Robust Segmentation of Freight Containers in Train Monitoring Videos

Qing-Jie Kong*,†, Avinash Kumar*, Narendra Ahuja*, and Yuncai Liu†

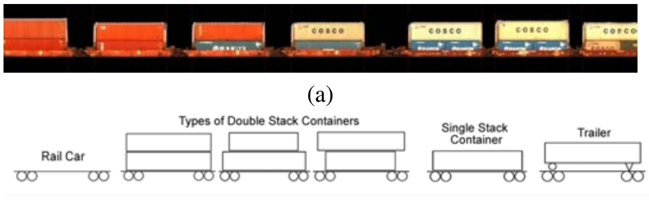
$*$ Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign, Urbana, IL 61801
† Institute of Image Processing and Pattern Recognition
Shanghai Jiao Tong University, Shanghai 200240, China
kongqingjie@gmail.com,{avinash,n-ahuja}@uiuc.edu,whomliu@sjtu.edu.cn

Abstract

This paper is about a vision-based system that automatically monitors intermodal freight trains for the quality of how the loads (containers) are placed along the train. An accurate and robust algorithm to segment the foreground of containers in videos of the moving train is indispensable for this purpose. Given a video of a moving train consisting of containers of different types, this paper presents a method exploiting the information in both frequency and spatial domains to segment these containers. This method can accurately segment all types of containers under a variety of background conditions, e.g illumination variations and moving clouds, in the train videos shot by a fixed camera. The accuracy and robustness of the proposed method are substantiated through a large number of experiments on real data of train videos.

1. Introduction

Intermodal (IM) freight trains (Fig. 1(a)) are the largest and prominent freight vehicles in the North American Freight Railroads network. These trains are composed of *containers* placed on *rail cars* (see Fig. 1(b)). The typical length of rail cars ranges from 35m to 40m and each IM train consists of 100-125 rail cars, thus making the length of the overall train beyond 2 miles. Their operating speeds can reach 75-80 miles per hour (mph). Due to high speeds and the flow of air through the gaps between the containers, these trains suffer from large aerodynamic resistance. This causes high fuel consumption leading to expensive operating costs. Lai *et al.* [4] concluded that placing small containers over larger rail cars results in aerodynamically inefficient loading pattern of IM trains. They also proposed that an analysis of gap lengths between consecutive containers would be a good metric of characterizing the quality of a loading pattern. This analysis would be a feedback to improve loading pattern of IM trains at various railroad yards and thus help in cutting of operating costs. But due to long length of IM trains, obtaining gap lengths and evaluating the efficiency of the loading pattern manually are tedious tasks. This fact motivates the development of a method for automatic, reliable and efficient detection of loading patterns of containers and the gaps between adjacent containers.



(a)

(b)

Figure 1. Intermodal freight train. (a) An example of the train; (b) Rail car and different types of containers.

A vision-based automatic Train Monitoring System (TMS) can achieve this goal, as shown in the prototype developed by Kumar *et al.* [3]. Their system includes segmenting and extracting the foreground in image frames which consists of the containers on the train. This is followed by stitching the non-overlapping parts of successive frames, thus forming a mosaic that shows the entire train (Similar to Fig. 1(a)). The containers are then segmented and their specific types are identified, and finally, all the above information is used to calculate gap lengths [3]. Among all of the above tasks, segmentation of train containers is the most critical one, as it determines the performance of all the other tasks. The problem of accurate segmentation is made challenging by the following facts:

- IM train videos are captured throughout the year under varying outdoor weather conditions at different times.
- On a smaller scale, often there are illumination changes in videos due to movement of cloud shadows.
- Apart from accuracy, time efficiency of the system is also important. Each IM train is captured as a set of 10-15 videos depending upon the length of the train. Each video has 1024 image frames, each of size 640 × 480. As the number of trains captured in a day could be as high as 10, the system should be fast and efficient in processing large set of videos.
- The segmentation method should handle different container shapes and size (Fig. 1(b)).

978-1-4244-5498-3/09/$25.00 ©2009 IEEE

- The system should operate with uncalibrated cameras (i.e., with arbitrary camera angle and position) so that it could be deployed at any place near any main line.

