BAP Generation for Head Movement based on MPEG-4 SNHC

Seung-Uk Yoon, Sung-Yeol Kim, and Yo-Sung Ho

Kwangju Institute of Science and Technology (K-JIST)
1 Oryong-dong, Puk-gu, Kwangju, 500-712, Korea
{suyoon, sykim75, hoyo}@kjist.ac.kr

Abstract

In this paper, we propose a body animation parameter generation system for global head movement using head motion analysis and tracking. The proposed system consists of two separate layers: head motion analysis layer and 3-D model registration layer. Following the MPEG-4 SNHC standard, we generate the global head motion using body definition and animation parameters. In the implemented system, we acquire head motion data from a single camera and extract body definition parameters from an arbitrary VRML human model.

1. Introduction

The objective of this paper is to construct a body animation parameter (BAP) generation system that can provide a general and systematic framework for three-dimensional (3-D) synthetic human representation following the MPEG-4 SNHC standard. Main problems in the 3-D synthetic character animation system include how to get the object motion information and how to provide compatibility of 3-D data of synthetic characters. This paper addresses those problems by connecting two main areas: vision-based head motion analysis and MPEG-4 based 3-D model representation.

Research on integrating human motion analysis based on computer vision and 3-D model animation based on MPEG-4 SNHC face and body animation (FBA) is in the early stage. The MPEG-4 standard defines an FBA object to specify synthetic human face and body models. FBA aims to provide tools for model-based coding of video sequences containing the human face and body. Instead of representing the face and body as coded 2-D images, FBA object assumes a synthetic model that can be defined and animated by specific FBA parameters. These parameters are typically extracted or synthetically generated, coded and transmitted. MPEG-4 specifies a rich set of FBA parameters to define the model, texture, and animation of the face and body [1].

Huang et al. [2] suggested a model-based human motion analysis system to track and analyze motion of the human object in a video sequence in terms of body definition parameters (BDPs) and body animation parameters (BAPs) for MPEG-4 video encoder. They used 3-D cylinders as primitives to construct a synthetic human body model. Their model consists of 10 joints and total 22 degree of freedom (DOF), as shown in Fig. 1. However, this model has limited extensibility because it is operated only in their system. In other words, the model should be integrated before the system starts, because BAPs are calculated according to the pre-integrated model. In addition, the number of DOF is small compared to 186 DOF in the MPEG-4 SHNC FBA standard. In this paper, we construct a VRML parser to get an arbitrary human body model for extensibility.

Fig. 1. Simplified human model

2. BAP Generation System

Although there are some papers dealing with the estimation of motion parameters based on the MPEG-4 standard, explicit generation of BAP is a different and new approach. Although the MPEG-4 standard defines BDPs and BAPs, there is no description about how to animate 3-D models and how to acquire BDPs or BAPs. Therefore, our goal in this paper is to generate BDPs and BAPs explicitly and to test the operation of the generated parameters. We exploit VRML-based BDP generation and feature-based BAP generation methods. In order to test the system operation, we implement a BAP player working with BAPs provided by MPEG-4 and generated by the proposed system.

2.1 System Overview

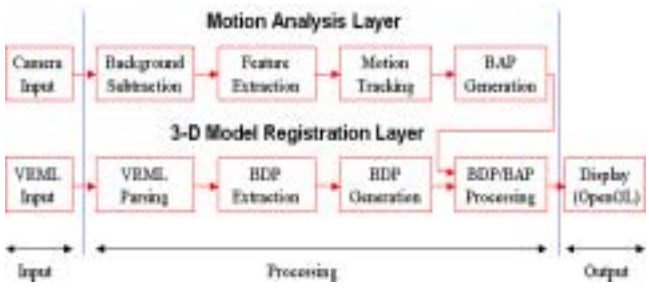


Fig. 2. Block diagram of the proposed system

The proposed system consists of three parts: input interface, data processing, and output part. Moreover, there are two main streams in our system: motion analysis layer and 3-D model registration layer.

2.2 Motion Analysis Layer

The objective of this stage is to generate body motion parameters explicitly and compactly. We employ a single USB camera, segment foreground objects, extract features from detected objects, track head motions, and finally generate BAPs. The preliminary constraints are: a single person for a camera, a limited range of distance between the camera and the human body, and indoor environments without any sharp illumination changes.

Input Interface: We use a USB PC camera as the input interface in our system to avoid problems of contact devices. Major benefits include cheap cost, easy manipulation, and user friendliness. However, the data accuracy is low. Fig. 3 shows the procedure of forming the input interface with one camera.

