Tracking Based Semi Supervised Learning using Background Subtraction - Classification (BSC) Model

Jinjian Zhai
Stanford University
jameszjj@stanford.edu

Brian Chung
Stanford University
bpchung@stanford.edu

Abstract

In this paper, we tackle the issue of using semi supervised learning in classifying objects with tracking information. Based off the earlier work of Teichman and Thrun, we modify the approach using LIDAR data to extracting useful object classification information from a single fixed camera source.

Our track data comes from background subtraction and segmentation of camera video data. Using semi supervised method, a few objects are labeled. Consequently, a lot of labeling data are obtained automatically using the tracking information and that data is used in training the object classifier.

The classifier will be based on both positive and negative samples, and the resulting objects are further tested and classified. Experiments are analyzed and the performance of the classification of different object classes are evaluated.

1. Introduction

Currently, there is a great need for high throughput classification of video data. Rather than manually hand labeling individual video frames, we seek to automate the process through model free segmentation and track classification. In particular, we separate the process of learning into three separate tasks.

Using background subtraction and bi-layer segmentation, a user will hand label a few of those objects and much more labeling data will be obtained automatically using the tracking information . The set of tracks are fed into a classifier which will also classify future objects as being part of that object class.

As seen in Teichman's work [2], tracking based semi supervised learning is surprisingly resilient to noise and has an uncanny ability to learn new useful instances of the classes.

Difficulties arise in proper background subtraction and proper handling of edge cases (*e.g.* intersecting tracks).

Accurate background subtraction will be vital to identifying proper tracks. Due to the fixed camera, we can obtain a background model easily from the data. We will begin with using Gaussian modeling for quick background subtraction. Shah's method of Bayesian object detection is researched [7]. Pending background subtraction, tracks of objects can be gleaned from each sample.

The classifier is an off the shelf classifier of Felzenszwalb [13]. Once a series of training data is sampled along the track of the object, new testing frames of the class are identified,

2. Background and Related Work

This section will introduce the literary background in Teichman's original project, background subtraction, and classification.

The project was extended from the Tracking-based semi-supervised learning project by Alex Teichman [1] at Stanford AI Lab. The original project solved the problem of track classification in dense 3D range data [6, 5, 4, 2, 3]. The semi-supervised object recognition method uses tracking information and dramatically re-

duces the amount of hand-labeled training data required to produce a reliable detector. The method is based on EM algorithm. It runs iteratively between training a classifier and extracting useful training examples from unlabeled data by exploiting tracking information. The final result can be based on three hand labeled training tracks of each object class and reach a final accuracy comparable to that of the fully supervised equivalent. But this project only used expensive laser range finder on Stanford's autonomous vehicle. In our project, we extend to cheap camera-based systems, which is more possible to be installed on vehicles.

In the part of background subtraction, we research the work of Lo [17], Cucchiara [18], Koller [19], Wren [20], Wang [9], Stauffer [21, 22], Criminisi [8], Elgammal [23], Han [24], Oliver [25] and Seki [26].

In Lo's paper [17], an automatic monitoring system is proposed for detecting overcrowding conditions in the platforms of underground train services. The system is designed to use existing closed circuit television (CCTV) cameras for acquiring images of the platforms. In order to focus on the passengers on the platform, background subtraction is used. A variance filter is also introduced to optimize the removal of background pixels.

In Cucchiara's paper [18], background subtraction methods researched to update the background model and to deal with shadows. The algorithm combines statistical assumptions with moving objects, apparent objects and shadows. Pixels belonging to such objects are processed differently in order to supply an object-based selective update. It also exploits color information to improve object segmentation and background update.

The are quite a few innovations in Criminisi's paper [8]. In his paper, the pixel velocities are not used, thus the need for optical flow estimation is removed. Instead, an efficient motion vs nonmotion classifier is trained to operate directly and jointly on intensity-change and contrast. Its output is then fused with colour information. The prior on segmentation is represented by a second order, temporal, Hidden Markov Model, together with a spatial MRF favouring coherence except where contrast is high. Thus the algorithm does real-time separation of foreground from background in monocular video sequences.

In Wren's paper [20], a real-time system for tracking and interpretation of people is built on a multi-class statistical model of color and shape. The system obtains a 2-D representation of head and hands.

