# Lightweight Monocular Obstacle Avoidance by Salient Feature Fusion

Andrea Manno-Kovacs Levente Kovács
Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI)
Kende u. 13-17, Budapest, Hungary
{andrea.manno-kovacs, levente.kovacs}@sztaki.mta.hu

## Abstract

*We present a monocular obstacle avoidance method based on a novel image feature map built by fusing robust saliency features, to be used in embedded systems on lightweight autonomous vehicles. The fused salient features are a textural-directional Harris based feature map and a relative focus feature map. We present the generation of the fused salient map, along with its application for obstacle avoidance. Evaluations are performed from a saliency point of view, and for the assessment of the method's applicability for obstacle avoidance in simulated environments. The presented results support the usability of the method in embedded systems on lightweight unmanned vehicles.*

## 1. Introduction

The goal of this paper is to present a monocular obstacle avoidance method based on the fusion of robust salient features deployable on lightweight unmanned vehicles.

Regarding monocular obstacle avoidance, there are several methods that either only use visual features or augment them with information from other sensors. For indoor environments, in [27] a navigation framework was presented using a single image for detecting stationary objects and ultrasonic sensing to detect moving objects, using the difference between the current and expected image for detecting static obstacles. [19] used low resolution color segmentation and object detection (trained for 8 object classes) for single camera obstacle avoidance. In [18] an indoor obstacle avoidance method is presented, where landmarks are extracted using feature points and approximate spatial obstacle contours are built using interest point matching among different frames. In [1] a monocular obstacle avoidance method is presented, where a series of captured frames are used to construct a dense depth map every second to aid navigating around objects, but all computations are performed offboard on a base station. In [34] obstacle avoidance was created using low resolution images (for color segmentation) to find ground objects and a sonar sensor for extracting depth information, while in [32] indoor obstacle avoidance was produced using optical flow extracted from image series for finding objects and estimating depth. For outdoor environments, in [26] a monocular obstacle avoidance method was introduced using supervised learning to learn depth cues followed by [20], where monocular obstacle avoidance was presented for aerial vehicles using Markov Random Field classification modeling the obstacles using color and texture features, training the model for obstacle classes with labeled images. In [6] a visual navigation solution was described, following a sequence of images acquired in a training phase, avoiding new obstacles using the camera and a range scanner. In [10] a monocular approach was presented for recognizing forest trails and navigating a quadrotor micro aerial vehicle in such an environment, by using a deep learning approach to recognize trail directions, not for avoiding obstacles. In [7] an obstacle avoidance approach was presented for autonomous watercrafts using a single camera, using optical flow to detect and track potential obstacles, based on an occupancy grid approach (using GPS and inertial sensors). In [28] a deep convolutional network was used for spatio-temporal cue analysis for object avoidance in traffic conditions, by saliency-based modeling of the importance of scene objects. [33] presents a salient object detection method on monocular imagery for planetary rover robots working on images with homogeneous background without the need for a priori training. [15],[16] introduced a monocular obstacle avoidance method for both indoor and outdoor environments, using a depth-like feature map (Dmap) based on relative focus maps, not requiring a priori training or learning of specific environments or objects categories. The current work also uses the relative focus maps as an element of the fusion process, and proposes a more robust and better performing solution.

As the analyzed environments may be diverse, it might be hard to make prior assumptions about specific surroundings. From a human vision point of view, the eyes are continuously fixating on the most important or salient areas or targets (such as obstacles), meanwhile filtering the less relevant visual information and calculating a saliency map. To

model such behavior, saliency-based solutions have been introduced for various environments. The method of this paper uses a single camera without other sensors. It does not work by detecting or recognizing objects, but on the overall analysis of a fused salient feature map to avoid possible collisions. The approach has the benefits of not being scenario-constrained, not requiring a priori training and not having any requirements or constraints regarding camera motion or obstacle classes. The method is also frame rate independent, since it processes single frames, thus it is applicable in embedded systems of various capabilities. We evaluated the proposed method through comparisons with saliency methods and evaluated its obstacle avoidance capabilities in simulated environments. The fused salient feature map incorporates information about feature points, texture, edges and local orientation provided by the Textural-Directional Harris based Features (TDHF) along with the relative depth information (Dmap) features, resulting in a more higher level and more reliable solution.

