Robust Real-time Human Hand Localization by Self-Organizing Color Segmentation

Ying Wu, Qiong Liu, Thomas S. Huang
Beckman Institute
University of Illinois at Urbana-Champaign
Urbana, IL 61801
{yingwu, q-liu2, huang}@ifp.uiuc.edu

Abstract

This paper describes a robust tracking algorithm used to localize human hand in video sequences. The localization system relies mainly on an automatic color-based segmentation scheme combined with the motion cue. An automatic self-organizing clustering algorithm is proposed to learn the color clusters unsupervisedly in the HSI space without specifying the number of clusters in advance. The schemes of growing, pruning and merging of 1-D self-organizing map (SOM) are facilitated to find an appropriate number of clusters in the forming stage of SOM. The training and segmentation in our approach is fast enough to make possible real-time applications. This segmentation scheme is capable of tracking multiple objects of different colors simultaneously. Motion cue is employed to focus the attention of the tracking algorithm. This approach is also applied to other tasks such as human face tracking and color indexing. Our localization system implemented on a SGI O2 R10000 workstation is reliable and efficient at 20-30Hz.

1 Introduction

Localization and tracking objects in video sequences are key issues in computer vision applications. In current virtual environment (VE) applications, keyboards, mice, wands and joysticks are the fundamental controlling and navigating devices. However, those devices are unnatural. Human body are being considered as a natural "device" in human computer interaction (HCI), which motivates the research of tracking, analyzing and recognizing human body motion[7, 12, 13]. Although the goal is to understand the human body movements, the first step to achieve this goal is to reliably localize and track human faces and hands in image sequences.

The difficulties in visual tracking come from clutter background and unknown lighting conditions. When it needs to track multiple objects simultaneously, the problem becomes even more challenging. The robustness, accuracy and speed are important to evaluate a tracking algorithm. Another difficult problem in visual tracking is matching objects in different image frames.

Different image information sources supply different cues to tracking algorithms. Edge-based approaches match the edges of objects in images and region-based approaches use image templates. In the small motion assumption, which assumes there is little difference between two images, these approaches can achieve accurate results. However, when this assumption does not hold, which is very often in practical applications, these algorithms will be lost and have to depend on some remedies to resume tracking. At the same time, edge-based and region-based tracking methods generally need more computational resources, which makes real-time application systems hard to realize by using these techniques.

On the other hand, blob-based tracking algorithms do not use local image information such as edge and region, but rely on color, motion and rough shape to segment objects from the background. They are computational efficient and robust. Since human hand movement always presents fairly large motion, edge-based or region-based tracking algorithms may not suitable for real-time hand tracking applications.

Although a tracking system can be benefit from high-level image processing activities such as recognition and understanding, we believe the process of localizing objects in our human vision is mainly a low-level activity.

Our localization algorithm is based on low-level image segmentation, which is necessary in tracking bootstrapping and in the cases where the small motion assumption does not hold. Color is a strong cue of segmentation. Some successful tracking systems are built

on color segmentation [4, 3, 11, 6]. However, there are still some challenging problems related to tracking by color-based segmentation [4], such as complicated background with color distraction and changing lighting conditions.

In this paper, we propose a robust real-time localization algorithm which applies to human hand tracking applications. Color cue and motion cue are integrated in a hierarchical clustering algorithm. Color clusters in the HSI color space are learned through an automatic self-organizing clustering algorithm without specifying the number of clusters in advance. A scheme to adapt lighting conditions is embedded in our algorithm. Motion cue is employed to focus the attention of the tracking algorithm. Our algorithm can also be applied to other tracking tasks. All of our experiments described here are real-time at 20 to 30 frames per second running on a SGI O2 R10000 workstation without any special hardware.

2 Localization System

Our localization system takes a hierarchical segmentation approach. Although our tracking system can work solely by color segmentation, motion segmentation and region-growing method are used to make the system more robust and accurate without introducing too much computation cost. Figure 1 shows the overview of the localization system.

