Real-time Detection and Interpretation of 3D Deictic Gestures for Interaction With an Intelligent Environment

Jan Richarz, Thomas Plötz and Gernot A. Fink
Dortmund University of Technology, Robotics Research Institute, Dortmund, Germany
{Jan.Richarz,Thomas.Ploetz,Gernot.Fink}@udo.edu

Abstract

We present a system that enables pointing-based unconstrained interaction with a smart conference room using an arbitrary multi-camera setup. For each individual camera stream, areas exhibiting strong motion are identified. In these areas, face and hand hypotheses are detected. The detections of multiple cameras are then combined to 3D hypotheses from which deictic gestures are identified and a pointing direction is derived. This is then used to identify objects in the scene. Since we use a combination of simple yet effective techniques, the system runs in real-time and is very responsive. We present evaluation results on realistic data that show the capabilities of the presented approach.

1. Introduction

The ”intelligence” of a smart room is often understood as the amount of technology that is built in. The more electronic devices are present and the more services it offers, the more ”intelligent” it is. We argue that a smart room will only be recognized as smart if its services can be accessed and controlled in an intuitive and natural manner. Therefore, in our smart environment ”FINCA” [9], we focus on man-machine interfaces that are as intuitive as possible. One of our goals is the development of a gesture-based interaction module.

In this paper, we present a real-time[1] multi-camera system that is able to reliably detect moving persons and locate their faces and hands. This is a major prerequisite for unconstrained vision-based 3D gestural interaction. In the current application, we use the system to recognize deictic gestures and infer a 3D pointing direction from them, which can then be used to control certain room functions (e.g. switching lights) or to identify referred objects. Our system combines several simple techniques to achieve robust detection. It is fast and responsive and can be used with arbitrary camera configurations. It also does not need any prior training, all necessary information is extracted on-line.

The remainder of the paper is organized as follows: First we shortly review related literature. Then, we present the architecture of our system followed by an experimental evaluation on realistic data. We conclude with a short summary.

2. Related work

While the development of gesture interfaces for human-machine interaction has gained great interest in recent years, there are surprisingly few projects that deal with the explicit recognition of pointing directions from deictic gestures. Often, gestural interaction is realized via the recognition of a fixed gestural ”control alphabet” of hand or body postures [6] or sign language recognition [8]. Dynamic gestures – i.e. spatio-temporal motion patterns of the body – are also frequently used in action recognition and surveillance systems [4] where the goal is not to control an interface, but to detect certain incidents in an observed scene.

Given the task to recognize object or environment references from deictic gestures, a reference can be defined via spatial proximity [3] or via explicit calculation of a pointing direction, which requires an accurate detection of body parts. A common technique is to fit a detailed body model to the data [5], which, in general, is computationally expensive.Therefore, some approaches use simplified models [7] or first reduce the amount of 2D data from different camera streams and then combine the results [11]. The latter is the approach we pursue in this paper. Our goal is to explicitly derive a 3D pointing direction without constraining the application to a certain camera setup and without imposing restrictions on the user. This also means that no auxiliary techniques like markers or tracking gloves are utilized.

3. System architecture

In order to efficiently detect deictic gestures, we do most of the computation on the individual 2D image streams. This greatly reduces the amount of 3D data that has to be processed. Furthermore, this process can easily be parallelized, yielding a system that can cope

[1] Requirement: Reaction to user actions within 1-2 seconds

with an arbitrary number of cameras. The first step is to detect areas of strong motion, to which then a face detector is applied. If a face is found, we extract a skin color description and combine it with the motion information to find hand candidates. These are then clustered, and the results of the image streams are combined in 3D using a ray intersection technique. We then identify deictic gestures based on contextual information and derive a pointing direction from them.

Motion detection: In our scenario, it is reasonable to assume a mostly static environment in which all moving objects are of potential interest. However, since our cameras are potentially moving, the system must be able to quickly adapt to new situations. Furthermore, for stability reasons, we want image regions exhibiting strong motion to remain marked as ”interesting” for some time (even if they become static again). Consequently, static background subtraction is not suitable.

