Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration

Mathias Kölsch and Matthew Turk
Department of Computer Science, University of California, Santa Barbara, CA 93106

Abstract

This paper introduces "Flocks of Features," a fast tracking method for non-rigid and highly articulated objects such as hands. It combines KLT features and a learned foreground color distribution to facilitate 2D position tracking from a monocular view. The tracker's benefits lie in its speed, its robustness against background noise, and its ability to track objects that undergo arbitrary rotations and vast and rapid deformations. We demonstrate tracker performance on hand tracking with a non-stationary camera in unconstrained indoor and outdoor environments. The tracker yields over threefold improvement over a CamShift tracker in terms of the number of frames tracked before the target was lost, and often more than one order of magnitude improvement in terms of the fractions of particular test sequences tracked successfully.

1. Introduction

We present a fast method for tracking the 2D location of unaugmented hands from monocular views. By integrating image cues obtained from optical flow and a color probability distribution, our method is able to follow rapid hand movements despite arbitrary finger configuration changes (postures). It can deal with dynamic backgrounds, some lighting changes, and significant camera motion such as from a hand-held camera during walking. It does not require a shape-based hand model, thus it is in principle applicable to tracking any deformable or articulated object. A more distinct and uniform object color increases performance but is not essential. We show extensive experiments with hands, demonstrating for our test sets up to 347% and 825% performance improvement over CamShift tracking [2], depending on the measurement method.

The tracker's core idea is motivated by the seemingly chaotic flight behavior of a flock of birds such as pigeons. While no single bird has any global control, the entire flock still stays tightly together, a large "cloud." This decentralized organization has been found to mostly hinge upon a simple constraint that can be evaluated on a local basis: birds like to maintain a minimum safe flying distance to the other birds, but desire not to be separated from the flock by more than another threshold distance [13].

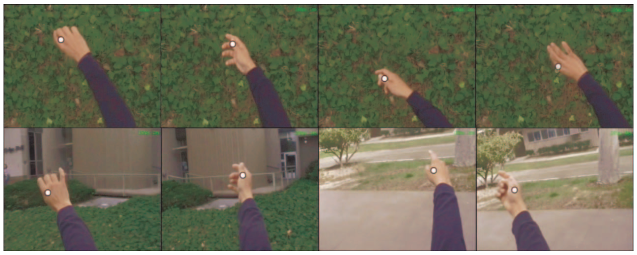

Figure 1: Tracking despite a non-stationary camera, hand articulations, and changing lighting conditions. The images are selected frames from sequence #5.

Our hand tracker consists of a set of small image areas, or features, moving from frame to frame in a way similar to a flock of birds. Their "flight paths" are determined by optical flow, and then constrained by observing a minimum distance from all other features and by not exceeding a maximum distance from the feature median. If these conditions are violated, the feature is repositioned to a location that has a high skin color probability. This fall-back on a second modality counters the drift of features onto nearby background artifacts that exhibit strong grey-level gradients.

The speed of pyramid-based KLT feature tracking [11, 16] allows our method to overcome the computational limitations of model-based approaches to tracking, easily achieving the real-time performance required for vision-based interfaces. The flocking behavior in combination with the color cue integration is responsible for the quality of the results: in our experiments, hands were tracked repeatedly for more than a minute despite all efforts to distract the tracker. Several examples are shown in the video clip that accompanies the paper and is available from the first author's web site.[1] A few frame snapshots are also shown in Figure 1.

Hand gesture human-computer interfaces have the potential to open new realms of applications and functionalities, especially for mobile and worn computing devices. Recognizing hand motions and configurations by means of computer vision is a particularly promising approach as it allows a maximum of versatility without encumbering the user. In prior work, we built a mobile computer that was

[1] currently at http://www.cs.ucsb.edu/~matz/RTV4HCI.wmv

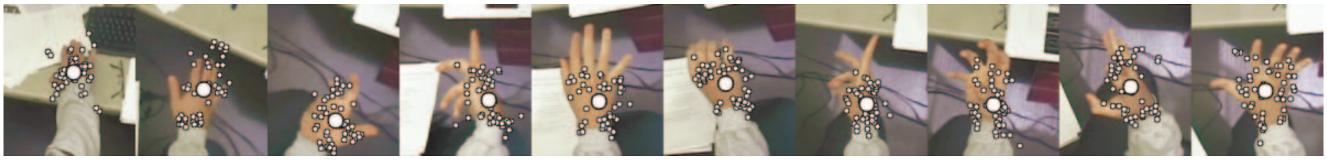

Figure 2: Snapshots from sequences #3 with highly articulated hand motions. 200x230 pixel areas were cropped from the 720x480-sized frames. The cloud of little dots represents the flock of features, the big dot is their mean. Note the change in size of the hand appearance between the first and fifth image and its effect on the feature cloud.

