2015 IEEE Intelligent Vehicles Symposium (IV)
June 28 - July 1, 2015. COEX, Seoul, Korea

Real-time small obstacle detection on highways using compressive RBM road reconstruction

Clement Creusot[1] and Asim Munawar[1]

Abstract— **Small objects on the road can become hazardous obstacles when driving at high speed. Detecting such obstacles is vital to guaranty the safety of self-driving car users, especially on highways. Such tasks cannot be performed using existing active sensors such as radar or LIDAR due to their limited range and resolution at long distances. In this paper we propose a technique to detect anomalous patches on the road from color images using a Restricted Boltzman Machine neural network specifically trained to reconstruct the appearance of the road. The differences between the observed and reconstructed road patches yield a more relevant segmentation of anomalies than classic image processing techniques. We evaluated our technique on texture-based synthetic datasets as well as on real video footage of anomalous objects on highways.**

I. Introduction

Autonomous driving research has made tremendous progress since the initial DARPA Grand Challenges in 2004, 2005 and 2007. As a matter of fact, most of the investigations in this domain have followed the premises of the DARPA initial challenges: To navigate at moderate speed in very challenging environments such as urban settings [1] or desert mountain trails [2]. The sensors used for perception are usually matched to such tasks and often include radar, LIDAR, or stereo as well as standard wide-angle monocular 2D cameras. While safe autonomous driving at low or medium speeds downtown is challenging, the technology developed for this task may not always suit high-speed driving scenarios. The main reasons for this difference are the sensors' range and accuracy. In short, the sensor range of the vehicle is directly linked to how far in the future it can predict and avoid events such as collisions. At high speed the car must be aware of obstacles at greater distances. For example, for a maximal safe deceleration of 0.8 g, at a speed of 150 km/h, a car would need at least 110 meters to come to a full stop, setting aside the latency for perception and decision processes.

Distant obstacle detection is possible for large objects (such as cars and trucks) using either radar or 2D vehicle appearance detection systems (such as HoG detectors) but it hasn't been done for small unknown and unpredictable obstacles. There are two main reasons for this: First, of-the-shelf active sensors cannot be used with high accuracy over long distances. For example, a LIDAR system such as the Velodyne HDL-64E [3] has a vertical angular resolution of about 0.4°. This means that the maximum distance at which it can detect 3 consecutive points on a small 20-cm vertical object is less than 15 meters. Second, class-specific object detection systems are bound to fail in detecting new types of objects. While some obstacles are more common than others (for example burst tire debris), it is impossible to predict exactly what might fall from a truck or a car onto the road. A class specific machine-learning-based detector works well for cars and pedestrians detection, since the whole class appearance is relatively compact. This will not work for random objects which have no predetermined shape or appearance.

Based on these two observations we decided to investigate the problem from the standpoint of a single class learning task. What we want to do is to detect the dual of the road, i.e. anything on the road that is not the road. In this paper we present a technique for road patch appearance reconstruction using a Autoencoding Neural Network, specifically a compressive Restricted Boltzmann Machine (RBM) trained exclusively to reconstruct the road. We show early evidence that such an approach can be more efficient than classic segmentation techniques for candidate obstacle detection.

Our main contribution is to investigate an uncharted type of technique for anomalous object detection on a learnable textured background. To the best of our knowledge, this is the first time such an idea is being applied to object detection.

In the next sections, we will discuss previous work in this area, introduce our method, and evaluate it on both synthetic and real data. The last section will focus on analyzing its limitations and the advantages of the method, and we will propose interesting future directions.

II. Previous Work

While the literature on pedestrian and car detector is prolific (please refer to [4] and [5] for respective reviews), there has been very little work on unknown anomalous distant object detection on the road. This is quite natural since the focus of autonomous driving has been on Low and Moderate Speed Driving (LSD, MSD) rather than High Speed Driving (HSD). In the LDS and MSD cases, active range sensors are often sufficient for obstacle detection.

