2015 IEEE 14th International Conference on Machine Learning and Applications

Robust Vehicle Tracking Using Perceptual Hashing Algorithm

Zheng Li†, Jian-Fei Yang†, Long Chen*, Juan Zha
School of Mobile Information Engineering
Sun Yat-sen University, Zhuhai, P. R. China, 519082
hsqmlz@foxmail.com, jianfei_mars@hotmail.com, chenl46@mail.sysu.edu.cn, zhaj0804@foxmail.com
*Corresponding author: Dr L. Chen
†These two authors contributed equally to this work.

***Abstract*—Vehicle tracking, significant in the computer vision using machine learning method, allows the vehicle to comprehend its immediate environment and therefore, enhances the intelligence of the vehicles and the safety of vehicle occupants. We propose a novel tracking algorithm that can work robustly under challenging circumstances such as road scene where several kinds of appearance and motion changes of a tracking object occur. Our algorithm is based on the perceptual hashing algorithm (PHA) and the color, low-frequency and rotation information are considered. By means of PHA, our tracker generates a single identification at each frame. The sliding windows produce a series of candidates between consecutive frames so that the new position of tracking object can be updated by comparing the binary code of candidates and identification. In the experiment, the quantitative and qualitative results are expressed by center location error (CLE) and VOC overlap ratio(VOR). Compared to the advanced tracker at present, PHA tracker shows its robustness when confronting violent changes of noise, illumination, background clutter and part occlusion, which demonstrates its state-of-the-art performance in the field of dynamic vehicle tracking.**

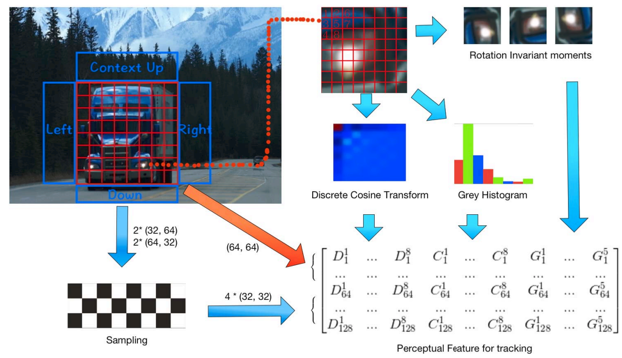

Fig. 1: Perceptual Feature Extraction

I. Introduction

The research of object tracking plays an important role in computer vision community due to its wide range of applications such as automatic surveillance, human-computer interaction, activity recognition and robotics. Among them, the application in Intelligent Transportation System(ITS) such as [1] [2] [3] [4] becomes more and more significant. The data from tachograph can be dealt with automatically and the development of driverless vehicles naturally depends on it as perceptual analysis. For a long time, the major challenges of visual tracking lie in the rapid and tremendous appearance variation due to noise, occlusion, illumination, background clutter and scale changes. In order to solve these problems, tracking method requires more accurate observation model as well as efficient tracking model with the help of machine learning. There are numerous algorithms proposed in recent years, which can be categorized into region-based, model-based, feature-based and active contour-based algorithm.

The region-based algorithm such as [5] segments the video objects and establishes the connections between segmented regions in consecutive frames to track the object. It performs sensible in multi-objects tracking but weak in occlusion. On contrast, the model-based algorithm like [6], which use the prior knowledge to obtain the model of the object, can deal with the occlusion problem. Therefore, this method is widely applied to human tracking. The limitations stems from its compulsory prior knowledge including appearance model and structure. The feature-based one [7] extracts the features of the object and combine them as high level characteristics. The matching of the features between frames guides the tracking procedure. However, the high complexity of computation decreases the utility. As for active contour-based algorithm [8], it proves efficient but sensitive to the initial condition. Although lots of advanced trackers based on traditional method, such as Frag [9], IVT [10], SemiT [11], TLD [12], VTD [13], LSK [14], Struck [15], VTS [16] and MTT [17], are presented, the visual tracking still has a long way to develop.

