Video-Rate Hair Tracking System Using Kinect

Kazumasa Suzuki, Haiyuan Wu, and Qian Chen

Faculty of Systems Engineering, Wakayama University, Japan
suzuki@vrl.sys.wakayama-u.ac.jp, {wuhy,chen}@sys.wakayama-u.ac.jp

Abstract. In this paper, we propose automatic hair detection and tracking system that runs at video-rate (30frame per-second) by making use of both the color and the depth information of the images obtained from a Kinect. Our system has three characteristics: 1) Using a 6D feature vector to describe both the 3D color feature and 3D geometric feature of each pixel uniformly; 2) Classifying pixels in images into foreground (e.g. hair) and background with K-means clustering algorithm; 3) Automatic selecting and updating the cluster centers of foreground and background before and during hair tracking. Our system can track hair of any color or style robustly in clustered background where some objects have color similar to the hair, or in environment where the illumination changes. Moreover, our algorithm can be used for tracking a face (or head) if the face (skin+hair) is selected as foreground.

Keywords: Hair Tracking, Hair Detection, Kinect, Video-Rate, the color and the depth information.

1 Introduction

The appearance of hair in image carries important information about people. In the case that the hair part of a head can be detected and tracked, the performance of head detection, personal identification, head pose estimation and many other facial image recognitions that use both the hair and the face will be improved significantly over the ones that only use face information (see Fig.11 and Fig.10). Compared with the face region, since there are not many constant features in hair region and the hair style and the color can be changed easily, it is difficult to build a model or to training some useful features for detecting or tracking hair. As a result of it, the researches about hair detection are much fewer than the ones about face detection, and we could not find any reports about tracking hair at video rate.

Y.Yacoob et.al.[1] detect the face and eyes from a frontal face image, than build its skin color model and hair color model by fitting a head model on to the detected face. The region having color similar to the hair color model is detected as hair. K.Kee etc.[2] present a probabilistic graph model to segment the hair and face regions from the background. This approach extends the traditional segmentation algorithms, such as Graph-Cut and Loopy Belief Propagation, by incorporating the color and location model. This method cannot realize video

S. Tominaga, R. Schettini, and A. Trémeau (Eds.): CCIW 2013, LNCS 7786, pp. 212–223, 2013.

rate tracking because of its high computation cost. P.Julian etc.[3] use a simple statistic model and active contour to detect an initial hair region. The color and the texture of hair are trained from the initial hair region, which is then used to detect the hair region with a pixel-wise segmentation. This method depends on the detail texture of the hair thus can only be applied to high resolution images.

Hua etc.[6] propose a tracking algorithm called as *K-means tracker*. By using a 5D feature vector to describe the color and the position of a pixel, a variable ellipse model to describe the search area, a per-frame K-means clustering based segmentation and automatic cluster center selection and update techniques, they realized video rate tracking for rigid and non-rigid objects. However, the K-means tracker often shows unstable behavior in the following situations.

- When some background objects around the target have color similar to the target, the pixels of them will be mistakenly classified as target pixels and the incorrect tracking result will be given.
- When the size of the target object becomes very big during tracking, the distance between the target pixels near the boundary and the target cluster center may become longer than the distance to the background cluster centers. In this case, those pixels will be mistakenly classified as non-target pixels.
- Another problem is that the K-means tracker needs some methods to specify the initial cluster centers before tracking.

In this paper we propose a method that can track hair regions of heads at video rate by solving the above problems of K-means tracker. In order to realize automatic initialization for the K-means tracker, we develop a high-speed face detection algorithm by making use of range information obtained from a 3D imaging sensor, such as a stereo camera or a Kinect. After the face has been detected, we initialize the cluster centers of foreground (e.g. hair) and background by using both the color and depth information. In order to solve the first problem, we introduce the depth information as an additional property of pixel and describe each pixel with a 6D feature vector that consists of 3D color elements and 3D world coordinates. By performing the K-means clustering in this 6D feature space, the background objects having color similar to the target that appear around the target in the image can be separated from the target effectively. For the second problem, we describe the 3D shape of a head with a spheroid. Before performing the K-means clustering to segment pixels in an image, we first transform the image coordinates (x, y) into 3D world coordinates by using the depth information. This makes the resulted distance not be influenced by the size of the target in the image. Since the detection and tracing of hair region is based on pixel-wisely classification, the hair of any style can be detected and/or tracked stably.