operated solely through hand gesture recognition with a head-worn camera, providing output in a head-worn display [9]. All other components need not be accessed and are stowed away in a conventional backpack. Robust hand detection (see [8]) initializes the vision system which then tracks the hand with the method described here. Key postures are recognized and – along with the 2D hand location – drive input to an application designed to support the mobile user. Posture recognition also serves as re-initialization of the tracking, reducing feature drift and accommodating for lighting changes. Critical aspects of the vision components are user independence, their robustness to arbitrary environments, and their computational efficiency as they must run in concert on a laptop computer, providing real-time and low-latency responses to user actions.

2. Related work

Rigid objects with a known shape can be tracked reliably before arbitrary backgrounds in grey-level images [1, 6]. However, when the object's shape varies vastly over time such as with gesturing hands, most approaches resort to shape-free color information or background differencing [4, 10, 14], thus being subject to unimodal failure modes, caused for example by a non-stationary camera. We use a multimodal technique that can overcome these vulnerabilities. Other multi-cue methods integrate for example texture and color information and can then recognize and track a small number of fixed shapes despite arbitrary backgrounds [3]. Our method tracks without a priori knowledge of possible postures and can handle any number of them. Depth information combined with color also yields a more robust hand tracker [5], yet stereo cameras are more expensive and cumbersome than the single imaging device required for our monocular approach.

Object segmentation based on optical flow (for example with normalized graph cuts [15]) can produce good results for tracking objects that exhibit a limited amount of deformations during global motions and thus have a fairly uniform flow [12]. Flocks of Features relaxes this constraint and can track despite concurrent articulation and location changes (see Figure 2).

Our method makes no attempt at estimating the articulation of the hand's phalanges (see for example [19, 17]), this is left for subsequent processing [9].

3. Method

The motivation for our new approach stems from the difficulty of tracking in real-time from a monocular view highly articulate objects such as hands during rapid movements. If the environment can not be constrained, for example, to a static or uniformly colored background, single-modality methods fail if only one assumption is violated. Our approach integrates two image cues in a very natural manner.

We chose pyramid-based KLT feature tracking as our first modality because it delivers excellent results on quickly moving rigid objects and because it can be computed very efficiently [11, 16]. The flocking feature behavior was introduced to allow for tracking of objects whose appearance changes over time, to make up for features that are "lost" from one frame to another because the image mark they were tracking disappeared. Since mere feature reintroduction within proximity of the flock can not provide any guarantees on whether it will be located on the object of interest or some background artifact, color as the second modality is consulted to aid in the choice of location. An overview of the entire algorithm is given in Figure 3.

3.1. KLT features and tracking initialization

KLT features are named after Kanade, Lucas, and Tomasi who found that a steep brightness gradient along at least two directions makes for a promising feature candidate to be tracked over time ("good features to track," see [16]). In combination with image pyramids (a series of progressively smaller-resolution interpolations of the original image [11]), a feature's image area can be matched efficiently to the most similar area within a search window in the following video frame. The feature size determines the amount of context knowledge that is used for matching. If the feature match correlation between two consecutive frames is below a threshold, the feature is considered "lost."

In the presented system, a hand detection method [8] supplies both a rectangular bounding box and a probabil-

```
input:
bnd_box - rectangular area containing hand
mindist - minimum pixel distance between features
n       - number of features to track
winsize - size of feature search windows

initialization:
learn color histogram
find n*k good-features-to-track with mindist
rank them based on color and fixed hand mask
pick the n highest-ranked features

tracking:
update KLT feature locations with image pyramids
compute median feature
for each feature
  if less than mindist from any other feature
     or outside bnd_box, centered at median
     or low match correlation
  then relocate feature onto good color spot
     that meets the flocking conditions

output:
the average feature location
```

