Real Time Limb Tracking with Adaptive Model Selection

Matheen Siddiqui and Gérard Medioni
University of Southern California
Institute for Robotics and Intelligent Systems
Los Angeles, CA 90089, USA
{mmsiddiq, medioni}@usc.edu

Abstract

We describe an efficient and robust method of tracking human forearms as skin colored regions. Of special consideration in the design of this system are real-time and robustness issues. We approach this as 2D tracking problem using skin color and edge information. Multiple 2D limb models are used to enhance tracking of the underlying 3D structure. This includes models for lateral forearm views (waving) as well as for pointing gestures. Experiments on test sequences demonstrate the efficacy of this approach.

1 Introduction and Related Work

The study of automatic and natural modes of interaction between humans and machines is an important field of research. A major contributing factor to this is the prevalence of machines in everyday life. As society advances, technology plays a larger supporting role and we increasingly find ourselves interfacing with machines and computers. Thus, there is a need for natural and social modes of interaction between machines and ordinary people. This is especially true when the targeted users might not be technologically savvy, or even have some disabilities preventing them from using traditional control devices and modalities [6].

For this purpose, it is important to not only locate potential users but also obtain information about their body configuration. Such information is not only useful for non-verbal communication but also in monitoring and activity recognition. However, a system designed to obtain such information needs to operate robustly and in real-time. This task is complicated by the high dimensionality of the upper-body, and the often unpredictable motion of the arms.

In our system we focus exclusively on the hands and forearms. To track these structures we make use of 2D trackers to track corresponding image regions. Using 2D trackers allows us to reduce dimensionally and overcome many singularities in fully articulated 3D models [7]. These trackers can be quite effective so long as the 2D shape is *similar* to the underlying 3D structure. To accommodate disparate views, we instantiate several 2D tracking models in parallel, each tuned to a particular view of the forearm.

Each individual tracker is akin to methods such as [4] and [1]. However, our algorithm accounts for the orientation of a region of interest, as in CAMShift [2]. Similar to [9], this is found via an optimization framework, however we directly use gradient based methods. While scale is not adapted directly in this process, it is implicitly accounted for in the model selection phase of our algorithm.

Furthermore, we explicitly model regions as skin blobs. This assumes users wear short sleeve shirts and their forearms are exposed. This assumption simplifies the detection of good visual features, as skin regions and boundaries can be extracted with greater ease, and occurs frequently enough to account for many situations. Skin color information is obtained from already detected faces, thus our system works without a learning step and on a variety of people.

The rest of this paper is organized as follows: In section 2 we present the details of our system. In section 3 we present the tracking models and model switching. In section 4 we demonstrate the effectiveness of this approach on test sequences. In section 5 we conclude and provide future directions of research.

2 Limb Tracking

With no prior knowledge of the limb locations, the entire image needs to be searched to find potential limbs. After this initialization, the limb can be tracked using local edge and skin color information. Edges are detected by simply using a Canny edge detector. To detect skin pixels, we first compute the skin likelihood of each pixel based on a hue-saturation space histogram trained on a detected face. Following this, we eliminate pixels that constitute the least likely 40%. The likelihood scores of the remaining pixels are then rescaled to be between 0 and 1.

The 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

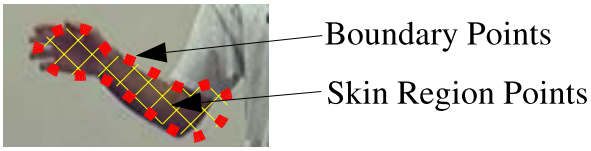


Figure 1. Segmented skin region and boundary.

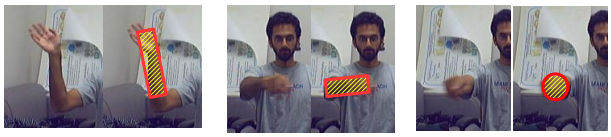


Figure 2. Tracking models.

To track a forearm, we model regions of interest as a collection of *feature sites* that indicate the presence of skin colored or boundary pixels. This is illustrated in Figure 1.

