2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Person Tracking on a Mobile Robot with Heterogeneous Inter-Characteristic Feedback

Juhyun Lee and Peter Stone
Department of Computer Sciences
The University of Texas at Austin
{impjdi,pstone}@cs.utexas.edu

Abstract— **For a mobile robot that interacts with humans such as a home assistant or a tour guide robot, tracking a particular person among multiple persons is a fundamental, yet challenging task. Uniquely identifying characteristics such as a person's face, may not be visible consistently enough to be used as the sole form of identification. Rather, it may be useful to also track more frequently visible, but perhaps less uniquely identifying characteristics such as a person's clothes. After learning various characteristics of a person, the tracking system is required to autonomously update itself with additional training data, since the learned features may change over space and time due to the mobile nature of the robot. In this paper, we introduce a novel algorithm for merging multiple, heterogeneous sub-classifiers designed to track and associate different characteristics of a person being tracked. These heterogeneous classifiers give feedback to each other by identifying additional online training data for one another, thus improving the performance of each classifier and the accuracy of the overall system. Our algorithm has been fully implemented and tested on a Segway base.**

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, one fundamental subtask is the ability to distinguish among individuals. This paper focuses on enabling a mobile robot to track an individual based on input from a video camera, a sensor that is becoming increasingly standard on modern mobile robots.

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 to detect it 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.

This paper introduces a novel algorithm 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.

Our algorithm has been fully implemented and tested on a mobile robot platform based on a Segway Robotic Mobility Platform. The robot was outfitted to participate in the RoboCup@Home competition held in Atlanta during the summer of 2007. In this event focusing on domestic robotics, two out of the six required tasks were related to person recognition and person tracking [1]. Due in large part to the general approach introduced in this paper, our robot finished in second place out of the eleven entries from ten countries.

The remainder of this paper is organized as follows. Section II provides a short overview of related techniques. In Section III, we introduce the concept of heterogeneous inter-characteristic feedback in domain-independent terms. We provide a proof-of-concept with a simple person tracker in Section IV. In Section V, we implement a person tracker with more classifiers for more challenging situations. Section VI presents experimental results illustrating the improved performance of our method over person tracking without inter-characteristic feedback. We summarize and evaluate our work in Section VII.

II. RELATED WORK

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 [2], [3], [4], [5]. These systems focus on algorithms tracking persons using a

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

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 [6], [7], [8], [9]. 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 [10] from faces similar to work proposed in [11], [12] 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 [13], and the idea has been extended to multiple reinforcement learning agents giving feedback to each other [14], [15]. In [16], multiple visual detectors (e.g. Grey vs. BackSub) are co-trained [17] 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 [18], [19], 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 [20], [21]. 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 [22]. 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.

III. CLASSIFICATION WITH HETEROGENEOUS INTER-CHARACTERISTIC FEEDBACK

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.

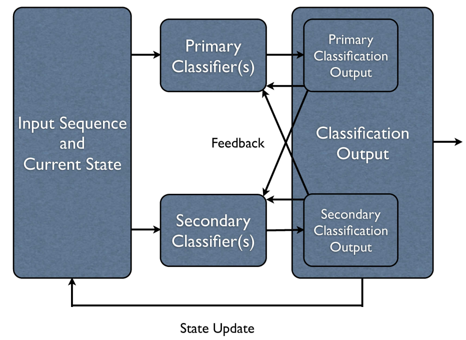

Fig. 1. Classification with heterogeneous inter-characteristic feedback

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. 1 illustrates our scheme.

Algorithm 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 (line 1–2), while the more expensive primary classifier is invoked whenever a more accurate classification is needed (line 3–7). If the condition of taking the branch is carefully chosen, near real-time performance can be achieved by avoiding an expensive classification tasks for the robot every 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

Algorithm 1 Classification with heterogeneous inter-characteristic 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: $\qquad$ **if** $(PriClass \neq SecClass)$ **then**
13: $\qquad\quad$ **if** $(PriClass.Confidence > SecClass.Confidence)$ **then**
14: $\qquad\qquad TrainSecChar(SecChar, Class)$
15: $\qquad\quad$ **else**
16: $\qquad\qquad Class \leftarrow SecClass$
17: $\qquad\qquad TrainPriChar(PriChar, Class)$
18: $\quad$ **else**
19: $\qquad TrainSecChar(SecChar, Class)$
20: **else if** $(SecClass \neq \emptyset)$ **then**
21: $\quad Class \leftarrow SecClass$
22: Update $State$
23: **return** $Class$

classification accuracy and/or $State$. Lines 14, 17, and 19 comprise the inter-characteristic 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 primary classifier alone.

