A Comparative Study of Conventional and Deep Learning Target Tracking Algorithms for Low Quality Videos

Chiman Kwan(✉), Bryan Chou, and Li-Yun Martin Kwan

Applied Research LLC, Rockville, MD, USA
chiman.kwan@signalpro.net,
choub90@gmail.com, martin.kwan97@gmail.com

Abstract. This paper presents a comparative study of several state-of-the-art target tracking algorithms, including conventional and deep learning ones, for low quality videos. A challenge video data set known as SENSIAC, which contains both optical and infrared videos at long ranges (1000 m–5000 m), was used in our investigations. It was found that none of the trackers can perform well under all conditions. It appears that the field of video tracking still needs some serious development in order to reach maturity.

Keywords: Deep learning · Target tracking · Low quality videos
Kalman filter

1 Introduction

Target tracking using optical and infrared imagers has many applications such as security monitoring and surveillance operations. Compared to radar [1, 2], multi-spectral [3, 4], and hyperspectral sensors [5, 6], optical and infrared imagers are low cost and easy to install and operate. In recent years, there are new tracking algorithms using track-learn-detect [7], compressive sensing [8, 9], deep learning [10], tracking by detection [11], and references therein. These algorithms have been proven to work well in benchmark data sets. However, the benchmark videos are of high resolution and high quality. In contrast, some realistic videos such as the SENSIAC videos [12] are of low quality in terms of resolution and environmental conditions.

One objective of this research is to compare some representative tracking algorithms in the literature using the SENSIAC data [12], which have both optical and infrared videos at various ranges. We do not have any preference on any particular tracking algorithms. In fact, we also include Kalman tracker [13–15], which is probably the oldest algorithm in the literature. Another objective is to see if deep learning approaches, due to recent hype in deep learning, are better than conventional algorithms.

This paper is organized as follows. In Sect. 2, we will briefly review the tracking algorithms in this study. Although the algorithms are not exhaustive, they are representative methods of many state-of-the-art algorithms in the literature. Section 3 focuses on an extensive comparative study using actual videos. Two performance metrics were used to compare different algorithms. Finally, some concluding remarks are included in Sect. 4.

T. Huang et al. (Eds.): ISNN 2018, LNCS 10878, pp. 521–531, 2018.
https://doi.org/10.1007/978-3-319-92537-0_60

2 Tracking Algorithms

The following approaches are by no means an exhaustive list of current methods. However, they are representative methods in target tracking in recent years. Some deep learning approaches were not included because our PCs do not have the necessary hardware or software to run them. We briefly outline the key ideas of each method below.

2.1 STAPLE Tracker [16]

For this algorithm, the histogram of oriented gradients (HOG) features are extracted from the most recent estimated target location and used to update the models of the tracker. Then a template response is calculated using the updated models and the extracted features from the next frame. To be able to estimate the location of the target, the histogram response is needed along with the template response. The histogram response is calculated by updating the weights in the current frame. Then the per-pixel score is computed using the next frame. This score and the weights, calculated before, are used to determine the integral image, and ultimately, the histogram response. Together, with the template and histogram response, the tracker is able to estimate the location of the target.

The STAPLE tracker [16] is able to successfully track the target of interest until the end of a video when there is no occlusion. Even with a camera that is not stationary, STAPLE [16] is able to keep a tight bounding box around the target and the bounding box appears to scale according to the target. However, the scaling of the bounding box is too little to be significant. There are some cases where the bounding box does not completely encase the entire target, but it will still follow and track the target when there is partial encasement by the bounding box. One major issue that STAPLE [16] suffers from is the case of occlusions. Once the target becomes occluded, STAPLE [16] is unable to redetect the target to track again after emerging from the occlusion. Overall, STAPLE [16] works well for targets that do not become occluded.

2.2 Long-Term Correlation Tracking (LCT) Tracker [17]

This algorithm starts by using the initial bounding box and expanding it to specify a search window. Features are then extracted from within the search window to estimate the target location. After the location has been computed, the scaling is then calculated for the bounding box. Then the program checks to make sure that the correct target is being tracked. If it is not, then the tracker performs redetection by finding possible states and chooses the most accurate state by comparing the confidence scoring for each state. After redetection, the appearance and motion model are updated. This update is performed regardless of whether or not the redetection module is performed. This cycle is continued until the end of the video.

