

Face Tracking in a Multi-Camera Environment

Petr Douthek¹, Tomáš Svoboda¹, Katja Nummiaro², Esther Koller-Meier¹ and Luc Van Gool^{1,2}

¹D-ITET/BIWI

Swiss Federal Institute of Technology Zürich
Switzerland

²ESAT/PSI-VISICS

Katholieke Universiteit Leuven
Belgium

Abstract

We present a system for the robust real-time tracking of human faces. The system utilizes multiple cameras and is built with low-cost standard equipment. A 3D tracking module that uses the information from the multiple cameras is the core of the presented approach. Endowed with a virtual zooming utility, the system provides a close-up view of a face regardless of the person's position and orientation. This best matching front view is found by comparison of color histograms using the Bhattacharyya coefficient. The tracking initialization and learning of the target histograms are done automatically from training data. Results on image sequences of multiple persons demonstrate the versatility of the approach. Telepresence, teleteaching or face recognition systems are examples of possible applications. The system is scalable in terms of the number of computers and cameras, but one computer/laptop with three low-cost FireWire cameras is already sufficient.

1 Introduction

Decreasing prices of powerful computers and cameras made multi-camera setups affordable for a larger group of users. Such systems are suitable for telepresence applications like teleteaching or videoconferencing as they can cover the entire scene. People can move freely, while one of the cameras provides a good view of their face the audience all the time. Exchange of information between cameras increases the system capabilities and robustness. Occlusion or clutter can be resolved by taking another viewpoint of the tracked object.

This paper presents a multi-camera system capable of showing a close-up of face(s) of human(s) moving in a smart room. The Aviary project [14] already presented results for such task, but it relies on special equipment like pan-tilt-zoom units, omnidirectional cameras and microphone arrays. The system is not portable. Comaniciu *et al.* [3] detect faces in multiple views of active cameras using color histograms which have to be initialized from a

database. The resolution of the resulting video stream decreases with an increasing distance from the head position which results in a lower amount of data to be transmitted. The information from multiple views is not used to calculate the 3D position which makes it more complicated to establish face correspondences between views.

The M2Tracker [9] tracks robustly multiple persons, but has no real-time capability and tracks the whole bodies so that people have to keep their distance to all the cameras. Cai and Aggarwal [2] track humans with a calibrated multi-camera setup by fitting a coarse 2D body model to the segmented image. The camera is switched when a person is about to leave the view. However, only one subject can be tracked at a time. Khan *et al.* [8] perform tracking and view switching with an uncalibrated multi-camera setup which is initialized by one person walking through the room to find the boundaries of the fields of view of the cameras.

We built our system [4] as a modular and portable system using low-cost consumer hardware. It can be quickly calibrated using the method described in [12]. We decided to employ 3D head tracking and color histogram comparison to track faces of people moving in the room. The tracking is based on detection of heads in images, establishing head correspondences across views and reconstructing the 3D position of the heads. The color histogram of a tracked head is compared to previously acquired target histograms to decide whether the particular camera sees a face or not. There are several advantages to combining the tracking with a histogram comparison:

- tracking multiple people – heads can be put into correspondence using 3D information, so that occlusions can be resolved
- initialization – it is difficult to detect the head position in the image initially or when it is lost if only its histogram is known; a projected 3D position supplied by the less unfortunate cameras gives a good initial location
- color histogram learning – we present a simple method for automatic learning of target color histograms based

on the 3D head positions

- face orientation – 3D positions of a head can be used to calculate the corresponding velocities, but more is needed to determine its facing direction – comparison of the tracked head with the target color histogram yields this information
- RGB histograms are computationally efficient, scale and rotation invariant and robust against partial occlusion.

The following steps are taken for each frame, see Fig. 1:

1. capture images from all cameras
2. local image processing servers segment the background, find connected components, calculate silhouettes around them and use this information to locate the heads
3. then they decide whether it is the face or the back of the head and send this decision together with the head position and its size to the central computer
4. process on the central computer adds heads to the existing 3D objects, creates new objects if necessary and removes old ones
5. the best camera for viewing the face is selected as well as a zoom-in area around the head, finally a request for this area is sent to the image processing server.

