In Proc. Asian Conf. on Computer Vision, Taiwan, 2000

An Adaptive Self-Organizing Color Segmentation Algorithm with Application to Robust Real-time Human Hand Localization

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

ABSTRACT

This paper describes an adaptive self-organizing color segmentation algorithm and a transductive learning algorithm used to localize human hand in video sequences. The color distribution at each time frame is approximated by the proposed 1-D self-organizing map (SOM), in which schemes of growing, pruning and merging are facilitated to find an appropriate number of color cluster automatically. Due to the dynamic backgrounds and changing lighting conditions, the distribution of color over time may not be stationary. An algorithm of SOM transduction is proposed to learn the non-stationary color distribution in HSI color space by combining supervised and unsupervised learning paradigms. Color cue and motion cue are integrated in the localization system, in which motion cue is employed to focus the attention of the system. 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.

Keywords: color segmentation, hand localization, transductive learning, self-organizing map

1. INTRODUCTION

In current virtual environment (VE) applications, keyboards, mice, wands and joysticks are the most fundamental input devices. However, those devices are either inconvenient or unnatural in the sense of 3D or high DOF inputs. Human body, such as hand, is being considered as a natural input "device" in human computer interaction (HCI), which motivates the research of tracking, analyzing and recognizing human body movements [7, 12, 13]. Although the goal of natural user interfaces is to recognize and understand the movements of human body, the first step to achieve this goal is to reliably localize and track human body parts, such as face and hand. Magnetic sensors have been used to supply some motion information directly, however, many magnetic sensors are plagued by magnetic interferences[13]. An alternative is vision-based interface (VBI). An example is vision-based gesture interface, in which some commanding inputs are represented by a set of hand gestures such as pointing, rotating, starting, stopping, etc. In order to recognize these hand gesture commands, the system should localize and track the motion of human hand. However, the articulation and non-rigidity of hand make this task non-trivial.

In most vision-based applications, localizing and tracking objects in video sequences are two of the key issues, since they supply inputs to the recognition part of application systems. Generally, localization is to estimate a bounding box for an object in image sequences, while tracking includes 2D tracking which estimates 2D motion parameters, 3D tracking which gives the position and orientation of the object in 3D space, and high DOF tracking which tracks the deformation of the object. In this sense, localization, which estimates position and size of the object in 2D space, extracts the most fundamental information of the object.

The difficulties in visual tracking come from complex backgrounds, unknown lighting conditions and deformation of the object. When it needs to track multiple objects simultaneously, the problem is made even more challenging. The robustness, accuracy and speed are important to evaluate a tracking algorithm.

Different image features of the object supply different cues in tracking algorithms. Edge-based approaches match edges of the object in different images, and region-based approaches use image templates. Under small motion assumption that assumes there is little difference between two consecutive frames, these approaches can achieve accurate results. However, when this assumption does not hold, which is very often in practice, these algorithms will be lost, and the recovery has to depend on some remedies. 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.

An alternative is blob-based tracking, which does not use local image information such as edge and region, but employs color, motion and rough shape to segment objects from the background. Although human hand is articulated, it is more uniform in the sense of color, which makes this approach computational efficient and robust.

Skin color is a strong cue in human tracking. Segmentation is necessary in tracking bootstrapping and in the cases when small motion assumption does not hold. A naive approach is to collect some skin color samples from one user, and skin color regions are expected to be separated from the background by thresholding the distances in color space. However, there is a large variation of skin color for different people. One of the solutions is to make a statistical model for skin color, and the model is tuned by collecting a very large training data set [4]. However, color may change with lighting conditions, which may mess up the skin color model in some cases. Another problem is that collecting such a large labeled training data set is not trivial.

Some successful tracking systems have been built based on color segmentation [5, 3, 11, 14]. However, there are some challenging problems related to tracking by color-based segmentation [5], such as cluttered background with color distracters and changing lighting conditions.

In this paper, we propose an adaptive color segmentation algorithm and a robust real-time localization system, which apply to human hand tracking and motion capturing applications. Different from the methods of constructing a unique skin color model, our proposed approach tries to adapt the non-stationary color distributions by transducing learned color models through image sequences. The color distribution at each time frame is approximated by our proposed 1-D self-organizing map (SOM), by which color clusters in the HSI color space are learned through an automatic self-organizing clustering algorithm without specifying the number of clusters in advance. In order to capture the non-stationary color distribution, the 1-D SOM is transduced by combing supervised and unsupervised learning paradigms. At the same time, motion cue is employed to focus the attention of the localization system. Our algorithm can also be applied to other tracking tasks.

