EFFICIENT ROAD MAPPING VIA INTERACTIVE IMAGE SEGMENTATION

O. Barinova, R. Shapovalov, S. Sudakov, A. Velizhev, A. Konushin

Moscow State University, Dept. of Computational Mathematics and Cybernetics
{obarinova, shapovalov, ssudakov, avelizhev, ktosh}@graphics.cs.msu.ru

Commission III, WG III/5

KEY WORDS: Automation, Video, Processing, Incremental, Learning, Object, Detection

ABSTRACT:

Last years witnessed the growth of demand for road monitoring systems based on image or video analysis. These systems usually consist of a survey vehicle equipped with photo and video cameras, laser scanners and other instruments. Sensors mounted on the van collect different types of data while the vehicle goes along the road. Recorded video can be geographically referenced with the help of global positioning systems. Road monitoring systems require special software for data processing. This paper addresses the problem of video analysis automation, and particularly the pavement monitoring functionality of such mobile laboratories. We show that computer vision methods applied to this problem help to reduce amount of manual labour during data analysis. Our method transforms video collected by mobile laboratory into rectified geo-referenced images of road pavement surface, and allows mapping of lane marking and road pavement defects with minimum user interaction. In our work the mapping workflow consists of two stages: off-line and online stage. In order to reduce user effort during error correction we take advantage of hierarchical image segmentation, which helps to delete false detections or mark missing objects with just a few clicks. Through continuous training of detection algorithm with the help of operator input error rate of automatic detection decreases; thus minimal input is required for accurate mapping. Experiments on real-world road data show effectiveness of our approach.

1. INTRODUCTION

Roadway monitoring systems are widely-used for supervising road pavement surface and repair planning. These systems usually include a complex of video cameras and other sensors mounted on a car as shown on Figure 1. The sensors record road pavement surface when travelling on a pavement at traffic speed.

Most existing software for road monitoring involves manual processing of video collected by these mobile laboratories. Operator manually marks objects like lane marking and pavement surface defects (potholes, cracking and patches) on each video frame. This procedure is laborious and takes plenty of time; therefore the task of automation of objects detection comes into focus. In this paper we consider the problem of automation of video analysis for pavement surface monitoring. We describe a tool which assists in utilising visual observation data of pavement surface and mapping lane marking and pavement surface defects.

Our main goal is to minimize effort of operator at the time of mapping lane marking and road defects while preserving accuracy of mapping result. The effectiveness of our method is achieved by intensive usage of computer vision techniques together with user-friendly interface that allows checking results of automatic detection and correcting errors if needed.

As long as direct mapping of lane marking and road pavement defects in video sequences faces severe difficulties, we transform video into rectified images of road pavement surface. These images are further processed during interactive mapping.

While to our knowledge there haven't been much research on topic of road defects detection, lane detection is a well-researched area of computer vision with applications in autonomous vehicles and driver support systems. Despite perceived simplicity of finding white markings on a dark road, it can be very difficult to determine lane markings on various types of road. These difficulties arise from shadows, changes in the road surfaces itself, and differing types of lane markings. A lane detection system must be able to pick out all manner of markings from cluttered roadways and filter them to produce a reliable estimate of the vehicle position and trajectory relative to the lane as well as the parameters of the lane itself such as its curvature and width.

Existing methods for lane marking detection are usually based on edge detection (McDonald, 2001) and gradient analysis (Lu, 2007). Use of edges makes detection results sensitive to noise, changes in lighting conditions and shadows. Another approach uses steerable filters (McCall, 2004) which can be convolved with the input image and provide features that allow them to be used to detect both dots and solid lines while providing robustness to cluttering and lighting changes.

As long as these methods were designed for autonomous vehicles, they aim at tracking of lane marking in video. In our work the goal is to detect lane marking in still images of road surface. Also our task is to detect precise contours of lane marking instead of just determining lane marking direction. This task is closely related to the field of semantic image segmentation, therefore the method we propose for detection is based on semantic segmentation of rectified road images.

Rectified images can differ substantially depending of roadway material, time of survey and weather conditions. Therefore automatic detection tuned on one road image can perform poorly on other images. For this reason we have developed a detection algorithm which is automatically tuned with the aid of user interaction in order to perform best on each particular road. This allows accounting for specific characteristics of every particular road, or even a road section.

