Inter-Classifier Feedback for Human-Robot Interaction in a Domestic Setting

Juhyun Lee, W. Bradley Knox, and Peter Stone

***Abstract*—For a mobile robot that interacts with humans such as a home assistant or a tour guide robot, activities such as locating objects, following specific people, and distinguishing among different people are fundamental, yet challenging robotic vision tasks. For complex object recognition and tracking tasks such as person recognition and tracking, we use the method of inter-classifier feedback to track both uniquely identifying characteristics of a person (e.g. face), and more frequently visible, but perhaps less uniquely identifying characteristics (e.g. shirt). The inter-classifier feedback enables merging multiple, heterogeneous sub-classifiers designed to track and associate different characteristics of a person being tracked. These heterogeneous sub-classifiers give feedback to each other by identifying additional online training data for one another, thus improving the performance of the overall tracking system. We implement the tracking system on a Segway base that successfully performed aforementioned activities to a second place finish in the RoboCup@Home 2007 competition. The main contribution is a complete description and analysis of the robot system and its implemented algorithms.**

***Index Terms*—Robotics vision, Human-robot interaction, RoboCup@Home**

I. Introduction

With the growing possibility of and demand for robots interacting in real-world environments, it is becoming increasingly important for robots to be able to interact with people. For robust human interaction in domestic environment, the robot must be able (1) to locate and identify common objects, (2) to follow people or guide people to places of interest, and (3) to distinguish the set of people with whom it commonly interacts while also successfully identifying strangers. In RoboCup@Home, an international competition designed to foster research on such interactive domestic robots, the robot has to show its performance in these tasks [1].

This paper presents the UT Austin Villa RoboCup@Home 2007 entry, a Segway-based robot and the second-place finisher in the competition. The robot demonstrated its ability to complete versions of all three of the tasks mentioned above. The main contribution of this paper is a complete description of the robot system and its implemented algorithms which enabled the robot's successful human-robot interaction in this broad, challenging, and relevant event, with an emphasis on the key component of our person recognition algorithm.

Detecting and/or tracking a particular person among multiple persons can be challenging for three reasons. The first reason is the noisy data. A person's most uniquely identifying visual feature is his or her face, which is not always present in a given video frame. Even if it is present, face detection algorithms may fail due to motion blur or bad lighting. The second reason is the demanding constraints of the task. Because a robot needs to operate in real-time with its limited processing power shared among all its tasks, the computational resources available for person tracking are constrained, thus limiting the algorithms that may be considered. The third reason is the mobile nature of the robot. The robot may only get to see a very limited view of a person under one lighting condition when it is trained. Worse, the trained characteristics of the person can change over space and time, due to pose and illumination changes. Then, the robot must be able to detect such changes autonomously and select new training data for its classifiers.

We use inter-classifier feedback for person tracking in a video stream that uses face recognition as a starting point, but augments it with tracking of more frequently visible, but perhaps less uniquely identifying characteristics such as the person's clothes. The main idea is that primary, uniquely identifying characteristics (e.g. faces) can be dynamically associated with secondary, ambiguous, possibly transient, but more easily computable characteristics (e.g. shirt colors). When primary characteristics are identifiable, they are re-associated with the secondary characteristics currently visible on the person. The secondary characteristics can then be used to track the person, even when the primary characteristics are not detected. We also show how each classifier helps the other classifiers to update their training data online to improve the overall performance of the system.

The main technical focus of our approach was on person tracking and recognition. As such, we focus in detail in this article on our algorithms for these tasks, including a novel method of combining classifiers of multiple characteristics of the person. The tasks our robot performed also required object tracking, for which we use the previously mentioned ARTags [14]. We summarize our use of ARTags in the context of their task-specific uses in Section VI.

The remainder of this paper is organized as follows. Section II describes the RoboCup@Home 2007 competition including its goals and format. After the motivation for a strong person tracking algorithm for mobile robots, we introduce the concept of heterogeneous inter-classifier feedback in domain-independent terms in III. We provide a proof-of-concept with a simple person tracker that we used in the competition in Section IV. Section V introduces the UT Austin Villa robot, including both hardware and software systems. Section VI

Juhyun Lee, W. Bradley Knox, and Peter Stone are with the Department of Computer Sciences, the University of Texas at Austin, Austin, TX 78712, the United States of America.

E-mail: {impjdi,bradknox,pstone}@cs.utexas.edu

Fig. 1. RoboCup@Home 2007 setting.

Task	Description
Navigate	Navigate to a commanded location
Manipulate	Manipulate one of three chosen objects
Follow and Guide a Human	Follow a human around the room
Lost and Found	Search for and locate previously seen objects
Who Is Who?	Differentiate previously seen and unseen humans
Copy-cat	Copy a human's movement in a game-like setting

Fig. 2. List of RoboCup@Home 2007 tasks.

describes our specific solutions for each task and our respective performances in them. Section VII discusses related works and Section VIII concludes the paper.

II. RoboCup@Home 2007

RoboCup@Home is an international research initiative that aims "to foster the development of useful robotic applications that can assist humans in everyday life" [1]. The eventual goal is to create fully functional robots that can assist humans at home in a variety of ways, performing any function that humans are currently hired to do, including assisted living and nannying. The RoboCup@Home community created a compelling and challenging set of tasks for the first year of the event in 2006 and raised the bar in 2007 [33].

In the 2007 competition, robots in a living room and kitchen environment (Fig. 1) had to complete up to four of six specified tasks. These tasks can be considered fundamental building blocks toward the complex behavior and capabilities that would be required of a fully functional home assistant robot. The specific tasks are described in Fig. 2.

Within each task, there were two levels of difficulty. The easier level, called the first phase, existed as a proof of concept and often abstracted away part of the problem (e.g. object recognition or mapping and navigation). The second, more difficult phase of each task was structured similarly to how the task would need to be performed in a real domestic setting. During each phase, there was a ten minute time limit to complete the task objectives.