In this paper, we address the problem of designing a robust and discriminative perceptual hashing algorithm and bestowing the responsibilities of extracting features for the tracking framework. The philosophy of our method is to discover and utilize the perceptual information, such as color and low-frequency information, to the efficient and stable visual tracking. The perceptual information such as [18] mainly refers to DCT coefficients and color distribution. Only the perception of tracking object results in weak performance of occlusion so the features of spatial context make sense. Moreover, to deal with the rotation invariance, geometric moments are introduced. For all features extracted, a hashing method reduces them to a binary matrix, the corresponding identification. In our tracking system, the object can be discovered in the consecutive frames while updating with the discriminative identification. The experiment bears out the accuracy and robustness compared with advanced tracking algorithm. Owing that our perceptual

978-1-5090-0287-0/15 $31.00 © 2015 IEEE
DOI 10.1109/ICMLA.2015.104

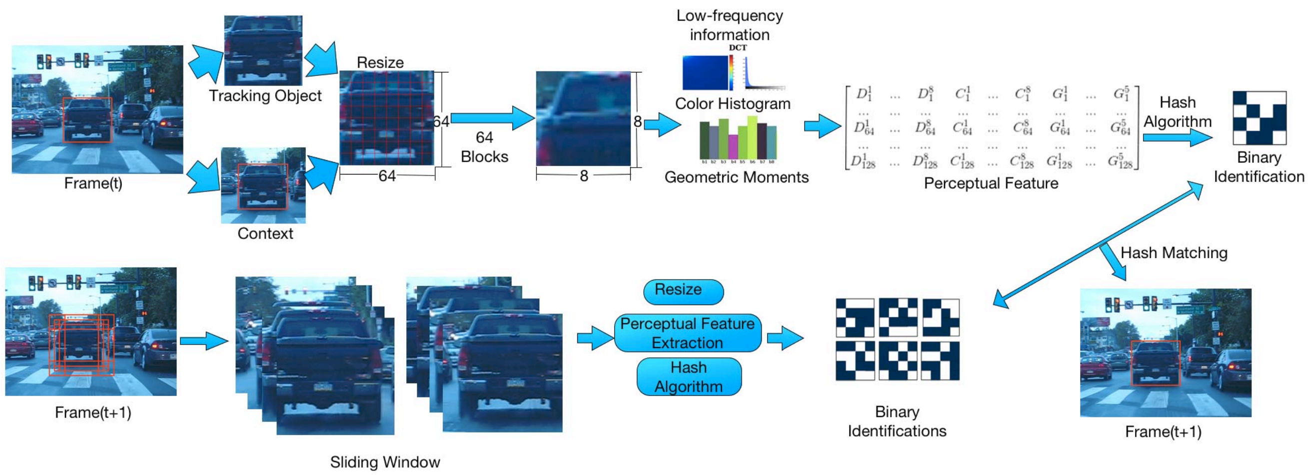

Fig. 2: Framework of the proposed Perceptual Hashing Tracking

hashing algorithm is insensitive to illumination, occlusion and background clutter, it is suitable for vehicle tracking, part of the automated perception systems, which makes difference in ITS. It allows the vehicle to comprehend its immediate environment and therefore, enhances the intelligence of the vehicles and the safety of vehicle occupants. Furthermore, the accurate tracking makes it possible to realize the ultimate goal of autonomous automobile driving.

In summary, we propose a robust and discriminative visual tracker that combines the perceptual feature, context information and geometric invariant moments to a binary code as unique identification via local-sensitive hashing algorithm [20]. The experiment performs its state-of-art, especially tackle the vehicle tracking. To our knowledge, it is the first time that the perceptual hashing algorithm is applied to vehicle tracking. Undoubtedly, the high performance and wide applicant range demonstrates its long-tern development.

Fig. 3: Severe occlusion may occur in the visual tracking.

II. Perceptual Hashing Tracking

The workflow of our method is summarized in Figure 2. In the preprocessing of each frame, the block image strategy makes it simple to process and evaluate the tracking object. Generally, the feature-based algorithm gives each frame a discriminative feature vector so we construct the vectors with perceptual feature of tracking object and context. Then, the perceptual feature is transformed into a series of binary codes via hash computation for fast matching.

