Vehicle Semantics Extraction and Retrieval for Long-term Carpark Video Surveillance

Clarence Weihan Cheong (✉)[1], Ryan Woei-Sheng Lim[1], John See[1], Lai-Kuan Wong[1], Ian K.T. Tan[1], and Azrin Aris[2]

[1] Center for Visual Computing, Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Malaysia,
clarence_han@hotmail.com, ryanlim0616@gmail.com, johnsee@mmu.edu.my, lkwong@mmu.edu.my, ian@mmu.edu.my

[2] VADS Lyfe, Telekom Malaysia Berhad, 60000 Kuala Lumpur,
azrin.aris@tm.com.my

Abstract. Car park video surveillance data provides plenty of semantic rich data such as vehicle color, trajectory, speed, and type which can be tapped into and extracted for video and data analytics. We present methods for extracting and retrieving color and motion semantics from long term carpark video surveillance. This is a challenging task in outdoor scenarios due to ever-changing illumination and weather conditions, while retrieval time also increases as data size grows. To address these challenges, we subdivided the search space into smaller chunks by introducing spatio-temporal cubes or *atoms*, which can store and retrieve these semantics at ease. The proposed method was tested on 2 days of continuous data from an outdoor carpark under various lighting and weather conditions. We report the precision, recall and F_1 scores to determine the overall performance of the system.

Keywords: Vehicle Semantic Extraction, Retrieval Systems, Carpark Surveillance

1 Introduction

The use of video-based traffic surveillance is becoming increasingly popular due to the low implementation cost. However, majority of these data are left unprocessed and kept in storage devices. Rich semantic data such as vehicle color, trajectory and type can be exploited for video and data analytics to provide deeper insights for surveillance and retrieval purposes.

Traditionally, to perform retrieval on surveillance videos, users need to provide description of the vehicle such as the time, place of the incident, vehicle registration plate, and color of the vehicle. Next, users would filter through all the retrieved results to manually identify the target event. This entire process is undoubtedly time consuming and labor intensive.

To overcome the inefficiency of such laborious methods, we propose a long-term surveillance analytics system that extracts and stores the semantics data

into a database and allow video clips of specific events to be retrieved using user-described queries.

First, our method performs background subtraction to extract foreground blobs that represents vehicles. Next, a filtering process is applied to remove any unwanted blobs such as pedestrians. They are then tracked frame by frame to generate their individual trajectories and to extract vehicle-specific semantics. Lastly, these semantics are segmented into spatio-temporal cubes (atoms) and stored in the database. We evaluated the reliability and performance of the proposed method over a span of 20 hours.

2 Related Works

While Intelligent Transportation System (ITS) is a popular research topic, there has been little research done for carpark scenes. When viewed regardless of the intended scene, vehicle semantics extraction and trajectory retrieval for surveillance video is a wide research field.

For vehicle color semantics, there are many different school of thoughts that arise from it. Authors in [2] and [5] approach this challenge by obtaining the histogram in HSV/HSL color space while other works [4, 6, 9] addressed it by designing deep learning methods. In an interesting work [8], color spaces were also used to detect moving shadows from urban surveillance video.

In the area of vehicle motion extraction and trajectory grouping, the authors in [11] and [2] quantized the moving direction of the objects into 4 and 9 directional bins respectively. In [3], a novel method of indexing trajectories into spatio-temporal cubes is introduced. A recent work by Castañón et al. [2] used other vehicle semantics such as size and persistence to query for anomalous and typical events.

3 Framework

The framework of our proposed method adheres to the typical top-down approach for automated video surveillance in carparks [7] which includes background subtraction, blob filtering, vehicle detection vehicle tracking. The semantic information from these vehicle blobs are extracted, segmented into atom-based cubes and stored in the database. This information can then be queried through a search interface. The overview of our framework is shown in Figure 1.

3.1 Background Subtraction, Vehicle Detection and Tracking

We describe a number of preparatory steps that were taken prior to the extraction of semantics. Firstly, background subtraction with a combination of adaptive learning and frame differencing [7] is performed to extract foreground blobs from each video frame. This strategy is computationally cheaper than optical flow, hence it improves on the overall efficiency of segmenting moving objects.

