Automatic Person Detection and Tracking using Fuzzy Controlled Active Cameras

Keni Bernardin, Florian van de Camp, Rainer Stiefelhagen
Institut für Theoretische Informatik
Interactive Systems Lab
Universität Karlsruhe, 76131 Karlsruhe, Germany
keni@ira.uka.de, fcamp@ira.uka.de, stiefel@ira.uka.de

Abstract

This paper presents an automatic system for the monitoring of indoor environments using pan-tilt-zoomable cameras. A combination of Haar-feature classifier-based detection and color histogram filtering is used to achieve reliable initialization of person tracks even in the presence of camera movement. A combination of adaptive color and KLT feature trackers for face and upper body allows for robust tracking and track recovery in the presence of occlusion or interference. The continuous recomputation of camera parameters, coupled with a fuzzy controlling scheme allow for smooth tracking of moving targets as well as acquisition of stable facial closeups, similar to the natural behavior of a human cameraman. The system is tested on a series of natural indoor monitoring scenarios and shows a high degree of naturalness, flexibility and robustness.

1. Introduction and Related Work

The tracking of persons with steerable cameras is an active research field with applications in many domains. These range from video surveillance, over automatic indexing, to intelligent interactive environments. In all cases, robust person tracking and the acquisition of high resolution images of target objects can serve as a poweful building block to support other techniques, such as gesture recognizers, face identifiers, head pose estimators, scene analysis tools, etc. In the last few years, more and more approaches have been presented to tackle the problems posed by unconstrained, natural environments and bring automatic camera tracking technology out of the laboratory environment and into real world scenarios.

Tsuruoka et al. [10] used a steerable camera to make more appealing recordings of presentations available to remote viewers. In addition to the active camera, which is used only for recording, a static camera is used for all image processing, target position computation, and control command generation. A fuzzy controller is used to steer the active camera in accordance with the observed situation. In contrast, the here presented system allows a much higher degree of freedom of the user and requires no additional fixed camera to estimate the target position.

A whole different level of freedom is reached by Castro et al. [11], which use fuzzy logic for target tracking with a mobile robot. Although the concepts for the fuzzy steering of a robot can not be directly mapped to a steerable camera, there are many similarities in the design of the controller. Cuevas et al. [12] realize a combination of both by setting up a steerable camera on a mobile robot. As opposed to the here presented system, their approach makes no use of the camera zoom, as the robot itself is expected to move closer to objects.

Perhaps the work that most closely relates to ours is that of Hampapur et al. [13]. Their system, which is designed for wide area surveillance, uses active cameras to make unaware close-up recordings of persons in a predefined space. Their approach does, however, rely on a system of fixed stereo cameras to deliver 3D person tracks, with which the active cameras are steered.

The here presented system differs from the aforementioned in its complete use of all camera degrees of freedom, as well as in the autonomous nature of its function: It does not rely on outside information to keep track of targets, but does this solely based on the information in the active camera image.

For the design of our active camera person tracking system, the following criteria were observed:

- It should be able to operate in realtime, with reaction times short enough to keep pace with the uncooperative, natural movement of tracked subjects.
- It should provide for smooth and natural camera motion, as a professional cameraman would.

1-4244-1180-7/07/$25.00 ©2007 IEEE

- It should be able to keep track of the target in natural indoor environments, with uneven lighting conditions and cluttered backgrounds, robustly handle occlusions and recover from tracking failures automatically.
- It should be able to interact with an external target selection system to switch the focus between persons, without relying on the external system for continuous or accurate position information.
- It should automatically and independently detect targets in the field of view of its active camera, and steer the camera based on this information.

Additionally, we expect the system

- to use only off-the-shelf cameras and hardware.
- to be usable in a wide range of environments and scenarios, without the need for major tuning or retraining.

The remainder of the paper is organized as follows: Section 2 presents the automatic detection, tracking, and tracker fusion techniques. Section 3 gives details about the camera calibration procedure and the fuzzy controlling scheme. In Section 4, the performance of the camera tracking system is demonstrated on a series of experimental scenarios and the usefulness of the approach is discussed. Finally, Section 5 gives a summary and a conclusion.