After the specific tasks, all teams performed a free-form demonstration in what was called the *Open Challenge*, during which they showed off their most impressive technical achievements to a panel of other team leaders. Each event was scored and five teams advanced to the Finals. In the Finals, the five finalists performed demonstrations for trustees of the RoboCup organization, who determined the final standings.

Our robot attempted three of the six possible RoboCup@Home tasks. These tasks were *Lost and Found*, *Follow and Guide a Human*, and *Who Is Who?*. Each task is described in the following subsections. Our specific approaches to the three tasks are detailed in Section VI.

A. Lost and Found

This task tested a robot's ability to find an object that had been "lost" in the home environment. We competed in only the first phase of the *Lost and Found* task. In that phase, a team would hide a chosen object somewhere in the living environment at least five meters from their robot and out of its view. If the referees approved the location, the task began. The task ended successfully when the robot had moved within 50 cm of the item and had announced that it found it.

B. Follow and Guide a Human

In *Follow and Guide a Human*, a robot followed a designated human as he or she walked throughout the home and then, optionally, returned to the starting position (thus "guiding" the human).

1) First Phase: A team member led his or her robot across a path determined by the competition referees. The leader was permitted to wear any clothing or markers he chose. Once the leader and the robot reached the destination, an optional extension was to have the robot return back to the starting point with the human-following.

2) Second Phase: The rules were the same except that the human leader was a volunteer chosen from the audience. Therefore, the algorithm needed to robustly identify a person without markers or pre-planned clothing.

C. Who Is Who?

The *Who Is Who?* task tested person-recognition capabilities on a mobile robot. Both phases of the task involved the robot learning to recognize four people, the referees rearranging the people and adding one new person (a "stranger"), and the robot subsequently identifying the four known people and the stranger accurately.

1) First Phase: The four people lined up side-to-side while a robot moved among them and learned their appearances and names. Once the robot finished training, the four people and a stranger were arranged into a new order by the referees. Then, the robot again moved among the people, announcing their names as each was identified. One mistake was allowed.

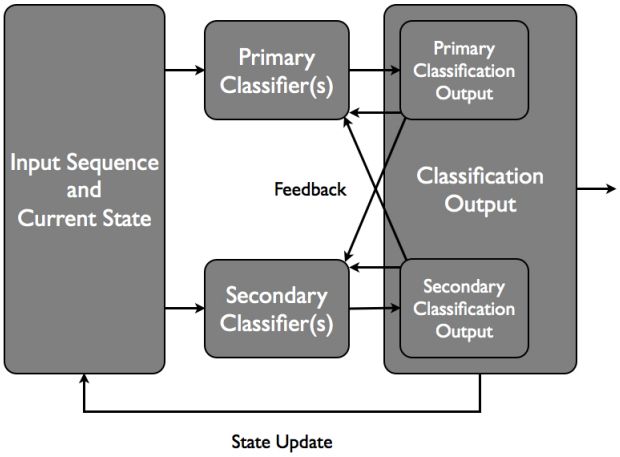

Fig. 3. Classification with heterogeneous inter-classifier feedback

2) Second Phase: The second phase was much like the first, but after the robot finishes training, the four known people and the stranger were placed by the referees in various locations around the entire living room and kitchen environment. The robot then had to search them out *and* correctly identify them.

III. Classification with Heterogeneous Inter-Classifier Feedback

A robust person tracking and recognition algorithm was essential in order to do well in two out of three tasks we decided to attempt. Before moving on to the implementation details of the person tracker we used, we describe the concept of heterogeneous inter-classifier feedback in domain-independent terms.

The overall system is a learning system which takes its current state and a part of the input sequence to compute its output and update its current state. During the output computation, an overall classifier is used which is built up from two or more heterogeneous sub-classifiers. Each sub-classifier solves its own classification problem by extracting different characteristics from the same input.

We divide the characteristics into two groups: primary and secondary. A primary characteristic must be a unique one that identifies a class. The classification problem of such primary characteristic may be computationally expensive, or susceptible to noisy input data. A secondary characteristic may be ambiguous, but computationally less expensive and more robust with respect to noise. Secondary characteristics can be introduced to leverage the shortcomings of a classification solely based on primary characteristics. This is also one of the main differences between our method and an ensemble. A secondary classifier is not used to vote for a better answer in case of an ambiguous classification result, but as a fall-back classifier for the times when the primary classifier returns no answer. There can be multiple characteristics in the same level, or more levels of characteristics may be introduced if the inter-characteristic relationship can be well-defined. Fig. 3 illustrates our scheme.

Algorithm 1 Classification with heterogeneous inter-classifier feedback (with 1 primary and 1 secondary classifier)

Require: $Input$: Input sequence, $State$: Current state
1: $SecChar \leftarrow ExtractSecChar(Input)$
2: $SecClass \leftarrow ClassifySecChar(SecChar)$
3: **if** $(IsPriCharRequired(State) =$ **true**$)$ **then**
4: $\quad PriChar \leftarrow ExtractPriChar(Input)$
5: $\quad PriClass \leftarrow ClassifyPriChar(PriChar)$
6: **else**
7: $\quad PriClass \leftarrow \emptyset$
8: $Class \leftarrow \emptyset$
9: **if** $(PriClass \neq \emptyset)$ **then**
10: $\quad Class \leftarrow PriClass$
11: $\quad$ **if** $(SecClass \neq \emptyset)$ **then**
12: $\quad\quad$ **if** $(PriClass \neq SecClass)$ **then**
13: $\quad\quad\quad$ **if** $(PriClass.Confidence > SecClass.Confidence)$ **then**
14: $\quad\quad\quad\quad TrainSecChar(SecChar, Class)$
15: $\quad\quad\quad$ **else**
16: $\quad\quad\quad\quad Class \leftarrow SecClass$
17: $\quad\quad\quad\quad TrainPriChar(PriChar, Class)$
18: $\quad\quad$ **else**
19: $\quad\quad\quad TrainSecChar(SecChar, Class)$
20: **else if** $(SecClass \neq \emptyset)$ **then**
21: $\quad Class \leftarrow SecClass$
22: Update $State$
23: **return** $Class$

Alg. 1 shows the basic structure of the algorithm we propose. $ExtractPriChar$ and $ExtractSecChar$ extract and return primary and secondary characteristics, respectively, of a given raw input. The returned characteristics are fed into each characteristic's classifiers $ClassifyPriChar$ and $ClassifySecChar$, respectively, which return the class label of the input. $TrainPriChar$ and $TrainSecChar$ are procedures for training the primary and the secondary classifier, respectively, with the training data and the class label. Finally, $IsPriCharRequired$ is a simple helper function that determines whether the heavy primary classifier should be run in the given cycle for performance reasons.