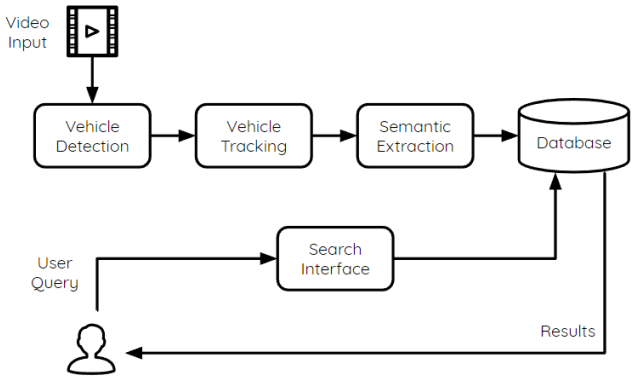


Fig. 1: Framework diagram

Since the focus is on carpark surveillance where large portions of the video footages may not contain any substantial movement, a frame skipping method was deployed to speed up the overall processing. The blobs then go through a series of morphological operations such as dilation and erosion to filter out noise and to fill up gaps in the blobs to generate the final foreground blobs.

Next, the blobs are filtered according to their sizes, positions and aspect ratios, where each of these parameters were determined empirically to suit the scene geometry. After that, the YOLO real-time object detector [10] is applied to differentiate between vehicles or non-vehicle blobs. This two-step approach is designed to filter out objects other than vehicles that are not of interest such as pedestrians and motorcycles. Finally, each blob is matched back to the trajectories using a tracking state machine proposed in [7].

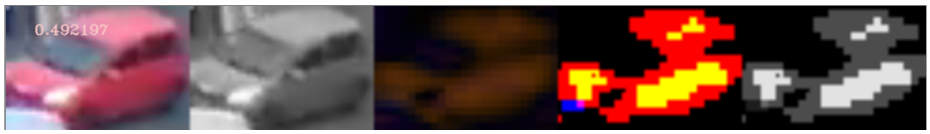
3.2 Object Specific Semantic Extraction

As object specific semantics from the scene provides deeper insights for surveillance purposes, this work currently focuses on two types of object specific semantics, namely the color and motion information.

I. Color information plays a significant role in the retrieval process as it is often one of the most common information given when a user tries to describe an object from an event in a scene. Extracting color information accurately is particularly challenging for outdoor scenes as the color information varies throughout the day due to ambient illumination and weather changes. Algorithm 1 summarizes our strategy for extracting color information.

When a vehicle is detected in the scene, a bounding box of the foreground blob is usually used to mark the location of the vehicle. However, due to the background subtraction method used, the final foreground blob appears slightly larger than the actual footprint of the vehicle. In order to obtain a closer estimation of the vehicle's dominant color, the bounding box is cropped by 30% to reduce some background information such as the road or vehicles around it.

(a) Achromatic vehicle (Gray/White)



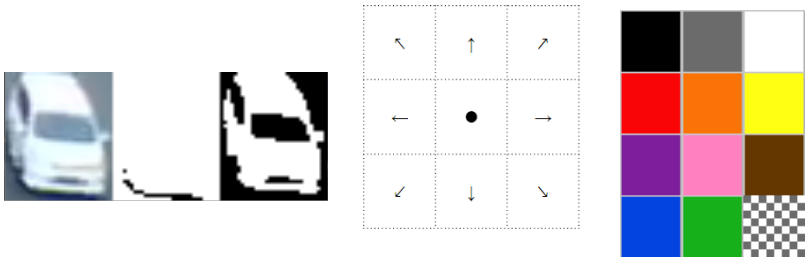
(b) Chromatic vehicle (Red)

Fig. 2: (From left) Original image; Grayscale image; Absolute difference; Binary threshold absolute difference; Threshold difference in grayscale

Subsequently, our strategy determines the dominant color of each vehicle blob by first undertaking a task to determine if the vehicle's dominant color belongs to the achromatic scale (black, gray and white color) or chromatic scale (other hues). We determine the absolute difference between the cropped image and its grayscale version, and then threshold each channel in RGB at an empirically-found intensity value of 35. The hint of significant values from this step indicates a substantial presence of chromatic hue. Then, we convert the thresholded image to a grayscale image, and determine the ratio of non-zero pixel values over total pixels. This process allows us to deduce the presence of strong chromatic hues and estimate if the vehicle belongs to the achromatic or chromatic subsets. A threshold pivot, T_{pivot} is empirically set at the 0.18 where if the ratio of non-zero pixel values is more than T_{pivot}, we can assume that the particular vehicle blob contains a strong chromatic hue, as illustrated in Figure 2.