The color segmentation algorithm based on self-organizing map and the transduction of SOM are discussed in section 2 and section 3 respectively. Our proposed hand localization system is presented in section 4. Some experimental results are shown in section 5 and the paper is concluded in section 6.

2. AUTOMATIC COLOR SEGMENTATION

Color has always been considered as a strong tool for image segmentation [2]. In a static scene, color difference is one of the easiest global cues to tell apart different objects. Color-based segmentation is nothing new, and its roots are almost as old as color video itself. Even today, colored markers are frequently used to facilitate locating objects in a cluttered video scene.

Because color is computational inexpensive, and it can give more information than a luminance-only image or an edge-segmented image, color-based segmentation is more attractive than edge-based and luminance histograming techniques. Histogram-like segmentation approaches such as Color Predicate (CP) [5] work well when appropriately thresholding the histogram. Although one threshold can be easily found in two-peak histogram that corresponds to simple background, it is still hard to handle cluttered background because finding good thresholds can be very complicated. Another approach is to make parametric color models by Gaussian distribution or Gaussian Mixtures [8, 4]. The problem is that there is not enough prior knowledge to determine the number of components of the distribution in advance.

Our color segmentation scheme is to approximate the color distribution of an image in the HSI color space by 1-D self-organizing map (SOM), in which each output neuron (or node) in SOM corresponds to a color cluster. Self-organizing map (SOM) [6] is mainly used for visualizing and interpreting large high-dimensional data set by mapping them to low-dimensional space based on a competitive learning scheme. SOM consists of an input layer and an output layer. Figure 1 shows the structure of 1-D SOM. 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. Through competition, the index of the wining node is taken as the output of SOM. Hebbian learning rule adjusts the weights of the winning node and its neighborhood nodes. SOM is highly related to vector quantization (VQ) and k-mean clustering. One good characteristics of SOM is its partial data density preservation if properly trained.

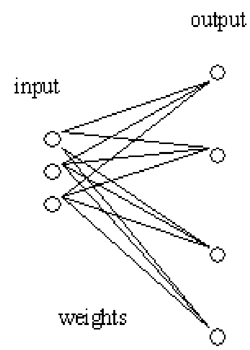

Figure 1. 1-D SOM structure

One of the problems of many clustering algorithms is that the number of clusters should be specified in advance. The success of the clustering algorithm depends on the specified number of clusters. It is the same case in the basic SOM algorithm. The more output neurons, the higher the resolution, since output neurons corresponds to clusters. Different number of cluster leads to different results of tessellation of the pattern space. If less neurons 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.

One possible approach to this problem is cross-validation. Although the structure of the SOM, such as the number of output neurons, is fixed each time, a good structure can be determined after validating several different structures. However, this approach does not offer flexibility to find an appropriate structure of SOM, and it is not fast. An alternative is to embed some heuristics to dynamically change the structure of SOM in training. Our algorithm can automatically find an appropriate number of clusters by the schemes of growing, pruning and merging.

Growing Scheme: Our algorithm is also a competitive learning scheme which deals with the problem of how to find the competition winner. In the 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:

$$\mathcal{D}(\mathbf{x} - \mathbf{w}_i) = ||\mathbf{x} - \mathbf{w}_i|| \quad (1)$$

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.

In standard SOM, the node with the smallest output is taken as the winner c.

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

In some cases, however, when the outputs of all nodes are nearly the same, determining winner by finding the smallest output is not suitable. In this situation, the input vector may be 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. So, it is not a robust way to make the smallest one as the winner. In this situation, a new node could be generated by taking the input as the weight vector of the new node, which is explained in figure 2.

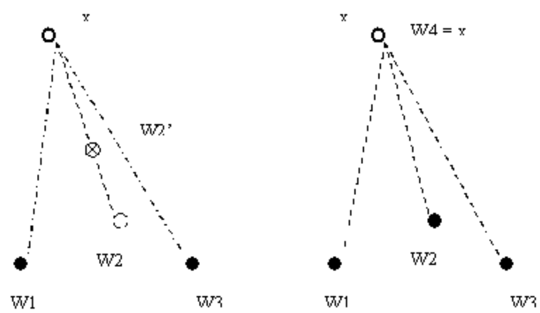

Figure 2. Growing scheme of SOM. $\mathbf{w}_i$ is the weight vector, and $\mathbf{x}$ is an input vector. (left) when the input vector is too far from every weight vector so that the output of all nodes are nearly the same, if current input $\mathbf{x}$ is included in any of the clusters, say $\mathbf{w}_2$, the weight vector of that cluster will be misplaced unnecessarily. (right) In this situation, a new node is created and $\mathbf{w}_4 = x$.

