JOURNAL OF KOREA MULTIMEDIA SOCIETY, VOL. 10, NO. 6, JUNE 2007(pp. 716-725)

Stereo Vision Based 3-D Motion Tracking for Human Animation

Seung Il Han[†], Rae Won Kang[††], Sang Jun Lee[†††], Woo Suk Ju[††††], Joon Jae Lee[†††††]

ABSTRACT

In this paper we describe a motion tracking algorithm for 3D human animation using stereo vision system. This allows us to extract the motion data of the end effectors of human body by following the movement through segmentation process in HIS or RGB color model, and then blob analysis is used to detect robust shape. When two hands or two foots are crossed at any position and become disjointed, an adaptive algorithm is presented to recognize whether it is left or right one. And the real motion is the 3-D coordinate motion. A mono image data is a data of 2D coordinate. This data doesn't acquire distance from a camera. By stereo vision like human vision, we can acquire a data of 3D motion such as left, right motion from bottom and distance of objects from camera. This requests a depth value including x axis and y axis coordinate in mono image for transforming 3D coordinate. This depth value(z axis) is calculated by disparity of stereo vision by using only end-effectors of images. The position of the inner joints is calculated and 3D character can be visualized using inverse kinematics.

Keywords: 3D Motion Estimation, Target Tracking, Stereo Camera, 3-D Character Animation, Inverse Kinematics, Color Model

1. INTRODUCTION

Computer vision is concerned with the theory and technology for building artificial systems that obtain information from images. In vision systems an important issue is its ability to behave like human vision system. The system detects the motion of human being or location of an object and responds correspondingly. Therefore we can reconstruct motion and location of the object which can be used in real world. Currently, multi-media business (movie, animation, game etc.) are continuously growing, so demand for advanced object detection device is increasing.

In this paper, our objective is to produce virtual 3D model which has motion similar to the objects in real world. The existing devices for object detection are based on sonar motion and magnetic capture, etc. These devices require wearing of sensors on various parts of body which is not feasible as sensors are big and heavy to move freely and they are high in cost. In our proposed system there is no need of marker or sensor, and it is low in cost. When general target is tracked, the system analyzes the human or object location using image processing techniques. It calculates 3D coordinate, in order to present the 2D data in virtual space. So 2D data has to be changed by stereo vision. First, target is defined by image on each camera. The detection of target is based on segmentation technique using HSI or RGB color model. But due to the presence of noise in the

※ Corresponding Author : Joon Jae Lee, Address : (617-716) Dongseo University San 69-1 Jurye2-dong, Sasang-gu, Busan, Korea, TEL : +82-51-320-1724, FAX : +82-51-316-2786, E-mail : jjlee@dongseo.ac.kr
Receipt date : Oct. 31, 2006, Approval date : Jan. 29, 2007
[†] Graduate School of Design & IT, Dongseo University (E-mail : d99003159@dongseo.ac.kr)
[††] Hangreentec Corp. (E-mail : hantc010@hanmail.net)
[†††] Pivonine Corp. (E-mail : helmet97@nate.com)
[††††] Division of Digital Contents, Dongseo University (E-mail : helmet97@nate.com)
[†††††] Graduate School of Design & IT, Dongseo University

images, the object cannot be detected clearly hence we detect the object clearly using blob analysis.

Using color model system there are few problems of overlapping point, which occur when targets overlap due to the occurrence of similar color. We propose an algorithm which solves the problem of overlapping point using acceleration vector, location data, blob size data, and define boundary. First, the define boundary algorithm protects contacting target from overlapping problem. If targets overlapping occur in spite of that, we can solve the problems using previous acceleration vector, location data, blob size data techniques. Moreover, this system detects depth-value (including x-axis, y-axis) which reconstructs three-dimensional coordinate, which orderly compute the elements of virtual reality based on the object of real world using stereo image data. This system tracks object present in images in real-time and tests 3D-model which is made by motion data. If we want to visualize 3D-model in virtual reality, motion data has to be correct.

2. STEREO VISION SYSTEM

The proposed system is based on a stereo vision system to obtain and track the feature of human end-effectors, such as a head, hands, and feet. This system consists of one personal computer and two color cameras. (See Fig. 1.)