However, few of existing methods can simultaneously satisfy all the above requirements.

In this context, this paper proposes a four-stage method for segmenting the containers within the videos of intermodal freight trains acquired from a fixed camera placed along the track-side. In the first stage, the periodic reappearance feature of the containers is exploited to identify and remove the background at the top and bottom of the containers. In the second stage, a background model is learned over the distribution of pixel intensity in a pre-calculated window location in the background image. This model is utilized to remove background in the gaps between consecutive containers. In order to handle varying illumination, the background model is updated at every image frame where the background appears with the gap. In the third stage, if a single stack (Fig. 1(b)) is present in an image frame, it gets detected by background subtraction. In the last stage, a post processing step is performed to obtain more accurate results for the foreground segmentation obtained earlier. Color information is also used to further improve the results. The effectiveness of our method is shown in experiments with numerous real videos, under conditions ranging from daytime blue sky without clouds to the cloud sky in the evening.

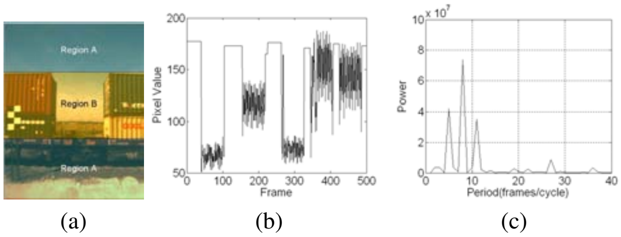
2. Conventional Foreground Segmentation Techniques

Foreground segmentation in videos is a traditional problem in computer vision. Below we mention a few classical approaches and why they fail for our case:

- **Template Based**: A template of the background can be stored before the IM train arrives in the field of view of the camera. Each image frame in the captured video could then be subtracted from the template, and the difference is thresholded to obtain foreground. But, due to length of the train and the movement of clouds in the video, there is considerable difference in the background present between the template and the image frames near the end of the video. Thus, the threshold parameter becomes very critical and is hard to set.
- **Gaussian Mixture Model(GMM) Based** [6]: This method maintains a mixture of Gaussian at each pixel location and classifies the pixel intensity, in the latest image frame being processed, at that location as belonging to either the background or foreground Gaussian. In our problem the containers can have similar intensities as present in the background. Thus, containers might get confused with the background. This effect is shown in the image frames located on the second row of Fig. 8, where some pixels lying in the container have been classified as belonging to background by the GMM method.
- **Energy Minimization Based**: Foreground extraction can be formulated as a energy minimization problem as in [2]. Although highly accurate, but owing to large processing time requirement of discrete optimization over a very large set of image frames, it cannot be applied to build a fast vision system in our case. This is basically a trade-off between accuracy and time efficiency, which cannot be ignored in our case.
- **Edge Detection Based** [3]: This method exploits the edge features of containers to guide the segmentation procedure. However, the detected edges are not always meaningful and accurate, and involve selection of parameters that are hard to generalize to apply well to all conditions.

3. Proposed Approach

3.1. Stage 1: Detecting Train Region



(a) (b) (c)

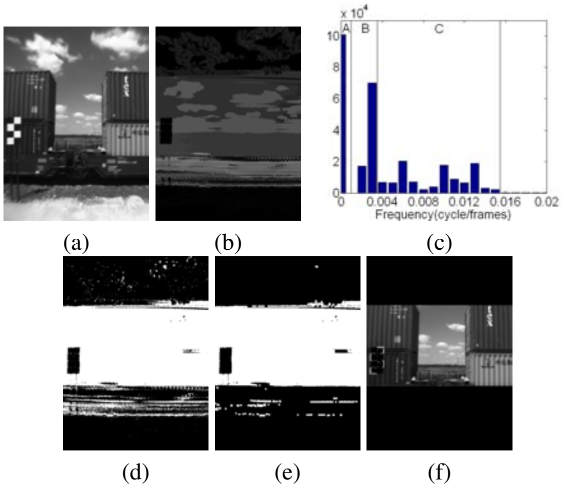
Figure 2. Time series of a pixel located in Region B. (a) Partition of the region in a frame; (b) Pixel signal in temporal domain; (c) Power spectrum of the signal.

