Narrative-level Visual Interpretation of Human Motion for Human-robot Interaction

Adrian Hilti[1], Illah Nourbakhsh[2], Björn Jensen[1], Roland Siegwart[1]

[1]Institut de Systèmes Robotiques
Ecole Polytechnique Fédérale de Lausanne
CH-1015 Lausanne EPFL, Switzerland

[2]The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213, USA

Abstract

Compelling human-robot interaction demands high level perception of human behavior by the robot. In this paper we describe a visual perception system that provides high-level, narrative interpretation of human behavior in relation to the robot. The vision system has been implemented using Firewire digital camera technology and has been tested in public venues at The Robotics Institute.

1. Introduction

Robot-human interaction is quickly becoming a serious research question because robot technologies have matured. Nursebots, Tourguide or Toy robots have been developed with the aim to interact with humans [TBB99] [NBG99]. But robots primarily depend upon ranging sensors that have little bandwidth and acuity in terms of three-dimensional motion and shape detection. As a result, the robots cannot easily interpret the behavior and intentions of the humans around them.

Recent advances in vision systems and the gain in computation power of today's computers allow making robots "see" their environment with a new level of high-level cognition. We propose that contemporary vision techniques be used to create a narrative interpreter that senses human motion around the robot visually and provides high-level perceptual output regarding human behavior.
Despite recent advances, the solution is not straightforward. Many vision techniques work well in a controlled laboratory environment; however, for this project we require high-level visual cognition that is robust to real-world dynamics (e.g. motion, illumination) as well as the wide variety of expected human behavior.

2. Approach

To obtain a high level interpretation of human motion in a video stream one must first detect humans. There are a variety of approaches to human detection, primarily focusing on face detection and recognition. The main approaches use color [MAG99] [Bra98] and also a mixture of stereo, color and pattern detection [DGH98]. Other techniques use neuronal networks [RBK95] or wavelet transforms [Sch99].

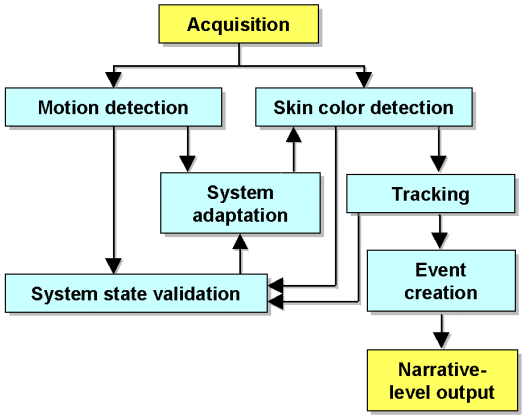

Fig. 1: System overview

In our system (figure 1), both pixel-based skin color segmentation and motion-based segmentation are used for human detection. Advantages of using skin color are that it is orientation invariant and one of the faster facial feature detection methods. It is therefore suitable for real time systems.

The human face normally presents a large skin surface in a flesh-tone. This tone is quite similar from human to human and even across various races [MAG99]. Using hue and saturation (HS) as inputs, the method transforms a color image into a filtered image, wherein each pixel is associated with a likelihood of being due to skin color [Poy96].

Although this method is stable under even illumination, changing lighting conditions have significant impact on the overall hue and saturation values of skin. Therefore, our algorithm for color-based segmentation implements continuous adaptation to compensate for exogenous changes. Both motion-based segmentation and the skin color-based segmentation methods provide feedback for adaptation using an algorithm that will be described in detail.

Identification of each individual human face is performed using a contour finding algorithm combined with image erosion and dilation techniques. Finally, a filtering step eliminates candidate faces that fail basic shape and position tests.

Following human identification, our method generates narrative-level outputs by tracking and interpreting the motions of humans over time. A trajectory-tracking algorithm translates long-term motions of the humans into high-level perceptual events ready for consumption by the robot. System outputs include unique identification of each successive human, followed by interpretation of the human's motions (e.g. approaching robot, stopping, gross motion, departing).

