

REPORT DOCUMENTATION PAGE			Form Approved OMB NO. 0704-0188		
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA, 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 27-07-2008		2. REPORT TYPE Final Report		3. DATES COVERED (From - To) 1-Sep-2007 - 31-May-2008	
4. TITLE AND SUBTITLE A New Particle Filter Based Algorithm for Image Tracking			5a. CONTRACT NUMBER W911NF-07-1-0503		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER 611102		
6. AUTHORS Scott Acton, Xinmin Liu, Zongli Lin			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAMES AND ADDRESSES University of Virginia Office of Sponsored Programs 1001 N. Emmett St. P.O. Box 400195 Charlottesville, VA 22904 -4195			8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211			10. SPONSOR/MONITOR'S ACRONYM(S) ARO		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S) 53334-CI-II.1		
12. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited					
13. SUPPLEMENTARY NOTES The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.					
14. ABSTRACT We have developed a novel algorithm for tracking an object, such as the UAVs, in a sequence of images. The development of such an algorithm is motivated by the idea behind the particle filter and the concept of feedback from control theory. We will thus temporarily refer to this algorithm as a particle filter with feedback (PFF) algorithm for image tracking. We will briefly describe our algorithm and compare its performance with some existing algorithms. This comparison indicates that the proposed tracking algorithm drastically outperforms the existing methods both in terms of tracking accuracy, robustness and tracking speed.					
15. SUBJECT TERMS image tracking, particle filter, control theory					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT SAR	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Zongli Lin
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER 434-924-6342

Report Title

A New Particle Filter Based Algorithm for Image Tracking

ABSTRACT

We have developed a novel algorithm for tracking an object, such as the UAVs, in a sequence of images. The development of such an algorithm is motivated by the idea behind the particle filter and the concept of feedback from control theory. We will thus temporarily refer to this algorithm as a particle filter with feedback (PFF) algorithm for image tracking. We will briefly describe our algorithm and compare its performance with some existing algorithms. This comparison indicates that the proposed tracking algorithm drastically outperforms the existing methods both in terms of tracking accuracy, robustness and tracking speed.

List of papers submitted or published that acknowledge ARO support during this reporting period. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Number of Papers published in peer-reviewed journals: 0.00

(b) Papers published in non-peer-reviewed journals or in conference proceedings (N/A for none)

Number of Papers published in non peer-reviewed journals: 0.00

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts): 0

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts): 0

(d) Manuscripts

Number of Manuscripts: 0.00

Number of Inventions:

Graduate Students

<u>NAME</u>	<u>PERCENT SUPPORTED</u>
Xinmin Liu	0.50
Alla Aksel	0.25
FTE Equivalent:	0.75
Total Number:	2

Names of Post Doctorates

<u>NAME</u>	<u>PERCENT SUPPORTED</u>
-------------	--------------------------

FTE Equivalent:

Total Number:

Names of Faculty Supported

<u>NAME</u>	<u>PERCENT SUPPORTED</u>	National Academy Member
Zongli Lin	0.11	No
FTE Equivalent:	0.11	
Total Number:	1	

Names of Under Graduate students supported

<u>NAME</u>	<u>PERCENT SUPPORTED</u>
-------------	--------------------------

FTE Equivalent:

Total Number:

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period:	0.00
The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:.....	0.00
The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:.....	0.00
Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):.....	0.00
Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:	0.00
The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense	0.00
The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:	0.00

Names of Personnel receiving masters degrees

<u>NAME</u>

Total Number:

Names of personnel receiving PHDs

<u>NAME</u>

Total Number:

Names of other research staff

NAME

PERCENT SUPPORTED

FTE Equivalent:

Total Number:

Sub Contractors (DD882)

Inventions (DD882)

A New Particle Filter Based Algorithm for Image Tracking

(Technical Final Report Submitted to Dr. Liyi Dai)

Xinmin Liu (Graduate Student), Zongli Lin (PI) and Scott Acton (Co-PI)

Charles L. Brown Department of Electrical and Computer Engineering

University of Virginia

Charlottesville, VA 22904-4743

1. Summary of the Report

We have developed a novel algorithm for tracking an object, such as the UAVs shown in Fig. 1, in a sequence of images. The development of such an algorithm is motivated by the idea behind the particle filter and the concept of feedback from control theory. We will thus temporarily refer to this algorithm as a particle filter with feedback (PFF) algorithm for image tracking. We will briefly describe our algorithm and compare its performance with some existing algorithms. This comparison indicates that the proposed tracking algorithm drastically outperforms the existing methods both in terms of tracking accuracy, robustness and tracking speed.