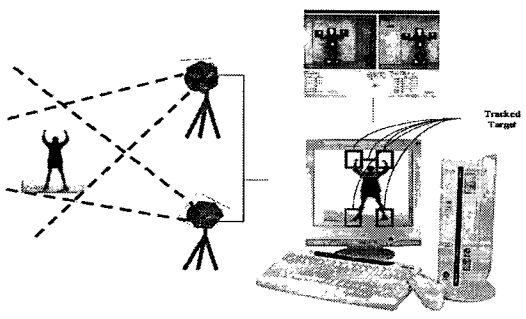


Fig. 1. System architecture.

The system is capable of tracking objects in the camera views in real time. First we use Zhang's camera calibration algorithm to calibrate the camera to remove distortion present in the cameras[1]. The graphical user interface of the system display the live color image from the camera on the computer screen. It then allows the user to select the target using the mouse. The selected target is then initialized, the system then will continuously track the target. The flow chart of this system is shown Fig 2. At the beginning of the algorithm, the camera initially determines the location which is calibrated using stereo geometry. The image is then captured from the stereo camera. After capturing the image, the segmentation algorithm extracts object to be tracked using HSI or RGB color model. The blob of tracking object is estimated after calculating centroid of segmented image. If target can be tracked, we estimate and compute new target value. Otherwise target is again estimated using hue values. New target value is made up using averaging three centroid of blob method.

After estimating and computing the new target value's, the matching is performed between the left and right output, if match is correct between the left and right object 3D output is shown, otherwise again new target is estimated. The inverse kinematics is then performed on 3D data to check whether all parts are connected or not, if all parts are connected then the 3D object estimated is correct otherwise again target is estimated using hue values. This system uses the epipolar geometry for stereo matching and has a distortion of camera, so we used Z.Zhang's calibration method in this paper[2].

3. IMAGE SEGMENTATION

Once we click the target on the image from the cameras, then we get the selected pixel value(R, G and B). And we compute the average color of

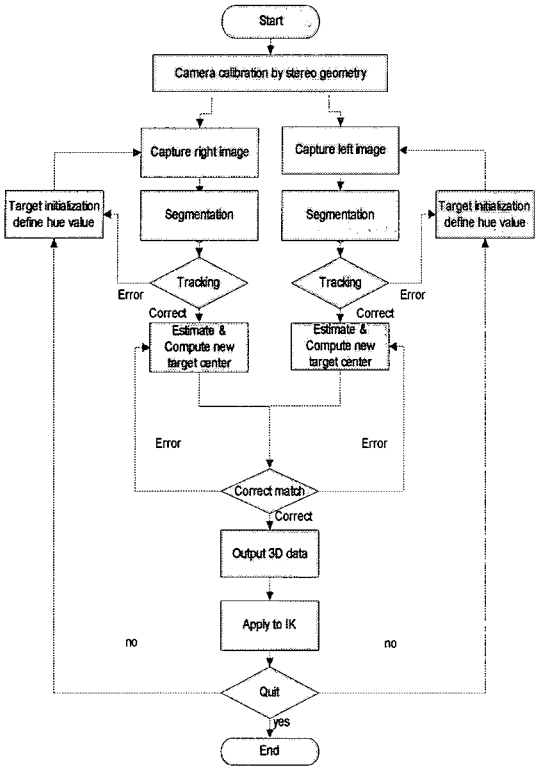


Fig. 2. Flow chart of the proposed system.

a small region around this point in the image. The color of this region is used for estimation of object tracking. The selected object is analyzed through HSI color model or RGB color model estimation. The HSI color model is used, when object cannot be found through RGB color model. If object is not sensitive to hue value, we would use RGB color model. (e.g. foot or center of body)

3.1 Segmentation of skin region by HSI color model

When we click the target on the image, we get RGB color. But we want HSI color model. Skin color has many noises. So, that can be not easily segmented thought RGB color model in the image. The HSI color model is available to acquire a more exact color data in computer vision. A color data is acquired more exactly like the human vision by HSI.

