Adaptive Human Shape Reconstruction via 3D Head Tracking for Motion Capture in Changing Environment

Kazuhiko Murasaki, Masamichi Shimosaka, Taketoshi Mori and Tomomasa Sato

Abstract— **This paper describes a human shape reconstruction method from multiple cameras in daily living environment, which leads to robust markerless motion capture. Due to continual illumination changes in daily space, it had been difficult to get human shape by background subtraction methods. Recent statistical foreground segmentation techniques based on graph-cuts, which combine background subtraction information and image contrast, provide successful results; however, they fail to extract human shape when furniture such as tables and chairs are moved. In this paper, we focus on the results of face detectors that would be independent of such background changes and help to improve the robustness under movement of background objects. We propose a robust human shape reconstruction method with the following two characteristics. One is iterative image segmentation based on graph-cuts to integrate head position information into shape reconstruction. The other is high-precision head tracker to keep multi-view consistency. Experimental results show that proposed method has enhanced human pose estimation based on reconstructed human shape, and enables the system to deal with dynamic environment.**

I. Introduction

Vision-based human pose estimation is expected to realize markerless motion capture, or the motion capture system without any device attached. There are many approaches for pose estimation [1], [2], such as silhouette based methods and volume based methods. Although these approaches succeed to estimate pose well enough, most approaches assume that human silhouette is captured by simple background subtraction, and cannot deal with difficult scenes, such as cluttered background, illumination changes and dynamic background. In fact, background images frequently changes in daily living space. Recent statistical foreground segmentation techniques based on graph-cuts, which combine background subtraction information and image contrast, provide successful results [3]; however, they fail to extract human shape when furniture in background such as tables and chairs are moved. To extract human silhouette robustly, we introduce additional information independent of background. Human head position could be significant to robust silhouette extraction because human head has common image features and is easy to be detected by cameras. We propose a robust human shape reconstruction method with the following two characteristics. One is iterative image segmentation based on graph-cuts to integrate head position information into human silhouette extraction. The other is high-precision head tracker to keep multi-view consistency of head poses.

Kazuhiko Murasaki, Masamichi Shimosaka, Taketoshi Mori and Tomomasa Sato are with Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan. {murasaki, simosaka, tmori, tsato}@ics.t.u-tokyo.ac.jp

There are many approaches to reconstruct human shape in spite of dynamic background. For example, voxel coloring which uses multi-view color consistency [4] and the fusion of multiple depth map acquired by stereo cameras [5] are able to reconstruct a target shape without silhouette extraction. Indeed, these reconstruction methods are not affected by background changes, but they take much more computational cost and need much more cameras. Moreover, it is necessary to extract human shape from whole shape of the target space through some additional clues. To achieve high-speed reconstruction, we employ silhouette-based approach, that is volume intersection. We tackle to extract human silhouette robustly with background and additional information. Recently, silhouette-based approaches with not only background information but also feedback from reconstruction results are proposed. Although feedback from reconstructed volumes [6] or feedback from estimated human pose [7] help us to extract human silhouette when background changes, these feedback are effective only when background changes are apart from human; furthermore, it is difficult to recover silhouette extraction if the feedback loop collapses. Our reconstruction method leverages head position estimated by textual features, which is independent of background information, to improve the robustness under movement of background objects.

Remainder of this paper is as follows. In section II, the overview of our motion capture system is introduced. In section III, we describe human silhouette extraction method via graph-cuts. Then, 3D head tracking method is described in section IV. In section V, experimental results in difficult situations are shown. Finally, our conclusions are discussed in section VI.

II. Human Pose Estimation via Multi-Cameras

Fig. 1 shows the flow of proposed motion capture system as the baseline of our research. First of all, multi-view cameras surrounding the target human fetch image sequences. Multi-camera setup is showed in Fig. 1. Then, 3D head position is estimated through multi-view head detection and classification. Next, 3D head position is used to extract human silhouette from camera images. Silhouette extraction is based on integration of multiple information such as color distribution, color contrast and positional constraint. Then, human shape is reconstructed by volume intersection through multi-view silhouette images. Human shape is expressed by a set of voxels. Finally human pose is estimated to fit human-shaped voxels. We employ Shimosaka's method [2] on pose

estimation. Our research focuses on shape reconstruction method in dynamic scenes. Our approach is based on human silhouette extraction based on head position and 3D head position estimation.