Periodicity is a significant feature of objects in motion, and it has been widely used for segmenting objects [1] and detecting pedestrians [5]. The regions in input image frames of an IM train video can be divided into two types: *Region A* is the region above and below the containers (see Fig. 2(a)); *Region B* is the region where the locomotives are imaged (see Fig. 2(a)). The goal of the first stage is to use periodicity of trains to remove Region A. For every pixel location in an image frame, the intensity values are accumulated across time and a time series is obtained. Let $I(i,j)(t)$ represent the intensity at a pixel location in frame t of size $M \times N$. The intensities $I(i,j)(t)$ are accumulated for the length of the video, thereby obtaining $M \times N$ time series. It is observed that the intensity at an image location belonging to Region A is relatively constant across the video. Thus, the time series belonging to pixels in these regions are expected to have less variance. Whereas, in Region B, where the containers and the gaps appear alternately as the video progresses, it is observed that the time series for pixels in this region consist of prominent crests and troughs (see Fig. 2(b)). Consequently, the proposed method is based on extracting frequency features from the time series data and applying it for foreground segmentation. However, before the time series can be processed, the noise in the time series signal of every pixel is filtered by the follow operation:

$$I'(i,j)(t{+}1) = \begin{cases} I(i,j)(t), \text{if } |I(i,j)(t+1) - I(i,j)(t)| < \varphi \\ I(i,j)(t+1), \text{otherwise.} \end{cases} \tag{1}$$

where $I'(i,j)(t+1)$ represents the new value in the frame $t+1$; φ denotes the threshold of controlling the filtering strength. From the experiments, it is found that this operation reduces the effect of high-frequency noise on feature extraction from time series data.

Next, Fast Fourier Transform (FFT) is computed for this signal and the period and power spectrum is obtained (Fig. 2(c)). For every time series obtained, only the most dominant frequency is assumed to be its inherent frequency and is stored for that location in a 2-D array of size $M \times N$. Thus, a spatial image is obtained where each location stores the most prominent frequency. This image is called the frequency image. A sample frequency image obtained for an IM train video is shown in Fig. 3(b), where the values have been normalized to lie between 0 and 255. It can be inferred from the frequency image that the frequency values of most of the pixels in Region B are higher than those in Region A.



(a) (b) (c)

(d) (e) (f)

Figure 3. Steps in the first stage. (a) A frame in a video; (b) Frequency image of the video; (c) Histogram of the frequency image; (d) Thresholded result; (e) Thresholded result after the morphological operations; (f) Final result of the first stage.

A pixel value histogram of the frequency image is calculated, as shown in Fig. 3(c). This histogram is analyzed to obtain a range estimate of the frequencies which belong to Region A and Region B. For our case, we obtain approximate ranges for the following cases:

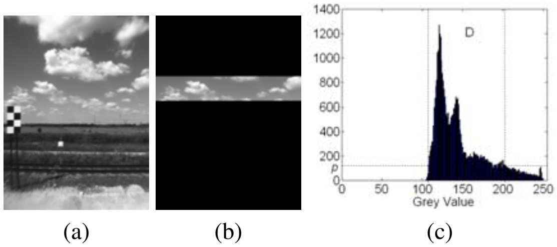
- Range A : frequency of the pixels which belong to sky and ground, whose frequency is 0.
- Range B : frequencies of the pixels belonging to clouds outside the train region.
- Range C : frequencies of the pixels in the train region.

Numerous experiments with frequency image were carried out in order to determine optimal threshold, using a large set of videos whose backgrounds and containers vary significantly. It was found that the frequencies of the train regions were always in the range between 0.004 and 0.02. Thus, the frequency image can be thresholded easily as shown in Fig. 3(d). In order to get smoother results, two times erosion and dilation operations are performed in sequence (see Fig. 3(e)). Finally, the train region (i.e., Region B) can be obtained by projecting the pixel values of Fig. 3(e) onto the y-axis. The final result of Stage 1 is shown in Fig. 3(f).

3.2. Stage 2: Removing Background Gap

The goal of the second stage is to remove the background between the containers. This task is difficult as in most cases the part of background region visible in the gaps is smaller than the size of the foreground, i.e., the container. The proposed method consists of the following three steps.