The approach we use is similar to the Motion History Images of Bobick and Davis [1].We keep a reference image $R_t(x, y)$ of the scene as model. For each new frame, a pixel-wise difference image $D_t(x, y)$ to the model is computed. This is then used in two ways. First, it is used to calculate the motion map M:

$$M_t(x,y) = \left\{ \begin{array}{rcl} \tau & : & D_t(x,y) > \theta \\ f(E_t(x,y)) & : & \text{otherwise} \end{array} \right.$$
$$\text{with} \quad E_t(x,y) = E_{t-1}(x,y) + 1$$

where $M_t(x, y)$ is the motion map pixel value at time t, τ the maximum motion value, θ a threshold, f an arbitrary decay function and E a pixel-wise decay counter which is reset to zero if the associated pixel is detected as foreground. Thus, we do not obtain a binary foreground-background segmentation, but a continuously-valued motion saliency map (see Fig. 1). A side-effect is that the response to motion can be tuned via the decay function. Second, D is used to update R with a given update rate. The effect is that, on the one hand, strong motions ”linger” in the saliency map for some time while, on the other hand, the model image is slowly adapted to match the scene. Thus, we can still detect connected regions for some frames if parts of the region ceased to move while foreground objects that do not move for a longer period of time will be learned to be part of the background. This mechanism also yields robustness against moderate lighting changes, since the system will adapt to them within a few frames.

In the resulting motion saliency map M, regions of interest (ROI) are identified using a quad-tree decomposition. While traversing the tree, the average saliency in the area defined by the current node is calculated. Since the nodes define rectangular subimages, this can be done very efficiently using an integral image of the input. Traversal of a branch is aborted when a node with an average saliency above a threshold is found, and this node is added to the ROI list. Adjacent ROIs are then merged to form bounding boxes around areas with strong motion. This algorithm is very fast and is able to detect an arbitrary number of ROIs of arbitrary sizes.

Face and hand detection: The largest ROI is kept for further processing, thus we assume that only one person is present. However, multiple separate ROIs could be processed in parallel to overcome this limitation. Persons occluding each other remain problematic, though.

We apply two Viola-Jones detectors [10] (frontal and profile) to the ROI to detect the person’s face. If multiple face hypotheses are obtained, the one with the highest combined motion/skin saliency in its detection region is chosen. If no face has been found, computation is aborted for this frame, unless a skin color histogram with a high trust value (see below) is available. In this case we use the color information to find likely face positions near the last known position.

Given the face, a HSV color histogram of the person’s skin color is computed. To be more robust against detection errors, we only take into account pixels inside an elliptic area smaller than the face and additionally weigh them with a Gaussian. Furthermore, a weighted average between the new and the old histogram is computed in each step which smooths temporal variations of the histogram bins. We also assign a trust value to the histogram which is increased if the new and old histograms are similar, and decreased otherwise.

Using the smoothed histogram, we compute a skin likelihood map and combine it (using weighted average) with the motion saliency. We then slide a rectangular window over the resulting combined saliency to find hand candidates. All candidates with an average saliency greater than a threshold and within 80% of the maximum value are kept. Since this procedure typically yields numerous overlapping candidate regions on hands, we apply a Mean Shift clustering algorithm [2] to reduce clusters to single detections.

3D projection and combination: So far, all computations have been carried out on the individual 2D image streams. The next step is to combine the detected faces and hands into a 3D representation. Note that the image streams are not synchronized. Given the cameras’ internal parameters and 3D room positions, we calculate rays from the cameras’ optical centers through the detection centers in the respective images using simple projective geometry. Rays belonging to the same object should intersect at the object’s 3D position. However, due to detection errors and the unsynchronized