Tracking is achieved by maximizing the consistency of the feature sites with the underlying image. This can be posed as an optimization problem over translation ($\mathbf{t}$) and orientation (θ) as expressed in the following equation:

$$\theta*, \mathbf{t}* = \arg\min \sum_{\mathbf{x}\in BP} D_{dist}(R(\theta)\mathbf{x}+\mathbf{t}) + \lambda \sum_{\mathbf{x}\in SP} F_{skinScore}(R(\theta)\mathbf{x}+\mathbf{t}) \quad (1)$$

where $R(\theta)$ is a rotation matrix, BP consists of boundary points, SP consists of skin points, and $D_{dist}()$ yields the distance to the nearest boundary point within the region of interest. This is efficiently calculated using a distance transform of detected edge points[5]. The term $F_{skinScore}(\mathbf{x})$ represents a function that is zero when the image has a skin colored pixel at location $\mathbf{x} = (x, y)$ and is large otherwise.

While a natural choice for $F_{skinScore}(\mathbf{x})$ is the negative logarithm of the skin pixel probability, $(-logP_{skin}(\mathbf{x}))$ we solve (1) using gradient based methods. Thus we need the $F_{skinScore}(\mathbf{x})$ to be *smooth*. This is achieved by using its continuous distance transform [5]:

$$F_{skinScore}(\mathbf{y}) = min_{\mathbf{x}}(\|\mathbf{x}-\mathbf{y}\| + -\alpha log(P_{skin}(\mathbf{x}))) \quad (2)$$

This transform tends to give smoother images that have basins around regions of high skin probability as shown in Figure 4. This improves both the speed of convergence when solving (1) as well as the range of convergence.

We solve (1) directly by using a Levenberg-Marquardt optimizer [8] with a fixed number of iterations. We also only compute D_{dist} and $F_{skinScore}$ in a fixed size region about the previous pose of the forearm. This region is large enough to accommodate movement while keeping computational costs low. Finally, the face is masked out to prevent regions from being attracted to it.

3 Tracking Models

We use the tracker described in section 2 to track the user's forearm. For this purpose an appropriate model is required. The advantage of using simple 2D models to track the 3D forearm is that they can be used efficiently and robustly so long as they *match* the underlying 3D structure. However, a single 2D model is not ideal for all views. For example a lateral, waving forearm in which the profile of the forearm is visible, is significantly different from tracking an arm where a user is pointing towards the camera, and only the hand is visible.

To effectively track a forearm as it moves in 3D, we utilize multiple 2D tracking models as shown in Figure 2. The model shown in (a) is designed to track laterally viewed forearms, which often occurs in waving gestures. In (c) the circular shaped model is designed to track forearms pointing towards the camera. In this case, only the hand is visible and the circle effectively tracks this hand. The model in (b), which is just a shorter rectangle, is useful for situations between (a) and (c).

3.1 Model Selection

After a forearm is initialized, we must track it using the models described in section 3. To do this we need to understand how to switch between them, and how well they account for the underlying visual features. We quantify this using a *fullness* score and a *mass missed* score. The fullness score is the percent of pixels in the model's interior that are skin colored (i.e. whose skin likelihood is above a threshold). This quantifies how well the model explains the data inside itself. The mass missed score is the ratio of the number of skin pixels outside the model but inside an expanded region of interest to the total number of pixel inside the expanded region of interest. This quantifies how well the model explains the data surrounding itself.

During the tracking process all three models are running. One is selected based on observations and the state transition diagram (Figure 3) and its state can affect the other models. The transition diagram is designed to track a forearm as it switches from waving to pointing at the camera.

Initially, we choose the fully lateral forearm model since a gesture is often initiated by waving to a device. The other models are initialized so that their centers coincide and are aligned with the fully lateral model. As the fullness score changes we can switch to the $3/4$ and then pointing model if necessary. Similarly, if the fullness score increases we move

Current State	Condition	Next State
A	FS_A>.6	A
A	FS_A<.6	B
A	FS_A<.6 and FS_B < C	C
A	FS_A<.6 and FS_B < .6 and FS_C<.5	D
B	FS_B > .6 and (FS_A > .8 or FS_A > FS_B)	A
B	FS_B > .6 and FS_A < FS_B	B
B	FS_B<.6	C
B	FS_B<.6 and FS_C < .5	D
C	FS_C > .2 and MM_C>.6 and FS_B>.6	B*
C	FS_C > .2 and MM_C<.6	C
C	FS_C < .2	D

*Translate A and B so that their centroids coincide with that of C.