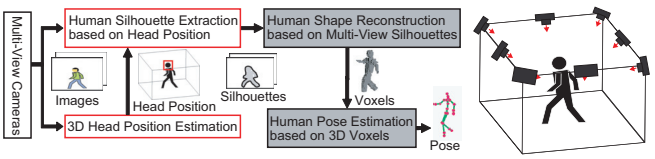

Fig. 1. Markerless Mocap and Multiple Camera Setup

III. Human Silhouette Extraction via Head Position

In order to extract human silhouette from a camera image, we use clue of head position in addition to background information. Our extraction method is based on graph-cuts to integrate multiple cues. Integration of pixel-level color likelihoods, image contrasts and positional constraint of head enables robust extraction in spite of background changes. Furthermore robust extraction makes it possible to update background images online.

A. Background Subtraction via Graph-Cuts

We introduce a method of background subtraction based on graph-cuts, that is the basis of our method. Image segmentation based on graph-cuts is performed by minimizing following energy function,

$$E(X) = \sum_{r \in I} D_r(x_r) + \lambda \sum_{(r,s) \in \varepsilon} S_{rs}(x_r, x_s) \qquad (1)$$

where I is a set of pixels in a image and ε is a set of combinations of neighboring pixels. X means a set of foreground / background labels x_r at pixel r. Graph-cuts calculate a label set X minimizing $E(X)$. $D_r(x_r)$ is a data term, which denotes a cost to label pixel r as x_r, and $S_{rs}(x_r, x_s)$ is a smoothing term, which denotes a cost of label changes between neighboring pixels. λ is a parameter which balances two terms. The data term and the smoothing term are defined as follows.

$$D_r(x_r) = \begin{cases} -\ln p_B(\boldsymbol{i}_r) & x_r = 0 \\ -\ln p_H(\boldsymbol{i}_r) & x_r = 1 \end{cases} \qquad (2)$$

$$S_{rs}(x_r, x_s) = |x_r - x_s| \cdot \exp(-\gamma d_{rs}), \qquad (3)$$

where p_B denotes background likelihood and p_H denotes human-area likelihood, d_{rs} is color difference between pixels. Minimizing S_{rs} causes label changes along strong contrasts. γ is a parameter for normalization.

Background likelihood p_B is expressed in YCrCb color space, and modeled by Gaussian mixture at each pixel. Each distribution is trained by iterative updating for each frame [8]. Moreover, introducing likelihood evaluation via CrCb, which is chromatic information, can deal with shadow effect [6].

Human-area likelihood p_H is also expressed in YCrCb color space, and modeled by Gaussian mixture. Because human color has no relationship with location, distribution is trained from whole pixels labeled as human area. Besides likelihood of whole human color, likelihood of skin color is employed. Skin color is learned from a head position.

Image contrast d_{rs} means color differences between pixel r and pixel s. To emphasize foreground contrast only, d_{rs} is designed to weaken contrast derived from background image [3].

Fig. 2 shows the result of background subtraction via graph-cuts. Comparative method using only background likelihood is affected by background flicker and shadow. On the other hand, the method via graph-cuts extracts noiseless and accurate silhouette. However, this approach cannot divide human region and object region when some objects except human appears like Fig. 2. To handle such background changing situation, our approach employs positional constraint of human head.

Fig. 2. Background Subtraction via Graph-Cuts

B. Constraint of Head Position

1) Estimation of Central Axis of Human Body: It is difficult to extract human body silhouettes directly from positional constraint of human head. Then, first of all, we estimate a central axis of human body by 3D head position. Assuming the human stands on the floor, 3 points in an image, which are a head P_H, a center of gravity P_G and a foot P_F , are estimated. The head position P_H is the projected point of estimated 3D head position in the camera image. The center of gravity P_G is calculated by background and human-area likelihoods. Probability of being human area, $p_H/(p_H + p_B)$ is calculated for each pixel, and center of the probabilities becomes P_G. The foot position P_F is calculated by P_H and P_G. The position where 3D head position is projected onto the floor is projected to P'_F in camera image, that is $\overrightarrow{P_H P'_F}$ is the projected vector of the vertical vector from the head to the floor. Using point P'_F, the foot position is defined as $P_F = P_G + \frac{1}{2}\overrightarrow{P_H P'_F}$. Fig. 3 shows estimated central axis based on the 3 points. Our approach is effective in various scenes such as standing, sitting and bending down.

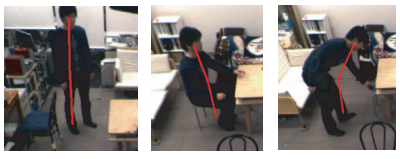

Fig. 3. Rough Human Area Based on Head Position

2) Graph-Cuts Modeling via Head Position: The central axis of the body can be used to evaluate human-area likeli-

hood of each pixels, then data term, which was defined as (2), is redefined as follows,

$$D_r(x_r) = \begin{cases} -\ln p_B(\boldsymbol{i}_r) + \eta(1 - q_H(r)) & x_r = 0 \\ -\ln p_H(\boldsymbol{i}_r) + \eta q_H(r) & x_r = 1 \end{cases} \quad (4)$$

where $q_H(r)$ denotes likelihood based on distance from the central axis, and η is a parameter adjusting the effect. The distance from the axis is described as $d_H(r)$, and the human-area likelihood $q_H(r)$ is defined as follows

$$q_H(r) = \exp(-\frac{d_H(r)^2}{2\sigma}) \quad (5)$$

where σ means thickness of the human body in camera image.