The steps 2-5 are described in this paper. Details of the synchronous image acquisition in step 1 can be found in [13].

2 2D Head Detection

First, foreground objects are segmented from the current image using a background model previously learned on an empty scene [5] i.e. the scene without people in it. The integration of an adaptive background modeling process based on [7], which does not require the scene to be empty for learning, is ongoing. Connected foreground components are then found together with silhouettes around them, see Fig. 2. Connected components can be optionally joined in the case of an oversegmentation which occurs mainly in badly illuminated scenes.

Each connected component is evaluated by using some natural assumptions about the typical human appearance: the quality of the component is decreased if it is wider than tall and components touching the top border of the image are penalized because it is likely that the head is not completely visible. The components are sorted by their likelihoods and those with a probability lower than one tenth of

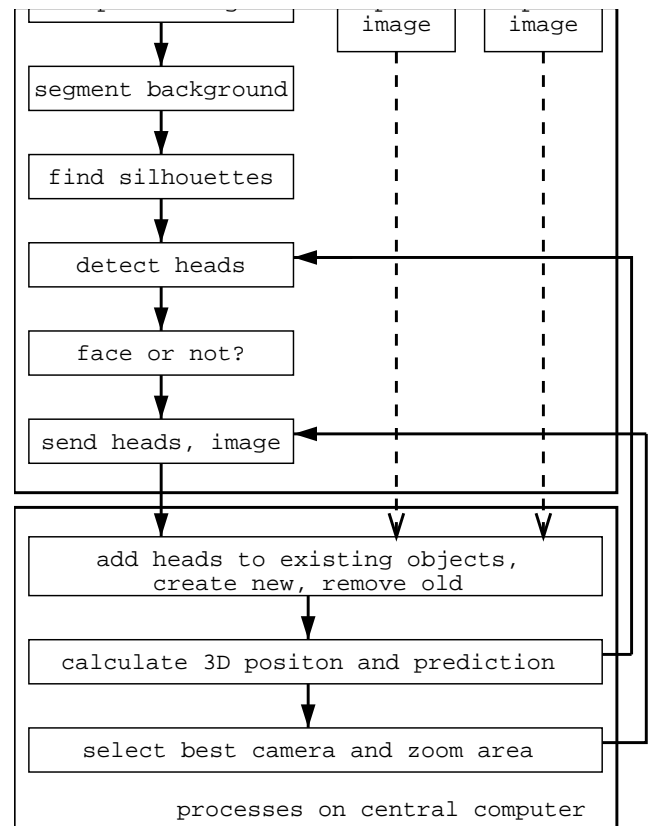


Figure 1: The diagram shows the sequence of operations performed for each frame.

the maximal value are discarded. When the maximal number of people moving in the room is known to be n , all but the n best components are discarded. For example, only one person is allowed in the room when learning the target histograms (described in Sec. 4).

For each of the human candidates the related silhouette is found and the top of the head is selected among the silhouette extremas. The vertical size of the head ellipse is calculated as one fifth of the whole connected component height. This size is a rough approximation and is only used when no near predicted head exists. See Sec. 3 for a description how the prediction is obtained. A fixed aspect ratio $horizontal_axis : vertical_axis = 0.75$ for the head worked well in our experiments.

Finally, heads detected in each view are sent to the central computer for 3D calculations.



Figure 2: Each incoming image is processed to find connected components in a segmented image to determine the silhouettes around them and to detect heads (white ellipse) at the top of the silhouettes. The rectangle is a bounding box around the segmented component.

