**The Investigation of Different Filters Based on Gesture Recognition System**

Chen Du, Wenjun Zhang, Jingyi Yang, Jinzhao Feng

**Abstract**

1. **Introduction**
2. **Gesture Recognition System**

We built our gesture recognition system which could recognize number gesture from one to five and show the corresponding numbers. This system is based on Matlab, which is one of the most popular computer vision platforms. The block diagram of the system is shown in Fig.1.

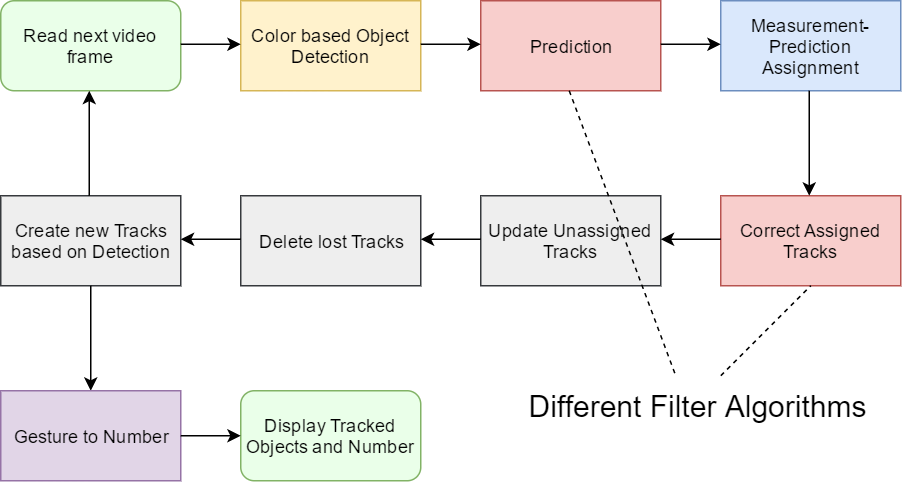


Fig. 1 Diagram of the gesture recognition system

This system could be separate into 3 parts: detection, tracking and recognition.

**2.1 Object Detection**

We have tested 3 detection methods including correlation bases detection [1], foreground detection [2] and color-based detection [3]. Since the correlation-based object detection needs a specific template for each object and is unstable when object or lens rotates, it appeared low precision for finger detection where lots of rotations and random motions happen. Similarly, the foreground detection has a demand for fixed camera, and is hard to tell the fingers from non-target moving objects interference. Color-based detection is widely used in scenes where the objects have specific color, and has the advantage of robustness for moving object detection compared with other two methods. Using a glove with specific colored fingers, the color-based detection showed high precision and robustness. So we choose the color-based object detection to capture the fingers and hand locations. In detail, in each frame, the RGB value of each pixel <rx, gx, bx> is compared with the target RGB value<rt, gt, bt>, and the Euclidean norm of the difference is calculated as the distance

dist\_x = ||rx-rt|| + ||gx-gt|| + ||bx-bt|| (所有x, t作下标) – (1)

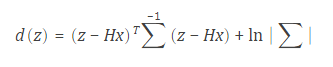
If the distance is within a given threshold, this pixel is judged as target (foreground).

--(2)

A mask of target is formed after all pixels have been judged. A blob analysis is performed on the mask and the finger-like connected components within given size will be output as the fingers and the centers are calculated as locations. Before blob analysis, we remove small connected components within a given area threshold (e.g. 100 pixels) to prevent false positive, then a morphologic closing is applied to remove the false negative holes caused by noise.

**2.2 Tracking**

We applied 4 kinds of filters which will be discussed in **section 3**. For multiple object tracking, one challenge is to correctly assign the predicted locations with the detected ones. Here we use the state-space distance [4] to compute the confidence.

