Real-time Hand Tracking using Flocks of Features

Bc. Andrej Fogelton*
Supervised by: Ing. Matej Makula, PhD.†

Faculty of Informatics and Information Technologies
Slovak University of Technology
Bratislava / Slovakia

Abstract

There is a growing demand to interact with computers in a more natural way. For example using hand gestures to interact with certain type of applications would be more efficient than old-fashioned keyboard and mouse. Hand tracking is one of the key problems in computer vision. We have analyzed many different approaches used for hand tracking. Flocks of features introduced by Mathias Kölsch and Matthew Turk can track human hand continuously during various movements and pose variations. It uses the Kanade Lucas Tomasi (KLT) tracker for features located on a human hand to track them in a frame sequence. It can handle fast tracking of non-rigid highly articulated objects such as hands. We propose an improvement to this algorithm by processing the frame using histogram back projection of the skin color prior to applying flocks of features (FoF). This modification provides better results with lower false positive error.

Keywords: Hand Tracking, Flocks of Features, Back Projection, Histogram

1 Introduction

In the last few years, there is a growing demand to control computers in a more interactive way then using just mouse and keyboard. One of the pioneers of the new way of interaction was Nintendo WII,[1] which uses infrared LEDs and infrared camera with a proximity sensor. This device is used in a game console offering a totally new way of game experiencing. For example, you can play tennis by holding WII remote controller in your hand instead of your tennis racket and play a match against your friend or the computer.

The other promising product was introduced by Microsoft. The new version of their game console XBOX 360 uses *Kinect*,[2] which has the ambition to become even more popular than WII. Kinect is a webcam extended with infrared light camera and infrared light projector. This projector illuminates the scene with infrared light and a special infrared light camera is able to compute the depth information from the image. This information can be used to interact with computer like never before. You drive a car or play almost everything what you want and with this camera the computer is capable of creating a model of human figure in real-time. This opens new possibilities of interaction with the computer.

We believe that in several cases using hands to interact with the computer will be much more efficient than old-fashioned mouse or keyboard. We want to make this solution accessible for larger population by using a web-cam. In order to deal with highly articulated objects, such as hands, effectively in the most common situations with arbitrary background, several requirements were given:

- background invariant,
- without gloves or any other markers,
- light invariant,
- ability to track both hands of the user real-time,
- hand shape (pose) invariant,
- hand size invariant.

In Section 2, we describe the present state of art in the hand tracking area. The algorithm *Flocks of features* is described more precisely in Section 3. In Section 4, we present our modifications to this algorithm to overcome some of its difficulties. We discuss the results of different methods in Section 5. Finally, in the last section, we propose the conclusion and the future work.

2 Related Work

There are several methods to track human hands. A tracker can be based on a few cues or their combinations; shape (contour), color, and behavioral knowledge about the hand. A lot of algorithms work fine, however there are lots of restrictions to the user's behavior or background. The aim of good interface is not to restrict users but to allow them maximum freedom and comfort. There are summarizing papers [12, 10] which bring brief informations on several techniques used for hand detection and hand tracking.

*fogelton@gmail.com
†makula@fiit.stuba.sk
[1] http://www.nintendo.com/wii
[2] http://www.xbox.com/en-US/kinect

2.1 Color Based Tracking

One of the first ideas to detect human hand is to use the color based filter. Every human hand has a specific color. The basic principle is to use a set of threshold ranges for every image channel separately. This solution is limited to several conditions. There should be no other objects having the same skin color characteristics in the captured image, lighting conditions have to be constant and the person has to be of the specific skin color. For example black people do not have the same range of thresholds like white people. These restrictions make this solution not very useful, but there is a number of possibilities to improve it.

Color Models

The most widely used is the RGB color model, in which every color is described by the intensity of three basic colors: Red, Green and Blue. High correlation between these components and luminance mixing with the chromaticity makes this color model very sensitive to the light condition changes [3]. There are several other models, which have the intensity of colors in a separate channel. Most common are HSV, HSL, YCbCr or normalized RGB.

Simple Color Classification using Randomized Lists

An interesting solution was presented in the paper *Robust hand tracking using a simple color classification technique*, which uses even different color model called *L*a*b** [19]. It is also quite invariant to luminance conditions with separately defined luminance value *L** and chromatic values *a** and *b**. Threshold ranges are setup with sample pixels from a hand, but this is not the only classification method used. This solution is based on clustering similar color pixels and defining a region of interest during the initialization process (in our case a human hand). This solution presents very good tracking results (Figure 1) of hand under various luminance conditions and with almost no false positive cases. However, we believe that this solution will stop working when the hand comes in contact with face and the classifying method would fail.