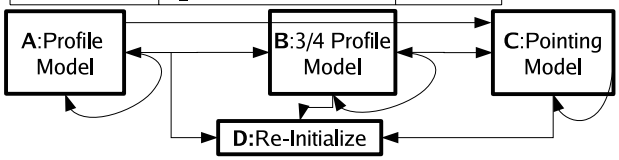


Figure 3. Model selection and state transitions. The table lists the conditions for transition between state. FS refers to the fullness score of a model, and MM refers to mass missed score.

from the $3/4$ profile model to the profile. However to transition out of the pointing state, its mass missed score must be above a threshold and $3/4$ profile models must have a high enough fullness score. When transitioning from the pointing state, both profile models' centroids are re-initialized to coincide with that of the pointing model.

In addition to switching between the models, we must also detect when the tracker loses track of the underlying forearm as shown in Figure 3. It is possible, for example, that one can move too fast and cause the tracker to lose its object. Knowing when this occurs is useful if this tracker where integrated in a larger system which could reset it

4 Results

This tracking system has been implemented as on a 3GHz Xeon System and currently runs between 10-25fps depending on the initialized scale of the system. As illustrated in the following sequences, this system has been extensively tested with various users and indoor settings.

In Figure 5(a) we shown an example in which the user points at a camera. The tracker is inaccurately initialized in frame 00. In frame 01 we see it was able to latch onto the nearby forearm which is subsequently tracked.

In Figure 5(b) the subject transitions between lateral and pointing positions. Between frames 00 and 08 subjects forearm transitions from a lateral (waving) position to a pointing position. Following this, in frames 10 to 16 the subjects forearm changes from pointing to a lateral view again.

In Figure 5(c) we show the system's results on a PETS sequence. Here the tracker stays in the fully lateral model.

5 Conclusion and Future work

We have described the design of a limb tracking system. The system works robustly and in real-time as demonstrated by the examples. We have successfully implemented this system in real-time processor on a Xeon 3GHz machine. The system works robustly and efficiently, and has been extensively tested qualitatively.

So far, we have successfully been able to track arms in single sequences. Our next steps, include feeding these results to a gesture recognition module[3]. We also are interested in extending this formulation for use with multiple cameras. Finally, more quantitative analysis of system performance will be performed using annotated test sequences.

6 Acknowledgments

This work was conducted in a collaborative project with the Electronics and Telecommunications Research Institute (ETRI), a Korean non-profit, government-funded research organization.

References

[1] S. T. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *CVPR*, pages 232–237, June 1998.

[2] G. R. Bradski. The OpenCV Library. *Dr. Dobb's Software Tools for the Professional Programmer*, November 2000.

[3] I. Cohen and H. Li. Inference of human postures by classification of 3d human body shape. In *IEEE Workshop on Analysis and Modeling of Faces and Gestures*, pages 74–81, Nice, France, 2003.

[4] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, May 2003.

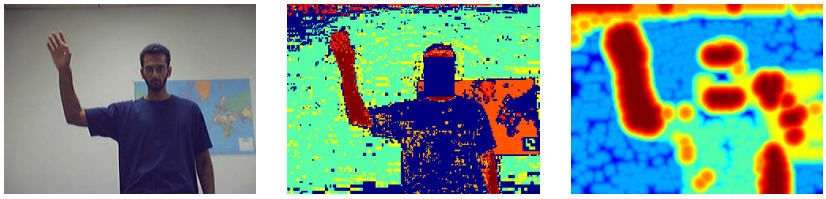
[5] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

[6] D. P. Miller. Assistive robotics: An overview. In *Assistive Technology and Artificial Intelligence, Applications in Robotics, User Interfaces and Natural Language Processing*, pages 126–136, London, UK, 1998. Springer-Verlag.

[7] D. D. Morris and J. M. Rehg. Singularity analysis for articulated object tracking. In *CVPR*, Santa Barbara, CA, June 1998.

[8] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 1992.

[9] H. Zhang, W. Huang, Z. Huang, and L. Li. Affine object tracking with kernel-based spatial-color representation. In *CVPR*, pages I: 293–300, 2005.



(a) (b) (c)

Figure 4. The negative log of Skin color probability before (b) and after (c) applying the continuous distance transform. The figure in (c) is more suitable for the optimization of equation (1) as the skin regions have smoother boundaries. As reference, the original frame is shown in (a). In these figures a red color indicates a smaller value while a blue color indicates a larger value.