In this paper, we implement a person tracker with this concept, but this scheme is fully general. For instance, as detailed further in Section VII, it could also apply to other domains such as computer networks.

IV. IMPLEMENTATION WITH 2 CLASSIFIERS

Having discussed the general concept of heterogeneous inter-characteristic 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. 1 is implemented for our domain as shown in Fig. 2.

The robot platform used is the Segway RMP equipped with a 1 GHz tablet PC for input/output processing, decision making, and control. The robot's primary sensor used in this paper is a video camera with a limited view of its surroundings ($56°$ horizontally and $45°$ vertically). Images are captured in RGB color space at 30 Hz with a resolution of 640×480 pixels. Multi-resolution images are used for different tasks. The robot remains stationary in this section.

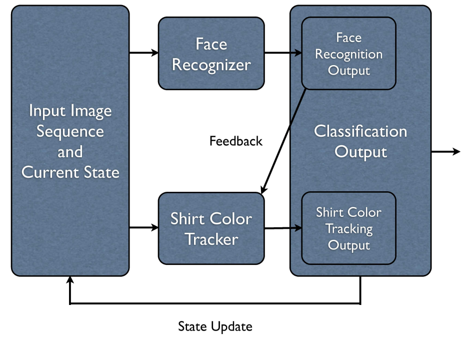

Fig. 2. Person tracking with 1 primary and 1 secondary classifier

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 [6]. 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 [6] does not suffer from a slight resolution decrease. The face recognition algorithm which extracts scale-invariant features (SIFT) [10] 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. 3 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

Fig. 3. 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).

Fig. 4. Example person tracking scenario. In (a) and (b), our algorithm tracks persons with and without the primary characteristic (faces). After leaving the camera's view, the two persons exchange their shirt, and re-enter the scene without showing their faces in (c). The algorithm classifies the persons by just the secondary characteristic. Once the primary characteristic is visible, it updates the secondary characteristic of each person in (d).

objects in the scene. By maintaining positive and negative samples separately, a more accurate RGB-to-person mapping can be generated than 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 [23] 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 took more than 3 seconds on our laptop which takes 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.

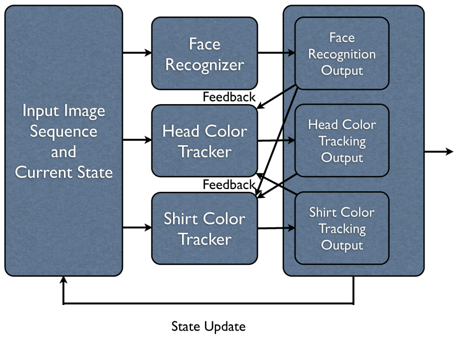

Fig. 5. Person tracking with 1 primary and 2 secondary classifiers

Fig. 6. Example person tracking scenario. Gray, white, and black bounding boxes are showing the primary characteristic (face), and the two secondary characteristics (face color and shirt color), respectively. In (a) and (b), our algorithm tracks a person with and without the primary characteristic, even in blurry images. The shirt color is not a good feature in a similarly colored environment like (c), but the face color classifier provides it with new negative samples, so that no false positives are returned (d).

As an example of autonomous re-training, we show the process of shirt learning and updating. Two persons are wearing a different shirt and train the person tracker. Each person is correctly classified even when the face is not visible to the camera. After the training and the brief testing, they leave the scene and exchange their shirts. When they re-enter the scene without showing their faces, the person tracker does not correctly classify them, but once the primary characteristic is seen, the tracker updates the secondary characteristic associated with each primary characteristic (Fig. 4).