If the initialization of target and non-target cluster centers is specified manually, our method can track arbitrary objects.

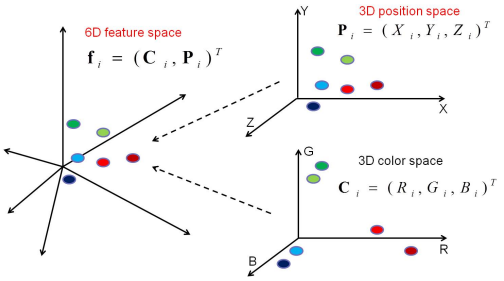


Fig. 1. 6-dimensional extension feature space

2 Pixel-Wise Clustering in 6-Dimensional Feature Space

In this paper, we assume that the target (e.g. hair) and background are spatially separated in the 3D space. We make full use of the color and depth information from a Kinect, and describe each pixel with a 6 dimensional feature vector: (R, G, B, X, Y, Z) (See Fig. 1), which is an extended version of the one in the original K-means tracker, where the feature vector is 5 dimensions (R, G, B, x, y). . By introducing depth information for describing pixels, we can tell the difference between the background objects that have similar color to the target even when they appear near the target object in the image, which was not possible in the original K-means tracker.

The position of a pixel (x, y) in an color image is first transformed to the 3D world coordinates system according to its Z in depth image and the calibrated camera parameters.

$$\mathbf{p}_2 = [x\, y]^{\mathrm{T}} \rightarrow \mathbf{p}_3 = [X\, Y\, Z]^{\mathrm{T}} \tag{1}$$

In the newly defined 6D feature space, each pixel is described with a 6D feature vector $\mathbf{f}_6 = [\mathbf{c}_3\, \mathbf{p}_3]^{\mathrm{T}}$ that presents its color $\mathbf{c}_3 = [R\, G\, B]^{\mathrm{T}}$ and the position in 3D world space $\mathbf{p}_3 = [X\, Y\, Z]^{\mathrm{T}}$ simultaneously. The distance $d(\mathbf{f}_6^a, \mathbf{f}_6^b)$ between pixel a and b is defined as the following formula.

$$d(\mathbf{f}_6^a, \mathbf{f}_6^b) = ||\mathbf{c}_3^a - \mathbf{c}_3^b||^2 + \alpha||\mathbf{p}_3^a - \mathbf{p}_3^b||^2, \tag{2}$$

where α is a constant weight factor for adjusting the balance of a color ingredient and a distance ingredient. $||\mathbf{c}_3^a - \mathbf{c}_3^b||^2$ is the Euclid distance in 3D color space, and $||\mathbf{p}_3^a - \mathbf{p}_3^b||^2$ is the Euclid distance in 3D world space.

Although the shape, size and hair style of human heads differs from each other, the 3D volume that a head occupying can be approximated with a spheroid with constant long axis and short axis. Therefore, by estimating the distance in between pixels in 3D space, we can separate the target from the background objects similar color and can remove the influence of the unstable behavior caused by the change of target size in image.