Figure 1: Hand clusterization [19].

Mean-Shift and CamShift

Mean-Shift is a robust color segmentation method [1] based on selected region matching. It is converging from an initial guess for location and scaling to the best match based on the color histogram probability. CamShift detects the mode in the probability distribution by applying Mean-Shift and dynamically adjusting the parameters of the target distribution. Mostly it is working fine but with very bad size precision and objects with similar color can easily distract the tracker. It is the standard, the results can be compared to.

Modified CamShift [18] takes into consideration a kind of gray model, which also represent the forecast of change in the image sequence. Better results are achieved because of this modification and there is no distraction due to the contact with face (Figure 2).

Figure 2: No distraction with face using Gray CamShift tracking [18].

2.2 Background Subtraction

This technique is used mostly in [11] with fingertip detection. It can be used when the background is given or stable without any other motion (hand only). This restriction makes it applicable only to a few situations.

2.3 Contour Based

Another cue on which we can base tracking is the contour. There are several solutions that use specific shape of a hand posture for detection and tracking. The basic technique to enhance image is the *Canny* edge detector. A more sophisticated one is the Oriented edge energy (Figure 3), which is the result of several filters [15]. It is hard to define all possible contours of a human hand and that is why contours are mostly used for posture estimation on selected region.

Figure 3: Original image, Canny detector, Oriented edge energy [15].

Condensation

The condensation approach [5] models the probability distribution with a set of random particles and performs all involved calculations on this particle set. Condensation presents very good results with proposed hand posture in Figure 4, but it is not designed to track human hand while posture changes a lot.

Figure 4: Sample of condensation tracking [5].

2.4 Model Based Detection

This principle is always used with some other cues, where we can use knowledge about the hand. For example in [2] it is used with skin color probability map to detect the hand and its posture, which depends on how many fingers are straight. This kind of model can be useful when the hand is pointed to the camera and the number of straight fingers can be clearly seen.

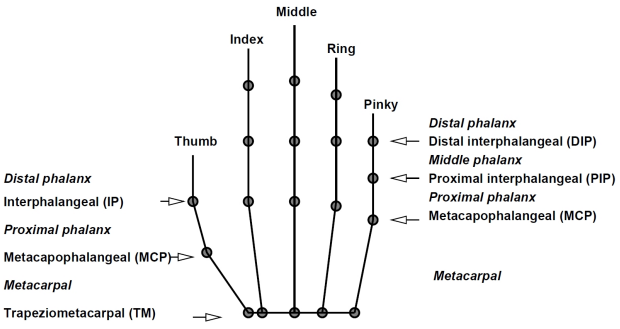

Figure 5: One of the skeleton hand models [12].

There are several types of hand models [12]. The skeleton model (Figure 5) is very common. They usually differ in number of points and vectors. The motion model [15] is another option, which can be used as another cue to extend the color based tracking.

3 Flocks of Features

Mathias Kölsch and Matthew Turk presented a *Fast 2D hand tracking with flocks of features and multi-cue integration* [7]. This algorithm can track the human hand without any artificial objects such as gloves. It is robust to various light conditions and furthermore a non stationary camera can be used. The tracker's core idea is motivated by the seemingly chaotic flight behavior of a flock of birds [13] such as pigeons. The minimum and maximum safe distance during the flight are defined. Features of the hand are also very close together like birds in a cloud [13]. The minimum distance between any two features and the maximum distance from the center (median) is defined. The median position of features is computed and the search using optical flow can be provide only up to the maximum distance from this position.

Listing 1: Flocks of features algorithm [8].

```
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
//k=3 was used

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:
median - the average feature location
```

Robust hand detection [8] is used to initialize this method. Very good results are achieved during rapid movements and with continuous pose changing of the human hand. An overview of the entire algorithm is listed in Listing 1.

3.1 KLT Features

The KLT tracking algorithm calculates a brightness gradient (sobel operator) along at least two directions for a promising feature candidate to be tracked over time [14, 16]. In combination with image pyramids (a series of progressively smaller-resolution interpolations of the original image), a feature's image area can be matched efficiently to the most similar area within a search window in the following video frame. If the feature match correlation between two consecutive frames is below a threshold, the feature is considered "lost". A hand detection method supplies both a rectangular bounding box and a probability distribution to initialize tracking.