2. Person Detection and Tracking in Moving Camera Images

As was stated above, the goal in our approach is to acquire a human target in the field of view of the camera automatically and to keep tracking the target using only the cues available in the camera image. Since the camera itself is constantly changing its orientation, object initialization and continuous tracking pose great challenges. Standard techniques such as foreground segmentation, motion detection or optical flow are not applicable as the background is constantly moving. Although some recent off-the-shelf camera models allow to obtain internal information about orientation and speed directly through their control interface, this information is usually not fast or reliable enough to compensate for motion-enduced errors in frame-level processing. Reliable edge detection is also difficult, and predefined color models, such as skin color, are impractical as the lighting conditions and the color signature of the scene change substantially with camera orientation.

The here presented system overcomes these problems by using motion and color invariant detectors for frontal faces in each camera frame to initialize and continuously update person models. It also uses a combination of trackers, relying both on color and on edge features to maintain correct tracks. In the following, the different system components are explained in detail.

2.1. Track Initialization

For the detection of frontal faces in the camera views, boosted cascades of classifiers based on Haar-like features, as decribed in [1, 2], are used. The image is continuously scanned using various detection window sizes and bounding rectangles for likely face candidates are obtained. These candidates are then filtered using color information, as described in the next subsection, to eliminate false positives. Once a person was detected, a track is intialized and only the area surrounding the track is scanned in subsequent frames. After a track is lost, scanning is performed again on the entire image.

The classifier cascades are also used to continuously update the color models for the tracked person, as they deliver reliable information about the face region. Whenever a face detection shows sufficient overlap with the tracking window, an update is made. Although adaptation is therefore only performed when the subject faces the camera, this strategy helps to avoid color model degradation, which occurs when wrongfully learning in non-person colors.

For our implementation, the frontal face classifier cascades were taken from the OpenCV [15] library. They allow for fast processing and high recall rates, as required for realtime applications such as ours.

2.2. Color Feature Tracking

2.2.1 Face Tracking

When a subject's frontal face is first detected in the initialization phase, the image pixels inside the detection window are used to build a color histogram of the target region. The analysis is made in HSV space, and the color values are sampled from a subwindow half the size of the original detection, to avoid training in background colors. In addition to the face histogram H, a background histogram H_{neg} is built to model non-face colors, using the pixels from the entire image.

After normalization, $H(x)$ can be seen as modeling $P(x|Face)$, and $H_{neg}(x)$ as modeling $P(x|\neg Face)$, for a given pixel x. By applying Bayes' rule, we can obtain the likelihood ratio of a pixel belonging to the face as

$$\frac{P(Face|x)}{P(\neg Face|x)} = \frac{P(x|Face)}{P(x|\neg Face)} * \frac{P(Face)}{P(\neg Face)} \sim \frac{H(x)}{H_{neg}(x)}. \tag{1}$$

For ease of representation, we directly compute the histogram $H_{filt} = H/H_{neg}$ which will, after normalization, be used to calculate backprojection maps on the input image. We refer to this step as "histogram filtering" [3].

H_{filt} is directly used to evaluate the quality of the original detection and eliminate false positives which would lead to stray tracks: The average backprojection value inside the detection window is calculated, and the initialization of a track is only made if this value exceeds a certain threshold (in our case 50%).

After a color model for the target object was built, tracking is made in subsequent frames using the meanshift [4] algorithm on the generated backprojection maps, reestimating both the position as well as the size of the object.

Additionally, the target's color model is adapted every time a new frontal face detection shows sufficient overlap with the actual tracking window. For this, the filtered color histogram for the new detection window H_{new} is computed as above, and the adaptation is made according to Eq. 2.

$$H_{adapt} = (1 - \alpha) \cdot H_{old} + \alpha \cdot H_{new} \tag{2}$$

Here, the learning factor α is determined automatically, based on the quality of the new detection. Again the average backprojection value of pixels inside the detection window P_{avg} is used, and the learnrate is defined as

$$\alpha = \frac{P_{avg}}{P_{total}} \cdot 0.5 \tag{3}$$

with P_{total} being the total number of pixels inside the detection window. In this way, the learnrate increases with the representational quality of the new color model and a maximum learnrate of 50% can be achieved.

2.2.2 Upper Body Tracking

In the envisioned application scenario for the active camera tracking system, the tracked subject can not be expected to keep facing in the direction of the camera. As the subject moves around and turns his head, or due to the viewing angle of the camera, his or her face may become heavily occluded or even invisible, causing the face tracker to fail. To compensate for this fact, as stated earlier, the system relies on a combination of trackers, running in parallel.

