Computational Intelligence, Volume 23, Number 4, 2007

RECOGNIZING SIMPLE HUMAN ACTIONS USING 3D HEAD MOVEMENT

JORGE USABIAGA, GEORGE BEBIS, ALI EROL, AND MIRCEA NICOLESCU

Computer Vision Laboratory, University of Nevada, Reno

MONICA NICOLESCU

Robotics Laboratory, University of Nevada, Reno

Recognizing human actions from video has been a challenging problem in computer vision. Although human actions can be inferred from a wide range of data, it has been demonstrated that simple human actions can be inferred by tracking the movement of the head in 2D. This is a promising idea as detecting and tracking the head is expected to be simpler and faster because the head has lower shape variability and higher visibility than other body parts (e.g., hands and/or feet). Although tracking the movement of the head alone does not provide sufficient information for distinguishing among complex human actions, it could serve as a complimentary component of a more sophisticated action recognition system. In this article, we extend this idea by developing a more general, viewpoint invariant, action recognition system by detecting and tracking the 3D position of the head using multiple cameras. The proposed approach employs Principal Component Analysis (PCA) to register the 3D trajectories in a common coordinate system and Dynamic Time Warping (DTW) to align them in time for matching. We present experimental results to demonstrate the potential of using 3D head trajectory information to distinguish among simple but common human actions independently of viewpoint.

Key words: human action recognition, 3D head movement, principal component analysis, dynamic time wrapping

1. INTRODUCTION

Human action recognition is a challenging problem with many important applications in video surveillance and human-computer interaction. There has been significant research over the last few years dealing with the problem of recognizing human actions from video. In general, complex spatio-temporal changes in dynamic environments must be processed and analyzed for inferring useful information. Moreover, a decision must be made whether a system would utilize 2D or 3D information. When studying the representation and recognition of signals in time, two techniques always show up in the literature: DTW and Hidden Markov Models (HMMs) (Oates, Firoiu, and Cohen 1999).

Gavrila and Davis (1995) have presented a system where several actions, including *waving hello, waving-to-come*, and *twisting*, were recognized by recovering 3D body pose using multiple cameras. Matching was performed using DTW. Rao, Yilmaz, and Shah (2002) have proposed a view-invariant representation and recognition approach for action recognition. Their system characterizes actions using *dynamic instants* and *intervals*. *Dynamic instants* are maxima of the spatio-temporal curvature, while *intervals* are the segments of a trajectory between *dynamic instants*. To extract the *dynamic instants* reliably, they employ anisotropic diffusion for smoothing the trajectories. Actions are first classified according to the number of instants and their sign sequence, while recognition is performed by calculating the rank of the observation matrix (Tomasi and Kanade 1993). Later on, Rao et al. (2003) proposed an approach for view-invariant alignment and matching of video sequences. Their system is capable of aligning video sequences obtained by different cameras at different time instances and employs DTW with a cost function based on epipolar geometry.

In Syeda-Mahmood, Vasilescu, and Sethi (2001), a region-based approach to human activity recognition was presented using only one camera. The idea was modeling actions in terms of *action cylinders* which were built by combining shape and motion information. Because action cylinders are not view-invariant in general, new actions are recognized by recovering the viewpoint transformation and treating the problem as model-based object recognition. A real-time, Bayesian approach to modeling human interactions was presented by Oliver, Rosario, and Pentland (2000). Each action model was built using synthetic data while an eigen-background approach and a Kalman filter were used to detect blobs corresponding to the human body and track them over time. To model human interactions, they combined a top-down with a bottom-up approach, both based on Coupled HMMs, using a closed feedback loop. More recently, Robertson and Reid (2006) presented a general system for human activity recognition where actions were described using trajectory information and motion descriptors. Human behavior was modeled as a stochastic sequence of actions using HMMs.

In the past, it has been demonstrated that simple human actions can be inferred by tracking the position of the head only. Specifically, Madabhushi and Aggarwal (1999) have demonstrated that head movement provides useful information for distinguishing among several simple but common human actions such as sitting-down, standing-up, walking, hugging, etc. The key idea is that the head moves in a characteristic way when humans perform certain actions. Using two cameras (i.e., frontal and lateral), they built a system that was able to recognize some simple actions using probabilistic models. Besides being viewpoint dependent, their system was based on a number of heuristics and assumptions that do not usually hold in practice (i.e., motion in the x-direction is independent of motion in the y-direction). Moreover, the position of the head was determined manually from frame to frame. A related approach appeared in Nait-Cherif and McKenna (2003) where head trajectory information was used to recognize simple actions in a meeting room. This system was also viewpoint dependent.

In general, detecting and tracking the head is simpler and faster than detecting and tracking other body parts like the hands/or the feet. This is mainly because the head has low shape variability and more visibility. This is especially true in surveillance applications where cameras are mounted at a high position, making it less likely for the head to be occluded. However, using head movement alone should not be expected to provide enough information for distinguishing among complex human actions. Therefore, the role of head movement should be considered mostly as complimentary rather than stand-alone, for example, it could be combined with 3D human pose information (Bowden, Mitchell, and Sarhadi 2000).

In this article, we extend the work of Madabhushi and Aggarwal (1999) by proposing a viewpoint invariant action recognition system based on 3D head trajectory information. Specifically, the main weakness of the approach in Madabhushi and Aggarwal (1999) is that it can recognize an action from two viewpoints only which limits its practical value. In contrast, our approach is capable of recognizing simple actions, independently of viewpoint, by tracking the position of the head in 3D using multiple cameras. Another difference between the approach in Madabhushi and Aggarwal (1999) and our approach is in the recognition stage. In Madabhushi and Aggarwal (1999), each action was modeled using probabilistic models, however, each a small number of training data was used to estimate the parameters of each model. Moreover, recognition was performed using Bayesian classifiers, under the restrictive assumption that each action can not take more than a fixed number of frames. In contrast, our recognition strategy is based in DTW which does not put any upper bounds on the number of frames needed to perform an action. By aligning different samples of the same action in time, we model each action by simply taking the average over all the samples of the same action. We demonstrate the proposed approach on a long video sequence containing multiple actions, without assuming that different actions have been pre-segmented.