Background Model. A reference image frame is chosen from the first few frames of the IM train video before the train arrives in the camera view. This reference image frame (as shown Fig. 4(a)) does not contain any foreground objects (i.e., IM containers). A rectangular window in this frame is selected where the containers are most likely to appear where the location is determined based on the information from the first stage of our method and constrained to regions containing the sky. This rectangular image is then used as an initial background image (Fig. 4(b)). Next, a histogram of the chosen background region is computed, as shown in Fig. 4(c). Once the histogram is produced, the principal intensity range of the background image in the selected window can be decided as follows: first set a threshold value p, then the range where the histogram value of the grey value is larger than p is defined to be the intensity range of the chosen background image, shown as the Range D in Fig. 4(c). By this means, around 90% grey value of the chosen background image can always be included in this intensity range, no matter what form the histogram is.



(a) (b) (c)

Figure 4. Background model. (a) A frame of background image; (b) Sub-Region chosen from the background image; (c) Histogram of the chosen background region.

Background Removal. Once the intensity range of the background is obtained, we can classify a pixel in Region B as background by comparing its pixel value in the current frame with the intensity range of the background. The first step of processing the current frame is to apply a median filter to remove salt and pepper camera sensor noise from the image. After this, a window whose position and size are same as the one learned in the background image is chosen in the current frame, and all pixels in this window are compared to the intensity range learned in the last step (Fig. 5(b)). A series of morphological operations consisting of erosion and dilation are applied to get smooth results (Fig. 5(c)). Finally, the left and right boundaries of the background region can be obtained by projecting the pixel values of Fig. 5(c) onto the x-axis, thereby obtaining the middle background region between two containers (Fig. 5(d)).

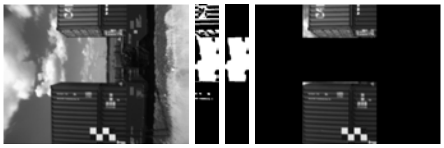


Figure 5. Background removal. (a) A frame in a video; (b) Result of the recognition (the white regions represent the regions that are recognized as the background); (c) Result after the morphological operations; (d) Segmentation result after the first two stages.

Background Update. Due to varying outdoor conditions, there are significant illumination changes in the background. To obtain robust performance, it is necessary to update the background model. In the IM video, a gap can be viewed as a part of the image region which is moving along with the containers. Since this gap is centered at different locations in an image frame, they span the complete background once in a while in a set of contiguous few frames. The detected background regions in these gaps can then be spliced together to rebuild a new background image, and the above-mentioned background model is repeated for update.

3.3. Stage 3: Detecting Single Stack

After the above two stages, most of the loading patterns (e.g., double stack, trailer) can be segmented successfully. However, the case of single stack still can not be tackled, as the position of the selected detected window in Stage 2 can only be limited to the region containing sky background, thereby the window can not find the single stack as shown in Fig. 6(b). This results from from the use of a simple yet effective background model (which contains only the sky region). In order to detect the single stack, an additional background subtraction step is explored after the first two stages. A detection window is set in the region of the single stack as its location is known to appear at certain image region. In the detecting window, the background subtraction is performed only in the region of the middle background which has been removed in Stage 2. A subtraction result after morphological operations is shown in Fig. 6(c). Then, the left and right boundaries of the single stack can be obtained by the same projecting method of pixel values as used in Stage 2, and the upper extent of the single stack is detected by using the refinement method presented in the next stage. One example of the final segmentation result is shown in Fig. 6(d). The background subtraction method used here is reliable, because first, the background in the single stack region is the distant view, in which both clouds in the sky and objects on the ground are nearly static in quite a little time; second, it is not necessary to exactly segment all of the foreground pixels of the single stack but just to find its quadrate foreground region, as is shown in Fig. 6(c).

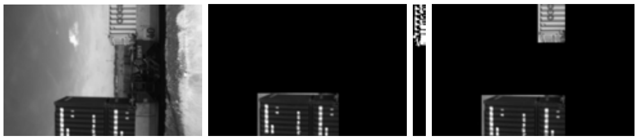


Figure 6. Single stack detection. (a) A frame in a video; (b) Segmentation result after the first 2 stages; (c) Result of the recognition (the white regions represent the regions recognized as the foreground); (d) Final segmentation result.