2. Existing Video Trackers

The GVF snake tracker was developed by Ray et al. (2002). It captures the object to be tracked through minimizing an energy function, defined on the basis of internal energy, external energy, shape, size, position, and sampling of the contour. Under most circumstances, the snake tracker is able to successfully track a rolling object.

The Monte Carlo tracker was developed by Cui et al. (2006). Based on the object movement information and the image intensity features, a specialized sample-weighting criterion is tailored to rolling object observed in vivo. In comparison with the snake-based trackers, as the noise intensity level increases, the performance of a snake tracker degrades more than that of the Monte Carlo tracker. More details on such a comparison can be found in Cui et al. (2006).



Fig. 1. Typical UAV images that represent typical targets in our research.

3. The Proposed Tracking Algorithm

The development of the algorithm was motivated by the idea behind the particle filter and the concept of feedback in control theory. We first predict the leukocyte position using the movement information of the previous steps. Samples of particles are then generated around the predicted position. Unlike in the Monte Carlo tracker where samples are generated randomly, here samples are generated by gridding an area around the predicted position. The number and the density of the samples are adjusted based on the previous movement information. At each of these sample points, radial edge detection is applied to determine if the point is within the target boundary. Weighted average among the positions of those sample points detected to be within the target boundary will be the filtered position of the center of the target at the current image frame. The weighting for a sample point is assigned according to a normal distribution with respect to its distance from the predicted position.

Various components of the algorithm are described in more detail as follows.

3.1. Sample Generation

Samples are generated around a predicted position of the target. We predict the target position using the movement information of previous steps. In the current stage, we still use the motion model of Cui et al. (2006). The target position is predicted by

$$\begin{aligned}\bar{x}_{c,t+1} &= \hat{x}_{c,t} + \alpha(\hat{x}_{c,t} - \hat{x}_{c,t-1}) + \beta(\hat{x}_{c,t-1} - \hat{x}_{c,t-2}) + (1 - \alpha - \beta)(\hat{x}_{c,t-2} - \hat{x}_{c,t-3}), \\ \bar{y}_{c,t+1} &= \hat{y}_{c,t},\end{aligned}$$

where $(\bar{x}_{c,t+1}, \bar{y}_{c,t+1})$ is the predicted target position in frame $t+1$, $(\hat{x}_{c,t}, \hat{y}_{c,t})$ is the estimated position in frame t , and α and β are non-negative constants. In the 2nd frame, when we don't have three previous frames, we will use the filtered position of the previous frame. Similarly, in the 3rd and 4th frames, we will use only the information of the previous 2 and 3 frames respectively.

We will generate the samples by gridding within an ellipsoid that is centered at the predicted position,

$$\frac{(x - \bar{x}_{c,t+1})^2}{a^2} + \frac{(y - \bar{y}_{c,t+1})^2}{b^2} \leq 1,$$

where $4 \leq a, b \leq 7$. The number and density of samples are adjusted by the previous measurement information. Shown in Fig. 2 is an illustration of a set of samples.

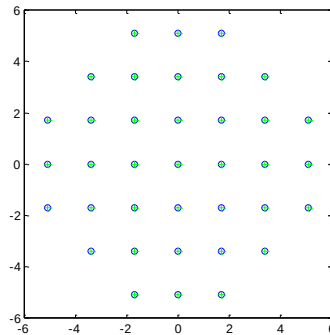


Fig. 2. A set of samples generated around the predicted position.

3.2. Image Intensity Measurement

Suppose (x, y) is the position of the target center. By performing radial edge detection around (x, y) , we can detect the target boundary. To do so, construct several line segments extending radially from (x, y) with coordinates $(l_{\theta,x}(k), l_{\theta,y}(k))$ (see Fig. 3),

$$l_{\theta,x}(k) = x + r(k) \cos \theta,$$

$$l_{\theta,y}(k) = y + r(k) \sin \theta, \quad \theta = \frac{k_1}{N} 2\pi, \quad k_1 = 0, 1, \dots, N_1 - 1,$$

$$r(k) = r_1 + \frac{k}{K} (r_2 - r_1), \quad r_1 < r_2, \quad k = 0, 1, 2, \dots, K,$$

where N is the number of line segments, $K+1$ is the number of points on each line, θ specifies the orientation of the line segment, and r_1 and r_2 are pre-specified values delimiting the length of the line segments.