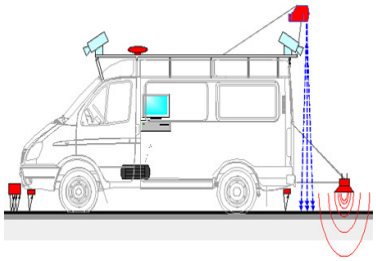
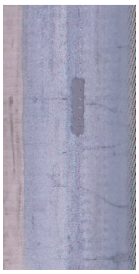


Figure 1. Road laboratory. Video cameras are mounted on the front side and on the back side of the car.

(a)



(b)

Figure 2. Video frame from one camera and a corresponding section of rectified road image.

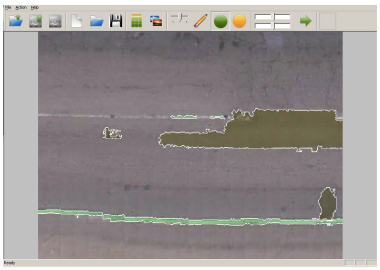


Figure 3. Lane marking and road defects are mapped with a minor user interaction.

The outline of mapping process for a user if the following: first automatic detection is applied to a small section of rectified road image, after that user checks the results of automatic method, corrects the errors if needed and then detection algorithm is adapted in order to take new data into account. After that user goes on to the following road section and the whole procedure is repeated again. Through continuous training of detection algorithm with the help of operator input error rate of automatic detection decreases; thus minimal input is required for accurate mapping. In order to reduce user effort during error correction we take advantage of hierarchical image segmentation, which helps to remove false detections or mark missing objects with just a few clicks.

The paper is organized as follows. Section 2 addresses the procedure of data acquisition and transformation of video sequences into rectified road images. Section 3 describes offline stage of our method. Section 4 gives details on user interaction with the system. Our method for lane marking and pavement surface defects detection is described in section 5. Section 6 is devoted to our machine learning algorithm, which helps to tune detectors on various road images individually. Experiments on real-world data collected by our mobile laboratory are described in section 7. Section 8 is left for conclusion and future work.

2. DATA ACQUISITION

In this work we have used a vehicle equipped with 4 video cameras with resolution 720x576px and Global Positioning System (GPS) on board. The cameras capture video of road surface and roadside, which can be accurately geographically registered by means of GPS. Figure 2 (a) shows an example of one frame of video obtained by a video camera mounted on a van and corresponding section of rectified road image.

Although all cameras in capture video, usage of video as input for mapping lane marking and road pavement defects has severe drawbacks. First, areal objects on road pavement surface suffer from projective distortion which degrades performance of detection algorithms. For example, rectangular pavement patches become trapezoids in video frame. Second, some elongated objects are not fully visible in any single frame of video sequence. Third, different objects are represented with different spatial resolution on the same video frame depending on their distance to the camera. To overcome these problems we transform video sequence into rectified image of the road pavement surface.

These images are obtained from video using perspective plane transformation. Resulting image is one long image in the full driven length. All rectified images are stored in raw format with time and distance information of all pixels. Figure 2 (b) shows an example of video frame obtained from one camera and a corresponding section of rectified image of road pavement surface.

As long as image processing algorithms (like image segmentation) used at subsequent stages of our workflow are memory and time consuming, long rectified road image are cut into non-overlapping small sections. Each part is about 0.5 megapixel image and represents an approximately 5-10 meters long section of road pavement surface. All these section images are further processed in chain, following vehicle path.

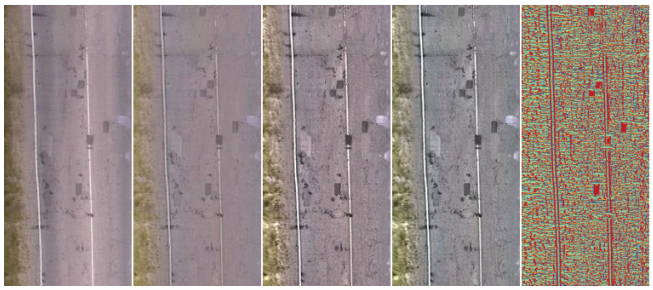
3. OFFLINE STAGE OF MAPPING PROCESS

In our work the mapping workflow consists of two stages: off-line and online stage. As long as we aim at interactive working time at the time of road mapping, all time-consuming operations required by both detection and learning are performed off-line. Offline stage happens once for each road data before user starts mapping road surface. This stage doesn't require any user assistance. Our detection algorithm is based on over-segmentation and classification of super-pixels, therefore offline stage includes image processing, image segmentation, and calculation of features for each image segment. Below these operations are described in more details.