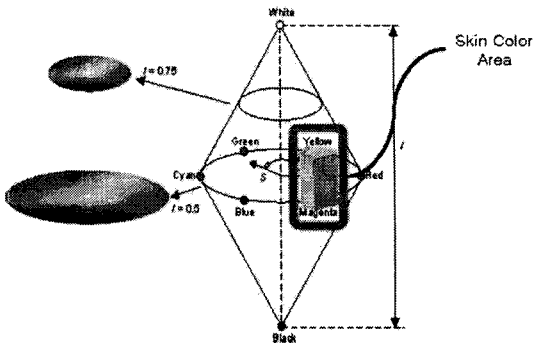


Fig. 3. HSI color model for skin color.

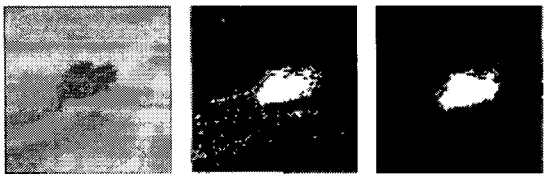
RGB color image input of CCD camera is changed to HSI color model and then hue, saturation and intensity values are calculated. HSI color model is used for detecting the object (e.g. hand, head etc). Removing intensity component and using only hue value reduce the effect of the illumination change. Also, the coordinate system is easy to be described in HSI color model. The color of target (human skin color) is the average value of sample set of pixel constituting the target.

3.2 Segmentation of others interest region by RGB color model

RGB color model formats an image data. So, we don't need to transform color model because feet and body center wear colorful shoes and cloth.

3.3 Blob analysis

We use mathematical morphology as a tool for extracting image component that are useful in the representation and description of region shape, such as boundaries, skeletons and the convex hull. Morphological image processing define two fundamental morphological operation, dilation and erosion in terms of the union (or intersection) of an image with a translated shape (structuring element). Fig. 4(a) is the original image, 4(b) is the image after applying threshold in hue value of HSI color model and 4(c) is the result of image after applying erosion and dilation.



(a) (b) (c)

Fig. 4. Image of hand detection.

4. TRACKING

Once we click the object present in the image, we can extract the color information of object and this color information is then used to track the objeet using segmentation. This color information determines target region to be tracked in the scene until it is reinitialized. For tracking the object, two consecutive frames are taken. Centroid of intersected region of two objects is calculated, centroid of difference between the second frame (object) with respect to first frame (object) is calculated and centroid of object in second frame is calculated then the average of centroids is calculated which is the approximated centroid of target. This result is in a very robust tracking system.

4.1 Target center using region

Since we are using HSI model, even if some noise is present we are still able to find the object. HSI model has three elements (hue, saturation, intensity). Intensity is changed by illumination and saturation is changed by Intensity. For robustness, this system should not consider saturation and intensity values which are sensitive elements. The hue value of a pixel in a color image is determined by the red, green and blue values present in the image buffer corresponding to the pixel. This color value will form a point in the two dimensional goniometer hue spaces. The hue of the target is then computed using sample set of pixels constituting the target. When the target moves and the illumination changes the hue of the target is likely to change. We use a computationally efficient hue matching function which allows us to compute whether a pixel hue matches the target hue within limits.

Once we have the hue value of the target and a color matching algorithm, we can find the pixel in given region of the image which matches with hue value of the target. We use the quantitative measure of color matching to find the centroid of these pixel positions. This gives us the most likely center of the target based on color only. If (i,j) are row and column coordinates of the pixel P_c (i, j), then for a given rectangular region the most likely target center based on color alone will be given by:

$$\text{Center}_p = \begin{bmatrix} X_p \\ Y_p \end{bmatrix} = \begin{bmatrix} \dfrac{\sum_{1}^{i*j} P(i,j,t) * i}{\sum_{1}^{i*j} P(i,j,t)} \\ \dfrac{\sum_{1}^{i*j} P(i,j,t) * j}{\sum_{1}^{i*j} P(i,j,t)} \end{bmatrix} \quad (1)$$

In order to track the object, we find position of the target by calculating average of the centroids given above. And then blob analysis is used to calculate the area of the object. And this process is recursively performed to track the object present in the frame. Fig. 5 show tracking of the object after calculating cetroid and then using blob analysis to calculate area[3,4].