The LCT tracker [17] is able to successfully track the target of interest until the end of the video. This algorithm has proven to be quite robust in that it is able to handle occlusions and a non-stationary camera. Although the LCT [17] is able to handle cases of light to moderate occlusion, it is unsuccessful when there is heavy occlusion. Such

as when a target is under a heavy shadow that spans multiple frames. It has been found that the LCT [17] is unsuccessful because of the dramatic changes the shadows make on the appearance model of the target. Another minor fault of this algorithm is the dynamic scaling of the bounding box. When the orientation of the target changes and the bounding box is scaled, there are some cases when the bounding box becomes too large and covers more area around the target of interest than desired. Overall, the LCT [17] algorithm is robust and able to handle most cases of occlusion.

2.3 Fusion of STAPLE and LCT

The Fusion of STAPLE [16] and LCT [17] merges the two algorithms into one program. We implemented this fusion algorithm. The reasoning for this merge is to combine the best features of the two algorithms and resolve the main issues that each algorithm suffers as individual programs. It just so happens that the issues with STAPLE [16] can be resolved by the LCT [17] and the issues with the LCT [17] can be resolved with the STAPLE [16]. This Fusion tracker is able to successfully track the target of interest until the end of the video while keeping a tight fitting bounding box around the target. This includes cases with light to medium occlusion.

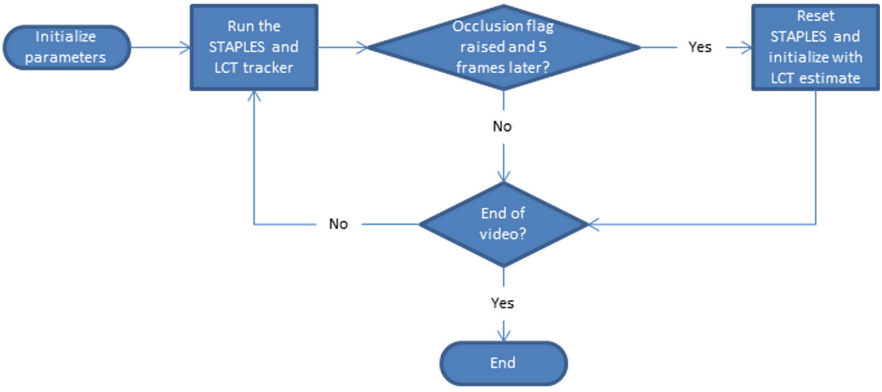


Fig. 1. Fusion of STAPLE and LCT tracking algorithms.

Figure 1 illustrates the fusion based tracker. The fusion tracker works by running the STAPLE [16] and LCT [17] trackers simultaneously. Since the bounding box information from the STAPLE [16] tracker has a more desirable result, STAPLE [16] is used to visualize the location of the target. The LCT [17] is used to detect occlusion and for redetection. Once occlusions are detected, a flag is raised and the program waits for 5 frames to pass so that the target has some time to emerge from the occlusion. Once the flag is raised and the 5 frames have passed, STAPLE [16] is reset and initialized with the location information from the LCT [17]. The purpose of resetting STAPLE [16] is to clear the history of the appearance and motion model. This cycle continues until the end of the video.

2.4 Kalman Tracker

Although Kalman tracker is easy to understand, we could not find a good Kalman tracker in the internet. So we implemented this by ourselves. The Kalman tracker is able to successfully track a moving target at close range until the end of the video if there are no occlusions and the camera is stationary. It has been found that the Kalman tracker has issues when the target is stationary because of its reliance on motion to be able to successfully track. Detection of motion is only performed every ten frames to ensure that there are notable differences between two frames. Overall, the Kalman tracker works for close range targets captured with a stationary camera.

Figure 2 illustrates the Kalman tracker. Given an initial position and velocity, a prediction of the next location is calculated for the first frame. Then, the same calculation is made for frames in between intervals of 10. For every other 10 frames, the Kalman filter parameters are updated using the motion of the target. More specifically, the measurement residual and Kalman gain are updated to predict a more accurate state estimate. This cycle continues until the end of the video.

