Moving Shadow Removal for Multi-Objects Tracking in Outdoor Environments

Mohamed Taha[1], Hala H. Zayed[2], M. E. Khalifa[3] and Taymoor Nazmy[4]
[1, 2] Faculty of Computers & Informatics, Benha University, Benha, Egypt
[3, 4] Faculty of Computer & Information Sciences, Ain Shams University, Cairo, Egypt

ABSTRACT
Shadow detection and removal has had great interest in computer vision especially in outdoor environments. It is an important task for visual tracking, object recognition, and many other important applications. One of the fundamental challenges for accurate tracking is achieving invariance to shadows. Two or more separate objects can appear to be connected through shadows. Many algorithms have been proposed in the literature that deal with shadows. However, the problem remains largely unsolved and needs further research effort.

This paper proposes a method for removing cast shadows from vehicles in outdoor environments. The proposed method employs the estimated background model of the video sequence and applies a Gamma decoding followed by a thresholding operation. Experimental results show the success of the proposed method in detecting and removing shadows robustly and leads to considerable improvements in multiple object tracking.

General Terms
Computer Vision, Pattern Recognition.

Keywords
Shadow Removal, Shadow Detection, Moving Shadow Removal, Object Tracking

1. INTRODUCTION
The advance of technology makes video acquisition devices better and less costly, hence increasing the number of applications that can effectively utilize digital video. Compared to still images, video sequences provide more information about how objects and scenarios change over time. Over the past years, many interesting applications of traffic monitoring and vehicle tracking using roadside cameras have been proposed and shown to be useful in many situations [1-3]. Vehicle tracking systems are presented in order to solve many traffic problems like vehicle routing [4] and accidents detection [5] to name a few. However, their effectiveness depends on video image processing algorithms that are capable of reducing common problems such as shadows, occlusion, illumination, reflection, and camera shaking. Among of them, shadows have proven to be a large source of error in vehicles detection and classification [6, 7]. It is not difficult for human eyes to distinguish shadows from objects. However, identifying shadows by computers is a challenging research problem. Shadow is a region of relative darkness that occurs when an object totally or partially occludes direct light from a light source [8]. Typically, there are two types of shadow: 'self' and 'cast'; self-shadow occurs on the object occluding the light, while cast-shadow is that generated by the object on the ground or other objects in the scene. Figure 1 shows the difference between the self-shadow (marked by red border) and cast-shadow (marked by blue border). Unfortunately, the similar characteristics between cast and self-shadows often complicate the computer's ability to distinguish between them.

Fig 1: Self-shadow and cast-shadow

In many computer vision applications, cast-shadow is more important than self-shadow which is usually identified as some pixels of the object itself. Cast-shadow can bring serious problems while extracting moving objects due to the misclassification of shadow points as foreground [9]. In some cases when the shadows stretch, two or more independent objects can appear to be connected together. Shadow can cause merging of objects, object shape distortion and object losses. Figure 2 shows a merging problem of two objects in a tracking process. It occurs as a result of the fusion of the right vehicle shadow with the left vehicle.

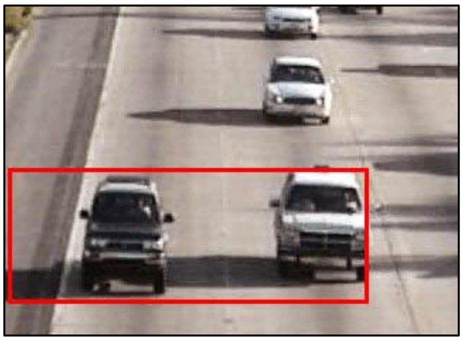


Fig 2: Two vehicles appear as one object in the tracking process because of shadow stretching

Understanding the content of images and videos requires detecting and tracking moving objects. When the objects of interest have a well-defined shape, template matching or more sophisticated classifiers can be used to directly separate the objects from the image [10]. The difficulties associated with shadow detection arise since shadows and objects share two important visual features: First, shadow pixels are detectable as foreground pixels as they are typically different from the background; second, shadows have the same motion pattern of the objects casting them [8, 9, 11]. In fact, shadow identification is a serious problem for both still images and image sequences (video) and has become an active research area [10].

It should be noted that the purpose for which shadow removal is utilized determines the restrictions of how the problem is solved. For example if the goal is just image enhancement, it can be allowed to have some guidance by a user to remove shadows. This will simplify the problem. However, in computer vision domain, the purpose is usually to extract meaningful information from an image. The artistic nature of the image is not so important as long as the resulting image is shadow free and the core information of the original image is kept. In this case, the shadow removal should be done automatically without any user intervention.

In case of still images [12], shadows are commonly analyzed to estimate geometric properties of the objects causing the shadow, and to enhance object localization and measurements. This is helpful in many applications such as in aerial image analysis for recognizing buildings [13, 14], and for obtaining 3-D reconstruction of the scene [15, 16]. In addition, another significant application is in 3-D analysis of objects to extract surface orientations [17] and to determine light source direction [18]. However, in the case of image sequences, shadows are referred to as moving shadows to differentiate them from that in still images [19] and the detection approach and steps are usually different from still images. In addition, the purposes of shadow detection in image sequences could be: change detection, scene matching or surveillance which is different from the detection purpose of still images. In moving shadow detection methods, generally a background subtraction step is usually present [19, 20]. There are several studies on moving shadow detection [19 - 26]. In vision-based vehicle detection systems, differentiating cast shadows from moving objects is significant and remains largely unsolved [27]. In this paper, a method for removing cast shadow from vehicles in outdoor environments is proposed. The method works by applying gamma decoding followed by a thresholding operation and employing an estimated background model of the video sequence.