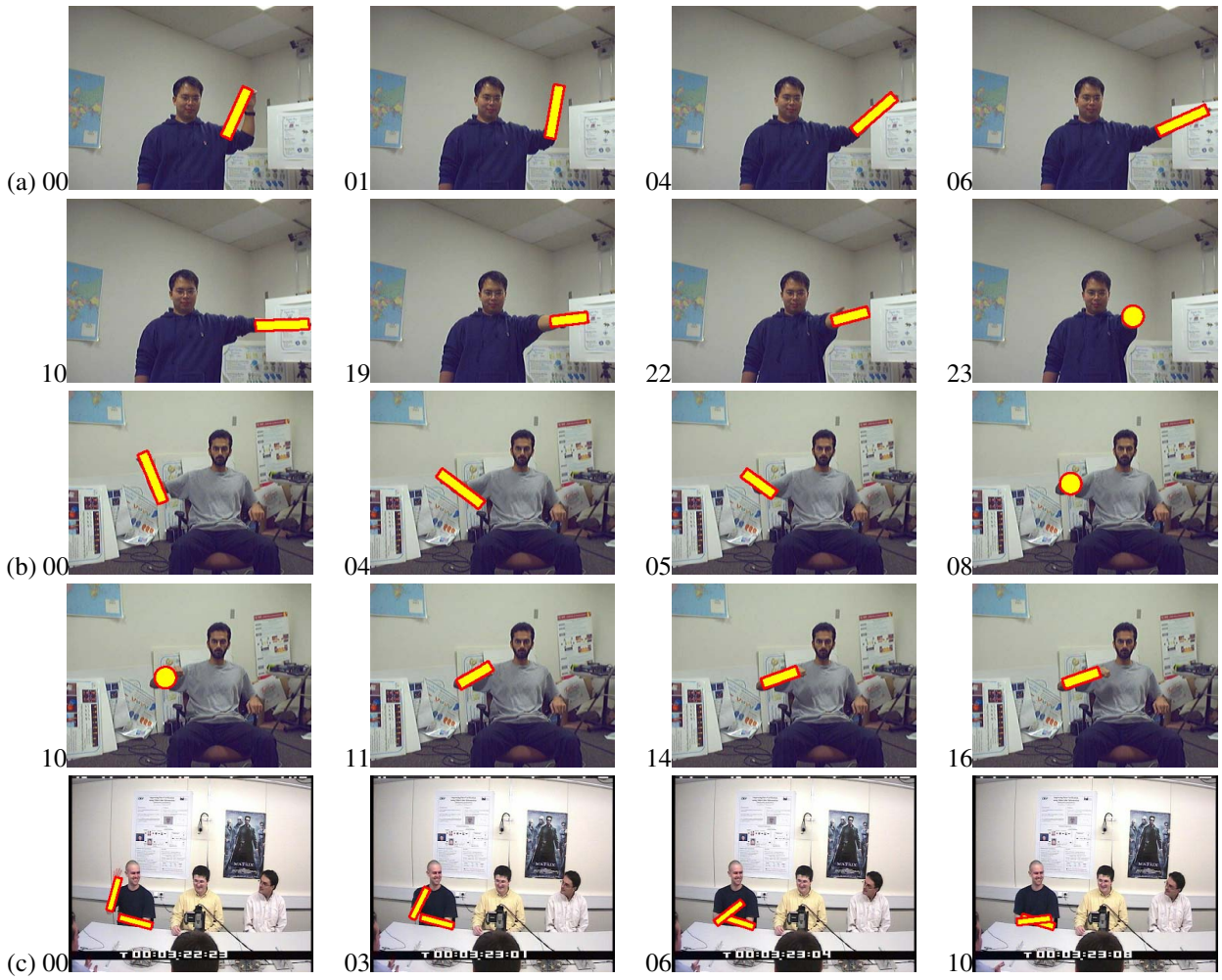


Figure 5. Results of the tracking system on various sequences. In (a) the system is inaccurately initialized in frame 00, but is able to recover in frame 01 and track as the user points at the camera. In (b) the user switches between waving and pointing and waving again and the tracker subsequently switches between lateral and pointing models. In (c) the results of the system on a PETS sequence are shown.