4.2 Acceleration of motion vector

The speed of the motion affects the rapid processing time of the blob. The search range of system has the least range which is limited for fast

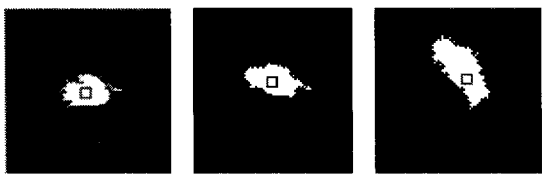


Fig. 5. Hand motion tracking.

processing. We know that next location of object computed by average of blob center isn't suitable for fast motion. If motion is fast, blob is located outside of search range. Therefore, acceleration of motion vector is efficient to find the target going out of its search range[3].

$$v_t = | C_t(i,j) - C_{t-1}(i,j) |$$
$$v_{t+1} = v_t + a(v_t - v_{t-1}) \quad (2)$$

Where C is blob center calculated by equation (1), V_t is distance of C, $\boldsymbol{a}$ is acceleration. When motion direction doesn't changed for a short while, but speed is decreased, acceleration needs to decrease. Decrease in speed can be determined, as blob center is near between pre-frame and current frame. Also, when motion vector is reversed, we remove inertia because the law of inertia disturbs tracking. If we don't remove inertia, searching area will move to a wrong direction. (Fig 7.)

4.3 Overlapping problem between Objects

There are 4 cases of object estimation by hue value. In case of (a), we estimate two hands and head. (b) is overlapped by two hands and head. (c) is overlapped by one hand and head, and the other hand. (d) is overlapped by two hands and head. We will solve overlap problem like (b), (c), (d) cases. When we start human tracking, the model is T-posed (a). In no overlap case, we know head and

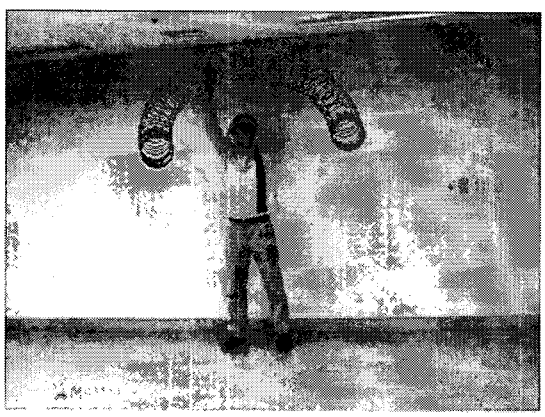


Fig. 6. Search of moving area.

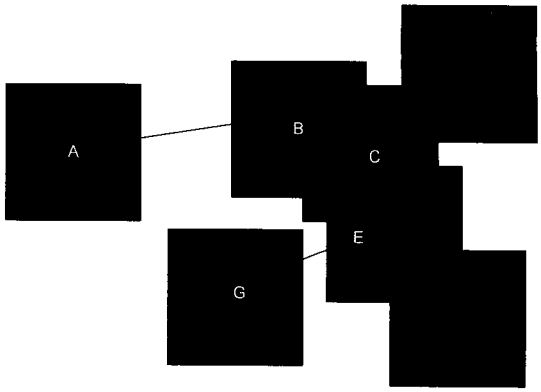


Fig. 7. Searching error by inertia

two hands, each blob centriod position and size. But when overlapping happen, some blobs size and center points will change. There exist only four cases after segmentation using hue values shown in Fig 8.

In Fig 10, when hands come over and separate from face, we calculate the position of the objects as given by above method. In Fig 10, as the face area is bigger so area of the hand gets neglected because it is small. Hence it is difficult to know if there are two objects. So, in our proposed method we take a small area near to the centroid of the objects into consideration to identify the different objects. But if the small area under consideration

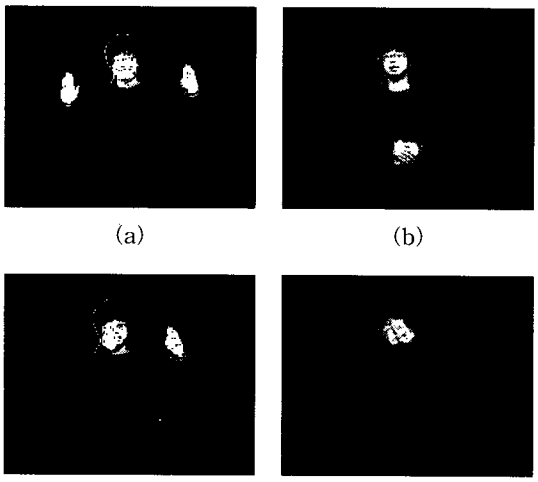


Fig. 8. Cases of overlapping between objects.

is also overlapped then acceleration vector principle is used to detect the object, the principle is discussed later.

