# Recent Advancements on Drivable Free Space Estimation Using Monocular Vision

Yasir Amir, Haroon Rasheed, Umair Shahid

***Abstract* – This paper presents a overview on recent work on drivable free space estimation with emphasis on monocular vision. Yao et. al. proposed an inference in MRF using various cues based on appearance, edges, spatial and temporal smoothness. Wolcott et. al. added more cues based on perceived motion using optical flow. While Levi et. al. proposed a new column wise regression approach using convolutional neural networks and stixels. All the techniques reviewed in this paper have large processing time, thus seriously limiting their practical application.**

***Index Term* – drivable free space, stixel, monocular vision.**

## I. INTRODUCTION

Motor vehicle accidents (MVA) are caused by driver inattention and poor judgment. Out of these accidents in the U.S 31% of MVAs are due to rear-end collisions, mostly when a front vehicle stops or slows down suddenly without giving the trailing vehicles any warning or very little time to react. Intelligent driver systems have proved to lessen the severity and frequency of accidents [6]. MVA prediction using computer vision is an important, challenging and emerging field, with a lot of potential for research.

In addition to MVA prediction, autonomous driving systems rely on robust techniques for their safe operation. Obstacle detection is the fundamental step in MVA prediction as well as autonomous driving systems. Dense laser scanners have been very successfully used in this respect. Stereo vision is also suitable technology however our focus here is on monocular camera due to cost and package size. Related to MVAs and autonomous driving, the most important question which need to be addressed is "what is the most critical information necessary to avoid obstacle" (and therefore accident). The most critical information is the drivable free space that can be immediately reached without collision. Monocular camera vision offers a cost effective solution for this problem.

### *A) Autonomous driving and obstacle detection*

Some notable work in relation to autonomous driving and obstacle detection was done by Prof. Amnon Shahua of Hebrew University and co-founder of Mobileye. Later on a few important contributions in this respect have been done by Jian Yao et. al. of Torronto University [24], Badino et. al. [17], Benenson et. al. [18], Levi et. al. [28] besides many others.

Yasir Amir, Haroon Rasheed and Umair Shahid are with department of Electrical Engineering, Bahria University Karachi, Pakistan.
Email: myasir.bukc@bahria.edu.pk, haroonrasheed.bukc@bahria.edu.pk, umairshahid.bukc@bahria.edu.pk. Manuscript received Feb 06, 2017 revised on April 20, 2017. Accepted on May 30, 2017.

Work of Prof. Shashua et. al.:

In 2003, they proposed a vision based adaptive cruise control system (ACC) using single camera, where they showed method of calculating range and range rate and also showed that how image geometry effected these quantities. The distance between the camera mounted vehicle and the target vehicle, $Z$ was as $Z = fH/y$ with $f$ being focal length of camera, $y$ being height of vehicle in image and $H$ being the height. This equation was derived on assumption that the horizontal line passing through the camera is parallel to road surface, which off course is not the case in real world scenarios. In addition there is vibration in camera due to vehicle motion, which must also be considered in analysis. Therefore the error in range $Zerr$ due to error of n pixels in location of location of contact point is $Zerr = Zn\text{-}Z$ which is:

$$Z_{err} = Z_n - Z = \frac{fH}{(y+n)} - Z = \frac{nZ^2}{(fH+nZ)}$$

It was observed that typically $n$ is nearly 1 and $fH >> nZ$ therefore $Zerr = nZ^2/(fH)$. This also shows error increases quadratically with distance. The percentage error in depth is given by $Zerr/Z*100$.

Example: Let us consider an example of a 640x480 image ($w$=640) with a horizontal field of view of fov=47 degrees gives a focal length in pixels

$f = (f_{mm}/sensor\text{-}width)w = (w/2)/\tan(fov/2)$,
$f = (640*0.5)/\tan(47*0.5*\pi/180)$

which rounds off to 736pixels. I we suppose the camera height is $H$=1.2m. With 1 pixel error which is error of 5% in depth is expected at a distance of

$Zerr = n\,Z^2/(fH) \rightarrow Z = (Zerr/Z)fH = 0.05*736*1.2 = 44.16m.$

An error of 5% at 44m is quite small as compared to human driver's error. At a distance of 100m the error would be $(Zerr/Z)*100 = nZ/(fH)*100 = 11.32\%$. Therefore ACC systems are quite practical. They also suggested method of determining bounds on accuracy to determine steps which must be taken to improve system performance. The system was tested on highways with promising results [1].

