MVA2015 IAPR International Conference on Machine Vision Applications, May 18-22, 2015, Tokyo, JAPAN

Egocentric articulated pose tracking for action recognition

HarukaYONEMOTO, Kazuhiko MURASAKI, Tatsuya OSAWA,
Kyoko SUDO, Jun SHIMAMURA, and Yukinobu TANIGUCHI
NTT Media Intelligence Laboratories, NTT Corporation
1-1 Hikari-no-oka, Yokosuka, Kanagawa, 239-0847 Japan
{yonemoto.haruka, murasaki.kazuhiko, osawa.tatsuya,
sudo.kyoko, shimamura.jun, taniguchi.yukinobu}@lab.ntt.co.jp

Abstract

Many studies on action recognition from the third-person viewpoint have shown that articulated human pose can directly describe human motion and is invariant to view change. However, conventional algorithms that estimate articulated human pose cannot handle ego-centric images because they assume the whole figure appears in the image; only a few parts of the body appear in ego-centric images. In this paper, we propose a novel method to estimate human pose for action recognition from ego-centric RGB-D images. Our method can extract the pose by integrating hand detection, camera pose estimation, and time-series filtering with the constraint of body shape. Experiments show that joint positions are well estimated when the detection error of hands and arms decreases. We demonstrate that the accuracy of action recognition is improved by the feature of skeleton when the action contains unintended view changes.

1 Introduction

In recent years, Head Mount Display (HMD) navigation systems have become popular for supporting daily life and the work of technicians. Current support systems display information corresponding to the user's operations. If the systems can recognize where the user is working and what he or she wants to do, the user can be notified in advance as what should or should not do or what they forgot to do. We develop a method that recognizes the user's actions from the video taken from the user's view (ego-centric video) to realize a better user interface. Common approaches for action recognition from egocentric video are based on object recognition because objects are very important cues in identifying what the user is doing and where he or she is [1, 2, 3, 4]. However, different actions can be performed on the same object and actions that do not involve an object might not be recognized. Our approach to describing human motion is to estimate the user's articulated pose. We assume that the camera is fixed to the user's head. Our proposal can extract the pose by integrating hand detection, visual self localization, and time-series filtering with the constraint of body shape.

2 Related work

Motion-based action recognition. Some methods use optical flow to obtain the movement features of hands [5] or track hands [6]. Sudeep et al. [6] distinguishes different actions on the same object, for example using a spoon to scoop or to stir. However, these methods are weak against view direction changes since these features are not view invariant; the same motion captured from different view directions yields different features.

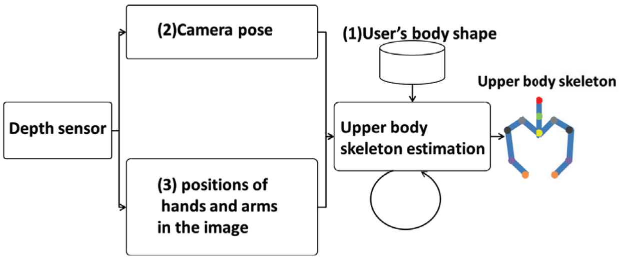
(2)Camera pose
(1)User's body shape
Upper body skeleton
Depth sensor
Upper body skeleton estimation
(3) positions of hands and arms in the image

Figure 1. Our framework of pose tracking

Action recognition using articulated human pose. Many studies have shown that accurate action recognition is made possible by the features of articulated human pose [7, 8, 9]. Xia et al. [10] proposed a view invariant feature, HOJ3D, and demonstrated that it offers significant view invariance and high accuracy. The articulated human pose obtained from ego-centric video is as useful as that from third person video. However, conventional approaches do not support ego-centric video [11, 9], since they assumes that the whole body is observed in the image.

In this paper, we propose a method that can estimate human articulated pose from ego-centric video, in which only a few parts of the body appear. Our proposal can extract the pose by integrating hand detection, visual self localization, and time-series filtering with the constraint of body shape.