Tracking and recovering the position of the head in 3D has received a lot of attention. In general, a head model is employed which could be purely geometrical or might contain additional information such as color and texture. In Brown (2001), the 3D position of the head was tracked by rendering a texture-mapped cylinder. Face detection was employed to re-initialize the tracker when it failed. A related approach was proposed in Cascia, Sclaroff, and Athitsos (2000). A 3D head tracking algorithm, which is robust partial occlusions, was proposed in Zhang and Kambhamettu (2002). An advanced geometric model was used to provide a better approximation of facial shape. In Terada, Oba, and Ito (2005), the head was modeled as a polygon mesh along with statistical color information from the skin and the hair. A particle filter was used to track the head in 3D using depth information recovered by a stereo camera. A cylindrical head model was employed in Kwon, Chun, and Park (2006). In Ohayon and Rivlin (2006), head geometry was modeled using a sparse set of 3D points which were acquired prior to tracking. By solving a camera pose estimation problem, the 3D position of the head was recovered. In Birchfield (1998), an elliptical head model was employed along with gradient and color information. A similar model was employed in Nait-Cherif and McKenna (2003) for head tracking and action recognition in a smart meeting room. A modified particle filter algorithm was employed for tracking.

With the exception of Birchfield (1998) and Nait-Cherif and McKenna (2003), the above methods are mostly appropriate for applications where the head is assumed to be close to the camera (e.g., human-computer interaction). In our application, we cannot make such an assumption, therefore, a detailed head model cannot be used. To demonstrate our approach, we have adopted the elliptical head model introduced in Birchfield (1998). To reduce noise, we smooth the 3D head trajectories using a Kalman filter with constant velocity Welch and Bishop (1995). It should be mentioned that more sophisticated approaches could be used for head modeling and tracking, however, our main objective in this study is to demonstrate the idea of employing 3D head movement for simple human action recognition.

The rest of this article is organized as follows. In Section 2, we present the experimental setup of our system. Section 3 describes the steps for recovering the 3D trajectory of the head while Section 4 presents the steps for modeling different actions. The procedure to recognize novel actions is given in Section 5. Our experimental results are presented in Section 6. Finally, our conclusions and directions for future work are presented in Section 7.

2. SYSTEM SETUP

Our experimental setup involves using multiple cameras to capture the location of the head from different viewpoints and estimate its 3D position. Given a sequence of frames corresponding to a specific action, the 3D trajectory of the head can then be extracted for action modeling and/or recognition purposes. Specifically, we used three Videre cameras to collect the training sequences to build a action model for each action. For testing, however, we used four Dragonfly cameras to allow the subjects more freedom in their movements but also to demonstrate that the proposed approach works well with different number and type of cameras. In general, a larger number of cameras can be used both for training and testing, depending on the environment and the application requirements. All cameras were calibrated and synchronized.

Although many camera models today provide an option for automatic synchronization, multiple camera calibration could be challenging depending on the application. One way to simplify this task is by assuming that all cameras see a common plane (Kanade, Rander, and Narayanan 1997). In this case, the position and orientation of the cameras with respect to the common calibration plane can be recovered easily. Then, calculating the relative position and orientation between cameras is straightforward. In our experimental environment, all the cameras were mounted high enough on the wall and tilted downward. This camera placement is very common for many surveillance applications. To calibrate the cameras, we placed a

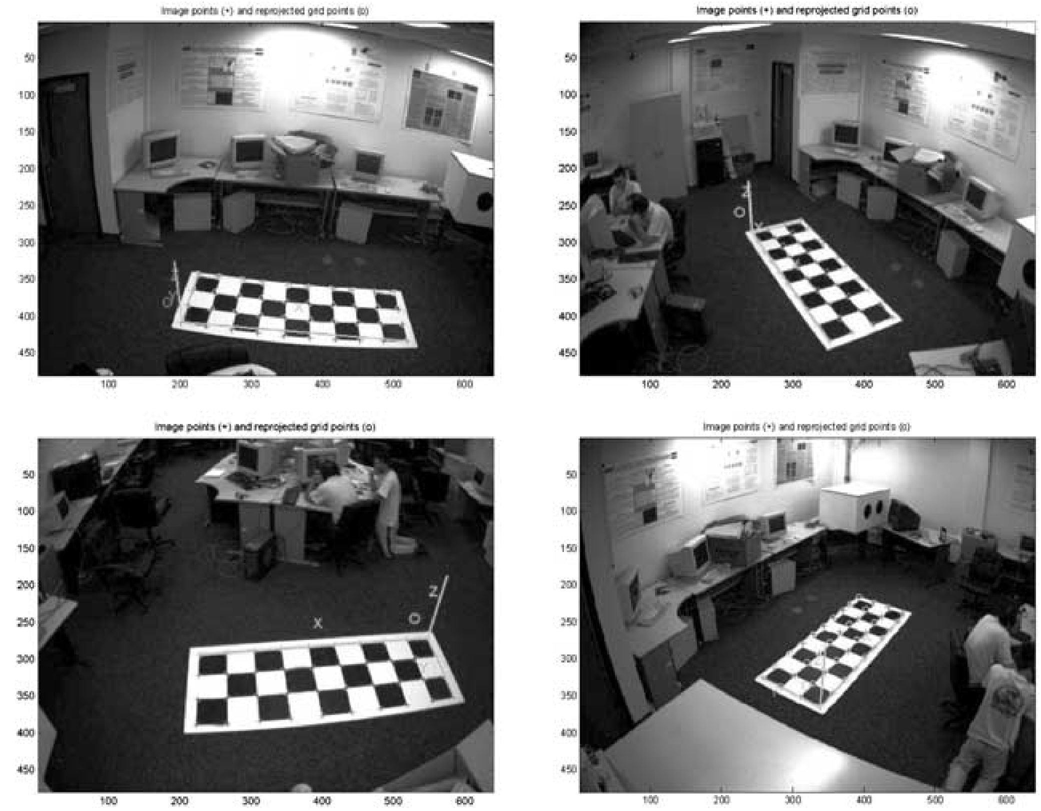


Image points (+) and reprojected grid points (o)
Image points (+) and reprojected grid points (o)
Image points (+) and reprojected grid points (o)
Image points (+) and reprojected grid points (o)
50
100
150
200
250
300
350
400
450
100
200
300
400
500
600
X
Z
O