Early work in the field of obstacle detection in highway environments heavily relies on stereo vision. J.A. Hancock 1997 [6] makes use of laser reflectance and stereo vision to detect road debris at long distances. T. William et al. [7] uses a multi-baseline stereo technique. The paper claims to detect 14-cm obstacles at a distance of over 100 m. Even the research in recent years uses stereo or Structure-From-Motion (SFM) for solving the problem. H. Kyutoku et al. 2011 [8] compares the previous and the current frame of a video to find any anomalies on the road. Subaru Eyesight TM

[1] Researchers at IBM Research Tokyo, Tokyo Research Lab, Toyosu, Japan {clement,asim}@jp.ibm.com

[9] is one of the commercial stereo-vision-based systems that detects large obstacles robustly. Although, such techniques could in theory detect any obstacle on the road, in practice, these techniques require a very clean road environment with accurate point matching for image warping and disparity computations. This is not practical for point matching since the real world images can be very noisy and the road may have relatively few but repeating features. Also, vehicle vibrations make camera calibration with long focal length very difficult as the two cameras shows independent motion in such situation.

Over the last decade, many researchers have tried to solve this problem using machine learning systems. Mobileye [10] is a commercially available system that detects large obstacles at small distances quite robustly by using only a monocular camera. However, this system can only recognize certain classes of objects such as vehicles or pedestrians. D. Forslund et al. 2014 [11] used far infrared (FIR) to find ROI on night's road scenes and then use boosting approach with sliding widows at multiple image scales to classify patches as animals or not. The technique uses very specific features to train the system and will therefore fail to detect unknown shapes or inanimate cold anomalies on the road.

Another major hurdle in using machine learning techniques to solve our problem is the absence of a comprehensive dataset. Commonly used datasets for vehicle detection [12], pedestrian detection [13] or scene object classification [14] are not sufficient for anomaly detection on the road. Most of the datasets give ground truth bounding boxes for objects of interest on the road. These datasets are meant to learn object features and classify them. In contrast, an anomaly cannot be learned and therefore, the road itself must be analyzed to find the road and non-road patches. For a dataset to be useful for this kind of task we would expect labels for the road surface and other features such as lane markings, cat's eyes reflectors and so on. The short focal length and low resolution of existing datasets also inhibit the situation for HSD.

III. Method

In this section we describe our anomaly detection system in detail. Please note that we focus here on the detection of anomalies, not the evaluation of their threat level or the decision making and actuations required to avoid them. At this stage we do not distinguish between a harmless plastic bag and dangerous solid debris. Our primary objective is to detect small non-road element on the road for further processing.

The inputs to our system are simple 2D color images of a road. Our approach has two main stages. In the first stage we try to generate a heat map that represent the likelihood of a patch being part of the road. For visualization we represent the heat map as a normalized grayscale image from white (road pixels) to black (non-road pixels). In the second stage, we use this map to perform a segmentation of potential obstacles. Our main focus is on stage 1 in this paper.

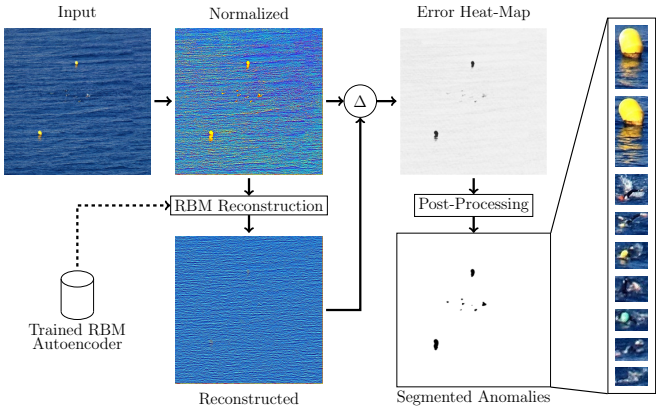


Fig. 1. Workflow of our anomaly detector. Please note that our technique can be used for many visual anomaly detection task on learnable textured background. Here we illustrate with an anomaly detection experiment on sea images. (Please zoom in for more details. The pictures in the second column are white balanced for display only.)