V. IMPLEMENTATION WITH 3 CLASSIFIERS

After our initial proof of concept implementation described in the previous section, we extend our basic classifier with an additional secondary characteristic of a person for more challenging situations when the robot is in motion and the trained classifiers need online updates, e.g. the shirt color and the background color are similar. We add a face color classifier that tracks the skin color and the hair color to track the person when both the face and the shirt color cannot be tracked. Extending our person tracker with a pants color tracker was an alternative, but pants are not visible as often as the person's face unless an additional camera is attached at the waist height. The robot platform remains the same, but it is in motion and follows the person in this experiment.

A. Additional Secondary Characteristic Tracking

The initial training of the face classifier is similar to the shirt classifier. Given the bounding box of the face, the pixels in it are sampled for the face color tracker to learn the face's skin color initially. As shown in Fig. 5, more feedback can be implemented among the classifiers. Since the feedback from the primary classifier to the secondary classifier is shown in Section IV, the only new feedback would be the feedback between the secondary characteristics. For showing the effect of the feedback between these, we assume that the two secondary characteristics have different colors. If the colors are similar, the feedback will not be able to give hints to the classifiers, and just operate without online improvement. Given color blobs of the face or the shirt, a rough estimate of their confidence can be obtained. For example, if the face bounding box is only 20 pixels wide, it is highly unlikely that the shirt bounding box is 320 pixels wide. Various additional state information, such as the motion cues of the characteristics, ($State$ in Algorithm 1) and the relationship between the characteristics (e.g. relative position and size) can issue a re-training.

With an additional secondary characteristic added, we show the update of secondary classifiers with a simple scenario. A person is learned on his face, face color, and shirt color. The person is correctly classified even in a blurry image caused by the person's and the robot's motion. The white color of the shirt learned in a dark environment is not a good feature with a similarly colored background. The shirt classifier returns the highlighted wall as a false positive. The face color tracker with a higher confidence due to its and primary characteristic's motion cues updates the negative samples histogram of itself and the shirt classifier. After the re-calculation of the RGB-to-person mapping, the false positive shirt blob in the background is not detected (Fig. 6). Empirically, we found that updating also the positive histogram based on the ambiguous face color blob is not robust enough.

VI. RESULTS

Upon start-up of the robot, all frames are processed by the face detector, because there is no known shirt in the beginning. While the face is trained, positive and negative face and shirt regions are sampled to build the histograms for the RGB-to-person mapping for the person trackers with additional characteristics. Once this process is finished, the ability to avoid performing face detection every frame can begin to improve the frame rate (Fig. 7 (a)). The person tracker with three characteristics has the lowest frame rate at first and is slightly slower than the one with two characteristics in the long run because it processes the most classifiers, but the extra overhead for the secondary classifiers is small, and is certainly worthwhile if it leads to better accuracy.

Fig. 7 (b) shows the results of a controlled experiment using our technique on a single subject. After having trained the robot, the person stayed in front of the robot for a while till $t = 10$. Then, he turned sideways and started walking. The robot lost track of the person, but then correctly tracked

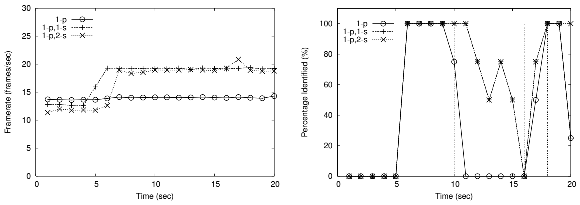

Fig. 7. (a) Frame rate and (b) person classification accuracy of person tracking accumulated at one-second intervals

the subject who was waiting for the robot, facing the camera ($t = 18$). After having been confident that the robot was successfully tracking him, the person turned again ($t = 20$). It can be observed that the face recognizer fails at $t = 10$ when the person stops showing his frontal face, but the person trackers with secondary characteristics still perform well. At $t = 16$, when the person was out of sight, all classifiers correctly drop to zero. Once the person is visible again and facing the camera, all classifiers rise to 100% identified. The person tracker with one characteristic once drops again incorrectly at $t = 20$ when the person turned sideways again. A person tracker with a face recognizer only is very fragile and the results strongly suggest using second characteristics along with the primary characteristic.

Our person tracker was used for our home assistant robot at the 2007 RoboCup@Home competition in Atlanta. During the competition, our robot successfully displayed the abilities to follow a human and to learn to distinguish among several people. Because competitions such as RoboCup@Home typically do not allow for enough task repetitions to obtain conclusive scientific results, they can only provide anecdotal evidence of a complete system's success. For this reason, we focus on controlled, comparative experiments in this paper. Nonetheless, our second place finish in the competition suggests that our algorithm presented in this paper may be useful in practice.