3. Implementation

The implementation can be described in terms of four main functional modules: human candidate detection, target tracking, parameter adaptation and narrative event creation.

3.1 Human candidate detection using skin color

For a given illumination, the skin reflectance of humans lies in a close hue saturation range even for different races [MAG99]. We transform the input RGB image into the hue saturation color representation. Hue and saturation are significantly more stable to changes in light intensity as compared to the RGB bands. But changing the camera parameters (white balance, hue, saturation) or external illumination will shift the skin color region considerably in hue-saturation space. Using appropriate adaptation techniques described below combined with motion detection for initial adaptation will help overcome these challenges.

Skin color probability
The given RGB input image is transformed into its hue saturation representation [IPL00]. Every pixel having its hue saturation value inside a certain range ("*color limitation window*") will be marked as 1, otherwise as 0, in a new binary skin color filtered image. This range has been chosen using first an initialization step and then continuous adaptation as described below.

Before segmentation of the image, the morphologic close function is applied to the raw binary image [IPL00]. This function eliminates small speckles and smooth shape borders, speeding up the contour-finding process that follows.

Segmentation - Contour finding in binary images *We use an algorithm described in [CV00]. It passes one time through the binary skin color filtered image and returns a sequence of contours represented by a chain of points and.*

Contour filtering
Evaluating false face candidates is done using the following filters that can be enabled or disabled as needed:

- **Position.** Shapes outside a clipping rectangle (e.g. lower body parts) are rejected.
- **Area.** Only faces within a specified range of pixel area are accepted.
- **Aspect ratio.** Relation between height and width of the bounding box. Typical values for faces are between 0.5 and 2.

The skin color contours pass through all three filters, whereas the motion contours only pass through the area and position filter. No aspect ratio filtering is used for motion, because person's motion silhouette can represent the whole body (described in "*Image differencing*").
The next step will be the tracking of identified skin color regions over time.

Fig. 2 Before skin color filtering and afterwards.

3.2 Tracking

Shape to person assignment algorithm. We instantiated a person object if a contour remains after filtering. Every person can only have one new contour attributed per frame. Tracking over time is realized using a scoring system. Every skin color contour in a new frame (that passed the filtering) will be scored in respect to present "persons" (assigned contours in last frames). The scoring is distance based.

Attributing a new contour to an existing person requires the following conditions fulfilled: the distance of the contour bounding box center to the person center has not to exceed a maximal distance and the contour has not to be attributed yet. After scoring every contour to a person into a table, all persons get their best-scored shape attributed.

Persons that do not get a shape attributed in this frame should not be deleted immediately. They could be detected again in one of the following frames. For solving this problem, the timeout concept has been introduced.

Timeout counter. Every person gets at its creation a counter set to 255. This value will be reduced in every new frame for all persons. If the timeout value of a given person goes below a threshold value, the person will be deleted. This guarantees clean up. On the other hand,

every successful assignment to a new contour will be rewarded with an increase of this timeout.

3.3 Adaptation

Starting in a new environment with new illumination and camera parameters requires *de novo* initialization of the system. This is a difficult task, because the skin color range in the hue – saturation representation cannot easily be known and must be measured. Our approach takes advantage of the secondary characteristics of human behavior, namely motion, to provide the feedback required for auto-initialization. Combined with continuous parameter adaptation based on the tracking of human targets, this results in a system that requires a minimum of human effort in order to robustly track humans in varying conditions.

Image differencing. The input image is converted to gray scale and image-closing algorithms [IPL00] are applied in order to eliminate small speckles and smooth the boundaries of objects without significantly changing their area.

This also decreases the calculation time for the following contour finding algorithm. The subtraction of the previous motion probability image produces a motion silhouette image. This image shows where motion occurred. At high processing frame rates, the silhouette image shows very thin silhouettes that makes further skin color searching difficult. To avoid this problem, we create a motion history image.

