TRAFFIC FLOW FROM A LOW FRAME RATE CITY CAMERA

Evgeny Toropov[1], Liangyan Gui[1], Shanghang Zhang[1,2], Satwik Kottur[1], José M. F. Moura[1]

[1]Carnegie Mellon University
Electrical and Computer Engineering
Pittsburgh, PA, USA

[2]Instituto Superior Técnico
Instituto de Sistemas e Robótica
Lisbon, Portugal

ABSTRACT

Traffic flow in a city is a rich source of information about the city. Cities are being instrumented with video cameras. They can potentially generate continuously large datasets to be processed (big data). This paper reports on our current work to detect traffic flow from an on-line low quality, low frame rate city video camera. The paper details a pipeline of four main steps – background subtraction, scene geometry, car detection, and car counting, and it illustrates results obtained with processing video from a single camera.

1. INTRODUCTION

The video generated by a low quality city camera (say between one and two frames per second and 100k pixels/frame) is roughly estimated to be about 1GB to 10GB of data per day. This is continuous and in "perpetuity." This paper considers the detection and counting of cars moving in a segment of a city road monitored by one such stationary camera. The video is low quality, noisy, with compression artifacts, and low frame rate. Due to the large depth of view of the camera, cars can be very small or very large. Figure 1 shows two consecutive frames. The low frame rate precludes tracking-based detection techniques as observed from Figure 1, where most of the cars are present only in one of the frames. Depending on the time of the day, traffic can be light or heavy, and illumination can vary dramatically throughout the day and night or due to very different weather conditions. We consider a video from an 8 hour span (10:00AM to 6:00PM), which reduces the range of illumination variations, but where there are significant shadows that appear and disappear.

The low quality and low frame rate of the cameras makes the problem significantly difficult, compared to traditional surveillance task. First, tracking-based approaches fail as detected vehicles are far apart in the image across successive frames. Second, matching the cars to avoid over-counting is difficult as their size and orientation differs vastly over frames.

Brief review of the literature. Background subtraction is limited by noise, variations in illumination, and shadows. Typical techniques for background subtraction include: 1) differentiating from each frame a background reference frame assumed to exist and with no foreground objects [1]; 2) gradient-based methods [2]; or 3) building background models with Gaussian Mixture Models (GMM) [3]. In this paper, we adopt a GMM-based method. Approaches for detecting vehicles include model based methods that use prior information [4] or deformable templates [5]. There are two methods for counting moving vehicles: tracking the paths of these vehicles [6]; and setting a virtual line and counting vehicles that pass the line [7, 8]. Because of the large variation in the size of the cars and the depth of the field of view in the images we consider, we partition each image into $1 + 4$ regions. The first is the near field of the image. In each of the remaining four, cars are of roughly the same size and moving in the same direction. To help with this image partition, we have an initial block that extracts the scene geometry.

Section 2 introduces the background subtraction procedure, Section 3 explains the scene geometry, Section 4 and Section 5 detail the detection and counting blocks, and Section 6 concludes the paper.

Fig. 1: A pair of sequential frames; one car correspondence is shown for clarity, smallest seen cars are 10 pixels.

2. BACKGROUND SUBTRACTION

The first step in our pipeline is to extract the regions of interest (ROIs) by subtracting the background from each frame in the video. We train adaptive background Gaussian Mixture Models (GMM) [9] to identify foreground pixels and obtain a foreground mask. Noise is filtered using erosion and subsequent dilation [10] with kernels of small radius. The refined foreground mask is used as an input for the scene geometry in Section 3. We then group foreground pixels to generate connected regions according to their sizes, moments, and shape information, and obtain the minimum bounding rectangles (blobs). We filter these blobs with the geometry constraints. In particular, we filter out cars that are significantly bigger or smaller than the expected size in the particular point being considered on the road. We also filter blobs by aspect ratio, which is approximately height/width $= 0.75$ in our case. These remaining blobs can aid in the detection process in Section 4.