The paper is organized as follows: Section 2 explores the related work found in the literature concerning cast shadow detection. Section 3 presents the proposed method in details. Section 4 discusses the experimental results and the performance evaluation of the proposed method. Finally, the conclusion is presented in Section 5.

2. RELATED WORK

The area of shadow detection and removal has made great progress in recent years. There has been significant work done lately that deals with the problem of moving cast shadows. In 2003, Prati et al. [11] conducted a survey on moving shadow detection algorithms. They categorized shadow detection methods in a two-layer taxonomy as shown in Figure 3. The first layer classification is algorithm-based taxonomy. It considers whether the decision process in the algorithm introduces and exploits uncertainty. The second layer classification is feature-based taxonomy where they mentioned the types of features used by each method among three broad classes: spectral, spatial and temporal features. Their main conclusion was that the simplest method was the one convenient for generalization. Moreover, to detect shadows efficiently in a specific environment, more assumptions yield better results. Consequently, there was no single robust shadow detection technique and it was better for each particular application to develop its own technique according to its nature.

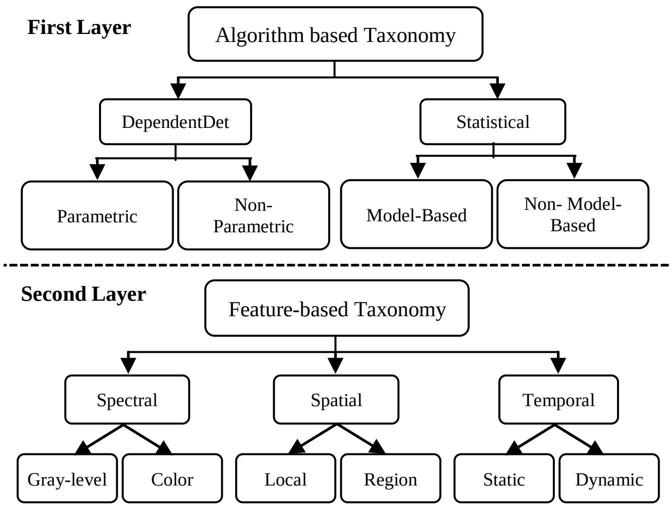


Fig 3: Prati [11] taxonomy of shadow detection algorithms

Sanin et al. [10] conducted a more recent review of shadow detection and removal techniques in 2012. The survey follows the work of Prati et al. [11] but with more recent publications, and a more comprehensive set of test sequences and experiments. In contrast to Prati review, Sanin review categorized shadow detection methods in a feature-based taxonomy into six classes: intensity, chromacity, physical properties, geometry, textures, and temporal features. They observed that the choice of features has greater impact on shadow detection results compared to the choice of algorithms.

Sanin evaluation [10] indicates that all shadow detection approaches have individual strengths and weaknesses. Out of these methods, the geometry-based technique has severe assumptions and is not generalizable to various environments, but it is an obvious choice when the objects of interest are easy to model and their shadows have different orientations. The chromacity-based method is simple to implement and computationally inexpensive, but it is sensitive to noise and less effective in low saturated scenes. The physical method tend to be more accurate than chromacity-based methods, but fails with objects having similar chromacity to that of the background. The small-region texture based method is potentially powerful for pixels whose neighborhood is textured, but may take longer to implement and is the most computationally expensive. The large-region texture based method produces the most accurate results, but has a significant computational load due to its multiple processing steps.

In addition, Al-Najdawi et al. [8] presented another survey of shadow detection methods in 2012. The algorithms are classified under a four-layer taxonomy based on object and environment dependency and domain of implementation. A further classification into monochrome/color, for pixel domain algorithms is given (see Figure 4). Performance of algorithms in different domain of implementations is compared quantitatively and qualitatively. Their conclusions significantly agreed with what Sanin et al. conclude but with some more additions. Their results reported that transform domain algorithms can be more robust to noise and are less complex as the number of features extracted are fewer and more precise. However, their performance is limited by the flexibility and the sharpness of shadow detected.

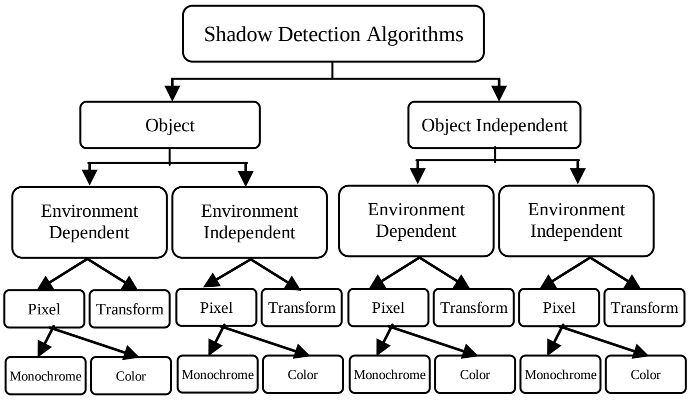


Fig 4: Al-Najdawi [8] taxonomy of shadow detection algorithms

3. THE PROPOSED METHOD

When studying the properties of shadow regions in terms of their features including intensity, color, and edge information, one can find that the intensity of a shadow pixel decreases when compared to the reference background [27]. In addition, the reduction rate changes smoothly between neighboring pixels. For the color feature when considering the HSV color space, both the hue component and the saturation component of shadow pixels change but within a certain limit [27]. In RGB color space, although all RGB values are lower in the shadow region than in the background region, the amount of reduction is not proportional. Shadow pixels are more saturated towards blue. As a result, shadow pixels falling on neutral surfaces, such as asphalt roads, tend to be more bluish. For the edge features, shadow regions generally do not have strong edges. Compared with the reference background, edge magnitude values of shadow pixels are lower, while shadows do not significantly modify the edge gradient at any pixel [27].