In the position of merging and splitting of different end-effectors at one point, for e.g. if suppose three objects are overlapped then to detect the objects, initially we have to find three blob areas and then calculate the center position of each blob. Then three end-effectors can also be recognized by the method proposed.

4.4 Proposed method to remove the overlapping problem

In some cases, we can get various blob's in same target area. In these blob's, we have to find the real end-effectors present. In Fig. 11, when two hands are merged and then splitted, it gives two cases to consider. First case, when two hands overlap, instead of getting two blobs only one blob is detected, then we must find two positions of objects. So our purpose is to select 1/4 position and 3/4 position of blob size in x-axis and select 1/2 position in y-axis, Second case, when two hands merged and 2 blob is detected, we calculate the center position of each hand twice. In some case error will happen when blob size changes.

We propose an algorithm for overlap error. Blob has some weaknesses. In general cases two blobs are merged, centroid of blobs become one. We will prevent two centroids to become one. (Fig 9.)

Boundary elimination algorithm uses the centroid of previous frame as the centroid of the new frame. The location of centroid of the object from previous frame and the same object in current frame are almost same, with little difference. This is because the search box is moving along with the object. (Fig 10.)

During the blob analysis for object tracking, we redefine boundaries of the different objects overlapping as shown in Fig 12. It is to protect interference of each blob. If two centroids point cross even after defining the boundary, we will use acceleration vector to prevent overlapping(Fig 13.).

When target 1 and target 2 appear to have same coordinate, the next algorithm will be used for separating the overlap. At this stage, we would

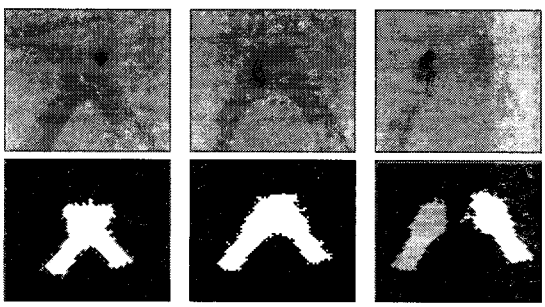


Fig. 9. Merge and split of two hands.

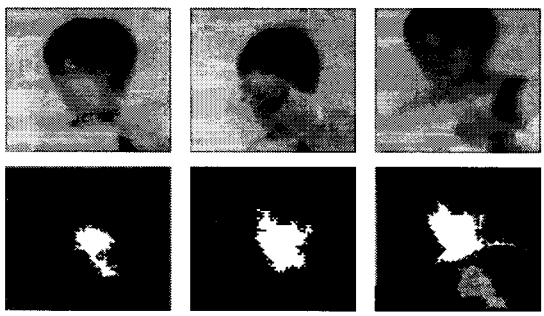
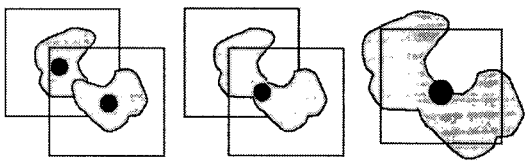


Fig. 10. Merge and split of hand and face



(a) (b) (c)

Fig. 11. Error in case that objects merge (a) Two objects, two centers (b) Two objects, one center (c) One object, one center.

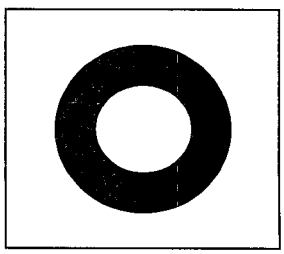
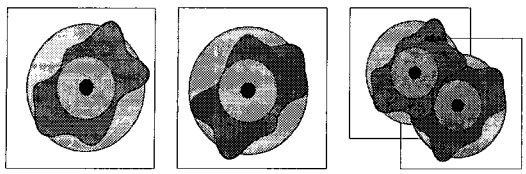


Fig. 12. Boundary elimination.