When a new target subject is acquired, similarly to the face model, a filtered color histogram for the upper body is initialized below the face detection, using an image subwindow of equal size, which has been shifted downwards by the height of the initial detection. In most cases, this subwindow contains the relevant color information to model the upper torso. This upper body histogram is filtered in a similar way as the face histogram, with the exception that the background histogram H_{neg} is not built using all image pixels. A region of 3 times the size of the original detection, which is expected to contain the upper body, is first masked out from the image, before the computation is made. This is because the upper body, in contrast to the face, can represent a considerable portion of the total image, and would bias the computed probability $P(x|\neg UpperBody)$ if included. Figure 1 shows the backprojection maps for the face and upper body models.

Figure 1. Backprojection maps for face and upper body, overlayed on the original image. For display purposes, the maps were thresholded and pixels exceeding the threshold colored red (face) and green (upper body). The circles represent the estimated size (only for the face) and center of the tracked region

Just as the face model, the upper body model is adapted every time a new detection closely matches the facial track. In contrast to the face, the upper body histogram backprojection does not allow a stable estimation of the size of the target, as the upper body may be cut off at the bottom of the image, slanted, etc, but it represents a more stable support for position estimation in the presence of noise, as the subject freely moves in the scene.

2.3. KLT Feature Tracking

The third of the tracker modules relies on KLT-features [6, 7]. These are essentially image regions that exhibit a strong gradient in both x and y directions. The tracker implementation realized here closely resembles that of [5], with the exception that no skin color probability is used to weigh the features. The same detection window as used for color tracking serves to initialize the KLT feature tracker. The detection window is searched for good features to track, and the found features are weighted from 100% to 0%, according to their distance to the window center, owing for the fact that features close to the border are more likely to belong to the background. To each feature, a small 10x10 pixel patch is stored and will be used for feature matching in the subsequent tracking steps. Figure 2 shows the initialization of a KLT-feature track inside a face detection window.

In tracking, a region of 3 times the estimated face size in the previous frame is searched for new features, and the features are scored through template matching of the model and image patches. When a match is found, the feature's confidence is increased by 10%. Likewise, the confidence is decreased by 10% if no matching pattern is found. Features with confidence scores below 20% are eliminated as new detected features are being added to the model with an initial score based on their distance to the track center.

The output of the KLT tracker is then computed as the median of the feature positions.

Figure 2. KLT features and their scores (0%-100%), shown as pixels of increasing brightness

2.4. Tracker Fusion and Track Termination Criteria

Based on the strengths and weaknesses of the separate trackers, the following selection strategy for tracker outputs has been implemented:

In general, the color-based meanshift trackers are used as they have shown to be quite reliable in the majority of cases. Rather than averaging their outputs, the system relies first on the face track, which allows for target size estimation, and falls back to the upper body track only when the face tracker fails. Additionally, basic geometric constraints are verified to detect a failure in the color tracking, e.g. when the upper body estimate lies above the face estimate, or both backprojection regions become very small. In such a case, the KLT feature track is used. Although the KLT tracker can not cope with rapid camera movement, it is less sensitive to lighting conditions and allows to keep track of slow targets until the color trackers can be reinitialized.

3. Active Camera Calibration and Fuzzy Control

The requirements to our tracking system were twofold. On the one hand, we wish it keep robust track of a target person even in the case of interference or occlusion, on the other hand, it should be able to quickly switch to and acquire a new target on demand. The command for a target switch could come from a higher order prioritizing system based on, e.g. attention driven trackers from fixed cameras, sound source localizers, simple motion sensors, etc, and would ideally be in the form of a more or less accurate 3D estimate of the target location.

To ensure correct target switching, it can be of great help to know the intrinsic and extrinsic parameters of the active camera at every point in time. This is also useful to obtain 3D position estimates for the tracked object, coordinate multiple cameras, etc.

In this system, the camera parameters are continuously updated using rotation and zoom information read from the camera. This is acquired at rougly 4fps from the camera control interface. Although this information is not fast or precise enough to be used in pixel-level image processing, it is more than sufficient for coordination with external systems during target switching.

3.1. Update of Camera Parameters

An initial calibration of the camera is performed in its rest position ($pan = tilt = 0°$) using standard calibration tools and a calibration checkerboard. In our case, the standard tools available in the OpenCV library [15] were used on automatically detected checkerboard images for computation of the intrinsic parameters [8], and the freely available *Camera Calibration Toolbox for Matlab* [14] was used to calculate the extrinsics. In this way, initial values for the 3D camera position T_{init} and rotation R_{init}, as well as focal length estimates at various discrete zoom steps $f_{x,0} \cdots f_{x,8}$ were obtained.