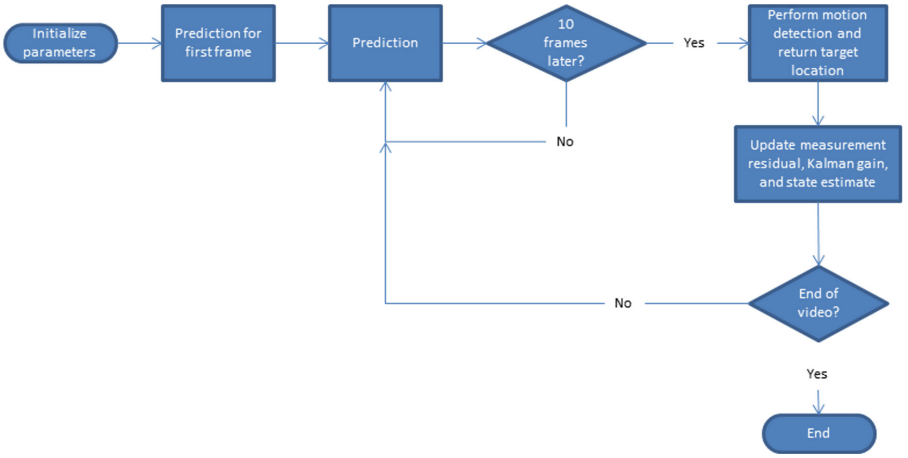


Fig. 2. Kalman tracking algorithm.

2.5 Hierarchical Convolutional Features for Visual Tracking (CF Tracker)

This is deep learning based tracker. As shown in Fig. 3, the CF tracker starts off the tracking by cropping out a search window from the first frame based on the initial position that is used as an input for the program. Once the search window has been established, convolutional features are extracted with spatial interpolation. Once this has been completed, a confidence score is computed for each VGG net layer. This score is used to estimate the closest target location for the next frame. Then another area is cropped out from the whole frame using the newest estimate and the convolutional features are extracted with interpolation to update the correlation filters for each layer. This cycle is repeated until the end of the video is reached.

The CF tracker performs similarly to the STAPLE and LCT tracker. It is able to track the target until the end of the video in most cases. This tracker is able to keep a bounding box around the target when the camera is not stationary and the scaling of the bounding box is adaptive, much like the STAPLE bounding box. However, the bounding box size changes are too small to be significant. One issue of this tracker is that the computational time is quite long. For one video of approximately 1,875 frames, the tracker takes about 30 min to complete. Although the tracker has not been tested on videos with occlusion, it appears that it would not handle occlusion very well due to the similar behavior with the STAPLE. Furthermore, the code does not have a function or algorithm for redetection.

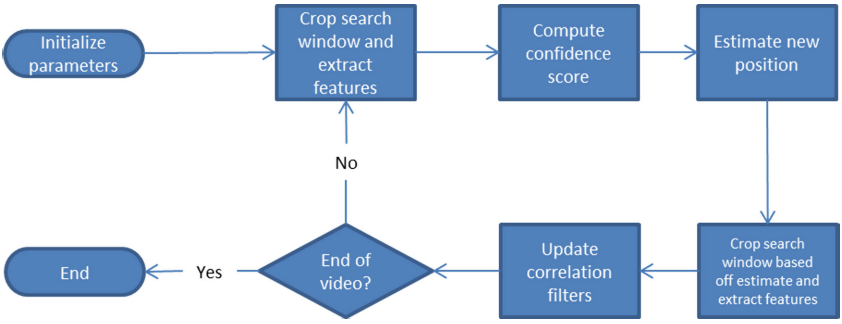


Fig. 3. Deep learning tracker.

3 Experiments

3.1 SENSIAC Database Description

All the tracking algorithms have been tested using the SENSIAC database [12], which contain different vehicles and human targets at multiple ranges. These videos are captured using both optical and mid-wave infrared (MWIR) cameras. This data set is available for purchase [12]. In this paper, we focus only on vehicles.

For vehicles, there are a total of nine targets. It is important to note that two targets were excluded because not all scenarios were available for the particular targets. These targets vary in size from a pickup truck to a tank. For each target, there are a total of 18 scenarios, nine for daytime and nine for night time. These nine daytime and nighttime scenarios vary in range from the target to the camera used to capture the video. The range starts from 1,000 m and ends at 5,000 m with an interval of 500 m. All the vehicles drive in a circular pattern at speeds specified within the ground truth files associated with each scenario. In total, there are 162 videos for vehicles.