3.4. Stage 4: Refining Segmentation Result

Since the upper extent is determined only by the highest container in a video, some defective results may be produced as shown in Fig. 7(c). Therefore, a refinement post-processing needs to be applied to obtain better results. To the segmentation result obtained after the first two stages (Fig. 7(c)), the background model and background removal operations in Stage 2 are repeated. The difference only lies in the window choice, as shown in Fig. 7(b). In this operation, the upper line of the window is supposed to be consistent with the upper limit of the train region derived from Stage 1. The lower line of the window should be lower than the upper line of the lowest container. The width of the window is same as that of the frame. By the second time of background model and removal, it can be known exactly whether the upper part of the foreground mask belongs to the container or not, and find the real upper limit of the container. The refined result is shown in Fig. 7(d).

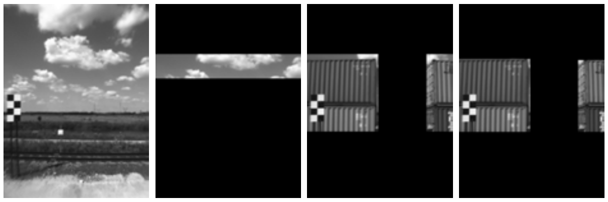


Figure 7. Refinement of segmentation results. (a) A frame of background image; (b) Sub-Region chosen from the background image; (c) Result before refinement; (d) Result after refinement.

3.5. Exploiting Color Information

Although the proposed method is effective in segmenting containers from grey-scale videos, it may not perform well when the intensity of the container is very similar to the background intensity. This can be easily overcome by using the complete RGB color channel instead of intensity in our method. Thus, the color information is combined in the proposed method to increase its accuracy on larger set of videos. There are several ways to exploit color information. For instance, we can perform the proposed segmentation algorithm respectively to the RGB channels, and then fuse their results. In addition, we may also model the background in the YUV color space to implement the segmentation. Our experiments show that the former approach is more robust than the latter one. For every pixel in the chosen window, we set its value at 1 if it is classified as a background pixel or 0 if it is judged as a foreground pixel. The final class label for that pixel location is determined by

$$C(i,j) = C_R(i,j) \bigcap C_G(i,j) \bigcap C_B(i,j) \qquad (2)$$

where $C_{RGB}(i,j)$ denote the class labels from the RGB channels at pixel (i,j); $\cap$ represents the AND operation.

4. Experiments

We validate the proposed method with 150 representative videos of IM trains where each train is captured in 10-15 videos at 15 fps. Each video consists of 1024 image frames, each of size 640 × 480. The videos encompass a wide range of background conditions, e.g blue sky without clouds, day with bright sunlight, static and moving clouds which can be dense or sparse during daytime and evenings, and typical rainy day with less illumination. For our experiments, the same group of parameters, which had been obtained from the training set beforehand, was used for different sorts of the backgrounds. But the algorithm in color information employed a different group of parameters from that in grey value. The experiments consisted of four parts:

Percentage of videos with successful segmentation of Region B from Region A (Stage 1).

The failure of Stage 1 was defined as the case when the boundary between Region A and Region B was incorrectly detected for a given video. Otherwise it was declared as a successful detection. We had successful detection in **96%** of the videos (total is 150). All the videos where Stage 1 failed was due to the fact that intensities of most of the learned containers were too close to that of the background. Due to this, the frequencies of most of the pixels in the train region fell out of the Range C, as defined for Stage 1 in Section 3.1, which is the range for the correct detection of trains. Thus the algorithm of Stage 1 treats some of pixels in the train region as belonging to background. But, it was observed that this situation happened only in the video in which *most of the* containers had the same intensity as that of the background, e.g. blue containers with clear blue sky in the background. In our experiments we seldom (4% of 150) encountered such videos. Thus, it can be concluded that Stage 1 of the method was quite robust.

Percentage of videos with successful segmentation of gaps (Stage 2-4).