The continuous update of camera parameters is then made in the following way:

For the extrinsic parameters, the actual rotation matrix is calculated from the latest camera pan and tilt information, by multiplying the initial rotation matrix with a "correction matrix" R_{corr} (Eq. 4),

$$R_{act} = R_{init} \cdot \begin{pmatrix} \cos(\beta) & \sin(\alpha)\sin(\beta) & -\cos(\alpha)\sin(\beta) \\ 0 & \cos(\alpha) & \sin(\alpha) \\ \sin(\beta) & -\sin(\alpha)\cos(\beta) & \cos(\alpha)\cos(\beta) \end{pmatrix} \tag{4}$$

with α the camera pan angle and β the tilt angle.

The focal length itself is not directly readable and is interpolated for the current camera zoom step from the discrete values $f_{x,0}$ to $f_{x,8}$, using a 4th order polynomial function. Figure 3 shows the results of interpolation. The camera itself is able to zoom to any continuous value within a range of 0 to 18 zoom steps, corresponding to minimum and maximum focal lengths. For our experiments, however, only the values up to step 12 were used, as the room dimensions comstrained the range of useful settings. Also, the discrete values $f_{x,9}$ to $f_{x,12}$ could not be calculated, as these magnification settings did not allow to fit the calibration object into the image anymore. Nevertheless, the interpolation results, even at zoom step 12, produced maximum deviation errors of only a few pixels, which was completely sufficient for the purpose of this tracking system.

3.2. Fuzzy Control

The advantage of the fuzzy controlling scheme over other techniques, such as PID controllers, etc, is that expert knowledge can be used, encoded in the fuzzy rules, to simulate the natural behavior of a human operator [9]. It

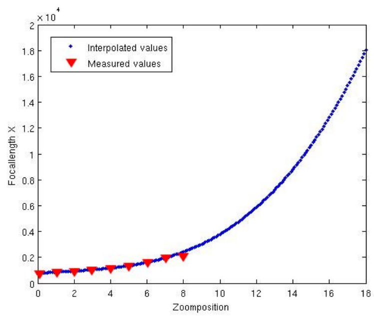


Figure 3. Interpolation of the focal length

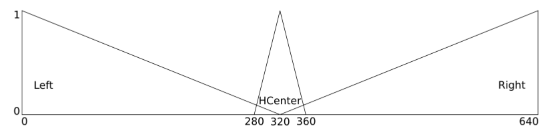
allows for much smoother camera handling in stable situations, while maintaining the ability to react quickly and keep the target in the image in emergency situations. It also allows to formulate the desired behavior of the system in simple terms, making the design process straightforward.

In our system, the input to the fuzzy controller are the x and y position, as well as the size of the target object in the image. Additionally, the gradients of these values are also fed to the controller. Likewise, the ouptut of the fuzzy controller are the required pan, tilt and zoom speeds for the camera. Using gradients and angle speeds allows for much more dynamic and smooth control as absolute positioning would, as the camera can adapt its rotation and zoom to match the relative speed of the target. Figure 4 shows the fuzzy sets for the input horizontal position in the image, horizontal speed, image size and gradient, in pixels, and for the pan velocity in degrees per second.

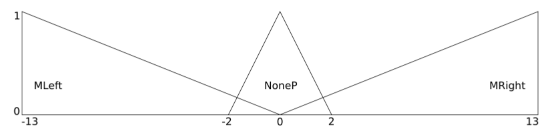
Based on these sets, the behavior of the system is determined by a set of rules connecting input values to expected outputs. The following lists a few sample rules:

- IF Left AND MLeft THEN FastLeft SlowOut
- IF Right AND MLeft THEN NoneP
- IF Fine AND NoneZ THEN NoZoom
- IF Big AND NoneZ THEN SlowOut
- IF Small AND Approaching THEN NoZoom
- IF Small AND Departing THEN FastIn
- ...