A vision based forward collision warning system was presented by Prof. Shashua *et. al.* where an algorithm was proposed to calculate time to contact (TTC) and possible course directly from position and size of vehicle. This information was obtained directly from image without 3-D representations of scene. Collision avoidance tests were

 ISSN – 1999-4974

performed on test tracks using Mobileye system [2]. In [5] the authors described a system which covers three major capabilities, namely forward collision, headway monitoring/ warning and lane departure. This paper not only described warning features but also sets rule for visual and acoustic human machine interface. This system was installed on commercial fleet and passenger vehicles and its results were quite promising.

The automotive industry requirements are strenuous and often contradicting and the computer vision systems should be able to meet those requirements. The algorithms are expected to have substantial computing power to function dependably in real-time applications under a wide range of day-night and weather conditions, using automotive industry qualified parts. These parts must be able to have long life and ability to with stand harsh operating conditions. Yet the cost must be kept low along with small system size and low power consumption. In order to come up to the challenge, almost all of these crucial requirements are met by a system EyeQ which was developed by Mobileye, which is a complete system on a chip, capable of supporting most of CV algorithms used in variety of related applications such as vehicle detection, lane and pedestrian detection etc. This system is described in detail in [5] where a process of designing an ASIC to support CV algorithms is also presented. The chip supported, lane detection, pedestrian and vehicle detection related applications. Novel algorithms may be developed for the chip which can exploit its advanced computational power. This chip contains two ARM processors (CPU) in addition to four vision computing engines (VCE). One ARM CPU is used for implementing algorithms while the other for communication to the vehicle and general IO.

As part of the 2006 DARPA Grand Challenge, Prof. Shashau and team, developed and tested a system in which they developed a real time system for "finding and tracking" amorphous paths in off-road conditions. They combined geometric projection with learning approach for identifying drivable regions in scene that are familiar. They used geometric projections to deduce yaw and pitch angles. Using 16 Walsh-Hadamard 4x4 binary kernels they performed texture analysis to segment out path regions from non-path regions. They used learning by example principle using boundary based components, which look for path bounding lines. Their combined approach made their vehicle capable of finding path even when vehicle is positioned out of path, a situation which is not frequently encountered by human drivers but by autonomous systems. [3]

An essential part of the collision avoidance system in urban scenarios is pedestrian detection. In another paper Prof. Shashua et. al. described a pedestrian detection system using monocular camera. They proposed an approach of single frame classification based on a novel technique which breaks down class variability by repeatedly training a set series of simple classifiers on clusters of training set. The system was tested and its performance was evaluated only for day time and single weather conditions [4].

*B) Drivable free space Estimation*

Drivable free space may be understood as the space that be immediately reached by (autonomous) vehicle without collision. Up till 2015, self-driving cars typically used Light Detection and Ranging (LIDAR) scanning in every direction to determine drivable free space, Google driverless car is one such example. LIDAR sensors provide plenty of useful information including point appearance. Without using LIDAR and relying only on camera to identify ground plane, is a challenging task [25]. Badino et. al. showed that stereo camera could be used to estimate the immediate drivable free space to replace LIDAR [7]. With LIDAR and/or stereo camera the computation of drivable free space becomes quite easy; however this is a nontrivial task with a single monocular camera. Most of the related previous work ([8],[9],[10],[11],[12] and [13]) in this respect used Laser range finder along with GPS based positioning.

In 1987, A. Elfes used the idea of "occupancy grid" [14], which refers to 2 dimensional grid where every cell models the occupancy evidence of environment. Occupancy grid is normally estimated via Laser range finders or ultrasonic sensors [15].