Stage 2-4 (Section 3.2, 3.3, 3.4) were validated using both grey and color information in a video. The combined results are shown in Table 1. Each of the test videos included around 8 containers. In the beginning, all train videos were classified into 8 classes depending on the type of background presented in them. These classes are shown in the first column of Table 1. For each of these classes, we obtain the success percentage by dividing the number of the *correctly segmented gap* (accompanying a container) to the total number of containers for the videos belonging to that class (last two columns of Table 1). A correctly segmented gap implies that the boundary between the container and background are found correctly as the container passes through the scene, as shown in the last row of Fig. 8. The exhaustive experimental results are enumerated in Table 1.

Table 1. Results of the second experiment. TNC = Total Numbers of the Containers; SR = Success Ratio.

Background Conditions	TNC	SR (Grey)	SR (RGB)
Day/No clouds/Blue sky	104	85.6%	96.2%
Day/Bright sunlight	61	88.9%	91.8%
Day/Heavy clouds	278	98.2%	100.0%
Day/Moving clouds	128	99.2%	100.0%
Day/General situation	323	98.8%	100.0%
Evening/Heavy clouds	98	100.0%	100.0%
Evening/Moving clouds	171	100.0%	100.0%
Rainy day/Water on lens	59	100.0%	100.0%
Total	1222	97.4%	99.3%

From the table, it is observed that the percentage success when the background is blue sky or bright sunlight is lower than the others. For the case of blue sky without clouds, the cause of erroneous gap detection, when only grey value information was used, was due to the presence of containers with similar intensity (similar to the erroneous case encountered in the first experiment). Even if color information was combined, there were still a few containers whose color was same as that of the blue sky. In the case of bright sunlight, the light reflected by the body of the containers made several light-colored containers bear similar illumination as the background. Apart from the above observations, Table 1 shows that combining color information greatly increased the accuracy and robustness of the proposed method.

Comparison of the proposed algorithm with the GMM based method [6].

The results are compared in Fig. 8, from which it can be inferred that the GMM based method have two inherent drawbacks for our case. One is that often the foreground is classified as background, since the color of the container are close to the color of the background (image located in the 1st row and 2nd column of Fig. 8). The other is that it often