To detect shadows, most algorithms convert RGB color space to HSV or YCbCr color spaces. These color spaces are characterized by their ability to separate the light intensity component from the chrominance components. Instead, the proposed method works on RGB color space to eliminate shadows since RGB color camera system is one of the most popular color spaces.

The proposed method incorporates estimated background model information and gamma decoding to detect cast shadow. This integration has greatly increased the scope of applicability and brought significant enhancements in the shadow-free images and the time of processing. The proposed method shows a significant elimination of the shadows in the frames of image sequences. Its basic idea is based on utilizing a common property of the shadow areas in outdoor scenes. That is, the pixels in shadow areas are darker than the rest of the image especially with strong sunlight. Even with dark moving objects, sunlight relatively brightens the pixels of the dark objects while sharpens the pixels of the shadow areas. The proposed algorithm simply involves utilizing Gamma decoding to focus the shadow areas and makes them more apparent to be separated later through a thresholding process. The algorithm uses Gamma decoding (where Gamma value $\gamma>1$) not Gamma correction (Gamma encoding where $\gamma<1$). Gamma decoding maps a wide range of bright values into a narrower range of values, which makes the image darker. However, Gamma correction maps a narrow range of dark values into a wider range of values, which makes the image lighter.

Figure 5 shows the steps of the proposed shadow removal method and Table 1 illustrates the list of symbols used in the algorithm. The inputs of the algorithm are the video frame that contains the shadow and the background image of the video sequence. The background image represents what the environment looks like without any foreground objects. In fact, it will be already estimated during the tracking process. Hence, it will not incur more overhead to the shadow removal algorithm. It is reasonable to assume that a clear view of the background can be obtained, or that the background can be estimated even in the presence of foreground objects [28]. The output of the algorithm is only the video frame after eliminating the shadow.

Algorithm: **Shadow Removal Algorithm**

Inputs: BK, fr_shadow

Output: fr_shadowless

Steps:

```
1.  // Compute the difference between fr_shadow and BK
    and convert the result to black & white image
    Diff=abs(BK- fr_shadow)
    Diff=im2bw(Diff)
2.  // Normalize the frame
    fr_shadow= fr_shadow / BK
3.  // Apply Gamma Decoding to the normalized frame
    fr_shadow=A * fr_shadow ^γ
4.  // Convert fr_shadow from RGB image to grayscale
    image
    fr_shadow= RGBtoGrayscale(fr_shadow)
5.  // Compute the suitable threshold using Otsu's method
    Thresh= Otsu's method (fr_shadow)
6.  // Thresholding fr_shadow
    For i=1 to fr_height
       For j=1 to fr_width
          If  Diff(i,j) > 0 then
             If  fr_shadow(i,j) > Thresh then
                fr_shadowless(i,j)= fr_shadow(i,j)
             else
                fr_shadowless(i,j)= 0
             end if
          end if
       end for
    end for
```

Fig 5: The proposed shadow removal algorithm.

Table 1. The symbols list used in the proposed Algorithm

Symbol	Meaning
BK	Estimated background
fr_shadow	Video frame that contains the shadow
fr_shadowless	Video frame after removing the shadow
fr_height	Frame height
fr_width	Frame width
A	A positive constant that controls the range values
γ	Gamma value

First, the difference between the video frame and the background image is computed in a pixel-by-pixel manner and the absolute value is taken. Large values in the resultant difference image indicate the existence of foreground objects. However, small values are usually noise due to environmental factors such as illumination and background clutter. Hence, these values can be further ignored. The resultant image is then converted to black and white image to facilitate its manipulation. This difference image will be used later in the thresholding process (step no. 6). In the next step, the video frame is normalized by dividing each pixel value by its corresponding value of the background image. This is a pre-processing step to apply Gamma decoding.

To focus the shadow areas, Gamma decoding is applied on the normalized frame. It is a nonlinear operation to improve the fidelity of the brightness value magnitudes. Using the equation showed in the third step in Figure 5, Gamma values (γ) larger than one make the image darker, while values smaller than one make dark regions lighter. In the proposed algorithm, a value larger than one is used to make shadow more apparent. By this process, the shadow pixels are getting darker and hence it can be easily separated. It compresses the low value pixels and stretches high value pixels. The input and output values of the Gamma equation are non-negative real values. The value of the constant A in the same equation controls pixel value range. In the proposed algorithm, the typical value one is used for A. In this case, inputs and outputs are typically in the range from 0 to 1. If a value greater than 1 is used for A, the range will be expanded to be from zero to A.

It should be mentioned that in order to stress the shadow areas, using Gamma decoding is more effective than adjusting image brightness. Applying Gamma decoding does not change the level of detail in the image. It adjusts the RGB value of each pixel in an image but not by the same amount. However, using brightness in the same case can make details wash out or fade to white or black. Since brightness just adds or subtracts the same value to/from each pixel, the image may lose information at the extremes.

After Gamma decoding, the resultant video frame is converted from RGB to grayscale image. The resultant grayscale image has a range from zero to 255. To make the thresholding process more robust, the threshold value should be automatically selected with each frame. The manual threshold setting method and offline learning based method cannot adapt to the variation of the environment in real-time. So in the proposed algorithm, a dynamic threshold is calculated using Otsu's method [29]. It is designed to select the optimum threshold for separation into two classes based upon maximizing the variance between them. It involves iterating through all the possible threshold values and calculating a measure of spread (intra-class variance) for the pixel levels each side of the threshold, i.e. the pixels that fall either in foreground or in background. The aim of this step in the algorithm is to find the threshold value where the sum of foreground and background spreads is at its minimum. It does not depend on modelling the probability density functions; however, it assumes a bimodal (i.e., two classes) distribution of gray-level values.