(a) (b) (c)

Fig. 13. Solving the overlapping problems by boundary elimination (a) Target 1 (b) Target 2 (c) Separating the overlap.

have acquires the average velocity of k previous frames. We calculate the difference between this average velocity and the velocity of 1 previous frame. If the result is less than or equals 0, the motion is set to be cross motion(b), otherwise the motion is set to be oppose motion(a). (Fig 14.)

$$\bar{A}_t = \left[\frac{\sum_{n=-k}^{0} v_{t+n}}{n} \right]_k \tag{3}$$

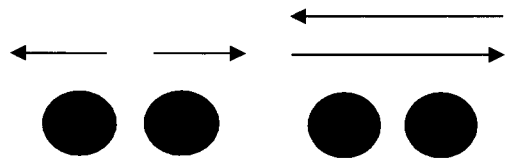
$$v_{t+1} = \bar{A}_t + a(\bar{A}_t - v_{t-1}) \tag{4}$$

where $\bar{A}_t$ is average of velocity and n is the number of frames contributed for getting average velocity[5].

5. STEREO MATCHING FOR 3D DATA

3D location can be calculated using stereo matching. The stereo camera uses 2 cameras (left and right camera).

Where *disparity* = $p_1 - p_2$ (p_1, p_2 defined using epipolar geometry), measure the difference in retinal position between the corresponding points in the two images.



(a) (b)

Fig. 14. Acceleration vector (a) Opposition (b) Cross.

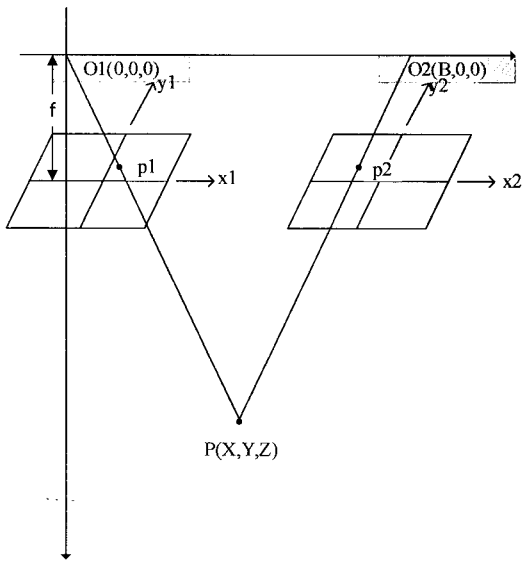


Fig. 15. Triangulation of stereo photogrammetry.

$$X = \frac{p_1}{(p_1 - p_2) * S} B$$

$$Y = \frac{y_1}{(p_1 - p_2) * S} B = \frac{y_2}{(p_1 - p_2) * S} B$$

$$Z = \frac{f}{(p_1 - p_2) * S} B \tag{5}$$

The X, Y, Z position of a point in the scene can be determined through triangulation between points in the images, Pl, P2 obtained by the left, right camera. Baseline is displacement between the cameras respectively. F is the focal length of the cameras. S is the Sensor size of CCD for applying real world. The object locations are a real coordinates with meter from the bottom. In this system, the motion data which is acquired by equation.(5) can use to the virtual reality motion[6-8].

6. EXPERIMENT RESULT

In this paper, the system is developed using: two FLEA digital cameras (Point Grey Corp. IEEE-1394 interface, image resolution 640*480, 30fps) and Pentium 4 CPU 2.8GHz computer. It captures

the motion of actor under common illumination. The Z.Zhang method is used for camera calibration. As shown in Fig. 19. ,by checking we select the positions of object for detecting and tracking. The image segmentation and the proposed algorithm is then applied, and centroid of object is calculated.

6.1 Processing time

We propose a small searching box in the image when motion tracks. The whole area didn't need to be searched in the image, because objects were defined by 6 small areas. The algorithm proposed is faster than the method being used in silhouette nalysis which is all area searching. (Fig. 16.)

6.2 Motion tracking

We solve the overlapping problem. In Fig. 17, we test three objects tracking(both hands and face).