In this paper, the shortest distance D_T from an unknown pixel $\mathbf{f}_6^u$ to the target center $\mathbf{f}_6^T$, and the shortest distance D_{NT} from $\mathbf{f}_6^u$ to the non-target cluster centers $\mathbf{f}_6^{NT}$ are defined as

$$D_T(\mathbf{f}_6^{\mathbf{u}}) = \min_{i=1\sim n}\{d(\mathbf{f}_6^{\mathbf{T}_i}, \mathbf{f}_6^{\mathbf{u}})\}, \tag{3}$$

$$D_{NT}(\mathbf{f}_6^{\mathbf{u}}) = \min_{j=1\sim m}\{d(\mathbf{f}_6^{\mathbf{NT}_j}, \mathbf{f}_6^{\mathbf{u}})\}, \tag{4}$$

Where, $\mathbf{f}_6^{\mathbf{T}_i}$ and $\mathbf{f}_6^{\mathbf{NT}_j}$ are target cluster center and non-target (background) cluster center, n and m is the number of target clusters and non-target (background) clusters, respectively. As show in Fig.5(a), the pixel $\mathbf{f}_6^u$ will be classified as a target pixel if D_T is less than D_{NT}, otherwise as a non-target pixel.

$$\mathbf{f}_6^u \rightarrow \begin{cases} \text{traget}; & \text{if} \quad D_T(\mathbf{f}_6^{\mathbf{u}}) < D_{NT}(\mathbf{f}_6^{\mathbf{u}}) \\ \text{non} - \text{traget}; & \text{otherwise} \end{cases} \tag{5}$$

3 Automatic Initialization of Cluster Centers for K-Means Clustering

3.1 Accelerating of Face Detection with Depth

There are two possible approaches to make face detection fast. The first one is to reduce the processing time for classifying each sub-image region. Many methods have been proposed for this purpose[8]. The second one is to reduce the number of sub-image regions to be classified in an image. In this research, we concentrate our attention on the second approach. We use the depth information (obtained from a Kinect) to reduce the number of the sub-image regions to be classified for face detection.

Assuming that a head has a constant 3D size W, the size of a face in the color image w can be calculated from the depth Z of the head with eq.(6).

$$w = \frac{f}{Z}W, \tag{6}$$

Where, f is the focal length of the camera. The depth Z can be obtained from range image directly (from Kinect). Therefore, we only need to classify the sub-image region of the estimated face size w at each position in an color image for detecting faces (See figure2). We use the library function of OpenCV[8] to detect frontal faces. Since the most unnecessary classifications can be effectively avoided, both the processing time for face detection and the false positive rate can be reduced significantly.

3.2 Automatic Initialization of Foreground Cluster Centers

In order to start tracking of hair using the method described in section **2**, we need a method to determine the initial cluster centers of foreground (e.g. hair) and background. In this research, we determine the necessary cluster centers by using the result of face detection.

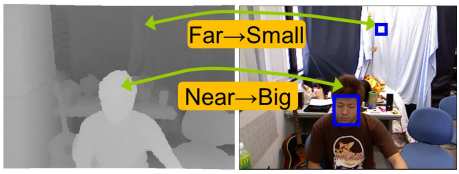


Fig. 2. Accelerating of face detection

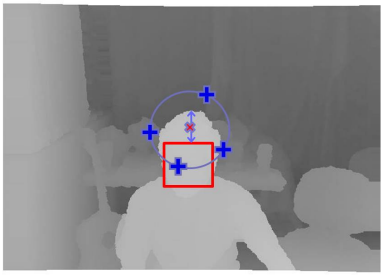


Fig. 3. Initialization of hair tracking

When a face has been detected, as shown in Fig.3, we will have a (red) rectangle of the detected face. Then, we find out the boundary of the face region in the depth image, and find out the intersection of the head boundary and the vertical line of the upper side of the face rectangle that crosses the mid-point of the upper side. The mid-point between the intersection and the mid-point of the upper side of the face rectangle is determined as the initial cluster center of the hair (red cross). The circle centered at the hair cluster center and its radius equals to the width of the face rectangle is used as the initial variable elliptic search area of the hair region.