Achromatic and chromatic color processing. Upon determining if the vehicle belongs to the achromatic scale, we then subjected the cropped image to both the black and white filters individually by applying binary thresholds set at empirically determined intensity levels of 50 and 170 respectively. Next, in similar fashion, the ratio of non-zero pixels upon filtering is used to determine if the vehicle is assigned to black, white or gray color term. Figure 3a shows how a white vehicle responds to a black and white filter.

As for the chromatic colors, we chose to utilize the HSV color space as it is visually more intuitive than the RGB color space. Here, we generated a 3-dimensional HSV histogram with 15 Hue bins, 8 Saturation bins and 8 Value bins. Based on the generated histogram, the maximum value of each bin from all 3 channels is assumed to correspond to the dominant color of the vehicle. However, since the vehicle is moving in the outdoor scene, the ambient and directional lighting (from sun and other light sources) contribute to slight variation of colors. To suit our problem, the dominant color for each frame of a tracked vehicle is averaged out throughout its trajectory.



(a) (b) (c)

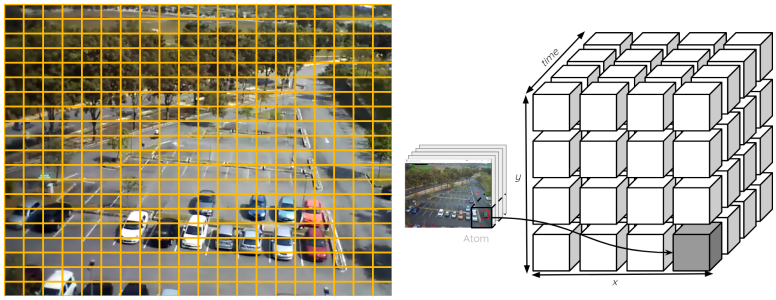
Fig. 3: (a) Black & white filter responses, (b) Directional bins, (c) 11 color categories [1]

Algorithm 1 Color Term Extraction

```
for Each blob in object do
    Shrink bounding box (crop image)
    Create a copy of the cropped image in grayscale
    Calculate absolute difference between cropped image & grayscale image
    Perform threshold on absolute difference to amplify difference
    Convert results into grayscale & calculate no. of non-zero pixels
    if Ratio of non-zero pixels > T_pivot then
        Calculate 3D HSV histogram                    //Chromatic Vehicle
        Locate maximum bin location of each channel
        Map the highest bin from each channel to Color Term
    else
        Perform black & white filter                  //Achromatic Vehicle
        Obtain ratio of non-zero pixels from both filters
        Determine Color Term
    end if
end for
Obtain average dominant color & return Color Term
```

We also note that the achromatic algorithm is an essential step because the 8 Values bins are insufficient to accurately represent vehicles with borderline dominant color as the brightness values may be widely distributed.

Color terms. Next, we addressed the problem of defining color terms by adopting the eleven common terms in English as described by a study done in 1969 by Berlin and Kay [1]. The color categories are white, black, red, green, yellow, blue, brown, purple, pink, orange, and gray. This definition enables us to quantize the range of colors to a fixed number of color categories while taking advantage of the atom-based structure (Refer to Section 3.3).