Badino et. al. used the notion of Stixel world [17] to compute free space and height of objects. Stixel world is the simplified model of the world using ground plane and a set of vertical sticks on ground representing the obstacles. According to them, there were several object descriptors like particles, quadrics, quadtrees, octtrees, patchlets or surfels that partly fulfilled the requirements but they did not attain the level of compactness they were striving for. Therefore they proposed a scheme to represent the 3-D environment in front of vehicle by a set of rectangular sticks or "stixels".

Each stixel stands vertically on ground, has a certain height and has a 3-D position with respect to the camera. The idea is that each stixel defines the border of free space thus approximating the boundaries of obstacles. A scene from an image with a width of 800 pixels, for instance, can be represented by 800/4=200 stixels, if width of each stixel is set to 4 pixels. In this way, this scheme compactly encloses two curves one encloses the drivable free space and runs on ground plane and the other encloses heights of all vertical obstacles which are located at the border line of free space. H. Hirschmuller used semi-global stereo matching to compute the stixel world [16]. In 2011, R. Benenson et. al. showed that stixel estimation can be done without using stereo depth map.[18]

Recently, in [26] Semantic Stixels a new vision-based scene model which is specifically focused on autonomous driving, was presented. Using stixels as primitive elements, the model figures out the geometric as well as semantic outline of a scene, providing a rich but compact generalization of both cues. For semantics they make use of a current deep learning-based scene labeling approach which provides an object class label for each pixel. They used stereo vision to derive pixel level disparity maps which are used to embed geometric information into the model. Their results show that the joint handling of the two cues on Stixel level produces a very compact representation yet at the same time maintaining correctness close to the two individual pixel level input data sources. Their framework was comparable to the related approaches in terms of real time operation and computational costs.


ISSN – 1999-4974

Hoiem et. al. had shown in 2005, that is quite reliably possible to classify a given pixel in image into sky, ground or building [22]. Recently, S. Achar et. al., S. Scherer et. al. and A. M. Neto et. al. proposed free space estimation by using Binary Classification. However their understanding of free space required space behind obstacles as a result efficient and exact inference is not possible ([19], [20] and [21]). Felzenszwalb and Veksler [23] proposed a scheme of modeling a scene using two horizontal curves that divide image into three regions namely top, middle and bottom. Although exact inference is possible, complexity of scheme does not render it suitable for real time applications [24].

*C) Free space estimation using monocular camera*

*Work of J. Yao et. al.* :

In this approach the problem was modeled in such a way that in an image with width=$w$ and height=$h$, '$w$' discrete variables ($y_i$) were taken, where $y_i \in \{1,..,h\}$ , set of $h$ discrete labels. It was also suggested that the states of $y_i$ could be further restricted since $y_i$ could never be above horizon. A simple way of calculating a bound for horizon line using the training images was suggested which was used to restrict the labels in inference procedure. A 1-D chain graph $G=\{V,E\}$ with vertices $V = \{1,\ldots,w\}$ was proposed, with edges $(i,i+1) \in E$ and $i \in \{1,..,w-1\}$. The features from $I_t$ and $I_{t-1}$ were used to calculate curve for image $I_t$. And an **energy function** was defined as:

$$E(y_i, I_t, I_{t-1}) = -\Sigma_{u\in U}\Sigma_i W_u^T \Phi_u y_i - \Sigma_{(i,j)\in E} W_p^T \Phi(y_i, y_j)$$

The first term is unary which uses edges, appearance and temporal information, while the pair wise potential (second term) encode the spatial information. The parameters for energy function are $\boldsymbol{w}=\{\boldsymbol{w_u}, \boldsymbol{w_p}\}$. The parameters were learned using structure prediction. The uninary potentials are explained next.

The first potential focuses on appearance, where the authors used two Gaussian Mixture Models (GMM) one for each road and sea. Each GMM had five components to model the probability of each pixel to be foreground or background. The parameters were learned using Expectation Maximization (EM). Location prior was used to enforce pixels that are always road. They used training data to determine this region and formulated a potential that focuses on the entropy of the distribution in patched around the labels/ pixels of interest. Entropy is calculated in a patch around pixel located at $(i,j)$, in terms of distribution of road or non road pixels, by

$$\varphi_{\text{appearance}}(y_i=k) = H(i,k)\Sigma^{h}_{j=k} H(i,j)$$

Entropy is supposed to be high close to boundary of road/ non-road pixels. They used a cumulative sum that favors pixels that are closer to the moving vehicle and those with non zero entropy, so as to determine a curve that passes through boundary between nearest series of obstacles and road.