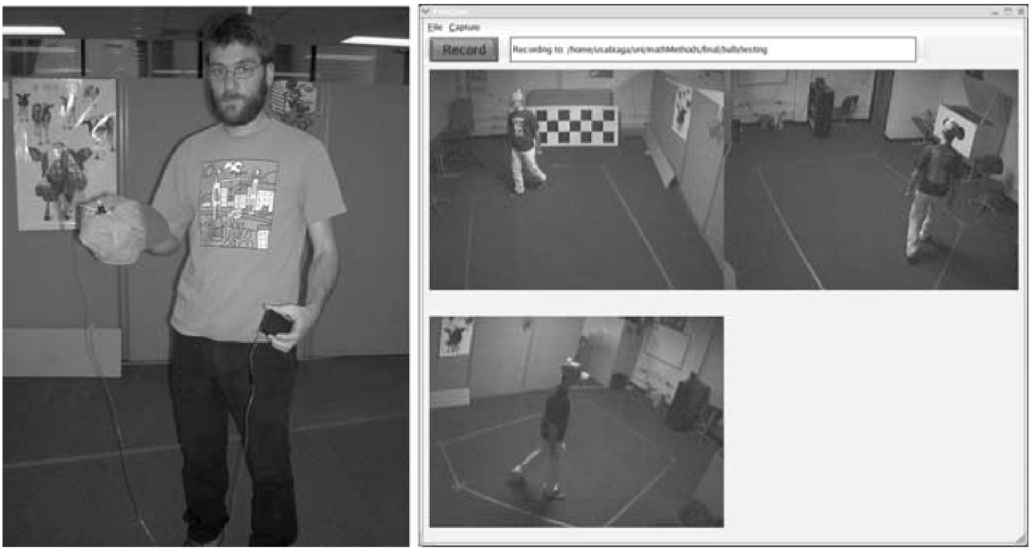
FIGURE 1. Illustration of setup for multiple camera calibration.

calibration board on the floor in such a way that it was visible by all cameras (i.e., see Figure 1). To compute the extrinsic and intrinsic camera parameters, we used Matlab's Calibration Toolbox, which is based on the calibration algorithms proposed by Zhang (2000) and Heikkila and Silven (1997).

3. 3D HEAD TRAJECTORY EXTRACTION

The first step in recovering the 3D trajectory of the head involves detecting its 2D location in each camera. Because the cameras were all synchronized, each 2D head location corresponds to the same physical 3D location of the head. To estimate the 3D location of the head from the corresponding 2D head locations, we use triangulation. To reduce reconstruction errors, we apply triangulation using each pair of cameras and fuse the estimates.

To evaluate the power of 3D head trajectory information for discriminating between different actions, special care was placed in building the action models without introducing head localization errors. Specifically, to detect the location of the head in each camera robustly and reliably during training, subjects were required to wear a hat with a lightbulb attached to it as shown in Figure 2. To detect the position of the head in each frame, we simply segment the position of the lightbulb by thresholding the intensity value. In all of our experiments, we used a fixed threshold equal to 200. To estimate the center of the lightbulb more reliably, we fit a 2D Gaussian and take the location having the maximum value as the true center of



Recording to: /home/ucabiaga/uni/mathMethods/final/bulb/testing

FIGURE 2. Detecting and tracking the head reliably to build accurate action models: (a) lightbulb attached to a hat, (b) subject wearing the hat for collecting the training sequences.

the lightbulb. It should be mentioned that during testing, we did not impose any restrictions on the subjects and the head was detected and tracked automatically using an elliptical head tracker (see Section 5).

To recover the 3D position of the head, we apply triangulation by considering pairs of cameras, assuming that the head is visible in both cameras (Trucco and Verri 1998). In our implementation, we consider all possible camera pairs. Because each camera pair yields an estimate of the true 3D position, the final estimate is obtained by averaging the individual 3D estimates as shown in Figure 3. The 3D trajectory of the head is then obtained by combining the 3D positions of the head over time.

4. ACTION MODELS

To model each action, we collected a number of training data (i.e., sequences) for each action by asking different subjects to perform the action several times. Then, each training sequence was processed to extract the 3D trajectory of the head as described in Section 3. The final model for each action was built by appropriately preprocessing the 3D trajectory training samples of that action and averaging them. Preprocessing involves the following three steps: (i) registering the 3D head trajectories by representing them in a common coordinate system, (ii) aligning them in time using DTW, and (iii) normalizing them.

4.1. Registration and Normalization of 3D Head Trajectories

Before combining the 3D trajectory training samples of an action into a single model, the samples must be registered first by rotating and translating them into a common coordinate system. Figure 4 illustrates the *sitting down* action performed by the same subject in two different ways, each time starting and ending at different positions and facing different directions. Registering the trajectories would be necessary to average them.