(a) 2D Atom Grids (b) Atom

Fig. 4: Atom Structure

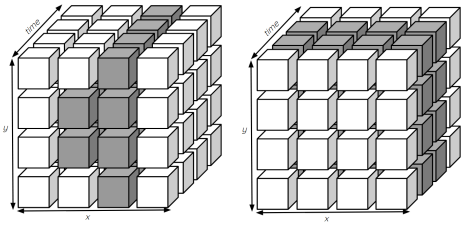
II. Motion information also plays an important role in the retrieval process as users would often describe the trajectory of a vehicle from a particular incident. Instead of generating a fine representation of the vehicle's trajectory (conventional motion vectors), we can store a coarser representation of motion information in the form of directional categories. We achieve this by quantizing the extracted motion information of the vehicle trajectories into 9 bins – 8 directional bins as well as one bin to denote minuscule and negligible motion, as shown in Figure 3b. The motion vectors are extracted from the centroid of the vehicle, with respect to its previous location one second ago; a minimum displacement of 5 pixels determines the presence of motion.

The advantage of such a method is that we are able to fully utilize the atom-based structure (Refer to Section 3.3) to locate motions of interest in an efficient manner. With this approach, we do not have to consider the various combinations of fine-grained motion trajectories which may occur in realistic outdoor scenes such as carparks.

3.3 Semantic Segmentation and Indexing

I. Semantic Segmentation In the proposed method, we adopted the concept of using video cubes or *atoms* from [2] as a high-level data structure that frames the data into a spatio-temporal search space. An atom is defined as a group of cells at a similar spatial location, that spans a certain fixed number of frames; hence forming a spatio-temporal 'cube'. Figure 4b illustrates the atom structure. Contrary to [2] which uses these atoms in a tree-like structure that associates each atom with its neighbors as the child nodes, we consider each atom discretely and independently of one another.

Since our video data (see Section 4.1) has a resolution of 640 $\times$ 480 pixels and frame rate of $10fps$, we analytically set the dimensions of each atom, α to $\alpha_{width} = 32$ pixels, $\alpha_{height} = 24$ pixels and $\alpha_t = 10$ frames, which represents the temporal duration of one second. We selected the resolution of the atom $(\alpha_{width}, \alpha_{height})$ as such so that the video resolution can be uniformly divide our video into 20 atoms across both its width and height. This approach allows us



(a) Region of Interest (b) Time Slicing

Fig. 5: Types of Queries

to distinctly identify each atom from a video by an index $(\alpha_{width}, \alpha_{height}, \alpha_t)$ (see Figure 4). Similarly, we can store specific occurrences of each semantic type (color, motion) with the same atom index. Though a vehicle blob may be encapsulated by several neighboring atoms, only the atom which corresponds to the centroid of the blob is stored. This is done with the consideration that the precise bounding box location is not essential for retrieving video shots.

II. Semantic Indexing In order to index the extracted data, we borrowed the idea similar to that of Locality-sensitive Hashing (LSH) used in [2] by grouping similar semantics together for quicker retrieval. Our proposed method espouses this by creating unique database tables for each of the 20 semantics (11 colors and 9 motion bins) with the source video, vehicle ID along with the individual atom indices as columns for each table. This allows us to rapidly locate specific atoms with the queried semantic without going through the entire database, as illustrated in Figure 5. The atom structure enables us to make queries based on a specific region of interest, or time slice.

3.4 Semantic Retrieval

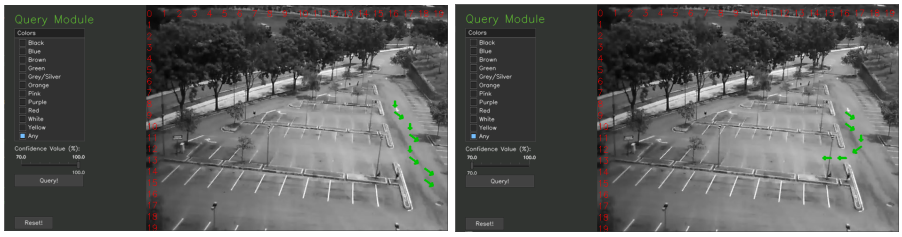
When a color query $\mathbb{C}$ is issued, all vehicles with matching colors are returned. When a trajectory query $\mathbb{Q}$ is issued, the possible atoms that are matched will be retrieved in temporal order. These retrieved atoms need to be merged to build video shots that will be returned. Assume $\mathbb{A} = \{\alpha_n, \alpha_{n+1}, \ldots, \alpha_N\}$ is a time-ordered set of retrieved atoms, we intend to piece together relevant atoms to build video shots $\mathbb{S}_i$. This is achieved by performing atom merging if the following condition is met:

$$\alpha_{n+1} \subset \mathbb{S}_i \qquad \textbf{if} \quad t_{\alpha_{n+1}} - t_{\alpha_n} < (5 * f) \tag{1}$$

where f is the video frame rate.