For the reconstruction task, several machine learning techniques could be used to learn road appearances. In this paper we use an RBM as an autoencoder for the input image layer. We explain the reason for this choice later in the paper. At this stage our goal is not to recommend a particular learning technique but to assess whether the global approach of detection by reconstruction make sense for practical applications and whether the road texture data can be learned.

A. Preprocessing

Our approach has an offline training part and an online part. Both are preceded by a pre-processing stage where the large input image is cut into small square patches of dimension $P \times P$ which are converted to float vectors, mean centered and normalized. In most of the experiments in this paper we use $P = 8$ with a stride $S = 6$. The sampling parameters have some effects on the smoothness of the resulting map as well as on the speed at which the system can run. Refer to section IV-E for further details.

B. Offline Training

For the training stage we prepared patches cut from a long YouTube video of a Japanese highway ([15]). We manually defined a selection mask for the video so that only patches near the center of the road area were selected for training (see Figure 2) This data was fed to a Gaussian binary Restricted Boltzmann Machine (RBM) with an input layer of size $L_{vis} = P \times P \times 3$ and hidden layers of size H. We set H to be relatively small for a compressed representation of the input. In our experiments, H is set to 20. The RBM is trained using Stochastic Gradient Descent. The cost function is the mean squared reconstruction error of the patch given random Gaussian corruption of the input as in [16]. The trained weights of the RBM are stored for use in the online part:

$$RBM_{model} = (W, b_{hid}, b_{vis}) \quad (1)$$

where W is the unit weights matrix and b_{hid} and b_{vis} are the hidden and visible biases vectors respectively.

Lets call x'_i the reconstruction vector:

$$x'_i = Sigmoid(x_i \cdot W + b_{hid}) \cdot W^T + b_{vis} \quad (2)$$

The objective of the training is determine (W, b_{hid}, b_{vis}) that satisfy:

$$\underset{W,b_{hid},b_{vis}}{\arg\min} \sum_i (x'_i - x_i)^2 \quad (3)$$

Here is our basic idea. If the RBM has been trained on a single positive class of patches $x_i \in A$, then the error should be smaller for patches belonging to that class than for random patches $\bar{x}_i \in \neg A$, i.e.:

$$\epsilon_{x_i} < \epsilon_{\bar{x}_i} \quad (4)$$

where ϵ is the reconstruction error. By looking at these values we can identify which patches were the least expected, i.e. the most anomalous.

The training is imperfect and presents many clutters. However these elements are a minority and since the system cannot learn much (due to its compressive nature) it can only learn things that are quite generic to the whole set (ideally a model of what a road patch is).

We choose an RBM model over other modern machine learning approaches because of its simplicity, popularity, ease of implementation but mainly because the online part can be executed quite fast without any hardware acceleration, making it a good candidate for real-time applications.

It is important to understand that having a too adaptive reconstruction method here would defeat our purpose. An adaptive system would be able to generalize well to new types of data. Hence, we intentionally want the system to be limited enough not to reconstruct anything besides road patches.

C. Online pipeline

In the online part the process follows the workflow presented in Figure 1. For visualization, we reconstructed the image from the computed patches at each stage of the process. Example of such pictures for road images are shown in Figure 6.