3) Iterative Reshaping the Human Silhouette: Like Fig. 2, when energy minimization is applied to whole image, background changes are extracted as human area wrongly. It is necessary to integrate the positional constraint into silhouette extraction in order to deal with such scenes with background changes. Based on active contour approach [9], we propose the approach to revise the extraction result iteratively by graph-cuts segmentation.

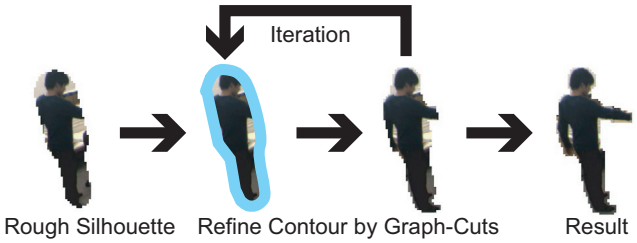

Fig. 4. Iterative Segmentation Approach

Fig. 4 shows the processing flow.

1) Based on the central axis of the body and its thickness, rough human silhouette is defined.
2) p_B, p_H, d_{rs} of pixels in the boundary region (blue area in Fig. 4) of the silhouette is calculated.
3) The boundary region is segmented by graph-cuts.
4) Step 2 and 3 are repeated until human silhouette converge.

At step 3, inner region is set to human area, and outer region is set to background to apply graph-cuts to boundary region. That is, if pixel r is in inner side, $p_B(\boldsymbol{i}_r) = 0, p_H(\boldsymbol{i}_r) = 1$ or if pixel r is outer side, $p_B(\boldsymbol{i}_r) = 1, p_H(\boldsymbol{i}_r) = 0$.

Fig. 5 shows the differences between global graph-cuts segmentation and iterative graph-cuts based on head position. Positional constraint achieves to extract only human silhouette in spite of background changes.

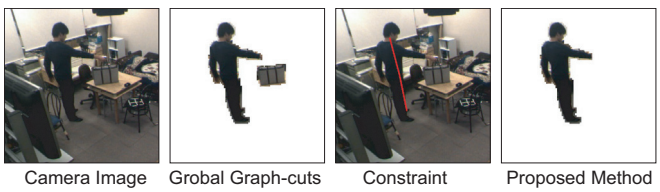

Fig. 5. Iterative Graph-Cuts vs. Global Graph-Cuts

C. Online Updating of Background Color Model

Our extraction method can divide human silhouette from changing background via head position. Taking advantage of robust human silhouette extraction, background and human color information is able to learned from the silhouette. In background region, background GMMs are updated gradually per pixel [8], and in human region, human color GMM is retrained by EM algorithm from the set of pixels labeled as human. Because there is a problem of adaptive background subtraction that static foreground is learned as background with time, it is difficult to learn background changes fast. However, our approach achieves fast update without making human area background by using silhouette extraction result. Moreover, because head position estimation is independent of background information, the loop of information update can be restored easily even if it collapses. Fig. 6 shows update of background images. Upper row shows camera images, and lower row shows background images. In this scene, a man brings a bag on a table and goes by. The bag on the table is learned as background soon after putting it on the table. However the static man sitting on a chair does not become background.

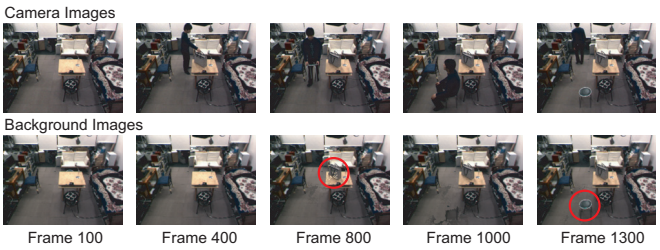

Fig. 6. Online Updating of Background Image

Background update affects not only learning after background changes but also accuracy of silhouette extraction while background is changing. Fig. 7 shows the difference of silhouette extraction while background is changing between without update and with update. In this scene, a man moves a chair and sit down on it, therefore background changes in the region of the chair and the shadow under it. Though positional constraint by head position (left of Fig. 7) cannot handle this effect, extraction with update of background produces good result (right of Fig. 7). This is because gradual change of background like shadows is learned quickly.