The computationally cheap, and thus more frequently invocable, secondary classifier can be used as the default (lines 1–2), while the more expensive primary classifier is invoked whenever a more accurate classification is needed (lines 3–7). If the condition of taking the branch is carefully chosen, near real-time performance can be achieved by avoiding a large classification expense each cycle. In case of a mismatch of the class labels returned by each classifier (line 12), the algorithm picks the class label with higher confidence depending on each characteristic's classification accuracy and/or $State$. Lines 14, 17, and 19 comprise the inter-classifier feedback which improves the classification performance of each classifier by adding more training data to the other class. In case all sub-classifiers do not return an answer, the overall classifier does not return an answer either. Our scheme does not try to find an answer if an answer cannot be determined from its sub-classifiers. However, our scheme still performs better than a

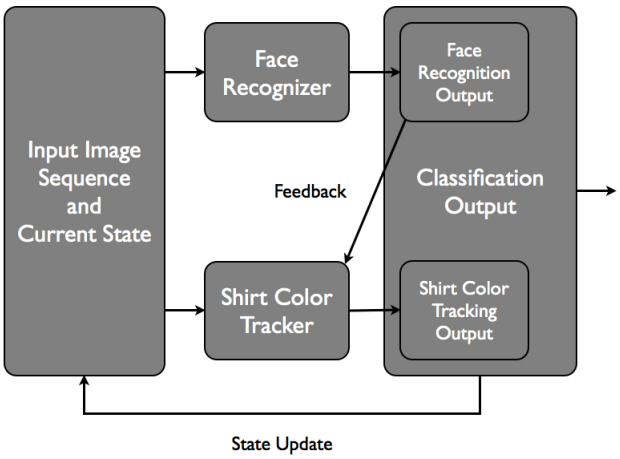

Fig. 4. Person tracking with 1 primary and 1 secondary classifier

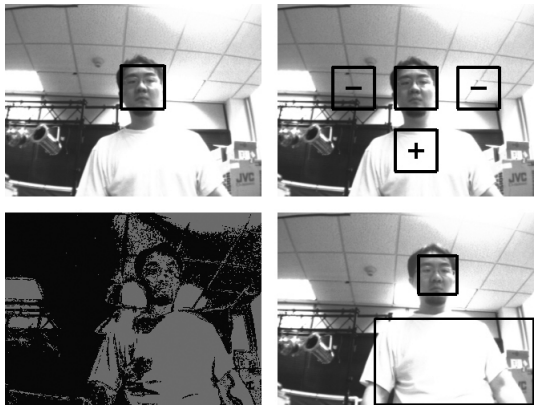

Fig. 5. Once the face is detected (a), the face's SIFT features are extracted to the face database and positive and negative regions of the shirt are sampled (b). The RGB-to-person mapping generated with the positive and negative histograms are shown in (c), and the shirt is detected in (d).

primary classifier alone.

IV. Person Tracking with Heterogeneous Inter-Classifier Feedback

Having discussed the general concept of heterogeneous inter-classifier feedback, next we apply the algorithm to a person tracking task. Since faces are unique, the primary characteristic for the person tracking task can be chosen to be the face. Since tracking the face alone is not sufficient to robustly track the person for previously mentioned reasons, a secondary characteristic of a person which is independent from the primary characteristic is chosen. Among different candidate characteristics, we choose the shirt of a person to be the secondary characteristic because it is easily visible, unless he or she is completely occluded by other objects. Fig. 3 is implemented for our domain as shown in Fig. 4.

A. Primary Characteristic Tracking

We divide the primary characteristic tracking task in two: the face detection and the face recognition. These correspond to $ExtractPriChar$ and $ClassifyPriChar$ in Algorithm 1, respectively. The face detection algorithm we use for the task is a boosted cascade of Haar-like features as discussed in [29]. It is implemented in the Intel Open Source Computer Vision Library, and shows a near-real-time performance (15 Hz) using limited resolution (160×120) images with our tablet PC. Extracting rectangular features from integral images as described in [29] does not suffer from a slight resolution decrease. The face recognition algorithm which extracts scale-invariant features (SIFT) [22] from cut-out face images suffers more from a resolution decrease. Rather than clipping the faces from the small 160×120 image used for the face detection, we extract the corresponding region in the original 640×480 image and extract the SIFT features of that region. These are used to distinguish among different faces by counting the number of matches during the recognition phase.

B. Secondary Characteristic Tracking

The secondary characteristic, a person's shirt, is trained when that person's face is successfully classified for several (e.g. 10) frames. Each person has his or her own positive and a negative histogram each with a size of $64 \times 64 \times 64$ RGB bins that contains the color information of the shirt the person is wearing. For example, a shirt with red and green stripes has high counts in $(63, 0, 0)$ and $(0, 63, 0)$. Fig. 5 shows which regions in an image are scanned for positive and negative samples of the shirt. Positive samples of the shirt colors are taken from a region as large as the face's bounding box, located 0.5 bounding boxes below the face. Negative samples are taken from two regions each as large as the face's bounding box, located 0.5 bounding boxes left and right of the face which should be the background or other objects in the scene. By maintaining positive and negative samples separately, a more accurate RGB-to-person mapping can be generated than by generating the mapping with positive samples alone. With this sampling scheme, we assume that the color of the shirt is relatively uniform in direction, i.e. we do not consider shirts having different colors in the front and in the back but we do not assume constant-colored shirts. We assume that each person has a distinctly colored shirt. In case there is more than one person having similarly colored shirts, the shirt of the latest person of interest is recorded, and the corresponding RGB values are mapped to that person.

