AUTOMATED RAILCAR INSPECTION

Centre for Visual Information Technology,
International Institute of Information Technology, Hyderabad, India
K.Sumanth, C. Navya Latha
sumanth@students.iiit.net, navya@students.iiit.net

Abstract — **Modeling of dynamic activities and automated inspections at all levels has been a research topic in the field of Computer Vision for many years. Most of the early works on activity representations come from the field of Artificial Intelligence. In order to understand the activities that arise out of the interactions of a configuration in fields like image analysis, speech recognition, handwriting recognition, motion detection they have to be modeled so that recognition is possible. The various features in the input have to be extracted and based on these observations recognition is done. In this project, the formalisms employed are varied depending on the required outputs and the complexity of the input. Many of the algorithms applied here are learning algorithms, edge detection and feature extraction with an added feature of modeling. This project deals with camera based automation in railcars.**

I. Introduction

Camera-based Automation will be on the upside over other automation systems, mostly because it takes care of today's needs as well as the new applications of tomorrow, commented Larry Graham, global manager of automatic identification for General Motors Corporation. Once you capture an image, there's so much more you can do with it.The advantages of using Camera based Automation are mani-fold. Since the camera based automation actually captures images, not only can the defects be found out, they can also be pinpointed providing us with an efficient feedback. The processing would be done in real time and so the time consumed would be saved along with the efficient utilisation of man-power. Since the human element is being removed, the results will be more objective in nature, thus helping us find the cause of defects. Also some of the defects can not be found out by the human eye. Moreover, it is not feasible to manually check each and every system for an anomaly.

II. Objectives

The objective we are looking at in this project consists of the following: (i) Acquisition of images of freight railcars for inspection of the safety appliances and other appliances. (ii) Identification of image characteristics associated with the health of the safety appliances. (iii) Identification of types of models useful to represent the railcar appearance when the safety appliances are in satisfactory condition, and if necessary, a model for unsatisfactory condition as well.

Figure 1: Image of the RailCar

The current objective is efficient background subtraction of the acquired video independent of the various paramaters of the video like the illumination conditions, weather conditions or any other contingencies in the background like a tree or a pole in the middle.

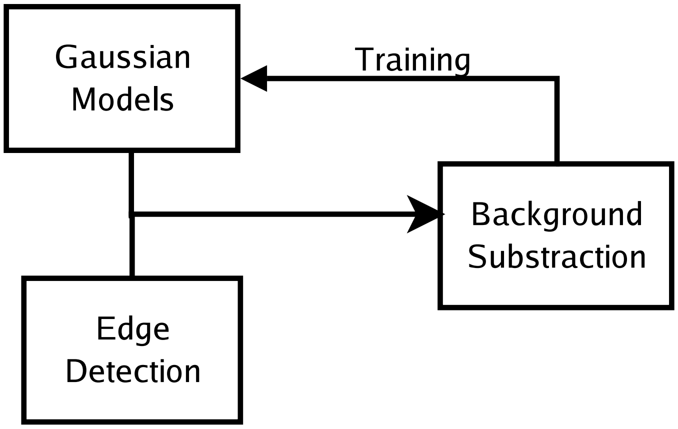

Figure 2: Block Diagram of Work Done so far.

III. Background

The long term objective of this project is to investigate machine vision technologies for automatic monitoring of railcar health. The focus of the proposed work here will be on automated, vision based monitoring of the health of the safety appliances.

The challenge of the problem arises from the large number, variety and locations of the appliances found in railcars, coupled with the wide range of car types and designs. Each such combination must be analyzed for being healthy, e.g., to determine if the appliances are in proper configuration. Further, the appliances must be inspected for safety, and recognized as safe or unsafe; in the latter case, the system must enumerate safety violations in terms of, e.g., missing, damaged or mislocated appliances, and provide pertinent details such as measurements of clearance distances and degrees of misalignment.

The first part is concerned with the design of an image acquisition system capable of acquiring the images of the railcar. The second part involves development of algorithms for recognition of healthy configurations of appliances. The third part is aimed at eliciting, from the results of the recognition algorithms, specific ways in which the appliances fail safety checks and associated parameters.