A. Frame Preprocessing

For the frame of t, we define the region of tracking object as $\mathbf{I^T}$ and the context as $\mathbf{I^C}$. The purpose of introducing the context is that severe occlusion may mislead the tracker to miss the target as Figure 3, especially for most current vehicle detection and tracking systems based on particle filter. The contexts of the tracking object are sampled in four directions so that the tracker is able to recognize the object by the surrounding environment when confronting occlusion. In order to obtain the same length of the final feature, we resize $\mathbf{I^T}$ to (64, 64). As the Figure 1 shows, the contexts are segmented from four directions of the tracking object with $(\frac{1}{2}length(\mathbf{I^T}), width(\mathbf{I^T}))$ and $(length(\mathbf{I^T}), \frac{1}{2}width(\mathbf{I^T}))$. Naturally, they can be resized to (32, 64) and (64, 32). In our block images strategy, each image of (64, 64) is separated into 64 blocks with the size of (8, 8). By way of the proper weight of object and context, we sample the blocks of context to decrease the number of them to 64. Therefore, the preprocessed data includes 64 blocks of tracking object $\mathbf{B^T}$ and 64 blocks of the contexts $\mathbf{B^C}$, from which the perceptual feature is extracted.

B. Perceptual Feature extraction

The perceptual feature makes preparations for the hashing computation and depicts the basic characteristics of the tracking object. In this part, we will calculate the perceptual feature P_T and P_C for each block of tracking object and the context. P is a 21-dimension vector including 8 DCT coefficients D, 8 color histogram H and 5 geometric moments M as shown in (1).

$$\mathbf{P}_i = \mathbf{H}_i + \mathbf{D}_i + \mathbf{G}_i, i \in \left\{\mathbf{I^T}, \mathbf{I^C}\right\} \quad (1)$$

1) Image histogram learning: The color information can be expressed by image histogram via statistical method. Each of its value is an integer representing the frequency of occurrence of a particular section of intensity values. The corresponding 8-dimension feature value $H(r)$ is defined as:

$$\mathbf{H}(r) = \sum_{p=1}^{W} Q(\mathbf{I}_p, r), r = 1, 2, ..., N_h \tag{2}$$

where W denotes the number of pixels of a block, N_h is the total number of sections ($N_h = 8$) and $Q(\mathbf{I}_p, r)$ is zero except when intensity value $\mathbf{I}_p$ (at pixel location p) belongs to section r.

2) Low-frequency information: In the signal processing, the low-frequency information generally gets obtained by Discrete Cosine Transform, which outputs the feature values with the equivalent count of inputs. Thus, all pixels of a block is used as inputs y_k. The DCT coefficients are found out through (3).

$$D_m = \sum_{k=1}^{64} y_k \cos\left[\frac{\pi}{8} m \left(k + \frac{1}{2}\right)\right] \tag{3}$$

We select 8 discriminative values in the upper left corner of DCT feature matrix where the low-frequency information gathers. Through DCT and feature selection, the 8-dimension vector $\mathbf{D}$, mainly standing for the substantive information of profile, is obtained.

3) Rotation invariance: Until now, the perceptual feature performs good in profile and color, but lack of rotation as well as scaling invariance. To boost the sphere of application, we bring in the geometric invariant moments [19], a set of statistical features of image shape. They are not sensitive to rotation, scaling and other normal operations and have many kinds of expression types such as Hu and Zernike moment. Considering the computation complexity in tracking issues, our method uses Hu moments as follows, and η_{ij} denotes the $(i+j)$ order normalized central moment of the image

$$\begin{aligned}
M_1 &= \eta_{20} + \eta_{02} \\
M_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
M_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - 3\eta_{03})^2 \\
M_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} - 3\eta_{03})^2 \\
M_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + 3\eta_{03})^2] \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{12} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
M_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
M_7 &= (3\eta_{12} - \eta_{30})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
\end{aligned}$$

By comparing the 7 Hu moments of original and rotated image, we select the M_1, M_3, M_4, M_5, M_6 as $\mathbf{G}_i$ as they are more invariant for rotation.

C. Locality-sensitive Hashing and Fast Matching

To compare different features of images fast, the perceptual features matrix is transformed to a binary code by locality-sensitive hashing (LSH), widely used in similarity search [20].