To simplify registration, it can be observed that the z direction (i.e., normal to the floor) is always common to all actions, as far as the subjects performing an action step on the ground. Therefore, registration is required in the $x - y$ plane only. To implement this idea, we

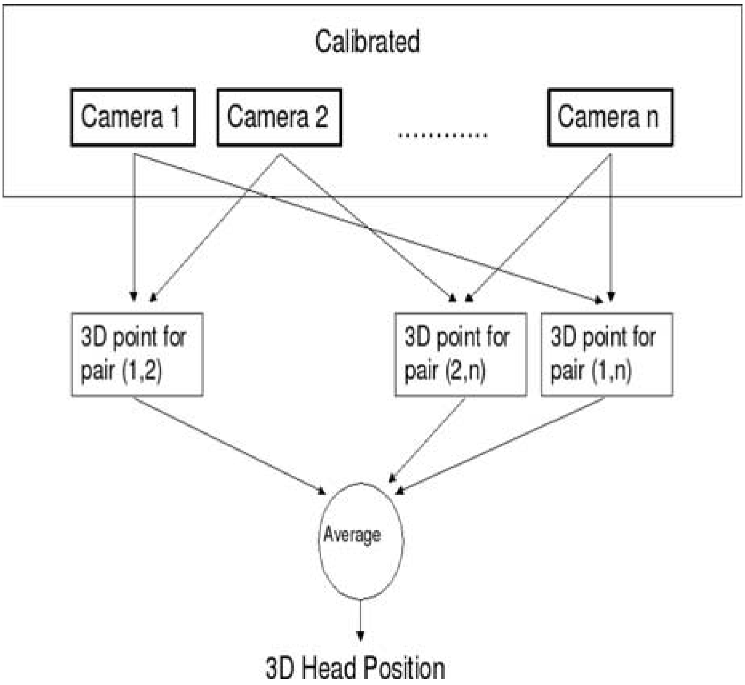


FIGURE 3. Estimating the 3D position of the head by reconstructing the 3D position of the head from every pair of cameras and averaging the results.

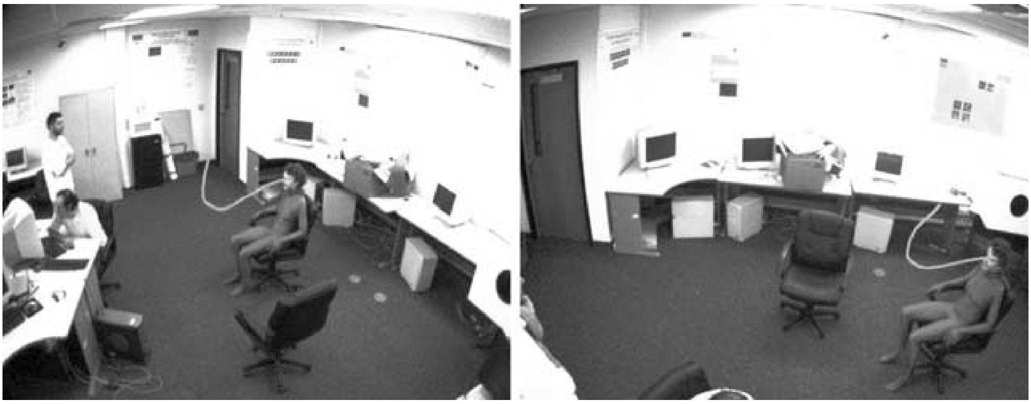


FIGURE 4. Two instances of the *sitting down* action performed at different locations of the room and facing different directions.

represent each action in a new coordinate system which is built by using PCA (Jain, Duin, and Mao 2000). The resulting PCA space contains only two principal components, x' and y', both perpendicular to z axis. Therefore, each training trajectory is represented in the $x' - y' - z$ space, which is rotation and translation invariant (i.e., centered at the first point of each trajectory), by projecting its x and y coordinates in the PCA space. Obviously, the direction of 3D trajectories (i.e., distinguishing between the starting and ending points of a trajectory) would be necessary to distinguish actions having similar shape (i.e., "sitting-down" and "sitting-up" actions where one is almost the reverse of the other).

4.2. Alignment of 3D Trajectories

Because different subjects can perform the same action in different speeds, it is also imperative to align the samples of each action in time before combining them into a single model. In this study, we employed DTW (Oates, Firoiu, and Cohen 1999) for aligning the actions in time. DTW is a well-known method in speech recognition for aligning two sequences with local differences in time. DTW works by allowing for locally stretching or shrinking a signal in time, to find the best alignment between two sequences. This has been shown to be far more superior than matching two sequences using simple distance measures such as the Euclidean distance.

Specifically, the purpose of DTW is to find the best alignment between two sequences, which is equivalent to finding the path starting at the first frame of each sequence to the last frames by minimizing a distance. The success of DTW depends on how the distance is computed. Instead of comparing the value of the candidate sequence at time t with the query at time t, it is calculated with respect to the values of the query in a time window centered at t.

DTW can be implemented efficiently using dynamic programming. First a table containing all the distances between frames of the two sequences is built. Then, a decision about the shortest path is made based on this table. The path minimizing the distance between two sequences gives the proper alignment and the distance is the measure of similarity. The table of distances for a query Q and a candidate C is build by computing for each element of each trajectory the following function:

$$\gamma(i, j) = d(q_i, c_j) + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\} \tag{1}$$

where d is a distance measure (e.g., Euclidean distance) and $\gamma(i, j)$ represents the error of aligning signals up to time t_i and t_j. Using a brute force approach, the table of distances is built for all possible pairs of frame-to-frame correspondences. However, this is not always necessary and some global constraints can be enforced to bound the best path search. In this work, we have adopted the Itakura band (Keogh et al. 2003) to apply some global constraints. We assume that the trajectories are segmented in time such that the first and last frames are already aligned. The resulting model has the same length (i.e., contains the same number of frames) as the training sequences.

4.3. Normalization of 3D Trajectories

After the 3D head trajectories have been registered and aligned in time, we normalize them so that the variation in the z' and x' directions are equal. Figure 5 shows several samples of the "sitting-down" action without any registration.