IV. Project Deliverables

We intend to build a system which can take in a video file as input and inspect the containers of the railcar for any damages in the safety appliances. We intend to make the system such that it can identify a particular kind of containers and model it.

We intend to do it with learning algorithms which would be based on the video and would be dynamic depending on the video. It should also model a design for the train and then check for the various patterns and the condition of the safety appliances.

V. Major challenges

- The major challenge of the problem would be to deal with the various types of architechtures and designs of the railcars. Also, there might be some paintings and posters on the railcars which would hinder the automation process.
- Background Extraction would sometimes become tough as there is no definite pattern for it.
- The automation system should be independent of the climatic conditions like snow, rain or the lighting conditions. Shadows will depend on the lighting conditions and the time of the day when the data is taken, which should be taken care of.
- The features required to ensure proper functioning of the safety appliances might not be present in a single frame and so some mosaicing has to be done.

VI. Procedure

The input to this system would typically be a video of a passing freight railcar. We would first extract images of the railcar and then determine the features that represent the various safety appliances. Next step would be to develop a set of image characteristics that will be considered for capturing distinguishing characteristics of safety appliances. Using image processing techniques those features can be determined and the working condition of those safety appliances can be evaluated. Learning algorithms can be used as well as new ones could be developed to train the system to detect the defects.

The first and one of the most critical parts of the implementation is the background extraction which is what we are currently working on.

VII. Implementation Details

The typical input for this system is an avi file of the moving train. Some initial data which has been collected using a preliminary system is already available with us. Hence, the first step would be extraction of the frames. It is a VC++ (6.0) project. The exe file is run with input parameters as avi file and the destination folder where to store the extracted frames. It outputs one frame at the end of each loop.

Now that we have done the frame extraction, the work that proceeds will be on the images. Next step is the background extraction i.e., elimination of the unnecessary part of the images. There are no specific

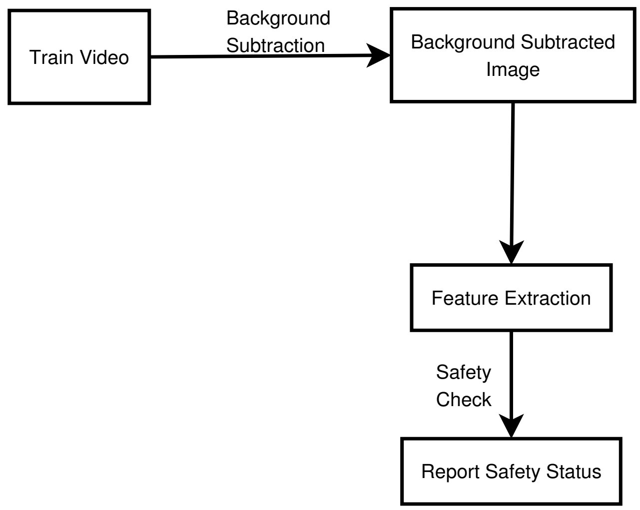

Figure 3: Block Diagram of the Project.

algorithms which can be applied directly to obtain the background subtracted image efficiently and with guarantee. The efficiency of the algorithm depends on the input video. An algorithm which gives very good results with a good video may fail miserably when there are some contingencies like a tree in the background or climatic conditions like snowfall or for that matter time of the day when the video is recorded. Hence, a robust algorithm has to be developed to overcome these problems and to get the background subtracted image efficiently irrespective of the contingencies in the video.

Following are some of the algorithms/approaches we tried to get best results.

A. Gaussian mixture model

Graphical models are powerful tools for processing images. An image texture classification method based on Multi Gaussian mixture models of sub-band coefficients is used in this model. Gaussian mixture models classify the background pixels into clusters. Training for the Gaussian mixture is done using the first N, say first 30 frames of the video, which would contain only the background. As soon as the train enters the frame the system determines if the new pixels are close to the background model and classifies it as background based on some threshold. Else the pixel is classified as foreground or as pertaining to the railcar. The number of Gaussian models that are to be used needs to be determined. After extensive analysis we found out that mixture models in the rang of 3-5 give the best accuracy for the background detection.