In Cui et al (2006), where the following one-dimensional edge detection operator is applied on each line segment,

$$e_{\theta}(k) = |I'_{\theta}(k-2) + 2I'_{\theta}(k-1) - 2I'_{\theta}(k+1) - I'_{\theta}(k+2)|.$$

Here we use a modified edge detection operator on each line segment

$$e_{\theta}(k) = I'_{\theta}(k-2) + 2I'_{\theta}(k-1) - 2I'_{\theta}(k+1) - I'_{\theta}(k+2),$$

where $I'_{\theta}(k)$ is the image intensity at point $(l_{\theta,x}(k), l_{\theta,y}(k))$ obtained by bilinear interpolation. The corresponding coordinate, denoted as $(e_{\theta,x}, e_{\theta,y})$, with the maximum $e_{\theta}(k)$, is the detected edge point for the orientation θ for a bright target (or with the minimum $e_{\theta}(k)$ for a dark target). An example of the radial edge detection ($N = 8$) is illustrated in Fig. 3.

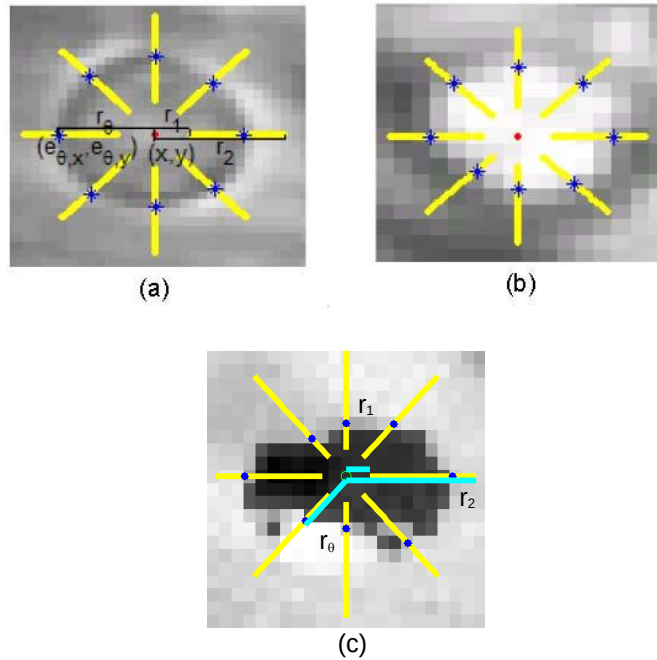


Fig. 3. (a) Detected dark leukocyte edge; (b) Detected bright leukocyte edge; (c) Detected vehicle edge.

3.3 Sample Weighting

To ensure a sample is in the inside of a bright target, all $e_\theta(k)$, $k=1,2,\dots,8$, should larger than a statically determined threshold. Considering the effect of the noise and clutter and weak image intensity features, if 7 of $e_\theta(k)$, $k=1,2,\dots,8$, are larger than a threshold, we will assume that the sample is inside the target. Similarly, if 7 of $e_\theta(k)$, $k=1,2,\dots,8$, are smaller than a threshold, we consider the sample is inside a dark target. If a sample is inside a target, the sample weighting is set to be a nonzero number, otherwise, it is set to be zero.

For a target with a nonzero weighting, we define $d_{t+1}^{(m)}$ to measure the displacement between $s_{t+1}^{(m)}$ and $(\bar{x}_{c,t+1}, \bar{y}_{c,t+1})$. The importance weight $\tilde{\pi}_{t+1}^{(m)}$ should be large when the displacement is small. In our algorithm, we define

$$\tilde{\pi}_{t+1}^{(m)} = e^{-\left(\frac{d_{t+1}^{(m)^2}}{2\sigma^2}\right)},$$

where $2 \leq \sigma \leq 5$. After normalization, the weighting we assign to the sample $s_{t+1}^{(m)}$ is

$$\pi_{t+1}^{(m)} = \frac{\tilde{\pi}_{t+1}^{(m)}}{\sum_{m'=1}^M \tilde{\pi}_{t+1}^{(m')}}.$$

3.4. Feedback in Image Processing

The concept of feedback is followed throughout our algorithm development. For example, the adjustment of the threshold in image intensity measurement and the determination of the density and number of samples are all based on the previous step data.