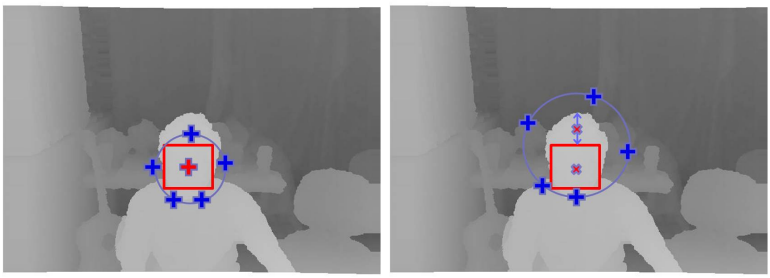
In the case of tracing hair, the number of hair clusters is ONE for one head. It will not change during tracking.

The initial background cluster centers (blue plus) are automatically determined on the circumference by using the method. The number is determined automatically according to the surrounding background. The details are given in the next section (Section 4). The number may change during tracking. By allowing changeable number background clusters, the hair tracking became more stable.

The target of this tracking method is not limited to hair. It can be used to track a face or a head (face+hair).

In the case of tracking a face, the number of face clusters is ONE for one head too. We use the center of the face rectangle as the initial cluster center of the target (face), as shown in Fig.4(a). The circumscribed circle (blue circle) of the face rectangle is used as the initial variable elliptic search area of the face. Similar to the case of tracking hair, the initial background cluster centers (blue plus) are automatically determined on the circumference.

In the case of tracking a head, the number of head clusters is TWO for one head. One is the hair cluster center and another is face cluster center, as described



(a) Initialization of face tracking; (b) Initialization of head tracking

Fig. 4. Initialization of face tracking or head tracking

in the case of hair tracking and face tracking, as shown in Fig.4(b). We let the circle center at the mid-point of the hair cluster center and the face cluster center and its radius equal to the distance between the center and the lower right corner of the face rectangle is the initial variable elliptic search area of the head region. The initial background cluster centers ("blue +") are automatically determined in the similar way as the case of face tracking or hair tracking.

4 Automatic Updating Non-target Cluster Centers during Tracking

The algorithm for the initialization non-target cluster centers and its update during tracking is based on the one of Oike et.al[4] and is extended to use the new 6D feature space. The flowchart of updating non-target cluster centers is shown in Fig.5(b). The number of the non-target cluster center candidate point s, i.e., the pixel count on a search area ellipse. The l^{th} candidate point is set to s_l, and its 6-dimensional feature vector is described with a simplified expression as $\mathbf{f}_{s_l}$.

Since the distance between the "already-selected" non-target cluster centers and the s_l is needed. First, we need to define the $1st$ point for a non-target cluster center on the search area elliptical outline. This can be done, for example, by finding the intersection of the horizontal axis and the ellipse.

$$\mathbf{f}^*_{N_1} = \mathbf{f}_{s_1}, \tag{7}$$

Where, $\mathbf{f}^*_{Nj}$ is a 6D feature vector of j^{th} non-target (background) cluster center.

Next, we calculate the shortest distance $D_T(\mathbf{f}_{s_l})$ from s_l to the target cluster centers, and the shortest distance $D_{NT}(\mathbf{f}_{s_l})$ from s_l to the "already-selected" non-target cluster center.

$$D_T(\mathbf{f}_{s_l}) = \min_{i=1\sim n}\{d(\mathbf{f}_{s_l}, \mathbf{f}^*_{T_i}\}, \tag{8}$$

$$D_{NT}(\mathbf{f}_{s_l}) = \min_{j=1\sim m'}\{d(\mathbf{f}_{s_l}, \mathbf{f}^*_{N_i}\}, \tag{9}$$

Where, $\mathbf{f}^*_{T_i}$ is a 6D feature vector of i^{th} target (i.e., hair) cluster center. n is the number of target clusters, m' is the number of "already-selected" non-target cluster centers.