The second potential focuses on edges where Canny edge detector was used to detect edges. This assists the the curve being looked for, to get lined up with the natural outline (contour) between free space and obstacles. There would be many edges in the image but the appropriate or the required ones are those that are located at the bottom of image closer to camera. This potential as formulated by the authors is:

$$\varphi_{\text{edge}}(y_i=k) = e(i,k)\Sigma^{h}_{j=k}\, e(i,j) \text{ with}$$

$e(i,j) = 1$ if an edge is located at pixel at $(i,j)$.

The third potential focuses on homography. For a pixel $p(i,j)$ be labeled in image $I_t$ and corresponding pixel $p(i',j')$ in $I_{t-1}$ the authors used homography to impose smoothness across images while maintaining 1-D chain graph during inference. A homography matrix was calculated using ground plane. In this way a one-to-one mapping between pixels on ground in both images ($I_{t-1}$ and $I_t$) could be obtained which also provided mapping of curve representing free space. The mapping is given by following homography mapping:

$$\begin{pmatrix} i' \\ j' \\ 1 \end{pmatrix} = H(t,t-1) \begin{pmatrix} i \\ j \\ 1 \end{pmatrix}$$

Here $H(.)$ defines the homography matrix. The homography potential is defined as:

$$\varphi_{\text{homography}}(y_i=j) = \varphi_u(y_i'=j')$$

Where $\varphi_u(y_i'=j')$ is the unary potential in previous image and with $y_i'=j'$ computed from previously given mapping. Homography was calculated in a RANSAC frame work using SIFT correspondences during the experimental testing.

The second term of energy function which is pair-wise, focuses on the spatial smoothness. It was noted that curve is smooth in absence of obstacles. In presence of obstacles, which happens in a few image columns only, the curve would be non-smooth. They employed a truncated quadratic penalty to reinforce the curve to be smooth. The potential is given by:

The loss augmented inference could be solved using dynamic programming since the loss decomposes into unary potentials.

$$\varphi_p(y_i, y_j) = \begin{cases} \exp(-\alpha(y_i - y_j)^2) & \text{if } |y_i - y_j| \le T \\ \lambda_d & \text{otherwise} \end{cases}$$

*Work of Levi et. al.:*

In their paper [28], Levi et. al. used single color camera in contrast to the existing methods based on 3-D sensing. Their main contribution is that they reduced the problem to a column wise regression problem. The regression is then


ISSN – 1999-4974

solved using convolutional neural networks (CNN). For testing they used the KITTI dataset and they showed that their approach is far better than the rest.

According to Levi et. al. since complete scene labelling (e.g road, sidewalk, building) is a tough task, they adopted a better approach using the concept of stixels and detecting a contact point (pixel) in the image columns that forms the border of obstacle and ground. Their approach is two step, where as a first step they divided the given frame into columns and solved the detection as a regression problem using CNN. This was termed as stixelnet. As a second step they improved results by imposing smoothness constraints and making use of interactions between neighboring columns. They introduced a loss function to train the neural network, which was based on "semi-discrete" representation of obstacle probability. Large quantities of labeled data are essential for training deep CNN. In order to fulfill this need they took advantage of laser scanners which eliminated the need for manual labeling however they further fine tuned the stixelnet by hand labeling. Their method proved to be even superior to stereo based approach using stixels. In the

"KITTI road segmentation challenge" their fine-tuned network was ranked second best, although it did not suitably model all cases.