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 $\mathbf{w}_i$. The competition winner can be found by:

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

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. In practice, a threshold is set to determine such nodes.

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:

- Initially set the number of nodes N to 2, which is according to two clusters, and randomly initialize the weights $\mathbf{w}_i = \mathbf{w}_i(0), i = \{1, 2\}$, where $\mathbf{w}_i(k)$ represents the weight vector of the ith node at the kth iteration.
- Draw an input $\mathbf{x}$ from the training sample set randomly to the SOM.
- Find the winner among the nodes using equation 4.
- If(winner!=NULL), adjust the weights of the winner node c and its two neighborhood node $c-1$ and $c+1$.

  $$\begin{aligned} \mathbf{w}_c(k+1) &= \mathbf{w}_c(k) + \eta(k)(\mathbf{x} - \mathbf{w}_c(k)) \\ \mathbf{w}_{c-1}(k+1) &= \mathbf{w}_{c-1}(k) + \eta(k)\,\alpha(k)(\mathbf{x} - \mathbf{w}_{c-1}(k)) \\ \mathbf{w}_{c+1}(k+1) &= \mathbf{w}_{c+1}(k) + \eta(k)\,\alpha(k)(\mathbf{x} - \mathbf{w}_{c+1}(k)) \end{aligned}$$

  where $\eta(k)$ is the step size of learning, $\alpha(k)$ is a neighborhood function, k is the counter of iteration.
- If there is no winner, grow a new node n according to the growing scheme. $\mathbf{w}_n(k+1) = \mathbf{x}$ and $N = N+1$.
- If a node is rarely win, delete it according to pruning scheme, $N = N - 1$.
- 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 vector, 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.

3. TRANSDUCTION OF SOM

One of the problems of tracking by color segmentation is that the unknown lighting conditions may change the color of the object. Even in the case of fixed lighting sources, the color may still be different since the object may be shadowed by other objects. If we could find some invariants to the lighting conditions, this problem could be easily solved. However, there are still no such invariants in the state-of-the-art. These situations bring some difficulties to the approach of making a unique and fixed skin color model, since the distribution of skin color is non-stationary through image sequences so that the statistics of the distribution are not fixed.

Since we neither assume static backgrounds nor fixed lighting conditions, the probability density functions of colors in a certain color space, such as HSI or normalized RGB, will be non-stationary. At each time frame, the distribution of color is modeled by the proposed 1-D SOM, in which each neuron represents a color cluster of current time frame. This 1-D SOM also offer a simple color classifier by competition among its output neurons, through which the image at current time frame can be segmented. However, this classifier may not be good for the next time frame because of the non-stationary density of color.

Model adaptation over time was ever addressed in [8], in which a Gaussian mixture model was used, and a linear extrapolation was employed to adjust the parameters of the model by a set of labeled training data drawn from the new frame. However, since the new image is not segmented, these labeled data set is hard to obtain.

Our solution to this problem is called *transduction of SOM*, which is to update the weights and structure of the trained SOM according to a set of new training data so that the transduced SOM captures the new distribution. The new training data set in the transduction consists of both labeled and unlabeled samples. The algorithm is described below.

- $\mathcal{W}^{(n-1)} = \{\mathbf{w}_i^{(n-1)}, i = 1, \cdots, C^{(n-1)}\}$ are the weights of SOM at time frame $n-1$. The training data set $\mathcal{X}^{(n)} = \{\mathbf{x}_k^{(n)}, k = 1, \cdots, N\}$ is drawn randomly from the image at time frame n. We use $\mathcal{W}^{(n)}$ to represent SOM at time frame n.
- The training data set $\mathcal{X}^{(n)}$ is classified by the SOM $\mathcal{W}^{(n-1)}$, and is partitioned into two parts: a labeled data set $\mathcal{X}_l^{(n)}$ and an unlabeled data set $\mathcal{X}_u^{(n)}$. If a sample $\mathbf{x}_k^{(n)}$ is confidently classified by $\mathcal{W}^{(n-1)}$, then put this sample to the set $\mathcal{X}_l^{(n)}$ and label it with the index of the winning neuron of $\mathcal{W}^{(n-1)}$; otherwise, put it to $\mathcal{X}_u^{(n)}$ and let it unlabeled.
- **Unsupervised updating**: The algorithm described in section 2 is employed to update $\mathcal{W}^{(n-1)}$ by the unlabeled data set $\mathcal{X}_u^{(n)}$.
- **Supervised updating**: The labeled data set $\mathcal{X}_l^{(n)}$ is used in this step. $(\mathbf{x}_k, l_k)$ is drawn from $\mathcal{X}_l^{(n)}$, where l_k is the label for $\mathbf{x}_k$. The winning neuron for the input $\mathbf{x}_k$ is c.