4.4. Action Models

To build a model for each action, first we collected several training sequences by having a number of subjects perform each action several times. Then, for each action, we used the longest training sequence as an initial template and aligned the rest training sequences of the same action with the template using DTW. The final model for each action was obtained by taking the average over all normalized trajectories of the action.

It should be mentioned that since we used the longest training sequence as an initial template for DTW, the action models look similar to the corresponding templates. Figures 6 and 7 show representative training samples and corresponding action models for the following four actions: *sitting-down, bending-down, squatting*, and *rising from squatting*. Each figure

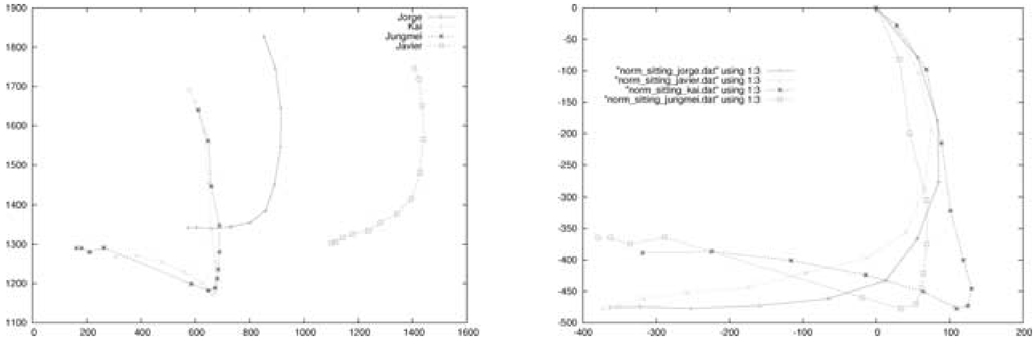
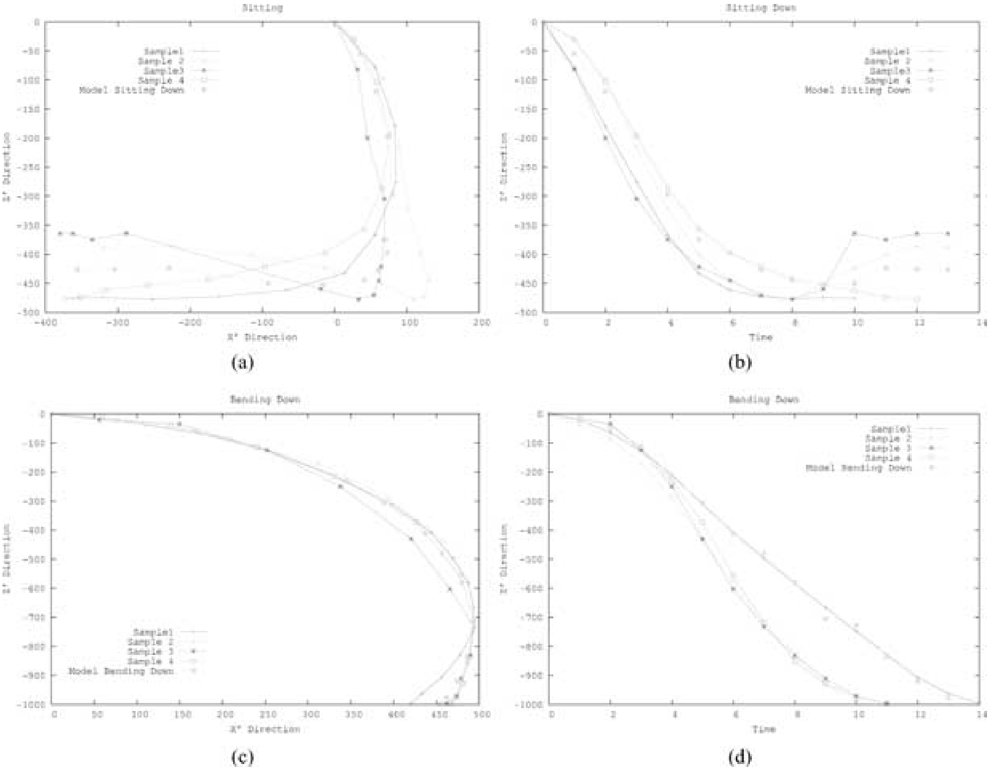


FIGURE 5. Representative trajectories corresponding to the "sitting down" action (a) $x - z$ plots, (b) $x' - z$ plots (i.e., registered, aligned, and normalized).



(a)

(b)

(c)

(d)

FIGURE 6. Representative sitting-down training samples and action model: (a) x' vs z, (b) time vs z; Representative bending-down training samples and action model: (c) x' vs z, (d) time vs z.

shows plots of the z component against both x' and time. Figure 8 shows all four models in the same plot for comparison purposes.

5. ACTION RECOGNITION

Given a new input sequence corresponding to some action, first we process the data to detect the head in each frame, estimate its 3D position, and finally recover the 3D head