Image processing

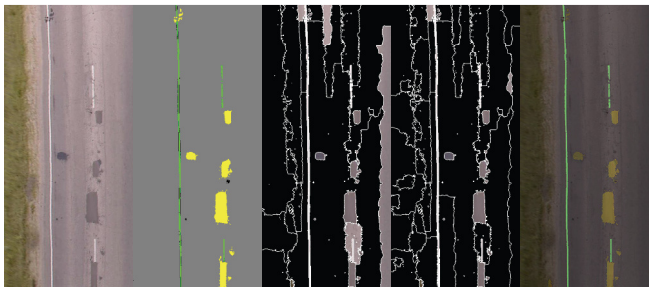
Roadway images are strongly differed to each other in color, brightness and texture. This fact substantially complicates the detection task. Therefore main goal of image preprocessing is to normalize images and put them into some standard state. Image processing includes luminance correction, contrast adjustment, colour correction and image smoothing. All these operations are performed in CIE-Lab colour space.

For luminance correction we use a modification of Retinex algorithm(Land, 1971) . Single-Scale Retinex has artifacts such as halos around dark objects and shadows around light ones, what damages detection in low-contrast images. Conventional Multi-Scale Retinex also has these artifacts when it has to deal with strong luminance changes. Since most of necessary lightness correction is caused by ruts on the road, brightness map is calculated using elongated median filter. It helps to reduce halos effect during luminance corrections (Figure 4 (b)).



(a) (b) (c) (d) (e)

Figure 4. Image processing stages: (a) Source image; (b) Retinex transformation result; (c) result of retinex and contrast adjustment stages; (d) overall image preprocessing result; (e) Texton map



(a) (b) (c) (d) (e)

Figure 5.Cascade classification stages. (a) – Input image, (b) – ground truth image, (c) – 1st cascade layer result, (d) – 2nd cascade layer result, (e) – overall algorithm result.

After Retinex correction mean value of luminance becomes equal to one. Before scaling L-component back to normal, contrast adjustment is achieved by squaring it. Then, L-channel is scaled back to normal (Figure 43(c)). This operation helps to make detection of road defects easier even in low-contrast images.

Colour correction uses conventional grey world algorithm. In Lab colour space it consists of the shift of colour components so as to make mean value of these components be equal to zero, instead of scaling colour components in RGB space. In the final stage bilateral filtration is used to smooth image without loss of important details (Figure 4(d)).

Image segmentation

The hierarchical structures is a powerful tool to analyze data in many applications. Several basic approaches to construction of such multi-level image structure exist. The first approach involves recursive segmentation. An image is segmented in a large scale, and then segments are independently split into pieces. Another approach involves successive segmentation of an image at several scales. But in this case large segments not necessarily represent combinations of smaller ones; this fact limits the scope of application of this method for segmentation.

In this work we used a method based on determination of strength of the boundaries between segments by means of the analysis of saddle points between density modes and merging segments that are weakly separated. For segmentation of the image in our work the hierarchical version of algorithm of mean shift, proposed in (Paris, 2007) is used.

This algorithm provides fast hierarchical segmentation on the basis of idea of the saddle point analysis. Results of this hierarchical segmentation are shown in Figure 5, where borders of segments at different levels of hierarchy are shown in white.

Features calculation

A number of various features are used for classification of segments. We use colour statistics, such as mean values of CIE Lab components and mean values of RGB components, colour variance, Lab components' percentiles.

To account for shape information we calculate coordinate statistics, such as mass centre, coordinate variance, elongation, orientation, area of the segment. Usage of information about neighbourhood of the segment is also very informative for road defects detection. Accordingly distance between mean values of colour components inside segment and inside its neighbourhood are also included in the list of features.

Texton histograms are also used in our system (Leung 1999). These features are proven to be highly effective in recognition task and are used nowadays in many detection and recognition systems (Criminisi, 2006).

Previously created filter bank is applied to the image; filter output vectors for each pixel are associated with the nearest texton vectors from previously trained universal texton dictionary. Then histogram of textons over the segment is used as feature for classification task. Figure 4(e) illustrates a resulting texton map, which is an image, where pixels are labeled accordingly to corresponding textons.