In Wang's paper [9], he talked about the automatic segmentation of foreground from background in video sequences by directly subtracting a mean background image from each frame, and retaining those parts of the frame that differ the most from the background.

In Stauffer's papers [21, 22], he talked about a common method for real-time segmentation of moving regions in image sequences by thresholding the error between an estimate of the image without moving objects and the current image. The model treats each pixel as a mixture of Gaussians and using an on-line approximation to update the model. Then they are evaluated to determine which are most likely to result from a background process. Although the mixture of Gaussians and kernel density estimation approach suffer from the lack of flexibility by fixing limiting the number of Gaussian components in the mixture.

In Elgammal's paper [23], he talked about methods to segment moving regions in image sequences taken from a static camera by comparing each new frame to a model of the scene background. The model is non-parametric and handles the situation that the background is not completely static but contains small motions such as tree branches and bushes. The model estimates the probability of observing pixel intensity values based on a sample of intensity values for each pixel. The model can also use color information to suppress detection of shadows. It should be noted that the system requires a large memory by maintaining a non-parametric representation of the density.

In Oliver's paper [25], he talked about a real-

time computer vision and machine learning system for modeling and recognizing human behaviors in a visual surveillance task. The system deals in particularly with detecting when interactions between people occur and classifying the type of interaction.

In Seki's paper [26], he presents a background subtraction method for detecting foreground objects in dynamic scenes involving swaying trees and fluttering flags. He tried to narrow the ranges by analyzing input images and to improve the detection sensitivity by employing the correlation of image variations at neighboring image blocks instead of chronological background image updating .

In the part of classification, we research the work of Felzenszwalb [12, 14, 15, 13]. He developed a learning-based system for detecting and localizing objects in images, representing objects using mixtures of deformable part models. The good part of the model is that they only needs bounding boxes for the training of objects in an image. The model is based on histograms of oriented gradients (HOG) low-level features, pictorial structures of deformable part-based models (as shown in Problem Set 4 of the class) and discriminative learning with latent SVM.

3. Background Subtraction/Image Segmentation

3.1. Approach

Despite the multitudes of segmentation methods such as color or texture separation, they tend to fail when deriving background models from motion based video. These models typically derive a "mean" image based off the set of frames and subtract individual frames by the mean image to obtain the foreground objects.

The background subtraction method typically fails for real world data. Items such as trees, birds, and other variant objects can create false positives. In other instances, false negatives arise from collisions or occlusions of the species.

Because of those issues, our group decided to base our work off an implementation of Billayer Segmentation of Live Video by Criminisi, et al. [8]. Criminisi's approach uses a probabilistic combination of motion, color, and contrast in a Hidden Markov Model.

Our videos are first read by the matlab code and transformed into a series of images:

$$z = z^1, z^2, \cdots, z^t \tag{1}$$

The frames of images can be represented as pixels:

$$z = z_1, z_2, \cdots, z_i, \cdots, z_n \tag{2}$$

The derivatives of frame z is:

$$\dot{z} = \dot{z_1}, \dot{z_2}, \cdots, \dot{z_i}, \cdots, \dot{z_n} \tag{3}$$

Therefore, the pixel derivative at time t is:

$$\dot{z_i^t} = |G(z_n^t) - G(z_n^{t-1})| \tag{4}$$

where G is a Gaussian kernel with sigma σ.

We can denote the spatial gradients of the pixels as:

$$\mathbf{g} = g_1, g_2, \cdots, g_i, \cdots, g_n \tag{5}$$

, where $g_i = |\nabla z_i|$.

Therefore, the motion observables are:

$$\mathbf{m} = \mathbf{g}, \dot{\mathbf{z}} \tag{6}$$

Segmentation is denoted by $\alpha \in (\mathrm{F}, \mathrm{B})$, where F is foreground and B is background.