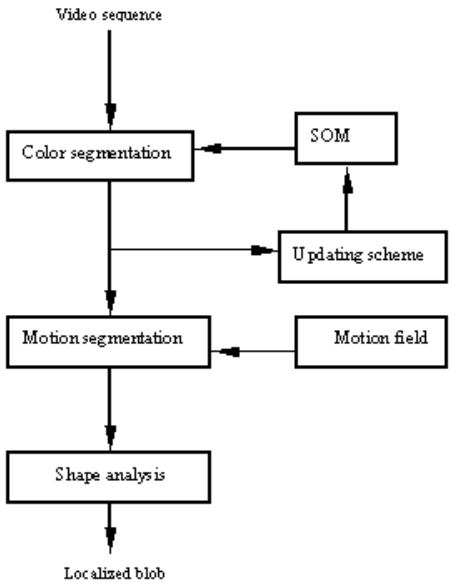

Figure 1: Localization system framework

The first frame taken by a camera is used to train the SOM by our self- organizing clustering algorithm, which will be described in next section. In this initialization stage, the colors in the scene are learned. In our experiments, the training stage is fast (less than 1 sec) with a 640x480 color image. The input of the SOM is the HSI value of a pixel, and the output is the index of the winning node of SOM by competition. Generally, it takes fewer than 6 nodes to segment indoor working environments.

The trained SOM is used to segment each image frame to find different color regions. This stage can be done on a lower resolution version of the image, which makes the segmentation faster. Morphology operators are also employed to filter the noise. The result of color segmentation is good enough to separate different color regions. In order to adapt to the lighting condition, several random samples taken from the segmented color regions are used to update the weights of SOM.

Since there may be many different colors in the working space, how to determine what to track is a problem. One possible solution is to specify an interested color region such as human hand or face. Second possible solution is to learn the skin color distribution from a large number of training samples. Another possible solution is to use some rules to automatically find an interested color from motion intension. If we detect a motion region by examine the image difference or optic fields, the color of that region is taken to be the interested color.

There are some cases in which several objects have nearly the same color. For instance, tracking two face and two hands is needed in recognizing sign languages. When the color segmentation algorithm separates them from the background, there are some ways to locate each region. One method is to use the same scheme of our self-organizing clustering to find the centroid of each isolated blob. Another way is to use a region-growing technique to label each blob.

3 Color Segmentation Algorithm

Color has always been considered as a strong tool for image segmentation [2]. In a static scene, color difference is one of the easiest globe cues to tell different objects apart. For instance, in the familiar commercial where the one bright red umbrella is surrounded by a virtually endless sea of black umbrellas, the red one is easy to be found because of its difference in color. Color-based segmentation is nothing new, and its roots are almost as old as color video itself. Video engineers have used chrome keying, the first color-based segmentation, practically since the advent of color video cameras. Even today, colored markers are

frequently used to facilitate locating objects in a cluttered video scene.

Because it is not computational expensive and color can give more information than a luminance only image or an edge image, color-based segmentation is more attractive than edge-based and luminance histograming techniques.

Histogram-like segmentation approaches like Color Predicate (CP) work well when appropriately thresholding the histogram. Although one threshold can be easily found in two-peak histogram which corresponds to simple background, it is still hard to handle complicated background because finding good thresholds can be very complicated. Our color segmentation scheme is based on 1-D self-organizing map to tessellate the HSI color space. The self-organizing map can learn the density distribution of the HSI color space, so that it is capable to identify multiple peaks in the distribution. We take the advantage of this property of SOM to learn the color distribution of the scene.

The self-organizing map (SOM) [5, 9]is mainly used for visualizing and interpreting large high-dimensional data set by mapping them to a low-dimensional space based on a competition learning scheme. SOM consists of an input layer and an output layer. The number of nodes in input layer is the same as the dimension of the input vector while the structure of the output layer can be 1-D or 2-D connected nodes that are connected to each input node with some weights. Figure 2 shows the structure of 1-D SOM. Through competition, the index of the wining node is taken as the output of SOM. The weights of the winning node and its neighborhood nodes are adjusted by Hebbian learning rule. It is highly related to vector quantization (VQ). One good characteristics of SOM is its partial data density preservation if properly trained [9].

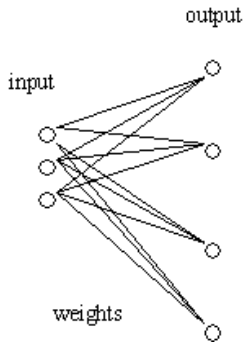

Figure 2: 1-D SOM structure

One of the problems of many clustering algorithm is that the number of clusters should be specified in advance. The more the clusters, the higher the resolution. Different numbers of clusters lead to different results of clustering. The success of the clustering algorithm depends on the specified number of clusters. If fewer nodes are used, data of lower density will be dominated by the patterns of higher density. On the other hand, if more nodes are used, the ordered mapping is hard to be obtained.

We propose an unsupervised self-organizing learning algorithm to tessellate the color space by a given training data set. Our algorithm can automatically find the appropriate number of clusters. The schemes of growing, pruning and merging are embedded in our algorithm.