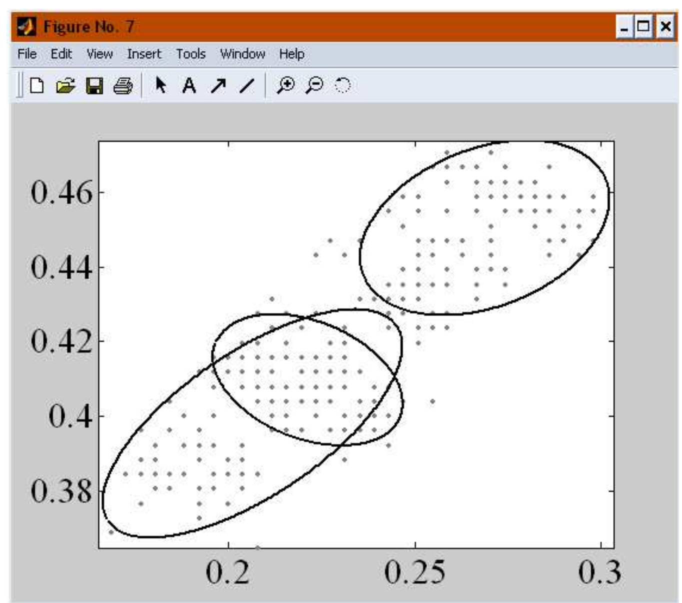

Figure 4: Gaussian Models

B. Edge Detection

Next approach was to detect the edges in the images. Detection of edges can help in getting the boundary of the railcar and hence can solve the problem of background subtraction because then we would know whether a particular pixel belongs to foreground i.e., the railcar or lies in the background.

- Using a Mask: The method we used for the edge extraction was Prewitt which gave best results among the other methods like LoG, Sobel and Canny filters. In this case, we used both grayscale and colour outputs to get efficient results. For the purpose of detecting weak edges, we saved the magnitude of the derivatives of the image using the prewitt operator instead of comparing with the cutoff. To detect the long, vertical edges we used a larger mask than the usual 3x3 mask. We tested it on the images with a various mask sizes and 21x21 mask gave the most satisfactory results.

Looking specifically for long edges would be ideal to find out when a container has entered or is about to leave the field of view of the camera. This would still not solve the background between containers problem. For this, some kind of contour of the railcar needs to be extracted. Such contours (extracted per frame) can be matched across frames to ensure that the contour is indeed that of the rail car's. All pixels outside these regions are background and can be learnt and a learning algorithm can be developed.

Assuming that during the first few frames of the train's arrival into the camera FOV, the background subtraction will be correct, this approach will further ensure that background learning does NOT occur on the pixels of the train. The final system can be a combination of the learning approach, and the above one. This forms a sort of control system, wherein the learning approach, classifies each pixel based on colour, and the above approach ensures whether such a classification is correct.

C. Improved Edge Detection

In the previous approach of background detection the edge detection was for container's vertical edges (sides of the container) worked well. The next idea is to extend this edge detection into a model of container detection by doing the following:

- Instead of producing a binary image output for edge detection, we will output an image with probabilities of each pixel belonging to an edge.The idea is to get a good estimate of the velocity of the train and identify if we can distinguish between whether a particular pixel is within a container or outside it, using just the edge information. We must process this "image" of probabilities to identify edges (ideally a set of points with high probability oriented in a vertical direction).
- After identifying, track the vertical edges corresponding to the front of the train. This can be done using the velocity estimate of the train. This velocity estimate is the amount of pixels the train moves per frame. We used non maximal suppression to detect the vertical edges better. Non maximal supression would involve using a small mask similar to a thin strip and checking if the value of the edge is greater than the neighbourhood pixels in the grey image that we have generated. This is to account for the weak edges present in the given image. In the non maximal suppression method, we process the edge detected gray images and look for the local maxima of 5 pixels along the vertical edge direction. The results were satisfactory and the performance best with a 19 x 19 or a 21 x 21 mask. The 11 x 5 masks did not work properly even in this case also. With a large mask, the noise i.e, the smaller edges were not detected and only the long vertical edges required were detected. We used a 5 x 1 mask for non maximal supression.