To detect the shirt of a person in a given scene, we map each RGB pixel to a person ID with the mapping generated as described in the previous paragraph, and find the largest continuous blob containing only 1 ID. This approach is a modification of color-blob segmentation [26] where the colors of interest are assigned the same label. The blob detection and recognition algorithm is a lightweight operation that is carried out in real-time, 25 to 30 frames per second with a 320×240 resolution image. A more sophisticated algorithm such as edge detection may also be applied, but it requires additional object classification which needs a computation close to the face recognition itself (e.g. the Canny edge detector runs in 15 Hz) which is not desirable for tracking a weaker characteristic.

Another SIFT matching algorithm could have been chosen to distinguish shirts, but we found the color information of shirts yields better classification than the gray-scale SIFT features.

C. Adaptive Characteristic Tracking Algorithm Selection

Heavier vision processing is undesirable, since it results in lower frame rates which leads to less reactive robot behavior. We use an adaptive characteristic selection scheme for the robot's vision to achieve a higher frame rate. By the nature of human motion, the face is either constantly visible if facing the camera with limited movement, or constantly unrecognizable or occluded if not facing the camera or moving rapidly, although there can be a transition period between the two states. The face detection algorithm we use shows an average frame rate of 15 Hz. If the face detector can be skipped every other frame without decreasing the detection rate, the average frame rate would increase up to 22.5 Hz. Referring back to Algorithm 1, $IsPriCharRequired$ is defined as "every other frame". To avoid compromising the person detection rate, the secondary shirt detector has to show an equal or better detection rate than the face detector. We found this to be true in relatively steady lighting conditions.

D. Autonomous Real-Time Training Data Selection

Although we introduce the notion of primary and secondary characteristics indicating the different weights of each characteristic, there is no guarantee that a lower weighted characteristic will positively impact other characteristics, and vice versa. The primary tracking system can give feedback to the secondary tracking system to choose new training data for accurate classification. In our person tracking application, the face recognizing algorithm which computes scale-invariant features in normalized gray-scale images is more robust to color changes caused by ambient brightness changes. On the other hand, the RGB-to-person mapping used for shirt tracking is highly susceptible to such changes. If a person's face is correctly recognized, but the shirt is not detected, the RGB-to-person mapping can re-learn the shirt's colors, or update the RGB values for better classification under the changed lighting condition.

Since SIFT features are sensitive to directed lighting, a person moving in an indoor environment may be classified as a different person where there is more directed lighting than ambient lighting. However, the shirt's colors sampled with a Gaussian distribution has a slightly wider range in this case, and thus is still visible with directed lighting. Since the shirt is already known to belong to a certain person, the false-negative unknown face is then added to the training data of the primary classifier. Although conceptually possible, we decided not to integrate the re-training of the face recognizer on our laptop. The re-computation of the probability density function in our face recognizer takes more than 3 seconds on our robot-mounted laptop and less than 1 second on a 2 GHz dual-core laptop. We found that the robot operates more smoothly without the re-training, since it does not have to stop frequently for the PDF computation. The effect of autonomous real-time training data selection is shown in the Finals described in Section VI.

Fig. 6. UT Austin Villa home assistant robot.

V. Platform

This section introduces the hardware and software systems of the UT Austin Villa RoboCup@Home 2007 entry, shown in Fig. 6. The robot consists of a Segway Robotic Mobility Platform (RMP) 100[1], supporting an on-board computer and various sensors. No other team used a Segway as its robotic platform. The Segway provides controlled power in a relatively small package. This suits a domestic environment well, for it is small enough to maneuver a living environment built for humans and powerful enough to reliably traverse varying indoor terrain including rugs, power cords, tile, and other uneven surfaces. The large wheels easily navigate small bumps that challenged other indoor robots during the competition.

The two-wheeled, self-balancing robot reaches speeds up to six mph, exerts two horsepower, and has a zero turning radius, freeing it from worry about getting out of tight corners and corridors. The Segway moves with two degrees of freedom, receiving motion commands in the form of forward velocity (m/sec) and angular velocity (radians/sec). It provides proprioceptive feedback in the form of measurements of odometry and pitch. With a payload capacity of 100–150 lbs., the Segway could easily carry several times the weight of its current load.

A 1 GHz Fujitsu tablet PC sits atop the Segway platform, performing all sensory processing, behavior generation, and the generation of motor commands on-board. It interfaces with the Segway via USB at 20 Hz.

[1]http://www.segway.com/rmp/

Two cameras and one laser range finder are available to sense the robot's environment. The Videre Design STOC camera[2] provides depth information, but is not used for the tasks and experiments described in this paper. Higher picture quality is obtained by the second camera, an inexpensive Intel webcam which sends 30 frames per second. The attached Hokuyo URG-04LX[3] is a short range, high resolution laser range finder that is well-suited for indoor environments. It collects 769 readings across 270° at 10 Hz. Also, a Logitech microphone and USB speakers are attached.

The Segway RMP 100 is based on the p-Series Segway line for human transport. Despite its power, the robot is quite safe, featuring safety mechanisms such as automatic shut-off, an emergency kill rope, and speed caps at both the hardware and software levels.

A multi-threaded program, written from scratch, operates the robot. The program's structure can be divided into five modules: the camera input processing, the laser range finder input processing, the motion input/output, speech output, the high-level behavior unit, and the GUI.