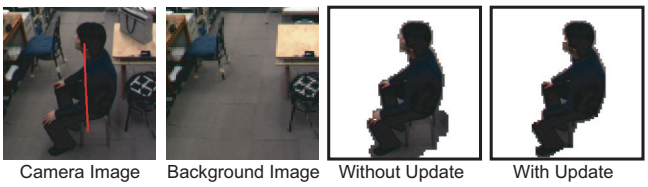

Fig. 7. Improvement of silhouette extraction by background update

IV. Head Position Estimation

As already discussed, our silhouette extraction method leverages head position as a clue of human position. That is because head (face) is the most distinctive body parts to

be detected in image, and it has common features among various people. Moreover, because it is at higher position in general, it is hardly occluded by some objects like furnitures. Therefore we tackle to estimate 3D head position through head detection in each camera image.

We propose high-precision multi-pose head classifier and integration method of multi-camera head detections to achieve robust estimation of 3D head position. Proposed method has following two features. First one is dataset clustering on training phase to minimize false classification rate, and the other is multi-view integration based on consistency of head pose estimated on each view.

A. Training of Multi-Pose Head Classifier

There are some problems to detect heads by cameras equipped on the ceiling. They are low-resolution image, variety of head pose and computational cost. Viola and Jones's face detection method [10] is famous for high-speed detection of low-resolution facial images. Their image classifier uses well-known rectangle features and is trained by Adaboost algorithm to achieve high-precision and fast detection. Various methods based on the same framework are proposed, then we also construct a head detector based on Viola and Jones's approach. As a solution to the problem of variety of head pose, it is general approach to train independent classifiers for all pose classes, however not only it takes time depending on a number of classes but also definition of pose classes affects classification accuracy. In our approach, tree-structured classifier is trained to achieve multi-pose multi-class fast detection, and a pose class is automatically divided appropriately in training phase. To divide a pose class, we have built a head image dataset with pose information, or Yaw and Pitch angles.

Practically, head detection from images is performed through sliding window search. This approach scans the image with a fixed-size window and applies the classifier to the subimage defined by the window. Head image is detected when the classification result is positive.

1) Feature Selection via RealAdaboost: Adaboost algorithm is a method to create a binary classifier by combination of many weak classifiers. From many candidates of rectangle features, the most effective feature is picked up and trained as a weak classifier, then connected to the main classifier. Several variants of Adaboost algorithm is proposed to improve its performance. FloatBoost [11] employs the feature elimination step to reduce number of weak classifiers. RealAdaboost [12] expands the output of weak classifiers from binary decision to probabilistic distribution. Our method employs RealAdaboost to select features and train weak classifiers. Each weak classifier outputs probability value as *confidence* based on the feature value histogram [13]. The flow of feature selection and training of weak classifiers is described below.

A training dataset is expressed as $S = (\boldsymbol{y}_i, z_i), i = 1, \ldots, N$, where $\boldsymbol{y}_i$ is an image input, and z_i denotes a label (head images are labeled as $z_i = 1$, others are $z_i = -1$). A main classifier composed of T weak classifiers $f(\boldsymbol{y})$ is defined as $F^T(\boldsymbol{y}) = \sum_{t=1}^{T} f_t(\boldsymbol{y})$, and its classification result is $z = \mathrm{sign}(F^T(\boldsymbol{y}))$. In RealAdaboost algorithm, we boost up the classification performance by adding weak classifiers one by one. First, we define weak classifiers $f(\boldsymbol{y})$. Weak classifiers output a value between -1 and 1 according to the feature value of rectangle feature. Its feature value is divided with equal width to n_c regions, and each region has *confidence* about output label. Weak classifiers output its *confidence* $c_j, j = 1, \ldots, n_c$ corresponding to the region of feature value $u_j, j = 1, \ldots, n_c$ as follows.

$$f(\boldsymbol{y}) = c_j, \text{ if } h(\boldsymbol{y}) \in u_j, j = 1, \ldots, n_c \tag{6}$$

where $h(\boldsymbol{y})$ denotes the feature value of $\boldsymbol{y}$. Selection of new feature $h_{t+1}(\boldsymbol{y})$ and training of new weak classifier $f_{t+1}(\boldsymbol{y})$ is performed by minimization of the loss function as follows.

$$L^{t+1} = \sum_{i=1}^{N} \exp(-z_i F^{t+1}(\boldsymbol{y}_i)) \tag{7}$$

Following value is defined by classification result on each label and each region.

$$W_{lj} = \sum_{i: z_i = l, h(\boldsymbol{y}_i) \in u_j} \exp(-l F^t(\boldsymbol{y}_i)) \tag{8}$$

Then, (7) is deformed as follows.