3 3D Head Tracking

The system manages a set of 3D objects that are being tracked. In each image, we compare all detected heads with the projected 3D objects. The head is assigned to the closest object o^* by calculating

$$o^* = \arg \max_{o \in \text{objects}} 2 - \frac{d(\text{head}, \text{projection}(o))}{l}, \quad (1)$$

where $l = \text{vertical_projected_size}(o)$ and d is the Euclidean distance. The head remains unmatched if the maximal value found in (1) is negative. This means that the 2D head is assigned to the closest 3D object if its distance from the projection of the 3D object is less than two times the size of the projected object.

New 3D positions of all objects are determined using N-view linear reconstruction [6]. The reconstruction is attempted for all pairs of heads not assigned to any object so far. A new object is created when the reprojection error of the reconstructed point is below a certain threshold.

Finally, the paths of all pairs of 3D objects are compared to decide if the objects are identical and thus should be joined. Objects which could not be reconstructed for several frames are removed.

The next 3D position of the head is predicted from previous positions using Kalman filtering. The head size is not predicted, a fixed size of the head based on the real human head size is used instead. The prediction of 2D heads in the next frame can be calculated by projecting the predicted 3D object into the cameras.

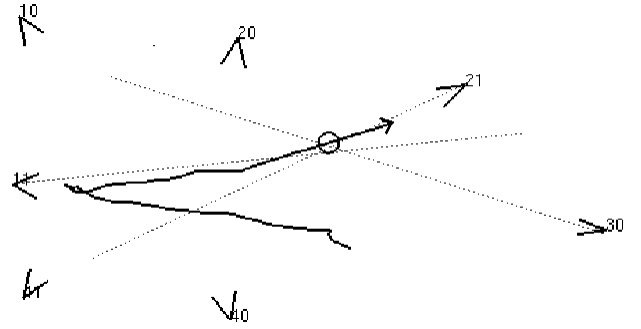


Figure 3: N-view reconstruction. This sketch shows a top view on the scene. Three heads in 3 different cameras are matched against the object (circle). The dotted lines show 3D rays through the 2D image position. They do not intersect exactly in one point due to detection and calibration errors. The 3D position is found by algebraic error minimization. Previous positions are stored and drawn as line. The velocity vector is shown as an arrow attached to the target position.

These head predictions are sent back to the local image processing servers to facilitate the retrieval of the head position and size in the next frame.

4 Face or not?

Although it is possible to calculate the heading direction from subsequent 3D head positions, the facing cannot be determined the same way, because the person may stand still or the head can turn independently of the body movement. Color histograms are employed to decide whether the face or the back of the head is seen in a particular view. Two target histograms – one for the face and one for the back – are therefore needed for each camera view, see Fig. 4. Each camera needs its own set of histograms. Camera parameters are adjusted by an automatic procedure to align color characteristics of the cameras [10]. Nevertheless, the color alignment is not powerful enough to compensate for all the lighting variations. Some cameras may see windows and some may not which induces strong contrast/brightness differences between images.

The similarity between the current head ellipse in the image and the target histogram is measured by the Bhattacharyya coefficient [1] which is implemented as proposed in [11]. Both the detected and the nearest predicted head are compared with the face and back target histograms after head detection described in Sec. 2. The most likely of these four combinations is chosen and the selected head ellipse is sent together with the “face/back” information and the related Bhattacharyya coefficient to the central server.

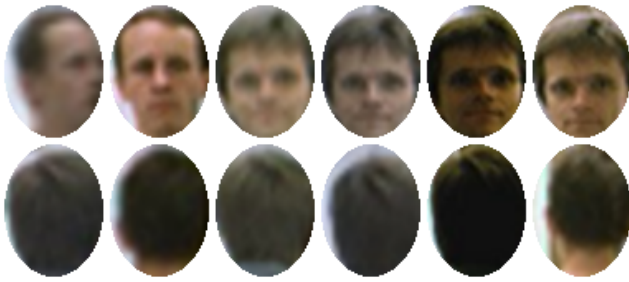


Figure 4: Learning histograms. The top row contains the image area from which the face color histogram of the target for each camera was initialized. The areas in the bottom row served for initialization of the histogram for the back for the head. The selection of head ellipses is an automated process which results in some errors – the misdetection of the head orientation is most significant in the first image. Therefore, the histogram is calculated as an average from several suitable heads. Notice the differences in contrast between cameras – they are caused by different illumination and camera parameters. The images of heads are resampled to the same resolution for better visibility.