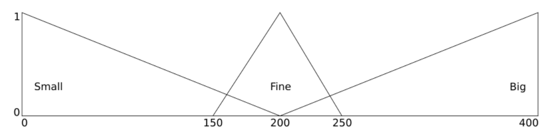
Apart from the usual rules for adjusting pan and tilt, one can see that specific behavior can be encoded, such as in the first rule: When the track is close to the left image border and is moving left, the danger of losing track is imminent and the camera should quickly move to the left, but it should also zoom out slowly, as a wider angle of view will automatically help to keep the track in the image. On the other hand, if the target is close to the right edge and is moving



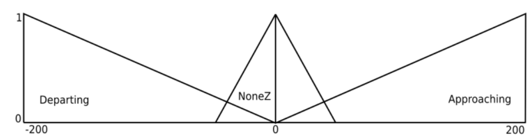
(a)



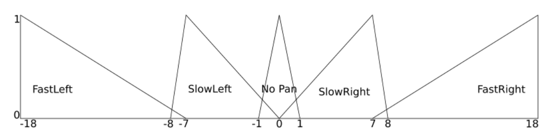
(b)



(c)



(d)



(e)

Figure 4. Sample fuzzy sets for camera control. Fig. 4(a): Fuzzy input set for the x-position of the person in the image in pixels. Fig. 4(b): Fuzzy input set for the horizontal speed of the person in the image in pixels/frame. Fig. 4(c): Fuzzy input set for the width of the face in the image in pixels. Fig. 4(d): Fuzzy input set for the change in face width in pixels/frame. Fig. 4(e): Fuzzy output set for the camera pan speed in degrees/s.

left, nothing should be done, as it is expected the target motion will bring it to the image center without the need for camera motion.

In our implementation, the fuzzy rules have been designed manually, and the fuzzy sets empirically adjusted to yield satisfactory results on a range of test scenarios.

4. Experimental Evaluation

To evaluate the effectiveness of the automatic active camera person tracking system, a series of sample scenarios were tested using one pan-tilt-zoom camera in a medium-sized seminar room. The room was relatively cluttered, with very uneven lighting, and with tables, chairs and technical equipment of various shapes and textures in the background. The camera itself was attached to one of the room walls at approximately 2m height. It is a SONY EVI-D70P delivering an interlaced PAL signal at 25fps. The camera images were deinterlaced and downsampled to 320x240 resolution. The camera is controlled through an RS-232 connector, has a pan range of $\pm 170°$, a tilt range of $+90°$ to $-30°$, and can rotate at up to $100°/s$. Its focal length can be continuously varied from 4.1mm to 73.8mm. In our implementation, the EviLib library [16] was used for camera access. All processing was done on a Pentium 3GHz dual core machine, and no tuning was made on the fuzzy rules or sets to adapt to the different test scenarios.

4.1. Scenario 1: Single person tracking

The goal in this scenario was to detect and track one person moving freely in the room, without explicit cooperation with the camera system. The subject could walk at normal pace, change direction, turn his back to the camera, etc, with distances to the camera varying from roughly 1 to 5m. Figure 5 shows a few key frames.

After succesful detection of a frontal face, the face and upper body models were correctly initialized. The estimated face and upper body centers are marked by green and red circles in the images, respectively. As the subject walks to the right of the image, the camera pans at the same speed to keep a smooth track of the face. As he turns to walk away from the camera, the system loses track of the face region, but keeps tracking the upper body region. In this period, the system keeps a constant zoom, as no size estimate of the head is availabe. As the side of the subject's face becomes visible again, the system succesfully recovers the face track.

As can be seen from the images, the lighting conditions on the subject's body vary considerably as he walks through the room. Nevertheless, the learned in color model is robust enough to allow a stable track.

In Figure 6, an example for tracking failure is shown. In this case, a too quick panning motion from the camera caused a tracking failure in the second frame. This in turn provoqued a faulty reaction from the fuzzy controller, and the camera motion, in turn, finally resulted in complete track loss. In such a case, the system is still able to recover as soon as a new frontal face detection can be made.

4.2. Scenario 2: Tracking through interference

This scenario served to test the robustness of the system in the case of several moving persons. Here, the tracked subject could pass in front of or behind other walking or immobile persons. The test showed that the system could keep correct track of the initial subject, even through complete occlusion (see Figure 7). This is mainly due to the difference in upper body color. As the target person's face becomes occluded, the face track wrongfully switches to the

Figure 5. Bridging a gap in face tracking

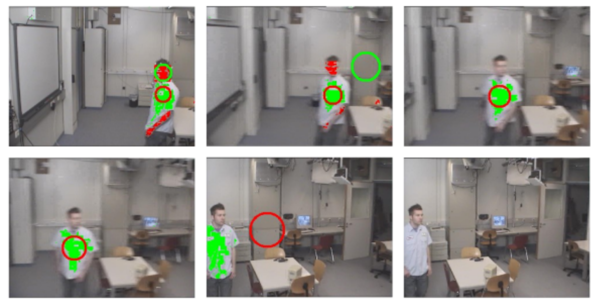


Figure 6. Tracking failure caused by extreme motion

Figure 7. Keeping track through occlusion by another subject

occluding person's face, but the error is quickly recovered as the upper body becomes visible again.