When the condition $D_T(\mathbf{f}_{s_l}) < D_{NT}(\mathbf{f}_{s_l})$ is true, it means that the pixel s_l is on the ellipse (background domain) is far from the existed non-target centers compared to the distance to the nearest target cluster. Then, the pixel s_l is count as a NEW non-target center.

$$\mathbf{f}^*_{N(m'+1)} = \mathbf{f}_{s_1} \qquad if \quad D_T(\mathbf{f}_{s_l}) < D_{NT}(\mathbf{f}_{s_l}) \tag{10}$$

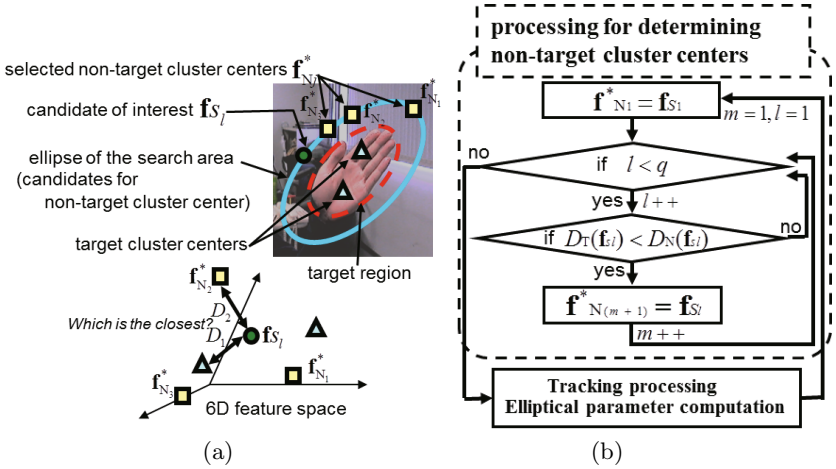


Fig. 5. (a) Comparison of distances in the 6-dimensional feature space; (b)Flowchart of updating non-target cluster centers. q is the number of pixels on the elliptical boundary of the search area.

This calculation is performed for each frame and all pixels $s_l; (l = 1, 2, \ldots q)$. Thereby, the arrangement and the number of non-target clusters suitable for track can be determined accommodative in each frame.

5 Experiments

5.1 Hair Tracking System

The system configuration of the experiment of the proposal method is shown in Fig.6.

In order to evaluate the performance of our hair tracking method, we applied to it several video sequences that containing heads with different hair style and hair color taken in many different environments. In all experiments, we confirmed that the tracking could be performed at video rate.

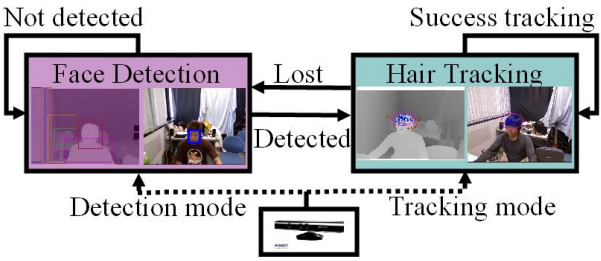


Fig. 6. System configuration

As shown in the figures from Fig. 7 to Fig.11, each ellipse is a variable ellipse search area of the object be tracked. The "Red +" shows the non-target cluster center. "Blue +" shows the target cluster center.

5.2 Results of Tracking Hair and Head in 3D

Fig.12 shows an example of tracking hair [1] and a head[2]. The blue pixels show the pixels that were classified as hair, and the orange pixels show the pixels that were classified as face. All the blue pixels and the orange pixels were show in a 3D space. As shown in the movies, our method could track the hair or the head robustly.

5.3 Results of Tracking Hair of Various Styles

Fig.7 shows an example of tracking multiple hair targets with different hair style and hair color[3]. There were three heads in the scene. The left one had long black hair, the right one had brown short hair and the farther mannequin had long brown hair. The blue pixels show the pixels which were clustered as hair. As shown in the movie, our method could track the hair regardless of color and style. The hair region could be extracted in pixel during tracking.