3.2 Vehicle Tracking Results

Although there are quite a few performance metrics for evaluating different trackers in the literature, only two different performance metrics were performed on each of the

scenarios. This is because only the ground truth center locations of the vehicles are available in the database. The first one was the distance precision (DP), which computes the average number of frames the estimate location was within range of the ground truth location. The second performance metric is the center location error (CLE), which computes the average distance between the ground truth location and the estimated location. The following tables are the averages of all targets for a particular range for each tracking algorithm.

Table 1. Averaged Center Location Error (CLE) for all cases. Optical videos.

CLE							
			Algorithms				
			LCT	STAPLE	Fusion	Kalman	CF
Ranges	Day	1000	26.2451	24.3939	18.1098	**5.5431**	28.1734
		1500	18.3854	15.8202	12.5347	**7.1036**	16.0348
		2000	9.4061	11.1048	7.0533	**3.3757**	12.8006
		2500	67.3546	**38.8866**	61.0648	97.7282	45.8926
		3000	57.8209	**13.6605**	21.7178	80.7708	22.0208
		3500	19.1763	**14.2882**	49.8143	87.3758	26.9078
		4000	32.0739	**6.7583**	49.2002	105.4559	20.6885
		4500	31.1081	**17.6424**	23.9859	66.0445	23.2431
		5000	35.2556	**24.7713**	33.1415	51.2991	29.2628

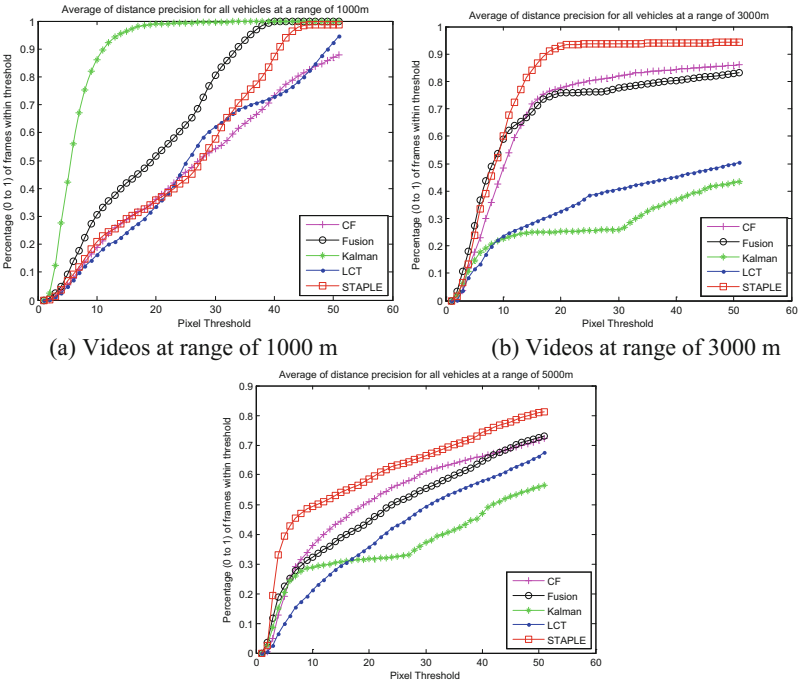
Table 2. Averaged Distance Precision (DP) at threshold of 20 pixels for all cases.

DP (20 pixel threshold)							
			Algorithms				
			LCT	STAPLE	Fusion	Kalman	CF
Ranges	Day	1000	0.3339	0.3574	0.5253	**0.9898**	0.3597
		1500	0.6352	0.6013	0.8247	**0.9860**	0.6350
		2000	0.9910	0.8650	**0.9998**	0.9986	0.8660
		2500	0.2700	**0.6502**	0.4535	0.2140	0.5336
		3000	0.3261	**0.9293**	0.7583	0.2536	0.7773
		3500	0.7830	**0.9100**	0.3890	0.2239	0.6539
		4000	0.4808	**0.9587**	0.3254	0.3459	0.7217
		4500	0.3591	**0.6466**	0.4985	0.1648	0.5196
		5000	0.3564	**0.5851**	0.4464	0.3192	0.5090