The previously unseen image is preprocessed into normalized patches that are row vectorized. Each patch x_i is autoencoded into the reconstructed patch x'_i (Equation 2).The error heat map is computed by taking the absolute differences between the input and reconstructed images.

$$\Delta_{x_i} = |x'_i - x_i| \quad (5)$$

The heat map based on these pixel differences can be used directly in the next stage. However reconstructing the full image from patches is computationally expensive. A faster solution is to consider only the mean of the pixel errors in each patch and remap these values to a two-dimensional image:

$$\epsilon_{x_i} = \sum_{0 \le k < L_{vis}} \Delta_{x_i,k} \quad (6)$$

This introduces a size reduction of the heat map similar to using a uniform convolution of size $P \times P$. In practice, the blurring effect introduced by this down-sampling does not significantly impact the detection of anomalies, since the target objects are larger than the patch size.

The post-processing steps are performed on the inverted reconstructed heat map (not the patches). The inversion is mainly used to help the visualization so that the obstacles appear black on a white background. First, we remap the intensity level of $[0, mean]$ to $[0, 1]$ so as to saturate as white the areas with insignificant reconstruction errors. The second step is to get a final segmentation from the heat map. Many techniques can be used for such task whether pixel-wise or context based. To avoid misunderstandings we do not use the final segmentation step in this paper and focus on the evaluation of our main contribution: the reconstruction error heat map.

For our visualization purposes the error Δ is shown as an average image and remapped to a 0-255 range. This is an imperfect representation of the underlying data. In addition, we usually show the heat map for the entire image. In practice the reconstruction would only be generated for the known road regions to save computational resources.

D. Training Data

Training data is an essential part of any machine learning approach. In order to learn the road appearance we used an online YouTube video [15]. It consists of a 1 h 40 m sequence of highway in Japan recorded from a car dashboard with a Panasonic GH4 camera at 4K resolution (3840×2160). We first temporally down-sampled the video to keep only one frame every 2 seconds (3054 frames) and resize by a factor of 0.5. We then selected a fix mask of 500×500 in the center road area. Each selected frame region is decomposed in squared 50×50 images with a stride of 25 producing 144 samples per frame. In the training phase these samples are randomly sampled for smaller patches of size $P \times P$ given the parameters of the experiment.

180.000 × 3840 × 2160 3054 × 500 × 500 439776 × 50 × 50

Fig. 2. The training samples are extracted from the video [15].

IV. Evaluation

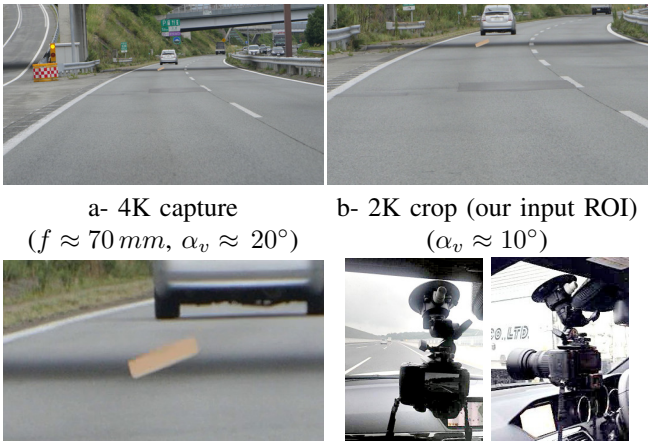
To evaluate our approach we first examine the quality of the reconstruction error heat map for classification tasks on small image patches. We then look at real video examples presenting anomalies to get a qualitative measure of the strength and limitations of our system.

A. Testing Data

For the testing data we use two different sources. First, some high resolution videos found on the Internet showing

obstacles on highways [17][18]. Second, video we recorded ourselves in conditions similar to [15] (4K resolution, Japanese highways). However while [15] used a relatively wide angle 12-35 mm lens, we used lenses between 70 mm and 150 mm. We used high resolution with powerful zoom to be able to detect distant objects on the road. The video capture system is shown in Figure 3. Unfortunately most of the data did not contain any significant anomalies. In 4 hours and 33 minutes of recording we encountered only one instance of a noticeable anomalous object for around 5 second as seen in the supplemental video and Figure 3-c. To compensate for the lack of test data, we downloaded high resolution video from YouTube such as [17], [19] and [18].