$$L^{t+1} = \sum_{j=1}^{n_c} (W_{+1j} e^{-c_j} + W_{-1j} e^{c_j}) \tag{9}$$

This loss function is minimized when

$$c_j = \frac{1}{2} \ln(\frac{W_{+1j}}{W_{-1j}}). \tag{10}$$

Plugging into (9), L^{t+1} becomes

$$L^{t+1} = 2 \sum_{j=1}^{n_c} \sqrt{W_{+1j} W_{-1j}}. \tag{11}$$

This value $Z = 2 \sum_{j=1}^{n_c} \sqrt{W_{+1j} W_{-1j}}$ can be a barometer of performance improvement when a new weak classifier is added. RealAdaboost pick up the feature which minimizes Z value and connect it to main classifier.

In practical implementation, because image dataset varies by boot-strapping, we use a variable ω_i in place of $\exp(-z_i F^{t+1}(\boldsymbol{y}_i))$ for each sample $\boldsymbol{y}_i$, and update the values iteratively to calculate Z value.

2) Tree Structured Classifier and Head Pose Clustering: Creating many classifiers corresponding to many pose classes to detect various poses of head causes heavy computational cost. Our approach creates a tree structured classifier for multi-pose classification at one time. Fig. 8 illustrates cascaded classifiers for multi-pose classification. Near the root of the tree, some rectangle features are shared among multiple pose classes near the root, and then detailed classification is decided after branching. This approach fasten head detection because it reduces number of rectangle features.

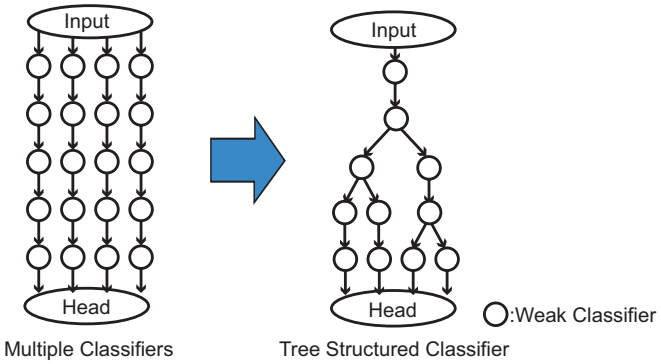

Fig. 8. Tree Structured Classifier

3) Training a Tree Structured Classifier: Wu's method [14] also trains tree structured classifier with dividing the dataset. It divides the dataset according to Z value, which denotes classification performance. If Z is higher than the threshold, the dataset is divided into two classes, and the classifier for each class is trained continuously. Then, trained classifier becomes like tree of weak classifiers as shown in Fig. 8. Wu describes that automatic clustering of dataset makes the performance higher than previous method with pre-defined pose classes. Classifier is trained as Fig. 9.

In Fig. 9, ψ means number of classes, Ψ is maximum number of classes, R is the target false positive rate and $\theta_{v,t}$ is the threshold for rejection of each weak classifier $f_{v,t}$. Firstly all of the dataset belong to single class, then the dataset is divided and a number of classes increases with adding a new weak classifier for each class. Trained classifier is structured like tree finally.

4) Head Pose Clustering Depending on Classification Performance: Although Wu's method employs k-means clustering to divide the dataset, we divide the head pose class according to the loss function after division. To split the dataset S_{+1v} belonging to class v into two classes, S_{+1A} and S_{+1B}, border value of two classes has to be searched. The head pose is expressed by yaw and pitch angles, and the border value is Θ_y or Θ_p correspondingly. When the dataset S_{+1v} is divided into S_{+1A} and S_{+1B} and new weak classifier for each is added, loss function (7) to decide the border value is as follows.

$$L^{t+1} = \sum_{j=1}^{n_c} (W^A_{+1j} e^{-c^A_j} + W^A_{-1j} e^{c^A_j} + W^B_{+1j} e^{-c^B_j} + W^B_{-1j} e^{c^B_j}) \tag{16}$$

$$W^A_{+1j} = \sum_{i:\boldsymbol{y}_i \in S_{+1A}, h(\boldsymbol{y}_i) \in u_j} \omega_i^{(t)} \tag{17}$$

$$W^A_{-1j} = \frac{|S_{+1A}|}{|S_{+1A} + S_{+1B}|} \sum_{i:\boldsymbol{y}_i \in S_{-1v}, h(\boldsymbol{y}_i) \in u_j} \omega_i^{(t)} \tag{18}$$

where outputs of new weak classifiers are c^A_j, c^B_j. W^B_j is defined in the same way as W^A_j. This function is minimized when

$$\min_{c^A_j, c^B_j} L^{t+1} = 2 \sum_{j=1}^{n_c} (\sqrt{W^A_{+1j} W^A_{-1j}} + \sqrt{W^B_{+1j} W^B_{-1j}}) \tag{19}$$