Stixelnet consists of five layers [28] where the first two layers are fully convolution with 64 filters in first layer and 200 kernels in the second layer, while the layers from three to five are fully connected with 1024, 2048 and 50 neurons, respectively. The network receives a vertical stripe ($I_s$) of the image. The stripe has width *w*, height *h* and the colors for each pixel with dimensions (*w*,*h*,*3*). In this paper they took (*w*,*h*,*3*)= (24,370,3). Taking zero as the first row from top and *h* as the bottom, the problem is finding in $I_s$ the vertical height *y* of the bottom of the closest obstacle called the obstacle position with *y* lies in interval [$h_{min}$,*h*], where $h_{min}$ represents the possible row of horizon (here $h_{min}$ =140). The output of the network is probability distribution over the interval [$h_{min}$,*h*] with *P(y)* = probability that obstacle in $I_s$ is at *y*. It was observed that color images gave better results as compared to gray scale while temporal information did not give any noteworthy performance advantage.

The first two layers are convolutional layers. The first layer convolves the input image of size 24x370x3 with 64 filters at each pixel position. The size of each filter is size 11x5x3. The second convolutional layer has 200 kernels of size 5x3x64. Maximum is then computed over regions which have no overlap between pooled regions by the max-pooling at the output of each layer with sizes 4x3 for the second layer and 8x4 for the first layer.

They solved the obstacle detection problem in two stages. In the first stage stixelnet provided the guess of road limit and obstacle and in the second stage global refinement was done using Conditional Random Field so as to get a globally consistent estimation. In CRF they optimized a potential (ref. to [28]) where the unary potential is the probability of obstacle position determined by stixelnet, while the pair-wise potential penalized discontinuities. The inference can be solved using Viterbi-Algorithm.

An important aspect of the problem is to find if a given pixel is road or non-road. In road segmentation the first two stages are same as in object detection while the third stage performs graph-cut segmentation on image in order to achieve higher accuracy.

For experiments they used KITTI dataset with 56 on road sequences consisting of a couple of hundreds of frames. About 6000 training images and 800 testing images were used. Their experiments show promising results.

*The work of Wolcott et. al.* :

Wolcot et. al. showed in [25] that a monocular grayscale camera could be used to partition a given image into disjoint sets of obstacles and ground plane. Their approach is close to [24] however they took advantage of perceived motion from optical flow in stream of images. In their work they focused on partitioning a stream of image frames into obstacles and prior map. They exploited a "textured prior map" to obtain "appearance models" and "optical flow likelihoods" which could then be integrated into an MRF frame-work. The prior map allowed evaluating ground likelihood by circumstancing belief on the expected appearance from the prior map. In contrast to [22], Wolcott et.al. introduced some new potentials such as "optical flow potential" besides additional potentials such as LIDAR, recursive potentials etc. in an effort to exploit perceived motion along with appearance. Their scheme could be executed at a frame rate of up to 8 frames per second yielding a processing time in range of 0.125-0.21 seconds per frame. Their proposed scheme was tested on a data set that had non-uniform lighting conditions in a challenging urban scenario. Experimental results showed improved robustness. As a future work they suggested that in order to segment objects lying above image partition, extracted optical flow vectors may be used.

## II. CONCLUSION

The techniques and methodologies discussed in this paper focus on monocular vision where the problem is quite challenging from computer vision perspective. Most of the techniques reviewed here, have large processing time thus seriously limiting their practical applications. The processing time of Yao et. al. is 0.1s per frame. Although Wolcott et. al. used more cues and gray scaled image their processing time was not better than former. Improving processing time is an important research area with a potential for a lot of work.

## REFERENCES

[1] Stein, Gideon P., Ofer Mano, and Amnon Shashua. "Vision-based ACC with a single camera: bounds on range and range rate accuracy." Intelligent vehicles symposium, 2003. Proceedings. IEEE. IEEE, 2003.

[2] Dagan E, Mano O, Stein GP, Shashua A. Forward collision warning with a single camera. Proceedings of the IEEE Intelligent Vehicles Symposium; June 2004; Parma, Italy. pp. 37–42.

[3] Alon, Yaniv, Andras Ferencz, and Amnon Shashua. "Off-road path following using region classification and geometric projection constraints." Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Vol. 1. IEEE, 2006.