We also created synthetic data by alpha mating random object images on top of highway background patches (see Figure 4).



a- 4K capture ($f \approx 70\,mm$, $\alpha_v \approx 20°$)

b- 2K crop (our input ROI) ($\alpha_v \approx 10°$)

c- Zooming in on an anomaly ($\alpha_v \approx 2°$)

d- Data Acquisition Panasonic DMC-GH4

Fig. 3. Details of a single frame presenting an anomaly (a,b,c) and recording setup (d)

B. Quantitative Analysis

Our main interest was to see how well the compressive RBM approach can reconstruct the road while not reconstructing non-road patches. Because there are currently no directly comparable state-of-art methods to compare with, we evaluated the main output of our method with RGB input data using standard classification techniques such as Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM). For each experiment, we used exactly the same baseline techniques with different input vectors computed from the same corresponding data. By doing this we were able to measure whether the distribution of our output is a better indicator of the labels (road vs non-road) than the raw input images. To measure this we use small image patches following the CIFAR layout (32x32 RGB patches). The positive class is composed of images containing plain roads (ROAD-RGB) sampled from manually labelled images. One negative class is composed of the actual CIFAR10 public dataset as a source of random non-road elements (CIFAR-RGB). Another negative class is composed of road patches (independent of the training) on which synthetic objects have been alpha-mated (OBS-RGB) (see Figure 4).

For these three datasets we computed the respective datasets of Δ error vectors from Equation 5 using our trained RBM: ROAD-Δ, CIFAR-Δ, and OBS-Δ. All of the datasets are split into a training and testing part of sizes 2.000 and 2.000. Each experiment involved two datasets (one positive and one negative class), we therefore used 4.000 samples for training and 4.000 samples for testing.

For both of our experiments we compared classification results using a linear SVM and an LDA technique. In Experiment 1 we compared the results obtained for ROAD vs. CIFAR. In Experiment 2 we compared the results obtained for ROAD vs. OBS. We show the classification results as standard Receiver Operating Characteristic (ROC) curves in Figure 5. Using the output of our system gives consistently better results than using the standard RGB image representation. The Area Under Curve (ROC-AUC) metric is always larger for our inputs. This means that the labels road/non-road are more linearly separable from the Δ error map than from the normal input images.

Please note that while x'_i can be encoded using only 20 values (the size of the hidden layer), $\Delta = |x'_i - x_i|$ also depends on the input and doesn't have a straightforward compressed representation. It is therefore difficult to measure the compression ratio of the overall representation relative to an RGB image.

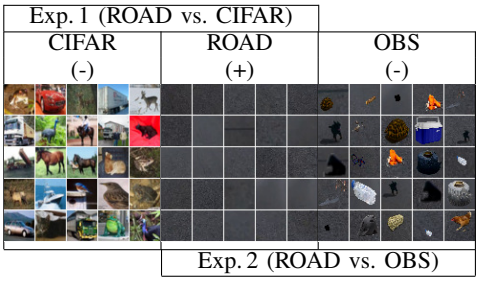
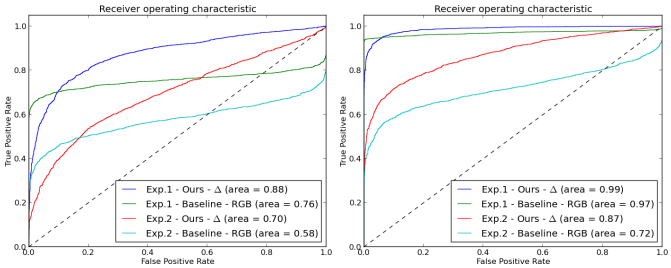


Fig. 4. Dataset patch RGB examples for the two quantitative experiments.



