Flocks of Features for Tracking Articulated Objects

Mathias Kölsch[1] and Matthew Turk[2]

[1] Computer Science Department, Naval Postgraduate School, Monterey, CA
matz@nps.edu
[2] Computer Science Department, University of California, Santa Barbara, CA
mturk@cs.ucsb.edu

Tracking non-rigid and articulated objects in live video is a challenging task, particularly because object geometry and appearance can undergo rapid changes between video frames. Color-based trackers do not rely on geometry, yet they have to make assumptions on the background's color as to avoid confusion with the foreground object.

This chapter presents "Flocks of Features," a tracking method that combines motion cues and a learned foreground color distribution for fast and robust 2D tracking of highly articulated objects. Many independent image artifacts are tracked from one frame to the next, adhering only to local constraints. This concept is borrowed from nature since these tracks mimic the flight of flocking birds – exhibiting local individualism and variability while maintaining a clustered entirety.

The method's benefits lie in its ability to track objects that undergo vast and rapid deformations, its ability to overcome failure modes from the motion cue as well as the color cue, its speed, and its robustness against background noise. Tracker performance is demonstrated on hand tracking with a non-stationary camera in unconstrained indoor and outdoor environments. When compared to a CamShift tracker on the highly varied test data, Flocks of Features tracking yields a threefold improvement in terms of the number of frames of successful target tracking.

Keywords: hand tracking, gesture recognition, human-computer interaction

1 Introduction

Flocks of Features is a fast method for tracking the 2D location of highly articulated objects from monocular views, for example, human hands. By integrating image cues obtained from optical flow and a color probability distribution, a flock of features is able to follow rapid hand movements despite arbitrary finger configuration changes (postures). It can deal with dynamic backgrounds, gradual lighting changes, and significant camera motion such as experienced with a hand-held camera during walking. It does not require a geometry model or a shape model of the target, thus it is in principle applicable to tracking any deformable or articulated object. Tracking performance increases with a more distinct and more uniform object color. The Flocks of Features method was first presented at the IEEE Workshop on Real-Time Vision for Human-Computer Interaction [13].

1.1 Flocking Behavior

The method's core idea was motivated by the seemingly chaotic clustering behavior of a school of fish or a flock of birds, for example, pigeons. While no single bird has any global control, the entire flock still stays tightly together. This decentralized organization has been found to mostly hinge upon two simple constraints that can be evaluated on a local basis: birds like to maintain a minimum safe flying distance to the other birds, but they also desire not to be separated from the flock by more than a certain maximum distance [19].

A Flocks of Features tracker utilizes a set of small image areas, or features, that move from frame to frame in a way similar to a flock of birds. The features' "flight paths" are determined by optical flow, resulting in independent feature movements. Thereafter, every feature's location is constrained to observe a minimum distance from all other features, and to not exceed a maximum distance from the feature median. If one or both of these conditions are violated, the feature is repositioned to a new location that is in compliance again. In addition, an attempt is made to select new locations with a high skin color probability. This consultation of a second image cue counters the drift of features onto nearby background artifacts as it might happen if these exhibit strong grey-level gradients.

The speed of pyramid-based KLT feature tracking (see Sec. 3.1 and [4, 17, 22]) allows Flocks of Features to overcome the computational limitations of model-based approaches to tracking, easily achieving the real-time performance that is required for vision-based interfaces. The flocking behavior in combination with the color cue integration is responsible for the quality of the results: in experiments (see Sec. 4), 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 is associated with the book and is available from the first author's web page.[3] A few frame snapshots are also shown

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

Fig. 1. Hand tracking despite a moving camera, hand rotations and articulations, changing lighting conditions and backgrounds. (The images are selected frames from sequence #5, see Table 1)

in Fig. 1. Section 5 covers extensive experiments with hands, demonstrating significant performance improvement over another popular tracking method, called CamShift [2].

1.2 HandVu

Human-computer interfaces that observe and utilize hand gestures have the potential to open new realms of applications and functionalities, especially for non-traditional computing environments such as augmented reality and 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. The tracker described here is an integral part of *HandVu*,[4] the first vision-based hand gesture interface that is publicly available and that allows quick and easy interface deployment (see [12]). For example, *HandVu* is used to operate a mobile computer [15] solely through hand gesture recognition. A head-worn camera provides the input, and a head-worn display in the same physical unit is responsible for the visual output. All other components need not be accessed and are stowed away in a conventional backpack.