We have tested our system not only in the Lab. but also in the open campus of our university, where many people; including children (boys and girls), old men, young and old women, and the results were very good.

In the experiments, where our method was implemented as a single thread program on a general PC, we confirmed that it became difficult to keep video rate performance when trying to track more than three targets.

[1] See `http://www.wakayama-u.ac.jp/~wuhy/hair3D.wmv`
[2] See `http://www.wakayama-u.ac.jp/~wuhy/head3D.wmv`
[3] See `http://www.wakayama-u.ac.jp/~wuhy/MultiHair.wmv`

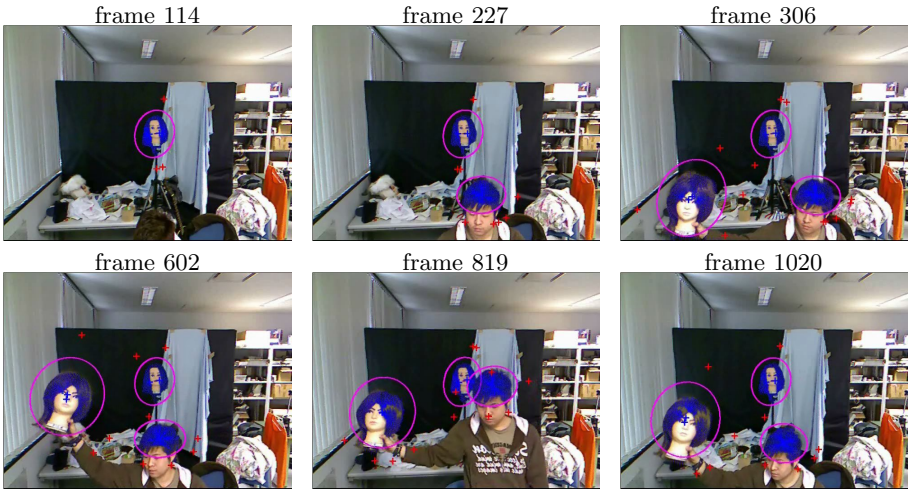
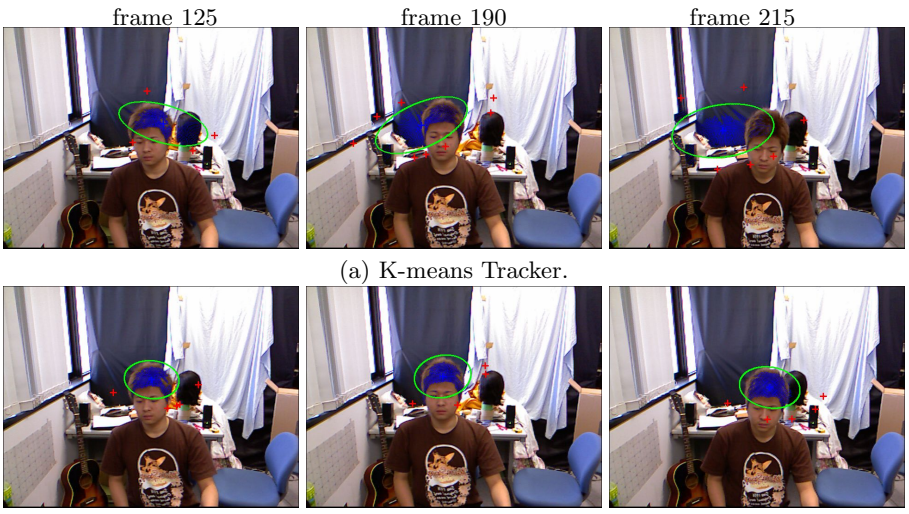


Fig. 7. Multiple target tracking



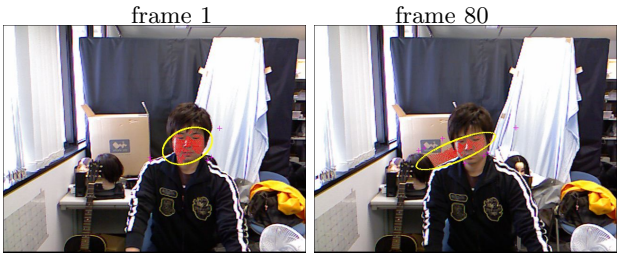
(a) K-means Tracker.