We use a function that stores for every white input pixel of the silhouette image the system time in a motion history image (cvUpdateMHIByTime function [CV00]). This way, we accumulate new silhouette images in one image. Experiments show that accumulating during 200ms, corresponding to approximately 3 frames at 15 frames per second acquisition rate provides acceptable results. Figure 3 depicts a sample motion history image of a person passing-by.

Fig. 3: Motion history image

The contour finding algorithm is applied to this image providing motion contours (cvFindContours function [CV00]). The contours bounding box areas are calculated and filtering is applied using pixel area as criteria. Motion areas being to small for adaptation, for example a person passing-by in the background, are filtered. Motion areas bigger than half of the screen are probably also false candidates and therefore filtered.

Initial skin color searching algorithm. The found filtered motion contours will help narrowing down the search for skin color regions.

We assume that a possible skin color region (the face) will be found in the upper part of a given motion contour. Therefore, we choose the topmost motion contour (1) seen in Figure 2. The rectangles (blocks) at (2) are tested one after another until a skin color evaluation criteria is accomplished. Otherwise the search will restart in the next image frame. The evaluation process is described next.

HS adaptation to rectangular region. For every new frame, a HS image is calculated, giving access to the HS values at a given coordinate of the image. For a given block (rectangle) the mean hue and saturation value of all HS pixels is calculated, the variance as well. If the variance of the HS values falls below a certain limit, adaptation will take place. Adaptation involves limiting hue and saturation values to a range that is limited based on the current, empirical measurements. We also refer to this range as the "*Color limitation window*". This technique allows an initialization of the color-filtering algorithm. Once this initial color window has been automatically registered, continuous adaptation can generally handle small and gradual variations in illumination as seen in real-world venues (e.g. motion of clouds, time of day). Tests demonstrate that this algorithm is primarily suited for persons standing in front of the camera and performs quite well. But a person passing-by is detected poorly, as described in section 4.

Continuous adaptation. Once the system is initialized, large-scale adaptation of the color window center position (mean HS values) is generally not required.

If a person is detected over a time of typically 10 seconds, adaptation is done for the person if its center speed (the bounding box center of the skin color contour) is below a certain value (figure 8). This guarantees that the person is not moving too fast resulting in a false adaptation to image background.

Once again we try to adapt to the face center. Knowing the person's skin color bounding box, a region inside this box is chosen reducing size by a factor of three.

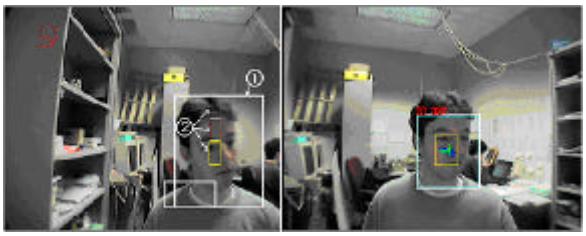

Fig. 4: Motion-based initialization. Continuous adaptation region

The adaptation only takes place if the variance criteria (as described in *"HS adaptation to rectangular region*") is accomplished, the person center speed is below a limit and the drift limitation is valid, as described next.

Mean HS Value Drift Limitation. Another implemented limitation to prevent false adaptation is done using a mean HS value drift limit. For every continuous adaptation attempted, the distance of the new calculated mean HS values and the old values are calculated. If this distance is bigger than a maximal value, no adaptation will occur. We only want to adapt to small changes of the mean skin color HS values.

The drift limitation is disabled for the initialization algorithm because large changes can be necessary.

Adaptation need evaluation (system state scoring)
An important aspect of this perception system, as it is intended for use in practical, real-world settings, is the ability to perform some level of self-diagnosis. Along these lines, an important first step is the ability of the system to sense its own adaptation failure. To detect poor system state, we have devised scoring functions based on expected scene interpretation and human behavior. Below are several techniques used to confirm the functioning of the end-to-end perception system.

Motion Without Detected Humans
If motion is detected but no skin color is present the system is probably not adjusted correctly, no people are detected.