Algorithm 1 Perceptual Hashing Tracking

```
Input: Frame(t), S^{t-1}
Sliding windows I^T and cosresponding I^C
repeat
    Resize the I^T to (64, 64), the I^C to (32, 64) and (64, 32)
    Divide them into 128 (8, 8) blocks
    repeat
        Calculate the grey histogram to obtain H
        Calculate the D via Discrete Cosine Transform
        Calculate the geometric moment G
        P_B = H_B + D_B + G_B
    until No block is unprocessed
    Combine the P_B to P
    Find the C_t through hashing algorithm
until No sliding window is available
Compare S^t with C_t to find the most similar one Ĉ_t
Update S^t with Ĉ_t
```

For each block feature $v_B \in \mathbb{R}^{16}$, we construct m corresponding hashing functions, one of which is define as follows:

$$h_{B_i}(v_B) = sign(w_{B_i}^{\mathrm{T}} v_B + b) \tag{4}$$

where $w_{B_i} \in \mathbb{R}^{16}$ is generated randomly between $[-1, 1]$ satisfying *Gaussian distribution* and b is set to 0. Each block B can generate m binary codes through m hashing functions $\{h_{B_1}(\cdot), h_{B_2}(\cdot), \cdots, h_{B_m}(\cdot)\}$.

Then, the binary code $\hat{\mathbf{C}}_{\mathbf{t}}$, composed of all the binary codes of blocks, becomes the unique single identification $\mathbf{S}_t$ at time t. According to the sliding windows $\mathbf{S}$ at the time of $(t+1)$, there are a set of candidate code defined as C_{t+1}^S. The position of tracking object is updated as follows:

$$\mathbf{S}^{t+1} = Match(\mathbf{C}_{t+1}^S, \mathbf{S}^t) \tag{5}$$

where $Match(\cdot)$ function selects the most similar identification between the object of t and series of sliding windows by calculating the minimum *Hamming distance*.

In general, Algorithm 1 displays the details. The proposed method highlights the importance of perceptual feature in our tracker and speed up the matching process through hashing algorithm. The next part will demonstrate fast speed, robustness and wide-application of our method after compared to most of the advanced tracker at present.

TABLE I: Property of experiment for each video

Datasets	Property
BlurCar1, 3, 4	**MB, FM**
Car, Car4	**IV, SV**
Car1	**IV, SV, MB, FM, BC, LR**
Car2	**IV, SV, MB, FM, BC**
CarDark, CarDark2	**IV, BC**
CarDay	**FM, SV, DEF**

Fig. 4: Qualitative tracking results of the five trackers over four representative frames of the eight video sequences (*BlurCar1, BlurCar3, BlurCar4, Car, Car2, Car4, CarDark, CarDay*) that are aligned from left to right and from up to down. Some trackers may lose the target as their tracking boxes disappear in some frames.

III. Experiments

A. Experiment Initialization

Using the 8 traffic video sequences which are publicly available [21] and 2 datasets made by us, our proposed method (PHA) was compared with 5 state-of-the-art tracking methods: Visual Tracking via Dense Spatio-Temporal Context Learning (STC) [22], Compressive Tracking (CT) [23], Locally Orderless Tracking (LOT) [24], Object Tracking via Sparsity-based Collaborative Model (SCM) [25], L1 Tracker Using Accelerated Proximal Gradient Approach (L1APG) [26]. Captured in the scene of road, these video sequences in Table I contain various events including Illumination Variation (**IV**), Scale Variation (**SV**), Deformation (**DEF**), Motion Blur (**MB**), Fast Motion (**FM**), Background Clutters (**BC**) and Low Resolution (**LR**).

Same initializations, which mainly refer to the tracking box, were set to all methods for fair comparison and best parameters led each method to optimal performance. For the initial condition of the proposed tracker, it performs object localization using a sliding-window-search scheme with a search radius of 30 pixels. The average running time of our implementation by OpenCV is about 0.02 second per frame on a workstation with an Intel Core i5 3.0GHz processor and 8G RAM. For each tracking region, three kinds of features are extracted, including DCT low-frequency information, grey histogram and geometric moments. Precisely, all the images are divided into 8×8 cells, each of which is concerned with a 21-dimensional vector. Therefore, for each sliding window, we have a 128×21 feature matrix (64×21 tracking information and the same number of the context information), by means of which the mean-hashing algorithm generates 128×21 binary codes as identification. Note that all aforementioned parameters are changeless throughout the test of all datasets.