1) All weights of samples are initialized as $\omega_i^{(1)} = 1/N$.
2) For $t = 1$ to T, do
   a) For all classes $v = 1, \ldots, \psi$, do
      i) For all candidates of weak classifier, compute following.

$$W_{lj} = \sum_{i:z_i = l, h(\boldsymbol{y}_i) \in u_j} \omega_i^{(t)} \tag{12}$$

$$Z = 2 \sum_{j=1}^{n_c} \sqrt{W_{+1j} W_{-1j}} \tag{13}$$

      ii) Select the weak classifier with smallest Z

$$f_t = \operatorname*{argmin}_f Z \tag{14}$$

      iii) Train the weak classifier by (10).
      iv) Update the weights of samples by

$$\omega_i^{(t+1)} = \omega_i^{(t)} \exp(-z_i f_t(\boldsymbol{y}_i)) \tag{15}$$

      v) Normalize the weights of samples.
      vi) Learn the threshold $\theta_{v,t}$ to reject as many negative samples as possible.
      vii) Remove the rejected samples from the dataset, and recollect samples as needed.
      viii) Finish the training for this class if false detection rate is smaller than R.
      ix) If $Z > \theta_Z$ for 3 times running about class v and $\psi < \Psi$, then divide the dataset of class v.
         A) Divide the dataset by some clustering method, and assign the labels v and $\psi + 1$ to the 2 new classes.
         B) Retrain all the previous weak classifiers for the new datasets.
         C) Add number of classes ψ.

Fig. 9. Tree Structured Classifier Training by RealAdaboost [14]

We define (19) as $\tilde{Z}$, and search for the border value to minimize $\tilde{Z}$ in order to split dataset for high-performance classification. Now we assume to search the border value Θ_y and divide the dataset depending on yaw angles for simplicity. To compute $\tilde{Z}$ value for all weak classifiers and for all candidates of border value, its computational cost is $O(N^2 \times n_h)$ by naive approach. Compared to the computational cost for feature selection $O(n_h \times N)$, it takes too much. We propose the method to search the border value fast by storing W_{+1j} values for each weak classifier and each border value. Our algorithm is described in Fig. 10.

To minimize (19), loss on a part of the dataset is computed for each weak classifier and minimum loss is stored as $M_i^{\mathrm{inc}}, M_i^{\mathrm{dec}}$. Then, optimal split is searched by minimizing (20). Sorting samples and storing minimum loss reduces

1) Prepare the storages $D^{\mathrm{inc}}_{r,j}, D^{\mathrm{dec}}_{r,j}, r = 1, \ldots, n_h, j = 1, \ldots, n_c$
$M^{\mathrm{inc}}_i, M^{\mathrm{dec}}_i, i = 1, \ldots, |S_{+1k}|$, where n_h denotes the number of weak classifiers.
2) Compute $W^{(r)}_{-1j}$ for all weak classifiers $h_r, r = 1, \ldots, n_h$ by negative samples.
3) Sort positive samples of the dataset S_{+1v} in ascending order by yaw angle.
4) For $i = 1, \ldots, |S_{+1k}|$, do
   a) For each weak classifier $h_r, r = 1, \ldots, n_h$, do
      i) If $h_r(\boldsymbol{y}_i) \in u_j$, add ω_j to $D^{\mathrm{inc}}_{r,j}$.
      ii) Compute $\hat{Z} = 2\sum_{j=1}^{n_c} \sqrt{D^{\mathrm{inc}}_{r,j} \frac{i}{|S_{+1v}|} W^{(r)}_{-1j}}$.
   b) Store the minimum $\hat{Z}$ as M^{inc}_i.
5) Then, sort samples in descending order.
6) Store M^{dec}_i by the same way.
7) Compute $\tilde{Z}$ about $i = 1, \ldots, |S_{+1k}|$ as follows.

$$\tilde{Z} = M^{\mathrm{inc}}_i + M^{\mathrm{dec}}_{|S_{+1v}|-i} \tag{20}$$

8) Search the sample i minimizing $\tilde{Z}$, and the border value is its pose parameter.

Fig. 10. Searching for the Border Value of Head Pose Classes

computational cost to $O(n_h \times N)$.

Fig. 11 shows the difference of head detection rate between Wu's method and proposed method. Experimental data is captured by ceiling cameras. There is one head in a test image like Fig. 13, and test samples are captured by sliding window. Correct head position is captured by optical motion capture. Fig. 11 denotes that improvement of dataset clustering approach enhances head detection rate.