Using the notations shown above, we can use a Conditional Random Field [27] posterior model:

$$p\left(\boldsymbol{\alpha}|\mathbf{z}, \mathbf{m}\right) \propto \exp - \left[\sum_{t'=1}^{t} E^{t'}\right] \tag{7}$$

Where E^t is the energy term [8]:

$$\begin{aligned} E^t =& E^t(\boldsymbol{\alpha}^t, \boldsymbol{\alpha}^{t-1}, \boldsymbol{\alpha}^{t-2}, \mathbf{z^t}, \mathbf{m^t}) \\ =& V^T(\boldsymbol{\alpha}^t, \boldsymbol{\alpha}^{t-1}, \boldsymbol{\alpha}^{t-2}) \\ &+ V^S(\boldsymbol{\alpha}^t, \mathbf{z^t}) \\ &+ U^C(\boldsymbol{\alpha}^t, \mathbf{z}) \\ &+ U^M(\boldsymbol{\alpha}^t, \boldsymbol{\alpha}^{t-1}, \mathbf{m^t}) \end{aligned} \tag{8}$$

, where V^T is the temporal prior.

$$\begin{aligned} &V^T(\boldsymbol{\alpha}^t, \boldsymbol{\alpha}^{t-1}, \boldsymbol{\alpha}^{t-2}) \\ =&\eta \sum_i^n \left[-\log p\left(\alpha_n^t | \alpha_n^{t-1}, \alpha_n^{t-2}\right)\right] \end{aligned} \tag{9}$$

$p\left(\alpha_n^t | \alpha_n^{t-1}, \alpha_n^{t-2}\right)$ are the probabilities of α^t with different combinations of α^{t-1} and α^{t-2}. η is the discount factor for non-independent pixels. $p\left(\alpha_n^t | \alpha_n^{t-1}, \alpha_n^{t-2}\right)$ can be trained from ground truth.

In Eq. 8, V^S is the spatial prior with Ising term [28, 29] derived from the natural tendency of segmentation boundaries with high image contrast:

$$\begin{aligned} &V^S(\boldsymbol{\alpha}, \mathbf{z}) \\ =&\gamma \sum_{(m,n \in \mathbf{C})} [\alpha_m \neq \alpha_n] \left(\frac{\epsilon + e^{-\mu ||z_m - z_n||^2}}{1+\epsilon} \right) \end{aligned} \tag{10}$$

In Eq. 8, U^C is the log of color likelihood [30].

$$\begin{aligned} &U^C(\boldsymbol{\alpha}, \mathbf{z}) \\ &= -\rho \sum_i^n \log p(z_n | \alpha_n) \end{aligned} \tag{11}$$

In Eq. 8, U^M is the motion likelihood [8]. Each time step's motion model is augmented in order to favor "coherence" frame to frame and forgoes traditional optical flow estimation.

$$\begin{aligned} &U^M(\boldsymbol{\alpha}^t, \boldsymbol{\alpha}^{t-1}, \mathbf{m^t}) \\ &= -\sum_i \log p\left(\mathbf{m}_n^t | \alpha_n^t, \alpha_n^{t-1}\right) \end{aligned} \tag{12}$$

By using energy minimization [8], terms are chosen in order to reduce the fragmentation of objects within the frames. The result of $\boldsymbol{\alpha}$ is estimated as:

$$(\hat{\boldsymbol{\alpha}}^1, \hat{\boldsymbol{\alpha}}^2, \cdots, \hat{\boldsymbol{\alpha}}^t) = \arg\min \sum_{t'=1}^{t} E^{t'} \tag{13}$$

3.2. Implementation

Instead of large amount of hand labeled training data in background subtraction, we use background subtraction technique, which was based on Wang's implementation of Bilayer Segmentation in live videos [9, 10]. So that we only need initial hand labeled data of the desired species.

Wang's implementation only tracks one object throughout the sequence. In order to improve performance, the ability to track multiple objects was added.

Furthermore, the implementation did not keep track of an object's trajectory throughout the sequence. This meant that an object in one frame did not have a sequence of events in history to compare to. Thus an abstracted information to keep object histories was added. Such information was used to generate the bounding boxes with images for the training of classification.

Finally, constraining filters to maintain the bounding box to the object selected in the initial frame are added so that much more resulting data from the bilayer segmentation can be used to train the classifier. The Matlab code was developed based on Wang [9, 10] and Criminisi [8]'s code.