Optical Camera Results. Table 1 summarizes the averaged CLE of tracking results of different algorithms at different ranges. It should be noted that there are a number of vehicles at each range. Smaller CLEs mean better performance. It can be seen that Kalman tracker works well for ranges less than or equal to 2000 m. STAPLE works well for ranges longer than 2000 m. Table 2 shows the averaged Distance Precision

(DP) at threshold of 20 pixels for all cases. Again, it can be seen that the Kalman tracker works well for short ranges and STAPLE works well for long ranges. Compared to the conventional trackers, the deep learning based tracker (CF) only performs moderately well in long ranges. The fusion approach works well only when there are occlusions, which are not present in the SENSIAC videos. In terms of computational speed, Kalman and STAPLE are the fastest, followed the LCT, fusion, and CF.

Figure 4 shows the averaged DP at three ranges. The trends are similar to what we observe in Tables 1 and 2.



(a) Videos at range of 1000 m

(b) Videos at range of 3000 m

(c) Videos at range of 5000 m

Fig. 4. Average DP at various ranges. Optical videos.

Infrared Camera Results. Different from the optical videos that have only day time videos, the infrared videos have both day and night time videos. Table 3 shows the averaged CLE results for all cases. In both day and night time cases, STAPLE performs quite well for all ranges. All other algorithms do not perform that well. Similarly, Table 4 shows the averaged DP results for all cases. Again, STAPLE performs well in almost all cases. Figures 5, 6 and 7 plot the DP results for different ranges. We can observe the similar trends mentioned above. In terms of computational speed, Kalman and STAPLE are the fastest, followed the LCT, fusion, and CF.

Table 3. Averaged Center Location Error (CLE) for all cases. Infrared videos.

CLE							
			Algorithms				
			LCT	STAPLE	Fusion	Kalman	CF
Ranges	Day	1000	32.6376	**24.6038**	53.2527	89.1767	71.2136
		1500	22.6851	**13.9725**	17.5248	75.4930	63.8537
		2000	**10.2987**	15.5561	38.2592	59.2863	23.4091
		2500	55.6768	**47.5998**	56.6269	115.8784	52.7754
		3000	52.3696	24.0137	**23.8464**	85.4935	42.6051
		3500	29.5317	**8.5811**	50.1563	82.4144	39.9273
		4000	29.2585	**10.6956**	13.4652	45.4083	32.8579
		4500	25.9339	**21.1822**	25.3266	67.7747	42.2674
		5000	29.6687	**20.6289**	20.7262	132.2764	27.0285
	Night	1000	**18.0577**	29.7589	27.5016	80.0698	19.7410
		1500	12.5835	17.6375	17.2377	**8.9518**	12.2514
		2000	**7.5399**	11.9845	11.9410	7.9800	8.9405
		2500	37.1904	8.4052	**8.3242**	33.3183	72.2683
		3000	37.9731	**7.3029**	19.4694	51.0178	45.3617
		3500	24.5018	**6.8555**	25.5366	53.5819	31.9712
		4000	15.5882	**5.6554**	17.9157	33.1909	11.8107
		4500	23.6879	**8.3042**	19.2317	30.6024	18.9383
		5000	30.1010	**15.4727**	15.5971	158.9508	53.6705

Table 4. Averaged Distance Precision (DP) at threshold of 20 pixels for all cases.