In *HandVu*, robust hand detection (see [14]) 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 the applications. Posture recognition also serves as re-initialization of the tracking, reducing feature drift and accommodating for lighting changes. Key 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.

[4] currently at http://www.cs.ucsb.edu/~matz/HGI/HandVu.html

2 Related Work

Rigid objects with a known shape can be tracked reliably before arbitrary backgrounds in grey-level images [1, 8]. 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 [5, 16, 20]. This makes assumptions about the background color or requires a stationary camera and a fixed background, respectively. Violation of just one of these assumptions has to be considered a unimodal failure mode. The Flocks of Features method, on the other hand, uses a multimodal technique to overcome these vulnerabilities. Other multi-cue approaches integrate, for example, texture and color information and can then recognize and track a small number of fixed shapes despite arbitrary backgrounds [3]. A flock of features tracks without a priori knowledge of possible postures and can handle any number of them. However, it makes no attempt at estimating the articulation of the hand's phalanges or finger configurations, this is left for subsequent processing (for example, see [25, 23, 15]).

Object segmentation based on optical flow can produce good results for tracking objects that exhibit a limited amount of deformations during global motions and thus have a fairly uniform flow [18, 5]. Flocks of Features relaxes this constraint and can track despite concurrent articulation and location changes (see Fig. 3). Depth information combined with color also yields a robust hand tracker [6], yet stereo approaches have their own limitations and are more expensive than the single imaging device required for monocular approaches.

The Flocks of Features approach is different from Monte Carlo methods (often called particle filters, condensation, or particle swarm optimization) [21, 7, 11]. The features in a flock react to local observations only and do not have global knowledge as the samples or particles in Monte Carlo methods do. The features' realm is the two-dimensional image data (optical flow), not a higher-level model space. But most of all, they move in a deterministic way, rather than probabilistically sampling their state space. Having said that, the repositioning of features that have violated the flocking conditions could be interpreted as an attempt to probabilistically model a global "distribution" of the tracked object (for example, the hand), with feature distance and color as its two marginalizations.

3 Method

The motivation for this approach stems from the difficulty of tracking highly articulated objects such as hands during rapid movements. This is particularly challenging when real-time constraints have to be met and only a monocular view in the visible light spectrum is available. 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. The approach that Flocks of Features takes integrates two image cues in a very natural manner.

The first image cue exploits the fact that object artifacts can be followed reliably if their appearance remains fairly constant between frames. The method of choice is a popular tracking method that was conceived by Kanade, Lucas, and Tomasi [17, 22], frequently referred to as pyramid-based KLT feature tracking. It delivers good accuracy on quickly moving rigid objects and it can be computed very efficiently. 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.

The second image cue is color: mere feature re-introduction 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. Placing these features at image locations that exhibit a color similar to the hand's color, however, increases the chances of features being located on hand artifacts. An overview of the entire algorithm is given in Fig. 2.

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 [22]). In combination with image pyramids (a series of progressively smaller-resolution interpolations of the original image [4, 17]), 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 mentioned *HandVu* system [12], a hand detection method [14] supplies both a rectangular bounding box and a probability distribution to initialize tracking. The probability mask is learned offline and contains 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. This 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.

```
input:
  bound_box - rectangular area containing hand
  hand_mask - probability of every pixel in bound_box to be hand
  min_dist  - minimum pixel distance between features
  n         - number of features to track
  winsize   - size of feature search windows

initialization:
  learn foreground color histogram based on bound_box and hand_mask,
     and background color histogram based on remaining image areas
  find n*k good-features-to-track with min_dist
  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 min_dist from any other feature
       or outside bound_box, centered at median
       or low between-frames appearance match correlation
    then relocate feature onto good color spot
       that meets the flocking conditions

output:
  the average feature location
```

Fig. 2. The Flocks of Features tracking algorithm. Good-features-to-track [22] are those that have a strong grey-level image gradient in two or more directions, that is, corners. k is an empirical value, chosen so that enough features end up on good colors; $k = 3$ was found to be sufficient. The offline-learned hand mask is a spatial distribution for pixels belonging to some part of the hand in the initialization posture

3.2 Flocks of Features

As one of the method's key characteristics, fast-moving and articulating objects can be tracked without the need for an object model.[5] Flocking 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: no two features must be closer to each other than a threshold distance, and no feature must be further from the median than a second threshold distance. Unlike birds that will gradually change their flight paths if these "flocking conditions" are not

[5] The color distribution can be seen as a model, yet it is not known a priori but learned on the fly.

met, affected features are abruptly relocated to a new location that fulfills the conditions. The flock of features can be seen in Fig. 3 as clouds of little dots.