In the case the target person stays occluded for longer periods of time, the upper body model would eventually adapt to the foreground person, but only if this person continually faces the camera, as no adaptation is made otherwise.

4.3. Scenario 3: Switching between speakers

This scenario served to test the speed of the system at switching and acquiring closeups of new target persons on demand. For this purpose, the system was given occasional switch commands, in form of expected 3D coordinates of the target speaker. In this test, the external hint was given manually, but it could just as well come from other modules, such as sound source localizers or other wide area trackers.

Figure 8 shows the test sequence. In frame 54, the cam-

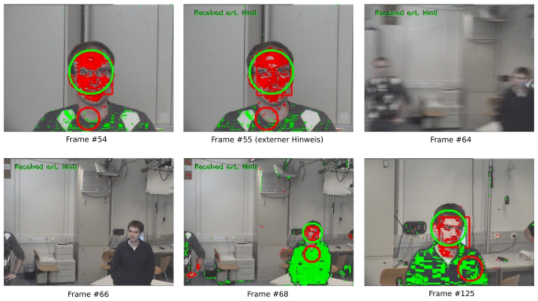


Figure 8. Acquiring speaker closeups on demand

era system tracks the initial speaker, adjusting the zoom factor to keep the face at ~ 100 pixels width. In frame 55, the external hint is received. The system first zooms out, then quickly pans to the new position (frames 64-66). In frame 68, the new speaker's face is detected and the system zooms in slowly until the desired face size is reached (frame 125). Although fast target switching was shown to be succesful in such a frontal setup, the requirement of frontal face detections for track intialization is a limiting factor in looser scenarios, as will be discussed in the following.

4.4. Scenario 4: Meeting recording

The goal in this scenario was to gain closeups of the respective speakers sitting at a meeting table. In this case, the subjetcs are located close to each other, but stay immobile throughout the sequence, making tracking easier. Again, the external switch signal was given manually. Figure 9 shows the sample sequence, where again the system zooms in on the first speaker (frames 364-408) before receiving the switch signal. It then zooms out (frame 719) and reorientates on the new speaker. However, it fails to acquire a frontal face for more than 1000 frames and only succeeds in zoming on the new speaker in frame 2042. Other speakers could not be acquired at all, as they always faced away from the camera. To achieve more reliable initialization, the inclusion of other types of detectors, at least for profile faces, would clearly be of benefit.

The sample scenarios have shown that as long as a detection can be achieved in reasonable time, the developed camera tracking system is able to keep track of a human target walking at reasonable speeds at any place in a medium sized room, through strong lighting variations, partial occlusions and interference from other subjects, and to recover from partial track losses robustly. It is capable of keeping smooth track of moving subjects as well as quickly obtaining high quality closeups of still targets, e.g. alternating speakers. The system's effectiveness is reached by the combination of a highly reliable person model initialization, cautious model adaptation, and the use of a mixture of trackers.