 --(3)

 --(4)

where z is the measurement and d(z) is the corresponding state-space distance, x is the predicted state, P is the state estimation error covariance, H is the measurement model, these metrics will be discussed in **section 3**. For each predicted state, we assign it to the measurement with lowest distance. For Kalman filters the assigned trackers will be corrected to get the tracked locations, while the unassigned trackers will just use predicted states as the tracked locations. For trackers which are unassigned for a certain number of continuous frames, we regard them as lost objects and delete them. For unassigned measurements new trackers will be created once they show up longer than a certain number of continuous frames.

**2.3 Convert gesture to number**

Once we have the tracked locations of the fingers and hand, we establish a 3x3 grid block based on hand location and count the finger numbers in each block, then output the corresponding meaning of the gesture, here the output contains numbers from 1-5. The connection between number count and the output is set manually, and is adjustable to add other meanings of gestures.

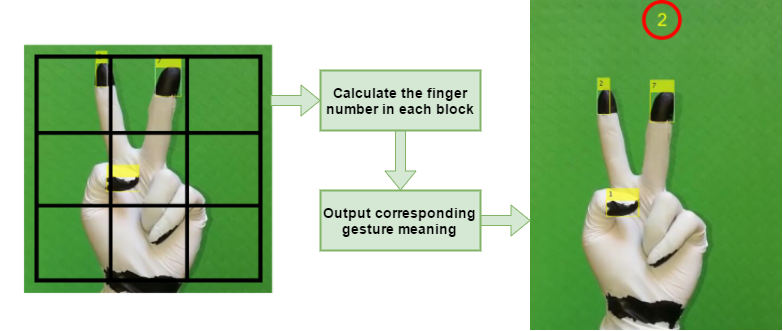


Fig. 2 Gesture-to-number conversion

1. **Filters Theories**
2. **Experimental Design and Result analysis**

To investigate the performances of different filters and confirm the hypothesis, we use the detection and tracking parts of our system to do a quantitative evaluation experiment, which aims to compare the tracking using different filters in various scenarios.

**4.1 Scenarios Selection**

Considering factors which could affect the tracking performances, we design and record videos in scenarios shown below:

1) Linear motion model: the objects perform simple linear motion, e.g. straight line motion. Note there is no perfect linear motion in real life, so the linear motion is actually approximately linear motion. Despite small vibration or disturbance, the movement in a short period of continuous frame could be regarded as linear.

2) Nonlinear motion model: the objects perform mostly random motion, e.g. hand doing random gestures. It’s hard or impossible to manually calculate the state transition model in this case.

3) Occluded objects: objects are usually occluded in real life videos. A robust tracker should be able to handle the occlusion. In this kind of videos, objects are occluded partially or entirely in some frames, the occluded frame ratio varies from 10% to 50%.

4) Low-light condition: light is a very important factor that could influence the detection results. In this condition, objects moves in low-light environment which makes it hard to tell the target color for each pixel.

By choosing one of 2 motion models and one of 3 interferences (no interference is one choice), we have 6 combinations of scenarios.

**4.2 Evaluation metric**

Many metrics for quantitative evaluation of target tracking have been proposed by researchers [5][6][7]. Since we focus on the tracking performance comparison, the precision plot which is used in recent year tracking benchmarks [8][9] is a suitable choice. The precision of a tracker means the percentage of frames whose estimated location is within the given threshold distance of the ground truth [8]. We are able to see the tracking performances with different error tolerances from precision plot directly.

**4.3 Parameters**

The system parameters are shown in Table. 1, for different trackers and videos, the noise value is selected based on the scenario. When the detector works well, i.e. there is enough light and no occlusion, the measurement noise is set to low value, while in low-light or occluded conditions, the measurement noise is set to high value. The motion noise is based on the goodness-of-fit of transition matrix/function to the motion model, for Kalman filter and Particle filter, motion noise is set to high for complex motion, and low for simple motion.