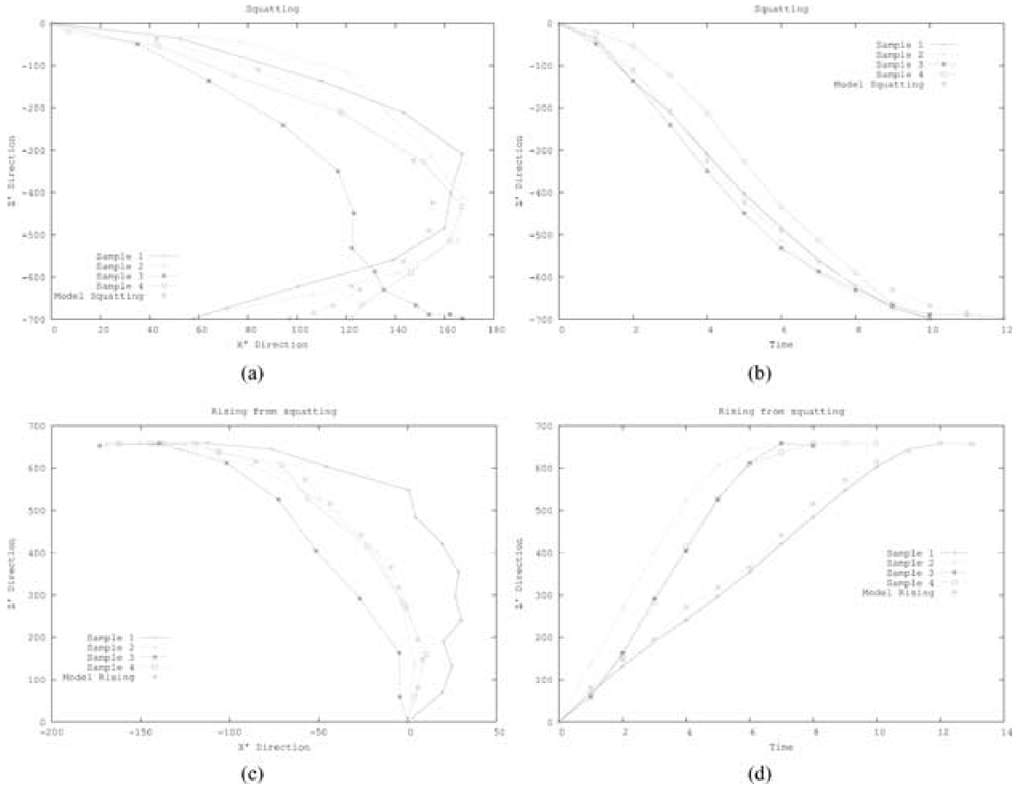


FIGURE 7. Representative squatting training samples and action model: (a) x' vs z, (b) time vs z; Representative rising-from-squatting training samples and action model: (c) x' vs z, (d) time vs z.

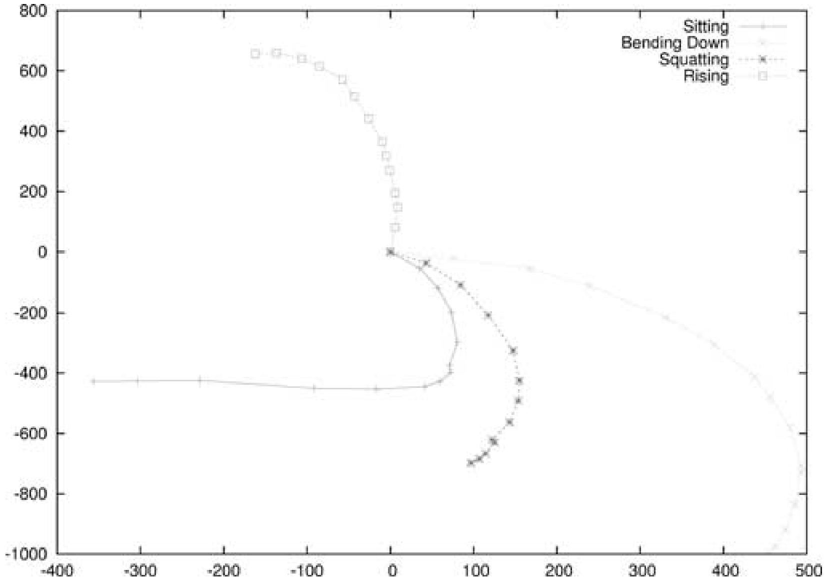


FIGURE 8. The four action models plotted in the x' vs z directions.

trajectory. In contrast to training where subjects have to wear a hat to extract the trajectory of the head very accurately, the head was detected automatically during testing using an elliptical head tracker (Birchfield 1998). To reduce noise, we smoothed the 3D trajectories using a Kalman filter with constant velocity (Welch and Bishop 1995).

Once the test 3D head trajectories had been filtered, they were compared against each of the action models to find the closest match. This step involves using the same steps as in training, that is, registering and aligning an input action with each of the action models as well as normalizing it. To account for differences due to performing an action at different speeds, additional effort is required (see next section).

6. EXPERIMENTAL RESULTS

In this section, we demonstrate the feasibility of the proposed approach by considering a small number of actions: *sitting-down*, *bending-down*, *squatting*, and *rising from squatting*. For demonstration, we have only considered the inverse action for "squatting" only (i.e., "rising from squatting"). This is because inverse actions have symmetrical trajectory shapes and can be recognized in the same way by simply flipping the starting and ending points.

During testing we had a subject, other than the ones who assisted us in collecting the training data performing all four actions several times each, one after the other. This represents a more realistic scenario instead of recording a separate video for each action (i.e., assuming already segmented actions). To make testing more interesting, we added a fourth camera and used different cameras as mentioned in Section 2. The test video sequence contained a total of 633 frames.

Because we did not know in advance where in the sequence a particular action starts and ends, we compared different parts of the sequence against each of the action models. In particular, for each frame *i* in the test sequence, we had to decide how many frames in the most recent history corresponded to the same action. This number could vary in general depending on the speed of the head as well as the frame rate of the camera (i.e., when different cameras are used for training and testing). Here, we considered different lengths of past history, each of them finishing at the current frame. The length values used for each action were computed statistically by analyzing the training data. Then, each subsequence was matched against the four action models. The smallest DTW-distance, for each action, was taken as the distance between the input action and the model for that subsequence.

It should be mentioned that the complexity of this step depends on the number of action models. However, indexing mechanisms could be employed to speed up this step, in a manner analogous to using indexing for object recognition (Bebis et al. 1998). The idea is that instead of having to search the space of all possible matches between the input and the models and explicitly reject invalid ones through verification, indexing could invert this process so that only the most feasible matches are considered. This can be done by arranging information about each model action in an index space offline such that only feasible matches can be found by indexing into this space during recognition.