Then thresholding process is performed over the obtained grayscale image. The new image frame (shadow free image) will have all pixels with values greater than the dynamic threshold. This process ensures the removing of cast shadows because the value of shadow pixels will be less than the threshold value. However, it keeps the pixels of both foreground image and background image. In object tracking, foreground objects are more important. Hence, the difference image obtained in step one is used as a filter during the thresholding process. That is, the thresholding is performed only on the pixels whose corresponding pixels on the difference image are white. By this way, thresholding is restricted to the foreground pixels only instead of the whole image pixels.

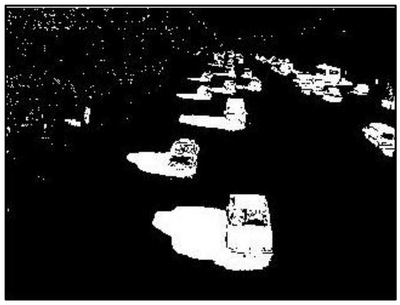
When studying the complexity of the proposed algorithm, we should analyze the complexity of its main parts. One of the main strengths of the algorithm comes from its utilization of well-known simple methods (Gamma decoding and Otsu’s method). Although Gamma decoding is a non-linear operator applied to an image, its computational complexity is linear O(n) where n is the number of pixels in the image. Also among all the segmentation methods, Otsu method is one of the most successful methods for image thresholding because of its simple calculation. It needs $O(m^k)$ time where m denotes the size of the compact image histogram and k denotes the number of thresholds to be determined [29, 30]. In the proposed algorithm, k equals to one. So, its order will be linear O(m). Hence, it is clear that the proposed algorithm is computationally feasible.

Figure 6 illustrates how the proposed algorithm works. The inputs of the proposed algorithm are shown in Figure 6.a and 7.b. A video frame taken from an outdoor surveillance camera is shown in Figure 6.a. The frame image contains a number of vehicles with their cast shadows. Note how shadows represent a change in both intensity and color of the prevailing illumination. Figure 6.b is the estimated background model obtained from the tracking process. The difference between the video frame and the estimated background is shown in Figure 6.c. The image shows the vehicles mask with their cast shadows (foreground objects). Also, some trees leaves appear on both sides of the road. This is because the motion of the leaves usually cause mismatch between the leaves positions of the estimated background model and leaves positions of the video frame. Then, the result of frame normalization is shown in Figure 6.d. Since the normalization process is performed by dividing the video frame pixel values by its corresponding pixel values in the estimated background, the white pixels indicate the matching positions between the video frame and the background model (their division result equals to approximately one). Figure 6.e shows the normalized video frame after applying Gamma decoding. Note how the shadow areas are getting darker. Focusing the shadow areas is the main role that Gamma decoding play in the proposed algorithm.

Before thresholding can be performed, the resultant video frame should be converted to a grayscale image as shown in Figure 6.f. Moreover, Figure 6.g shows the result of applying the dynamic threshold obtained from Otsu's method to the grayscale image. The white pixels represent those pixels with values greater than the dynamic threshold while the black pixels represent the shadow areas. This process alone is not enough. Referring to the difference image obtained in step one (Figure 6.c), the result can be filtered to obtain the foreground only without the background (Figure 6.h). Also, Figure 6.i shows the final frame after removing shadows. One can detect how the proposed algorithm eliminates the cast shadows efficiently. Although the shadow removal may not be perfect, the effect of shadows is so greatly neutralized so that many tracking algorithms can easily benefit from it.

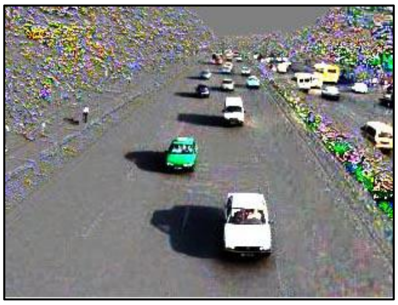
(a) Video frame that contains the cast shadow

(b) Estimated background model



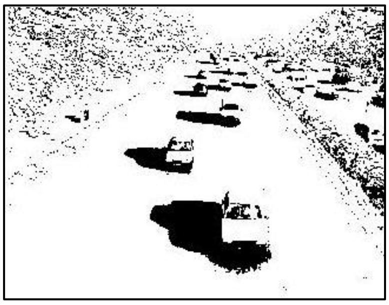
(c) Difference between the video frame and the background image

(d) Normalized video frame

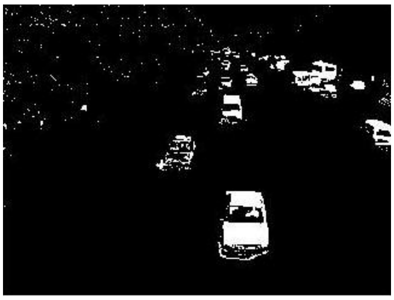


(e) Video frame after Gamma decoding

(f) Converting the resultant video frame to grayscale image



(g) The white pixels indicate the pixels with value greater than the threshold



(h) The resultant frame after thresholding

(i) The final frame after removing cast shadow

Fig 6: The proposed shadow removal algorithm

4. EXPERIMENTAL RESULTS

In this Section, the performance of the proposed method is analyzed. Several experiments were conducted on several datasets that contains different color vehicle, different background and different light source direction. The experiments were implemented on a 2.27GHz Intel Core i5 PC with 4GB memory, running under Windows 8 Enterprise. The algorithm is coded using MATLAB 8.1.0.604 (R2013a).

The first set of experiments aims to directly test the performance of the proposed method algorithm qualitatively. This is achieved by running the proposed algorithm on different video sequences and see its efficiency in removing cast shadows. The video sequences are selected so that it contains vehicles of different dimensions, different colors, and different lighting conditions.