Table 1. System paramters

|  |  |
| --- | --- |
| **System Parameter** | **Value** |
| Motion Noise | 10~1000 |
| Measurement Noise | 10~10000 |
| Video(Frame) Size | 486x864 |
| Frame per second | 12 |

The filter parameters are shown in Table 2. As has been discussed, the performance of three kinds of Kalman filters depends on the correct transition matrix/function, but for most non-linear motion, there is no general model to calculate transition function for EKF and UKF. To estimate the transition function of the hidden motion model, we assume the state of the target is approximately an autoregressive process

Using labelled ground truth, a feedforward neural network is applied to get the estimation of the transition function *f*(parameterized by w). The residual error after convergence was taken to be the motion noise.

Table 2. Filter parameters

|  |  |
| --- | --- |
| **Filters** | **Parameters** |
| Kalman Filter | Constant Acceleration Model |
| Extened Kalman Filter | f approximated by single-layer FNN |
| Unscented Kalman Filter | f approximated by single-layer FNN  = 0.001, , |
| Particle Filter-50 | Particle Number = 50 |
| Particle Filter-500 | Particle Number = 500 |
| Particle Filter-5000 | Particle Number = 5000 |

We apply particle filters with 3 amounts of particles with linear transition function.

**4.4 Results analysis**

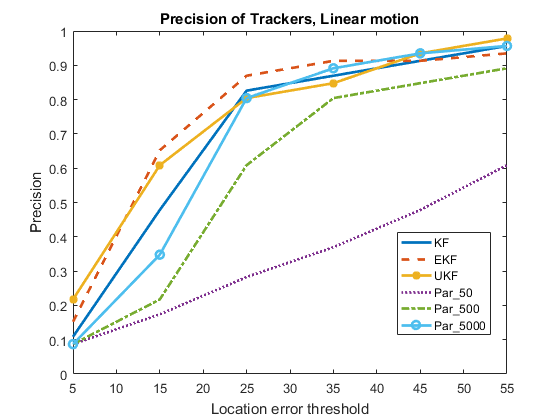


Fig. 3 Precision of Trackers in Linear motion scenario

Fig. 3 shows the precision of trackers in linear motion scenario. All three kinds of Kalman filters have good performances. The little difference among them is because the motion model is not perfect linear, and EKF and UKF could better capture the little vibration, when the threshold become higher, there is no obvious difference among KF, EKF and UKF. Particle filters with 5000 and 500 particles has also good and acceptable performances. But particle filters with 50 particles has relatively poor precision because it’s hard to calculate the correct location using such few number of particles, even little noise would influence the result.

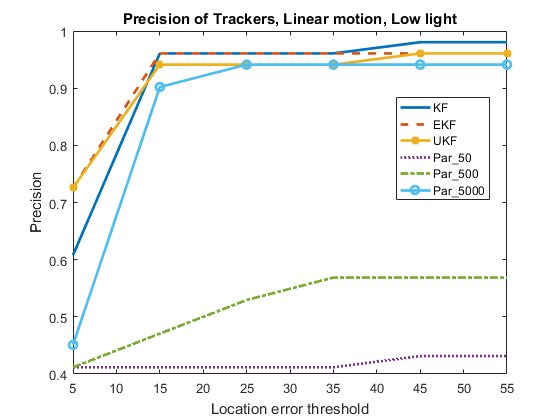


Fig. 4 precision of Trackers in Linear motion and Low Light scenario

Fig. 4 shows the results in low light scenario. We can see the 3 Kalman filters and particle filter with 5000 particles are still robust with good precision. But the particle filters with 500 particles and 50 particles have poor performances. This is because in low light condition, it’s more difficult to tell the accurate difference of color, i.e. the particle filter without enough particles have low accuracy when calculating the location.