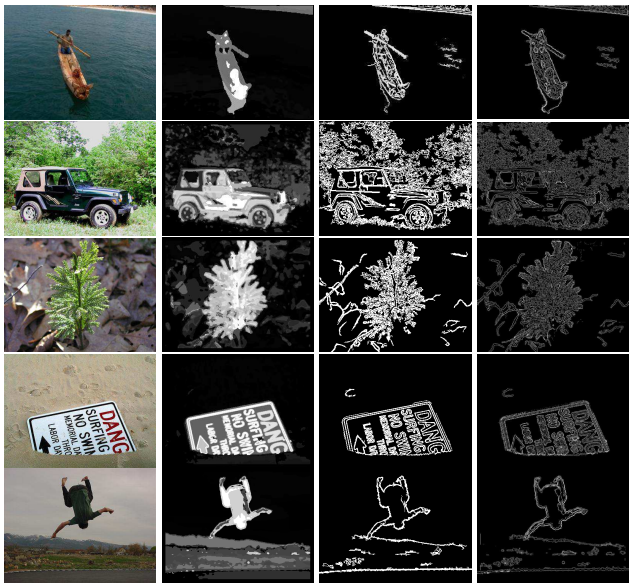
## 2. Salient Features for Obstacle Avoidance

In the following we present the steps of the Textural-Directional Harris based Feature map (TDHF) and its fusion with the relative focus feature map (Dmap) - Sec. 2.3 - and its application for monocular obstacle avoidance.

### 2.1. Textural-Directional Feature Map

Since we do not assume to have a priori information about obstacles, we use a bottom-up saliency approach, biologically inspired and task-independent, which is driven by low-level image features which can be normalized and combined, using different models and scales to calculate a saliency value for image pixels. Recent state-of-the-art saliency methods are using various features, like contrast [5] or texture [31]; others use less traditional ones, like analyzing the log spectrum of the image [12]. In [23] was shown that orientation information from the gradients in the vicinity of the interest points is a valuable feature for object representation: interest points are calculated as the local maxima of a modification of the Harris characteristic function [11], emphasizing both edges and corners in a balanced manner [14], then, based on orientation information, relevant edges can be emphasized for creating a feature map by fusing edges with other features.

Based on [23],[14], we will use the Textural-Directional Harris based Feature (TDHF) model, calculated as a fusion of structural and textural features. The structural part contains improved edge data based on the modified Harris characteristic function, interest point set and the main local orientation map. The textural part is based on the texture distinctiveness map of [31].



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

Figure 1. Main visual steps of the TDHF calculation: (a) is the original image; (b) $T$ texture distinctiveness map; (c) improved $S$ structure feature map; (d) TDHF feature map.

#### 2.1.1 Texture Distinctiveness Map

The statistical texture distinctiveness model was introduced in [31],[8] based on a sparse texture model of the image, where rotation-invariant, neighborhood-based textural representations are used to define representative texture atoms in the image and to build a sparse model of 20 textures. After extracting the texture model, the $T(x,y)$ texture map computes the distinctiveness of each texture compared to the others, where a higher value defines a more distinct region. The distinctiveness value of an atom is assigned to all of its pixels, resulting in the $T$ map. Fig. 1 (b) shows examples for extracted texture features.

#### 2.1.2 Harris Based Feature Map and Direction Feature Extraction

To emphasize the structural information of the image, [14] proposed a modification of the Harris detector's characteristic function for object boundary detection. The local maxima of the Modified Harris for Edges and Corners (MHEC) characteristic function formalizes an interest point set with points on object boundaries. The proposed characteristic function is based on the $\lambda_1$ and $\lambda_2$ eigenvalues of the Harris matrix [11]:

$$H_{\text{mod}} = \max(\lambda_1, \lambda_2). \quad (1)$$

Structural information represented by the $H_{\text{mod}}$ function will be used in the proposed feature model. By extracting its local maxima, the point set $P_{\text{MHEC}}$ (red pixels in Fig. 2