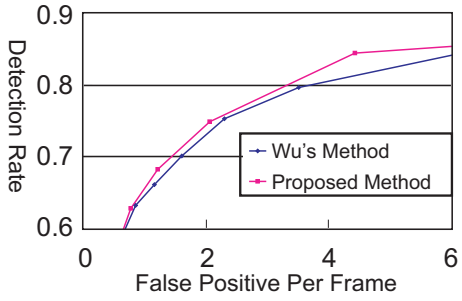

Fig. 11. ROC Curve of Head Detection

B. Integration of Multi-View Detection

3D head position is estimated as follows.

1) Preparing 3D Position Candidates: After head detection on each camera image, candidates of 3D head position can be calculated by triangulation using a pair of detections. In multi-view camera system, human face can be seen by only a few cameras actually, then using two views to estimate 3D position reduces the influences of false detections. If there are more than one candidates, each candidate is evaluated about multi-view consistency. Moreover, we apply image tracking based on subspace tracking [15] to track undetected head. This image tracking is applied to the view which was used to estimate 3D position in previous frame.

2) Head Pose Estimation: Before deciding the optimal candidate, pose of each candidate is roughly estimated by multi-view classification result. Global head pose is also expressed by yaw and pitch angles as local head pose in image is. A global pose is evaluated by consistency of classification results on multi-view head images.We use how many weak classifiers the input image passes as classification score, therefore the consistency about each global pose is computed as Fig. 12.

1) Prepare sufficient pose candidates expressed by yaw and pitch.
2) For each pose candidate, do
   a) Compute the corresponding pose class in each view, and count how many weak classifiers of the class are passed.
   b) Decide the consistency score from the mean of three maximum passage rates.
3) The most appropriate pose is estimated to be the candidate which produces the highest consistency score.

Fig. 12. Head Pose Estimation based on Multi-View Classification

3) Evaluation of Candidates via Pose Consistency: Appropriate head pose is estimated for each 3D position candidate. Likelihood of each candidate is defined based on consistency score q as follows.

$$p_{\mathrm{head}}(q) = \rho^{1-q} \tag{21}$$

where ρ means head likelihood of random image, and is set to 10^{-7}. The higher $p_{\mathrm{head}}(q)$ is, the more appropriate the candidate is. We apply this evaluation function to temporal filtering based on dynamic programming [16], then most appropriate 3D head position is selected.

4) Evaluation of Head Position Estimation: We evaluate head position estimation on three movie sequences, normal walking, dynamic walking and sitting down. Our method based on multi-view classification consistency is compared with Potamianos' method [16], which evaluate each candidate about Bhattacharyya distance of color histograms. Both methods use our head detector. Table I shows the results of evaluation. There are number of sequence frames, mean error in all frames [mm], number of frames where position error is over than 100mm and its percentage to all frames in the table. Numbering 1 denotes our method and 2 denotes Potamianos' method. 100mm is the threshold to judge whether estimated position is inside of head in camera images. The table shows that our approach improves the accuracy about mean error, furthermore number of frames with over 100mm error is

only 2%. Therefore our method achieves enough accuracy to reconstruct human shape robustly.

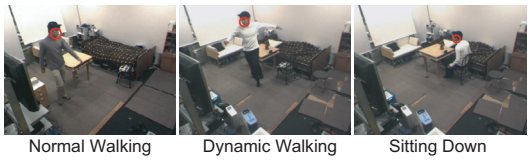

Fig. 13. Examples of Head Position Estimation

TABLE I
ESTIMATION ERROR OF HEAD POSITION

Scene	Normal Walk	Dynamic Walk	Sit Down
Num. of Frames	342	238	661
Mean Error 1	**38mm**	**36mm**	**45mm**
Mean Error 2	44mm	37mm	48mm
Over 100mm 1	**6 (1.7%)**	6 (2.5%)	**14 (2.1%)**
Over 100mm 2	12 (3.5%)	**1 (0.4%)**	23 (3.5%)

V. EXPERIMENTAL RESULTS OF HUMAN POSE ESTIMATION

We made experiments to estimate human pose in living environment by proposed shape reconstruction method. 8 cameras are equipped on the ceiling of the room and they captures 640 $\times$ 480 image. In silhouette extraction process, captured image is compressed to 160 $\times$ 120, and computational time is reduced to achieve almost 20 FPS processing for each camera.

Fig. 14 shows three examples of our experiments. From top down, it shows the example of holding a bag, turning on a desk light and pulling up a chair. Shape reconstruction by simple background subtraction (middle column) is affected by environmental changes and reconstructs excess shape. On the other hand, our method based on head position (right column) achieves better shape reconstruction with removal of effect of environmental changes. Fig. 14 shows the sequential pose estimation results on a scene with background changes. In this scene, a man brings a bag on a table and then moves a chair. Pose estimation results are superimposed on the input images. When simple background subtraction applied, some excess shape caused by background changes affect pose estimation results continuously. In contrast, proposed method provides successful results because it suppresses the effect of background changes and updates background information soon. In addition, we tried other scenes such as bringing a chair, taking off and putting on a jacket, reading a book and folding a blanket. Although pose estimation fails in a few scenes, it recovers soon through moving of human and it does not collapse.