Initializing the target histograms manually for multiple cameras would be unpractical. Therefore, we designed a method for semi-automatic initialization. A person walks towards the cameras without turning his/her head so that the heading direction is equal to the facing direction. The target face histogram is initialized from a head ellipse if the person is moving towards the camera with sufficient speed, see Fig. 5.

An average histogram is calculated if the situation described above occurs in multiple frames. Low speed usually means that the person is turning and the estimation of the heading vector is unreliable. The target histogram of the back of the head is calculated in the same way.

A symmetry test is employed to increase the robustness of the histogram learning. The color distribution of both the face and the back of the head should be symmetric over the vertical axis, as shown in Fig. 6. Therefore, the head ellipse is divided into the left and the right half-ellipse, a histogram for each part is calculated separately and the halves are compared using the Bhattacharyya coefficient. Heads without sufficient symmetry are discarded because it is likely that the side of the head was detected or, only a part of the head is inside the ellipse and the rest is background.

5 View Selection and Zooming

The knowledge of the head position and the face orientation allows the system to select the camera where the face is best visible. It also allows the system to perform a virtual zoom

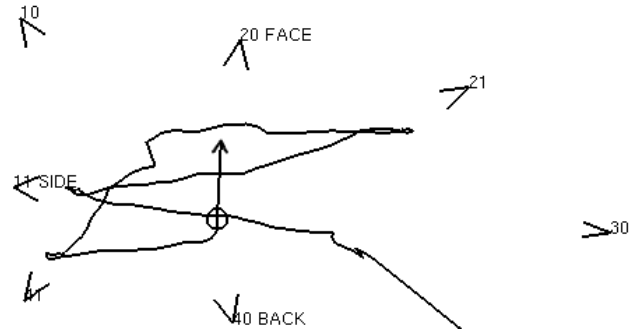


Figure 5: A snapshot of the learning procedure. The tracked person is moving towards the camera 20 which learns the target histogram in this frame. The camera 40 is learning the histogram for the back of the head. If histograms of the side of the head are enabled the camera 11 will learn it, in contrast to camera 30 where the distance to the tracked person is too big. The remaining cameras do not see the head or the angle is not suitable for learning.



Figure 6: A test on the color symmetry of the head can reveal misdetections. The first face is detected correctly and the color distribution in its left and right half is almost the same. The same holds for the correctly detected back of the head in the third ellipse. Only half of the face is seen in the second ellipse and a side of the head is detected in the last ellipse instead of the back. In both cases the color distributions of the left part differ from the distributions of the right part.

on the head and the area around it. A virtual zoom has the disadvantage of producing images with a lower quality, but an optical zoom would only make sense when paired with a pan-tilt unit. The cost of a sufficiently agile version of such a complex camera is comparable to the cost of our whole system including computers. Furthermore, an advantage of the virtual zoom is that information from the whole field of view of the camera is still available even when it is “zooming”.

At each frame, the 2D heads belonging to each of the tracked 3D objects are compared and the one which is a face and has the highest Bhattacharyya coefficient is chosen, see Fig. 7. This decision is propagated back to the relevant image server which is requested to “zoom in” on the face and send the corresponding image cutout to the central computer. This decision-request-send loop causes a delay



Figure 7: Color histograms of detected heads are compared to the target histograms and a “face/back” decision is made. The view with the most confident face decision is selected.

of one frame and, hence the *predicted* position is used to compensate it. A free space is added around the predicted head, see Fig. 8, which is proportional to the head size and is larger in the direction of movement as people prefer seeing a person walking towards the center over a person walking away from it.