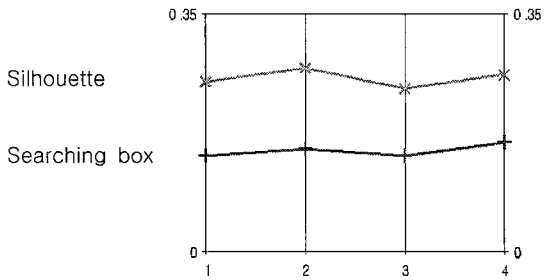


Fig. 16. Processing time(second).

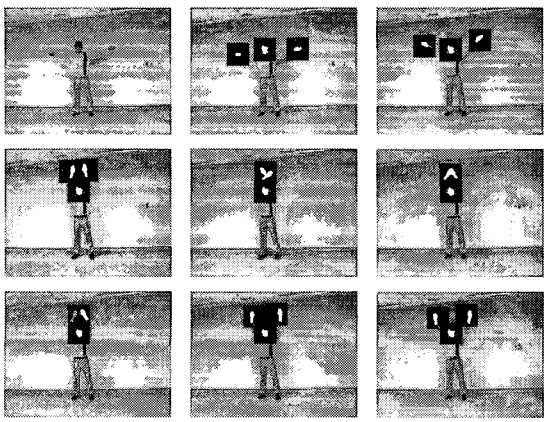


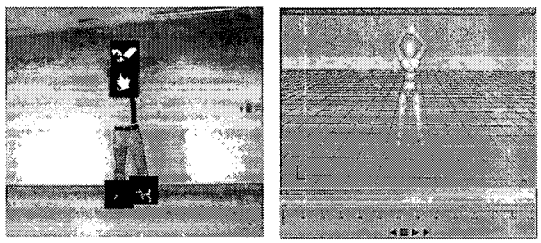
Fig. 17. Tracking result to image sequences.

Fig. 18 shows the character is in corrective position. The 3D character correctly poses, when objects which are both hand overlap.

It is possible to calculate mathematically an angle of joint which links the end-effector to a joint of body thought Inverse Kinematics. It is possible to track the actual motion of model. Model's head, hand and foot are tracked. These data are used to calculate a motion of virtual model. By using Inverse Kinematics when a motion is calculate, it is possible to move the model exactly like a real motion[9,10].

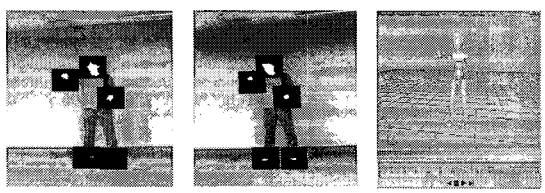
As shown in the Fig. 6. the end-effectors of human body can be tracked robustly. Fig. 19. (c) shows the 3D character produced after gathering information from left image and right image shown in Fig. 19. (a), (b). Fig. 19. (c) shows the 3D character which is rotated at an angle of 15 degrees, because this character has the 3D motion data though stereo matching.

Fig. 21. shows the T shape of the 3D character, in application program, produced after gathering information from left image and right image shown in Fig. 20 (a), (b) .



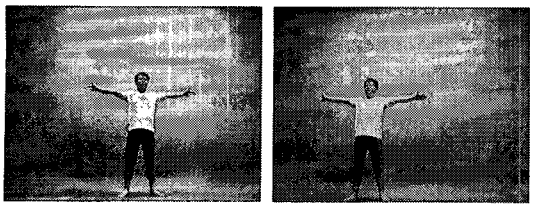
(a) (b)

Fig. 18. Applying arm motion (a) Tracking result (b) 3D plot.



(a) (b) (c)

Fig. 19. Animation result. (a)Right image (b)Left image, (c) Applying arm motion.



(a) (b)

Fig. 20. Applying 3D character after matching. (a) Right image. (b) Left image.