3 Proposed method

Our method has two main processes; extract cues to estimate articulated pose, then extract pose by using the cues. Cues are (1) user's body shape; Figure 1-(1), (2) camera pose; Figure 1-(2), and (3) hand and arm position in the image; Figure 1-(3). (3) hand and arm position in the image is shown in Figure 2. (2) camera pose and (3) hand and arm position in the image is usually obtained by a visual self-localization technique [12] and a hand and arm detector [13], respectively. In this paper, to evaluate the accuracy of pose estimation in an ideal environment, we get these cues and the input image data from a modeling program of human pose. The input motion data is obtained by a motion

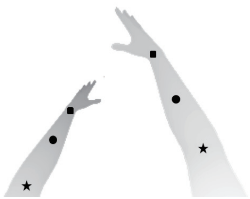

Figure 2. The cues of hand and arm positions in the synthesized depth image. square: the position of hand joint, sphere: the interior-divided position of hand and arm joint, star: the position of elbow.

capture system and input to the modeling software. The ego-centric image is synthesized by rendering the CG data of human pose that the modeling software outputs. We virtually obtain (2) camera pose and (3) hand and arm position by using a modeling program of human pose.

3.1 Human skeleton model

Articulated pose is generally described using a tree structure. We define a tree structure whose root node is the head. The root node is assumed to be the camera mounting point (Figure 3); the other nodes represent joints. This model consists of 11 joints; head, neck, chest, L/R collar, L/R shoulder, L/R forearm, L/R hand (L: Left, R: Right). The head has 6 degrees of freedom (rotation and translation), while the other joints have just three to cover rotation around the x, y, z axes. The parameters to be estimated are all rotation parameters of joints except for head because the parameter of the head is computed by camera pose. We write the set of these parameters as the pose parameter vector. The pose parameter vector at time t is described as

$$\boldsymbol{\Theta}_t = (\boldsymbol{\theta}_{t,1}, \boldsymbol{\theta}_{t,2}, ..., \boldsymbol{\theta}_{t,j}, ..., \boldsymbol{\theta}_{t,10}), \quad (1)$$

where j is joint index, and t is frame number. The rotation parameters have restrictions related to the range of joint movements. The restriction is represented by using α and β which means upper and lower limit, respectively.

$$\alpha_j \leq \theta_{t,j} \leq \beta_j. \quad (2)$$

The lengths between neighbor joints, i.e. user's body shape, are represented as the offset from the parent's joint when all rotation parameters are 0.

$$\boldsymbol{o}_j = (X_j, Y_j, Z_j, 1), \quad (3)$$

where $\boldsymbol{o}_j$ is represented using a homogeneous coordinate system. Impossible poses are omitted by considering the restriction and predefining $\boldsymbol{o}_j$. In HMD scenarios, the device is intended for personal use so it is natural to assume that the device knows the user's shape parameters, $\boldsymbol{o}_j$. Therefore, we need to estimate only pose parameters $\boldsymbol{\theta}_t$.

3.2 Articulated pose tracking

Model-based tracking is used to combine the cues and a skeleton model. By using a skeleton model, we can utilize the known dynamics of human motion and omit the impossible joint positions as candidates. Articulated pose is estimated by using a particle filter [14]. We denote the state vector at time t by x_t and the sequence of observation vectors up to time t by $Y_t = \{y_1, y_2, ..., y_t\}$. In the particle filter framework, the posterior probability $p(x_t|Y_t)$ is approximated by a finite set of samples $\{x_t^{(i)}\}_{i=1}^n$ (called particle) with importance weights $\pi_t^{(i)}$, where n denotes the number of particles and i is particle index. At each time t, particles are updated by two steps; the prediction step and the update step. In the prediction step, N particles are generated by replacing of each particle at time $t-1$ with its equivalent computed from the motion model. In the update step, the weight of each particle is computed and particles at time t are resampled from these predicted particles according to these weights. The new set of particles describes the approximation of posterior distribution. In our framework, the state vector is pose vector $\boldsymbol{\Theta}_t$ and an observation sequence consists of depth images and the 2D positions of hand and arm in the image.