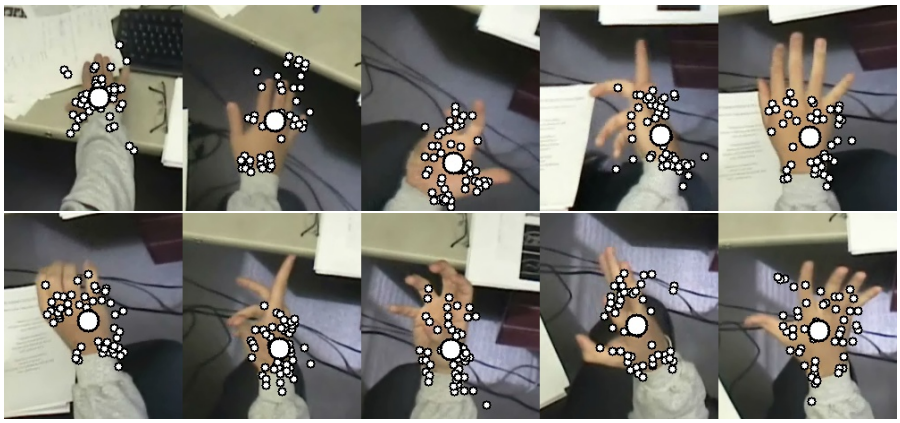

Fig. 3. These images are taken from individual frames of the video with highly articulated hand motions, sequence #3. Areas with 200x230 pixel were cropped from the 720x480-sized frames. The cloud of little dots represents the flock of features, the big dot is their mean. Note the change in size of the hand appearance between the first and fifth image and its effect on the feature cloud

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. Similarly, 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.

Chosing the *median* over the mean location to enforce the maximum-distance constraint is advantageous because of its robustness towards spatial outliers. In fact, the furthest 15% of features are also skipped for 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

When the hand is first detected, the observed hand color is learned in a normalized-RGB histogram and contrasted to the background color. The background color is sampled from a horseshoe-shaped area around the location

where the hand was detected (see Fig. 4). 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 most applications for *HandVu* assume a forward- and downward-facing head-worn camera, this assumption is reasonable. It was ensured that in the initialization frames of the test videos (which also included other camera locations) the reference area did not show the tracked person's skin. The color distribution was not restricted in subsequent frames. The segmentation quality that this dynamic learning achieves is very good for as long as the lighting conditions do not change dramatically and the reference background is representative for the actual background. The color cue is not a good fall-back method in cases where skin-colored objects that were not within the reference background area during learning come into view shortly thereafter.

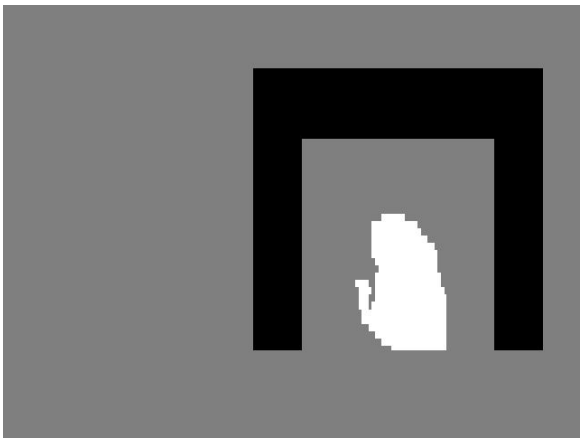

Fig. 4. The areas for learning the skin color model: The color in the hand-masked area (white) is learned in the foreground color histogram. The pixelized look stems from scaling the 30x20 sized hand mask to the detected hand's size. The background color histogram is learned from the horseshoe-shaped area around the hand (black); it is presumed to contain only background. The grey area is not used

The color information is used as a probability map (of a pixel's color belonging to the hand) in three places. First, the CamShift method – which Flocks of Features was compared to – solely operates on this modality. Second, 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 Sec. 4.

Third, 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 percent 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 color classification causes gracefully degrading tracking performance: only relocated features suffer while most features will continue to follow grey-level artifacts.