3.5. Some Observations

The success of tracking algorithm of Cui et al. (2006) is highly dependent on the accuracy of the predicted position of the target. If the target is a few pixels away from predicted position, the tracker will lose the target. The Cui et al. (2006) method also needs the target positions in the first two frames to initialize the process to make sure the predicted position is accurate.

Our proposed tracker can track a target even when it is many pixels away from the predicted position. Also, we need only the target position in the first frame to begin the tracking.

4. Comparison with Monte Carlo Tracker and GVF Snake Tracker

We will evaluate the performance our proposed particle filter with feedback tracker (PFF) with those of the Monte Carlo (MC) tracker and the GVF snake tracker on 30 sequences. Each sequence consists of 90 frames. This comparison will show the superior performance of the PFF tracker both in terms of tracking accuracy and tracking speed. The drastically reduced time required by the PFF tracker makes real-time tracking possible.

All simulations were carried out in Matlab 7.1.0.246 (R14) on PC with an Intel Core 2 CPU 2GHz and 1GB RAM. Each tracker is evaluated in the following three aspects.

- (1) Percentage of the frames tracked.
- (2) Number of sequences (out of 30 sequences) with all frames tracked

(3) Time taken to process each sequence.

Shown in Fig. 4 is the comparison of percentage of frames tracks. As seen in the figure, with registration, the PFF tracker tracks 16% more frames than the MC tracker, and 18% more frames than the GVF snake tracker. Without registration, the performance of the PFF tracker remains almost the same, while the performances of the MC tracker and the GVF snake tracker degrade drastically. This indicates that the PFF tracker is not only much more accurate but also much more robust.

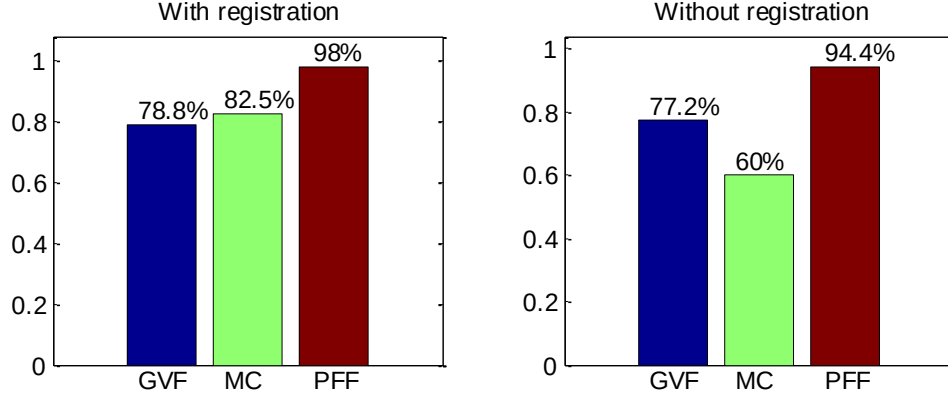


Fig. 4. Percentage of the frames tracked.

Shown in Fig. 5 is the number of sequences (out of 30 sequences) with all 90 frames tracked, both with and without registration. With registration, the PFF tracker is able to track all 90 frames in 23 out of 30 sequences, the MC tracker is able to track all 90 frames in 18 sequences and the GVF snake tracker is able to track all 90 frames only in 14 out of 30 sequences. Without registration, the number of sequences with all 90 frames tracked remains the same, while this number for the GVF snake tracker decreases to 1/3 and the number for the MC tracker decreases to 1/2.