4. ONLINE STAGE OF MAPPING PROCESS

At online stage automatic detection algorithm is applied to parts of rectified road image. User examines results of automatic detection on one image part and corrects detection errors if needed. Then automatic detection is adapted to new data. After that user goes on to the next part of the road and again analyses and corrects results of automatic detection. Accordingly automatic detector is continuously tuned in order to capture specifics of particular road.

Our system provides various facilities for making process of error correction easier for the user. The GUI contains a control which lets user change segmentation level. Operator is able to mark ground truth in a less detailed level and then specify it in a more detailed one. It makes user work more efficient.

Another facility allows controlling tradeoff between detection rate and false positive rate individually for lane marking and road defects. For example, user can increase detection rate of road defects detection (thus increasing false positive rate) by moving a slider. The change in detection rate is performed by changing a threshold on classifier output for road defects on the last cascade layer. This feature helps to significantly reduce amount of manual work in the beginning of online stage, when classifiers show instable performance.

5. LANE MARKING AND PAVEMENT DEFECTS DETECTION ALGORITHM

Our approach is based on cascade classifiers. The idea of cascades is derived from (Viola, 2002). General workflow of cascades is the following. There is ordered set of classifiers, where every subsequent classifier is more "complex" than the preceding one ("complexity" of the classifiers is defined depending on specifics of data or application). Input data

array is passed through these classifiers in turn; each classifier eliminates the data that confidently does not belong to the target class, the remained data is passed to the following, more "complex" classifier, for more thorough examination, etc.
The general idea of cascades involves detection of one target class that implies binary classification. In our task the cascade is applied to a problem of separating objects of two different classes from a background; that requires three-class classification. It is important to notice, that the background class in our task dominates significantly over classes of lane marking and pavement defects. This finding suggests modifying the scheme of cascades used in (Sudakov, 2008) in order to allow detection of several classes of objects.

Cascade workflow

At the offline stage the image had been segmented using the method from (Paris, 2007) into homogeneous regions, and several scales of segmentation are available. The largest scale segmentation is used on the first layer of cascade, the most detailed segmentation scale is used on the last layer. Segmentation at each subsequent scale is a subdivision of segmentation at the preceding scale, therefore we have a sequence of enclosed segments (hierarchy). Each cascade layer corresponds to a certain scale of hierarchy of segmentation and a binary classifier.
Those segments that have not been rejected at the preceding layers of cascade are classified into two classes: objects of interest (including lane marking and road surface defects) and background. The goal of classification is to reject the segments that do not contain pixels of objects of interest. For this purpose the threshold on the classifier output is set up so that the detection rate is close to 100 %.
This procedure is repeated up to the last cascade layer and then multi-class classification is applied. Segmentation corresponding to the last cascade layer is detailed enough to capture precise bounds of lane marking and pavement defects. Moreover, the majority of background segments are rejected at the preceding layers, so the number of background segments passed to the last layer approximately equals to the number of lane marking segments and segments of road covering defects. Therefore our cascade operational scheme also helps to solve a problem of imbalanced classes thus helping to achieve better classification performance. The workflow of cascaded segmentation is illustrated in Figure 5.

6. ON-LINE LEARNING

Online learning algorithms (Domingos, 2000, Oza,2005) process each training example once ‘on arrival’ without the need for storage and reprocessing, and maintain a current model that reflects all the training examples seen so far. Such algorithms have advantages over typical batch algorithms in situations where data arrive continuously. They are also useful with very large data sets on secondary storage, for which the multiple passes through the training set required by most batch algorithms are prohibitively expensive.
In order to enable user-aided tuning of object detection we incorporated on-line learning algorithm in the core of the system. As long as we aim at interactive time of classification and learning, the following requirements for the online-learning algorithm arise. First, online classifier should not store previously seen training examples. Second, learning time should not depend on the number of examples already seen by the learner. Thus we chose online random forest over Hoeffding trees (Domingos, 2000) as it meets both these requirements. Below we describe how online classifiers are used in our cascaded detection method.