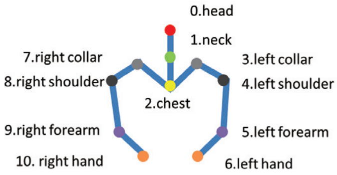

Figure 3. The tree structure for the articulated pose of the upper body.

3.2.1 Prediction step

In the prediction step, N particles are predicted from the previous set of particles using the motion model. We employ a static model as the motion model since joint angles move only slightly given the frame rate. Particles are predicted by adding Gaussian noise to each rotation parameter of $\theta_{t,j}$. This Gaussian distribution has zero-mean and variance $\frac{r}{C}$, where r means the range of motion of joint angle and C is a constant. These predicted samples satisfy the joint angle restriction (2).

3.2.2 Update step

The likelihoods of particles are evaluated by the results of hand detection and depth images. The joint positions of hands and arms are projected onto the image from each predicted sample. We denote this position by c and the 2D position of hands and arms in the image by c'. c and c' is a vector describing the x, y coordinates of the position in the image. We denote $a^{(i)}$ as the distance in pixels between c and c'. We denote the depth value at c' by $d_o^{(i)}$ and the distance between camera and joint (hand and arm) positions in camera coordinate as $d_\theta^{(i)}$. The likelihood of each predicted particle is described as

$$q(\Theta_t^{(i)}|y_t) = \prod_{k \in K} f(a_k^{(i)}) \times \chi(d_{\theta,k}^{(i)}, d_{o,k}^{(i)}), \quad (4)$$

where $f(\cdot)$ is a Gaussian distribution with zero-mean and variance σ, and k is joint index corresponding to

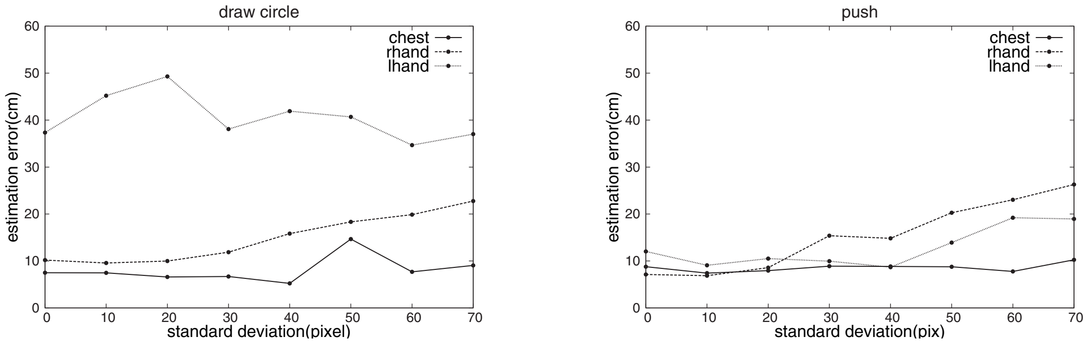

Figure 4. Result of articulated pose estimation from ego-centric video. x-axis plots the standard deviation, in pixels, of detection error for hands and arms. y-axis plots the error, in cm, of estimated 3D joint positions.