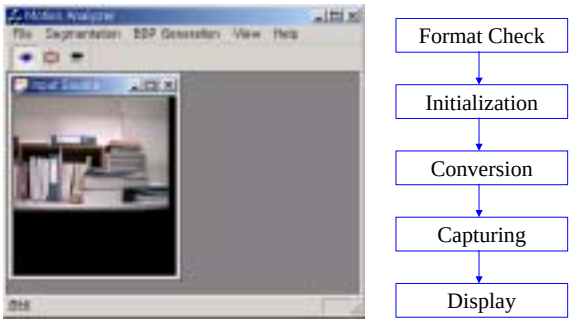


Fig. 3. Input interface

After our system detects the camera, we check whether it is a USB camera whose input format is I420 YUV type. Once the camera is properly detected and checked, it is initialized by the system. Then, input images are filed up in the system buffer. Because the buffer has only a limited size of one single image, the previous input image is overwritten by a new input one. Then, the CPU idle process is running to display the continuously coming input sequence into the monitor.

Background Subtraction: The second step is background subtraction. In our system, we exploit a simple and fast automatic subtraction method because we focus on motion parameter generation and movement analysis. We perform our experiment under indoor environments without severe illumination changes. Here, we assume there is no abrupt head movement; however, we allow reflection of lights, shadows, and complex background.

The specification of the input sequence is as follows: (1) the frame rate is 15 fps, (2) the image resolution is 160x120, and (3) the total 150 frames are used as the reference sequence. After capturing the reference frames, the statistical average and variance of each color channel (RGB) is calculated. We assume that background brightness levels are changing independently. If they have the normal distribution, background characteristics can be calculated by finding a sum of pixel values, $S_{(x,y)}$, and a sum of squares, $Sq_{(x,y)}$, for every pixel location (x,y). The average value is calculated by

$$m_{(x,y)} = S_{(x,y)} / N \tag{1}$$

where N is the number of the frames collected. The standard deviation is computed by,

$$\sigma_{(x,y)} = sqrt(Sq_{(x,y)} - (S_{(x,y)} / N)^2) \tag{2}$$

Using these statistical characteristics of each color channel, we can compute the average value and the variance of reference frames and store them [3]. Then, we can find foreground pixels by

$$sqrt(\sigma_i) > \alpha \cdot sqrt(\sigma_r) \tag{3}$$

where σ_i is the square root of the difference between the input image and the pre-calculated average mean image at (x,y), σ_r is the average variance of reference frames, and α is the subtraction threshold in our system. When Eq. (3) is satisfied, the pixel located at (x,y) in the current frame belongs to a moving object. After pixels in the incoming image are classified into two categories, foreground objects are represented by the white color. At the same time, background objects are represented by the black color.

Pre-processing and Feature Extraction: After classifying pixels into foreground and background objects, we find features from the segmented head image. In the previous work of Huang et al. [2], they applied the morphological operation in the pre-processing stage. They assumed that only one human object is extracted accurately. However, it is difficult to extract the exact foreground region in normal illumination environments. In addition, the center of a bounding box does not represent the object center properly. Since the meaning of the bounding box center is ambiguous, we use a contour-based approach to solve these problems.

In the beginning, our system finds every contour in the segmented image profile that contains all foreground objects. Each contour is represented by the chain code. Then, the system calculates moments of contours. Once the moment of a contour is computed, it is easy to calculate the contour area because the contour area can be directly calculated from the zero-th order moment of the contour. The moment of order $(p;q)$ of an arbitrary region R is given by [3].

$$v_{pq} = \iint_R x^p \cdot y^q dxdy \tag{4}$$

Then, the appropriate contour is selected by applying area constraints. After finding the proper region, we determine the centroid by computing contour moments. The mass center is expressed by

$$x_c = M_{10} / M_{00},\ y_c = M_{01} / M_{00} \tag{5}$$

where M_{00} is the zero-th moment and, M_{10} and M_{01} are the first moments for x and y, respectively.

Head Roll Motion Tracking: We track head roll motion that is depicted in Fig. 4. The head has a single joint, namely, a neck joint. Head roll movement is a type of the global motion.