TABLE II: Center location error (CLE) (in pixels) (average at all frames). Red fonts indicates the best performance while the blue fonts indicates the second best one.

Datasets	Ours	CT	STC	L1-APG	LOT	SCM
Car	14	78	24	67	112	47
Car1	12	45	23	16	28	59
Car2	20	98	19	21	25	42
Car4	11	109	9	137	78	43
BlurCar1	10	92	7	12	11	98
BlurCar3	21	82	220	78	121	63
BlurCar4	31	76	19	23	56	61
CarDay	19	93	43	31	24	34
CarDark	4	21	5	2	7	17
CarDark2	24	103	40	42	49	67
Average CLE	17	80	40	43	51	53

B. Qualitative Performance Comparison

Figure 4 shows the qualitative tracking results of the 5 trackers overall representative frames of traffic video sequences. The specific dataset *CarLongDark*, totally 4412 frames, is a very long video which test the robustness and illumination variation. The Figure 5 shows that only our tracker succeeds in all the frames without losing the target.

C. Quantitative Performance Comparison

In the empirical comparison of trackers, two popular evaluation criteria are used: center location error (CLE) and VOC overlap ratio (VOR) between the predicted bounding box B_p

Fig. 5: Sample frames of a sequence owning 4412 frames. Although the scale, illumination and deformation changes a lot in some frames, the proposed tracker manages to recover the correct position.

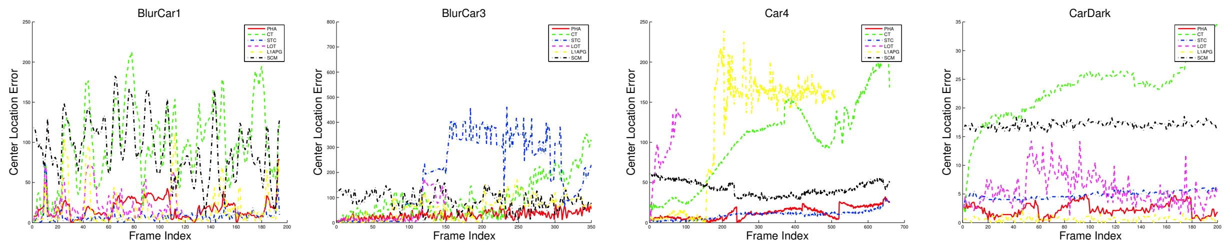

Fig. 6: Quantitative comparison of the 6 trackers *(PHA represents ours)* in CLE on the 4 video sequences. (The data that cannot cover all frames means to lose the target)

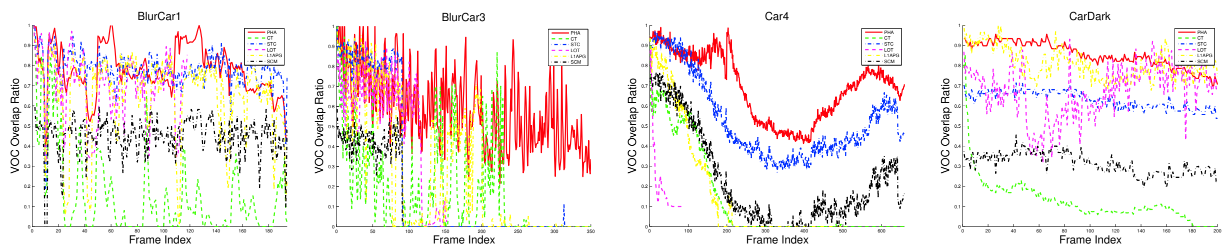

Fig. 7: Quantitative comparison of the 6 trackers *(PHA represents ours)* in VOR on the 4 video sequences.

and ground truth bounding box B_{gt}. The success rate refers to the situation when VOR is greater than 0.5.

$$VOR = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \tag{6}$$

To avoid accident, each video sequence is processed by every tracker twice. Table II and Table III show the quantitative results in which our tracker achieves the best or second performance in most sequences both in terms of center location error and success rate. By visualizing the concrete data, Figure 6 and 7 demonstrates the robustness of our method and better capability of vehicle tracking. Although STC performs excellent in the experiment, it may fall down such as *BlueCar3*. Furthermore, the average high score of VOR shows the good adaptability to complex and multivariate environment.