Figure 3: The Flocks of Features tracking algorithm. k is an empirical value, chosen so that enough features end up on good colors; we use $k = 3$. The fixed hand mask is a known spatial distribution for pixels belonging to some part of the hand in the initialization posture.

ity distribution to initialize tracking. The probability mask was learned offline and states for every pixel in the bounding box the likelihood that it belongs to the hand. Next, approximately 100 features are selected within the bounding box according to the goodness criterion and observing a pairwise minimum distance. These features are then ranked according to the combined probability of their locations' mask- and color probabilities. The *target number* highest-ranked features form the subset that is chosen for tracking. Its cardinality will be maintained throughout tracking by replacing lost features with new ones.

Each feature is tracked individually from frame to frame. That is, its new location becomes the area with the highest match correlation between the two frame's areas. The features will not move in a uniform direction; some might be lost and others will venture far from the flock.

3.2. Flocks of Features

The main innovation presented in this paper, Flocks of Features, allows for tracking of fast-moving and articulating objects without the need for an object model. (The color distribution can be seen as a model, yet it is not known a priori but learned on the fly.) It is a way of enforcing a loose global constraint on the feature locations that keeps them spatially confined. During tracking, the feature locations are first updated like regular KLT features as described in the previous subsection and their median is computed. Then, two conditions are enforced at every frame: no two features must be closer to each other than a threshold distance, and no feature must be further from the feature median than a second threshold distance. Unlike birds that will gradually change their flight paths if these "flocking conditions" are not met, our method abruptly relocates affected features to a new location that fulfills the conditions. The flock of features can be seen in Figure 2 as clouds of little dots.

The effect of this method is that individual features can latch on to arbitrary artifacts of the object being tracked, such as the fingers of a hand. They can then move independently along with the artifact, without disturbing most other features and without requiring the explicit updates of model-based approaches, resulting in flexibility and speed. Too dense concentrations of features that would ignore other object parts are avoided because of the minimum-distance constraint. On the other hand, stray features that are likely to be too far from the object of interest are brought back into the flock with the help of the maximum-distance constraint.

We chose the *median* over the mean location to enforce the maximum-distance constraint because of its robustness towards spatial outliers. In fact, we also remove the furthest 15% of features from the median computation to achieve temporally more stable results. However, the location of the tracked object as a whole is considered to be the *mean* of all features since this measure changes more smoothly over time than the median. The gained precision is important for the vision-based interface's usability.

3.3. Color modality and multi-cue integration

At hand detection time, the observed hand color is learned in a normalized-RGB histogram and contrasted to the background color as observed in a horseshoe-shaped area in the image around the hand. This assumes that no other exposed skin body parts of the same person who's hand is to be tracked is within that background reference area. Since our applications mostly assume a forward- and downward-facing head-worn camera, this assumption is reasonable. We ensured that it was met for our test videos, which also included other camera locations. The segmentation quality that this dynamic learning achieves is very good for as long as the hand's lighting conditions do not change dramatically and the reference background is representative for the actual background. For example, wooden objects that are not within the reference background area during learning will frequently be classified incorrectly as foreground color.

The color information is used as a probability map (of a pixel's color belonging to the hand) in three places. Firstly, the CamShift method which we compared our tracker to solely operates on this modality. Secondly, at tracker initialization time, the KLT features are placed preferably onto locations with a high skin color probability. This is true even for the two tracking styles that did not use color information in subsequent tracking steps, see Section 4.

Thirdly, the new location of a relocated feature (due to low match correlation or violation of the flocking conditions) is chosen to have a high color probability, currently above a fixed 50% threshold. If this is not possible without repeated violation of the flocking conditions, it is chosen randomly. A change in lighting conditions that results in poor classification with the learned distribution causes gracefully degrading tracking performance: only relocated features suffer while most features will continue to follow grey-level artifacts. For speed reasons we do not take the grey-level gradient information – the goodness-to-track – into account at this point anymore. However, we presume that this would not significantly improve tracking because in application the features automatically move to those locations after a few frames.

This method leads to a very natural multi-modal integration, combining cues from feature movement based on grey-level image texture with cues from texture-less skin color probability. The relative contribution of the modalities can be controlled by changing the threshold of when a KLT features is considered lost between frames. If this threshold is low, features are relocated more frequently, raising the importance of the color modality, and vice versa.