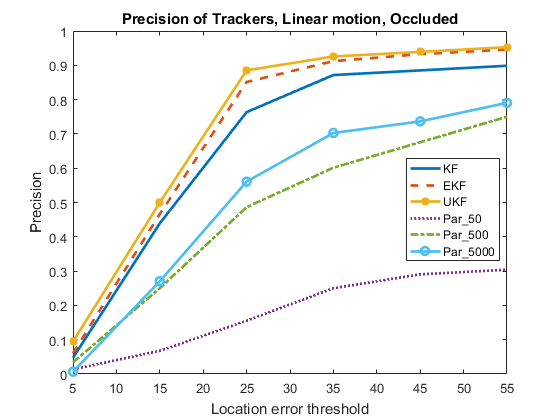


Fig. 5 precision of Trackers in Linear motion and Occluded scenario

From Fig. 5 we can see that UKF and EKF outperform KF as a result of more accurate prediction in occluded frames. While the 3 particle filters have relatively low precision. This is because when object is occluded particle filter will try to find the most similar region to target in background, i.e. the result is dominated by noise.

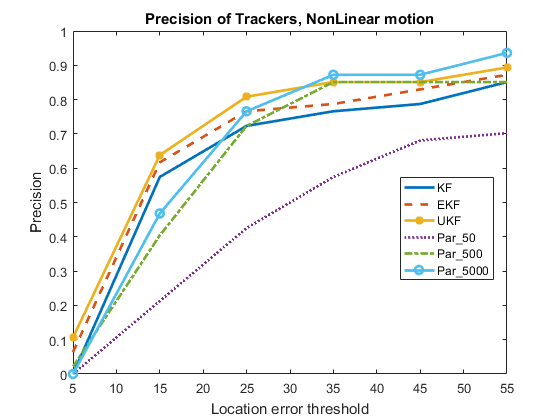


Fig. 6 precision of Trackers in Nonlinear motion scenario

Fig. 6 shows the results in nonlinear motion scenario, we can see clearly UKF > EKF > KF because UKF has the most accurate approximation to the nonlinear model, confirming the hypothesis. While particle filters with 5000 and 500 particles both has good performances.

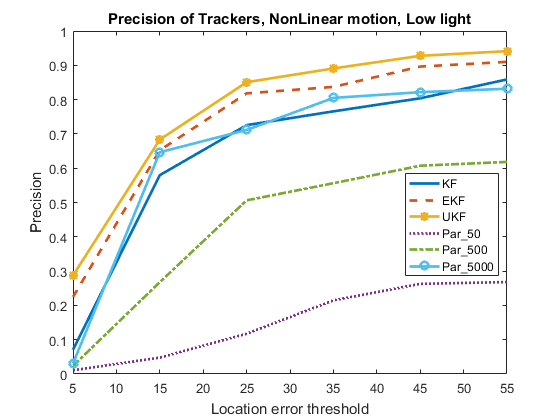


Fig. 7 precision of Trackers in Nonlinear motion and Low light scenario

Fig. 7 shows the results in nonlinear motion and low light scenario. Compared with Fig. 6, we can see three Kalman filters are still robust while UKF > EKF > KF. Particle filter with 5000 particles is still of good precision, even though influenced by low light inference. But the particle filters with 500 and 50 particles have low precision because the low light interference.

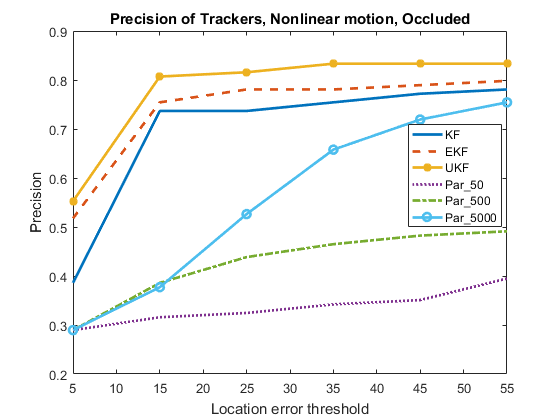


Fig. 8 precision of Trackers in Nonlinear motion and Occluded scenario

From fig. 8, we can see in occluded nonlinear motion scenario, it’s obvious that Kalman filters are more robust than particle filters, while UKF > EKF > KF. Particle filters again have low precision because of occlusion.