In our model the following object classes has been used:

1. pedestrain
2. biker
3. golf cart / mini van
4. bus / truck

Due to classification requirement of resolution, some classes are more successful in testing than others, which will be shown in the next section. Objects of all four classes are labeled and a lot of training data are obtained automatically.

130 minutes of video were fetched from the top of Hoover Tower [11] on Stanford campus. FFMPEG [16] was used to compile the video into useful movie clips with right format. The frame per second (fps) is 30 and the resolution are compressed to 480$\times$ 272 from 1920$\times$1088 original

format to fit with the memory limitation of background subtraction code in Matlab.

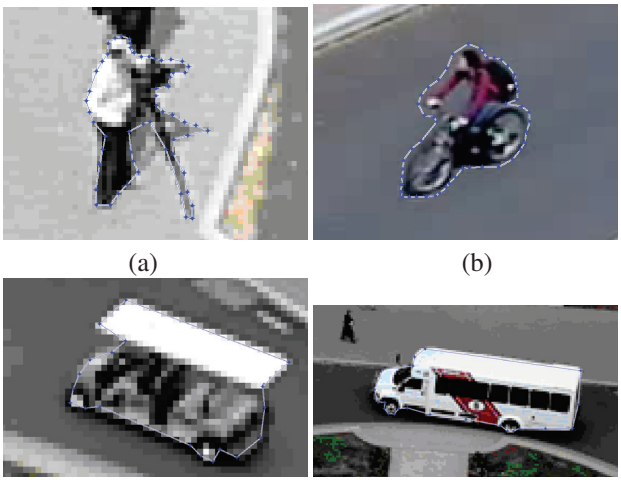

Figure 1. Example of masks for pedestrain (a), a bicycle rider(b), golf cart(c) and bus(d).

As shown in Fig. 1, the desired object in the first frame of the movie clips are enclosed by a polygon. The average length of the movie clips is 20 seconds, which is 600 frame images in a 30 fps format. Instead of manually masking 600 for the training set, we only need to mask one object for each video. It is two orders as simply as the traditional training process.

Both the background subtraction and classification are done on the linux system Ubuntu 11.10 version with Matlab R2010B and GCC-4.4 compiler. Video processing functions of Matlab were heavily used to decode the .avi and .mov files. The energy minimization code of Criminisi [8] is written in C++ and the Wang's wrapper [10] compiles them in Matlab, so a huge amount of time is saved comparing to pure Matlab processing code.

The tracking information of desired objects are provided in an array and are fitted with the corresponding pictorial information in time sequences. Then all the training results from hundreds of videos are exported in a unified numbering system to be feed into the classifier. The tracking information includes the time stamp, object classes, center position, bounding box generated from the mask sequence and original pictures.

3.3. Segmentation Experiment Results

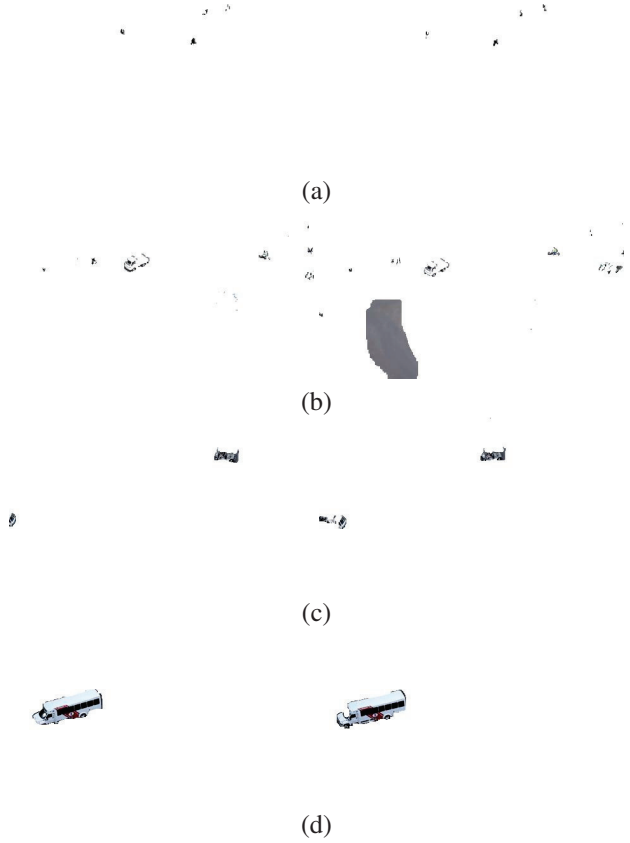

Figure 2. Some examples of mask results for pedestrain (a), a bicycle rider(b), golf cart(c) and bus(d).