Feature relocation does not take the grey-level gradient information, the goodness-to-track, into account to save processing time. However, this is presumed to not significantly improve tracking because in application the features automatically move to those locations after a few frames.

The described consultation of the color cue leads to a very natural multimodal 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 described in the following is to assess Flocks of Features' performance in comparison to a frequently used, state of the art tracking method. A CamShift tracker [2] was chosen 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. Therefore, five tracking styles were compared:

- **1 – CamShift:** The OpenCV implementation of CamShift [2] was supplied with the learned color distribution. A pilot study using a fixed HSV histogram yielded inferior results.
- **2 – 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.
- **3 – KLT features with flocking behavior:** As style 2, but the constraints on minimum pairwise feature distance and maximum distance from the median were enforced (see Sec. 3.2).
- **4 – KLT features with color:** As style 2, but resurrected features were placed onto pixels with high skin-color probabilities (see Sec. 3.3).
- **5 – Combined flocking and color cue:** The actual Flocks of Features, this tracker combines styles 3 and 4 as described in Sec. 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 guaranteed equal starting conditions to all styles.

Feature tracking was performed with three-level pyramids in 720x480 video, which arrived at the tracking method 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, some of the experiments investigated different parameterizations of the Flocks of Features method. In particular, the the following independent variables were studied: the number features tracked, the minimum pairwise feature distance, and the feature search window size.

4.1 Video Sequences

A total of 518 seconds of video footage was recorded 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 an indoor laboratory environment and at various outdoor locations, the backgrounds including walkways, random vegetation, bike racks, building walls, etc. The videos were 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 [14]. Excerpts of the sequences can be found in the video associated with this chapter (see Introduction).

Table 1. The video sequences and their characteristics: three sequences 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. The videos had varying amounts of skin-colored background 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 in seconds and the maximum number of frames that the best method tracked a given sequence are stated in the last two columns

id	outdoors	posture changes	skin background	total length	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

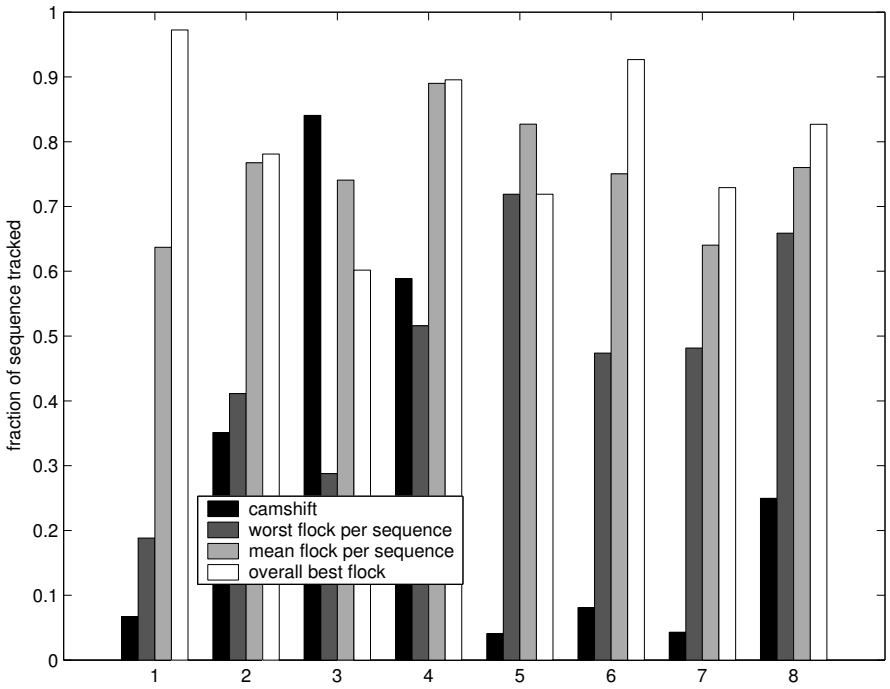

Fig. 5. This graph shows the time until tracking was lost for each of the different tracking styles, normalized to the best style's performance for each video sequence. Groups 1-7 are the seven video sequences. Group 8 is the sum of all sequences, normalized to the sum of each sequence's best style's performance. The Flocks of Features method tracks the hand much longer than the comparison tracker

5 Results

Tracking was defined 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. Thus, the tracking results were visually inspected and manually annotated.