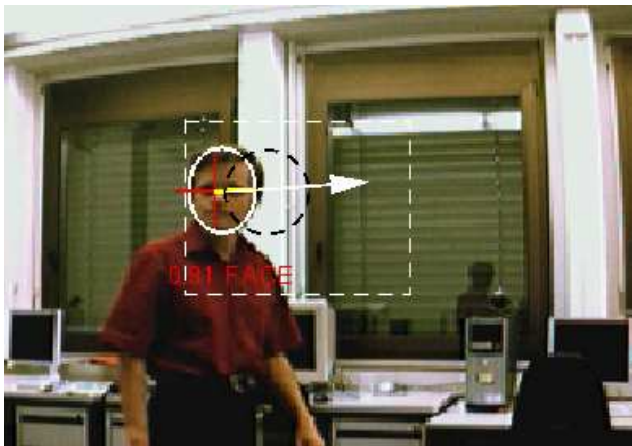


Figure 8: The selection of a zoom area is based on the prediction of the next head position (black ellipse). A certain minimal size of the area is required to maintain an acceptable resolution. Some free space is added around the predicted head, the margins are larger in the direction of movement (arrow). When adding top and bottom margins bottom is slightly preferred to show the shoulders rather than an empty space above the head.

To prevent disturbing rapid switching between the cameras, we multiply the Bhattacharyya coefficient of the head

in the previously selected camera by a certain “switching resistance coefficient” (the value of 1.05 is currently used) before selecting the head with the highest coefficient.

When multiple objects are tracked, we maintain the optimal view for each of them separately.

6 Experiments

A multi-camera setup with seven cameras attached to four desktop computers was used for our experiments. The cameras are placed around the working area in one room, see Fig. 9 for the spatial arrangement. We used a capturing frequency of 5 frames per second. The system may operate at a higher frequency but is currently slowed down by saving all images to disk.

Two experiments are shown, the first one features a single person walking in the room and turning the head independently of his body movement. It demonstrates the ability to track the head based on the segmentation and to select the view using color histograms, see Fig. 10.

The second experiment tested multiple object tracking in a sequence with two persons walking in the room. See Fig. 11 for the selected close-ups of faces. The corresponding paths are drawn on the floor-plane. The current algorithm is limited in the number of people as the silhouettes start to merge more often in densely populated scenes. The robustness is better when the segmented persons are separated in most of the views. If too many silhouettes are connected, then the tracking fails.



Figure 10: Keeping face visible regardless of the motion direction. The top row displays the images from cameras 10 (a), 41 (b) and 20 (c) selected in frames 263, 268 and 291 of our “head turning sequence”, see Fig. 9 for the camera positions. The bottom row shows the zoom-in areas. The face view is maintained even though the face orientation deviates from the motion direction.

7 Conclusion

We built a mobile multi-camera system using standard low-cost consumer hardware and performed real-time face tracking experiments on it. We combined a 3D tracking based on a segmentation with color histogram comparison to overcome the problems with the learning, the initialization and the constraint on a single tracked object that many trackers have.

In the future we want to employ a probabilistic reasoning to process the information that is already available with better results. Improvements are planned also for the target histogram learning phase. We would like to be able to learn new faces on-the-fly without any special learning sequence. We also want to introduce face and pedestrian detectors in order to increase the system selectivity. Also, we plan to look into the problem of handling more crowded scenes.

Acknowledgments

This work has been supported by ETH Zürich project BlueC and European Union project STAR.