In light traffic, when the background to foreground area ratio is a two-digit value, the background GMM does learn a robust model. However, during the 5 pm rush hour (figure 2a) the ratio of background to foreground gets closer to 1, and some parts of the road are constantly hidden by cars. Therefore, GMM does not learn the true road (figure 2b). We take advantage of the fact that our cameras are stationary. We subtract the background image from each frame, re-

This work is partially supported by ONR grant N00014-12-1-0903. The third author is supported by a fellowship from the Fundação de Ciência e Tecnologia (FCT) from Portugal through the CMU|Portugal Program.
{etoropov, lgui, shanghaz, skottur, moura}@andrew.cmu.edu

Fig. 2: From left to right: a) frame in heavy traffic; b) noisy GMM-generated background; c) reference picture of empty road; d) map of brightness adjustment; e) resulting clean background.

sulting in a ghost-like appearance of cars (figure 4). Next we explain how to generate a clean background in all traffic conditions.

Generating a clean background image from a frame is not always a trivial task. To solve this problem, we use a reference picture with empty road taken at times of light traffic – for example, at 10 am (figure 2c). We transfer this reference picture of empty road to other times of the day after brightness and color adjustments (figure 2d). Weather and time of the day change the illumination of the road smoothly but non-uniformly. Therefore, color and brightness adjustments of the reference picture must be non-uniform as well.

First we generate a noisy background image from the GMM (figure 2b). Pixels too different from the reference picture are considered as noise and are masked. Then we apply a Gaussian blur with a large $\sigma = 50$ to both the reference picture (figure 2c) and to the GMM-generated background, in order to average the color in the neighbourhood of every pixel. The masked pixels do not contribute to the Gaussian blurring. The difference between the two blurred images (figure 2d) is then added to the reference picture. The final clean background image is shown in figure 2e. The goal of generating the clean background image is to be able to create "ghost images" from input frames, thus the quality of the background image is evaluated in terms of how clean the dark parts of the ghost images are. The mean absolute value of the dark pixels of the ghost image is 2.8 ± 1 for the method described here. It is much lower than 4.6 ± 1 when the reference image is used without illumination adjustment and is close to the level of color noise 1.7 ± 1.

3. SCENE GEOMETRY

Working with a single, stationary camera provides the luxury to learn and understand the geometry of the scene. By scene geometry, we mean the location and extent of the road along with lane structure, as seen by the camera. The usefulness of such an understanding is two-fold. First, it provides context for the entire system. As the camera always sees a fixed stretch of the road, incorporating such knowledge about the scene will aid car detection and counting, leading to a more robust system. Second, it gives additional information about the possible sizes and location of candidate detections that can be used to reduce false positives (see figure 3a). For example, we can identify false alarms when the size of a candidate detection is suspiciously larger than the expected size, given the location of the detection in the fixed viewpoint. In this paper, we take advantage of this single camera setting and understand geometry with identifying the road lanes.

Overview. Identification of the lanes from a single view camera can be done using two broad approaches−making use of visual cues like lane markers and dividers that are painted in white or yellow and standout from the background [11],[12]; and tracking the vehicle moving along the lanes across various frames [13], [14][1].

These methods place restrictive assumptions or are not feasible in our current setting. First, depending solely on visual cues for lane detection forces a need for clear and distinct lane marks, which might not always be true for the given stretch of road. Because scene geometry is significant in improving car detection, unclear or no road marks for the lanes render the system ineffective. Second, low frame rate of the camera we consider disallows approaches that rely on tracking moving vehicles to be successful. Additionally, our camera is uncalibrated. In this paper, we propose a novel approach to detect and identify lanes using the first few frames of the input video stream, as described next:

1. The pipeline begins with estimation of the vanishing point and dominant edges of the road. These edges roughly define the extent of the road, helping us narrow our search for moving cars within these bounds.
2. With the help of the vanishing point estimates, we generate an Inverse Perspective Map (top view) of the road using planar homography, thereby eliminating the perspective effect.
3. We find the candidate pixels that possibly correspond to vehicles after subtracting the background (see section 2). We then compute the vertical lines in the top view where there is minimum or no foreground over a few frames, and identify them as the lane boundaries.