LDA Results SVM Results

Fig. 5. ROC curves for experiment 1 and 2. Left graph shows the LDA results while right graph show SVM results.

C. Qualitative Results

We tested our system on full size videos not related to the training data. The lighting, roads, recording settings, camera positions, resolutions, and so on were different. This is a very challenging test for any system.

What we want to demonstrate here is that the compressed representation of the road texture learned by the system is generic enough to be used for new and previously unseen roads. You can see examples of such detection in Figures 6 and 7.

Empirical observations of the reconstruction experiments on real images lead us to several conclusions. The system learns to some extent the color of the road. Colors that are unlikely to be seen on the road yield higher errors in general. This is independent of the amount of blur. As a rule, uniform patches yield smaller errors, while highy textured images produce larger errors. Edges generate the largest errors. However, different type of edges are not treated equally. The edges of shadows often seen on the road have smaller reconstruction errors than the edges of objects, as seen in Figure 7. A human-designed system would have to incorporate different rules for different cases, which seems unlikely to work in the long run in real life scenarios. Our data-driven approach to road reconstruction, although preliminary, appears much more robust by nature than human-designed approaches for anomaly detections.

D. Limitations

One obvious limitation of our approach is that by not having a negative training class the boundaries of the positive class remain stuck to the outer shell of the road samples in the feature space. In other words, the system can only generalize unseen patches that lie within the training manifold. This limitation is clear in figure 8. This scene from a YouTube video was filmed at dusk and the whole scene is immersed in the distinctive reddish lighting of dusk. Such lighting did not appear in the training video. Therefore all those slightly redder road patches are discarded as being non-road patches. However the road patches that lies in the shadows of the car and the bridge do not reflect this red light and are correctly reconstructed.

To solve this kind of problem our system should learn all of the possible road appearances. This works well for highways that have clean and uniform appearances but might be more difficult on messier road data. A complementary approach would be to use the within frame (or within video segment) road appearance to detect the anomalies. Indeed, while the roads appearance might vary widely, its texture remains locally self-similar within a frame or a short sequence of frames.

Fig. 8. Example of failure on video [19] due to an insufficient training coverage. Dusk lighting has never been seen during training. All road patches under the reddish sunlight are misclassified while road patches in the shadows are properly reconstructed.

TABLE I
COMPUTATION TIME PER MODULE (IN SECOND PER FRAME)

	Full images				Masked images	
Resolution	VGA 600x480	WSGA 1024x600	HD 1360x768	FHD 1920x1080	HD 1360x768	FHD 1920x1080
#Patch	7821	16830	28702	57101	5325	10829
Patch Extraction	.044	.096	.165	.350	.031	.063
Normalization	.031	.067	.132	.264	.021	.045
Reconstruction	.059	.129	.229	.456	.040	.082
Error Map	.014	.030	.063	.124	.009	.018
Total	.149	.323	.591	1.195	.103	.210

Another limitation concerns uniform non-textured surfaces. These surfaces are so simple that they can be easily reconstructed by the system even if they have never been seen before. This leads to low reconstruction errors in flat uniform areas, for example in the sky.

E. Speed

In our online process pipeline, the most computationally expensive parts are to cut patches within the input image and run the RBM reconstruction. For a fixed resolution, the speed of our system is more or less linear with the number of patches used per image. A higher number of patches (increased overlap) leads to a smoother error heat map but slower computation.

Our RBM was trained using the pylearn2 Framework [20] in python. The computation time for each of our modules is given in Table I. Time are given for a single Intel i7 CPU core at 3.33 GHz, the values are given in second per image and are averaged over 300 images. In the two last columns, we assume that the system knows the road position within the image. In that case we use the binary mask of the road to restrict the sampling of patches. This mask cover 20% of the images in average. With mask selection and without any harware acceleration, the system runs at almost 10 fps for a HD resolution input.