Growing Scheme: Our algorithm is a competitive learning scheme which has to deal with how to find the competition winner. In the basic SOM algorithm, the output of a node is the distance between the input vector and the weight vector of the node. The distance measurement can be defined as:

$$\mathcal{D}(\mathbf{x} - \mathbf{w_i}) = ||\mathbf{x} - \mathbf{w_i}|| \tag{1}$$

where $\mathcal{D}$ is a distance measurement between the input vector $\mathbf{x}$ and the weight vector $\mathbf{w_i}$ of node i of SOM. The measurement here is Euclidean distance, however, other distance measurement can also be used.

The node with smallest output is taken as winner c.

$$c = arg \ \min_i \mathcal{D}(\mathbf{x} - \mathbf{w_i}) \tag{2}$$

In some cases, when the outputs of all nodes are nearly the same, determining the winner by finding the smallest output is not suitable. In this situation, the input vector is too far from every weight vector or in the center of the convex hull of the weight vectors. If current input is included in any of the clusters, the weight vector of that cluster will be misplaced unnecessarily by adjusting the weight. Refer to Figure 3. So, it is not a robust way to make the smallest one as the winner. In this situation, a new cluster is generated by taking the input as the weight vector of the newly created node. By comparing the mean value and the median value of the outputs of all nodes, we make a rule to detect this situation. So, the competition can be described as:

$$y_i = \mathcal{D}(\mathbf{x} - \mathbf{w_i}) \ \forall i \tag{3}$$

where y_i is the output of the ith node with weight vector w_i. The competition winner can be found by:

$$c = \begin{cases} arg \ min_i \mathcal{D}(\mathbf{x} - \mathbf{w_i}), \ if mean(y) \approx median(y); \\ NULL, \ otherwise. \end{cases} \tag{4}$$

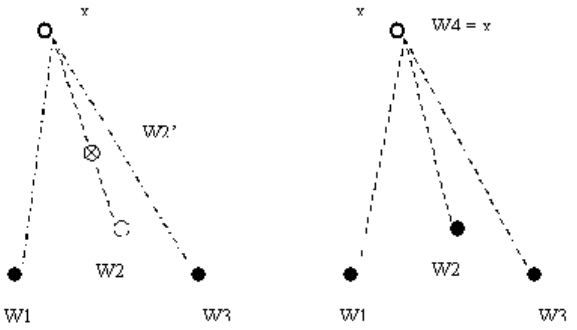

Figure 3: Growing scheme of SOM. w_i is the weight vector, and x is an input vector. (left) when the input vector is too far from every weight vector so that the output of all nodes are nearly the same, if current input x is included in any of the clusters, say w_2, the weight vector of that cluster will be misplaced unnecessarily. (right) In this situation, a new node is created and $w_4 = x$.

Pruning Scheme: In the training process, when a node is rarely to be a winner, it means that this cluster has very low density or can be taken as noise. So, this kind of nodes can be pruned.

Merging Scheme: In the training process, the distance between two weight vectors of each two nodes are calculated. If two weight vectors are near enough, we can merge these two nodes by assigning the average of the two weights to the new node.

Algorithm: The algorithm is summarized as below:

- Set the number of nodes to 2, according to two clusters, and randomly initialize the weights $\mathbf{w_i} = \mathbf{w_i}(\mathbf{0}), fori = 1, 2$, where $w_i(t)$ represents the weight vector of the ith node at the tth iteration.
- Draw an input x from the training sample set randomly to the SOM.
- Find the winner among the nodes through finding the maximum output of each node using equation 4.
- If(winner!=NULL), adjust the weights of the winner node c and its two neighborhood node $c-1$ and $c+1$.

$$\mathbf{w}_c(t+1) = \mathbf{w}_c(t) + \eta(t)(\mathbf{x} - \mathbf{w}_c(t))$$
$$\mathbf{w}_{c-1}(t+1) = \mathbf{w}_{c-1}(t) + \eta(t)\ \alpha(t)(\mathbf{x} - \mathbf{w}_{c-1}(t))$$
$$\mathbf{w}_{c+1}(t+1) = \mathbf{w}_{c+1}(t) + \eta(t)\ \alpha(t)(\mathbf{x} - \mathbf{w}_{c+1}(t))$$

  where $\eta(t)$ is the step size of learning, and $\alpha(t)$ is a neighborhood function.

- If there is no winner, grow a new node n according to the growing scheme. $\mathbf{w}_n(t+1) = \mathbf{x}$
- If a node is rarely win, delete it according to pruning scheme.
- Calculate the distance between each two nodes and perform merging scheme.