D. Effect Analysis

According to the aforementioned comparison in both quality and quantity, the proposed tracker have revealed excellent performance for vehicle tracking. Specifically, we clarify omnifarious evaluations corresponding to all classic tracking challenges.

Rotation and pose variation. The car in the *Car, Car2* and *CarDay* sequences undergoes in-plane rotation. In the *Car* sequence, when violent rotation variation happens (See *#1* and *#73*), STC, L1-APG and our method pass in the beginning. However, the performance of L1-APG gets worse as time passes by while STC fails at the next frames in the face of one single error. Thus, the geometric moments we introduce in the main theory make sense and finally only ours gets through the challenge.

TABLE III: Success rate (SR) (%).

Datasets	Ours	CT	STC	L1-APG	LOT	SCM
Car	73	20	60	30	22	40
Car1	83	38	75	80	72	45
Car2	86	15	100	100	46	37
Car4	82	15	40	19	9	18
BlurCar1	98	7	98	84	82	32
BlurCar3	63	31	25	29	72	30
BlurCar4	70	10	98	90	77	43
CarDay	86	45	75	80	85	60
CarDark	100	2	100	100	88	0
CarDark2	92	10	45	65	70	40
Average SR	83	20	71	67	62	39

Illumination, scale and pose variation. There are large illumination variations in the video sequences of *Car2, Car4* and *CarDark*. Especially, the appearance of the object in the *Car4* changes tremendously due to the ambient lights and shadows (See *#204* and *#342* in the *Car4* sequence). Both STC and our method adapt to these illumination variations well. Further to say, the excellent flexibility of illumination helps nighttime tracking a lot. Likewise, severe scale and pose variation happen in the *CarDay* and only our method performs favorably. Note that our method overcomes the difficulties of scale variation in the road scene just via sliding-window updating in these sequences. We don‘t introduce any features except geometric moments to solve the problem of scale-variation but the low-frequency information shows good suitability.

Background clutter and abrupt motion. In the *BlurCar1, BlurCar3* and *BlurCar4*, the running cars undergo fast movements and abrupt motion. In the *BlurCar1*, our proposed method achieves the best performance at nearly each frame (even indistinguishable *#11* and *#23*). Despite of the other tracker not losing the target, they cannot track the vehicle precisely during abrupt motion for they just make adjustments after the challenging variation. Hence, we consider the proposed method to be fast and robust.

IV. Conclusion

In this paper, we have proposed a robust visual tracker that learns the discriminative binary codes of every frame for the vehicle tracking system. To generate the single identification of each frame, a robust and effective perceptual hashing algorithm is applied. To imitate the perception of human visual system, we select the low-frequency, color and rotation information to obtain the perceptual feature. To perform the feature extraction, we build a perceptual algorithm based on DCT, grey histogram and geometric moments. In consideration of occlusion, we capture the feature of not only tracking object but also contexts. Compared with several progressive tracker, we empirically certify that our tracker is able to achieve more accurate and robust results in challenging complicated traffic video datasets. Thus, our tracker is very suitable for application in intelligent vehicle detection and tracking system. In the future, we are supposed to construct a framework for car perception using our tracker, which will play more practical and convincible roles in ITS.

V. Acknowledgment

The work described in this paper was supported by grants from National Natural Science Foundation of China (Grants No. 41401525), Natural Science Foundation of Guangdong Province (Grants No. 2014A030313209) and CRSRI Open Research Program (Program CKWV2014226/KY).

References

[1] J. Klein, C. Lecomte, and P. Miche, “Fast color-texture discrimination: Application to car tracking,” in *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, Sept 2007, pp. 546–551.

[2] H. Yazdi, M. Lotfizad, and M. Fathy, “Car tracking by quantised input lms, qx-lms algorithm in traffic scenes,” *Vision, Image and Signal Processing, IEE Proceedings -*, vol. 153, no. 1, pp. 37–45, Feb 2006.