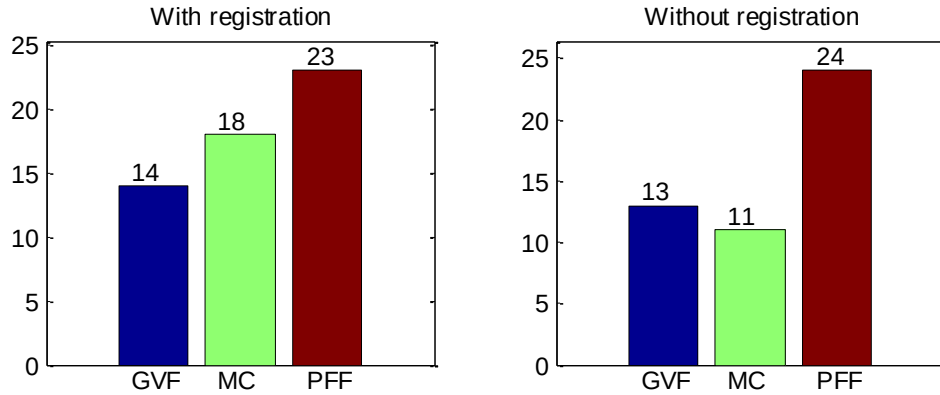


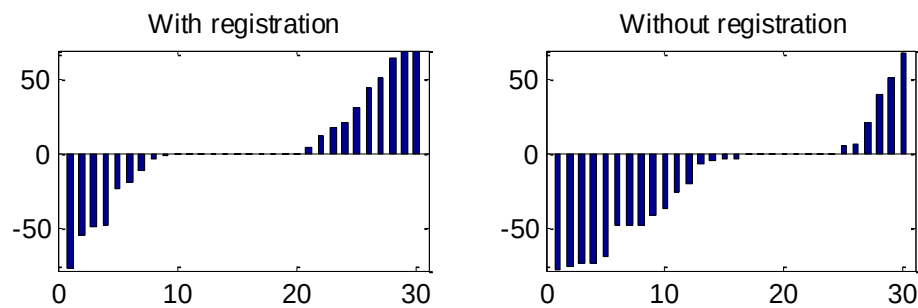
Fig. 5. Number of sequences (out of 30 sequences) with all 90 frames tracked.

Shown in Fig. 6 are the differences in numbers of frames tracked by the three different trackers for each of the 30 sequences. The top two graphs show that the MC tracker and GVF tracker outperform each other in about the same number of sequences and by about a same margin. The middle two plots show that the PFF tracker drastically outperforms the GVF snake tracker. In particular, with registration, in only two of the 30 sequences does the GVF snake tracker track a few more frames than the PFF tracker. On the other hand, there are 8 sequences in which the PFF tracker tracks over 40 more frames than the GVF snake tracker, and in another two sequences, the

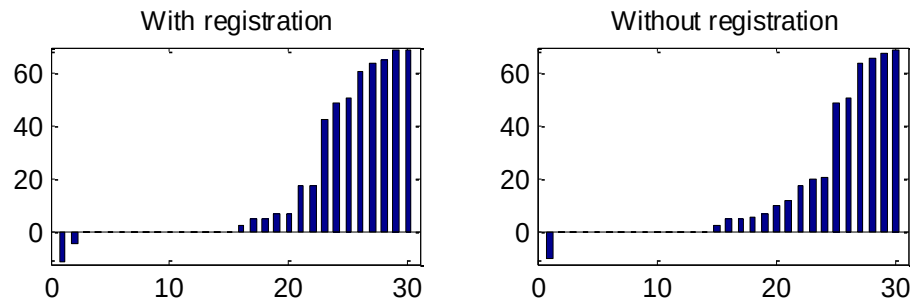
PFF tracker tracks 20 more frames than the GVF snake tracker. Without registration, the PFF tracker outperforms the MC tracker and the GVF snake tracker even more drastically. The bottom two plots show that the PFF tracker also outperforms the MC tracker by a similar margin as it outperforms the GVF snake tracker.

Shown in Fig. 7 is the average time required for computation in each sequence. In measuring these times, the times used to read data from the hard disk is not included. As seen in Fig. 7, the PFF tracker is about 34 times faster than the GVF snake tracker and over 56 times faster than the MC tracker.