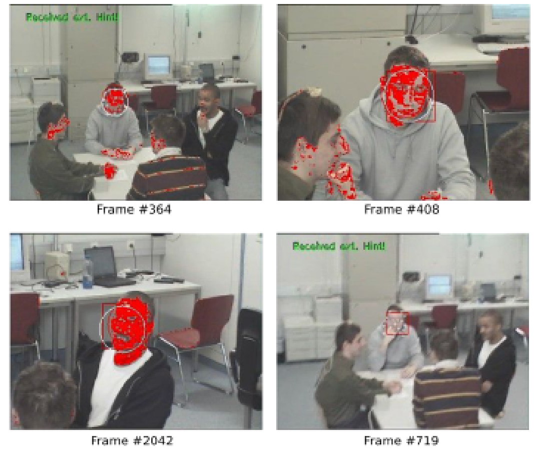


Figure 9. Switching between speakers in a meeting

5. Summary and Conclusion

In this paper, we have presented an automatic system for the monitoring of indoor environments using an off-the-shelf pan-tilt-zoomable camera. It uses boosted cascades of Haar-feature classifiers and color histogram filtering to achieve reliable initialization of person tracks even in the presence of camera movement. It uses a combination of 3 types of trackers: adaptive color feature trackers for the face and upper body, and a KLT feature tracker, to ensure robust tracking and track recovery in the presence of camera movement, illumination changes, occlusion or interference. The parameters of the active camera are recomputed on the fly, and a fuzzy controlling scheme allows for smooth tracking of moving targets, rapid switching between targets, as well as acquisition of high quality closeup views, similar to the natural behavior of a human cameraman. The system has been tested on a series of natural indoor monitoring scenarios, including the tracking of subjects in the presence of heavy interference and the recording of active speakers in a meeting scenario. It showed a high degree of naturalness and flexibility, was able to quickly acquire and track subjects, and recover from tracking errors. Future enhancements should include the addition of other types of person detectors for initialization, e.g. profile face or upper body detectors, and the actual 3D-tracking of subjects, e.g. in a federation of cameras.

6. Acknowledgement

The work presented here was partly funded by the *European Union* (EU) under the integrated project CHIL, *Computers in the Human Interaction Loop* (Grant number IST-506909).

References

[1] Rainer Lienhart and Jochen Maydt, *"An Extended Set of Haar-like Features for Rapid Object Detection"*. IEEE ICIP 2002, Vol. 1, pp. 900–903, Sep. 2002.

[2] Paul Viola and Michael Jones, *"Rapid Object Detection using a Boosted Cascade of Simple Features"*. International Conference On Computer Vision And Pattern Recognition, 2001.

[3] Kai Nickel and Rainer Stiefelhagen, *"Pointing Gesture Recognition based on 3Dtracking of Face, Hands and Head Orientation"*. 5th International Conference on Multimodal Interfaces, Vancouver, Canada, Nov. 2003.

[4] Dorin Comaniciu and Peter Meer, *"Mean Shift: A Robust Approach Toward Feature Space Analysis"*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 5, May 2002.

[5] Mathias Kölsch and Matthew Turk, *"Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration"*. In IEEE Workshop on Real-Time Vision for Human-Computer Interaction (at CVPR), 2004.

[6] B. D. Lucas and T. Kanade, *"An Iterative Image Registration Technique with an Application to Stereo Vision"*. In Proc. Imaging Understanding Workshop, pages 121-130, 198

[7] J. Shi and C. Tomasi, *"Good features to track"*. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, Seattle, June 1994.

[8] R. Y. Tsai, *"A versatile camera calibration technique for high-accuracy 3d machine visionmetrology using off-the-shelf tv cameras and lenses"*. IEEE Journal of Robotics and Automation, RA-3(4), pp. 323-344, August 1987.

[9] Earl Cox, *"Fuzzy fundamentals"*. IEEE Spectrum, 1992, pp. 58–61

[10] Shinji Tsuruoka, Toru Yamaguchi, Kenji Kato, Tomohiro Yoshikawa, Tsuyoshi Shinogi, *"A Camera Control Based Fuzzy Behaviour Recognition of Lecturer for Distance Lecture"*. Proceedings of the 10th IEEE International Conference on Fuzzy Systems, December 2001, Melbourne, Australia.

[11] Nuno de Castro, Rodrigo Matias, M. Isabel Ribeiro, *"Target tracking using fuzzy control"*. Proceedings of the Scientific Meeting of 3rd. Robotics National Festival, ROBOTICA2003, Lisbon, May 2003.

[12] Erik V. Cuevas, Daniel Zaldivar, Raul Rojas, *"Fuzzy condensed algorithm applied to control a robotic head for visual tracking"*. International Symposium on Robotics and Automation, IEEE, ISRA 2004, Queretaro, Mexico, August 25-27, 2004

[13] Arun Hampapur, Sharath Pankanti, Andrew W. Senior, Ying-li Tian, Lisa Brown, Ruud M. Bolle, *"Face Cataloger: Multi-Scale Imaging for Relating Identity to Location"*. IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS 2003), July 2003, Miami, FL.

[14] Jean-Yves Bouguet, *"Camera Calibration Toolbox for Matlab"*, `http://www.vision.caltech.edu/bouguetj/calib_doc`

[15] OpenCV - Open Computer Vision Library, `http://sourceforge.net/projects/opencvlibrary/`

[16] EVILib, `http://sourceforge.net/projects/evilib/`