Vanishing point estimation. The primary step in understanding the scene geometry is to estimate the vanishing point along with the two outermost dominant edges on either side of the image. It is well known that perspective projection causes parallel lines, road lanes in our case, to meet in a point. The dominant edges detected are likely to correspond to the outermost extents of the road. We use the vanishing point detector proposed in [15] for this task. The authors use soft voting, based on local texture features for vanishing point detection and then go on to estimate the dominant edges by finding the two lines with maximum texture orientation consensus. We refer the reader to [15] for further details. Since voting is done over all the pixels on the image, the estimation of vanishing points may slightly vary for different frames. We use the first N_v frames to compute the vanishing point and dominant edges, and then perform RANSAC [16] to select the best fit. Figure 3b shows the resulting image after this process.

Top view generation. Using the estimated vanishing point and dominant edges, we generate a top view of the road through a planar homography transformation. Working with the top view offers two significant advantages: i) We can get away with the perspective effect where the lanes seem to meet at a point. ii) We now have vertical lanes that can be easily identified through known techniques.

[1] Although [12] and [14] work with a vehicle mounted camera, as opposed to the surveillance camera in the current problem, they still provide the intuition behind the difference between the two approaches.

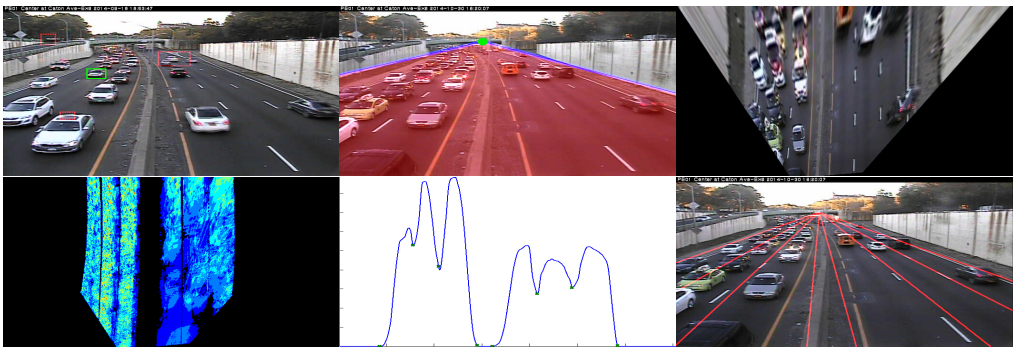

Fig. 3: *From left, top row:* a) Validating detections using scene geometry. *Red:* Possible false positives−unexpected size or location; *Green:* Acceptable car detection as size is within tolerance; b) Estimated vanishing point (*green*), dominant edges (*blue*), possible extent of road (*red*); c) Top view resulting from homography transformation[2], cars mostly lie in their corresponding lanes even though homography causes distortion, the bleeding into neighboring lane is minimal; *From left, bottom row:* d) Accumulated foreground over the first N_d frames, notice the clear minima along the directions; e) Plot after summing foreground map along the columns of (d), minima (*green*) correspond to lane boundaries; f) Final result after detecting the lanes.

The latter is true, as we assume that the lanes are parallel or nearly parallel. It must be noted that homography is valid strictly only for points on the road plane whereas pixels corresponding to the cars will get distorted. As the vehicles typically move along the center of a particular lane, there is minimum 'leakage' of foreground into the neighbouring lanes, justifying its use in our approach. Figure 3c illustrates this in greater detail.

Road lane detection. The final step is to detect the lanes using foreground extracted and the homography that generates the top view of an image. The task is to identify the vertical lines, i.e., columns in figure 3c, which correspond to lane boundaries. Since vehicles generally move well within the lane, they contribute to higher number of foreground pixels at the lane centers creating local minima near the lane boundaries that we desire to identify (figure 3d). We apply the homography transformation to the extracted foreground image. For every vertical line (each column in figure 3c), we keep track of the total number of foreground pixels belonging to that line, across the first N_d frames, i.e., we sum up all the rows to get a histogram of foreground pixels for each vertical line. Many such histograms, one for each new frame, are added to obtain the distribution of the foreground pixels for these N_d frames. We smooth out this accumulated histogram to avoid detecting multiple minima at the same place, and then find the points of local minima labeling them as lane boundaries (figure 3f).