References

- [1] F. Aherne, N. Thacker, and P. Rockett. The bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika*, 32(4):1–7, 1997.
- [2] Q. Cai and J.K. Aggarwal. Tracking human motion in structured environments using a distributed-camera system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(12):1241–1247, November 1999.
- [3] D. Comaniciu, F. Berton, and V. Ramesh. Adaptive resolution system for distributed surveillance. *Real Time Imaging*, 8(5):427–437, 2002.
- [4] Petr Doubek, Tomáš Svoboda, and Luc Van Gool. Monkeys — a software architecture for ViRoom — low-cost multicamera system. In James L. Crowley, Justus H. Piater, Markus Vincze, and Lucas Paletta, editors, *3rd International Conference on Computer Vision Systems*, number 2626 in LNCS, pages 386–395. Springer, April 2003.
- [5] Nico Galoppo von Bories, Tomáš Svoboda, and Stefaan De Roeck. Real-time segmentation of color images — implementation and practical issues in the blue-c project. Technical Report 261, Computer Vision Laboratory, Swiss Federal Institute of Technology, March 2003.
- [6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.
- [7] Michael Harville, Gaile G. Gordon, and John Woodfill. Foreground segmentation using adaptive mixture models in color

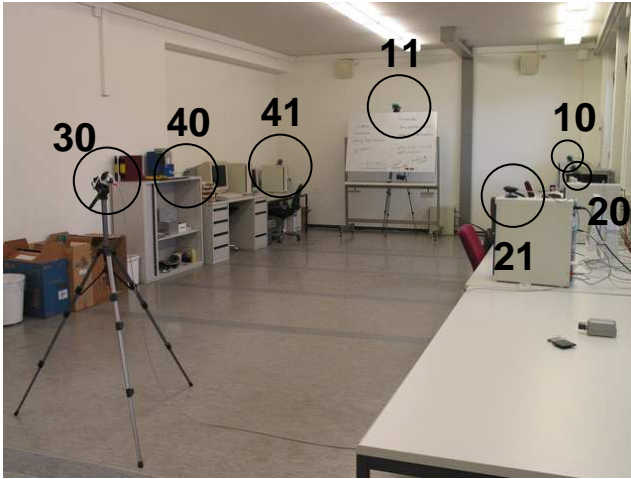


Figure 9: The camera layout used for the experiments. Each of our seven cameras is encircled and denoted with its Id number. These numbers are used on the top-view sketches.

and depth. In *IEEE Workshop on Detection and Recognition of Events in Video*, pages 3–11, 2001.

- [8] Sohaib Khan, Omar Javed, Zeeshan Rasheed, and mubarak Shah. Human tracking in multiple cameras. In *International Conference on Computer Vision*, July 2001.
- [9] Anurag Mittal and Larry S. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *The seventh European Conference on Computer Vision, ECCV2002*, volume 1 of LNCS, pages 18–36. Springer, May 2002.
- [10] Harsh Nanda and Ross Cutler. Practical calibrations for a real-time digital omnidirectional cameras. In *Technical Sketches, Computer Vision and Pattern Recognition*, December 2001.
- [11] K. Nummiaro, E. B. Koller-Meier, and L. Van Gool. An adaptive color-based particle filter. *Journal of Image and Vision Computing*, 21(1):99–110, 2003.
- [12] Tomáš Svoboda. Quick guide to multi-camera self-calibration. Technical Report 263, Computer Vision Lab, Swiss Federal Institute of Technology, Zurich, July 2003. <http://www.vision.ee.ethz.ch/~svoboda/SelfCal>.
- [13] Tomáš Svoboda, Hanspeter Hug, and Luc Van Gool. ViRoom — low cost synchronized multicamera system and its self-calibration. In Luc Van Gool, editor, *Pattern Recognition, 24th DAGM Symposium*, number 2449 in LNCS, pages 515–522. Springer, September 2002.
- [14] Mohan M. Trivedi, Ivana Mikic, and Sailendra K. Bhonsle. Active camera networks and semantic event databases for intelligent environments. In *IEEE Workshop on Human Modeling, Analysis and Synthesis (in conjunction with CVPR)*, June 2000.