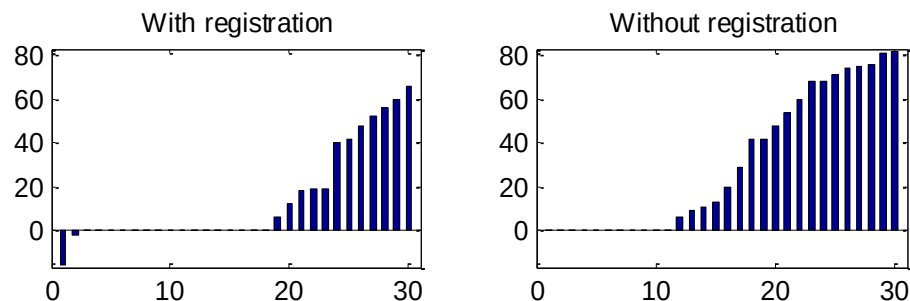
In order to compare in relatively fair manner, the main time-consuming part of the algorithms should all be written in a same programming language. We have chosen C. The main codes of both the MC tracker and the PFF tracker were written in C. The available GVF snake tracker is in the form of m-files (gvf.m and movesnake.m), which requires an average of 15.2683 seconds to track a sequence. The most time-consuming code in GVF snake tracker is the GVF algorithm. When we replaced gvf.m by a C code, gvf.c, found at <http://www.iacel.ece.jhu.edu/resources/>, the required time decreases to 5.24 seconds. We have also rewritten movesnake.m into movesnake.c. Using both gvf.c and movesnake.c, the required time further decreases to 3.14 seconds (see Fig. 8). The time for the GVF tracker as shown in Fig. 6 for comparison is this further reduced time of 3.14 seconds.



Number of frames tracked by MC tracker - Number of frames tracked by GVF tracker



Number of frames tracked by PFF tracker - Number of frames tracked by GVF tracker



Number of frames tracked by PFF tracker - Number of frames tracked by MC tracker

Fig. 6. The differences in numbers of frames tracked by three different trackers for each of the 30 sequences.

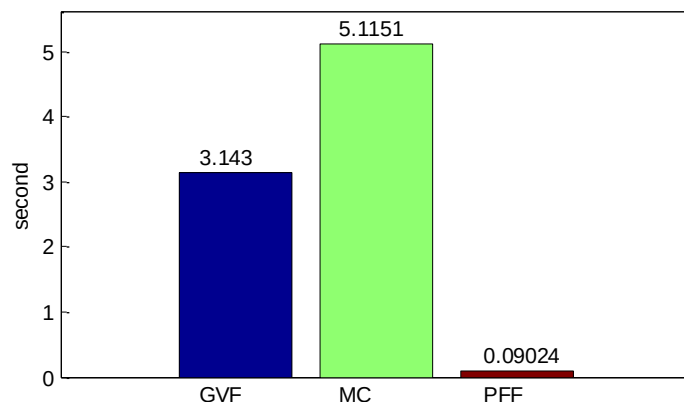


Fig. 7. Average time required for tracking in each sequence.

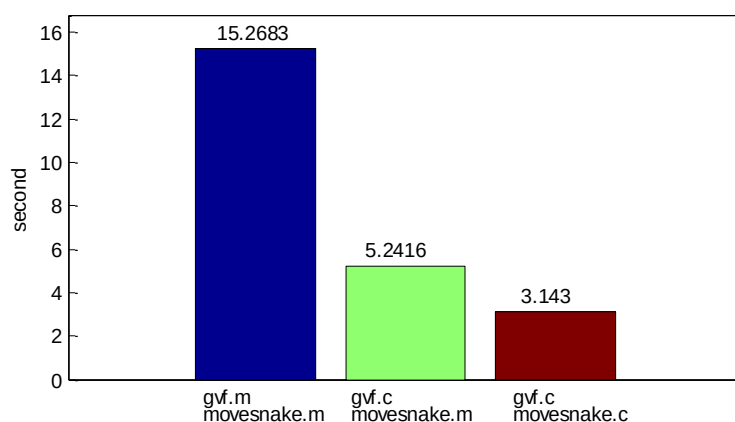


Fig. 8. Average time per sequence required by the GVF snake tracker with different codes.

5. Multiple Target Tracking

The PFF tracker is also effective in tracking multiple targets. Shown in Fig. 9 is the performance of the PFF tracker in tracking 8 UAVs in a video sequence. The PFF tracker tracks all 8 targets. On the other hand, the MC tracker tracks only 6 targets, and misses 2 targets. More specifically, it tracks target #7 in 32 frames out of the 79 frames, and tracks target #8 in 49 frames out of 150 frames. The GVF tracker has similar performance as the MC tracker. It tracks 6 targets, and misses 2 targets (track target #5 in 46 frames out of the 100 frames, and tracks target #8 in 49 frames out of 150 frames). We observe that both the GVF and MC trackers miss targets when targets pass under the tree or are close to road side.