4. Experiments

The main objective of the experiments was to assess our tracker's performance in comparison to a frequently used, state of the art tracker. We chose the CamShift tracker [2] method because it is widely available and because it is representative of single-cue approaches. The contribution of both the flocking behavior and of the multi-cue integration to the overall performance was also of interest. We therefore compared five tracking styles:

- **CamShift:** We supplied the OpenCV implementation of CamShift [2] with the learned color distribution. A pilot study using a fixed HSV histogram yielded inferior results.

- **KLT features only:** The KLT features were initialized on the detected hand and subject to no restrictions during subsequent frames. If their match quality from one to the next frame was below a threshold, they were reinitialized randomly within proximity of the feature median.

- **KLT features with flocking behavior:** As above, but the constraints on minimum pairwise feature distance and maximum distance from the median were enforced at every frame (see Subsection 3.2).

- **KLT features with color:** As plain KLT features, but resurrected features were placed onto pixels with high skin-color probabilities (see Subsection 3.3).

- **Combined flocking and color cue:** Our main contribution, this tracking style combines the above two methods as described in Section 3.

All styles used color information that was obtained in identical ways. All KLT-based styles used the same feature initialization technique, based on a combination of known hand area locations and learned hand color. This guarantees equal starting conditions to all styles.

Feature tracking was performed with three-level pyramids in 720x480 video, which arrived at our DirectShow filter at approximately 13fps. The tracking results were available after 2-18ms processing time, depending on search window size and the number of features tracked.

Aside from comparing different tracking styles, we also experimented with different parameterizations of our method. While further investigation is necessary, we present first results of varying the following independent variables: the number features tracked, the minimum pairwise feature distance, and the feature search window size.

4.1. Video sequences

We recorded a total of 518 seconds of video footage in seven sequences. Each sequence follows the motions of the right hand of one of two people, some filmed from the performer's point of view, some from an observer's point of view. For 387 seconds (or 4979 frames) at least one of the styles successfully tracked the hand. Table 1 details the sequences' main characteristics. The videos were shot in our lab and at various outdoor locations, the backgrounds including walkways, random vegetation, bike racks, building walls, etc. The video was recorded with a hand-held DV camcorder, then streamed with FireWire to a 3GHz desktop computer and processed in real-time. The hand was detected automatically when in a certain "initialization" posture with a robust hand detection method [8]. Excerpts of the sequences can be found in the video accompanying this paper (see Introduction).

5. Results

We define tracking to be lost when the mean location is not on the hand anymore, with extremely concave postures being an exception. The tracking for the sequence was stopped then, even though the hand might later have coincidentally "caught" the tracker again due to the hand's path intersecting the erroneously tracked location. Since the average feature location can not be guaranteed to be on the center of the hand or any other particular part, merely measuring the distance between the tracked location and some ground truth data can not be an accurate measure for determining tracking loss. We thus visually inspected and manually annotated the results.

id	outdrs	post.	skin bg	len	max tracked	
1	no	no	yes	95s	79.3s	1032f
2	no	yes	yes	76s	75.9s	996f
3	no	lots	little	32s	18.5s	226f
4	yes	yes	little	72s	71.8s	923f
5	yes	yes	yes	70s	69.9s	907f
6	yes	yes	yes	74s	31.4s	382f
7	yes	yes	yes	99s	40.1s	513f

Table 1: The video sequences. Three were taken indoors, four in the outdoors. In the first one, the hand was held in a mostly rigid posture (fixed finger flexion and orientation), all other sequences contained posture changes (column "post."). The videos had varying amounts of skin-colored background ("skin bg") within the hand's proximity. Their full length is given in seconds, counting from the frame in which the hand was detected and tracking began. The maximum time and number of frames that the respectively best method tracked a given sequence are stated in the last column.

5.1. Better than CamShift

Figure 4 illustrates our method's performance in comparison to a CamShift tracker that is purely based on color. The leftmost bar for each of the seven sequences shows that CamShift performs well on sequences three and four due to the limited amount of other skin-colored objects nearby the tracked hand. In all other sequences, however, the search region and the area tracked quickly expand too far and lose the hand in the process.