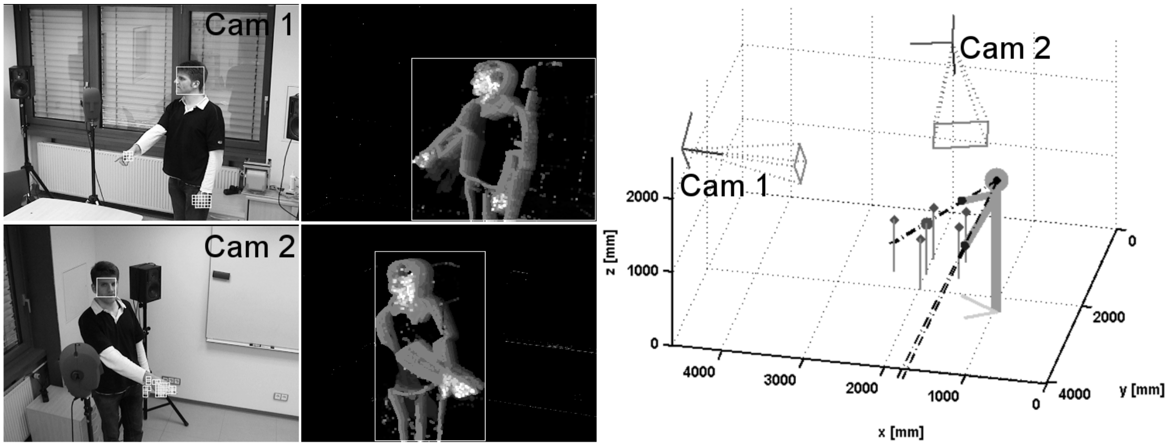

Figure 1. Left to right: input images with face and hand detections, motion/skin saliency with ROI, 3D projection result. Estimated pointing directions are depicted by dotted black lines, the dots visualize the marker positions. The referred marker is shown with a bigger dot.

image streams, the rays will generally be skewed. So we seek combinations that have minimum mutual distance. We calculate the connecting perpendicular line for each combination and interpolate the 3D position of the corresponding detection as the center point of this line. Rays from false detections will normally not yield a valid combination because their distance to all other rays will be too large. Fig. 1 shows an example of the 2D detections and the resulting 3D representation.

The 3D pointing direction is defined by the line connecting face and hand. We do not explicitly detect deictic gestures (although we plan to do so in future work), but recognize them context-based. Normally, inside the FINCA, the context is given via a speech command. For this paper, we define context via object references. If, in a given temporal window of 8 frames, an object reference occurred (i.e. the pointing ray intersects a scene object) at least 2 times, we infer that this object was pointed at. If different object references are detected, the one that has the majority of entries is selected.

4. Evaluation

For the sake of clarity we evaluated the 2D and 3D stages separately. For the 2D stage, we recorded short image sequences inside the FINCA each showing one person gesticulating unconstrainedly in the camera's field of view. The images are 378 by 278 pixels and were recorded at system frame rate (approx. 6 to 8 fps on a standard dual-core desktop with an unoptimized threaded implementation). We used 3 different cameras showing substantially different views of the room and recorded 10 different persons on different days and under varying lighting conditions. This set contains 3053 images that were annotated manually.

To evaluate the quality of the 3D reconstruction, we performed a pointing experiment where 6 different people were asked to point at 6 markers placed approx. 40 cm from each other on a table. 2 different camera setups were used. This set contains 2342 image pairs.

2D stage: The data was labeled using rectangles around the inner face area and enclosing visible hands. A face detection is considered valid if its bounding box entirely contains the annotation and lies inside a rectangle with 3 times the size of the annotation. A hand detection is valid if its center lies inside a labeled hand area. A hand is counted as found if at least one valid detection is present that belongs to it, otherwise it is reported as missed. The same holds for faces. False positives are detections that cannot be assigned to any of the annotated areas.

Table 1 shows the results. ”Faces” and ”hands” denote the total number of annotated faces and hands that were detected by the algorithm. The hand detection stage largely depends on the success of the face detection. In order to assess the performance of this stage, we present the scores calculated only for those images in which the face was detected correctly (”without face detection errors”). Also shown are the hand detection results after clustering has been applied to the candidate regions (”clustered”). Note that clusters with less than 2 supporting detections are omitted. The total number of correct and false hand hypotheses (”hand hyp.”, i.e. candidate regions) is shown in the bottom part. The majority of false hand hypotheses originates from only very few frames. Overall, there are 69 images with more than 10 false positives. So, on average we get one defective frame result every 43 frames.