We also introduce a Confidence Value (CV) which sets the sensitivity level of accepting a video shot as among the retrieved results. For each shot, we accept each retrieved shot $\mathbb{S}_i$ if it fulfills the following condition:

$$CV < \frac{\text{length}(\mathbb{S}_i)}{\text{length}(\mathbb{Q})} \times 100\% \tag{2}$$



(a) Motion Test Case 1 (TQ1) (b) Motion Test Case 2 (TQ2)

Fig. 6: Search interface for the proposed framework

This provides a margin of error when performing the query which acts as a trade-off function. A lower CV results in returning a larger set of results but at the expense of an increase in retrieved shots, and vice versa.

Search Interface. The proposed methods were realized in a form of a search interface, which was designed to allow users to construct a query by tracing the trajectory and selecting colors which fit their intended vehicle description. Figure 6 shows the interface, with the green lines showing the user-selected trajectory query. The underlying atom-based structure allows queries to be formed in a way which emulates the semantics extraction process, eliminating the need for query parsing.

4 Experiments

4.1 Dataset

This section describes the video data used in the development and evaluation of the proposed method. We collected a new video dataset consisting of videos recorded from a university's outdoor carpark area over a duration of several months. A single stationary camera was set up to record the video on weekdays throughout the week from 8:30AM to 6:30PM. These videos were recorded in a compressed H.264 MPEG-4 format with a resolution of 640 $\times$ 480 pixels and frame rate of $10fps$. Figure 7a shows a wide range of challenges found in the recorded video data: severe morning and afternoon shadows, rainy weather, and reflections. Due to the scale of experiment, we selected 2 days of video data (totaling 20 hours) for the work in this paper.

4.2 Experiment Methodology

To validate our method for vehicle color and motion retrieval, the ground truth states were manually labeled by a few annotators and cross-checked to arrive at a consensus. This allows us to validate the efficacy of our proposed automated method against human observations. Our system was implemented on an Intel i7 machine with 16GB RAM, GeForce GTX 1060 GPU. In order to analyze both color and motion semantics individually they are evaluated separately to measure their individual performances.

Fig. 7: (a) Various carpark scene challenges throughout the day (severe shadow, weather condition and reflections) (b) Sample screenshots of 3 retrieved shots (left, center, right columns) for query TQ2 (turning into junction)

Table 1: Ground truth distribution vehicle colors ordered by occurrence

Color	Gray	Black	White	Red	Blue	Orange	Yellow	Green	Pink	Purple	Brown
#	365	182	150	60	19	15	13	10	9	7	7
%	43.6	21.7	17.9	7.2	2.3	1.8	1.6	1.2	1.1	0.8	0.8

Color Retrieval

To measure the performance of the color retrieval module, we follow through the pipeline by extracting unique objects from each of the 11 color tables. The retrieved results are then compared against the ground truth. The ground truth distribution of vehicle colors is shown in Table 1.

Motion Retrieval

We specified 2 specific motion paths or trajectory queries (TQ) that we intend to validate: TQ1) Heading southward (see Fig. 6a) & TQ2) Turning in a junction (see Fig. 6b). The distribution of these test cases are 252 (86.3%) and 40 (13.7 %) trajectories for TQ1 and TQ2 respectively.

To measure the performance of the motion retrieval method, we performed evaluation on both TQ1 and TQ2 without consideration for the vehicle color. These experiments were tested on a few CV values (70%, 80%, 90%) and different number of atom query inputs to test the impact of trajectory details.