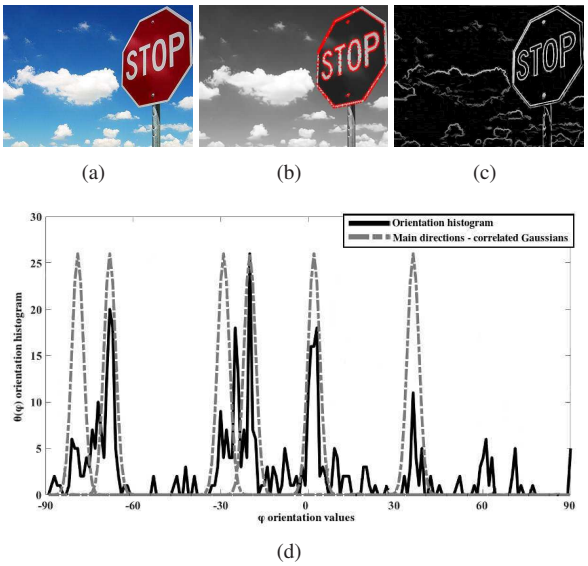



Figure 2. Calculating the relevant orientations of the ROI: (a) input image; (b) MHEC point set (red dots), the $\varphi$ orientation values are calculated for these pixels; (c) the $D_{\mathrm{MFC}}$ improved directional edge map; (d) shows the $\vartheta(\varphi)$ orientation histogram in black and the correlated Gaussian functions for the salient directions in gray.

(b)) represents important contours and relevant direction information can be defined by analyzing their vicinities:

$$P_{\mathrm{MHEC}} = \left\{ p_i : H_{\mathrm{mod}}(p_i) > T_{\max} \ , \ p_i = \underset{r \in b_i}{\operatorname{argmax}} \{ H_{\mathrm{mod}}(r) \} \right\}, \tag{2}$$

where a pixel $p_i$ is a member of the set, if it has larger value than its 8-connected neighbors and it exceeds an adaptive $T_{\max}$ threshold (calculated by Otsu's method [29]).

Local direction as a feature has been adapted earlier for edge and contour detection, however some [4],[30] cannot handle multiple orientation cases (like corners), and some [25],[3] calculate the orientation value on the pixel-level, losing the scaling nature of the feature. Other methods apply histogram binning [35] which is only a loose estimation. In our case the task is twofold: 1). proper direction information has to be extracted, and 2). an edge detection method is required which can handle directions and contours.

For the first point, we use an algorithm for direction feature extraction [24], using the $P_{\mathrm{MHEC}}$ point set for salient direction extraction. We analyze local gradient orientation density (LGOD) [2] in a small $W_n(i)$ neighborhood ($n \times n$) around the members of the $P_{\mathrm{MHEC}}$ point set and assign the main direction to the $i^{\mathrm{th}}$ point:

$$\varphi_i = \underset{\varphi \in [-90,+90]}{\operatorname{argmax}} \{ \lambda_i \}, \tag{3}$$

$$\lambda_i(\varphi) = \frac{1}{N_i} \sum_{r \in W_n(i)} \frac{1}{h} \cdot \| \nabla g_r \| \cdot k \left( \frac{\varphi - \varphi_r^{\nabla}}{h} \right), \tag{4}$$

where $\nabla g_i$ is the gradient vector for the $i^{\mathrm{th}}$ point with $\| \nabla g_i \|$ magnitude and $\varphi_i^{\nabla}$ orientation, $N_i = \sum_{r \in W_n(i)} \| \nabla g_r \|$ and $k(\cdot)$ is a non-negative, symmetric function, chosen as a Gaussian smoothing kernel with $h = 0.7$ bandwidth parameter.

After calculating the $\varphi_i$ for each point, we obtain a $\vartheta(\varphi)$ orientation histogram, representing the main orientations of the image (black in Fig. 2 (d)). To calculate the salient orientation a Gaussian function ($\eta(.)$, with $m$ mean, $d_\vartheta$ standard deviation) is correlated iteratively to the $\vartheta(\varphi)$ [24] to maximize $\alpha(m)$ (Fig. 2 (d) in gray):

$$\alpha(m) = \int \vartheta(\varphi) \eta(\varphi, m, d_\vartheta) \, d\varphi. \tag{5}$$

The $m$ mean represents the most correlating orientation. The iteration stops if: 1). the correlated Gaussians cover a fixed ratio (80%) of the $P_{\mathrm{MHEC}}$ points; 2). the $\alpha$ correlation rate is starting to decrease. The result of the process is a set of salient orientations, the input for a direction selective edge detection algorithm (described in the following).