[3] S. Hameed, O. Khalifa, M. Ershad, F. Zahudi, B. Sheyaa, and W. Asender, “Car monitoring, alerting and tracking model: Enhancement with mobility and database facilities,” in *Computer and Communication Engineering (ICCCE), 2010 International Conference on*, May 2010, pp. 1–5.

[4] H. Li, P. Bai, and H. Song, “Car tracking algorithm based on kalman filter and compressive tracking,” in *Image and Signal Processing (CISP), 2014 7th International Congress on*, Oct 2014, pp. 27–31.

[5] W. Yu, X. Tian, Z. Hou, A. Huang, and X. Liu, “Region edge histogram: A new feature for region-based tracking,” in *Signal Processing (ICSP), 2014 12th International Conference on*, Oct 2014, pp. 1180–1185.

[6] H. Cho, P. Rybski, and W. Zhang, “Vision-based bicycle detection and tracking using a deformable part model and an ekf algorithm,” in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, Sept 2010, pp. 1875–1880.

[7] J. Ohmer and N. Redding, “Gpu-accelerated klt tracking with monte-carlo-based feature reselection,” in *Digital Image Computing: Techniques and Applications (DICTA), 2008*, Dec 2008, pp. 234–241.

[8] K.-H. Seo and J.-J. Lee, “Real-time object tracking and segmentation using adaptive color snake model,” in *Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE*, Nov 2005, pp. 5 pp.–.

[9] A. Adam, E. Rivlin, and I. Shimshoni, “Robust fragments-based tracking using the integral histogram,” in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, June 2006, pp. 798–805.

[10] S. Bai, R. Liu, Z. Su, C. Zhang, and W. Jin, “Incremental robust local dictionary learning for visual tracking,” in *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, July 2014, pp. 1–6.

[11] B. Zeisl, C. Leistner, A. Saffari, and H. Bischof, “On-line semi-supervised multiple-instance boosting,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 1879–1879.

[12] Z. Kalal, J. Matas, and K. Mikolajczyk, “P-n learning: Bootstrapping binary classifiers by structural constraints,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 49–56.

[13] J. Kwon and K. M. Lee, “Visual tracking decomposition,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 1269–1276.

[14] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, “Robust tracking using local sparse appearance model and k-selection,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, June 2011, pp. 1313–1320.

[15] S. Hare, A. Saffari, and P. Torr, “Struck: Structured output tracking with kernels,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 263–270.

[16] J. Kwon and K. M. Lee, “Tracking by sampling trackers,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 1195–1202.

[17] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, “Robust visual tracking via multi-task sparse learning,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 2042–2049.

[18] O. J. L. P.-f. Wen Zhen-kun, Zhu Wei-zong, “A robust and discriminative image perceptual hash algorithm,” *Genetic and Evolutionary Computing (ICGEC), 2010 Fourth International Conference on*, 2010.

[19] X.-X. N. Bin Zhang, Yang Xin, “Image perceptual hash algorithm based on target character,” *International Conference on Communication Technology (ICCT)*, 2011.

[20] Y. Hua, B. Xiao, D. Feng, and B. Yu, “Bounded lsh for similarity search in peer-to-peer file systems,” in *Parallel Processing, 2008. ICPP ’08. 37th International Conference on*, Sept 2008, pp. 644–651.

[21] Y. Wu, J. Lim, and M.-H. Yang, “Online object tracking: A benchmark,” in *CVPR*, 2013.

[22] Q. L. D. Z. Kaihua Zhang, Lei Zhang and M.-H. Yang, “Fast visual tracking via dense spatio-temporal context learning,” *European Conference on Computer Vision*, 2014.

[23] K. Zhang, L. Zhang, and M.-H. Yang, “Fast compressive tracking,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 10, pp. 2002–2015, Oct 2014.

[24] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, “Locally orderless tracking,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 1940–1947.

[25] W. Zhong, H. Lu, and M.-H. Yang, “Robust object tracking via sparsity-based collaborative model,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 1838–1845.

[26] C. Bao, Y. Wu, H. Ling, and H. Ji, “Real time robust l1 tracker using accelerated proximal gradient approach,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 1830–1837.