In our segmentation algorithm, training data set is collected from one color image, and each data vector is weighted HSI value, i.e. $\mathbf{x} = \{\ \alpha H,\ \beta S,\ \gamma I\}$, where we set $\alpha = \beta = 1$ and $\gamma = 0.1$. Pixels with large and small intensities are not included in the training data set, because hue and saturation become unstable in this range. Once trained, the 1-D self-organizing map is used to label each pixel by its HSI value. The pixel label is the index of the node in the self- organizing map.

4 Performance and Experiments

Our color segmentation algorithm has been tested with a large variety of pictures. And our localization system which integrates this color segmentation algorithm has run under a wide range of operating conditions. Experiments show that our color segmentation algorithm is fast, automatic and accurate, and the tracking system is robust, real-time and reliable. This color segmentation algorithm can also be applied to other segmentation tasks.

4.1 Performance of Color Segmentation

The only parameter we should specify is the maximum number of clusters. If the scene is simple, we set the maximum number of clusters to 2 or 3. If the scene is complex, we set it to 10 or more. In between, we use 6.

Figure 4 show some segmentation results. Left column are the source color images, middle column are the segmented images, and right column are the separated color regions. The color of a segmented color regions are the average color of that region. Each pixel in the source images is assigned a label by our color segmentation algorithm, and this label is used as a mask to separate the corresponding color region. Our segmentation algorithm works well from those experiments. When the background has less color distraction, this algorithm finds the exact color regions. Since texture is not used in the segmentation, the segmentation results will be noisy when there is color distraction in the background. Hand and face images are taken from a cheap camera in the indoor environment in our labs. Our algorithm can also successfully segment hand region and face region.

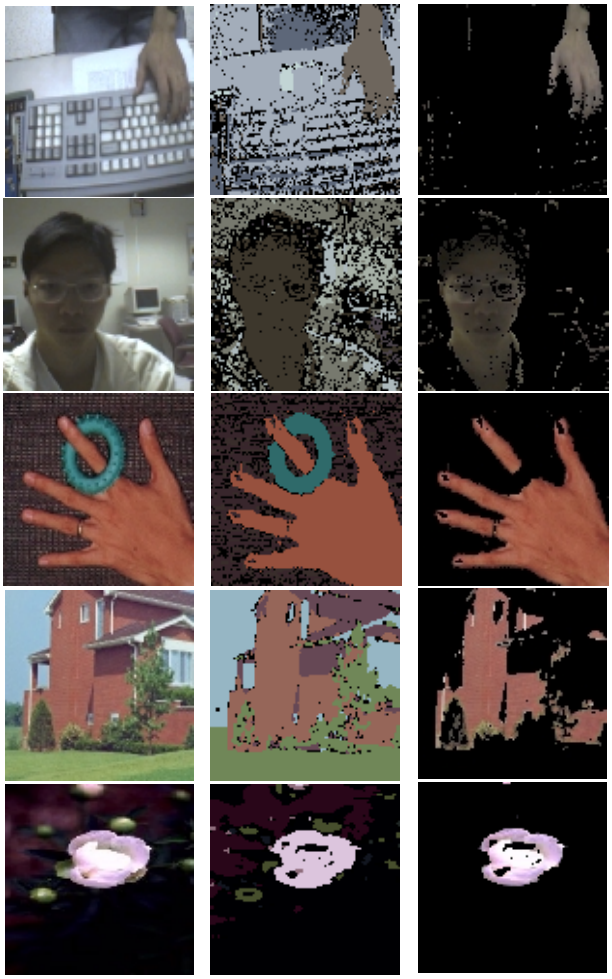

Figure 4: Color segmentation results. Left column are source color images, middle column are segmented images and right column are interested color regions.

4.2 Performance of Tracking

A typical application scenario of hand tracking is display controlling or 3-D mouse in desktop. A camera mounted at the top of the desktop computer looks below at the keyboard area to give an image sequence of moving hand. Another typical application is to track human face. Our tracking algorithm is able to localize multiple objects, which is useful to track moving human.

Since our tracking algorithm is essentially a global segmentation scheme, it does not rely on the tracking result of previous frames. Even if the tracker may get lost in some frames for some reasons, it can recover by itself without interfering the subjects. In this sense, the tracking algorithm is very robust. If the lighting condition changes dramatically, the color segmentation algorithm may fail. However, since the hue and the saturation are given more weight than intensity and the weight vectors in the self-organizing map are being updated with time, our tracking algorithm is insensitive to the change of lighting conditions by shading a light. Insufficient lighting and very dark background may bring some trouble to the color segmentation algorithm, since the hue and the saturation become unstable. The localization results of one experiment is given in Figure 5. In this experiment, a hand is moving around with interfering of a moving book. The book is also shading the light so that the color of skin is changing. The blue boxes are the bounding boxes of the interested color region.