D. Line Identification

After the non-maximum supression implementation, the next step was to be to use a line fitting algorithm or otherwise and determine the end points of those edges.

Implementation Details :

1. The binary image contaning the edges after Prewitt Horizontal operator and non-maximal supression would be given as the input.

2. Region growing was used to determine the edge points. The region growing was done by looking at the pixels in the 6 directions, i.e, 3 pixels on the top of the current pixel and the bottom 3 pixels out the eight neighbourhood pixels. This is because we are looking at only the vertical edges.

3. Only edges and their corresponding points with lengths greater than a specified length would be detected.

4. These edges would be displayed with a red cross at the edge end pixels and the edges would be displayed in red.

E. Edge Tracking

Each edge is given a random colour, and is tracked in the next frame. A set of colors are specified initially

Figure 5: Image of the Train

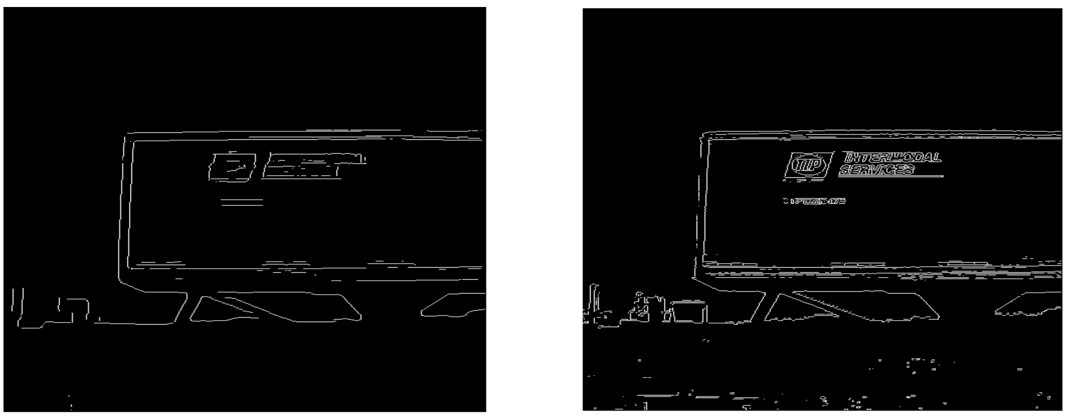

Figure 6: Edges Using 17x17 mask and 3x3 Prewitt mask

and each edge is given a different color. We find an edge (velocity +/- threshold) pixels away from the current x coordinate of the current edge. (Assuming the train moves horizontally). The features that we are looking at for a edge in the next frame to be the same as the edge in the current frame are:

- A comparison of the sizes of both the edges, based on the percentage value which can be specified as a parameter to the function.

- We have computed the distance between the end points and the average middle pixel and not the distances for all the points since we can asssume that the edges will be fairly straight. If the absolute value of the distance minus the velocity is less than a threshold, which is the size of our search say 5 pixels.

  That edge is coloured with the same colour as the current edge. Any edge that newly enters the current frame is a new edge is coloured separately.

F. Currently working On

We are now testing the above implementations on the available data applying different masks to the frames after the train enters. This is the Edge detection part which we have already done. Now, we keep a note of these edges and try to track them in the next frames. In the next frame, we search for these edges at a position (velocity +/- threshold) pixels away. All the edges that are present in the next frame also will belong to the

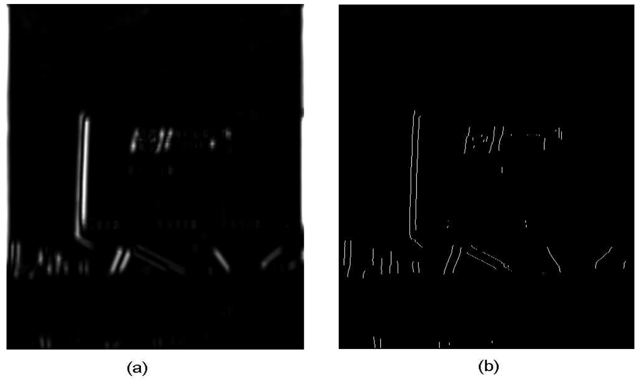

Figure 7: (a) Grey Scale Image of the Edges Detected. (b) Non maximal supressed edges