$$\mathbf{w}_c^{(n)} = \begin{cases} \mathbf{w}_c^{(n-1)} + \alpha(\mathbf{x}_k - \mathbf{w}_c^{(n-1)}), \ if\ c = l_k; \\ \mathbf{w}_c^{(n-1)} - \alpha(\mathbf{x}_k - \mathbf{w}_c^{(n-1)}), \ if\ c \neq l_k; \end{cases}$$

After several iterations, the SOM at time frame $n-1$ is transduced to n.

4. LOCALIZATION SYSTEM

Our localization system is based on color segmentation. And motion segmentation and region-growing method are also employed to make the system more robust and accurate without introducing too much computation cost. Figure 3 shows the overview of the localization system.

The first frame taken by a camera is used to initially train the SOM by the proposed self-organizing clustering algorithm that has been described in section 2. In this initialization stage, the color distribution in the scene is initially mapped. In our experiments, the training is fast (less than 1 sec) with a 640x480 color image. The inputs of the SOM are the HSI value of pixels, and the outputs are the indexes of winning nodes of SOM through competition. Typically, it takes less than 6 nodes to segment indoor working environments.

For each newly captured color image at time frame t, the SOM is transduced by the algorithm described in section 3.

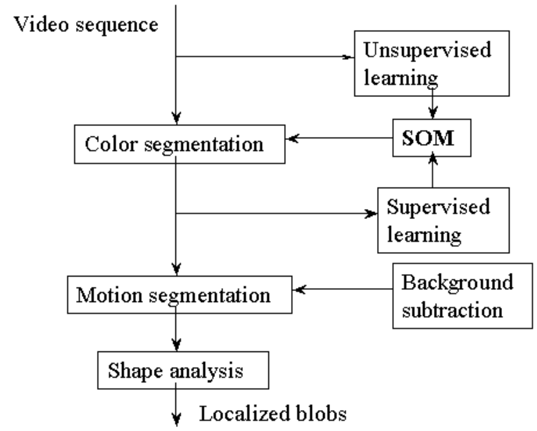

Figure 3. Localization system framework

Such SOM is used to segment the input image to find different color regions. This stage can be done on a lower resolution image the make the segmentation faster. Morphology operators are used to get rid of noise. After each pixel has been labeled, the SOM should be updated again by the supervised updating scheme described in the section 3. The labeled training data set is randomly selected from the segmented image, ignoring those that are too bright or dark.

Since there may be many different colors in the working space, and if the system does not specify which color to track, how to determine what to track is a problem. One possible solution is to specify a color region such as human hand or face. Another solution is to use some rules to automatically find an interested color from motion intention. If we detect a motion region by examine the frame difference or optic field, 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 faces or 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 or use some heuristics to find bounding boxes.

5. PERFORMANCE

Our color segmentation algorithm has been tested with a large variety of pictures. And our localization system that integrates this color segmentation algorithm has run under a wide range of operating conditions. The global hand tracking system based on our color segmentation supplies some inputs for our articulated hand motion capturing algorithm[12]. Experiments show that our color segmentation algorithm is fast, automatic and accurate, and the proposed localization system is robust, real-time and reliable. This color segmentation algorithm can also be applied to other segmentation tasks.

5.1. Performance of Segmentation

One 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.