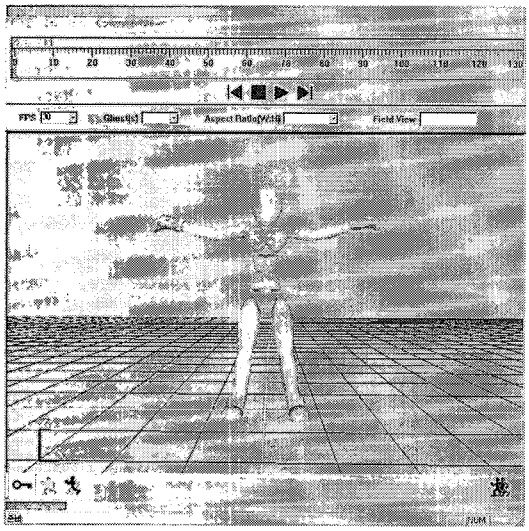


Fig. 21. Application program for shape of T.

7. CONCLUSIONS

In this paper, we introduced marker free motion capture system. The existing motion tracking using computer vision doesn't solve overlapping problem effectively in case of same color objects. So, they avoid scenes when objects overlap. But we propose ways which solve the problem of object overlapping and use the inverse kinematics algorithm. In this way, user can act freely during motion tracking. And the algorithm proposed is faster than the method which is used in silhouette analysis. The system can perform efficiently in real-time environment, because it is detecting and tracking the end- effectors of human body using small blob area. Also the system is insensitive to change of illumination.

ACKNOWLEDGEMENTS

This research was supported by the Program for the Training of Graduate Students in Regional Innovation which was conducted by the Ministry of Commerce Industry and Energy of the Korean Government.

REFERENCES

[1] Zhang, "A flexible new technique for camera calibration," *Microsoft research*, Vol. 4, pp. 5-9 December 2, 1998.

[2] C.Wren, A.Azarbayejani, T.Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. On PAMI*, Vol. 19, No. 7, pp. 780-785, July 1997.

[3] C.J.Park, S.E. Kim, and I.H. Lee, "Real-time marker-free motion capture system using blob feature analysis," *Proceedings of SPIE*, pp. 237-246, February 2005.

[4] George V.Paul Glenn, J. Beach, and Charles J.Cohen, "A realtime object tracking system using a color camera," *IEEE computer society*, March, 2001.

[5] N.I.Badler, M.Hollick, and J.Granieri, "Real-time control of a virtual human using minimal sensors," *Presence*, Vol. 2, pp. 82-86, 1993

[6] T.Horparasert, I.Haritaoglu, D.Harwood, L. Davis, C.Wren, and A.Pentland, "Real-time 3D motion capture," *Second workshop on perceptual interfaces*, pp. 2-3, 1998.

[7] R.H. Lee, S.E. Kim, C.J. Park, and J.H. Lee, "Real-time motion generation of virtual character using 3D position information of end-effector," *SCI2002*, July 2002.

[8] Emanuele Trucco and Alessandro Verri, *Introductory techniques for 3-D computer vision*, Prentice Hall, New Jersey 1998.

[9] C.Welman, "Inverse kinematics and geometric constraints for articulated figure manipulation," *Simon Fraser University*, Sept. 1989.

[10] Vera B. Anand, *Computer graphics and geometric modeling for engineers,* Sept. 1992.

Seung Il Han

2007 Dongseo Univ. (BS)
2007~Dongseo Univ. (MS)
Interesting area : image processing, Computer vision

Rae Won Kang

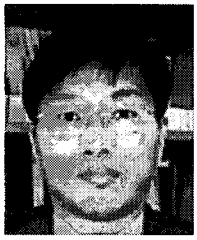
2005 Dongseo Univ. (BS)
2007 Dongseo Univ. (MS)
2007~
Interesting area : image processing, Computer vision

Sang Jun Lee

2004 Dongseo Univ (BS)
2006 Dongseo Univ. (MS)
2006~Pivonine software engineer
Interesting area : image processing, Computer vision

Woo Suk Ju

1998 Kyungnam Univ. (BS)
2005 Dongseo Univ. (MS)
2005~Dongseo Univ.
Interesting area : computer game, image processing, computer animation



Joon Jae Lee

1986 Kyungpook Nat'l. Univ. (BS)
1990 Kyunpook Nat'l. Univ. (MS)
1994 Kyunpook Nat'l. Univ. (Ph.D)
1998~1999 Georgia Institute of Technology. Visiting Professor
2000~2001 PARMI Corporation. Research Manager
1994~Dongseo Univ. Associate Professor
Interesting area : image processing, 3-D computer vision, and fingerprint recognition