1. **Conclusion**

For linear motion scenario, all three Kalman filters perform well; for nonlinear motion, UKF > EFK >KF corresponding to the different precisions of approximation to the model. And all three kinds of Kalman filters have good stability in low light and occluded conditions. Particle filter needs sufficient amount of particles to ensure robustness, and it has good performance for both linear and nonlinear motion model. But once in low light and occluded conditions, the performance of particle filter declines sharply. For future work, there are many ways to improve the track precision. For example, using RGB-D camera, it’s possible to reconstruct the 3-D model of target and get better tracking results [10].

**Reference**

[1]

@article{comaniciu2003kernel,

title={Kernel-based object tracking},

author={Comaniciu, Dorin and Ramesh, Visvanathan and Meer, Peter},

journal={IEEE Transactions on pattern analysis and machine intelligence},

volume={25},

number={5},

pages={564--577},

year={2003},

publisher={IEEE}

}

[2]

@mastersthesis{gallego2009foreground,

title={Foreground segmentation and tracking based on foreground and background modeling techniques},

author={Gallego Vila, Jaime},

year={2009},

school={Universitat Polit{\`e}cnica de Catalunya}

}

[3]

@inproceedings{khan2001object,

title={Object based segmentation of video using color, motion and spatial information},

author={Khan, Sohaib and Shah, Mubarak},

booktitle={Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on},

volume={2},

pages={II--II},

year={2001},

organization={IEEE}

}

[4]

@article{welch1995introduction,

title={An introduction to the Kalman filter},

author={Welch, Greg and Bishop, Gary},

year={1995}

}

[5]

@article{bernardin2008evaluating,

title={Evaluating multiple object tracking performance: the CLEAR MOT metrics},

author={Bernardin, Keni and Stiefelhagen, Rainer},

journal={EURASIP Journal on Image and Video Processing},

volume={2008},

number={1},

pages={1--10},

year={2008},

publisher={Springer}

}

[6]

@inproceedings{li2009learning,

title={Learning to associate: Hybridboosted multi-target tracker for crowded scene},

author={Li, Yuan and Huang, Chang and Nevatia, Ram},

booktitle={Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on},

pages={2953--2960},

year={2009},

organization={IEEE}

}

[7]

@article{babenko2011robust,

title={Robust object tracking with online multiple instance learning},

author={Babenko, Boris and Yang, Ming-Hsuan and Belongie, Serge},

journal={IEEE transactions on pattern analysis and machine intelligence},

volume={33},

number={8},

pages={1619--1632},

year={2011},

publisher={IEEE}

}

[8]

@inproceedings{wu2013online,

title={Online object tracking: A benchmark},

author={Wu, Yi and Lim, Jongwoo and Yang, Ming-Hsuan},

booktitle={Proceedings of the IEEE conference on computer vision and pattern recognition},

pages={2411--2418},

year={2013}

}

[9]

@article{wu2015object,

title={Object tracking benchmark},

author={Wu, Yi and Lim, Jongwoo and Yang, Ming-Hsuan},

journal={IEEE Transactions on Pattern Analysis and Machine Intelligence},

volume={37},

number={9},

pages={1834--1848},

year={2015},

publisher={IEEE}

}

[10]

@article{takimoto20163d,

title={3D reconstruction and multiple point cloud registration using a low precision RGB-D sensor},

author={Takimoto, Rog{\'e}rio Yugo and Tsuzuki, Marcos de Sales Guerra and Vogelaar, Renato and de Castro Martins, Thiago and Sato, Andr{\'e} Kubagawa and Iwao, Yuma and Gotoh, Toshiyuki and Kagei, Seiichiro},

journal={Mechatronics},

volume={35},

pages={11--22},

year={2016},

publisher={Elsevier}

}