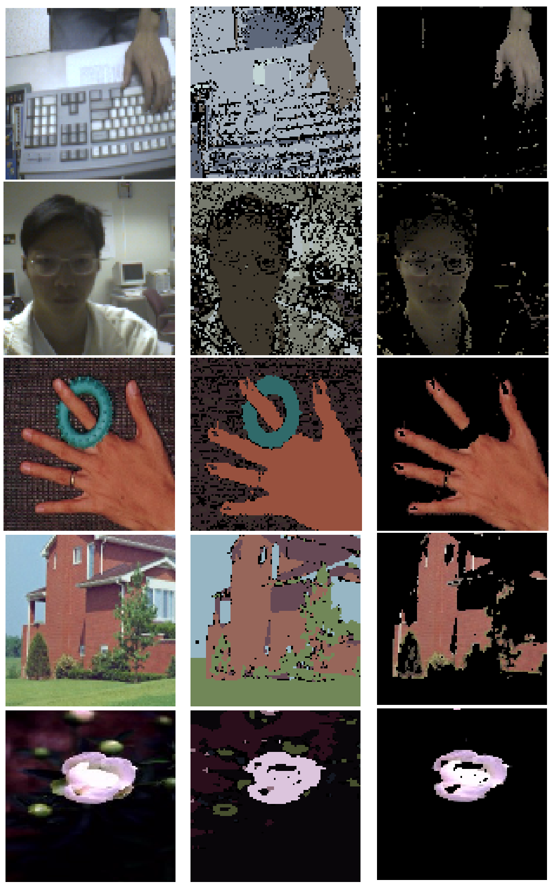

Figure 4. Color segmentation results. Left column are source color images, middle column are segmented images and right column are interested color regions.

Figure 4 show some segmentation results. Left column are source color images, middle column are segmented images, and right column are separated color regions. The colors of segmented color regions are the average colors of these regions. 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 through these experiments. When the background has less color distracters, this algorithm finds exact color regions. Since texture is not used in the segmentation, segmentation results will be noisy when there is color distracter texture 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.

5.2. Performance of Localization

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

Since our localization system is essentially based on a global segmentation algorithm, it does not largely rely on the tracking results 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.

Our proposed system can handle changing lighting condition to some extend because of the transduction of the SOM color classifier. At the same time, since the hue and saturation are given more weight than intensity, our system is insensitive to the change of lighting intensity such as the objects are shadowed or the intensity of the light source changes. However, there are still some problems. Insufficient lighting, too strong lighting, very dark and bright background may bring some troubles to the color segmentation algorithm, since hue and saturation become unstable and the system does not give more weights to intensity. If the lighting condition changes dramatically, the color segmentation algorithm may fail since the transduction cannot be guaranteed.

One localization results of our experiments is given in Figure 5. In this experiment, a hand is moving around with the interference 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. A demo sequence can be downloaded at http://www.ifp.uiuc.edu/˜yingwu.

Our localization system is very robust and efficient from this experiment in which the background of the scene is cluttered. 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 conditions, the tracker may be lost. However, it can quickly recover to continue working. Different skin tones do not affect our system. The first image with the interested color region is used to initially train the SOM so that it can work with nearly any users, which has been tested in our other experiments.

6. CONCLUSION

Localization of interested objects in video sequences is essential to many computer vision applications. Cluttered background, unknown lighting conditions and multiple moving objects make tracking tasks 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 presented a robust

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.

localization system based on self-organizing color segmentation and SOM transduction. A 1-D SOM is used to tessellate the HSI color space automatically. Images are segmented by this 1-D SOM through a competition process and each pixel of the image is labeled by the index of the winning node. Since the lighting condition and the background are not fixed, generally, the distribution of colors in the image sequence is not stationary. In order to capture the non-stationary color distribution, the 1-D SOM is transduced by combing supervised and unsupervised learning paradigms. Experiments show that our localization system is capable of reliable tracking multiple objects in real time on one-processor desktop SGI O2 workstation.

The transduction of SOM classifier is not mature, and it needs more efforts to find a better way to combine supervised and unsupervised learning schemes. 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 system offers a bounding box of the interested objects. Shape analysis of localized objects will be extended to estimate the 3D motion of the objects.

7. ACKNOWLEDGMENTS

This work was supported in part by National Science Foundation Grant IRI-9634618 and Grant CDA-9624396.

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] M.Jones and J.Rehg, “Statistical Color Models with Application to Skin Detection”. *Compaq Cambridge Research Lab. Technical Report CRL 98/11*, 1998

[5] 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

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

[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] Y.Raja, S.McKenna, S.Gong, “Colour Model Selection and Adaptation in Dynamic Scenes”, *Proc. ECCV’98*, 1998

[9] 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.

[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

[14] Ying Wu, Thomas S. Huang ”Robust Real-time Human Hand Localization by Self-Organizing Color Segmentation”, *IEEE Int’l Workshop on Recognition, Analysis and Tracking of Face and Gestures in Real-Time Systems*, Corfu, Greece, 1999