Overall, the results are satisfactory. Note that most missed hands have either been idle for several frames

Table 1. Detection results for 2D stage.

	annotated	found (%)	missed (%)
faces	3008	2334 (77.6)	674 (22.4)
hands	5806	3850 (66.3)	1956 (33.7)
without face detection errors			
hands	4540	3546 (78.1)	994 (21.9)
clustered	4540	3405 (75.0)	1135 (25.0)

	total	correct (%)	false pos. (%)
hand hyp.	24339	18625 (76.5)	5714 (23.5)
without face detection errors			
hand hyp.	19858	17229 (86.8)	2629 (13.2)
clustered	3665	3406 (92.9)	259 (7.1)

Table 2. Detection results for 3D stage

	annot.	correct (%)	false (%)	missed (%)
obj	132	94 (71.2)	18 (13.6)	20 (15.2)
ref	1144	506 (44.2)	168 (14.7)	470 (41.1)
With detected head position:				
obj	114	94 (82.5)	18 (15.8)	2 (1.7)
ref	841	506 (60.1)	168 (20.0)	167 (19.9)

(and therefore very likely do not carry any information), are in front of the face (where no hand candidates are searched for obvious reasons) or are shadowed by the body. The main gesticulating hand is found quite reliably and the system can cope with single missed detections in between.

3D stage: Since it is not possible to infer precise ground truth data for human pointing gestures due to their approximative and person-dependent nature, we cannot report absolute precisions. Instead, we present overall detection rates where a detection is valid if the person was pointing at that moment and the correct marker is identified. The weak point in the whole detection process is the face detector, which has difficulties dealing with in-plane rotation and tilting of faces. Also, its detections vary in size and position and therefore yield inaccuracies in the head position estimates which limit the achievable accuracy of the system and are difficult to assess. Since our goal is to evaluate the pointing estimation process (not the face detector), we choose from a large dataset those sequences where the face detector yields satisfactory performance, resulting in the testset described above. The results are shown in table 2. The object detections (where a complete pointing sequence belonging to a certain marker is counted as one object) is reported as ”obj”, ”ref” are frame-wise object references (i.e. single detections). The misses are mainly due to head detection failures (as can be seen in the bottom part of table 2 where we omit the frames in which no head was found). The system yielded 223 false positive references, mainly due to smooth transitions between gestures. These can rather easily be eliminated by incorporating gesture segmentation.

5. Summary

We presented a multi-camera system that is able to infer the 3D positions of a person’s face and hands in real-time from almost arbitrary camera configurations (limited by the requirements of the face detector, which will be replaced in future versions). It, thus, enables marker-less and unconstrained pointing-based interaction with an intelligent environment. Our experimental results show that the system achieves reliable detection, and we successfully utilized it to identify scene objects via deictic gestures. Note that, once a direction is derived, it can be used for multiple purposes, i.e. controlling room services or directing the room’s attention to a certain area. So far, only settings with 2 cameras were evaluated, but the extension to multi-camera scenarios is straightforward.

References

[1] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Trans. PAMI*, 23(3):257–267, 2001.

[2] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. PAMI*, 24(5):603–619, 2002.

[3] N. Hofemann et al. Recognition of deictic gestures with context. In *Proc. 26th DAGM Symposium, LNCS*, vol. 3175, pp. 334–341, 2004.

[4] W. Hu et. al. A survey on visual surveillance of object motion and behaviors. *IEEE Trans. Systems, Man and Cybernetics*, 34(3):334–352, 2004.

[5] R. Kehl et al. Full body tracking from multiple views using stochastic sampling. In *Proc. IEEE Conf. CVPR*, pp. 129–136, 2005.

[6] R. Lockton and A. W. Fitzgibbon. Real-time gesture recognition using deterministic boosting. In *Proc. British Machine Vision Conf.*, pp. 817–826, 2002.

[7] K. Nickel and R. Stiefelhagen. Real-time person tracking and pointing gesture recognition for human-robot interaction. In *Computer Vision in HCI, ECCV Workshop on HCI, LNCS*, vol. 3058, pp. 28–38, 2004.

[8] S. Ong and S. Ranganath. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Trans. PAMI*, 27(6):873–891, 2005.

[9] T. Plötz. The FINCA. http://www.finca.irf.de.

[10] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. CVPR*, pp. 511–518, 2001.

[11] C. Wu and H. Aghajan. Model-based human posture estimation for gesture analysis in an opportunistic fusion smart camera network. In *Proc. Int. Conf. Advanced Video and Signal based Surveillance*, pp. 453–458, 2007.