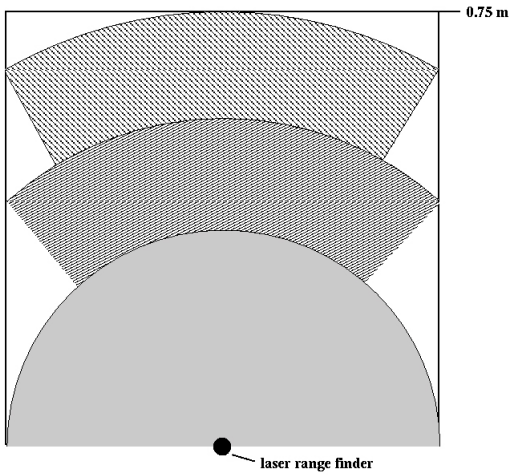

Fig. 7. The laser range finder data is checked for occupancy at three different ranges and angles to approximate a rectangle. The rectangle was a bit wider and deeper than the Segway for safety.

VI. Approach and Performance

This section describes the strategies and algorithms the Segway used in the tasks described in Section V. All tasks were performed in the same home environment (Fig. 1).

A. Lost and Found

In *Lost and Found*, a robot searched for a known object that had been placed in an unknown location in the home environment. The task setup is described in Section II. Our robot competed in the first phase of *Lost and Found*.

1) First Phase: We chose to use an ARTag marker as the target object [14]. ARTag is a system of 2D fiducial markers and vision-based detection. The markers are robustly detected from impressive distances (more than 5 m at 320×240 resolution in our lab with a 20 cm$\times$20 cm marker) with varying light and even partial occlusion. Each marker is mapped to an integer by the provided software library. We did not observe any false positives from our ARTag system.

For the *Lost and Found* task, our robot searched the environment using a reflexive, model-free algorithm that relied on a fusion of range data and camera input. The Segway moved forward until its laser range finder detects an obstacle in its path. It would then look for free space, defined as an unoccupied rectangular section of the laser plane 75 cm deep and a few centimeters wider than the segway, to the left and right and turned until facing the free space. If both sides were free, the robot randomly chose a direction. If neither side was free, it turned to the right until it found free space. Algorithmically, free space was determined by a robustly tuned set of pie pieces in the laser data which overlapped to approximate a rectangle (see Fig. 7).

We placed the object on a table at the opposite end from where the Segway began. A straight line between the two would have passed through a television, shelves, and a kitchen table. The robot had neither prior knowledge of the object's location nor any model of the environment. The Segway successfully completed its search with more than three minutes to spare. Of the six teams that attempted *Lost and Found*, only three teams, including our team, completed it.

B. Follow and Guide a Human

In this task, a robot followed behind a human as he or she walked around the home environment, winding around the furniture. Its setup is described in Section II. The Segway attempted both the first and second phases of the *Follow and Guide a Human* task.

1) First Phase: We attempted only the following (not guiding) portion of this first phase. We did not attempt the extension because time constraints and technical difficulties left the Segway without functional mapping software. (No team finished the extension of returning back to the starting point.) Again, we used an ARTag marker on the shirt of the leading human. The robot flawlessly followed the human leader, without touching furniture or the human. Six of eight teams that attempted the first phase of *Follow and Guide a Human* completed this portion of the task.

2) Second Phase: Without the ARTags of the first phase, the robot instead trained and used a shirt classifier as described in Section IV (Fig. 8). Since we anticipated following a human with his back turned, and thus never return a positive classification result, the face recognition component of our person recognition algorithm was not used. This is an example of the secondary classifier acting as a fall-back classifier, when the primary classifier does not return any result (Fig. 9).

In the competition, the referees chose an African-American volunteer wearing a white shirt. This choice presented two problems that each were sufficient to break our algorithm.

The first problem was that the Viola and Jones' face detection algorithm was unable to detect the human's dark-skinned face. The face detector extracts contrast-based features

[2]http://www.videredesign.com/vision/stereo_products.htm
[3]http://www.hokuyo-aut.jp/02sensor/07scanner/urg.html

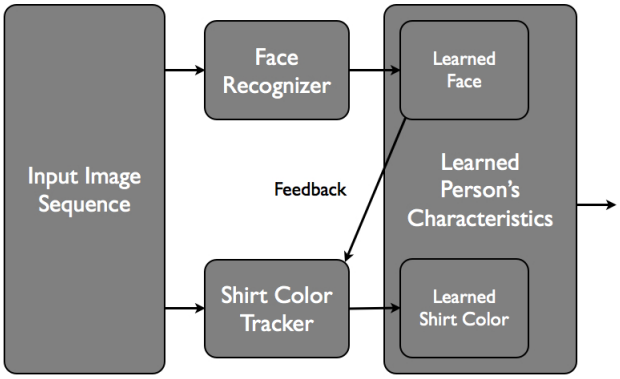

Fig. 8. Learning a person for the *Follow and Guide a Human* task and the *Who Is Who?* task. The face learner indicates the location of the face in the image, and thus the location of the person's shirt. The shirt color tracker can then learn the person's. Note that the state update arrow from Fig. 4 is removed since no motion cue is involved during training.

from a potential face location in the image and uses those features to classify the location as a face or not. However, the lighting was too dark for the detector to capture both bright and dark regions in a dark-skinned face. Without detecting a face, the robot merely waited to see one. We restarted the robot with a LED flashlight attached below the camera to add contrast to the volunteer's face. This time, the face detector managed to locate the face in the video images, and the shirt classifier learned the white shirt.

Tracking the volunteer's shirt was also problematic. His white shirt blended with the background, much of which was white as well. Collecting negative samples helps discriminate between similar colors to some extent, but the shirt and large elements of the background were too alike for the algorithm to handle. Instead of tracking the volunteer's shirt as intended, the robot classified a large portion of the wall as the person and was unable to follow the volunteer.

The choice of volunteer revealed weaknesses in our shirt-following algorithm. However, in the *Open Challenge* and Final rounds, we demonstrated that, given a human leader with light to moderately dark skin and a shirt color that is distinguishable from the background colors, the robot could follow a person for whom it has no a priori data.

