بسم الله الرحمن الرحيم

Name	Mohamed Saleh	Loie Hesham
Student ID	1111113245	1091105774

Research Methodology Assignment-I
Tutorial Section TC209
Number of Pagers is 3 (including this page)
Name of the Journal is **Elsevier** *http://www.elsevier.com/journals*

Contents

Hands and face tracking for VR applications

Javier Varona , Jos M. Buades , Francisco J. Perales

April 11, 2013

Abstract

In this paper, the author presents a new method of 3D human limbs tracking system. This 3D tracking system is more efficient than some previews similar system. these systems come with many disadvantages,for instance,the user has to put on a lot of equipment which lead to a limitation in the movement. To over come this limitation, the author developed a virtual reality device that detect the skin-color pixels presented in the real-time image and with a color segmentation module, it will label the face and hands skin-color blobs, and this will create a 2D tracking results. By using the 2D frame and a set of hypothesis the system build the 3D positions of the limbs.

1 Problem Statement

To track and identify hands and face of the user and to obtain a real-time 3D position, and that without requiring any special suits or any type of markers.To do so, we should build a blob-based representation of hands and face by detecting skin-color pixels in the image follows by other steps before getting the final visualization results in **VRML** format and H-Anim avatar compliant.

2 Claimed Contributions

In order to solve the tracking problem, the author find that we should build a blob-based representation of each extreme limb (hands and face) condition by detecting the skin-color pixels in the image. Then, use the data association algorithm and a set of hypothesis built from the extreme limbs states at the previous frame. To calculate these hypothesis, we apply a simple prediction scheme to the detected extreme limbs. At last, by triangulating the extreme limbs 2D image position in cameras we get the 3D positions.

3 Directly-related work

Here is some of the important related works:

1. Perales FJ. Human motion analysis and synthesis using computer vision and graphics techniques. State of art and applications. Proceedings of the world multiconference on systemics, cybernetics and informatics (SCI2001), 2001.
2. Bretzner L, Laptev I, Lindeberg T. Hand gesture recogni- tion using multiscale colour features, hierarchical models and particle filtering. Proceedings of the fifth IEEE international conference on automatic face and gesture recognition (FGR.02), 2002.
3. Okay K, Satoy Y, Koikez H. Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems. Proceedings of the fifth IEEE international conference on automatic face and gesture recognition, 2002.

4 Methodology

4.1 Hands and face tracking algorithm

process goes thought many major steps, and those step are:

4.1.1 Skin-color segmentation module

Detect the skin-color of the actor using only one image for modeling the actor's skin-color. Next, convert the pixels from RGB-space to HSL-space to take color values (The hue and Saturation values) for each pixel. Using hue and saturation values for skin-color segmentation cause two main problems. First, that human skin hue values are near the red color, the value that they have is near 2π radians. We can avoid that by rotating the hue values to π radians. Second, the hue gets unstable when the saturation values are getting near the 0. Once the two previous steps are done, we have a sample set to learn the skin-color distribution and to obtain the best results we use a Gaussion model, where n is the number of samples, i is composed of the hue and saturation values, $\mathbf{x}_i = (h_i, s_i)$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}((x_i - \bar{x}), (x_i - \bar{x}))$$

Once we find $\bar{x}$ and Σ from the above formulas, we can calculate the probability that the new pixel is skin by using: (x is skin)

$$P(x) = \frac{1}{\sqrt{(2\pi)^2|\Sigma|}} x \exp(-\frac{1}{2}((x - \bar{x})|\Sigma^{-1}|(x - \bar{x})'))$$

We connected components algorithm to the probability to get the blob representation of the extreme limbs. Finally, we compute the attributes of each blob to represent the state of the actor's extreme limbs, and that we be used by the data association process.

4.1.2 Data association module

The data association module's main objective is to produce the labels that represent the extreme limbs. It detects any new coming extreme limbs in the scene , or any extreme limbs that have left the scene . Each limb is represented by a vector $Sl = (Pl, wl, l)$ Where (l) is the extreme limb's label $P = (Px, Py)$ the limb position in 2D $W = (W, H)$ size of limb in pixels is the limb's angel in the 2D plane How the association algorithm works ? For each frame in a certain time (t) , it associate/combine the skin-color pixels , then build a hypothesis of the extreme limb in time (t-1)

$$P(t) = P(t) + P(t-1)$$

$$P = P(t) + P(t-1)$$

The above equation means that an extreme limb have a constant speed on the image plane. Therefore at a time (t) we have a set of hypothesis
$H = h_l, l \leq 3$ Where $H_l = (\hat{p}_l, w_l, \theta_l)$
And assuming that at a time (t) , (m) blobs have been detected we will have the following set

$$B(b_1, , b_j, , b_M)$$

Each blob will be labeled as B_j and each B_j identify a set of connected skin-color pixels of one limb, in order to not mix up between each limb blobs in case two limbs crossed or occludes another limb ,That data association process must set a relationship between the hypothesis (h_l)and blobs (b_j) in time.To do this we set a an approximation to the distance from the $X = (x, y)$ image pixel to the hypothesis m and normalize the image pixel coordinates $t = x - p$, $n = R.t'$, Where $\mathbf{R} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$ Then we find the crossing point $C = (C_x, C_y)$ between the hypothesis and the normalized image pixel by calculating the angel between the normalized image pixel and the hypothesis So we get the following result :

$$C_x = w\cos\alpha$$

$$C_y = h\sin\alpha$$

To calculate the distance between an image pixel and a hypothesis we use the following equation

$$d(\mathbf{x}, \mathbf{h}) = ||\mathbf{n}|| - ||\mathbf{c}||$$

4.1.3 3D-point reconstruction

Estimating the 3D position for the extreme limbs using the standard geometry and the 2D position all the images. By now the 3D position of the limbs is computed, the next thing is to label the blobs. And before we reconstruct the 3D points its very important to adjust the camera stereo, and we do that by using the Zhengyou Zhang algorithm, which is cheap and easy to use.

4.2 Visualization using H-Anim

The 3D information that obtained from the camera stereo pair is needed to display the data in real time on the workbench. We use the **H-A**nim standard and collaborate it with standard **VRML** model so we can get a human virtual avatar. the object from that is to recover accurately the 3D postion and orientations of limbs.

5 Conclusion

In this paper, the author reviled a brand new 3D tracking system for human hands and face in real time. which works by going through the following procedures . First, the system analysis the image or frame of the user to set different labels and marks on the user's color pixels.After that, the detected skin pixels goes through a process called color segmentation to identify the hands and face pixels in real time ,now we have a 2D image of the limbs captured in each one of the two stereo pair cameras. The final step, is producing a 3D image of the limbs that will be done by combining both pictures and results coming from the stereo pair cameras and using standard geometry algorithms. As a result we have a 3D tracking system that works in real time .

6 Reference

1. HANIM 1.1 Compliant VRML97. http://ece.uwaterloo. ca/h-anim/index.html.
2. Raimi A. Fast connected components on images. http:// xenia.media.mit.edu/9rahimi/connected.
3. Bouget JY. Camera Calibration Toolbox. http://www. vision.caltech.edu/bouguetj.