On-line learning of cascaded segmentation

In section 3 we described the workflow of cascaded algorithm for object detection, supposing that all classifiers are already trained. Here we describe the training phase of cascaded detection method.
The main problem here is what data should be used for training of classifier at each particular layer of cascade. There are two difficulties with providing training data to classifiers at cascade layers. First, we should take into account all segments which contain target class because if we do not provide enough samples of target class at the training stage, classifier wouldn’t be able to detect them at classification stage. This can lead to severe error of first kind.
The second problem is lack of target samples at all cascade layers in comparison to number of background samples. This class imbalance can lead to additional increase of error of first kind. This means that cascade will miss large amount of target objects. Therefore we need to consider special techniques for balancing class distributions. Our solution for both these problems is the following. We train classifier corresponding to each cascade layer using the data passed to a corresponding cascade layer by preceding version of cascade which had not been adapted to last portion of data. Then, all segments which contain marking and defects are added to the training set on each layer. In order to better balance classes’ distribution we use cost-sensitive online random forest, described below.

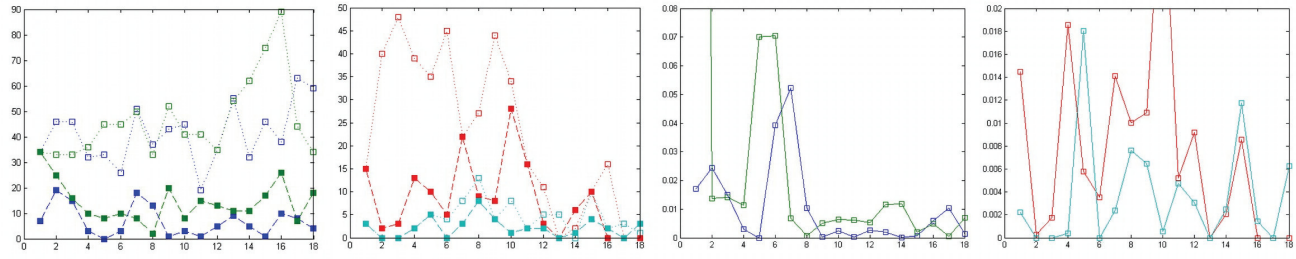
On-line random forest

In this work we use ‘one vs all’ algorithm for multi-class classification on the last cascade layer. This enables using binary classifiers on the lowest tier of the system. Those classifiers should be able to learn even some first portions of a training set efficiently to give a reliable classification result. Also, as mentioned above, these classifiers should handle imbalanced classes’ data.
We use on-line random forest classifiers at all stages of cascade. Our version of online random forest resembles an online bagging algorithm proposed by (Oza, 2005). We modified this algorithm in order to allow balancing the classes. This is achieved by assigning parameters of exponential distribution individually to each class in on-line bagging algorithm.
This procedure akin to random resampling is equivalent to introducing different penalty costs for misclassification of objects of each class. In this work costs are calculated in inverse proportion to a number of samples in the class. Also we use a random set of features for every weak classifier like in Random Forest algorithm (Brieman, 2001). This, together with using Hoeffding trees (Domingos, 2000) as a weak learner, helps to achieve stable classification results and reduce training and classification time.

7. - EXPERIMENTS

Image base

For the experiments we used four road images. They differ in quality and marking and relative areas of defects. We tested our system on the first 18 parts of every road image. All parts



(a) (b) (c) (d)

Figure 6. First pair of pictures: User clicks per screen required to obtain correct road mapping subject to number of image parts already seen by our learnable detection algorithm (a) - *road1*, *road2* data (b) - *road3* and *road4* data. Solid squares show necessary number of clicks if our detection algorithm is applied before user input. Empty squares show necessary number of clicks without use of automatic detection algorithm. Y-axis shows estimated number of clicks and x-axis represents number of processed images. Second pair of pictures: Error of automatic detection algorithm subject to number of image parts already seen by learnable detection algorithm (c) - *road1*, *road* data (d) - *road3* and *road4* data. The error is measured as a fraction of image square misclassified by our detection algorithm. At all plots green lines correspond to *road1* and blue lines correspond to *road2*. Y-axis shows error rate and x-axis represents number of processed images.

are about 0.5 megapixel size and correspond to 5-10 meters of road surface. Some of image parts with results of our automatic detection are shown in Figure 8.