In every frame, we know the number of detected motion contours and skin color contours (after filtering). We use two counters, a motion and color contour counter. These counters will be increased by a certain value for every frame if motion and skin color are present in the actual frame and decrease if not. If the motion counter comes to a limit and the skin color counter is zero re-initialization occurs. If instead skin color is detected and no motion the system also needs re-initialization because people generally move eventually.

Contour Errors
The presence of a contour that covers more than the half of the screen over a long time (20 seconds) is likely due to wrong system adaptation. Probably the system adapted to the background.

A counter is used that increases in every frame that presents such a big contour. If the counter reaches a limit value, new motion-based initialization is triggered.

Human Acquisition and Loss Thrashing
If the number of new assigned persons per frame is high (more than 5 persons created) and the number of deleted persons is also high (more than 5 persons removed), the system is probably in an unstable state. The adaptation parameters are borderline correct and motion-based re-initialization is activated to change system parameters.

False positives
The average traveled distance measured for all present persons during the last 5 seconds is a good indication of correct system state. If this number is low, there are non-moving inanimate objects probably being mistaken for humans. When this value drops below a lower limit in a given frame a counter will be increased. It will be decreased if not. Reaching a limit, motion-based initialization takes place.

Number of Humans Detected
The optics and algorithm impose a geometric limit on the number of humans that can be visible to the camera system at a sufficiently close distance to be correctly tracked. Based on these physical limits, if the number of detected humans is larger than 6 for more than 20 seconds, the system parameters are likely to be incorrect. Again, we use a counter that increases successively and, upon reaching an arbitrary threshold, triggers motion-based initialization.

3.4 Event creation
The creation of narrative-level events is tightly linked to the tracking process. Every event is associated with a person. This enables annotating a person with an event history. Every person's three-dimensional tracked path is also recorded, useful for further processing, analysis or visualization. Each person's trajectory along the visual plane is computed by simply tracking the center of mass of the respective contour along the image. The person's distance from the camera is estimated based on the area of the face contour. Although imprecise as an absolute measure of distance, this is an effective way to measure change in distance.

Entering and exiting. At every creation of a new person, its center position will be checked for left side, right side or center of the screen, leading to the corresponding events EV_ENTERS_RIGHT, EV_ENTERS_LEFT and EV_APPEARED.

The Remove function described earlier generates one of three events based on the last known location of the person: EV_EXITS_LEFT, EV_EXITS_RIGHT and EV_LOST. The geometric limits discriminate these events are set manually.

Approaching and distancing. As soon as a new created person stops on screen the person's contour area value is stored. After every successful contour assignment, the actual contour area will be compared to the stored contour area value. If the person approaches for example and the ratio between initial and present contour area grows by more than a given factor, the EV_APPROACHES event is triggered. For example, a human standing still two meters away will trigger an

EV_APPROACHES event after moving forward approximately 40 centimeters.

The same mechanism is used for the EV_DISTANCES event. These events set the stored contour area value to the new actual area value allowing to have successive approaching or distancing events.

If a person contour area gets bigger than a factor times the whole screen area, EV_VERYCLOSE is created. This event is generated when the human is approximately 30 cm from the camera lens.

Person information

At each cycle, the system communicates all events that have been triggered. In addition, for each person being tracked, the system reports:

ID	Traveled horizontal distance
Position	Center of mass
Area (z-value)	Main axis
Speed	Main axis rotation angle
Motion track history	Time on screen

Table 1: Person information

This additional information will be extremely useful to an active mobile robot in, for example, enabling the robot to direct its gaze and gestures in the direction of a specific person or group of people.

3.5 Processing speed

Hardware. We use a Sony DFW-VL500 Digital Firewire camera for image acquisition combined with a Ratoc Firewire PCMCIA card and a Dell Inspiron 5000e Laptop running Win98 or Win2000 at 600Mhz. Image acquisition is done using drivers developed at CMU [UN00].