5.1 General Performance

Figure 5 illustrates the method's performance in comparison to the 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.

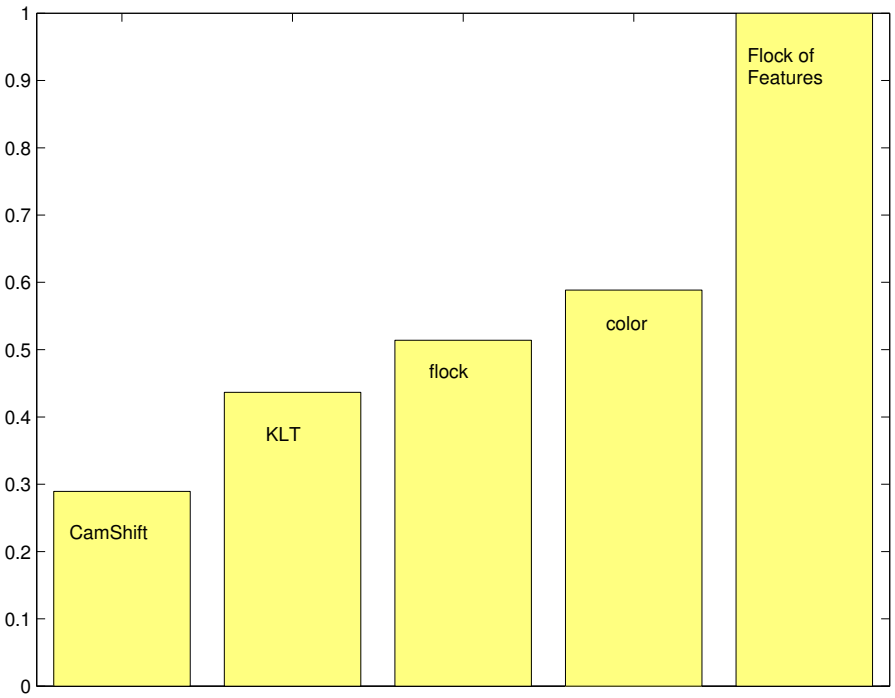

Fig. 6. Contribution of flocking versus color to the Flocks of Features' performance which combines both flocking and color information. Shown is the normalized sum of the number of frames tracked with each tracker style, similar to the eighth group in Fig. 5. Tracking with the Flocks of Features method distinctively shows synergy effects over the other methods' performances

Next, the relative contributions of the flocking behavior and the color cue integration on the combined tracker's performance were investigated. Figure 6 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 vast improvements over the CamShift tracker across the board.

5.2 Parameter Optimizations

Figure 7 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.