C. Who Is Who?

The *Who Is Who?* task tested a mobile robot's ability to meet and later recognize humans. To learn the faces of multiple people, we train a face classifier for each person as described in Section IV. For *Who Is Who?*, the output of the multiple-face classifier is the set of identities which had a number of SIFT feature matches above an empirically determined threshold. If the output set is empty, then the threshold is lowered and the classifier is re-run.

Given the set of candidate identities, a shirt classifier takes over. This classifier gathers samples as described in Section IV, but otherwise the shirt classifier is different, having been modified to eliminate blob selection. Since the face is required for classification in this task, the shirt pixels are simply taken from below the face as in training. For each candidate identity, the Euclidean distance between the average RGB values of the pixels on the persons shirt (a 3-tuple) and the average RGB values of the specific identity's shirt samples is calculated. If at least one candidate's shirt distance is above a shirt threshold, then the candidate with the shortest distance is chosen as the identity of the person. If none are above the shirt threshold, the person is announced as a stranger. This is an example of the secondary classifier being a fall-back classifier in case the primary characteristic based classification result is not confident enough (Fig. 10).

1) First Phase: In the first phase, we chose the four people and their shirts. We gave them strongly distinguishable shirt colors – red, green, blue, and yellow. Our robot correctly identified four of the five people. The stranger was misidentified as one of the known people.

We believe this error occurred specifically on the stranger for two reasons. First, the volunteer's SIFT features matched many features of at least one of the known people. Second, the volunteer's shirt was colored similarly to the person whose SIFT features were similar. With both the primary characteristic (the face) and the secondary characteristic (the shirt) testing as false positives, the person tracker did not correctly classify the stranger.

Of seven teams that attempted this task, some of which used commercial software packages, only one other received points by identifying at least four of the five people.

2) Second Phase: The training of the second phase is the same as in phase one, except the persons were chosen randomly by the committee. The testing is especially more challenging in the second phase. The five people (four known and one stranger) are not standing in a line anymore, but are instead randomly distributed throughout the home environment.

As in the *Lost and Found* task, we used a stochastic search to look for candidate people as recognized by positive identification from the face detection module. During the allotted time, the robot found one of the people and correctly identified him. No other team identified a single person during the second phase.

D. Open Challenge

Once all teams had attempted their specific tasks, each competed in what was called the *Open Challenge*. This consisted of a presentation and free-form demonstration. Going into this event, after receiving scores from the specific tasks, UT Austin Villa ranked third of eleven. A jury of the other team's leaders ranked us second for the *Open Challenge*. The robot's demonstration was a simplified version of the one performed in the Finals, so it will not be described.

E. Finals

The top five teams competed in the Finals. Having ranked third in the specific tasks and second in the open challenge, UT Austin Villa advanced along with Pumas from UNAM in

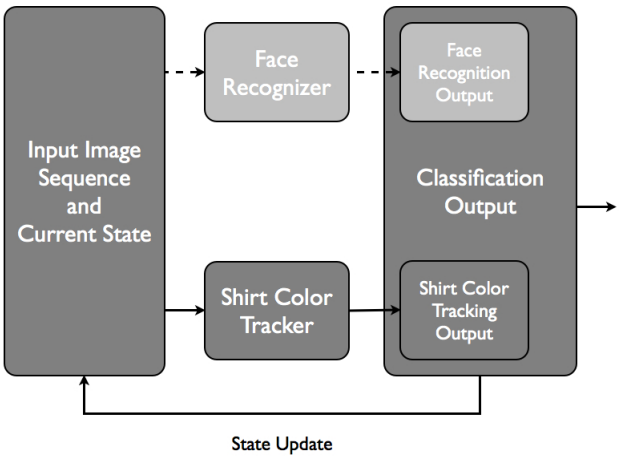

Fig. 9. Tracking a person for the *Follow and Guide a Human* task. The face recognizer is disabled, because the person will never show his face during this phase. The system relies on the shirt tracker only. Otherwise, the scheme is similar to Fig. 4.

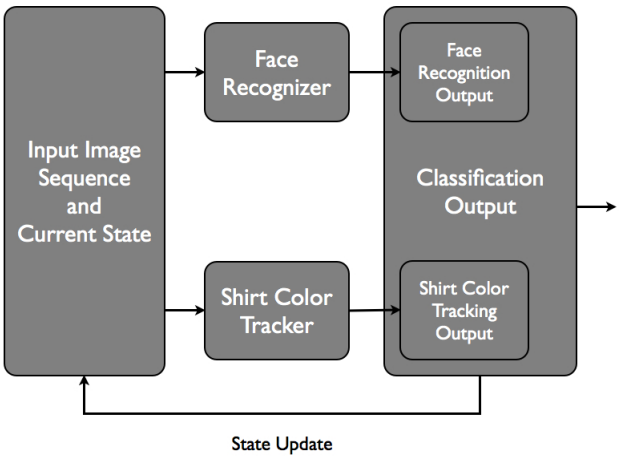

Fig. 10. Recognizing a person for the *Who Is Who?* task. Note that the feedback is disabled, since no re-training is desired. Otherwise, the scheme is similar to Fig. 4.

Mexico, AllemaniACs from RWTH Aachen in Germany, RH2-Y from iAi in Austria, and Robot Cognition Lab from NCRM in France. The Finals were judged by a panel of trustees of the RoboCup organization, all well-known robotics researchers.

Before describing the demonstration itself, we begin with some motivation for the scenario we focused on. Accurate person-recognition will be essential in any fully functional home assistant robot. For instance, if a person refers to himself or another by a pronoun (i.e. "Get *me* my medicine."), the robot needs to know who is being referenced. Asking for identification each time would be cumbersome and unnatural. Instead, the robot should identify the person by context as a human would. This context includes, among other things, visual data, which our algorithm uses.