Shown in Fig. 10 is the tracking speeds of all three trackers, indicating the superior performance of the PFF tracker.

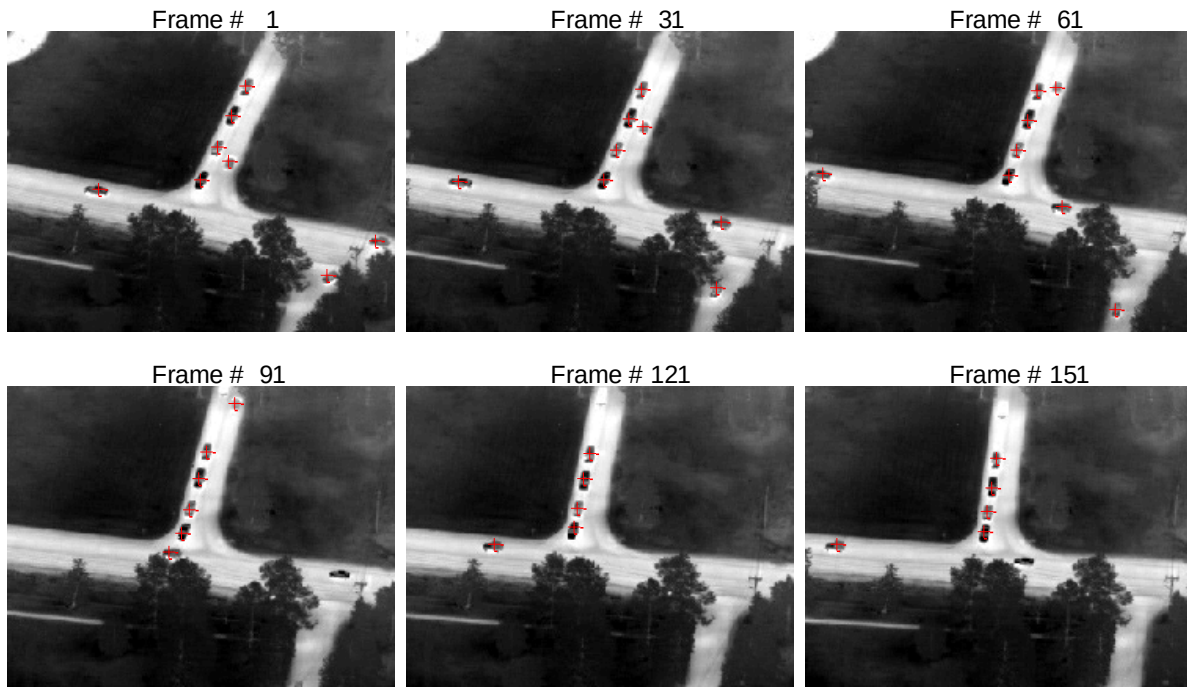


Fig. 9. The PFF tracker tracks all 8 targets in a 5 second segment of a typical UAV video

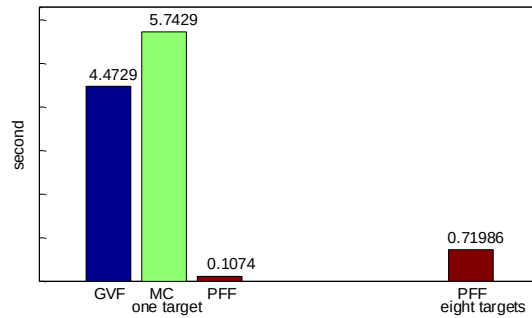


Fig. 10. Time required in tracking.

References

- [Cui et al. 2006] J. Cui, S.T. Acton and Z. Lin, "A Monte Carlo approach to rolling leukocyte tracking in vivo," *Medical Image Analysis*, Vol.10, No. 4, pp.598-610, 2006.
- [Ray et al. 2002] N. Ray, S.T. Acton and K. Ley, "Tracking leukocytes in vivo with shape and size constrained active contours," *IEEE Transactions on Medical Imaging*, Vol. 21, No.10, pp.1222-1235, 2002.