The probability mask states for every pixel in the bounding box the likelihood that it belongs to the hand. Features are selected within the bounding box according to their

ranking and observing a pair wise minimum distance. These features are being ranked according to the combined probability of their locations and color. Highly ranked features are tracked individually per frames. Their new locations become the area with the highest match correlation between the two frame's areas.

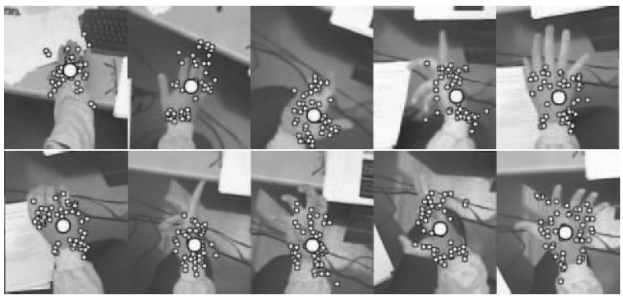

Figure 6: Snapshots of sequences with hand motions; the cloud of little dots are features and the big dot is their median [7].

Individual features can latch onto arbitrary artifacts of the object being tracked, such as fingers of a hand. Their movement is independent along with the artifact, without disturbing other features. Too dense concentrations of the features that would ignore other object's parts are avoided due to the minimum distance constraint. But stray features that are too far from the object of interest are brought back into the flock with the maximum distance constraint. To get more stable results, about 15% of the furthest features from median computation have to be removed. The speed of pyramid-based KLT (Kanade, Lucas, Tomasi) [14, 16] feature tracking allows to overcome the computational limitations of tracking the model-based approaches and achieving real-time performance.

3.2 Color Classification

During calibration process, a hand color is observed and the normalized-RGB histogram is created. Using this technique exclusively is not a very good solution because it can detect objects with similar color histogram such as wooden objects or other parts of the human body. The color information is used as a probability map. At tracker initialization time, the KLT features are placed preferably onto locations with high skin color probability. New location of a relocated feature is chosen with high color probability (more than 50%). Changing light condition can cause bad tracking performance, but only in case of relocated features because most of the features will continue to follow gray-level artifacts. This method combines cues from feature movement based on gray-level image texture with cues from texture-less skin color probability. It depends on the algorithm parameters how often features are relocated and on the importance of the color modality.

This algorithm was used to interact (Figure 7) with a wearable computer [9]. A webcam was placed at the head mounted display, so the hand size was approximately constant. It can be used to track both hands [4] or even other objects, where the skin color is replaced by a given sample. The problem is that it is not size invariant due to the threshold for the maximum distance from the center of the flock.

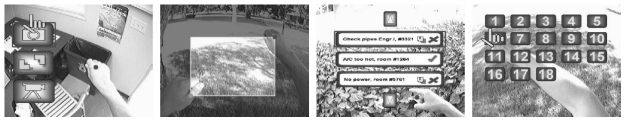

Figure 7: Screen shots from glasses of given application [9].

3.3 Flocks of Features with Appearance

New version of FoF was introduced, where another cue was added [6]. Haar features detector [17] was trained on a set of images of hands with different pose variations. This is used side by side with FoF. Positive detection of hand is evaluated after passing several stages (Figure 8). This adapted algorithm uses probability given not only by skin color, but also by the detector (the last passed stage of AdaBoost). The presented solution reaches better results than the original flocks of features in all tested cases [6].

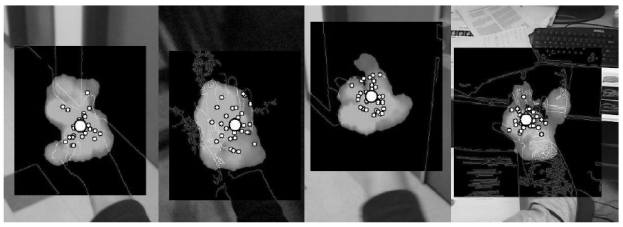

Figure 8: The appearance-based prior for selected hand images [6].

4 Modifications of Flocks of Features

The original FoF uses gray-level image for KLT features tracking. Due to this procedure we found the FoF algorithm to be vulnerable to edges occurring in the background. During movements over strong edges, a lot of KLT features can be relocated into incorrect positions. This leads to an incorrect median relocation and tracking failure (Figure 9).