Figure 2. $\chi(d_{\theta,k}^{(i)}, d_{o,k}^{(i)})$ is the function given as

$$\chi(d_{\theta,k}^{(i)}, d_{o,k}^{(i)}) = \begin{cases} f'(|d_{o,k}^{(i)} - d_{\theta,k}^{(i)}|) & \text{if } d_{\theta,k}^{(i)} \geq d_{o,k}^{(i)}, \\ 0 & \text{if } d_{\theta,k}^{(i)} < d_{o,k}^{(i)}, \end{cases} \quad (5)$$

where $f'(\cdot)$ is a Gaussian distribution with zero mean and variance σ'. The likelihood function returns larger values as distance $a^{(i)}$ shrinks and the depth value in the image approaches the distance between camera and joint (hand and arm) positions. The poses are omitted if the distance between the joint position and camera is smaller than the depth value from the depth sensor. The weights of all particles are normalized as

$$\pi^{(i)} = q(\hat{\Theta}_t^{(i)}|y_t) / \sum_i q(\hat{\Theta}_t^{(i)}|y_t), \quad (6)$$

where y_t describes the input depth image and the cue of hand and arm positions. N particles are generated at time t by choosing a particular sample from predicted samples according to the weights. This new set of samples describes the posterior probability.

4 Action recognition

We employ Xia's framework [10] for action recognition to get its view invariant features. They proposed HOJ3D, which is a pose vector representing the occupancy of joints relative to the root joint, i.e. hip center in their method. We compute HOJ3D in each frame and quantize the results. We then train a Hidden Markov Model (HMM) using the quantized HOJ3D sequence as the observed data. When a test sequence is given, an action label is returned for it by classifying it as the action that has the largest posterior probability.

$$decision \;\; = \;\; \underset{l=1,2,\ldots,M}{\arg\max}\, P(V|\lambda_l). \quad (7)$$

V is a test sequence and λ_l are the trained parameters of each HMM; M is the total number of actions.

5 Experimental evaluation

5.1 Experimental settings

We used Poser Pro 2014[1] as the modeling program to build the dataset. A motion capture system (OptiTrack[2]) was used to build the dataset. All synthesized data was generated for one human model and $\boldsymbol{o}_j$ matched that of the human model.

Articulated pose tracking. We captured 4 actions (draw circle, horizontal arm wave, push, two hand wave) including dynamic motion of hands, arms and upper body. They were collected from 1 person. We took the variance in prediction of pose parameter to be $C = 30$. The restriction of rotation parameters, α and β in (2), was set to be the same as default values in Poser's model. The variances of Gaussian distribution in computing likelihood (4, 5) were arbitrary constants. We synthesized the detection error of hand and arms (corresponding to Figure 2) by adding Gaussian noise to the correct positions of hand and arm in the image. These 2D positions are used as the hand and arm position cues in the image.

Action recognition. We captured the motion of several assembly operations including 5 actions (open cover, attach HDD, screw, attach memory, close cover). These actions were performed twice by 2 different people and this data was used for HMM training. Test data are collected by taking the actions performed for one time by the two people the same as in training data. To evaluate the proposal's robustness against view change, the test data included view changes. We compared two HOJ3D features; one sets the chest joint position as the spherical coordinate origin (Chest-HOJ3D) while the other sets the camera position as the origin (Camera-HOJ3D). We consider the former as the skeleton feature and the latter as the non-skeleton feature.

5.2 Results

Articulated pose tracking. Figure 4 shows the estimation error of the joints when the standard deviation of the detection error of hands and arms increases. The hand joints may lie in the image but not the chest joint. The results of draw circle and push are shown as representative of the other motions. In draw circle, the error in left hand position is large because the left hand does not appear in the image, but this does not affect the estimation of other joints. In both actions, the poses are stably estimated when the standard deviation of detection error is under 20. We conjecture

[1] `http://my.smithmicro.com/poser-pro-2014.html`

[2] `https://www.naturalpoint.com/optitrack/`

Table 1. Accuracy of action recognition (average F-score). 5 actions associated with device assembly are evaluated using motion capture dataset. 'no view change': the test data does not contain view changes, 'view change': the test data contains view changes.

	no view change	view change
Chest-HOJ3D	0.75	0.67
Camera-HOJ3D	0.91	0.21

Table 2. Accuracy of action recognition (F-score). 5 actions associated with device assembly are evaluated using estimated skeleton data. The test data contains view changes.