Figure 9 shows a plot of DTW-distances between input trajectories and each of the action models. When a minimum has been reached for a particular action model, then, if it is below a certain threshold, we assume that this action has been executed. Due to the fact that some actions, if not properly segmented in time, could look quite similar, small differences for some actions might occur. However, if there are two minimums close together (e.g., frame 451), these are further compared and only the one having the smallest distance was considered valid.

Although our testing environment was different from training, all actions in the test sequence were detected and recognized successfully: "sitting-down" was detected at frame

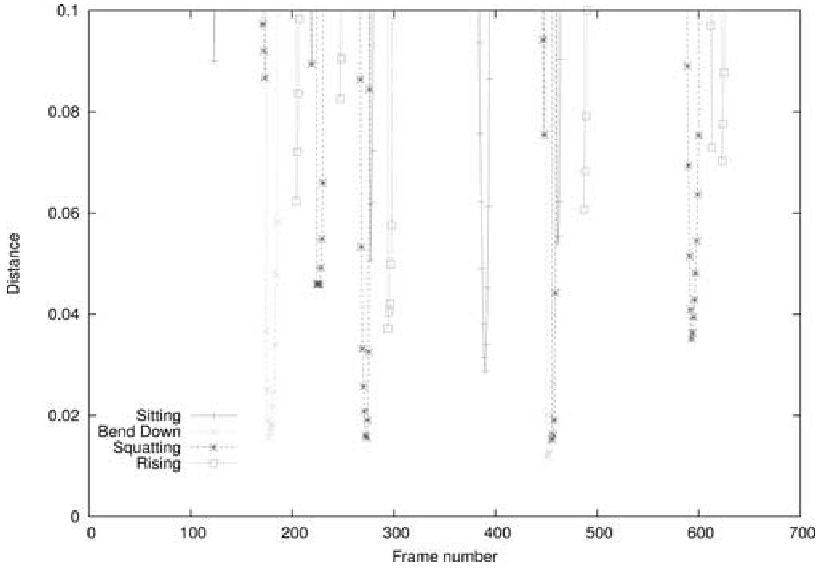


FIGURE 9. Plot of DTW-distances between the test sequence containing 633 frames and each of the four action models. The subject performed all four actions several times, all of which were detected and recognized successfully (see text for details).

123, "bending-down" was detected at frame 177, "squatting" was detected at frame 227, "rising from squatting" was detected at frame 235, "squatting" was detected at frame 273, "rising from squatting" was detected at frame 284; "sitting-down" was detected at frame 390; "bending-down" was detected at frame 451; "squatting" was detected at frame 593; and "rising from squatting" was detected at frame 623. Figure 10 shows several examples from the test sequence where specific actions were detected and recognized correctly.

7. CONCLUSIONS AND FUTURE WORK

We presented a system for recognizing several simple human actions by analyzing the movement of the head in 3D. The proposed system recovers the 3D trajectory of the head by estimating the position of the head in 3D using multiple cameras. Then, it registers the trajectories in a common coordinate system using PCA and aligns them in time using DTW. Finally, it performs normalization prior to recognition. Our experimental results demonstrate the potential of the proposed approach. It should be emphasized again that although the head movement can not be used to discriminate between complex actions, it does provide useful information which could be combined with other cues for building more sophisticated action recognition systems.

For future work, we plan to extend the proposed approach in several ways. First, we plan to estimate the speed of the head to segment different activities in a video sequence more efficiently instead of simply considering different length subsequences. Second, instead of modeling actions by simply averaging the normalized sample trajectories, we plan to modify Itakura's global constraint to take standard deviation into consideration as well. Third, we plan to experiment with more actions as the main purpose of this study was to demonstrate feasibility. Fourth, we plan to perform larger scale experiments using more data both for

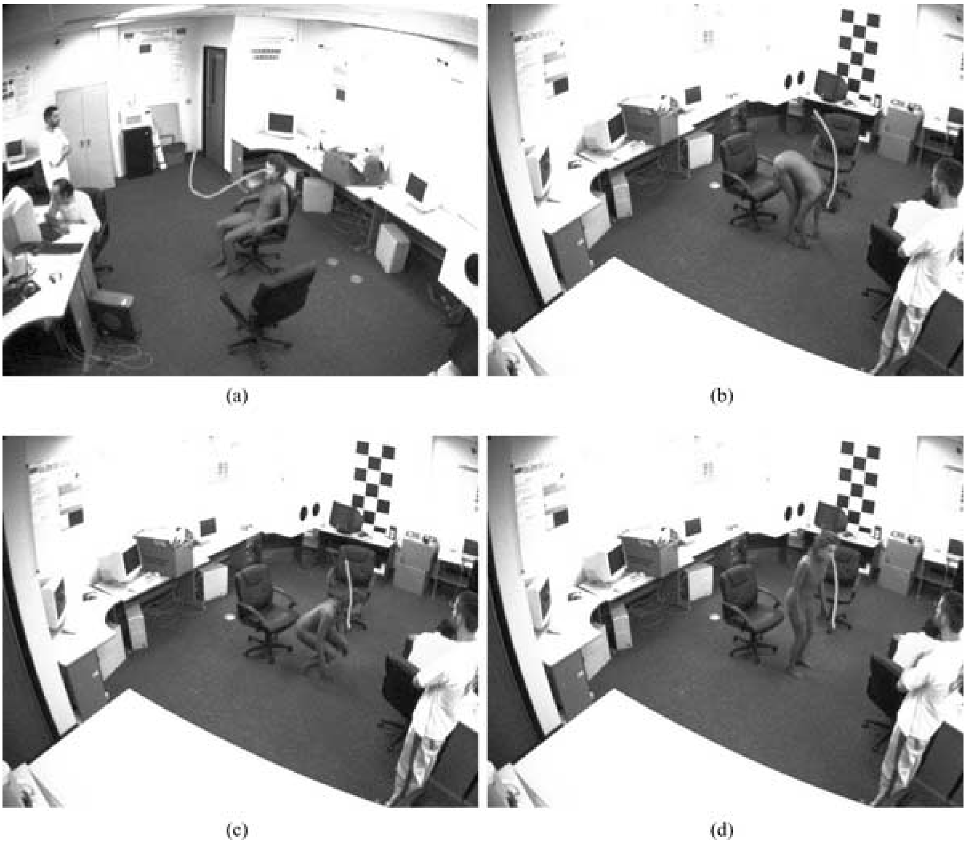


FIGURE 10. (a) *Sitting-down* at frame 123, (b) *Bending-down* at frame 177, (c) *Squatting* at frame 227, (d) *Rising from squatting* at frame 236. Once an action was recognized successfully, the 3D trajectory of that action was projected onto the last frame of the action for illustration purposes.