As seen in Fig. 2, there are still difficulties in some proper segmentation. In Row (b), Though the objects (bicyclists) are properly tracked, the background is often added in as part of the sequence. Therefore, some fixes include utilizing box detection and further localizing using time stamp information are implementation.

Fig. 3 shows the result after utilizing the time stamp information for proper bounding boxes. The segmentation correctly tracks objects despite changes to the frames such as rotation. We can see from Fig. 3 that different objects move around the circle with good tracking result.

4. Object Classification

Although the background subtraction algorithm is invariant to rotation, the classifier is not.

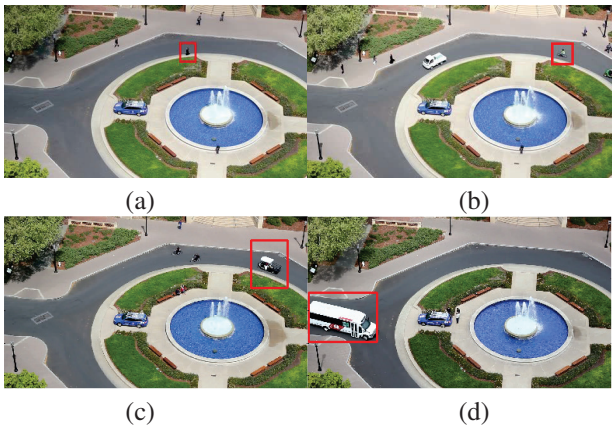

Figure 3. Some examples of bounding boxes results for pedestrain (a), a bicycle rider(b), golf cart(c) and bus(d) after time stamp correction.

We have to generate objects for each angle to precisely train the classifier. In the first section of this study, we generated a training set with bounding boxes assigned by the background subtraction model. Then the rest of the frames are used for the testing set.

Because the training process is time consuming, we don't need to generate all the frames. Instead, we generated one ground truth bounding box for each m frames:

$$m = \frac{t \cdot \text{fps} \cdot \theta}{\text{deg}} \tag{14}$$

, where t is the time for the object to rotate deg angles, $\text{fps} = 30$ is the frame per second, and θ is the desired angle accuracy.

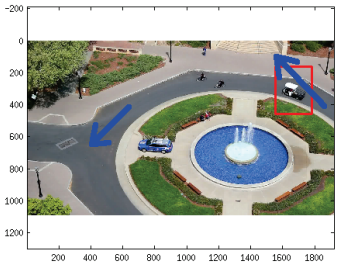

Figure 4. The rotation angle of objects is 90° in the scene.

As show in Fig. 4, $\text{deg} = 90^\circ$. Among all videos, we have the average $\bar{t} = 10s$. Therefore, with a desired degree step of $\theta = 3^\circ$:

$$\begin{aligned} m &= \frac{t \cdot \text{fps} \cdot \theta}{\text{deg}} \\ &= \frac{10 \times 30 \times 3}{90} \\ &= 10(\text{frames}) \end{aligned} \tag{15}$$

4.1. Approach - Discriminatively Trained Parted Based Model

The classification coding was developed based on the project of Felzenszwalb [12, 15, 13]. The model involves linear filters and dense feature maps. The linear filter is used to score the position (x, y). The dense feature map G is an array of d-dimentional feature vectors computed from the dense grid of the locations in an image.

The response or score [14]is defined as:

$$\begin{aligned} &score(p_0, \cdots, p_n) \\ &= \sum_{x',y'} F\left[x', y'\right] \cdot G\left[x + x', y + y'\right] \\ &= \sum_{i=0}^{n} F_i' \cdot \phi(H, p_i) - \\ &\sum_{i=1}^{n} d_i \cdot \phi_d(dx_i, dy_i) + b, \end{aligned} \tag{16}$$

, where $p_i = (x_i, y_i, l_i)$ defines the location and level of the i-th filter. $z = (p_1, \cdots, p_n)$ forms a pyramid of features.