	open cover	attach HDD	screw	attach memory	close cover	average
Chest-HOJ3D	0.91	0.69	0.25	0.37	0.15	0.48
Camera-HOJ3D	0.86	0.00	0.57	0.06	0.21	0.34

that this is because the proposed method (1) considers the smooth movement of joints through its use of time-series filtering, (2) omits impossible poses by the skeleton model, the restriction of joint movement and predefined user's shape parameter, and (3) computes the likelihood of each particle by using the results of the multi joint detector and the depth value.

Action recognition. The results of action recognition are shown in Table 1. In a preliminary experiment, we used the motion capture data as the joint's 3D position. The results show that data containing view changes decreases the accuracy of both chest-HOJ3D and the camera-HOJ3D. The former is more accurate than the latter. This is because hand motion relative to the camera position can not be distinguished from camera motion relative to hand position; the motion of the hand against the chest position is camera motion invariant. Table 2 shows the result of action recognition using skeleton estimated by the proposed method (noise added to hand and arm position cues is 0). If the data contains view changes, Chest-HOJ3D offers higher overall accuracy as the preliminary experiment shows.

6 Conclusion

We proposed an articulated pose tracking framework that offers novel motion features for action recognition to be extracted from ego-centric videos. Even though some joints of the upper body cannot be seen in first person views, our solution is effective as it integrates hand detection, camera localization and a predefined human body model and employs particle filtering to track the user's full upper body. Experiments on synthesized depth videos showed that pose can be stably estimated even in the presence of hand position errors due to our use of model-based tracking and time-series filtering. The results gained from action recognition trials showed that describing motion relative to the chest is useful in capturing actions in the face of unintended view changes. In the future, we aim to improve overall joint estimation accuracy and apply our method to real data.

References

[1] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *CVPR*, pp. 2847–2854, 2012.

[2] A. Fathi and M. J. Rehg. Modeling actions through state changes. In *CVPR*, pp. 2579–2586, 2013.

[3] K. Matsuo, K. Yamada, S. Ueno, and S. Naito. An attention-based activity recognition for egocentric video. In *CVPR*, pp. 565–570, 2014.

[4] A. Fathi, Y. Li, and J. M. Rehg. Learning to recognize daily actions using gaze. In *ECCV*, pp. 314–327, 2012.

[5] S. Narayan, M. S. Kankanhalli, and K. R. Ramakrishnan. Action and interaction recognition in first-person videos. In *CVPRW*, pp. 526–532, 2014.

[6] S. Sundaram and W.M Cuevas. High level activity recognition using low resolution wearable vision. In *CVPRW*, pp. 25–32, 2009.

[7] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Learning actionlet ensemble for 3d human action recognition. *IEEE Trans. PAMI*, Vol. 36, No. 5, pp. 914–927, 2014.

[8] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR*, pp. 588–595, 2014.

[9] A. Yao, J. Gall, and L. V. Gool. Coupled action recognition and pose estimation from multiple views. *Int. J. Comput. Vision*, Vol. 100, No. 1, pp. 16–37, 2012.

[10] L. Xia, C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *CVPRW*, pp. 20–27, 2012.

[11] J. Shotton, T.Sharp, A.Kipman, A.Fitzgibbon, M.Finocchio, A.Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, Vol. 56, No. 1, pp. 116–124, 2013.

[12] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard. Real-time 3d visual slam with a hand-held rgb-d camera. In *the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*, Vol. 180, 2011.

[13] G.Rogez, J.S.Supancic III, M.Khademi, and D.Ramanan J.M.M.Montiel. 3d hand pose detection in egocentric rgb-d images. In *ECCVW*, 2014.

[14] A.Doucet, N.De Freitas, and N.Gordon. *An introduction to sequential Monte Carlo methods*. Springer, 2001.