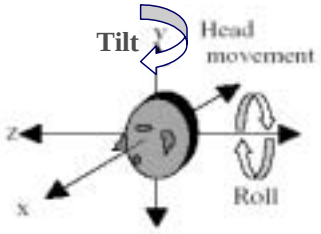


Fig. 4. Classification of head movement

We compute the orientation of the probability distribution for every input image using the Bradski's method [4]. We can calculate the orientation angle for head roll motion by

$$\theta = \frac{\arctan\left(\frac{2\left(\frac{M_{11}}{M_{00}} - x_c y_c\right)}{\left(\frac{M_{20}}{M_{00}} - x_c^2\right) - \left(\frac{M_{02}}{M_{00}} - y_c^2\right)}\right)}{2} \quad (6)$$

$$M_{20} = \sum_x \sum_y x^2 I(x,y)\,;\; M_{02} = \sum_x \sum_y y^2 I(x,y) \quad (7)$$

Head Tilt Motion Tracking: Head tilt movement is the rotation of the head based on the Y-axis, as shown in Fig. 4. In order to track head tilt motion, we compute head ratios. We assume the default posture and limited angle variations. Then, we calculate *Default Ratio*, *Max L_Ratio*, and *Max R_Ratio*. Moreover, we measure *L_Ratio* and *R_Ratio* of the head. Fig. 5 represents these parameters.

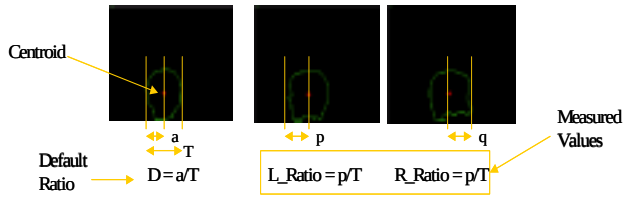


Fig. 5. Head ratio calculation for head-tilt movement

Criteria for determining the head direction are: if (L_Ratio/D) > 1, the direction is left; if (R_Ratio/D) > 1, the direction is right. Finally, we can get the angle of rotation by

$$\varphi(\text{deg}) = 90 \times L(R)\_Ratio / MaxL(R)\_Ratio \quad (8)$$

BAP Generation: As mentioned before, we track head roll and tilt motion. The joint related to those movements is the skullbase in BDP. The matched BAP is c7_roll whose BAP number is 116. The unit of each BAP value is expressed by

$$BAP = \theta(\varphi)[\text{deg}] \times 100000 / \pi \quad (9)$$

Since we calculate angles of head motions by Eq. (6) and Eq. (8), we can get BAP values by Eq. (9).

2.3 3-D Model Registration Layer

VRML Input and Parsing: We use an arbitrary virtual reality modeling language (VRML) 2.0 file of the human model as the input to the 3-D model registration layer. From the VRML file, we extract names, locations of joints and segments, face geometry, and connectivity data. These data construct the internal structure of the BDP file. The overall structure of our VRML parser is shown in Fig. 6.

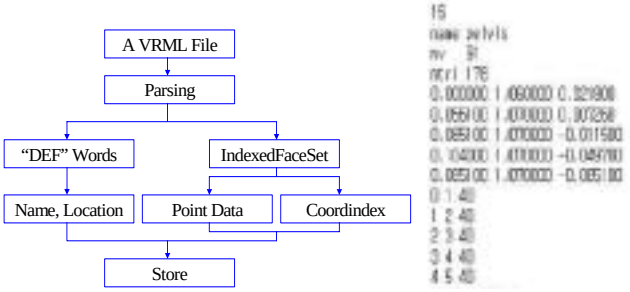


Fig. 6. The parsing procedure and a BDP structure

When we parse the point data, face geometry information, coordIndex, and connectivity of each face correctly, we can construct the BDP file from those data.

BDP Generation: After paring the VRML file, the system generates the BDP file automatically. The BDP file structure used in our system is shown in the right side of Fig. 6. The first line of the structure means the number of segments of the 3-D model. The segment name, the number of vertices, and the number of triangles appear for each segment. Then, face geometry data and connectivity information are stored. This block structure is repeated until reaching the last segment. Since the file structure of BDP is not defined in the MPEG-4 SNHC standard, we generate our own format based on the definition of BDP in the MPEG-4 standard.

BDP/BAP Parsing and Processing: To display the human model movement in the screen, it is needed to read and analyze BDP and BAP data. Detail procedures for reading and processing of the BDP and BAP file are as follows.

