SwRI People Tracking System Basic User’s Guide

This document outlines the basic functionality of SwRI’s people tracking system. The people tracking system consists of two separate sensor systems, an industrial PC, a POE switch, and a suite of software. The software was developed and tested on Linux version Ubuntu 12.04 , using Willow Garage ROS Fuerte. ROS undergoes numerous upgrades to its core and to its many packages. Users should be aware that updates often break existing software.

The software and all the necessary drivers should be installed on the PC delivered to NIST by SwRI. In addition, ROS and many of the launch files require certain environment variables to be set in order to run correctly. For the default user named nist with password nist, these have all been set using the .bashrc. For a new installation, all the ROS specific environment variables will be set upon installation.

However, several new environment variables are set to allow ROS scripts, or launch files to be independent of specific paths to data. This is important for ROS bag files which are often so big that it is not advisable to make copies. The tracking system’s launch files expect the following environment variables to be set:

1. NIST\_CLASSIFIERS -The directory where all the trained classifiers and configuration files for the detection nodes and nodelets are stored.
2. NIST\_BAG\_DIR - The directory where any bag files are stored
3. NIST\_DATA\_DIR – The directory for where the training images are stored
4. NIST\_EXCAL – Where the extrinsic calibration data is stored

To install the hardware,

1. Secure the three cameras such that the frame is level and 6 feet high.
2. Connect the two Basler cameras via Ethernet cable to any one of the powered ports (yellow) on the Netgear switch.
3. Connect the PC to the switch via any port, powered or not.
4. If desired, connect the Switch via any port to an external Ethernet port.
5. Apply power to the switch.
6. Connect the ASUS sensor USB port on the back side of the PC (FRONT USB PORT INCOMPATIBLE)
7. Connect a monitor and keyboard to the PC as usual

To run the software, open a terminal and type any one of the following

1. roslaunch System\_Launch run\_kinect\_node.launch
2. roslaunch System\_Launch run\_kinect\_nodelet.launch
3. roslaunch System\_Launch run\_stereo\_node.launch
4. roslaunch System\_Launch run\_stereo\_nodelet.launch

The first two options use the kinect sensor, while the second two use the stereo camera imagery to track people. The node.launch files execute variants of the software which use individual ROS nodes for detection, while the nodelet version uses the ROS nodelet functionality for detection of people. Nodelets use shared memory to transfer data between software, while nodelets send messages via Ethernet. Both were implemented in order to compare performance. Nodelets should outperform nodes.

Regardless of which option is selected, one should observe a window labeled “Tracker.” There may be other visualizations active which were used for debugging. After a short startup, the Tracker window should display the scene being observed by either the cameras or the Asus sensor. When people enter the scene, blue boxes should appear to outline the people being tracked. The number at the top of the box indicates the ID of the person being tracked. Should a person leave the field of view, or become occluded briefly, the tracker should recognize that the new detection is the same as a previous detection, and retain the original ID. Once a track is lost for a sufficient amount of time, a new ID number will be assigned. The estimated velocity appears at the bottom of the box.

In addition to the display, a myriad of other data is available for either display or analysis. For example, for each person being tracked, there is a time-stamped ROS message containing all the information defined by the requirements document, concluding the location of the person relative to the camera frame. Some data streams must be turned on by setting appropriate ros parameters.

# Software Summary

The runtime detection and tracking software is comprised of numerous packages. Some are native to Fuerte others were optional installations, and others were developed by SwRI for under this effort. Here we document only those under contained in the packages directory.

The following packages are used to generate the sensor data

1. aravis\_camera\_driver
2. aravis
3. kinect\_data

The following packages form a cascade of classifiers to detect the presence of people.

1. consistency – Generates rois with correct size to range ratio
2. HaarAda – An Adaboost classifier using Haar-like features on intensity
3. HaarDispAda – An Adaboost classifier using Haar-like features on disparity
4. HaarSvm – A Support Vector Machine using Haar-like features on intensity
5. HogSvm – A Support Vector Machine using Histograms of Oriented Gradients on intensity as features

Each of the packages has an associated launch directory. The launch directory contains scripts for training and debugging each detection node. The most often used launch scripts which appear in all 5 classifiers are train.launch and roi\_display.launch. The first is used to train the classifier, and is described in the training document. The second is used to display the regions of interest produced by the classifier. It provides a qualitative indication of how well each stage of the cascade is performing. Other launch files exist for running individual classifiers from directories and bags. These extra launch scripts are not maintained. They may or may not work, but because they were once useful, and because they are easily resurrected, they have not been deleted. For example, in the HaarAda package there is a run\_from\_dir.launch file which by default replays the imagery from a training directory along with its rois and allows one to see how well the system recalls the examples used for training. By setting directory parameter to point to an alternate directory containing labeled data, one could evaluate this individual classifier's performance on that data.

The object\_tracking package tracks the individual, assigns IDs and determines their location and velocity. Only a few options are available for this package. First, one may elect to store the tracking data to a file by setting the CVS\_Filename parameter to true. Another option is to display images or not by setting Show\_Images to true. Finally, setting Calculate\_Covariance to true produces the covariance data, at the expense of computational time.

The following packages provide general utilities used elsewhere

1. roi\_msgs - Defines messages between classifer nodes
2. roiViewer – Provides utility to view regions of interest found by a classifier to contain a person
3. Camera\_info\_manager – Indentical to standard ROS packages, with one line changed.

Three packages were developed to enable training of the classifiers.

1. Labeler -- A GUI for labeling data to create training sets
2. roiPlayer – A ros node which plays the labeled data in order to train each classsifier
3. image\_dump – A utility to extract the color and disparity images from bags