(b) proposed method

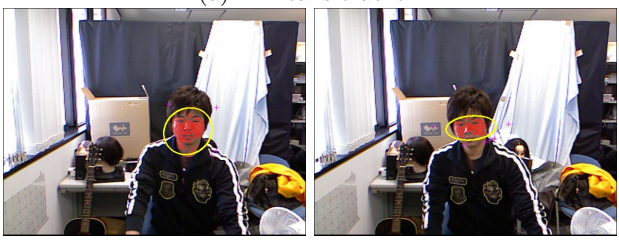
Fig. 8. The comparative experimental results of hair tracking with similar color background (black-out curtain)

5.4 Experiments for Testing the Influence of Existence of Background Objects Having the Color Similar to the Target

For testing the influence of existence of background objects having the color similar to the target, we compared the performance of our method with K-means tracker for the same video sequence.



(a) K-means tracker.



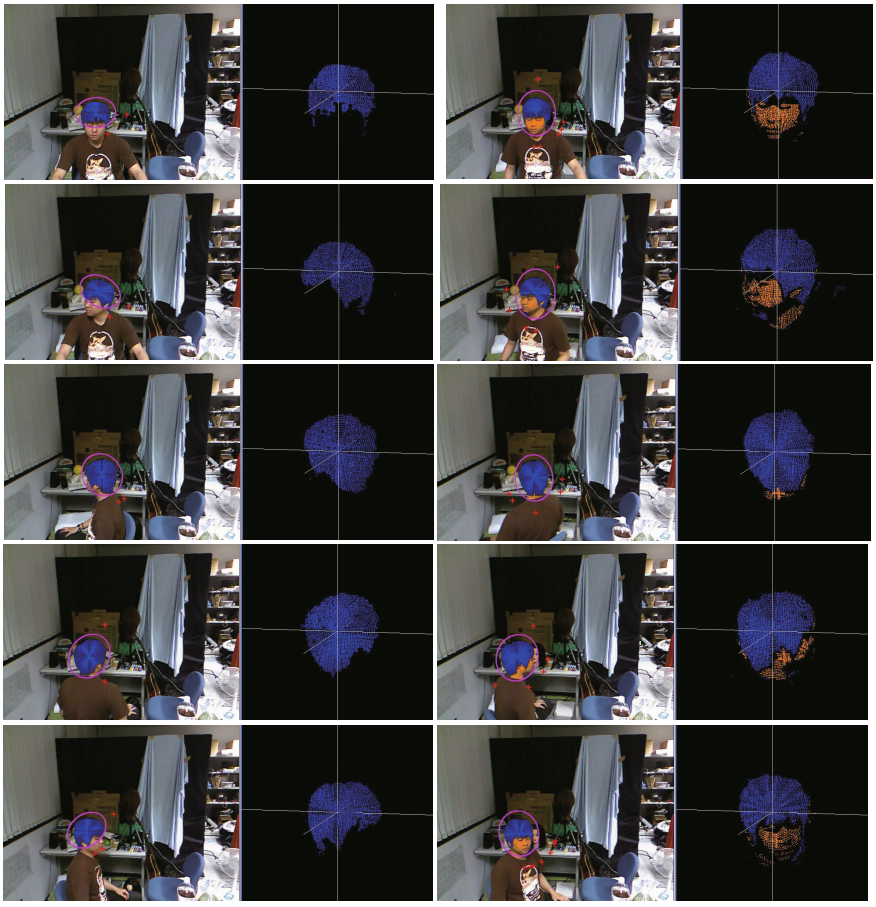
(b) our method.