Figure 7 and Figure 8 show the results of running the proposed algorithm on two video sequences (bungalows, highway) provided by “Changedetection.net” (a change detection benchmark dataset available at http://wordpress-jodoin.dmi.usherb.ca/dataset/) [31]. The two testing inputs are uncompressed AVI video files. The resolution of each video frame of “bungalows” and “highway” is 360 × 240 and 320 × 240 respectively. The moving objects for both video sequences are vehicles. The first column of the two figures shows the original frames. The second column shows the frame after applying Gamma decoding. The third column shows the foreground objects found in the frame after removing cast shadows. As it can be seen from the two figures, the shadows are removed quite effectively. However, there are a number of artifacts introduced into frames number 852 and 1074 in Figure 8. The algorithm performance is degraded with the existence of dark moving objects. Some dark parts of the vehicles are misidentified as shadow area but it does not affect the tracking process at all. Actually, in tracking, we do not need a one hundred percent shadow free object. We just need to prune the shadow effect in a way that prevents presenting two objects as one object as a result of a shadow merge.

Fig 7: The results of applying the proposed shadow removal algorithm on "bungalows" video sequence

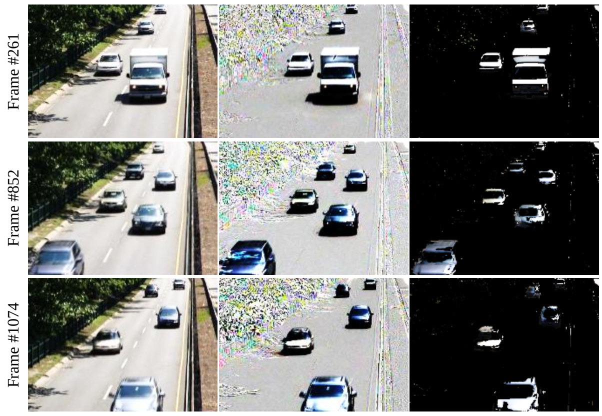


Fig 8: The results of applying the proposed shadow removal algorithm on "highway" video sequence.

In the second set of experiments, to compare the performance of the proposed shadow removal algorithm with existing methods quantitatively, we calculate the shadow detection rate η (Eta) and shadow discrimination rate ζ (Zeta), which are the performance metrics presented in the benchmark paper [11] by Prati et al. Since that time, these two metrics are used widely as standard metrics for testing the performance of shadow detection algorithms [32-36]. They are defined as follows:

$$\text{Shadow detection rate } (\eta) = \frac{TP_s}{(TP_s + FN_s)}$$

$$\text{Shadow discrimination rate } (\xi) = \frac{TP_f}{(TP_f + FN_f)}$$

where TP and FN stand for true positive and false negative pixels with respect to either shadows (S) or foreground objects (F). This means TP_s is the number of pixels which are determined correctly as shadow pixels; TP_f is the number of pixels which are determined correctly as foreground object pixels. FN_s is the number of errors in which a shadow pixel is defined as an object pixel, and FN_f is the number of false detection which identified an object pixel as a shadow pixel.

The shadow detection rate is concerned with labelling the maximum number of cast shadow pixels as shadows. The shadow discrimination rate is concerned with maintaining the pixels that belong to the moving object as foreground. In general, η is reduced with increasing ζ, and ζ is reduced with increasing η; thus, η and ζ are a reciprocal relationship.

Table 2. Descriptions of testing videos

	highway I	**highway III**
Frames Number	440	2227
Frame Resolution	320 × 240	320 × 240
Frame Rate	14 fps	10 fps
Scene Type	Outdoor	Outdoor
Scene Surface	Asphalt	Asphalt
Objects Type	Vehicles	Vehicles
Objects Size	Large	Small
Shadows Size	Large	Small

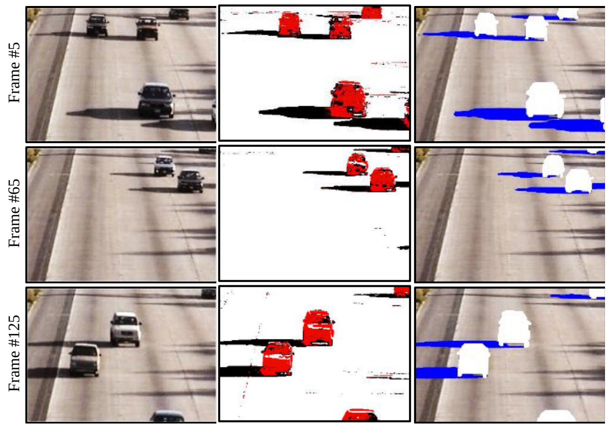


Fig 9: The results of applying the proposed algorithm on "highway I" video sequence: original frame, shadow detection result, ground truth image from left to right, respectively

The proposed approach has been tested on two benchmark video sequences "highway I" and "highway III" (see Table 2 for a detailed description of the two videos). The data set is provided by the Computer Vision and Robotics Research Laboratory of UCSD (http://cvrr.ucsd.edu/aton/shadow). Figure 9 and Figure 10 show the results of applying the proposed algorithm on "highway I" and "highway III" respectively. In each figure, the first column shows the original frame. The second column shows the shadow detection result after applying the proposed algorithm (red pixels indicates the foreground object while black pixels indicates the shadow pixels). Finally, the third column shows the ground truth images used to evaluate the proposed algorithm performance (white pixels indicates the foreground object while black pixels indicates the shadow pixels). The ground truth masks is obtained from http://arma.sourceforge.net/shadows/. Note that the size of vehicles in the video sequence shown in Figure 10 is much smaller compared with vehicles size in Figure 9. However, moving cast shadows of the objects in both figures have been identified effectively.