The deformation features are :

$$\phi_d(dx, dy) = \left[dx, dy, dx^2, dy^2\right] \tag{17}$$

, where (dx, dy) is the displacement of i-th part to base position:

$$(dx_i, dy_i) = (x_i, y_i) - (2(x_0, y_0) + v_i) \tag{18}$$

Furthermore the score of a pyramid z is expressed as the product of model parameter β and a vector H as $z = \beta \cdot \psi(H, z)$:

$$\beta = (F_0', \cdots, F_n', d_1, \cdots, d_n, b) \tag{19}$$

$$\psi(H, z)$$
$$= \phi(H, p_0), \phi(H, p_1), \cdots, \phi(H, p_i), \cdots, \phi(H, p_n),$$
$$- \phi_d(dx_1, dy_1), \cdots, -\phi_d(dx_n, dy_n), 1 \tag{20}$$

By using the latent SVM model, we can get the scoring function:

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Psi(x, z) \tag{21}$$

Here β is a vector of model parameters and z are latent values [12]. The binary label for x is obtained by thresholding the score.

β is first trained from labeled samples D by minimizing:

$$L_D(\beta) = \frac{1}{2}||\beta||^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i)) \tag{22}$$

where $\max(0, 1 - y_i f_\beta(x_i))$ is the standard hinge loss and C is the weight of the regularization term.

4.2. Classification Experiment Results

Classification testing was performed both intra-class and inter-class test samples. In practical terms, images were divided into testing sets containing individual classes as well as testing sets containing multiple classes. Because of the intense computational demands required in the classification, the classification tests were performed for both Bus and Cart classes.

For single class tests, each image was given a positive score (a bounding box covering at least 50% of the object was found) or a zero score (no correct bounding box found for the object). For multiple class testing, a third score was added for negative classifications (the incorrect object was classified).

Out of the 13 long movie clips in the bus testing set, the classification results were very promising.

Class	Positive Match	No match
Bus	238 (89.47%)	28 (10.53%)
Cart	181 (42.99%)	240 (57.01%)

Table 1. Single-class Testing Results

Class	Positive Match	No match	Negative Match
Bus	71 (79.78%)	6 (6.74%)	12 (13.48%)
Cart	30 (62.5%)	18 (37.5%)	0 (0.0%)

Table 2. Multi-class Testing Results

Due to the apparent size and ease in grabbing features from the bus's markings, the classifier performed well. Most of the No-Match scores were also well localized but failed to adequately cover the bus area.

On the other hand, cart testing did not perform well, most likely due to the small size of the objects.

Multiclass results for the bus were as expected. The number of false negatives were high due to feature similarity between the bus and cart classes. Tests run without either buses or carts yielded negligible false positives from the cart and bus classes.

Figure 5. A multiclass test image in which the bus classifier chose the cart object

The results of pedestrain and bike are not good due to their smaller size in the image. The segmentation process yielded bounding boxes that were too small or too far localized from the actual objects. For future results, these bounding boxes would have to be manually manipulated (defeating the purpose of semi-supervised learning) or improved with the segmentation code.

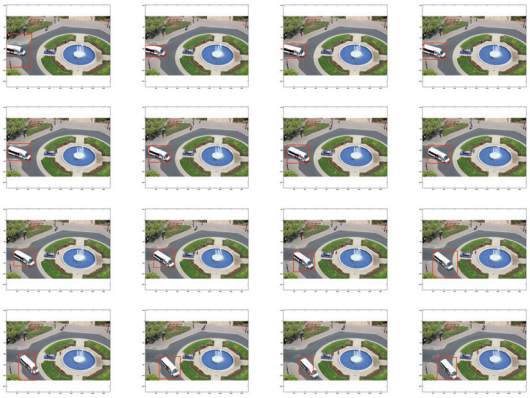

Figure 6. The classification testing result of bus in a movie clip. The pictures are shown for every 5 frames starting from the 4th frame.

Figure 7. The classification testing result of cart in a movie clip. Less than half of the test cases are classified due to the small size of the cart.