4. CAR DETECTION

Choice of detectors. We now consider the car detector. It is desirable to detect a bounding box for every visible car in every frame. The main challenges arise because of heavy traffic, very low frame rate (around 2 sec. per frame), low image quality, and image compression artifacts. Relatively heavy traffic makes cars block one another in the image limiting applicability of background-based techniques [4]; the low frame rate effectively makes tracking of already detected cars not feasible, and the low image quality and image compression artifacts add additional noise to the image. Given these limitations, we look for a combination of several different approaches that produce a satisfactory result in most cases in our application

Given a camera, a detector should be able to find both very small cars on road segments farther away from the viewer, and big cars, closer to the viewer. In our videos a human can distinguish a car

Fig. 4: left: original frame fragment with and without subtracted background (car "ghost"); right: examples of car patches for training cascade detectors

10-15 pixels wide (figure 1), which is not as small as expected due to color noise and compression artifacts. At the same time, in the front of the frame, even small parts of cars become visible. A detector should also work in both light and heavy traffic conditions. We found that in very light traffic a detector based on background subtraction (like [4]) is capable of finding cars throughout the frame, while for heavier traffic, a Viola-Jones cascade detector [17] can be used for the farther away parts of the frame. The Viola-Jones detector, however, has poorer performance in closer parts of the image. For these, it is possible to use more complicated approaches like part-based models [5], but a simple background subtraction based detector will also perform well since individual cars in the front of the image and the road between them are usually visible even in heavy traffic. In the remaining part of the section, we explain how we build the Viola-Jones cascade detectors, and combine the cascade detectors with blobs provided in Section 2.

Viola-Jones cascades. We now consider the Viola-Jones cascades of Haar features. The cascades are trained and used with ghost-like images of cars with subtracted background (figure 4.) This mitigates the effect of such background features as lane markings on car appearance and reduces the number of false positive detections on the background. A clean background is generated at all times as described in Section 2. Next we explain how we trained the cascade detectors.

[2]We avoid pixels close to the vanishing point as they result in huge distortions due to perspective effects

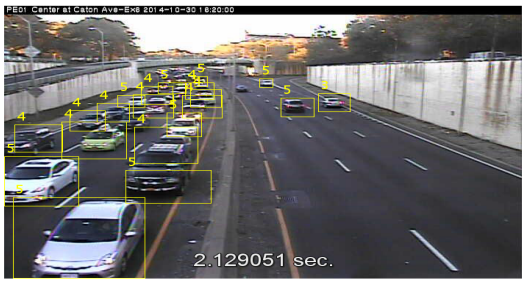

Fig. 5: Example of combining the two cascade detectors and the background subtraction based detector

Training the cascade detectors. We collected training data with blobs obtained in Section 2. Data were collected from several cameras at light traffic conditions, when the background detector accuracy is high. Since the available data is unlimited, we adjust parameters for very high accuracy with lower recall in order to minimize the human input. The remaining false positives are then manually filtered out. We harvested over 6000 car detections after processing 5 hours of video from different cameras (figure 4.)

We trained two cascades of Haar features. One cascade model fires on straight-looking cars, and the other – on cars viewed under some angle to the camera. Accordingly, we extracted two subsets of car patches from the available training data. We assigned a mask to each of the two cascade detectors based on the area of detector's applicability.

Combining detectors. At detection time, the two Viola-Jones models and the background based detector (3 detectors total) each process the provided frame and return their detections. We threshold the candidates based on expected size, expected ratio, and proximity to image boundary, and use standard non-maximum suppression [18] to generate a final set of detections. Figure 5 presents an example of detection results in the difficult case of heavy traffic. With the 19 visible cars, there are 17 correct detections along with 3 false positives. The other 2 cars were not detected.