We tried to avoid this kind of failure by ranking features based on skin color probability also during tracking and not only at the initialization procedure, but this did not lead to the expected results. We used histogram (calculated from a given region – hand palm) from the initialization procedure to create a probability map by applying

Figure 9: FoF algorithm is vulnerable to edges occurring in the background due to bad KLT features relocation.

back projection during tracking on every frame. One slight difference is use of the HSV color model instead of normalized-RGB, because it is more common in these kind of application while using histogram. The HSV color model and normalized-RGB have similar characteristics in terms of luminance invariance.

4.1 Image Processing

We realize that if we use the probability map instead of the original image, we will get rid of the edges in the background because they are not skin colored and they did not appear in the back projected image. The idea to run FoF on this probability map (Figure 10) has been proved to be a step forward. Look at the resulted image on Figure 10, there is a lot of noise. To reduce this noise we use basic image processing operations: erosion and dilation (Figure 11). From testing we can say that the best ratio *reducing noise/reducing skin regions* is achieved when applying erosion and then dilation, each one only once. This is also called the operation open.

Figure 10: Result of histogram back projection with noise.

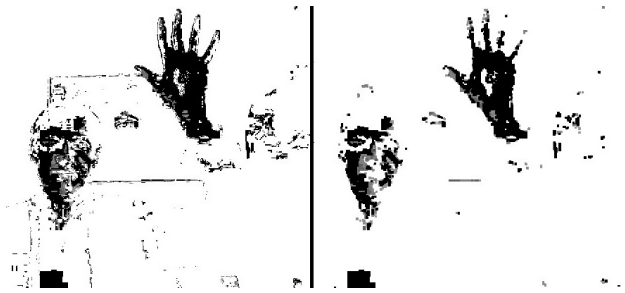

Figure 11: Removing noise with erosion and dilation.

Because of this modification we do not need to rank features in the initialization procedure and we can be almost sure that every feature will be located somewhere in the skin region. The modified algorithm is listed in Listing 2.

Listing 2: Modification of Flocks of features algorithm.

```
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
create back projected image
find n good-features-to-track with mindist

tracking:
create back projected image
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:
median - the average feature location
```

5 Results

We present a sequence of images with tracking results (Figure 12). We considered tracking to be lost when the median came out of the hand palm for more than one second. Our modification can also fail like the original one (Figure 13), mostly because of rapid movements of the hand over face or other skin colored objects. An ordinary webcam is able to achieve 30 frames per second, but this is not enough for rapid movements. The reason for the KLT tracking failure in this case is the optimization to look for a new location of a given feature in range of 10 pixels (the bigger the range, the more time it takes to compute). That is the reason why it considers other skin colored parts as hand and the median is disrupted.

Another case of tracking failure is when the median slides off the hand palm (Figure 14), because the given person is not wearing a shirt with long sleeves. This tracking is based mostly on color cue and this disruption is very common using this principle. The solution is to add another cue which would represent the hand contour; it should be some general contour with help of which we could determine the border between hand palm and forearm. This could be our task in the future.

The HSV color model provides luminance invariation, but there are situations when the histogram back projection is not working well. For example, when a human hand is moving too far from the camera, the luminance from the

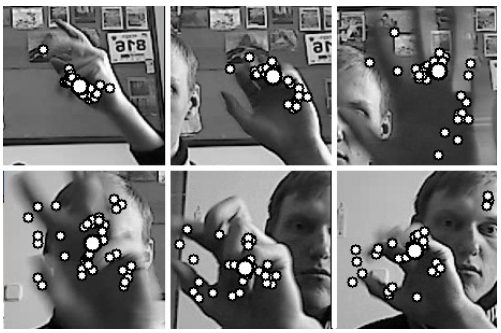

Figure 12: Example of hand tracking(200×200 image parts are cutted out from 640×480 images).

Figure 13: Tracking failure due to rapid movements.

hand changes rapidly and we can see only small parts of the hand (Figure 15) on the back projected image. This can also lead to tracking failure. This unpleasant effect can be also seen when the main source of light is not over head and luminance can change a lot by rotating the hand.

5.1 Comparison

In general it is hard to compare methods, as they have different advantages and disadvantages. We did some comparisons between CamShift, our implementation of FoF and our modification of FoF. We did not add new features when others where considered lost. Our aim is to objectively verify the ability of tracking various hand postures in a frame sequence while trying to disrupt the tracker by movements across face and the other hand.

