter-clockwise (error>0), (3) stop (error=0), (4) moves clockwise (error<0), (5) stop (error=0), (6) moves counter-clockwise (error>0), (7) abruptly moves clockwise (error<0), and (8) stop at last. The relative error is based on pixel. The speed here is rotation speed.

Conclusion & Future Work

❖ Improving the proposed solution for tracking to obtain better performance.

❖ Developing more advanced context reasoning and inference mechanism based on artificial intelligence and machine learning techniques.

❖ Including more contexts other than the visual contexts and Perfecting the system architecture to fulfill complex tasks.

Acknowledgement

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) and industrial and government partners, through the Healthcare Support through Information Technology Enhancements (hSITE) Strategic Research Network.