V. CONCLUSION

We have presented a new approach for obstacle detection based on reconstruction of road patches using a compressive RBM. While still at a early stage, we show that the approach has great potential to detect anomalous artifacts on the road. It is also more elegant than many other techniques, since it is driven by the data rather than by man-made sequential recipes.

While anomaly detection is required for obstacle detection it is not always sufficient. Being able to distinguish between a flat anomaly that represents no danger (such as a piece of cardboard Fig 7) and a 3D obstacle that can potentially be dangerous (such as a burst tyre Fig 6) is very important to avoid false positive alerts and their potential consequences (autonomous braking and lane changes). The solidity and degree of hazard of an anomaly is difficult to evaluate with a single 2D image. Future work will investigate the use of stereo vision (in particular uncalibrated vertical stereo) as well as video-based 3D inferencing to estimate the volume and configuration of the anomaly on the road. In terms

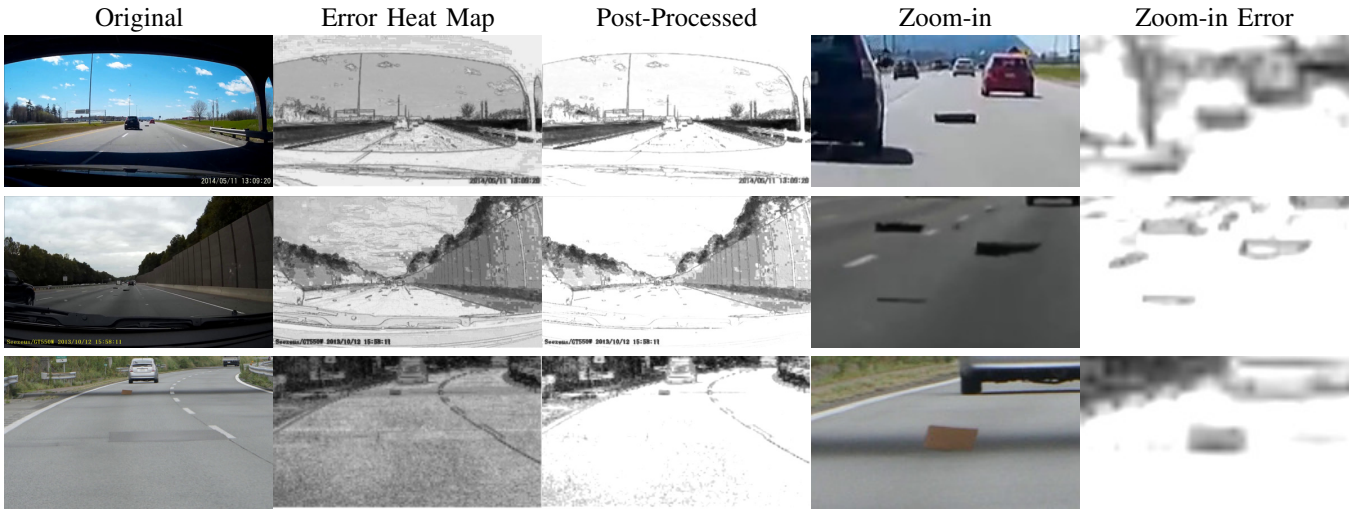


Fig. 6. Example on three real anomaly videos.