We made 6 pairs of videos with different length from 400 to 1000 frames. Each pair consists of similar videos. One is with a man wearing long sleeves shirt ('a' labeled videos), the other wearing t-shirt ('b' labeled videos). Videos and tracking results can be downloaded from my website.[3] One pair was made outside, the others inside. They are aimed at different conditions like changing the size of the hand, rapid movements or moving background. Complete list of all tested videos can be found at Table 1. The results are processed in the graph (Figure 16), where the height of columns means the number of frames till tracking failure.

[3] http://henryi.yweb.sk

Figure 14: Median slided off the hand palm.

Figure 15: Example of bad back projection due to the luminance variation of the skin.

Conclusions and Future Work

We analyzed different hand tracking solutions with the aim to find a solution which could be used as an input to interact with computer. The main goal is not to restrict the user with behavior rules. It should be possible to track human hand without wearing any gloves or forcing the users to hold their hand in one posture all the time to get tired easily. Flocks of features showed up to be a very robust algorithm, but it can be easily disrupted with strong edges in the background. Our idea to process an image with histogram back projection led to a significant improvement in tracking efficiency and decreasing false positive KLT features tracking. But there are still several issues to handle.

We proofed, that running FoF on back projected image is better in almost all cases (Figure 16) than the original FoF and it can handle more reliable tracking than CamShift in all cases. In one case original FoF showed to be better due to bad luminance conditions (Back projection is not working well, when the luminance condition change a lot from initialization procedure.) and the gray-level image used for tracking is more suitable for these kind of

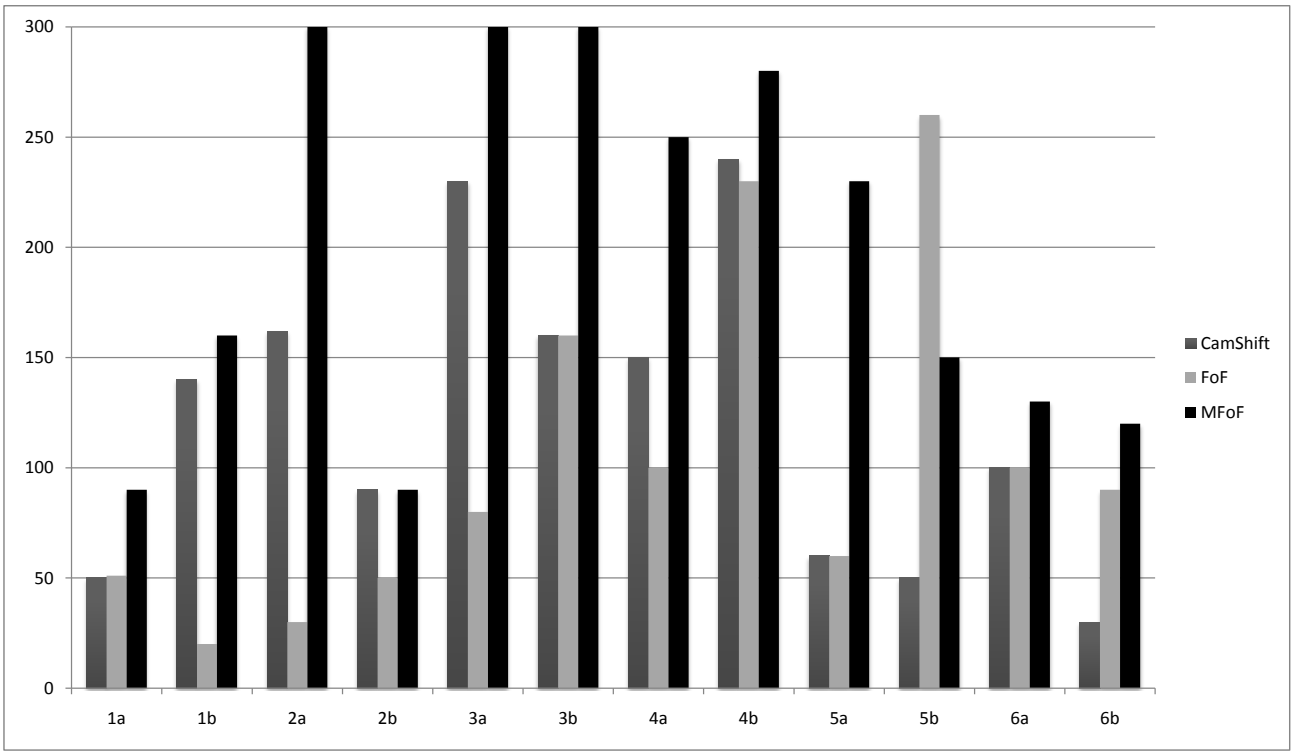

Figure 16: Testing results (Modified FoF was able to track the whole video sequence in cases: 2a, 3a, 3b).

number	conditions
1	rapid movements of the hand
2	size of the hand is changing a lot
3	arbitrary movements
4	arbitrary movements
5	moving background
6	outside lightining conditions

Table 1: Explanation of listed videos

situations.