Evaluation metrics: We used 3 evaluation metrics - Precision, Recall as well as the F_1 score to determine the overall performance of the system. Correct matches will be regarded as true positives, tp. False positives, fp is the total number of retrieved results minus the true positives, while false negatives, fn is the total number of correct results minus true positives. Precision, Recall and F_1 score is computed as:

$$\text{Precision} = \frac{tp}{tp+fp} \; ; \; \text{Recall} = \frac{tp}{tp+fn} \; ; \; \text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

4.3 Experiment Results and Discussion

The performance of the proposed method is computed by comparing the annotated ground truth against the retrieved results.

Color Retrieval

Table 2 reports the confusion matrix of the color retrieval task. The overall precision stands at 54%, with a recall of 36% and F_1 score of 39%. The cells in Table 2 marked in green indicate the highest count of correctly predicted colors, while the cells marked in red indicate the highest count of incorrect predictions for each color. Our method is able to predict correctly a majority of cases for seven out of eleven colors.

Based on the obtained results, we learn that the T_{pivot} needs to be adjusted as too many chromatic vehicles were classified as achromatic vehicles which in turned affected the overall performance of the proposed method. We hypothesize that better results can be obtained by careful adjustment of T_{pivot} or attempt to learn a suitable color model as in [6, 9] for this particular scene. We observe that vehicles with lighter and darker shades were particularly difficult as they do not contain enough chromatic hues to arrive at a correct prediction.

While the classification of achromatic versus chromatic vehicles faced considerable difficulties, the black & white filters provide a considerably good result when determining the different categories of achromatic vehicles which may be useful for processing in grayscale. Based on our observation, most errors usually occur when the vehicles are in locations where the intensity of shadows overpowered the lighter-shade vehicles in terms of coverage area.

Table 2: Confusion matrix for color retrieval task

		Predicted Color										
		Gray	Black	White	Red	Blue	Orange	Yellow	Green	Pink	Purple	Brown
Actual Color	Gray	**236**	61	68	0	0	0	0	0	0	0	0
	Black	48	**134**	0	0	0	0	0	0	0	0	0
	White	26	4	**120**	0	0	0	0	0	0	0	0
	Red	27	25	0	2	0	0	0	0	4	**2**	0
	Blue	3	10	0	0	**6**	0	0	0	0	0	0
	Orange	8	3	0	0	0	**3**	0	0	0	1	0
	Yellow	3	1	2	0	0	0	**7**	0	0	0	0
	Green	5	1	4	0	0	0	0	**0**	0	0	0
	Pink	1	0	0	**3**	0	0	0	0	**5**	0	0
	Purple	3	3	0	0	0	0	0	0	0	1	0
	Brown	3	4	0	0	0	0	0	0	0	0	**0**

Result	Precision	65.01	54.47	61.86	40.00	100.00	100.00	100.00	N/A	55.56	25.00	N/A
	Recall	64.66	73.63	80.00	3.33	31.58	20.00	53.85	0.00	55.56	14.29	0.00
	F1 Score	64.84	62.62	69.77	6.15	48.00	33.33	70.00	N/A	55.56	18.18	N/A

Motion Retrieval

The retrieved trajectory shots are validated against the annotated ground truth labels to obtain our results. As it is difficult to pinpoint the exact "scene" where the test cases occur, we use a time window of $\pm T$ seconds to indicate a range whereby a retrieved motion can be correctly matched to the ground truth label. We fixed $T = 5$ in our experiments, similar to that used in the tracking evaluation of [7].

Table 3 shows the results of the motion retrieval task with varying Confidence Values (CV) and varying number of atom inputs in the trajectory query. For TQ1, the total number of atom inputs varied from 5 to 8 inputs while TQ2 is represented by a shorter trajectory of 4 to 6 inputs as it concerns a junction turning query. For TQ1 & TQ2, the overall average precision is around 89% & 50% while the recall is at 27% & 59% respectively. Based on the F_1 scores, the results show that the proposed retrieval method works best when the CV is at the lowest (70%) with an atom input length of 5. Figure 7b shows some sample snapshots representative of the retrieved shots for TQ2.