Software. The system is based on the Intel Image Processing Library (IPL), the Intel Open Source Computer Vision Library (OpenCV) and written in Microsoft Visual C++ 6 SP4. IPL provides Intel Processor optimized low level image buffer management and processing functions. OpenCV is used for contour finding and motion history functions. System feedback is provided by speech output using the Microsoft Speech SDK 5 Text-to-speech engine. Graphical 3 dimensional output is created using OpenGL.

Speed improvements. Fast processing time is a key element for successful tracking. Processing will be done on a downscaled image that reduces the amount of calculation drastically. The initial image after acquisition at 640x480 pixels is reduced by the factor 5 to 128x96 pixels. The present computer has a processing time average of 15ms per frame (without acquisition). Acquisition and processing takes around 67ms corresponding to 15fps. This frame rate includes all image filtering, candidate person identification, person tracking and narrative generation.

4. Experiments & Analysis

The system was implemented and tested in unmodified sections of the Robotics Institute at Carnegie Mellon University (Pittsburgh, PA). In the first experiment, the system was mounted on a tripod in a hallway, perpendicular to the hallway's main direction of travel. The goal was to evaluate the system's ability to recognize human passers-by using motion-based initialization followed by adaptation. Ideally, the system would generate narrative describing persons as entering from one side and exiting to the other side.

The performance of the visual interpretation system was only acceptable when the "color limitation window" parameters were adjusted manually, and furthermore when the camera hue parameters were adjusted to compensate for a washed-out blue channel due to sunlight. The motion-based initialization technique was impractical because the face region in the motion profile of a human passing-by is situated in the motion contour front part. An improvement of our algorithm that may overcome this limitation would be to calculate the motion vector of each moving object, then search for skin color regions in the appropriate subsection of the moving object (i.e. the leading edge rather than the trailing edge).

Human Frontal View. In the second real-world experiment, we intended to demonstrate the narrative generation system in a suitable venue that would include sufficient frontal face views. Our chosen venue was a monthly Robotics Institute Buffet. The camera was mounted across from the buffet table, with a clear view of attendees.

For a total of 35 minutes, the visual interpretation system autonomously identified and tracked humans, generating meaningful narrative level summaries of the human's behavior. Figure 5 shows an image during this test in which two tracked persons have been identified and assigned unique ID numbers. The trajectory of each person is shown, as well as full-scale horizontal lines indicating the region that is being searched for humans.

Continuous adaptation was performed continuously on the person presenting the largest contour, and this adaptation was successful both in ensuring smooth tracking of this "principal" person and in adjusting for exogenous illumination changes that may have affected camera characteristics such as white balance.

The system is robust to multiple moving persons, as shown in Figure 5, where person 2 passes behind person 1. Although person 2 is lost, person 1, the largest contour and therefore closest person is tracked successfully.

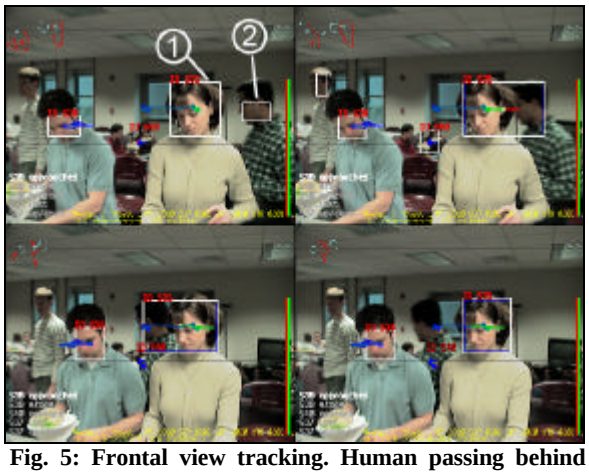

Fig. 5: Frontal view tracking. Human passing behind tracked person.

During the 35 minute test, a total of 2,027 events were generated. 527 candidate persons were identified. Of these candidate, 90 were tracked for longer than 10 seconds, in every case a correct identification of a person. These 90 long-term persons were associated with 894 events, yielding an average of 9.9 events per candidate.