#### 2.1.3 Direction Selective Edge Map

Referring to the above, we need an edge detection which can exploit the salient directions and can handle formations with multiple orientations (like corners) on a higher, object level. We apply the Morphological Feature Contrast (MFC) operator [36], which is able to distinguish background textures and isolated salient features. A linear extension of MFC, also introduced in [36], is able to extract linear features in defined directions. MFC has separate operators for bright and dark features, defined as the difference of the original signal and one of its envelopes. After removing texture details and extracting potential edges, we apply the mentioned linear filter for linear feature detection. This emphasizes edges in the salient orientations and background information is reduced, resulting in a less noisy and directional feature enhanced $D_{\mathrm{MFC}}$ edge map (Fig. 2 (c)).

#### 2.1.4 Textural-Directional Harris Based Feature Map (TDHF)

We fuse the obtained $D_{\mathrm{MFC}}$ edge map with the $H_{\mathrm{mod}}$ (Eq. 1) Harris based feature map to include boundary information. Both functions are rescaled to reduce their range, and the directional-structural feature map has the following form (visual examples in Fig. 1 (c)):

$$S = \max(\max(0, \log(D_{\mathrm{MFC}})), \max(0, \log(H_{\mathrm{mod}}))). \tag{6}$$

To further improve the salient feature map, we also incorporate textural information in the Textural-Directional Harris based Feature (TDHF) model:

$$f_{\mathrm{TDHF}} = \gamma |\nabla S(x,y)| + (1-\gamma) |\nabla T(x,y)|, \tag{7}$$

Figure 3. (a) Example images from the MSRA dataset, rectangles showing the regions extracted based on Dmap, (b) the Dmap feature maps.

where the $\gamma$ is a balancing parameter between structural and textural parts (a constant $\gamma = 0.3$ is used). Visual examples for TDHF maps are shown in Fig. 1 (d).

### 2.2. Relative Focus Feature Map (Dmap)

The second element of the fused salient feature map is obtained by the so called relative focus map extraction method [17]. The method was originally introduced for relative classification of image regions based on their estimated blurriness, producing an intensity map with higher values representing more in-focus regions. It was also shown to be usable for separating differently textured regions. Later it was also used for monocular obstacle avoidance [15],[16], where the produced feature map (denoted by Dmap) was the basis for suggesting possible movement directions for ground and aerial robots. For the Dmap method, we use $32 \times 32$ image blocks with 16 pixel overlap, and run 10 iterations on each block. Then, we use the obtained local reconstruction errors in a linear classification producing the Dmap feature map. Fig. 3 shows examples for generated maps for images from the MSRA dataset [22].

### 2.3. Fused Salient Feature Map (Dmap+TDHF)

The final feature map Dmap+TDHF is the result of the fusion of the Textural-Directional Harris based Feature map (TDHF) with the relative focus feature map (Dmap). We build on TDHF's capability of producing a feature map inherently including feature point, texture, edge and local orientation information thus providing a powerful source of information that, when combined with the Dmap's relative depth information, results in a fused map that incorporates both object-based and relative depth data.

The fused feature map is produced through the following steps:

1. Extract Dmap ($f_{Dmap}$).
2. Extract $f_{\text{TDHF}}$.
3. Filter $f_{\text{TDHF}}$ to produce $f'_{\text{TDHF}}$:
   (a) Extract blobs from $f_{\text{TDHF}}$.
   (b) For each blob, assign its maximum intensity value to the whole region.
   (c) Morphologically dilate the resulting image (with a small $3 \times 3$ rectangular structuring element to eliminate eventual small holes), denoting the result as $f'_{\text{TDHF}}$.
4. Fuse the feature maps:

$$f_{Dmap+TDHF} = \varepsilon \cdot f_{Dmap} + (1-\varepsilon) \cdot f'_{\text{TDHF}}. \quad (8)$$

Figure 4. (a) Example images from the MSRA dataset, rectangles showing the final regions, (b) the filtered $f'_{TDHF}$ maps, (c) the final Dmap+TDHF feature maps.

Based on extensive empirical tests, we propose a constant $\varepsilon = 0.5$ (except in the rare cases when the TDHF produces an empty map, then we use 1.0 as a fallback).

Fig. 4 shows some visual examples of the produced fused feature maps, and the resulting marked rectangular salient regions. During the evaluations from a saliency perspective (Sec. 3.1) these will be the maps and resulting regions that will be used when comparing with other saliency approaches. When using these maps for obstacle avoidance, they will be looked upon from a reverse perspective, with the goal being to avoid eventual collisions, as will be detailed in the following section.