We analyzed these results from various perspectives and we find that our proposed method performs reasonably well at retrieving a user described trajectory motion at high precision, but at the cost of a lower recall rate when CV increases. This is likely due to its over-sensitivity towards the exact query given. From the experiment, we also learnt that the queries should be expanded to include neighboring atoms so as to provide a better chance at obtaining a higher recall rate with good precision. This appeals towards the subjective nature of trajectory-based querying where the users of such an interface would naturally draw a general direction of the query instead of a precise path.

Table 3: Results of motion retrieval task with varying CV and number of atom inputs

			CV: 70%			CV: 80%			CV: 90%		
			Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
No. of Input	TQ1	5	93.82	61.53	**74.32**	95.34	33.19	49.24	95.34	33.19	49.24
		6	90.09	36.84	52.29	90.09	36.84	52.29	89.13	16.59	27.98
		7	87.27	38.86	53.78	88	17.81	29.62	87.87	11.74	20.71
		8	86.88	21.45	34.41	84.61	13.36	23.07	89.65	10.52	18.84
	TQ2	4	16.28	80	27.06	28.69	73.33	41.25	28.69	73.33	41.25
		5	65.3	71.11	**68.08**	73.33	48.88	58.66	73.33	48.88	58.66
		6	55.81	53.33	54.54	55.81	53.33	54.54	57.69	33.33	42.25

5 Conclusion and future work

This paper proposes a framework for extracting and retrieving color and motion semantics from an outdoor long-term car park setting. We demonstrated

methods that were able to retrieve queries to a good measure of precision under various lighting and weather conditions. However, there is room for improvement in the recall ability for both the color and motion semantics.

Our future directions are aimed at fine tuning the proposed method for better performance over a longer span of time, i.e. weeks or months. With that, alternative methods that are more data-dependent may be plausible, such as learning a scalable color term extraction model.

Acknowledgment

This work is supported in part by Telekom Malaysia Research & Development Grant No. RDTC/160903 (SHERLOCK) and Multimedia University.

References

1. Brent, B., Kay, P.: Basic color terms: Their universality and evolution. Univ of California Press (1991)
2. Castañón, G., Elgharib, M., Saligrama, V., Jodoin, P.M.: Retrieval in long-surveillance videos using user-described motion and object attributes. IEEE Transactions on Circuits and Systems for Video Technology 26(12), 2313–2327 (2016)
3. d'Acierno, A., Saggese, A., Vento, M.: Designing huge repositories of moving vehicles trajectories for efficient extraction of semantic data. IEEE Transactions on Intelligent Transportation Systems 16(4), 2038–2049 (2015)
4. Dehghan, A., Masood, S.Z., Shu, G., Ortiz, E., et al.: View independent vehicle make, model and color recognition using convolutional neural network. arXiv preprint arXiv:1702.01721 (2017)
5. Feris, R.S., Siddiquie, B., Petterson, J., Zhai, Y., Datta, A., Brown, L.M., Pankanti, S.: Large-scale vehicle detection, indexing, and search in urban surveillance videos. IEEE Transactions on Multimedia 14(1), 28–42 (2012)
6. Hu, C., Bai, X., Qi, L., Chen, P., Xue, G., Mei, L.: Vehicle color recognition with spatial pyramid deep learning. IEEE Transactions on Intelligent Transportation Systems 16(5), 2925–2934 (2015)
7. Lim, R.W.S., Cheong, C.W., See, J., Tan, I.K.T., Wong, L.K., Khor, H.Q.: On vehicle state tracking for long-term carpark video surveillance. In: IEEE Int. Conf. on Signal and Image Processing Applications. To appear (2017)
8. Moghimi, M.K., Pourghassem, H.: Shadow detection based on combinations of hsv color space and orthogonal transformation in surveillance videos. In: Intelligent Systems (ICIS), 2014 Iranian Conference on. pp. 1–6. IEEE (2014)
9. Rachmadi, R.F., Purnama, I.: Vehicle color recognition using convolutional neural network. arXiv preprint arXiv:1510.07391 (2015)
10. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788 (2016)
11. Zhang, T., Liu, S., Xu, C., Lu, H.: Mining semantic context information for intelligent video surveillance of traffic scenes. IEEE transactions on industrial informatics 9(1), 149–160 (2013)