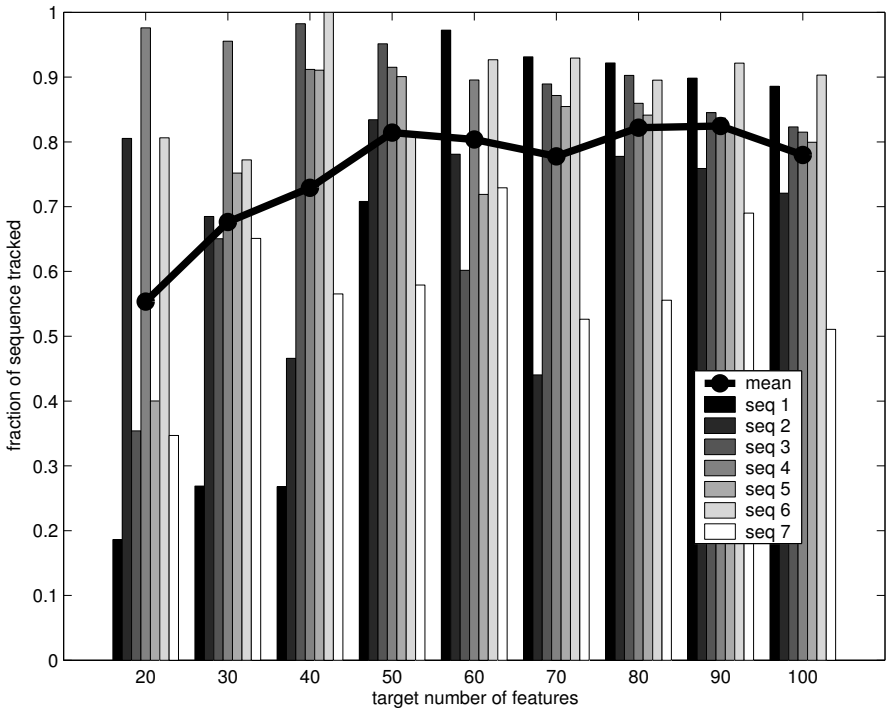

Fig. 7. How varying the number of features influences the performance for each of the video sequences. The KLT features were updated within an 11x11 search window and a pairwise distance of 2.0 pixels was enforced. The bars are normalized for each sequence's best tracker, of which only one (40 features, sequence #6) is shown here

In a related result (not shown), no significant effect related to the minimum pairwise feature distance was observed 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 on the size of its articulations. These values were constants in the experiments since they were conducted exclusively on hands. Further, the hand sizes did not vary by more than a factor of roughly two. An example for scale change are the hand appearances in the first and fifth image in Fig. 3. The window size has two related implications. First, 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. 8 does not explicitly show these effects. One possible explanation is that other factors play a role in how well the sequences fared, or the effect is not strong enough for the size of the data. On the other hand, the general trend is very pronounced and the tracker parameters were chosen 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 about 20 percent increase over KLT, as witnessed by comparing KLT features with and without flocking (see Fig. 6). 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 the presented 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 the experiments as the features flocked 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

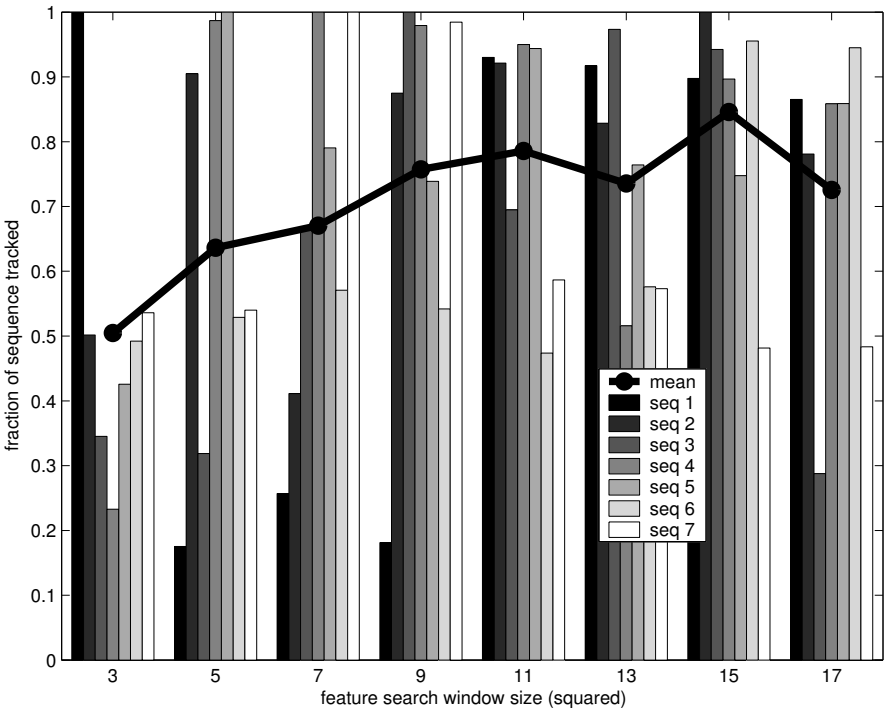

Fig. 8. How tracker performance is affected by search window size (square, side length given on x-axis). Larger window sizes improve tracking dramatically for sequences with very rapid hand location changes (sequences 3, 4, 5), but tracking of fast or complicated configuration variations suffer with too large windows (sequences 3, 7)

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 [9, 24]. The Flocks' output 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 [10] or Monte Carlo methods (see Sec. 2) 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 opti-

mal parameters also depend on the size of the hand, which is assumed to vary after initialization with no more than approximately a factor of two in each dimension. It is left for further investigation to quantify these relationships and to derive optimal parameters for different object sizes.

The Flocks of Features approach was conceived for coarse hand tracking for a time span in the order of ten seconds. 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 *HandVu*, the complete vision-based gesture interface (see [15, 12]), every successful posture classification re-initializes tracking and thus extends the tracking period into the long-term range.