The other bars are from twelve Flocks of Features trackers with 20-100 features and search window sizes between 5 and 17 pixels squared. Out of these twelve trackers, the worst and mean tracker for the respective sequence is shown. In all but two sequences, even the worst tracker outperforms CamShift, while the best tracker frequently achieves an order of magnitude better performance (each sequence's best tracker is normalized to 1 on the y-axis and not explicitly shown). The rightmost bar in each group represents a single tracker's performance: the overall best tracker which had 15x15 search windows, 50 features and a minimum pairwise feature distance of 3 pixels.

Next, we investigated the relative contributions of the flocking behavior and the color cue integration on the combined tracker's performance. Figure 5 indicates that adding color as an additional image cue contributes more to the combined tracker's good performance than the flocking behavior in isolation. The combination of both techniques achieves the overall vast improvements over the CamShift tracker.

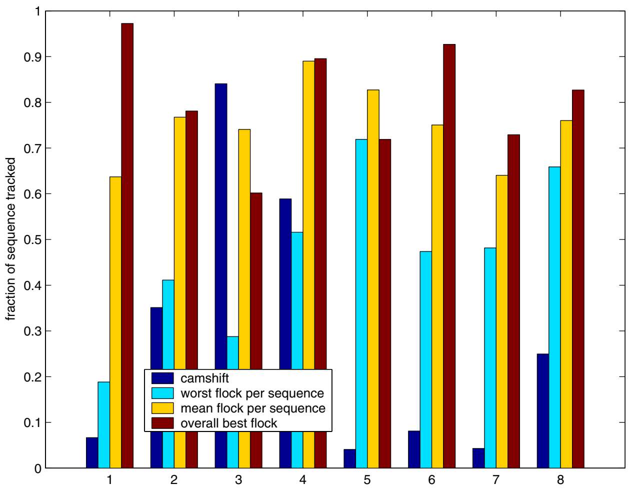

Figure 4: This graph shows the time until tracking was lost for each of the different tracking styles, normalized to the best style's performance for each video sequence. Groups 1-7 are the seven video sequences. Group 8 is the sum of all sequences, normalized to the sum of each sequence's best style's performance. Our Flocks of Features track the hand much longer than the comparison tracker.

5.2. Parameter optimizations

Figure 6 presents the tracking results after varying the target number of features that the flocking method maintains. The mean fraction's plateau suggests that 50 features are able to cover the hand area equally well as 100 features. The search window size of 11x11 pixels allows for overlap of the individual feature areas, making this a plausible explanation for no further performance gains after 50 features.

In a related result (not shown), we observed no significant effect related to the minimum pairwise feature distance in the range between two and four. Smaller threshold values however (especially the degenerative case of zero) allow very dense feature clouds that retract to a confined part on the tracked hand, decreasing robustness significantly.

Just as the previous two parameters, the search window size should ideally depend on the size of the hand and possibly the size of its articulations. We did not dynamically adjust these values since our experiments were conducted exclusively on hands, which were also within a size factor of about two of each other (an example for scale change are the first and fifth image in Figure 2). The window size has two related implications. A larger size should be better at tracking global motions (position changes), while a smaller size should perform advantageously at following finger movements (hand articulations). Second, larger areas are more likely to cross the boundary between hand and background. Thus it should be more difficult to pronounce a feature lost based on to its match correlation. However, Fig-

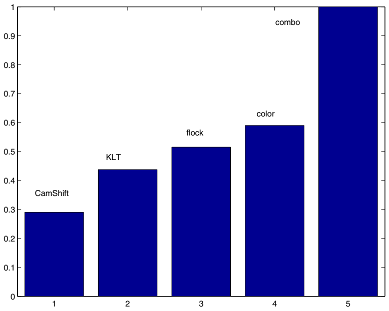

Figure 5: Contribution of flocking versus color to the combo tracker's performance. Shown is the normalized sum of the number of frames tracked with each tracker style, similar to the eighth group in Figure 4. The combo tracker distinctively shows synergy effects over the other trackers' performances.