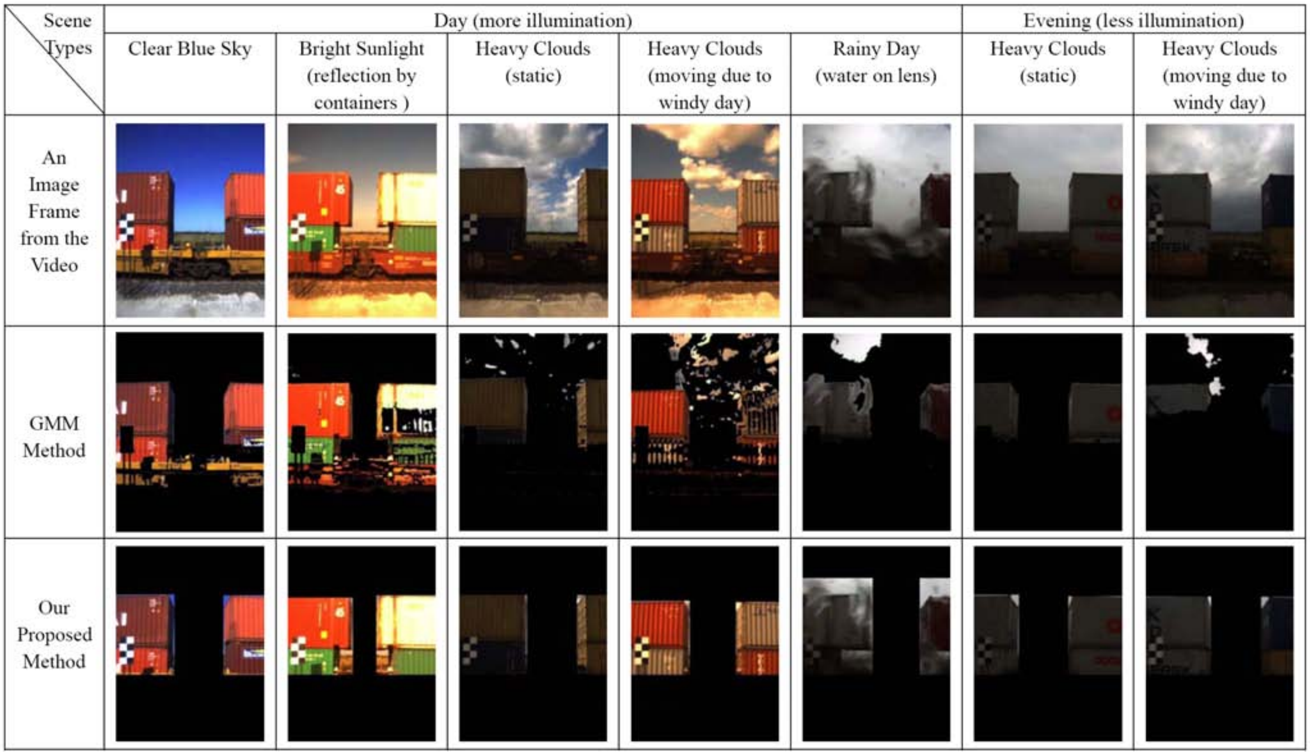

Scene Types
Day (more illumination)
Evening (less illumination)
Clear Blue Sky
Bright Sunlight (reflection by containers)
Heavy Clouds (static)
Heavy Clouds (moving due to windy day)
Rainy Day (water on lens)
Heavy Clouds (static)
Heavy Clouds (moving due to windy day)
An Image Frame from the Video
GMM Method
Our Proposed Method

Figure 8. Segmentation results compared to those obtained using the GMM based background subtraction (row 2) [6].

segments other moving objects except for containers, such as moving clouds. This is because the GMM does not use any high level information about the scene for segmentation, e.g. rectangular shape of containers. It should also be noted that these problems can not be solved by simply applying larger morphological operators or adding some constrains, because both the misclassified regions in the container body and the cloud regions are often very large (see image in the 2nd row and 4th column in Fig. 8). However, the proposed method can overcome these problems and robustly perform under varieties of background conditions.

Time efficiency of the whole algorithm.

The operation speed of the proposed algorithm was measured with an Intel(R) Core2 Due CPU 2.53GHz processor and 3GB RAM. The average processing speed is 4 frames per second (fps), which is close to real time processing.

5. Conclusion

We have proposed a robust and efficient method that combines the information in frequency domain and spatial domain to segment train containers in IM train videos. This method takes advantage of the periodic-motion feature of containers to detect the regions corresponding to them, and then removes the background region between consecutive containers, by first estimating and then using a model for the background defined in terms of the histogram of the background image. Experiments with real-world train videos validate the robustness and accuracy of the proposed algorithm. The proposed module is being integrated into a real time vision system for intelligent train monitoring.

Acknowledgements

The authors thank the support of BNSF gratefully. This work is also supported in part by China National 973 Program of 2006CB303103, China NSFC Key Program of 60833009 and China National 863 Project of 2009AA01Z330.

References

[1] O. Azy and N. Ahuja, “Segmentation of periodically moving objects,” in *Proc. 19th Int. Conf. Pattern Recognition*, 2008.

[2] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222-1239, 2001.

[3] A. Kumar, N. Ahuja, J. M. Hart, U. K. Visesh, P. J. Narayanan, and C. V. Jawahar, “A vision system for monitoring intermodal freight trains,” in *Proc. IEEE Workshop Appl. Comput. Vision*, 2007, pp. 24-29.

[4] Y. C. Lai, C. P. L. Barkan, J. Drapa, N. Ahuja, J. M. Hart, P. J. Narayanan, C. V. Jawahar, A. Kumar, and L. Milhon, “Machine vision analysis of the energy efficiency of Intermodal Freight trains,” *J. Rail Rapid Transit*, vol. 221, pp. 353-364, 2007.

[5] Y. Ran, I. Weiss, Q. Zheng, and L. S. Davis, “Pedestrian detection via periodic motion analysis,” *Int. J. Comput. Vision*, vol. 71, no. 2, pp. 143-160, 2007.

[6] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 1999, pp. 246-252.