Person recognition must be robust. Facial recognition alone is not enough, since humans will sometimes be facing away from the robot's camera. Similarly to our previously described algorithm for the *Who Is Who?* person recognition task, we again use shirt color as a secondary classifier. Whereas before it was used to differentiate people after comparing their faces, here we demonstrate using it to identify a person when he turns his back to the robot's camera.

Person recognition, in addition to being robust, must be flexible. Rigidly learning a person's exact appearance at one moment will likely not be sufficient to identify him or her after a significant change in appearance. Changes in human appearance can be roughly categorized into two types. One occurs quickly, like the changing of clothes every day or cutting one's hair. The other type of change occurs very gradually and includes growing older and losing or gaining weight. Although we created an algorithm to handle certain cases of both types, the five minute window of our demonstration limited us to creating a scenario that includes only quick changes.

Our scenario was designed to display our algorithm's robustness and adaptability. Specifically, it shows person identification using shirt color as a secondary classifier in the absence of the primary classifier, the face. It also mimics the daily (or so) occurrence of a human changing clothes, showing the robot adapt to this change in the secondary classifier. Lastly, it shows that the Segway robot can effectively follow a recently learned person without markers, as we unfortunately were unable to show during the second phase of the *Follow and Guide a Human* task. The only differences were that we used a lighter-skinned human and shirt colors which stood out from the colors of the background (as opposed to brown-skinned and white-shirted).

Before the demonstration, we again presented a short talk about the robot and our algorithms. A video of the presentation and demonstration can be found at our team web page[4].

The demonstration involved two people, one with whom the robot intended to interact and another who was unrelated to the robot's primary task (stranger). At the beginning, the robot trains classifiers for the intended person's face and shirt. It then follows the learned person based on only shirt color when face is not visible, first with a green shirt and later with a red shirt. The Segway twice gets "passed" to a stranger, whose back is turned (i.e. face invisible) and is wearing the same shirt color. Each time, it follows the stranger until it can see his face. At that point, the face classifier returns a negative classification and supercedes the shirt classifier, and the robot announces that it has lost the learned person and turns away to look for him. Upon finding the original person based on a positive facial classification, it retrains the person's shirt, subsequently stating whether the shirt color has changed.

In the demonstration, the interaction between the face and shirt classifiers was different than in the *Who Is Who?* task. In that task, the shirt classifier refined the results of the face classifier, choosing from possibly several candidate identities. In this demonstration, however, the shirt classifier worked when the robot did not detect a face in its vision input. Also when both classifiers were running (a face and a shirt are detected) but gave contradicting results, the shirt classifier would re-train using samples obtained from the face classifier.

[4]http://www.cs.utexas.edu/~AustinVilla/?p=athome

Team	Final Score
AllemaniACs	256
UT Austin Villa	**238**
Pumas	217
RH2-Y	199
Robot Cognition Lab	190

Fig. 11. RoboCup@Home 2007 final results

This demonstration shows a full implementation of our scheme as depicted in Fig. 4.

The panel of judges scored the presentations and demonstrations of each finalist, determining each team's final standing in RoboCup@Home 2007. We finished in second place (Fig. 11). Of the top three teams, we had a couple of unique characteristics. Our team size of three people was half that of the next smallest team. We were also the only team in the top three that was competing for the first time. We were very successful as well in the specific tasks in which we competed. We received more points than any other team in the person-recognition task of *Who Is Who?* and accomplished all tasks that we attempted in the first phases of *Lost and Found* and *Follow and Guide a Human*.

VII. Related Work

A variety of home assistant robots have been created in the past decade. Many exhibited impressive specific capabilities. Care-O-bot II [16] brought items to a human user and took them away in a domestic setting. It also functioned as a walking aid, with handles and an interactive motion system that could be controlled directly or given a destination. Earlier systems include HERMES [6] and MOVAID [10].

Person following specifically has received much attention from researchers. A recent laser-based person-tracking method was developed by Gockley et al. [15]. Their robot Grace combined effective following with social interaction. A vision-based approach similar to our own was created by Schlegel et al. [25]. In their system, the robot also tracked shirts using color blobs, but the shirts had to be manually labeled in the training images. Some more recent approaches have used stereo vision and color-based methods to track humans [12], [19].

Person tracking is an extensively researched area in computer vision. Several person tracking systems detecting the number of persons and their positions over time use a combination of foreground/background classification, clustering of novel points, and trajectory estimation [11], [17], [27], [32]. These systems focus on algorithms tracking persons using a stationary camera from a relatively distant, high viewpoint from which most of the people's bodies are consistently visible. In contrast, we consider a camera mounted on a mobile robot that may be moving in close proximity to and often at a lower vantage point than the people in question.

In this setting, the target person's unpredictable movement, the robot's inaccurate motion, obstacles occluding the target, and inconsistent lighting conditions can cause the robot to frequently lose sight of its target. To relocate its target after such out-of-sight situations, the robot must be capable of re-recognizing the person it was tracking. For such person recognition, faces are the most natural identifier, and various studies have been conducted on face recognition [29], [20], [4], [28]. Although these systems achieve reasonably high accuracy with well-aligned faces, they are infeasible for a real-time robotic platform due to heavy computation of face alignment or facial component extraction. Instead of recognition methods relying on careful alignment, we extract SIFT features [22] from faces similar to work proposed in [23], [5] and recognize faces by counting the number of matching SIFT features which is performed in near real-time.

To address the brittleness of tracking faces in light of changing poses and inconsistent lighting, we augment a face classifier with other classifiers, e.g. a shirt classifier. Previous work on integrating multiple classifiers has shown that integrating multiple weak learners ("ensemble methods") can improve classification accuracy [24], and the idea has been extended to multiple reinforcement learning agents giving feedback to each other [9], [18]. In [21], multiple visual detectors (e.g. Grey vs. BackSub) are co-trained [7] on each other to improve classification performance. These methods typically focus on merging classifiers that aim to classify the same target function, possibly using different input features. In contrast, the classifiers we merge are trained on different concepts (e.g. faces vs. shirts) and integrated primarily by associating their target classes with one another in order to provide redundant recognition, as well as to provide dynamically revised training labels to one another. Tracking faces and shirts is a known technique [13], [30], but we express the scheme in general terms and focus on the interaction of the classifiers.