### 2.4. Using Dmap+TDHF for Obstacle Avoidance

A contribution of this paper is the use of the fused Dmap+TDHF feature map for automatic monocular obstacle avoidance. The main targeted platforms of this approach are mobile sensing units, typically unmanned ground and aerial robots. To use the above generated Dmap+TDHF feature maps for avoiding collisions, our goal is to take the produced maps and find regions in them which likely do not contain near objects, i.e, we are looking for regions which are non-salient, thus have the lowest intensity in the produced feature maps.

Following [15], we scan the resulting fused feature map and we partition it into $3 \times 3$ regions $R_m$ ($m = 1...9$), and we look for regions that contain the maximum number of pixels that are below the detection threshold: $R = \max(SR_m)$, where $SR_m = |\{R_m(i) | R_m(i) < \tau\}|$; $R_m(i)$ are the pixels in region $R_m$ and we set $\tau$ to 20%. If more regions produce the same maximum $R$ value, then the region will be picked

which has the lower sum of feature map intensities that fall below the detection threshold.

Based on the above, we can propose a movement direction, towards the region that has been selected. There can be situations when no such region exists: either when there is no such region at all (the map suggests an unavoidable obstacle), or when we find such a region, but its area is too small (usually we only accept regions with area larger than 5% of the frame). In such situations we propose a stop condition, which means the method cannot suggest a movement direction. When the method can find a movement direction to propose, it will signal such directions as FWD, E, or W (forward, right turn, left turn). In case of a "STOP" signal the robot will make 20 degree left turns until a movement direction can be proposed.

## 3. Evaluations

We present the results of evaluations regarding the proposed Dmap+TDHF feature map from two points of view. The first part deals with evaluating the obtained feature map from a saliency point of view, for which we generate the feature maps for the user labeled images of the Microsoft Research MSRA Salient Object Database[1],[22] and compare it with other methods. In the second part, we evaluate the performance of the proposed feature map for obstacle avoidance in simulated environments, comparing with other saliency-based methods. The main objective here is to showcase a practically usable monocular obstacle avoidance solution, showing that its usability rests on its good performance from a saliency perspective.

### 3.1. Saliency-based evaluation

The MSRA dataset consists of a larger (20840 images labeled by three users) and a smaller (5000 images labeled by nine users) subset. To show the performance of the proposed feature map, we compare it with several other salient map generation methods: the Dmap method [16], the Saliency Toolbox (SBOX)[2] [13] (taking its 'Winner Take All' outputs similarly to [22]), the Histogram Based Contrast (HC) and the Region Based Contrast (RC) methods of [5], and the Spectral Residual (SR) method [12]. There are several other salient region extraction methods, among which we selected recent methods that also have usable sources available that could be ported to different platforms.

For our purposes, i.e., monocular obstacle avoidance, the main goal is not to produce pixel perfect salient regions, but to be able to broadly estimate such regions so they can be avoided. Thus, for each method, we fit rectangles on the obtained feature maps (keeping the top 90% of the generated intensity maps and fitting rectangles over the obtained maps), and compare such rectangular regions with the similar rectangular ground truth labels of the MSRA dataset. First, for each image, we take all the ground truth regions created by users (3 and 9 respectively), and we create an average rectangular region for each image, which will be the average rectangle of all the user-provided regions. Then we extract the region boundaries with the proposed and the compared methods for each image.

Then, we use the similarity between regions to compare the ground truth and the generated regions. For comparisons, we use the same metrics as in [15], namely the Boundary Displacement Error [9] (BDE) and the Jaccard-index (i.e., intersection over union, IOU) [21] (J). Fig. 5 shows visual examples for accepted regions for images of the dataset for all methods.

We included numerical comparison results in Table 1. The first value ("%") is the region acceptance rate according to the metrics in [15] (i.e., regions have at least a 25% overlap and the centers of mass are close together), while the other values show the Jaccard/IOU similarity and the BDE difference values. For the smaller dataset, the proposed method achieves the highest acceptance rate, while being a close second and third according to the Jaccard and BDE values respectively.