[4] Shashua, Amnon, Yoram Gdalyahu, and Gaby Hayun. "Pedestrian detection for driving assistance systems: Single-

 ISSN – 1999-4974

frame classification and system level performance." Intelligent Vehicles Symposium, 2004 IEEE. IEEE, 2004.

[5] Stein, Gideon P., et al. "A computer vision system on a chip: a case study from the automotive domain." Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on. IEEE, 2005.

[7] Che-Tsung Lin, Yu-Chen Lin, Long-Tai Chen & Yuan-Fang Wang, "Front Vehicle Blind Spot Translucentization Based on Augmented Reality," 2013 IEEE 78th Vehicular Technology Conference, pp. 1-6, Sep 2013.

[8] H. Badino, V. Frank, R. Mester, "Free Space computation using stochastic occupancy grids and dynamic programming", In ICCV workshop on Dynamic Vision 2007.

[9] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski, "Self-supervised monocular road detection in desert terrain", In RSS, 2006.

[10] A. Angelova, L. Matthies, D. Helmick, and P. Perona, "Fast terrain classification using variable-length representation for autonomous navigation", In CVPR, 2007.

[11] R. Hadsell, P. Sermannet, J. Ben, A. Erkan, M. Scoffier, K. Kavukcuoglu, U. Muller, and Y. LeCun. "Learning long range vision for autonomous off-road driving", JFR, 2009.

[12] D. Silver, J. A. Bagnell, and A. Stentz, "Learning from demonstration for autonomous navigation in complex unstructured terrain", IJRR, 2010.

[13] J. Michels, A. Saxena, and A. Y.NG, "High speed obstacle avoidance using monocular vision and reinforcement learning", In ICML, 2005.

[14] D. Kim, J. Sun, S. M. Oh, J. M. Rehg, and A. F. Bobick, "Traversability classification using unsupervised on-line visual learning", In ICRA, 2006.

[15] A. Elfes, "Sonar-based real-world mapping and navigation", Journal of Robotics and Automation, 1987.

[16] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics: Intelligent Robotics and Autonomous Agents, The MIT Press, 2005.

[17] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information", PAMI, 2008.

[18] H. Badino, U. Franke, and D. Pfeiffer, " The stixel world - a compact medium level representation of the 3d-world", In DAGM, 2009.

[19] R. Benenson, R. Timofte, and L. Gool, "Stixels estimation without depth map computation", In ICCV, 2011.

[20] S. Achar, B. Sankaran, S. Nuske, S. Scherer, and S. Singh, "Self-supervised segmentation of river scenes", In ICRA, 2011.

[21] S. Scherer, J. Rehder, S. Achar, H. Cover, A. Chambers, S. Nuske, and S. Singh, "River mapping from a flying robot: state estimation, river detection, and obstacle mapping", Autonomous Robot, 2012.

[22] A. M. Neto, A. C. Victorino, I. Fantoni, and J. V. Ferreira, "Real-time estimation of drivable image area based on monocular vision", In IV, 2013.

[23] D. Hoiem, A. A. Efros, and M. Hebert, "Automatic photo pop-up", ACM Trans. Graph., 2005.

[24] Felzenszwalb and Veksler, "Tiered scene labeling with dynamic programming", In CVPR, 2010.

[25] J. Yao, S. Ramalingam, Y. Taguchi, Y. Miki, R. Urtasun, "Estimating Drivable Collision-Free Space from Monocular Video", IEEE WACV, Jan 2015.

[26] Wolcott, Ryan W., and Ryan M. Eustice. "Probabilistic Obstacle Partitioning of Monocular Video for Autonomous Vehicles", Proceedings of the British Machine Vision Conference, York, UK. 2016.

[27] Schneider, Lukas, et al. "Semantic Stixels: Depth is not enough." Intelligent Vehicles Symposium (IV), 2016 IEEE. IEEE, 2016.

[28] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables", JMLR, 2005.

[29] Dan Levi, Noa Garnett, and Ethan Fetaya, "Stixelnet: A deep convolutional network for obstacle detection and road segmentation", In Proc. British Mach. Vis. Conf., pages 109.1–109.12, Swansea, United Kingdom, September 2015.

 ISSN – 1999-4974