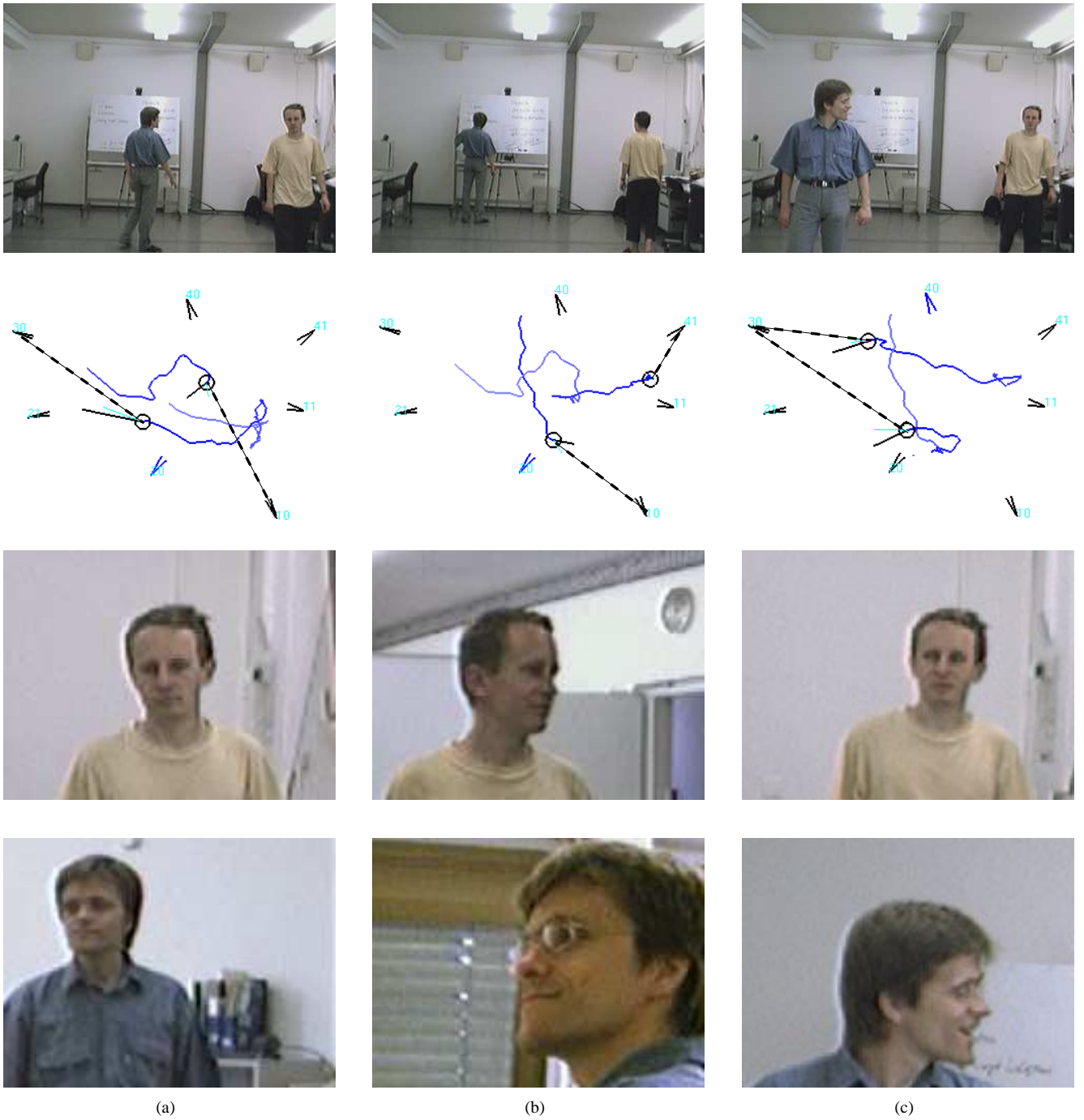


Figure 11: The columns represent frames 287 (a), 365 (b) and 435 (c) in the two person sequence. First row shows an overall view captured by the camera 30. The second row shows the top-view sketch with the current positions of the objects and their paths with predicted displacement. Dashed line links the person with the selected camera. Two bottom rows contain close-ups of both persons selected by the system.