Moreover, the test videos presented in [33] are used to evaluate the performance of the proposed algorithm. It includes three video sequences: video 1, video 2, and video 3. The resolution of each video frame is 320×180 pixels. Figure 11 shows the results of applying the proposed algorithm on these different video sequences. The first row shows the original frames in videos, the second is shadow detection results, and the third is ground truths. From the figure, it is obvious that the proposed algorithm is feasible in detecting the shadow regions. Table 3 lists comparative results of the proposed algorithm with some state-of-the-art methods [33-36]. It achieves an average shadow detection rate (η) equal to 93% and an average shadow discrimination rate equal to 88%. In addition, the average of the two rates is often used as a single performance measure [32-34]. The combined score is about 91%, which seems so promising results comparing to the existing methods. The results prove that the proposed algorithm habits excellent detection performance and is superior to other existing algorithms.

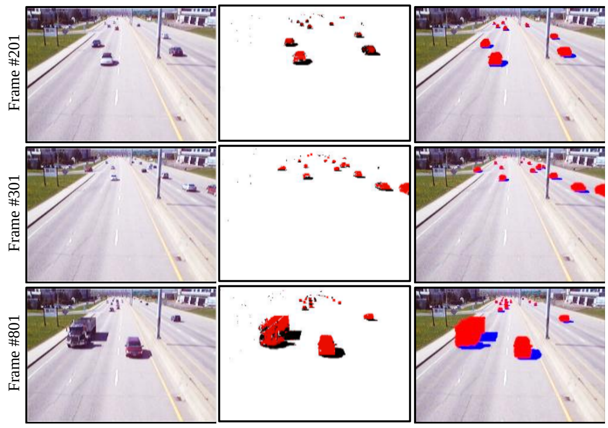


Fig 10: The results of applying the proposed algorithm on “highway III” video sequence: original frame, shadow detection result, ground truth image from left to right, respectively

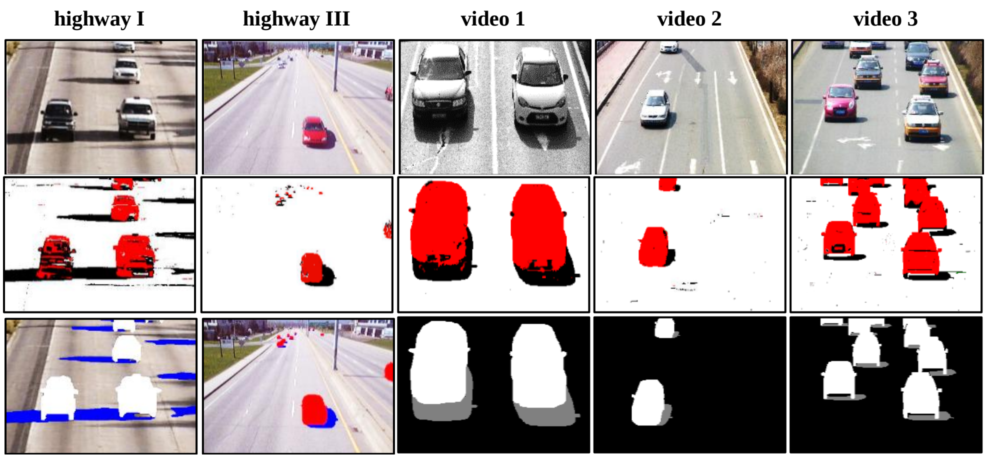


Fig 11: The results of applying the proposed algorithm on different video sequences: original frame, shadow detection result, ground truth image from top to bottom, respectively

Table 3. Comparison of the proposed algorithm with existing methods

Algorithm	highway I			highway III			Video 1			Video 2			Video 3		
	η (%)	ζ (%)	Mean (%)	η (%)	ζ (%)	mean (%)	η (%)	ζ (%)	mean (%)	η (%)	ζ (%)	mean (%)	η (%)	ζ (%)	mean (%)
Sun et al [34] (2010)	89	47	68	72	63	68	84	65	75	85	49	67	83	48	66
Sanin et [35] (2010)	82	94	88	65	92	79	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Choi et al [36] (2010)	86	89	88	84	91	88	72	72	72	88	92	90	90	77	84
Jia et al [33] (2013)	87	85	86	85	81	83	89	93	91	90	98	94	94	82	88
Proposed	92	88	**90**	93	87	**90**	94	89	**92**	97	91	**94**	91	86	**89**

In all previous experiments, the time taken to process each frame of the video is calculated. The average frame processing time (in milliseconds) per sequence is 12.41, 7.15, 11.53, 8.46, and 13.37 for highway I, highway III, video 1, video 2, and video 3 respectively. The variation in average processing time comes from the variation in both number of moving objects and number of frames in each video sequence. The average processing time for all the test video sequences is about 10.58 milliseconds for each frame. This means that the proposed algorithm will be feasible for real time tracking applications. Moreover, using a faster programming language like C++ and some optimization of the code, the processing time of the algorithm could be improved.

In the fourth set of experiments, the objective was to test the effect of using the proposed shadow removal algorithm on the tracking process. Figure 12 shows the tracking results after processing without and with applying the proposed shadow removal algorithm, respectively. Note the colored rectangles indicate the detected objects. In comparison with the results without using the proposed algorithm, in each frame shown in the figure, two vehicles appear as one object in the tracking process because of shadow stretching. In addition, in frames number 6 and 245, some moving shadows (identified by orange and blue rectangles respectively) have been wrongly detected as foreground object. On the other hand, the proposed algorithm has successfully detected the real objects and neutralized the negative influences by shadows.