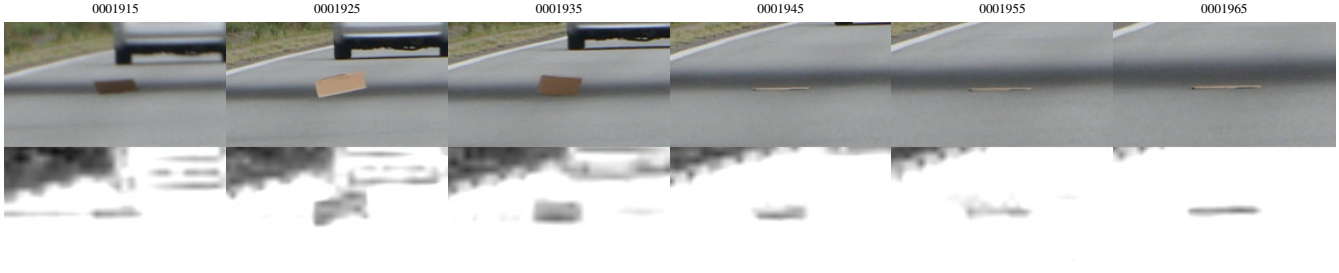


Fig. 7. Example of a detection sequence on a flying piece of cardboard sampled every 10 frames. Zoomed-in for visualization only. Notice that the part under the bridge shadow is properly reconstructed as being the road.

of texture reconstruction, it appears that patches within the same image or temporal sequence are highly correlated. Using such information might further increase the detection performance of our system.

Our final word concerns the data. An ideal situation would be to compare methods without resorting to synthetic data. However obstacles on the road in non-controlled environments are both rare and dangerous. Because they are rare it is difficult to gather enough high-resolution data to construct a large test dataset. Because they are dangerous we can not easily recreate such events on real highways. We think that large data collections of highway footage and the creation of labeled datasets of anomalies will be a good way to stimulate further research in this area.

References

[1] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, 1st ed. Springer, 2009.

[2] ——, *The 2005 DARPA Grand Challenge: The Great Robot Race*, 1st ed. Springer, 2007.

[3] Velodyne lidar. http://velodynelidar.com/lidar/.

[4] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *PAMI*, 2012.

[5] S. Sivaraman and M. Trivedi, "A review of recent developments in vision-based vehicle detection," in *Intelligent Vehicles Symposium (IV)*, 2013.

[6] J. Hancock, "High-speed obstacle detection for automated highway applications," CMU, Tech. Rep., 1997.

[7] T. Williamson and C. Thorpe, "Detection of small obstacles at long range using multibaseline stereo," in *IEEE International Conference on Intelligent Vehicles*, 1998.

[8] H. Kyutoku, D. Deguchi, T. Takahashi, Y. Mekada, I. Ide, and H. Murase, "On-road obstacle detection by comparing present and past in-vehicle camera images," in *MVA*, 2011.

[9] Subaru eyesight: Driver assist technology. http://www.subaru.com/engineering/eyesight.html.

[10] D. B. Yoffie, "Mobileye: The future of driverless cars," Harvard Business School Case 715-421, October 2014.

[11] D. Forslund and J. Bjarkefur, "Night vision animal detection," in *Intelligent Vehicles Symposium Proceedings*, 2014.

[12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[13] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *CVPR*, 1999.

[14] Newovision2 dataset. http://ilab.usc.edu/neo2/dataset/neovision2-dataset-contents.pdf.

[15] Wataken777. (2014) Youtube - "tokyo express way gh4 4k. https://www.youtube.com/watch?v=UQgj3zkh8zk. (3840 x 2160).

[16] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Comput.*, vol. 23, no. 7, pp. 1661–1674, July 2011.

[17] B. Xia. (2013) Youtube - "dashcam test tire debris". https://www.youtube.com/watch?v=7Ebj2OXI3I0. (1920 x 1080).

[18] _djel. (2014) Youtube - "dash cam - avoiding debris on highway". https://www.youtube.com/watch?v=Hw6Ck_BTKEs&. (1920 x 1080).

[19] P. Rustchynsky. (2013) Youtube - "dashcam - hitting debris on the motorway". https://www.youtube.com/watch?v=bIm-ffb-SKs. (1920 x 1080).

[20] I. J. Goodfellow, D. Warde-Farley, P. Lamblin, V. Dumoulin, M. Mirza, R. Pascanu, J. Bergstra, F. Bastien, and Y. Bengio, "Pylearn2: a machine learning research library," *arXiv:1308.4214*, 2013.