BDP Reading and Processing: In order to use BDP data in the OpenGL environments, we should calculate the normal vector of each face and the scaling factor of the 3-D human model. The system computes two vectors from the three vertices constituting a triangle face to calculate the normal vector of each face. Then, perpendicular vectors are calculated and the normalized normal vector of the face is obtained from those perpendicular vectors. Finally, the vertex normal is computed by averaging adjacent face normal vectors. This procedure is repeated for all triangular faces and vertices for all segments. The next process is the calculation of the model size factor. In order to show the human model with the proper size in the display screen, we compute differences between the maximum and minimum value of the model, normalize, and then scale them.

BAP Reading and Processing: BAP contains information about moving segments, the amount of movement for each segment, and the frame number. First of all, the system reads the BAP file generated from the motion analysis layer. While the BAP file is read, the system checks the BAP number whose value is one. It indicates the BAP relating to the head. Then, various tables and data are loaded into the system. They include segment numbers, segment names, joint numbers, joint names, a BAP type definition table, a BAP number and name recording table, and a BAP relation table. The BAP relation table includes that which BAP belongs to which segment and the type of BAP. BAP has one of eight different types and each type represents the rotational axis. These type data are contained in the BAP type definition table and the BAP relation table. The latter is constructed with the following format.

```
r[index].bap_num = 11

r[index].segment_num = 6

r[index].type = 4
```

Fig. 7. Format of the BAP relation table

Fig. 7 shows components of the BAP relation table. "r" stands for "relation" and the index has the value from 1 to 169. The total number of BAPs processed in our system is 169. Each number on the right side of the equal sign represents the BAP number, the segment number relating to that BAP, and the type number included in the type definition table, respectively. Then, the system reads the BAP value frame by frame. The system acquires the joint location, the rotation axis, and the angle of rotation by reading the above tables. Those data are employed to animate the 3-D synthetic model with OpenGL. The described procedure is shown in Fig. 8.

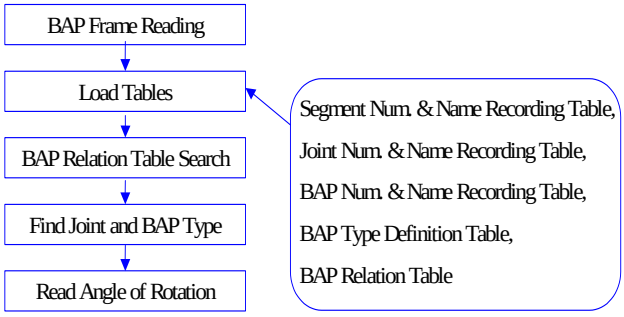


Fig. 8. BAP parsing and processing procedure

3. Experimental Results and Analysis

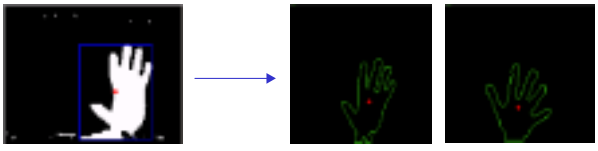
We generate BDPs from the VRML file and acquire BAPs by tracking head motions. In order to track head movement and represent the synthetic character, we design and implement two layers. Compared to previous approaches, our BAP generation scheme is explicit and direct.

3.1 Motion Analysis Layer

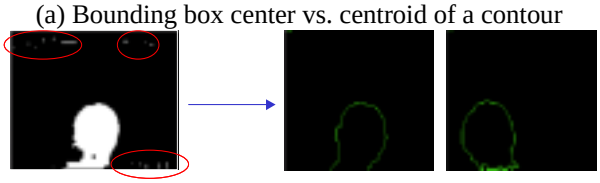
In the motion analysis layer, we use a USB PC camera as our input interface. The resolution of an image is 160x120 and each image has 3 color channels: R, G, and B channel. Then, we capture 150 reference frames, train them, and classify pixels of the incoming image. Fig. 9 shows the result of the background subtraction.

Fig. 9. Falsely detected foreground objects

In Fig. 9, we can see the falsely determined foreground pixels, which are located above the segmented hand. The unwanted region is caused by reflection of lights. The number shown in Fig. 9 is the threshold value. We exploit the contour-based area thresholding technique to remove these noises. Then, we compute not the bounding box center, but the centroid of the foreground object using contour moments. Here, we assume that the distance between a body object and a camera is limited and an area of a body object is bigger than noise regions. The found contour of the hand is shown in Fig. 10, where we can see that noises are properly removed by the area thresholding technique.