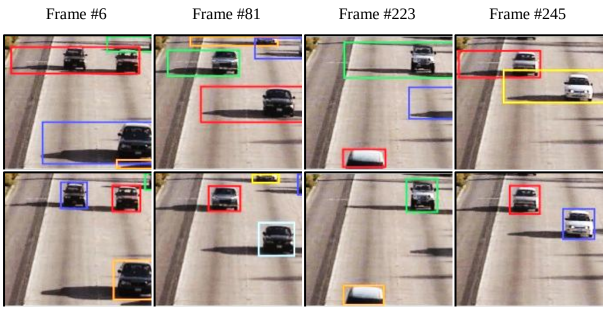


Fig 12: The tracking results on "highway I" video sequence first row: without shadow removal, second row: with the proposed algorithm

5. CONCLUSION AND FUTURE WORK

Removing cast shadows has become an unavoidable step in the implementation of many computer vision algorithms especially tracking systems. Separate objects can be connected through shadows, which can confuse object recognition. Hence, it is relevant to note that when the shadow detection performance is relatively good, shadow removal can in fact lead to considerably better tracking performance compared to cases without using shadow removal, regardless of the tracking algorithm. In this paper, a method for removing cast shadow from vehicles in outdoor environments is introduced. The method works by applying a gamma decoding followed by a thresholding operation and employing an estimated background model of the video sequence.

The proposed method is tested with several real data sequences that contains vehicles of different colors, different background and different light source direction. Also, several experiments have been designed to measure the algorithm's performance. The results obtained from the implementation of the proposed algorithm have shown effective shadow removal and applicability of the proposed technique. The main advantage of this algorithm lies in its computational inexpensiveness. Also, a comparison with some existing algorithms has been conducted in terms of measuring shadow detection rate η and shadow discrimination rate ζ and the results show that the proposed algorithm outperforms the other algorithms.

However, the algorithm's performance may be degraded with deep dark moving objects. In such cases, even though the shadow removal may not be perfect, the effect of shadows is so greatly reduced that many existing algorithms can easily benefit from it. As a future work, this can be handled by a second thresholding of the lower intensity pixels obtained from applying Otsu's method. These pixels above the resultant threshold are considered to be as part of the vehicle, while those pixels occupying the absolute lowest pixel range are considered to be the cast shadow.

6. REFERENCES

[1] Francesc Serratosa, René Alquézar, and Nicolás Amézquita, "A Probabilistic Integrated Object Recognition and Tracking Framework," In the International Journal of Expert Systems with Applications, Volume 39, Issue 8, pp. 7302-7318, June 2012.

[2] Zhongna Zhou, T.X.Han, and Zhihai He, "Semi-Supervised Learning for Robust Car Windshield Tracking and Monitoring in Live Traffic Videos," In Proceedings of the 19th IEEE International Conference on Image Processing (ICIP), Orlando, Florida, USA, pp. 489-492, September 2012.

[3] Kostia Robert, "Video-Based Traffic Monitoring at Day and Night Vehicle Features Detection Tracking," In Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems (ITSC'09), St. Louis, USA, pp. 1-6, October 2009.

[4] King-Wah Pang, "An Adaptive Parallel Route Construction Heuristic for the Vehicle Routing Problem with Time Windows Constraints," In the International Journal of Expert Systems with Applications, Volume 38, Issue 9, pp. 11939-11946, September 2011.

[5] Bin Chen, QingYao Wei, Cao He, and JianBo Wang, "Study of Expressway Traffic Simulations System Influence of Accident," In Proceedings of the 2010 International Conference on Optoelectronics and Image Processing (ICOIP'10), Volume 02, pp. 411-415, November 2010.

[6] Ryan P. Avery, Guohui Zhang, Yinhai Wang, and Nancy L. Nihan, "Investigation into Shadow Removal from Traffic Images," In the International Journal of Transportation Research Record, pp. 70-77, 2007.

[7] S. A. Lakhotiya, M. D. Ingole, and M. S. Joshi "Human Object Tracking using Background Subtraction and Shadow Removal Technique," In the International Journal of Computer Science and Applications, Volume 6, Number 2, pp. 117-121, April 2013.

[8] Nijad Al-Najdawi, Helmut E. Bez, Jyoti Singhai , and Eran.A. Edirisinghe, "A Survey of Cast Shadow Detection Algorithms," In the Journal of Pattern Recognition Letters, Volume 33, Issue 6, pp. 752–764, April 2012.

[9] Ye-Peng Guan, "Texture and Space-Time Based Moving Objects Segmentation and Shadow Removing," In Proceedings of the Third International Conference on Advances in Swarm Intelligence (ICSI'12), Shenzhen, China, Volume Part II, pp. 244-253, June 2012.

[10] Andres Sanin, Conrad Sanderson, and Brian C. Lovell, "Shadow Detection: A Survey and Comparative Evaluation of Recent Methods," In the Journal of Pattern Recognition, Volume 45, Issue 4, pp.1684-1695, April 2012.

[11] Andrea Prati, Ivana Miki, Mohan M. Trivedi, Rita Cucchiara, "Detecting Moving Shadows: Algorithms and Evaluation," In IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: 25, Issue: 7, pp. 918 - 923, July 2003.

[12] Qiang He, and Chee-Hung Henry Chu, "A New Shadow Removal Method for Color Images," In the Journal of Advances in Remote Sensing, Volume 2, Number 2, pp. 77-84, June 2013.

[13] Shen-Chi Tien, and Tsorng-Lin Chia, "Positioning a Point Target in an Aerial Image," In Digital Signal

Processing Journal, Volume 19, Issue 5, pp. 874-889, September 2009.

[14] Kartik Chandra, Neeharika Adabala, and Kentaro Toyama, "Aerial Image Relighting: Simulating Time of Day Variations," In Proceedings of the 24th International Conference on Advances in Computer Graphics (CGI 2006), Hangzhou, China, pp. 594-605, June 2006.

[15] Hiroshi Kawasaki, and Ryo Furukawa, "Shape Reconstruction and Camera Self-Calibration Using Cast Shadows and Scene Geometries," In the International Journal of Computer Vision, Volume 83, Issue 2, pp. 135-148, June 2009.