Size Invariant Flocks of Features

Due to the maximum distance threshold constant, flocks of features has a problem with moving the hand closer and further from the camera. This can be avoided by finding proper algorithm which would calculate this threshold depending on the density of features.

Contour Cue

We want to add a third cue to the flocks of features which would represent the contour of the hand palm. This will help us to avoid the unpleasant situation when features are detected outside the palm and the median is distracted and moved outside the palm of the hand (our aim is to keep the median inside the hand palm).

Tracking Lost Detection

After adding the contour cue we can setup tracking failure detection, which will be based on the median coordinates. Tracking could be considered lost when the median is moved outside the hand palm contour.

Acknowledgements

This work was supported by grant KEGA 244-022STU-4/2010.

References

[1] Dr. Gary Rost Bradski and Adrian Kaehler. *Learning opencv, 1st edition*. O'Reilly Media, Inc., 2008.

[2] Lars Bretzner, Ivan Laptev, and Tony Lindeberg. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, FGR '02, Washington, DC, USA, 2002. IEEE Computer Society.

[3] Reza Hassanpour, Asadollah Shahbahrami, and Stephan Wong. Adaptive gaussian mixture model for skin color segmentation. *World Academy of Science, Engineering and Technology*, 41, 2008.

[4] Jesse Hoey. Tracking using flocks of features, with application to assisted handwashing. In *British Machine Vision Conference (BMVC)*, 2006.

[5] Michael Isard and Andrew Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998.

[6] Mathias Kölsch. An appearance-based prior for hand tracking. In Jacques Blanc-Talon, Don Bone, Wilfried Philips, Dan Popescu, and Paul Scheunders, editors, *Advanced Concepts for Intelligent Vision Systems*, volume 6475 of *Lecture Notes in Computer Science*, pages 292–303. Springer Berlin / Heidelberg, 2010.

[7] Mathias Kölsch and Matthew Turk. Fast 2d hand tracking with flocks of features and multi-cue integration. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on*, pages 158 – 158, 27-02 2004.

[8] Mathias Kölsch and Matthew Turk. Robust hand detection. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 614 – 619, 2004.

[9] Mathias Kölsch, Matthew Turk, and T. Hollerer. Vision-based interfaces for mobility. In *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, pages 86 – 94, august 2004.

[10] Fariborz Mahmoudi and Mehdi Parviz. Visual hand tracking algorithms. In *Proceedings of the conference on Geometric Modeling and Imaging: New Trends*, pages 228–232, Washington, DC, USA, 2006. IEEE Computer Society.

[11] Shahzad Malik and Joe Laszlo. Visual touchpad: a two-handed gestural input device. In *Proceedings of the 6th international conference on Multimodal interfaces*, ICMI '04, pages 289–296, New York, NY, USA, 2004. ACM.

[12] V.I. Pavlovic, R. Sharma, and T.S. Huang. Visual interpretation of hand gestures for human-computer interaction: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):677 –695, jul 1997.

[13] Craig W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21(4):25–34, July 1987.

[14] Jianbo Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593 –600, 21-23 1994.

[15] Björn Stenger. Template-based hand pose recognition using multiple cues. In P. Narayanan, Shree Nayar, and Heung-Yeung Shum, editors, *Computer Vision ACCV 2006*, volume 3852 of *Lecture Notes in Computer Science*, pages 551–560. Springer Berlin / Heidelberg, 2006.

[16] Carlo Tomasi and T Kanade. Detection and tracking of point features. *Image Rochester NY*, pages Technical Report CMU–CS–91–132, April 1991.

[17] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511 – I–518 vol.1, 2001.

[18] Jiajun Wen and Yinwei Zhan. Vision-based two hand detection and tracking. In *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, ICIS '09, pages 1253–1258, New York, NY, USA, 2009. ACM.

[19] Miaolong Yuan, Farzam Farbiz, Corey Mason Manders, and Ka Yin Tang. Robust hand tracking using a simple color classification technique. In *Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, VRCAI '08, pages 6:1–6:5, New York, NY, USA, 2008. ACM.