The reported frame rate was limited by the image acquisition and transmission hardware and not by the tracking algorithm. During a second set of experiments with live video capture and processing, consistent frame rates of 30Hz were achieved (color firewire camera in 640x480 resolution). Higher frame rates allow "superlinear" performance improvements because KLT feature tracking becomes increasingly faster and less error prone with less between-frame object motion.

7 Conclusions

Flocks of Features is a new 2D tracking method for articulated objects such as hands. The method integrates two image cues, motion and color, to surpass the robustness of unimodal trackers towards lighting changes, background artifacts, and articulations. It operates indoors and outdoors, with different people, and despite dynamic backgrounds and camera motion. The method does not utilize a geometric object model, but, instead, enforces a loose global constraint on otherwise independently moving features. It is very fast (2-18ms computation time per 720x480 RGB frame), resulting in high frame rates (typical 30Hz on a 3GHz Xeon), but also leaving CPU cycles for other computation. For example, the vision-based hand gesture interface *HandVu* combines Flocks of Features with hand posture recognition methods in order to tap more than just the location of the hand for human-computer interaction. These novel interfaces point the way for natural, intuitive communication with machines in non-traditional environments such as wearable computing and augmented reality.

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. Stan Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 232–237, June 1998.
2. Gary R. Bradski. Real-time face and object tracking as a component of a perceptual user interface. In *Proc. IEEE Workshop on Applications of Computer Vision*, pages 214–219, 1998.
3. Lars Bretzner, Ivan Laptev, and Tony 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. Peter J. Burt and Edward H. Adelson. The Laplacian Pyramid as a Compact Image Code. *IEEE Trans. on Communication*, 31(4):532–540, 1983.
5. Ross Cutler and Matthew 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.
6. 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.
7. M. Isard and A. Blake. Condensation – Conditional Density Propagation for Visual Tracking. *Int. Journal of Computer Vision*, 1998.
8. Michael Isard and Andrew Blake. A mixed-state CONDENSATION tracker with automatic model-switching. In *Proc. Intl. Conference on Computer Vision*, pages 107–112, 1998.
9. 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.
10. R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME Journal of Basic Engineering*, pages 34–45, 1960.
11. James Kennedy and Russell Eberhart. Particle Swarm Optimization. In *Proc. IEEE Intl. Conference on Neural Networks*, volume 4, pages 1942–1948, 1995.
12. Mathias Kölsch. *Vision Based Hand Gesture Interfaces for Wearable Computing and Virtual Environments*. PhD thesis, Computer Science Department, University of California, Santa Barbara, September 2004.
13. Mathias Kölsch and Matthew Turk. Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration. In *IEEE Workshop on Real-Time Vision for Human-Computer Interaction (at CVPR)*, 2004.
14. Mathias Kölsch and Matthew Turk. Robust Hand Detection. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, May 2004.
15. Mathias Kölsch, Matthew Turk, and Tobias Höllerer. Vision-Based Interfaces for Mobility. In *Intl. Conference on Mobile and Ubiquitous Systems (MobiQuitous)*, August 2004.
16. Takeshi Kurata, Takashi Okuma, Masakatsu Kourogi, and Katsuhiko Sakaue. The Hand Mouse: GMM Hand-color Classification and Mean Shift Tracking. In *Intl. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems*, July 2001.
17. Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proc. Imaging Understanding Workshop*, pages 121–130, 1981.

18. Francis K. H. Quek. Unencumbered Gestural Interaction. *IEEE Multimedia*, 4(3):36–47, 1996.
19. C. W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 21(4):25–34, 1987.
20. J. Segen and S. Kumar. GestureVR: Vision-Based 3D Hand Interface for Spatial Interaction. In *Proc. ACM Intl. Multimedia Conference*, September 1998.
21. Caifeng Shan, Yucheng Wei, Tieniu Tan, and Frédéric Ojardias. Real Time Hand Tracking by Combining Particle Filtering and Mean Shift. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, 2004.
22. Jianbo Shi and Carlo Tomasi. Good features to track. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, June 1994.
23. 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.
24. Christopher R. Wren and Alex P. Pentland. Dynamic Models of Human Motion. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pages 22–27, April 1998.
25. Ying Wu and Thomas S. Huang. Hand Modeling, Analysis, and Recognition. *IEEE Signal Processing Magazine*, May 2001.

Index