(a) Bounding box center vs. centroid of a contour



(b) Noise removal through the area thresholding

Fig. 10. Centroid calculation and area thresholding

3.2 3-D Model Registration Layer

In this layer, we employ a VRML file as an input. We test the system operation with the Nancy model, presented in Fig. 11. The Nancy model consists of total 16 segments and 17 joints. It is more elaborate than the previous model, depicted in Fig. 1. Since one of our goals is to construct a BAP player compatible with complex BAPs provided by the MPEG-4 standard, the model structure is important. Such a simplified model cannot represent complex body movement. Moreover, the previous model must be pre-integrated into the system; however our system does not require a pre-integrated model. In other words, our system has high extensibility. The extensibility is a key factor of the system because we intend to expand the current framework for the whole body movement.

Fig. 11. Test model: Nancy

3.3 Operation Test of the BAP Player

We construct a BAP player based on the MPEG-4 standard and test the operation with various BAPs. Fig. 12 and Fig. 13 show results of animation with BAPs provided by the standard. They are applause.bap and sit_r_talk.bap. In case of the complex BAP like sit_r_talk.bap, our BAP player work well. Fig. 14 represents the test result for BAPs generated by the motion analysis layer in our system. The left column shows the animation result for head roll movement. The right column is the result for head-tilt.bap.

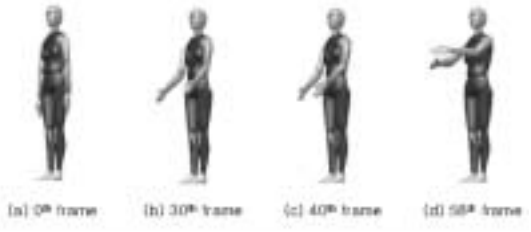


Fig. 12. Result for applause.bap provided by MPEG-4

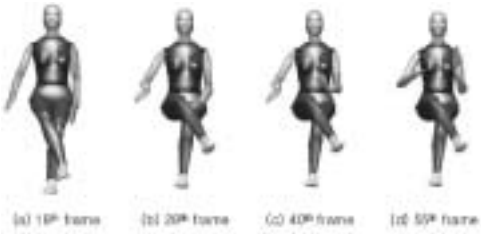


Fig. 13. Result for sit_r_talk.bap provided by MPEG-4

Fig. 14. Results for head-roll.bap (left column) and head-tilt.bap (right column) generated by the system

4. Conclusions

In this paper, we have proposed a BAP generation system for head movement based on the MPEG-4 SNHC standard, connecting vision-based human motion analysis with parameter-based 3-D model representation. Conventional synthetic character animation systems using contact devices have several defects. Although contact devices provide accuracy, they are expensive and not user friendly. There are various vision-based approaches trying to avoid those problems. However, they do not consider issues about data compression, transmission, and the international standard. Thus, they depend on specific S/W or H/W environments and do not provide compatibility. Therefore, we use a computer vision device as the input interface and analyze head motions to generate BAPs.

A contour-based approach is efficiently used to calculate contour areas, contour moments, and the orientation of head movement. Then, BAPs are generated explicitly by this information. Following the MPEG-4 SNHC standard, BDPs and BAPs are processed to animate the 3-D human model with OpenGL. Our BAP player is operated properly with BAPs provided by the standard and generated by the system. Moreover, the proposed system has an ability to utilize not only internally stored specific 3-D models, but also any VRML human model.

Although we have handled some problems of conventional methods, our system has some limitations at present. We have tracked partial head movements and the motion is mainly 2-D in spite of our system fully supports 3-D. We will expand our system for the whole body motion and make an effort to increase the accuracy of motion information.

Acknowledgement

This work was supported in part by K-JIST, in part by the Ministry of Information and Communication (MIC) through the Realistic Broadcasting Research Center at K-JIST, and in part by the Ministry of Education (MOE) through the Brain Korea 21 (BK21) project.

References

[1] C. Tolga, E. Petajan, and J. Ostermann, "Very low bitrate coding of virtual human animation in MPEG-4," Proceedings of ICME, pp. 1103 - 1106, July 2000.

[2] C.C. Huang and C.C. Lin, "Model-based human body motion analysis for MPEG–4 Video Encoder," Proceedings of ITCC, pp. 435 - 439, April 2001.

[3] Open Source Computer Vision Library Reference Manual, Intel Corporation, 2001.

[4] G.R. Bradski, "Computer vision face tracking for use in a perceptual user interface," Intel Technical Journal. Q2, 1998.