training and testing. In the same context, we will consider building multiple models per action by splitting the training data for each action into a number of clusters in the spirit of Gavrila and Davis (1995). This would be useful, for example, when the within-variance of an action is very high. Finally, we plan to perform comparisons with competitive approaches such as HMMs.

REFERENCES

BEBIS, G., M. GEORGIOPOULOS, M. SHAH, and N. DA VITORIA LOBO. 1998. Indexing based on algebraic functions of views. Computer Vision and Image Understanding (CVIU), **72**(3):360–378.

BIRCHFIELD, S. 1998. Elliptical head tracking using intensity gradients and color histograms. *In* IEEE Conference on Computer Vision and Pattern Recognition, pp. 232–237.

BOWDEN, R., T. MITCHELL, and M. SARHADI. 2000. Non-linear statistical models for the 3D reconstruction of human pose and motion from monocular image sequences. Image and Vision Computing, **18**(9):729–737.

BROWN, L. 2001. 3D head tracking using motion adaptive texture-mapping. *In* IEEE Computer Vision and Pattern Recognition Conference (CVPR), pp. 998–1005.

CASCIA, M., S. SCLAROFF, and V. ATHITSOS. 2000. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models. IEEE Transactions on Pattern Analysis and Machine Intelligence, **22**(4):322–336.

GAVRILA, D., and L. DAVIS. 1995. Towards 3D model-based tracking of humans in action. *In* IEEE International Workshop on Face and Gesture Recognition, pp. 272–277.

HEIKKILA, J., and O. SILVEN. 1997. A four-step camera calibration procedure with implicit image correction. *In* IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1106–1112.

JAIN, A., R. DUIN, and J. MAO. 2000. Statistical pattern recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence, **22**(1):4–37.

KANADE, T., P. RANDER, and P. NARAYANAN. 1997. Virtualized reality: Constructing virtual worlds from real scenes. IEEE Multimedia, Immersive Telepresence, **4**(1):34–47.

KEOGH, E., B. CELLY, C. RATANAMAHATANA, and V. ZORDAN. 2003. A novel technique for indexing video surveillance data. *In* ACM SIGMM Workshop on Video Surveillance, pp. 98–106.

KWON, O., J. CHUN, and P. PARK. 2006. Cylindrical model-based head tracking and 3D pose recovery from sequential face images. *In* International Conference on Hybrid Information Technology, pp. 135–139.

MADABHUSHI, A., and J. AGGARWAL. 1999. A bayesian approach to human activity recognition. *In* 2nd IEEE International Workshop on Visual Surveillance, pp. 25–30.

NAIT-CHERIF, H., and S. MCKENNA. 2003. Head tracking and action recognition in a smart meeting room. *In* IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, p ?

OATES, T., L. FIROIU, and P. COHEN. 1999. Clustering time series with hidden Markov models and dynamic time warping. *In* IJCAI-99 Workshop on Neural, Symbolic and Reinforcement Learning Methods for Sequence Learning, pp. 17–21.

OHAYON, S., and E. RIVLIN. 2006. Robust 3D head tracking using camera pose estimation. *In* 18th International Conference on Pattern Recognition (ICPR), pp. 1063–1066.

OLIVER, N., B. ROSARIO, and A. PENTLAND. 2000. A Bayesian computer vision system for modeling human interactions. IEEE Transactions on Pattern Analysis and Machine Intelligence, **22**(8):831–843.

RAO, C., A. GRITAI, M. SHAH, and T. SYEDA-MAHMOOD. 2003. View-invariant alignment and matching of video sequences. *In* IEEE International Conference of Computer Vision (ICCV), pp. 939–945.

RAO, C., A. YILMAZ, and M. SHAH. 2002. View-invariant representation and recognition of actions. International Journal of Computer Vision, **50**(2):203–226.

ROBERTSON, N., and I. REID. 2006. A general method for human activity recognition in video. Computer Vision and Image Understanding (CVIU), **104**(2):232–248.

SYEDA-MAHMOOD, T., M. VASILESCU, and S. SETHI. 2001. Recognizing action events from multiple viewpoints. *In* IEEE Workshop on Detection and Recognition of Events in Video, pp. 64–72.

TERADA, K., A. OBA, and A. ITO. 2005. 3D human head tracking using hypothesized polygon model. *In* IEEE International Conference on Systems, Man and Cybernetics, pp. 1036–1401.

TOMASI, C., and T. KANADE. 1993. Shape and motion from image streams: A factorization method. *In* Proceedings of the National Academy of Sciences, **90**:9795–9802.

TRUCCO, E., and A. VERRI. 1998. Introductory Techniques for 3-D Computer Vision. Prentice-Hall, Upper Saddle River, NJ, pp. 143–145.

WELCH, G., and G. BISHOP. 1995. An Introduction to the Kalman Filter. University of North Carolina at Chapel Hill, Tech. Rep. TR 95-041.

ZHANG, Z. 2000. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, **22**(11):1330–1334.

ZHANG, Y., and C. KAMBHAMETTU. 2002. Robust 3D head tracking under partial occlusion. Pattern Recognition, **35**:1545–1557.