Experiments setup

We have developed a testing framework which emulates user activity at the on-line stage. Given the classification results and ground truth data it starts with automatic thresholds adjusting. Gradient descent algorithm is used to determine a set of thresholds that minimizes total area of misclassified objects. Then user interaction is emulated as follows. At first, our framework corrects all errors of automatic detection which can be amended by relabeling segments of the coarsest segmentation scale. Then testing framework emulates user-aided error correction at subsequent segmentation scale. This procedure is repeated up to the most detailed segmentation scale. Total number of clicks required for errors correction is calculated as a sum of click counts at all segmentation scales. This statistic measures overall usability of our tool for road mapping.

One can see total clicks count per image part measured on 4 roads from our image base in Figure 6 (a, b). *Road1* and *road2* contain greater number of road defects than *road3* and *road4*, therefore larger number of user clicks is required for mapping last two roads. We have compared number of clicks required to achieve accurate mapping when user corrects errors of our automatic detection algorithm with the number of clicks required for mapping road surface from scratch when no automatic detection is performed. One (?- или It can be seen) can see that usage of automatic detection algorithm leads to advance in usability of mapping tool.

As a matter of fact, road marking can be usually found perfectly after processing the second or the third part image. So, the problem of road defects detection is more challenging. Figure 6 (c, d) demonstrates misclassified area of road defects subject to number of image parts seen by detection algorithm.

Figure 7 illustrates false positive and false negative error rates of road defect by pixels on *road1* data. This picture represents usual behaviour of our system. The rate of detected defects increases over time when more defects examples shown to automatic detection algorithm.

In summary, overall error tends to decrease while the number of handled images grows. The system usually starts to distinguish road defects since two or three images have been handled. Some road images like *road3* and *road4* contain a small amount of road defects (some image parts do not contain them at all). Although learning process is slowed down and benefit of using interactive system is reduced on such kind of roads however, usage of automatic detection result still remains beneficial.

8. CONCLUSIONS AND FUTURE WORK

We have presented a tool for efficient interactive mapping of road defects and lane marking on rectified images of road pavement surface. Intensive use of computer vision methods on different stages of our data processing workflow increases usability of the tool.

The most significant drawbacks of our tool is the limitation of using segments in user interaction stage and incapability to correct detection results on sub-segment level. Also our system currently is unable to accommodate to changes of the road structure, e.g. illumination level changing. This drawback can be eliminated if we provide on-line classifier with concept adapting.

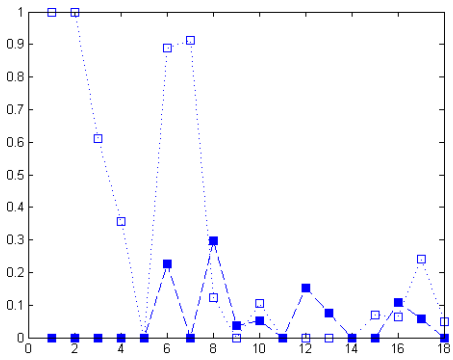


Figure 7. False positive and false negative rates on *road1* data subject to a number of handled road sections. Y-axis shows error rate and x-axis represents number of processed images.

9. REFERENCES

Breiman, L., 2001. Random Forests. *Machine Learning*, 45(1), pp. 5–32.

Criminisi, A., Shotton, J., Winn, J., Rother, C., 2006. TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation. *In proc. of European Conference on Computer Vision,* Graz, Austria.

Domingos, P., Hulten, G., 2000. Mining high-speed data streams. *In proc. of the VI ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* Boston, USA. pp. 71 - 80.

Land, E.H., McCann, J.J. 1971. Lightness and Retinex Theory. *Journal of the Optical Society of America*, 61(1), pp. 1-20.

Leung, T, Malik, J. 1999. Recognizing Surfaces Using Three-Dimensional Textons. *In proc. of the International Conference on Computer Vision*, Kerkyra, Corfu, Greece Vol. 2, pp. 1010-1118,

Oza, N.C. 2005. Online Bagging and Boosting. *In proc. of IEEE International Conference on Systems, Man, and Cybernetics, Special Session on Ensemble Methods for Extreme Environments.* - New Jersey, USA, vol. 3, pp. 2340–2345

Paris, S., Durand, F. 2007. A Topological Approach to Hierarchical Segmentation using Mean Shift. *In proc. of Computer Vision and Pattern recognition*, Minneapolis, Minnesota, USA.

Sudakov, S., Barinova, O., 2008, Velizhev, A., Konushin, A. Semantic segmentation of road images based on cascade classifiers, *In proc. of Pattern Recognition an Image Analysis*, Nizhny Novgorod, Russia. vol. 2, pp. 112-116.