DP (20 pixel threshold)							
			Algorithms				
			LCT	STAPLE	Fusion	Kalman	CF
Ranges	Day	1000	0.4533	0.3361	0.5508	**0.7006**	0.4058
		1500	**0.8022**	0.6575	0.7530	0.6873	0.3521
		2000	**0.9077**	0.7704	0.6330	0.5085	0.6684
		2500	0.2952	**0.4457**	0.3660	0.2935	0.2466
		3000	0.3306	0.6832	**0.6915**	0.0999	0.3058
		3500	0.5648	**0.9211**	0.3554	0.1018	0.2832
		4000	0.4543	**0.8544**	0.8038	0.1949	0.3678
		4500	0.4119	**0.6080**	0.4651	0.0276	0.3276
		5000	0.3751	0.5880	**0.5940**	0.2469	0.5035
	Night	1000	**0.5554**	0.3552	0.4456	0.4715	0.5247
		1500	0.7544	0.5654	0.6362	**0.9860**	0.8069
		2000	0.9620	0.7824	0.8503	0.9666	**1.0000**
		2500	0.5775	0.8954	**0.9027**	0.6246	0.4941
		3000	0.5303	**0.9304**	0.8347	0.4708	0.5638
		3500	0.6578	**0.9599**	0.7717	0.3769	0.5397
		4000	0.7932	**0.9619**	0.6293	0.4003	0.8157
		4500	0.5329	**0.8743**	0.5227	0.3767	0.6773
		5000	0.3797	**0.7411**	0.7426	0.0094	0.5394

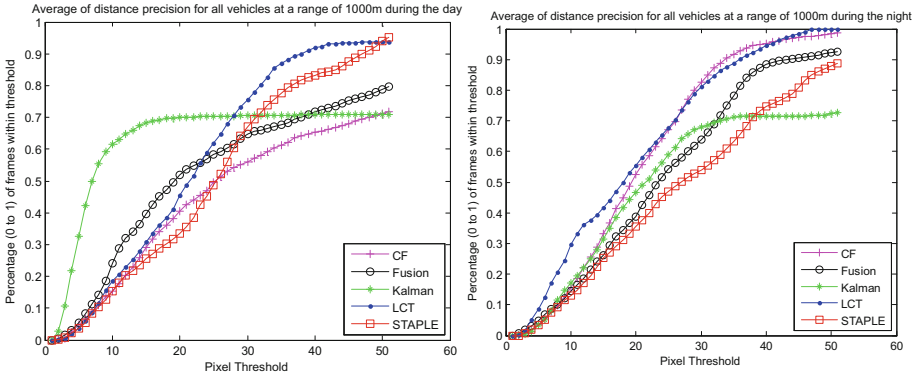


Fig. 5. Average distance precision for videos at range of 1000 m. Left: Infrared videos at day time; right: infrared videos at night time.

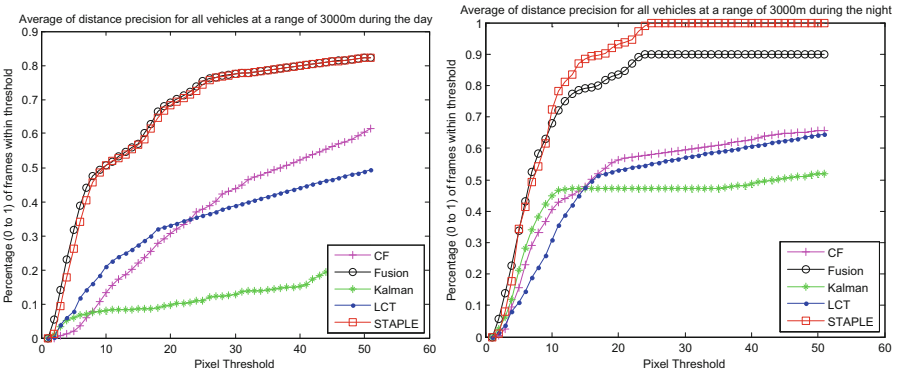


Fig. 6. Average distance precision for videos at range of 3000 m. Left: Infrared videos at day time; right: infrared videos at night time.

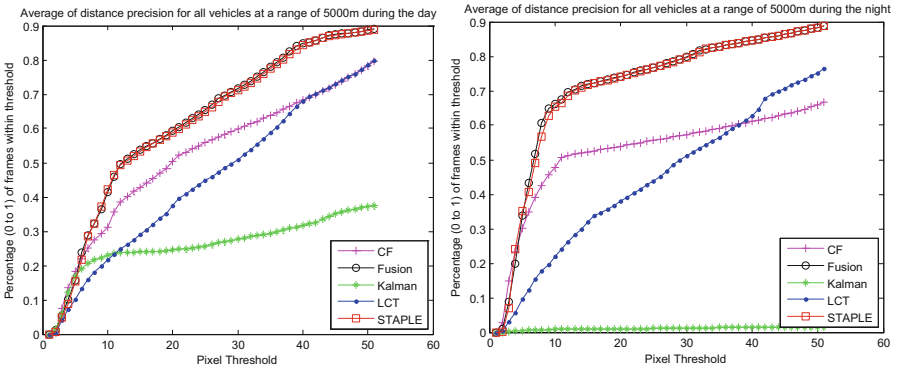


Fig. 7. Average distance precision for videos at range of 5000 m. Left: Infrared videos at day time; right: infrared videos at night time.