[16] Christian Wöhler, Kia Hafezi, "A General Framework for Three-Dimensional Surface Reconstruction by Self-Consistent Fusion of Shading and Shadow Features," In Pattern Recognition Journal, Volume 38, Issue 7, pp.965-983, July 2005.

[17] Mark S. Drew, Yacov Hel-Or, Tom Malzbender, Nasim Hajari, "Robust Estimation of Surface Properties and Interpolation of Shadow/Specularity Components" In Image and Vision Computing Journal, Volume 30, Issue 4-5, pp. 317-331, May 2012.

[18] James P. O'Shea, Martin S. Banks, and Maneesh Agrawala, "The Assumed Light Direction for Perceiving Shape from Shading," In Proceedings of the 5th Symposium on Applied Perception in Graphics and Visualization (APGV '08), Los Angeles, California, USA, pp. 135-142, August 2008.

[19] R. Girisha, and S. Murali, "Self Shadow Elimination Algorithm for Surveillance Videos Using ANOVA F Test," In Proceedings of the Third Annual ACM Bangalore Conference (COMPUTE '10), Bangalore, India, Article No. 12, January 2010.

[20] M. Xiao, C. Z. Han, L. Zhang, "Moving Shadow Detection and Removal for Traffic Sequences," In the International Journal of Automation and Computing, Volume 4, Issue 1, pp. 38-46, January 2007.

[21] Pankaj Kumar, "Intrinsic Image Based Moving Object Cast Shadow Removal in Image Sequences," In Proceedings of the 2011 International Conference on Digital Image Computing: Techniques and Applications (DICTA '11), pp. 410-415, December 2011.

[22] Xue Li, Yu-Jin Zhang, and Bao-Di Liu, "Robust Moving Cast Shadows Detection and Removal with Reliability Checking of the Color Property," In Proceedings of the Sixth International Conference on Image and Graphics (icig'11), Hefei, Anhui, China, pp.440-445, August 2011.

[23] Y. Matsushita, K. Nishino, K. Ikeuchi, and M. Sakauchi, "Illumination Normalization with Time-Dependent Intrinsic Images for Video Surveillance," In IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Volume 26, Issue 10, pp.1336-1347, October 2004.

[24] Yang Wang, "Real-Time Moving Vehicle Detection with Cast Shadow Removal in Video Based on Conditional Random Field," In IEEE Transactions on Circuits and Systems for Video Technology, Volume 19, Issue 3, pp.437-441, March 2009.

[25] Chin-Teng Lin, Chien-Ting Yang, Yu-Wen Shou, Tzu-Kuei Shen "An Efficient and Robust Moving Shadow Removal Algorithm and its Applications in ITS," In EURASIP Journal on Advances in Signal Processing, Volume 2010, Article No. 39, February 2010.

[26] A. J. Joshi, S. Atev, O. Masoud, N. Papanikolopoulos, "Moving Shadow Detection with Low- and Mid-Level Reasoning," In Proceedings of the IEEE International Conference on Robotics and Automation, Roma, Italy, pp. 4827-4832, April 2007.

[27] Chen Wei-Gang, Xu Bin, "Detecting Moving Shadows in Video Sequences Using Region Level Evaluation for Vision-Based Vehicle Detection," In Proceedings of the 2010 Fifth International Conference on Frontier of Computer Science and Technology (FCST '10), pp. 142-146, August 2010.

[28] Vikas Reddy, Conrad Sanderson, and BrianC Lovell, "A Low-Complexity Algorithm for Static Background Estimation from Cluttered Image Sequences in Surveillance Contexts," In EURASIP Journal on Image and Video Processing, Special issue on advanced video-based surveillance archive, Volume 2011, Article No. 1, January 2011.

[29] Mehmet Sezgin,and Bulent Sankur, "Survey Over Image Thresholding Techniques and Quantitative Performance Evaluation," In the Journal of Electronic Imaging, Volume 13, Issue 1, pp. 146-168, Jan 2004.

[30] Kuo-Liang Chung, Wan-Yu Chen, "Fast Adaptive PNN-based Thresholding Algorithms," In Pattern Recognition Journal, Volume 36, Issue 12, pp. 2793 – 2804, December 2003.

[31] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection.net: A New Change Detection Benchmark Dataset," In Proceedings of the IEEE Workshop on Change Detection (CDW-12), Providence, Rhode Island, USA, pp. 1-8, June 2012.

[32] Jia-Bin Huang, and Chu-Song Chen, "Moving Cast Shadow Detection Using Physics-Based Features," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, Florida, USA, pp. 2310–2317, June 2009.

[33] Yuhong Jia, Xiaoxi Yu, Jiangyan Dai, Wanjun Hu, Jun Kong, and Miao Qi, "A Novel Moving Cast Shadow Detection of Vehicles in Traffic Scene," Intelligent Science and Intelligent Data Engineering Lecture Notes in Computer Science, Springer, Volume 7751, pp. 115-124, 2013.

[34] Bangyu Sun, Shutao Li, "Moving Cast Shadow Detection of Vehicle Using Combined Color Models," In the Proceedings of the Chinese Conference on Pattern Recognition (CCPR), China, pp. 1–5, October 2010.

[35] Andres Sanin, Conrad Sanderson, and Brian C. Lovell, "Improved Shadow Removal for Robust Person Tracking in Surveillance Scenarios," In Proceedings of the International Conference on Pattern Recognition, Istanbul, Turkey, pp. 141–144, August 2010.

[36] JinMin Choi, Yung Jun Yoo, Jin Young Choi, "Adaptive Shadow Estimator for Removing Shadow of Moving Object," In the Journal of Computer Vision and Image Understanding, Volume 114, Issue 9, pp.1017-1029, September 2010.