Failure cases. We trained only two Viola-Jones cascade models, which cover most cases for this particular camera. The failure cases for Viola-Jones include trucks and buses, dusk and night conditions, as well as dense traffic. Some of the vehicles in those cases are successfully detected by the background based detector, such as the car in the front of the camera in figure 5. On the other hand, the background based detector may detect a group of car as a single vehicle, and it may see two cars in place of a single one in case it has a large gray front window. Most mis-detections are successfully filtered out using geometry constraints.

5. CAR COUNTING

The last block of the pipeline counts the number of cars in a given video sequence based on the cars detected by the detector block. There are two main challenges – low frame rate and low resolution. While low frame rate renders the traditional tracking ineffective, lack of resolution hinders matching using common features [19] [20]. To overcome these challenges, we propose a probabilistic counting model, including following steps: i) matching cars in the new frame and in the previous frame to distinguish new cars from seen cars; ii) counting the new cars in the new frame and adding them to the total count of cars in the video sequence.

To improve the accuracy of car matching, which is the key component of the probabilistic counting model, we simultaneously extract multiple representative features, utilize appropriate distance measurements, and consider geometry constraints. For the feature extraction, we combine the HOG features with color histogram. To measure histogram similarity, we apply the Earth Mover's Distance (EMD) [21]. Compared to traditional similarity metrics such as the χ^2 distance, EMD is more robust, being based on several desirable features including: the support of adaptive binning and support of partial matches, which make EMD more effective to measure the similarity in car matching. Regarding geometry constraints, we pairwisely compute the matching score between two cars based on geometry information such as lanes, speed, and potential driving distance. In order to combine the appearance similarity, and geometric constraints, we compute a weighted score based on the color matching score, HOG matching score and geometry matching score. A cutting line is adaptively set in frames based on geometry information, and cars farther than the cutting line are not considered. The final result is matches between detected cars in sequential frames and the total number of cars in a given video sequence. Figure 6 illustrates the matching results for a pair of sequential frames.

Quantitative Evaluation. In the experiments, we evaluate the proposed algorithm on a manually labelled representative video sequence. The ground truth vehicle count is 321. The proposed algorithm counted 275 of them, and detected 16 more false positives.

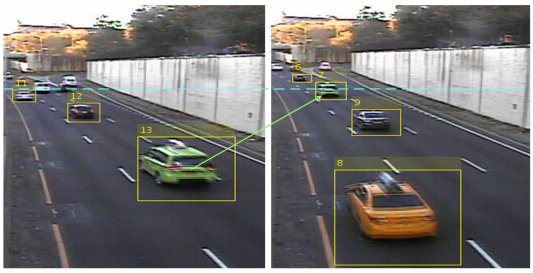

Fig. 6: Counting results for a pair of sequential frames. Only detected cars below the cutting (dashed) line are counted. In the left frame, the number of cars is 3; In the right frame, 1 car is matched with a car in the previous frame. 2 cars are not matched and are denoted as new cars, which makes the total of 5 in the pair of frames.

6. CONCLUSION

This paper describes a pipelined system to detect and count cars from a low quality, low frame rate, stationary city video camera. This is a challenging problem compounded by the depth of camera, compression artifacts, noise, variations in illumination, low frame rate, among other limitations. The video we consider spans an eight hour period of a single day, which limits the range of illumination variations. We combined two complementary methods for background subtraction to address different traffic conditions. The solution presented exploits the scene geometry extracted as explained in Section 3 and uses multiple detectors, each addressing a part of image where car sizes and traffic direction are considered more homogeneous. We also presented a traffic counter. The representative results with real video from a city camera provide accurate identification and count of cars. In our future work, we will consider night and day as well as weather variations, and videos from other cameras.

7. REFERENCES

[1] Chomtip Pornpanomchai, Thitinut Liamsanguan, and Vissakorn Vannakosit, "Vehicle detection and counting from a video frame," in *International Conference on Wavelet Analysis and Pattern Recognition, ICWAPR 2008, Hong Kong, China*. IEEE, August 2008, vol. 1, pp. 356–361.