Viola, P., Jones, M. 2002, Robust Real-time Object Detection, *International Journal of Computer Vision*.

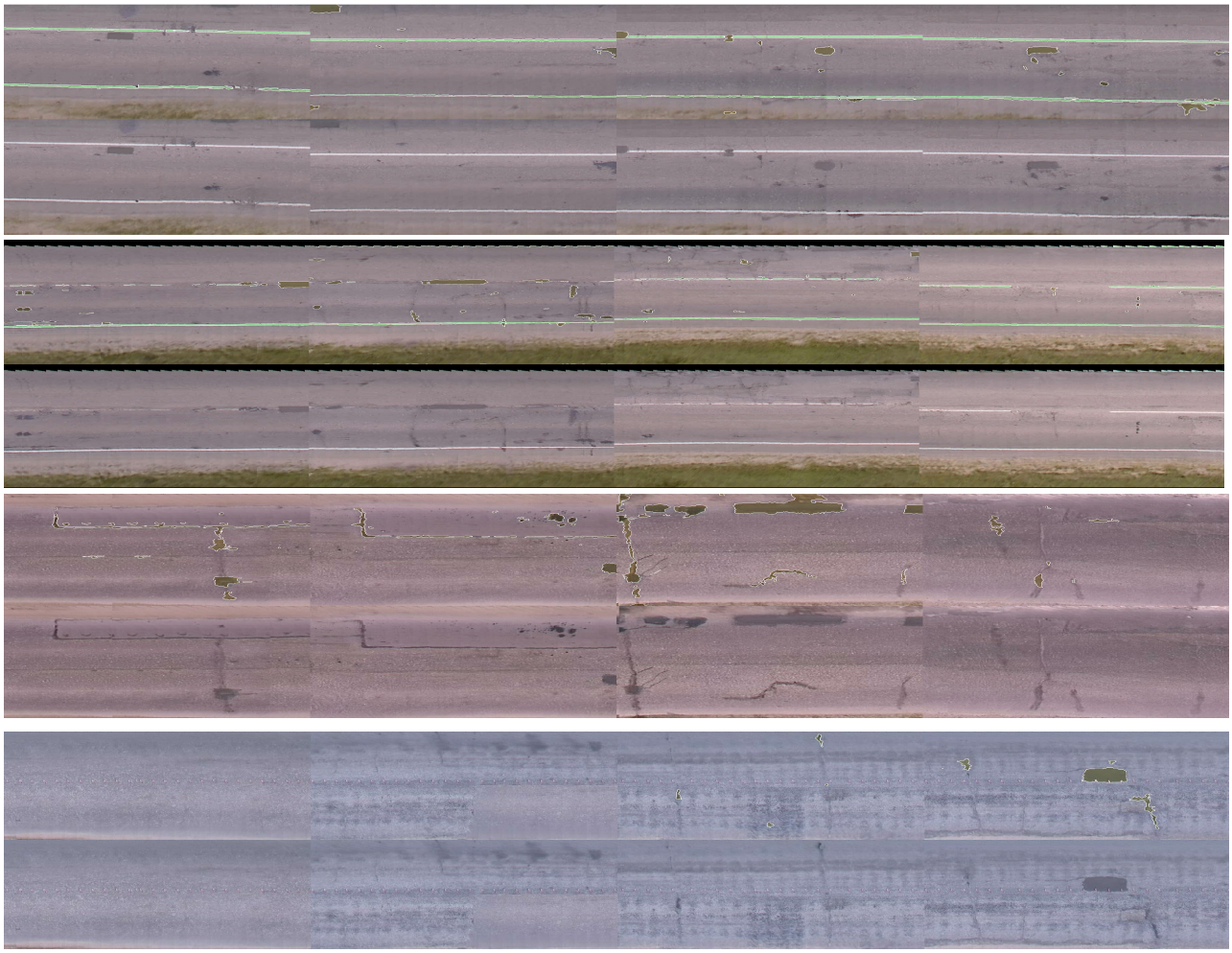


Figure 8. Results of automatic detection of road defects and lane marking. From top to bottom: *road1* data, *road2* data, *road3* data, *road4* data. Image parts number 3, 5, 10 and 18 are shown together with automatic detection results before manual correction. Lane marking is shown in green with blending, road defects are shown in brown with blending. Picture is better viewed in color and magnified.