Figure 8. Another better classification testing result of cart in a movie clip. In this case, cart is successfully classified from the bus object. The last few frames were lost though due to the closeness between cart and bus.

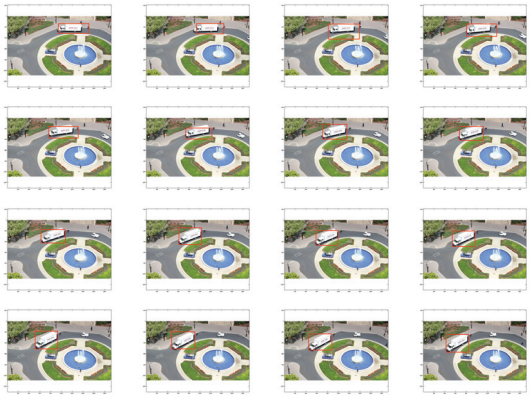

Figure 9. Another classification testing result of multiple classifications. In this case, truck is successfully classified from the cart object and the classification are correct for all the frames. Because this video clips is longer than others (due to the low speed of the truck), the sample is one frame out of each 10 tested frames.

Class	$\bar{w}$ †	$\bar{h}$ †	$\sqrt{\left(\frac{w}{2w_I}\right)^2+\left(\frac{h}{2h_I}\right)^2}$ ‡
Pedestrain	12	17	0.0337
Biker	19	14	0.0325
Cart	34	37	0.0767
Bus	120	90	0.2074

†: $\bar{w}$ and $\bar{h}$ are the average width and height of all the bounding boxes.
‡: w_I and h_I are the width and height of the frame.
Table 3. The bounding boxes actual size and relative size compared with resolution of the video.

4.3. Invariant Discussion

The average bounding box sizes of different object classes out of the 480×272 movie resolution are shown in Table 3.

Therefore, the relative object size over the image size should be $\frac{\text{size}_{\text{obj}}}{\text{size}_{\text{img}}} > 10\%$ and the pixel cluster should have at least 2000 pixels to get good testing result in the classifier. It should be noticed that the image was taken from the top of the Hoover tower which is over 285 feet high. However, when the camera are used in the vehicles, the object will be much bigger and the result will be better.

It is pertinent to note that the classifier was highly variant to rotational transformations of the objects. If the classifier was trained on a set of images of a bus, then even for a 10° real-world

bus rotation, the classifier often did not detect the bus. Therefore, we used $\theta = 3^\circ$ in Equation 15.

Because of the HOG algorithm, the model is invariant to illumination. Here are a few pictures of the scenes in different weather condition. The object of cart and bus can both be classification sucessfully.

Figure 10. Random weather conditions make the movie clips different in illumination. Yet the BSC model can successfully classify objects regardless of illumination changes.

5. Conclusion

The background subtraction/classifier (BSC) model is researched in the report. First a large amount of traffic videos at the Hoover tower were processed. Second, the background of the videos are subtracted by the semi-supervised method using the bi-layer segmentation model. Third, the result of the bacground subtraction was treated using the time stamp information to generate thousands of accurate training smaples automatically. Fourth, the discriminative trained part based (DTPB) model is trained using the latent SVM method . At last, new videos are tested using the DTPB model and the results are analyzed based on their accuracy and performance.

The BSC model is robust on objects larger than 10% of the screen size and is potentially very useful on the vehicle to replace the expensive laser tracking sensor by the cheap CMOS-based cameras.

References

[1] Alex Teichman, Sebastian Thrun. *Tracking-based semi-supervised learning*. Robotics: Science and Systems (RSS), 2011.

[2] Alex Teichman and Sebastian Thrun. *Practical object recognition in autonomous driving and beyond* IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), 2011.

[3] Alex Teichman, Jesse Levinson, and Sebastian Thrun. *Towards 3D object recognition via classification of arbitrary object tracks* International Conference on Robotics and Automation (ICRA), 2011.

[4] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J. Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, Michael Sokolsky, Ganymed Stanek, David Stavens, Alex Teichman, Moritz Werling, and Sebastian Thrun. *Towards fully autonomous driving: systems and algorithms* Intelligent Vehicles Symposium, 2011.