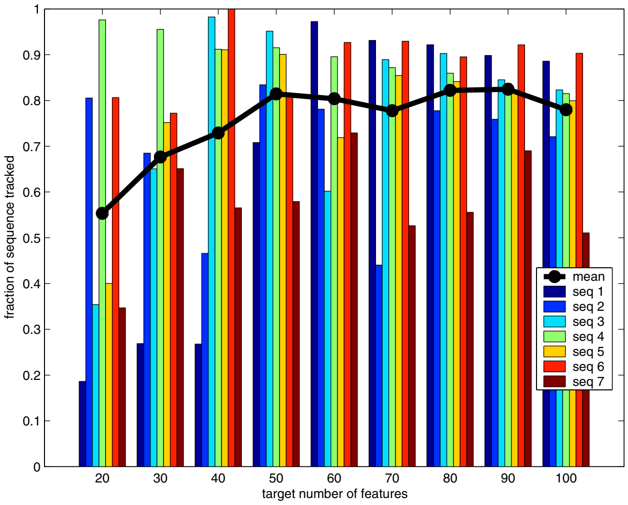

Figure 6: How varying the number of features influences the performance for each of the video sequences. The KLT features were updated within an 11x11 search window and a pairwise distance of 2.0 pixels was enforced. (The bars are normalized for each sequence's best tracker, which might not be shown here.)

ure 7 does not explicitly show these effects. We suspect that other factors play a role in how well the sequences come off, which warrants further investigation. On the other hand, the general trend is very pronounced and we chose the tracker parameters accordingly.

6. Discussion

The experiments showed that the performance improvement must be attributed to two factors. First, the purely texture-based and thus within-modality technique of flocking behavior contributes as witnessed by comparing KLT features with and without flocking (see Figure 5). Second, the cross-modality integration adds to the performance, visible in improvements from flocking-only and color-only to the combined approach.

A perfect integration technique for multiple image cues would reduce the failure modes to simultaneous violations of *all* modalities' assumptions. To achieve this for our method and its on-demand consultation of the color cue, a failure in the KLT/flocking modality would have to be detectable autonomously (without help from the color cue). To the best of our knowledge, this cannot be achieved theoretically. In practice, however, each feature's match quality between frames is a good indicator for when the modality might not be reliable. This was confirmed by our experiments as we could observe the features to flock towards the center of the hand (and its fairly stable appearance there) as opposed to the borders to the background where rapid appearance changes are frequent.

The presented method's limitations can thus be attributed to two causes, undetected failure of the KLT tracking and simultaneous violation of both modalities' assumptions. The first case occurs when features gradually drift off to background areas without being considered lost nor violating flocking constraints. The second case occurs if the background has a high skin-color probability, has high grey-level gradients to attract and capture features, and the tracked hand undergoes transformations that require many features to reinitialize.

Flocks of Features frequently track the hand successfully despite partial occlusions. Full object occlusions can be impossible to handle at the image level and are better dealt with at a higher level, such as with physical and probabilistic object models [7, 18]. Our method improves the input to these models, providing them with better image observations that will in turn result in better model parameter estimates. Enforcing temporal consistency by applying a Kalman filter or particle filtering methods is another way to improve tracking robustness.

There is a performance correlation between the target number of features, the minimum distance between features, and the search window size. The optimal parameters also depend on the size of the hand, which we currently assume to vary after initialization with no more than approximately a factor of two in each dimension. In our future work, we will attempt to quantify these relationships and derive optimal parameters for different object sizes.

Our method was designed for coarse hand tracking for a time span in the order of one minute. It is to provide 2D position estimates to an appearance-based posture recognition method that does not require an overly precise bounding box on the hand area. Thus, it was sufficient to obtain the location of some hand area, versus that of a particular spot such as the index finger's tip. In our complete vision-

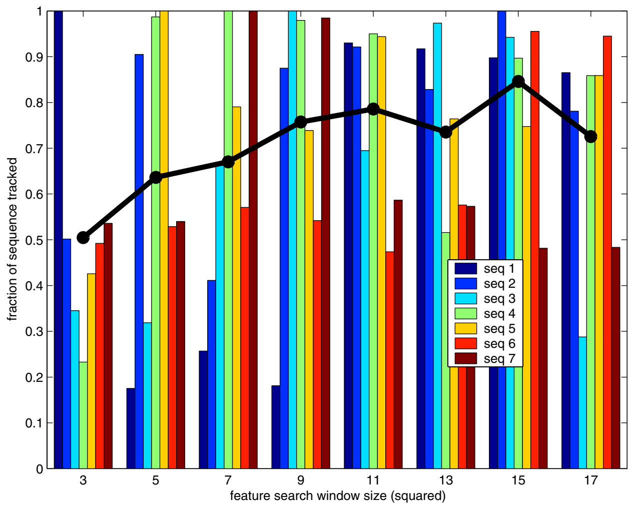

Figure 7: How tracker performance is affected by search window size (square, side length given on x-axis). Larger window sizes improve tracking dramatically for sequences with very rapid hand location changes (sequences 3, 4, 5), but tracking of fast or complicated configuration variations suffer with too large windows (sequences 3, 7).

based gesture interface (see [9], also briefly described in the Introduction), every successful posture classification re-initializes tracking and thus extends the tracking period into the long-term range.