Figure 5: Results of localization with 18 frames taken from image sequences. A moving hand with interfering of a book is localized. The blue boxes are the bounding box of the interested color region.

Our localization system is very robust and efficient from this experiment in which the background of the scene is complicated. Since a book is interfering the hand by shading the light, our localization system can still find a correct bounding box. Sometimes, due to

the sudden change of lighting condition, the tracker may be lost. However, it can quickly recover. Different skin tones do not affect our system. The first image with the interested color region is used to train the SOM so that it may work with nearly any users, which has been tested in many other experiments.

5 Conclusion and Future work

Localization of the interested objects in video sequences is essential to many computer vision applications. Complicated background, unknown lighting condition and multiple moving objects make the tracking task challenging. Computer vision techniques supply good ways to human computer interaction by understanding the movement of human body, which requires a robust and accurate way to track the human body such as hand and face. This paper present a robust tracking system based on the self-organizing color segmentation. A 1-D SOM is used to clustering the HSI color space automatically. Images are segmented by this 1-D SOM through the competition of each node. Each pixel of the image is labeled by the index of the winner node. Since our color segmentation is globe, it does not rely on some local image information such as edges so that there is no need to extract features and match the features in segmentation stage, which makes the algorithm cost efficient and robust. It seems that this scheme is more consistent with the principle of our human vision. Our localization system is mainly based on this color segmentation scheme. Experiments show that our localization system is capable of reliably tracking multiple objects in real time on a one-processor desktop SGI O2 workstation.

Since the process of competition among all nodes is essentially parallel, the tracking system can be made much faster by parallel implementation of the competition process. Currently, our localization algorithm can find a bounding box of the interested objects. Shape analysis of localized objects will be extended to estimate the 3D motion of the objects.

Acknowledgment

This work was supported in part by National Science Foundation Grants CDA-96-24396 and IRI-96-34618.

References

[1] Yusuf Azoz, Lalitha Devi, Rajeev Sharma, "Tracking Hand Dynamics in Unconstrained Enviornments", *International Workshop on Face and Gesture Recognition*, pp.274-279, 1998

[2] Dorin Comaniciu, Peter Meer, "Robust Analysis of Feature Spaces: Color Image Segmentation" *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico*, June 1997, 750-755.

[3] Kazuyuki Imagawa, Shan Lu, Seiji Igi, "Color-Based Hands Tracking System for Sign Language Recognition", *International Workshop on Face and Gesture Recognition*, pp.462-467, 1998.

[4] Rick Kjeldsen, John Kender, "Finding Skin in Color Images" *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pp.312-317, 1996

[5] T.Kohonen, "Self-Organized Formation of Topologically Correct Feature Maps", *Biological Cybernetics*, 43:59-69, 1982.

[6] V.Pavlovic, G.Berry, and T. S. Huang, "Fusion of audio/visual information for human-computer interaction," *Proc. Workshop on Perceptual User Interfaces (PUI), Banff, Alberta, CA, Oct.* 1997, pp. 69-71.

[7] Vladimir I. Pavlovic, R.Sharma, T.S.Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review," *IEEE PAMI*, Vol.19, No.7, July, pp.677-695, 1997.

[8] C.Rasmussen, G.Hager, "Joint Probabilistic Techniques for Tracking Objects Using Multiple Part Objects", *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, 1998.

[9] H.Ritter and K.Schulten, "Convergence Properties of Kohonen's Topology Preserving Maps: Fluctuations, Stability, and Dimension Selection", *Bioological Cybernetics*, 60(1):59-71, 1988.

[10] M.J.Swain and D.H.Ballard, "Color Indexing", *Int. J. Computer Vision*, Vol.7, No.1, pp.11-32, 1991.

[11] C. Wren, A. Azarbayejani, T. Darrel, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body", *In Photonics East, SPIE Proceedings* vol.2615, Bellingham, WA, 1995.

[12] Ying Wu, Thomas S. Huang, "Capturing Articulated Human Hand Motion: A Divide-and-Conquer Approach", *IEEE Int'l Conf. on Computer Vision*, Corfu, Greece, 1999

[13] Ying Wu, Thomas S. Huang, "Human Hand Modeling, Analysis and Animation in the Context of HCI" *IEEE Int'l Conf. on Image Processing*, Kobe, Japan, 1999