VI. CONCLUSION

In this work, we propose a novel human pose estimation via head position information. Though it has been difficult to segment human silhouette and neighboring background changes, it can be solved by the iterative segmentation approach based on graph-cuts with head position. We also

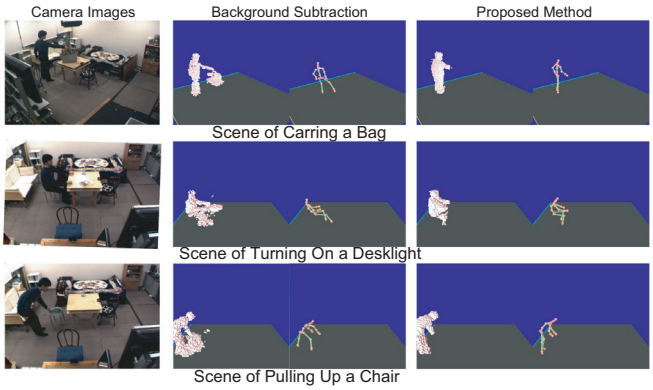

Fig. 14. Examples of Human Shape Reconstruction and Pose Estimation

contribue to develop robust 3D head position tracker by boosting. High-precision multi-class head detector and 3D position estimation based on multi-view consistency are leveraged as the 3D head position tracker. The experimental results show that use of head positions is appreciated to reconstruct human shape in spite of background changes.

REFERENCES

[1] T. Moeslund, A. Hilton, and V. Krüger. A Survey of Advances in Vision-Based Human Motion Capture and Analysis. *CVIU*, 104(2–3):90–126, 2006.
[2] M. Shimosaka, Y. Sagawa, T. Mori, and T. Sato. 3D Voxel Based Online Human Pose Estimation via Robust and Efficient Hashing. In *Proc. of ICRA*, pages 3577–3582.
[3] J. Sun, W. Zhang, X. Tang, and H. Shum. Background Cut. In *Proc. of ECCV*, volume 3952, pages 628–641, 2006.
[4] S. Seitz and C. Dyer. Photorealistic Scene Reconstruction by Voxel Coloring. In *Proc. of CVPR*, page 1067, 1997.
[5] P. Narayanan, P. Rander, and T. Kanade. Constructing Virtual Worlds using Dense Stereo. In *Proc. of ICCV*, 1998.
[6] T. Feldmann, L. Dießelberg, and A. Wörner. Adaptive Foreground/Background Segmentation Using Multiview Silhouette Fusion. In *Proc. of DAGM*, pages 522–531, 2009.
[7] M. Bray, P. Kohli, and P. Torr. PoseCut : Simultaneous Segmentation and 3D Pose Estimation of Humans Using Dynamic Graph-Cuts. In *Proc. of ECCV*, volume 3952, pages 642–655, 2006.
[8] P. KaewTraKulPong and R. Bowden. An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow. In *Proc. of AVBS*, 2001.
[9] N. Xu, R. Bansal, and N. Ahuja. Object Segmentation Using Graph Cuts Based Active Contours. In *Proc. of CVPR*, volume 2, pages 46–53, 2003.
[10] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Proc. of CVPR*, pages 511–518, 2001.
[11] S. Li and Z. Zhang. FloatBoost Learning and Statistical Face Detection. *IEEE Trans. on PAMI*, 26:1112–1123, 2004.
[12] R. Schapire and Y. Singer. Improved Boosting Algorithms Using Confidence-rated Predictions. *Machine Learning*, 37:297–336, 1999.
[13] B. Wu, H. Ai, C. Huang, and S. Lao. Fast Rotation Invariant Multi-View Face Detection Based on Real Adaboost. In *Proc. of FGR*, pages 79–84, 2004.
[14] B. Wu and R. Nevatia. Cluster Boosted Tree Classifier for Multi-View, Multi-Pose Object Detection. In *Proc. of ICCV*, pages 1–8, 2007.
[15] D. Ross, J. Lim, R. Lin, and M. Yang. Incremental Learning for Robust Visual Tracking. *IJCV*, 2008.
[16] G. Potamianos and Z. Zhang. A Joint System for Single-Person 2D-Face and 3D-Head Tracking in CHIL Seminars. In *Proc. of CLEAR*, pages 105–118, 2007.