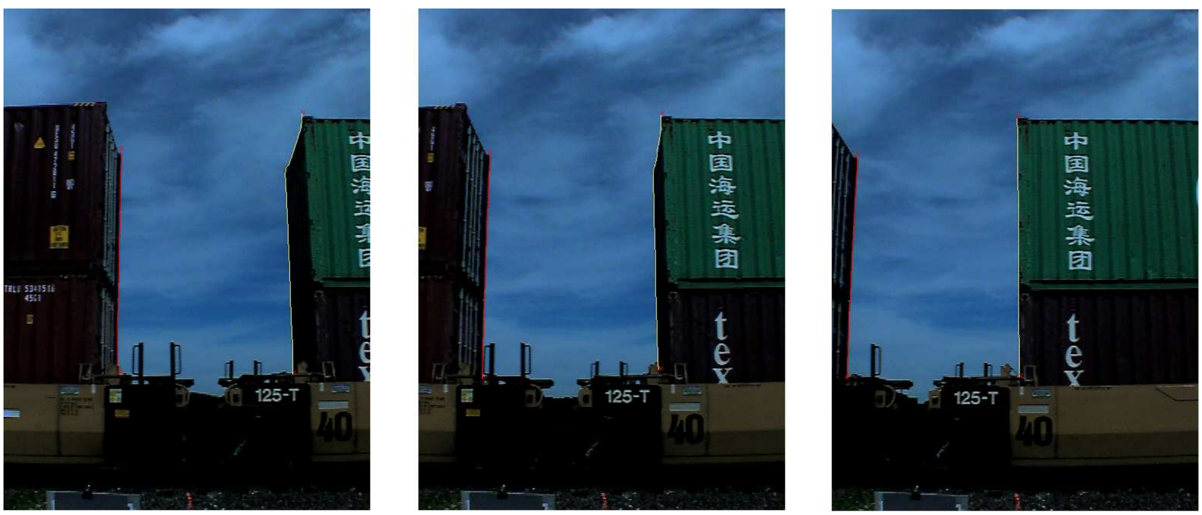

Figure 8: Edges Tracked Over Frames (Represented in Red and Yellow)

train and the rest are discarded. The background pixels that are identified at this stage will be used to train the system so that the background model can be updated.

VIII. Next Month's Plan

The next step would be to identify the contour of railcars robustly. We would be able to classify pixels as belonging to train/background and thus ensure that background learning is proper.

We need to make the edge detection more robust, typically by having a smaller mask so that small edges also are detected and then eliminate the spurious edges through comparison with previous frames, and using the velocity of the train.

- We are working towards making the algorithm more robust and also at intergrating the different clues that we are able to extract from the video. We are making use of the color aspects (Gaussian Mixture Models), velocity and shape aspects (Edge Detection). Hence instead of producing a binary image output for edge detection, we will output an image with probabilities of each pixel belonging to an edge.The idea is to get a good estimate of the velocity of the train and identify if we can distinguish between whether a particular pixel is within a container or outside it, using just the edge information. We must process this "image" of probabilities to identify edges (ideally a set of points with high probability oriented in a vertical direction).

- Apply a smaller mask to the first frame after the train enters, and keep these edges. Now in the next frame, search for these edges at a position (velocity +/- threshold) pixels away. All the edges that are present in the next frame also, will belong to the train and the rest will be discarded.

- Also, for the contour detection for the rail car bottom, try to implement some algorithms based on Snakes algorithm, or some other contour extraction techniques.

IX. References

(1) R. Gonzalez and R. Woods Digital Image Processing, Addison-Wesley Publishing Company, 1992, p 199. (2) Video Modelling and Segmentation Using Gaussian Mixture Models Xiaoran Mo, Roland Wilson, 17th International Conference on Pattern Recognition (ICPR'04) - Volume 3 pp. 854-857