The achieved frame rate was limited by the image acquisition and transmission hardware and not by the tracking algorithm. Higher frame rates will allow further performance improvements because KLT feature tracking becomes much faster and less error prone with shorter between-frame latencies.

7. Conclusions

We presented a real-time 2D tracking method for hands in monocular views. A flock of KLT features is maintained with the aid of a learned color distribution as a second image cue. A certain amount of size and color constancy after initialization is expected, but the method is robust to most other influences: it operates indoors and outdoors, with different people, and despite dynamic backgrounds and camera motion. The method requires no object model and thus might be applicable to tracking other very deformable and articulated objects such as human bodies. Its computation time of 2-18ms per 720x480 RGB frame leaves room for other computation such as hand posture recognition methods in order to build higher-fidelity vision-based gesture interfaces.

8. Acknowledgments

This work was partially supported under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48.

References

[1] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 232–237, June 1998.

[2] G. R. Bradski. Real-time face and object tracking as a component of a perceptual user interface. In *IEEE Workshop on Applications of Computer Vision*, pages 214–219, 1998.

[3] L. Bretzner, I. Laptev, and T. Lindeberg. Hand Gesture Recognition using Multi-Scale Colour Features, Hierarchical Models and Particle Filtering. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pages 423–428, Washington D.C., 2002.

[4] R. Cutler and M. Turk. View-based Interpretation of Real-time Optical Flow for Gesture Recognition. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pages 416–421, April 1998.

[5] S. Grange, E. Casanova, T. Fong, and C. Baur. Vision-based Sensor Fusion for Human-Computer Interaction. In *Intl. Conference on Intelligent Robots and Systems*, October 2002.

[6] M. Isard and A. Blake. A mixed-state CONDENSATION tracker with automatic model-switching. In *ICCV*, pages 107–112, 1998.

[7] N. Jojic, M. Turk, and T. Huang. Tracking Self-Occluding Articulated Object in Dense Disparity Maps. In *Proc. Intl. Conference on Computer Vision*, September 1999.

[8] M. Kölsch and M. Turk. Robust Hand Detection. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, May 2004.

[9] M. Kölsch, M. Turk, T. Höllerer, and J. Chainey. Vision-based Interfaces for Mobility. Technical Report TR 2004-04, University of California at Santa Barbara, February 2004.

[10] T. Kurata, T. Okuma, M. Kourogi, and K. Sakaue. The Hand Mouse: GMM Hand-color Classification and Mean Shift Tracking. In *Second Intl. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems*, July 2001.

[11] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proc. Imaging Understanding Workshop*, pages 121–130, 1981.

[12] F. K. H. Quek. Unencumbered Gestural Interaction. *IEEE Multimedia*, 4(3):36–47, 1996.

[13] C. W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics*, 21(4):25–34, 1987. SIGGRAPH '87 Conference Proceedings.

[14] J. Segen and S. Kumar. GestureVR: Vision-Based 3D Hand Interface for Spatial Interaction. In *The Sixth ACM Intl. Multimedia Conference*, September 1998.

[15] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *Proc. Intl. Conference on Computer Vision*, pages 1154–1160, 1998.

[16] J. Shi and C. Tomasi. Good features to track. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, June 1994.

[17] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Filtering Using a Tree-Based Estimator. In *Proc. 9th International Conference on Computer Vision*, volume II, pages 1063–1070, Nice, France, October 2003.

[18] C. R. Wren and A. P. Pentland. Dynamic Models of Human Motion. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pages 22–27. IEEE Computer Society, April 1998.

[19] Y. Wu and T. S. Huang. Hand Modeling, Analysis, and Recognition. *IEEE Signal Processing Magazine*, May 2001.