[2] Kantip Kiratiratanapruk and Supakorn Siddhichai, "Practical application for vision-based traffic monitoring system," in *6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2009, Pattaya, Thailand*. IEEE, May 2009, vol. 2, pp. 1138–1141.

[3] Pablo Barcellos, Christiano Bouvié, Fabiano Lopes Escouto, and Jacob Scharcanski, "A novel video based system for detecting and counting vehicles at user-defined virtual loops," *Expert Systems with Applications*, vol. 42, no. 4, pp. 1845–1856, 2015.

[4] Jin-Cyuan Lai, Shih-Shinh Huang, and Chien-Cheng Tseng, "Image-based vehicle tracking and classification on the highway," in *International Conference on Green Circuits and Systems, ICGCS 2010, Shanghai, China*. IEEE, June 2010, pp. 666–670.

[5] Akihiro Takeuchi, Seiichi Mita, and David McAllester, "On-road vehicle tracking using deformable object model and particle filter with integrated likelihoods," in *2010 IEEE Intelligent Vehicles Symposium, IV 2010, San Diego, US*. IEEE, June 2010, pp. 1014–1021.

[6] Belle L Tseng, Ching-Yung Lin, and John R Smith, "Real-time video surveillance for traffic monitoring using virtual line analysis," in *IEEE International Conference on Multimedia and Expo. ICME 2002. Lausanne, Switzerland*. IEEE, August 2002, vol. 2, pp. 541–544.

[7] Thou-Ho Chen, Yu-Feng Lin, and Tsong-Yi Chen, "Intelligent vehicle counting method based on blob analysis in traffic surveillance," in *Second International Conference on Innovative Computing, Information and Control. ICICIC 2007. Kumamoto, Japan*. IEEE, September 2007, pp. 238–238.

[8] Deng-Yuan Huang, Chao-Ho Chen, Wu-Chih Hu, Shu-Chung Yi, and Yu-Feng Lin, "Feature-based vehicle flow analysis and measurement for a real-time traffic surveillance system," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 3, no. 3, pp. 279–294, July 2012.

[9] Chris Stauffer and W Eric L Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* IEEE, 1999, vol. 2.

[10] Robert J Schalkoff, *Digital image processing and computer vision*, vol. 286, Wiley New York, 1989.

[11] Wook-Sun Shin, Doo-Heon Song, and Chang-Hun Lee, "Vehicle classification by road lane detection and model fitting using a surveillance camera.," *Journal of Information Processing Systems (JIPS)*, vol. 2, no. 1, pp. 52–57, 2006.

[12] Mohamed Aly, "Real Time Detection of Lane Markers in Urban Streets," in *IEEE Intelligent Vehicles Symposium*, June 2008, pp. 7–12.

[13] Jakub Sochor, "Fully automated real-time vehicles detection and tracking with lanes analysis," in *Proceedings of Central European Seminar on Computer Graphics (CESCG)*, Vienna, Austria, 2014, pp. 59–66, TU Vienna.

[14] J. C. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 20–37, 2006.

[15] Hui Kong, Jean-Yves Audibert, and Jean Ponce, "Vanishing point detection for road detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, FL, USA, 2009, pp. 381–395.

[16] Martin A. Fischler and Robert C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[17] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Dec. 2001, vol. 1, pp. 511–518.

[18] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros, "Ensemble of exemplar-svms for object detection and beyond," in *2011 IEEE International Conference on Computer Vision (ICCV), Barlecona, Spain*. IEEE, November 2011, pp. 89–96.

[19] Gautam S. Thakur, Mohsen Ali, Pan Hui, and Ahmed Helmy, "Comparing background subtraction algorithms and method of car counting," in *arXiv*. 1202.0549v1 [cs.CV], Jan. 2012.

[20] Takayuki Katasuki, Tetsuro Morimura, and Tsuyoshi Ide, "Bayesian unsupervised vehicle counting," *In Technical Report, IBM Research RT0951*, 2013.

[21] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40(2), pp. 99–121, 2000.