Fig. 9. The comparative experimental results of face tracking with similar color background (corrugated paper box)

Fig. 10. Tracking a face only

Fig. 11. Tracking a face and its hair

Fig.8 shows the results of hair tracking. The blue pixels show the pixels that were classified as hair. The black curtain (background) had the similar color but different depth to the hair (target). The K-means tracker failed to update the target ellipse correctly while our method gave the correct results.



(a) tracking hair; (b) tracking head.

Fig. 12. Results of tracking hair and head in 3D

Fig.9 shows the results of face tracking. The results of K-means tracker are shown on the left column [4] in Fig.9, and the results of our method are shown on the right [5]. In Fig.9, the corrugated paper box had a similar color but a different depth to the target (face). When the target moved towards to the box, the K-means tracker classified some pixels of the box (background) as target pixels and then failed to update the target ellipse correctly, while our method worked stably for the same sequence.

[4] See `http://www.wakayama-u.ac.jp/~wuhy/old5DTrackingHair.m1v`
[5] See `http://www.wakayama-u.ac.jp/~wuhy/new6DTrackingHair.m1v`

5.5 Comparative Experiments of Tracking Face and Tracking Head

We also did some comparative experiments of tracking a face and tracking a head (face+hair), using our method The blue pixels show the pixels that were classified as face, and the orange pixels show the pixels that were classified as hair. [6]. As shown on the top of Fig.10, in the case of tracking of a face, when the face region became small or turned to back, the tracking would fail. However, as shown on the bottom of Fig.11, the tracking of the head (face+hair) was carried out successfully for the same sequence.

6 Conclusion

In this paper, we have proposed a video rate hair tracking algorithm. By making use of depth information, we can detect front face fast enough for initializing and starting the tracking of a hair region, a face or a head. Since the target cluster center are obtained automatically, the hair (or face, head) with arbitrary color can be tracked. By using the color and depth information to cluster each pixel into target or background, we can track hair with arbitrary hairstyles, and can track target (hair, or face, head) stably even when the search area is mixed with background pixels that have color similar to the target.

If the initializations of target and non-target cluster centers are specified manually, our method can track arbitrary objects, including non-rigid objects and objects holes, at video-rate.

Acknowledgments. This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), 24500205.

References

1. Yacoob, Y., Davis, L.S.: Detection and Analysis of Hair. PAMI 28(7) (2006)
2. Lee, K., Anguelov, D., Sumengen, B., Gokturk, S.: Markov random field models for hair and face segmentation. In: FG (2008)
3. Julian, P., Dehais, C., Lauze, F., Charvillat, V., Bartoli, A., Choukroun, A.: Automatic Hair Detection in the Wild. In: ICPR (2010)
4. Oike, H., Wu, H., Wada, T.: Adaptive Selection of Non-Target Cluster Centers for K-means Tracker. In: 19th Int. Conf. on Pattern Recognition (2008)
5. Suzuki, K., Qi, Y., Wu, H.: Tracking Face and Hair using Extended K-means Tracker, MIRU, Demo, pp.1699–1700 (2011)
6. Hua, C., Wu, H., Chen, Q., Wada, T.: K-means tracker: A General Algorithm for Tracking People. Journal of Multimedia 1(4), 46–53 (2006)
7. Wu, H., Suzuki, K., Wada, T., Chen, Q.: Accelerating Face Detection by Using Depth Information. In: Pacific Rim Symposium on Advances in Image and Video Technology, pp. 657–667 (2009)
8. Viola, P., Jones, M.: Robust Real-Time Face Detection. IJCV 57(2), 137–154 (2004)
9. OpenCV, `http://opencv.willowgarage.com/wiki/`

[6] See `http://www.wakayama-u.ac.jp/~wuhy/FaceOrHeadTracking.wmv`