VII. CONCLUSIONS AND FUTURE WORK

Robots that interact with humans need to understand human motion. One of the basic tasks that has to be accomplished is tracking a certain person to which the robot has given its attention. Various research has been conducted for tracking a person which can be a difficult task for real-time robotics vision.

In this paper, we have described a method to robustly track a person among multiple different persons by using independent characteristics of a person: the face, the face color, and the shirt color. By combining these characteristics and allowing them to give feedback to each other, the person tracking algorithm exhibits a more stable detection rate than a system without such feedback between the features. The interaction between the three characteristics improves the frame rate, the detection rate, and the classification accuracy of the robotic vision, which ultimately leads to a more reactive and correctly-behaving robot.

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-characteristic feedback.

As mentioned in Section III, our heterogeneous inter-characteristic feedback system may also apply to other domains such as computer networks. For a worm detection system, the worm signature can be considered the primary characteristic for the classification problem. Since the regular expression matching of worm signatures is a heavy operation, a lighter classifier using simple heuristics for detecting anomalies in the network traffic can be used as the secondary characteristic for worm classification.

ACKNOWLEDGMENTS

This research is supported in part by NSF CAREER award IIS-0237699 and ONR YIP award N00014-04-1-0545. The authors gratefully acknowledge Youngin Shin for his face recognizer, W. Bradley Knox and Mohan Sridharan for their contribution to our RoboCup@Home project, and Peter Djeu for his helpful comments.

REFERENCES

[1] "Robocup@home rules & regulations." [Online]. Available: http://www.ai.rug.nl/robocupathome/documents/rulebook.pdf
[2] T. Darrell, G. Gordon, M. Harville, and J. Woodfill, "Integrated person tracking using stereo, color, and pattern detection," *IJCV*, 2000.
[3] I. Haritaoglu, D. Harwood, and L. Davis, "W4: Real-time surveillance of people and their activities," *PAMI*, 2000.
[4] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *CVPR*, 1999.
[5] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *PAMI*, 1997.
[6] P. Viola and M. Jones, "Robust real-time face detection," *IJCV*, 2004.
[7] D. Lee and H. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, 1999.
[8] P. Belhumeur, J. Hesphana, and D. Kriegman, "Eigenfaces vs. fisher-faces: Recognition using class specific linear projection," *PAMI*, 1997.
[9] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *CVPR*, 1991.
[10] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, 2004.
[11] J. Luo, Y. Ma, E. Takikawa, S. Lao, M. Kawade, and B. Lu, "Person-specific sift features for face recognition," in *ICASSP*, 2007.
[12] M. Bicego, A. Logorio, E. Grosso, and M. Tistarelli, "On the use of sift features for face authentication," in *CVPRW*, 2006.
[13] R. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictors," *Machine Learning*, 1999.
[14] R. Crites and A. Barto, "Elevator group control using multiple reinforcement learning agents," *Machine Learning*, 1998.
[15] S. Kumar and R. Miikkulainen, "Dual reinforcement q-routing: An on-line adaptive routing algorithm," in *ANNIE*, 1997.
[16] A. Levin, P. Viola, and Y. Freund, "Unsupervised improvement of visual detectors using co-training," in *ICCV*, 2003.
[17] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *COLT*, 1998.
[18] M. Everingham, J. Sivic, and A. Zisserman, "Hello! my name is... buffy – automatic naming of characters in tv video," in *BMVC*, 2006.
[19] S. Waldherr, R. Romero, and S. Thrun, "A gesture based interface for human-robot interaction," *Autonomous Robots*, 2000.
[20] C. Chong, S. Mori, K. Chang, and W. Barker, "Architectures and algorithms for track association and fusion," *AES*, 2000.
[21] E. Waltz and J. Llinas, *Multisensor Data Fusion*. Artech House, 1990.
[22] S. Aboutalib and M. Veloso, "Towards using multiple cues for robust object recognition," in *AAMAS*, 2007.
[23] M. Sridharan and P. Stone, "Real-time vision on a mobile robot platform," in *IROS*, 2005.