4 Conclusions

In this paper, we address target tracking for low quality videos. The low quality is caused by long range data acquisition as well as environmental conditions due to poor illumination and camera motions. Five representative trackers were used in our comparative study. Two performance metrics (center location error and distance precision) were used in our experiments. It was observed that one tracker known as STAPLE performed quite well whereas the deep learning based tracker did not work as well as STAPLE. A somewhat surprising results is that the Kalman tracker also works well up to 2000 m for optical videos. It is our belief that the field of target tracking still needs a lot of research, including deep learning based methods.

Acknowledgments. This research was supported by US Air Force under contract FA8651-17-C-0017. The views, opinions and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

1. Zhao, Z., Chen, H., Chen, G., Kwan, C., Li, X.: Comparison of several ballistic target tracking filters. In: Proceedings of American Control Conference, pp. 2197–2202 (2006)
2. Zhao, Z., Chen, H., Chen, G., Kwan, C., Li, X.: IMM-LMMSE filtering algorithm for ballistic target tracking with unknown ballistic coefficient. In: Proceedings of SPIE, Signal and Data Processing of Small Targets, vol. 6236 (2006)
3. Dao, M., Kwan, C., Koperski, K., Marchisio, G.: A joint sparsity approach to tunnel activity monitoring using high resolution satellite images. In: IEEE Ubiquitous Computing, Electronics & Mobile Communication Conference, pp. 322–328 (2017)
4. Perez, D., Banerjee, D., Kwan, C., Dao, M., Shen, Y., Koperski, K., Marchisio, G., Li, J.: Deep learning for effective detection of excavated soil related to illegal tunnel activities. In: IEEE Ubiquitous Computing, Electronics & Mobile Communication Conference, pp. 626–632 (2017)
5. Qu, Y., Qi, H., Ayhan, B., Kwan, C., Kidd, R.: Does multispectral/hyperspectral pansharpening improve the performance of anomaly detection? IEEE International Geoscience and Remote Sensing Symposium, pp. 6130–6133 (2017)
6. Zhou, J., Kwan, C., Ayhan, B.: Improved target detection for hyperspectral images using hybrid in-scene calibration. J. Appl. Remote Sens. **11**(3), 035010 (2017)
7. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-Learning-Detection. IEEE Trans. Pattern Anal. Mach. Intell. **34**(7), 1409–1422 (2012)
8. Mei, X., Ling, H.: Robust visual tracking and vehicle classification via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **33**(11), 2259–2272 (2011)
9. Zhang, K., Zhang, L., Yang, M.-H.: Real-time compressive tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7574, pp. 864–877. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33712-3_62
10. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical convolutional features for visual tracking. In: Computer Vision (ICCV) (2015)

11. Li, X., Kwan, C., Mei, G., Li, B.: A generic approach to object matching and tracking. In: Campilho, A., Kamel, M.S. (eds.) ICIAR 2006. LNCS, vol. 4141, pp. 839–849. Springer, Heidelberg (2006). https://doi.org/10.1007/11867586_76
12. SENSIAC. https://www.sensiac.org/external/products/list_databases.jsf
13. Lewis, F.L.: Optimal Estimation. Wiley, Hoboken (1986)
14. Zhou, J., Kwan, C.: Tracking of multiple pixel targets using multiple cameras. In: 15th International Symposium on Neural Networks (2018)
15. Kwan, C., Lewis, F.L.: A note on kalman filtering. IEEE Trans. Educ. **42**(3), 225–227 (1999)
16. Bertinetto, L., et al.: Staple: complementary learners for real-time tracking. In: Conference on Computer Vision and Pattern Recognition (2016)
17. Ma, C., Yang, X., Zhang, C., Yang, M.H.: Long-term correlation tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, pp. 5388–5396 (2015)