Events were generated at a frequency appropriate for social robot interaction. During these 35 minutes, the average event frequency was 0.96 Hz. Of course, arbitrarily set thresholds such as distance-travelled have a significant impact on event frequency (e.g. EV_APPROACHES). Table 2 below provides an excerpt of the event log generated during the Buffet. Further examples of generated narrative and accompanying video are available at [AHwww].

appears	483	stops	485
stops	483	approaches	485
enters left	484	moves	485
distances	483	stops	485
exits right	483	left/right	485
exits left	484	up/down	485
enters left	485	exits right	485

Table 2: Event log excerpt

5. Conclusion

We have implemented a visual interpretation system that provides high-level narrative interpretation of human motion. Skin color segmentation and motion-based segmentation are used for detection and tracking of humans. A novel aspect of the system is its heuristic set of system validation measures, coupled together with a re-initialization method and a continuous adaptation technique. Together, these competencies enable the system to recover from poor system state, robustly continuing to perform high level interpretation in changing conditions. The system has been tested in public venues at The Robotics Institute, and has been shown to perform well in the case of frontal views.

Future research issues include the integration of a web server allowing remote observation and the use of USB cameras for image acquisition. Further, the system will be implemented on real robot systems to test direct human-robot interaction.

Acknowledgements

Thanks go to Iwan Ulrich (CMU), who developed the drivers for the Firewire camera, Jean-Christophe Zufferey and Simon Sulser who provided help in programming questions. Thanks also go to The Robotics Institute, the Robot Educational Lab and Ben Wegbreit for equipment and support.

References

[AHwww] http://iaestepc4.epfl.ch/Diploma/Project.html, contains documentation, source code and demonstration video sequences of the system in action

[NBG99] I. Nourbakhsh, J. Bobenage, S. Grange, R. Lutz, R. Meyer, A. Soto. *An Affective Mobile Educator with a Full-time Job.* Artificial Intelligence, 114 (1-2), pp. 95-124. October 1999

[RBK95] H.A. Rowley, S. Baluja, T. Kanade. *Neuronal Network-Based Face Detection,* Carnegie Mellon University, 1995

[Sch99] H.W. Schneidermann. *A Statistical Approach to 3D Object Detection Applied to Faces and Cars.* CMU, 1998

[DGH98] T. Darrell, G.G. Gordon, M. Harville, J. Woodfill. *Integrated person tracking using stereo, color, and pattern detection.* CVPR, 1998

[UN00] I. Ulrich, I. Nourbakhsh. *Firewire Untethered: High-Quality Images for Notebook Computers.* Advanced Imaging Magazine, pp. 69-70, 2000

[MAG99] M. Störring, H.J. Andersen, E. Granum. *Skin colour detection under changing lighting conditions.* 7th Symposium on Intelligent Robotics Systems, 1999

[Poy96] C. A. Poynton. *A Technical Introduction to Digital Video.* New York: Wiley, 1996.

[TBB99] S. Thrun, M. Bennewitz, W. Burgard, *MINERVA: A Second-Generation Museum Tour-Guide Robot,* CMU, 1999

[IPL00] *Intel Image Processing Library Reference Manual,* Intel, 2000

[CV00] *Open Source Computer Vision Library Reference Manual*, Intel, 2000

[Bra98] G.R. Bradski. *Computer Vision Face Tracking For Use in a Perceptual User Interface*, Microcomputer Research Lab, Intel Corporation, 1998

[Fin98] G.D. Finlayson et al. "Comprehensive Colour Image Normalization", Proc. ECCV'98 Fifth European Conference on Computer Vision, Volume I, pp. 475-490, Springer, 1998

[Yan98] J. Yang et al. "Skin Color Modeling and Adaptation", Computer Vision-ACCV'98, Proc. of the Third Asian Conference on Computer Vision, Vol. 2, pp. 687-694