[5] Honglak Lee, Rajat Raina, Alex Teichman, and Andrew Y. Ng. *Exponential family sparse coding with application to self-taught learning* International Joint Conference on Artificial Intelligence (IJCAI), 2009.

[6] Michael Park, Sachin Chitta, Alex Teichman, Mark Yim *Automatic configuration recognition methods in modular robots* International Journal of Robotics Research (IJRR), 2008.

[7] Sheikh, Y.; Shah, M.; *Bayesian object detection in dynamic scenes* CVPR. vol. 1, pp. 74 - 79, 2005.

[8] A. Criminisi, G. Cross, A. Blake and V. Kolmogorov. *Bilayer Segmentation of Live Video.* CVPR, 2006

[9] Y. Wang, P. Perona and C. Fanti. *Foreground-Background Segmentation of Video Sequences.*

[10] http://www.vision.caltech.edu/projects / yiw / FgBgSegmentation/

[11] http://robots.stanford.edu/teichman/neovision2 / overhead_videos.tar

[12] P. Felzenszwalb, R. Girshick, D. McAllester and D. Ramanan. *Object Detection with Discriminatively Trained Part Based Models.* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, No. 9, September 2010

[13] http://www.cs.uchicago.edu/ pff/latent

[14] P. Felzenszwalb, D. McAllester, D. Ramaman. A Discriminatively Trained, Multiscale, Deformable Part Model. Proceedings of the IEEE CVPR 2008.

[15] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan. *Object Detection with Discriminatively Trained Part Based Models.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, iss. 9, pp. 1627 - 1645, 2010.

[16] http://ffmpeg.org/general.html

[17] B.P.L. Lo and S.A. Velastin, *Automatic congestion detection system for underground platforms*, Proc. of 2001 Int. Symp. on Intell. Multimedia, Video and Speech Processing, pp. 158-161, 2000.

[18] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, *Detecting moving objects, ghosts and shadows in video streams*, IEEE Trans. on Patt. Anal. and Machine Intell., vol. 25, no. 10, Oct. 2003, pp. 1337-1342.

[19] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russel, *Towards Robust Automatic Traffic Scene Analysis in Real-Time*, in Proceedings of Intl Conference on Pattern Recognition, 1994, pp. 126131.

[20] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, *Pfinder:Real-time Tracking of the Human Body*, IEEE Trans. on Patt. Anal. and Machine Intell., vol. 19, no. 7, pp. 780-785, 1997.

[21] C. Stauffer, W.E.L. Grimson, *Adaptive background mixture modelsfor real-time tracking*, Proc. of CVPR 1999, pp. 246-252.

[22] C. Stauffer, W.E.L. Grimson, *Learning patterns of activity using real-ime tracking*, IEEE Trans. on Patt. Anal. and Machine Intell., vol. 22, no. 8, pp. 747-757, 2000.

[23] Elgammal, A., Harwood, D., and Davis, L.S., *Non-parametric Model for Background Subtraction*, Proc. of ICCV '99 FRAME-RATE Workshop, 1999.

[24] B. Han, D. Comaniciu, and L. Davis, *Sequential kernel density approximation through mode propagation: applications to background modeling*, Proc. ACCV -Asian Conf. on Computer Vision, 2004.

[25] N. M. Oliver, B. Rosario, and A. P. Pentland, *A Bayesian Computer Vision System for Modeling Human Interactions*, IEEE Trans. on Patt. Anal. and Machine Intell., vol. 22, no. 8, pp. 831-843, 2000.

[26] M. Seki, T. Wada, H. Fujiwara, K. Sumi, *Background detection based on the cooccurrence of image variations*, Proc. of CVPR 2003, vol. 2, pp. 65-72.

[27] J. Lafferty, A. McCallum, and F. Pereira. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. In Proc. 18th International Conf. on Machine Learning, pp. 282289. 2001.

[28] Y. Boykov and M.-P. Jollie. *Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images*. In Proc. Int. Conf. on Computer Vision, 2001.

[29] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Trans. Graph., 23(3):309314, 2004.

[30] Y. Boykov, O. Veksler, and R. Zabih. *Fast approximate energy minimization via graph cuts*. IEEE Trans. Pattern Anal. Mach. Intell., 23(11):12221239, 2001.