There are various data fusion techniques for detecting objects in the environment. Multi-sensor fusion combines readings of multiple sensor devices to improve accuracy and confidence [8], [31]. In our method, we use one input from a single sensor device that is processed in multiple ways. Techniques such as MCOR combine multiple cues for object recognition in the environment [2]. Unlike their approach of adjusting the weight of each cue, we assign static weights to each classifier, but update the classifiers with additional training data using inter-classifier feedback.

VIII. Conclusion and Future Work

The main contribution of this paper was the complete description of our Segway-based platform that performed successfully in the RoboCup@Home 2007 competition. Leveraging our main technical innovation of using co-training classifiers for different characteristics of a person (face and shirt), it was able to follow a person, distinguish different people, identify them by name, and ultimately combine these abilities into a single robust behavior, adapting to a person changing his or her clothes.

The proposed vision algorithm makes use of the shirt or the face color as a fixed secondary characteristic. We have shown how the system adapts when a secondary classifier fails, if for example the background is similar to the shirt color. However, if people have similar shirts, other vision algorithms need to be considered adaptively. Switching the algorithm online

would be another interesting application of the inter-classifier feedback.

Though the Segway is adept at identifying previously seen humans, it lacks general object recognition capabilities, instead relying on the ARTag system. Future work that gives the robot the ability to learn and later recognize objects other than people would greatly increase its ability to interact within the home environment.

Mapping capabilities will also be necessary on any fully functional domestic robot. One option is Kuipers' Hybrid Spatial Semantic Hierarchy, a system that provides simultaneous localization and mapping, path planning, and an abstraction from its occupancy grid to an idea of places and portals [3]. Other packages are available as well.

ACKNOWLEDGMENT

The authors would like to thank Youngin Shin for his real-time face recognizer and Mohan Sridharan for the robot's initial development. We are also grateful for Selim Erdogan's invaluable help with our robot's operating system and hardware integration. This research is supported in part by NSF CAREER award IIS-0237699 and ONR YIP award N00014-04-1-0545.

REFERENCES

[1] Robocup@home rules & regulations.
[2] S. Aboutalib and M. Veloso. Towards using multiple cues for robust object recognition. In *International Conference on Autonomous Agents and Multiagent Systems*, 2007.
[3] P. Beeson, M. MacMahon, J. Modayil, A. Murarka, B. Kuipers, and B. Stankiewicz. Integrating multiple representations of spatial knowledge for mapping, navigation, and communication. AAAI technical report ss-07-04, The University of Texas at Austin, 2004.
[4] P. Belhumeur, J. Hesphana, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.
[5] M. Bicego, A. Logorio, E. Grosso, and M. Tistarelli. On the use of sift features for face authentication. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop*, 2006.
[6] R. Bischoff. *Design Concept and Realization of the Humanoid Service Robot HERMES*. 1998.
[7] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Conference on Learning Theory*, 1998.
[8] C. Chong, S. Mori, K. Chang, and W. Barker. Architectures and algorithms for track association and fusion. *IEEE Aerospace and Electronic Systems Magazine*, 2000.
[9] R. Crites and A. Barto. Elevator group control using multiple reinforcement learning agents. *Machine Learning*, 1998.
[10] P. Dario, E. Guglielmelli, C. Laschi, and G. Teti. Movaid: A mobile robotic system for residential care to disabled and elderly people. In *MobiNet Symposium*, 1997.
[11] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *International Journal of Computer Vision*, 2000.
[12] V. Enescu, G. Cubber, K. Cauwerts, H. Sahli, E. Demeester, D. Vanhooydonck, and M. Nuttin. Active stereo vision-based mobile robot navigation for person tracking. *Integrated Computer-Aided Engineering*, 2006.
[13] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy – automatic naming of characters in tv video. In *British Machine Vision Conference*, 2006.
[14] M. Fiala. Artag, a fiducial marker system using digital techniques. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.
[15] R. Gockley, J. Forlizzi, and R. Simmons. Natural person-following behavior or social robots. In *Human-Robot Interaction*, 2007.
[16] B. Graf, M. Hans, and R. Schraft. Care-o-bot ii–development of a next generation robotic home assistant. *Autonomous Robots*, 2004.
[17] I. Haritaoglu, D. Harwood, and L. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.
[18] S. Kumar and R. Miikkulainen. Dual reinforcement q-routing: An on-line adaptive routing algorithm. In *Artificial Neural Networks In Engineering*, 1997.
[19] H. Kwon, Y. Yoon, J. Park, and A. Kak. Human-following mobile robot in a distributed intelligent sensor network. In *IEEE International Conference on Robotics and Automation*, 2005.
[20] D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 1999.
[21] A. Levin, P. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. In *IEEE International Conference on Computer Vision*, 2003.
[22] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.
[23] J. Luo, Y. Ma, E. Takikawa, S. Lao, M. Kawade, and B. Lu. Person-specific sift features for face recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2007.
[24] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictors. *Machine Learning*, 1999.
[25] C. Schlegel, J. Illmann, H. Jaberg, M. Schuster, and R. Worz. Vision-based person tracking with a mobile robot. In *British Machine Vision Conference*, 1998.
[26] M. Sridharan and P. Stone. Real-time vision on a mobile robot platform. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.
[27] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.
[28] M. Turk and A. Pentland. Face recognition using eigenfaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1991.
[29] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 2004.
[30] S. Waldherr, R. Romero, and S. Thrun. A gesture based interface for human-robot interaction. *Autonomous Robots*, 2000.
[31] E. Waltz and J. Llinas. *Multisensor Data Fusion*. Artech House, 1990.
[32] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.
[33] T. Zant and T. Wisspeintner. Robocup x: A proposal for a new league where robocup goes real world. In *RoboCup*, 2006.