For the larger dataset, the proposed method also achieves the higher acceptance rate, while having the best Jaccard value and a close second in BDE. Overall, we can state that the proposed Dmap+TDHF method has a good saliency performance and for our purposes it is good for extracting regions to be avoided: it produces contiguous regions with somewhat wider boundaries (suggested by larger BDE values), which in our case for obstacle avoidance is actually a positive property. This will also be supported by the collision rate statistics.

### 3.2. Obstacle avoidance evaluation

For evaluating the proposed feature map's performance for obstacle avoidance we implemented the Dmap+TDHF, Dmap, HC, RC and SR methods for ROS (the Robot Operating System[3]). We performed tests of the methods in simulated environments. For the simulations, we used a virtual environment running in Gazebo[4], with a simulated TurtleBot ground robot equipped with a camera. The bot was also equipped with bumpers to detect ground truth collisions. Fig. 6 shows sample images for the environment.

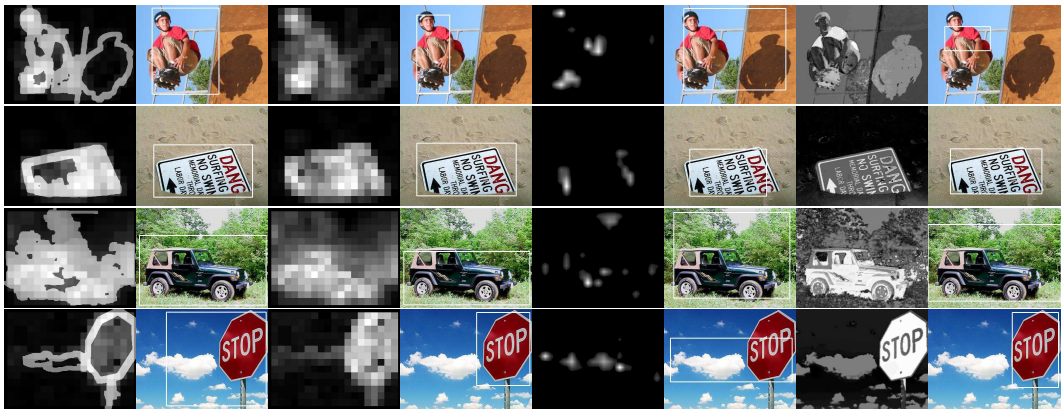
First and foremost, our goal is to achieve a low collision rate, thus creating a method that enables the autonomous vehicle to browse around, rarely hitting obstacles. Fig. 7 (a) shows collision rates. Results contain the averages from all 3 rooms of the virtual environment for a total of 1500 movements for each algorithm performed by the TurtleBot.

[1] http://research.microsoft.com/en-us/um/people/jiansun/salientobject/salient_object.htm

[2] http://saliencytoolbox.net

[3] http://www.ros.org

[4] http://gazebosim.org




(a) (b) (c) (d) (e) (f) (g) (h)

Figure 5. Example images with obtained regions based on the extracted maps with (a,b) the proposed Dmap+TDHF; (c,d) Dmap [16]; (e,f) SBOX [13]; (g,h) HC [5].

| | % | $\mu$BDE | $\sigma$BDE | $\mu$J | $\sigma$J |
|---|---|---|---|---|---|
| | MSRA set B (5000 images, 9 labels) | | | | |
| SBOX | 83.42 | 41.3 | 19.17 | 0.5 | 0.16 |
| HC | 82.08 | 34.33 | 24.74 | **0.58** | 0.22 |
| RC | 81.56 | **28.39** | 20.32 | 0.55 | 0.24 |
| SR | 24.3 | 46.45 | 19.64 | 0.36 | 0.16 |
| Dmap | 83.28 | 45.75 | 25.41 | 0.45 | 0.19 |
| Dmap+TDHF | **97.3** | 37.55 | 26.19 | 0.55 | 0.23 |
| | MSRA set A (20840 images, 3 labels) | | | | |
| SBOX | 88.37 | 40.18 | 19.22 | 0.51 | 0.16 |
| HC | 86.76 | 35.76 | 23.42 | 0.56 | 0.19 |
| RC | 80.3 | **33.46** | 20.78 | 0.5 | 0.22 |
| SR | 25.53 | 47.67 | 19.19 | 0.36 | 0.16 |
| Dmap | 83.98 | 44.97 | 23.46 | 0.46 | 0.18 |
| Dmap+TDHF | **97.14** | 34.15 | 23.6 | **0.58** | 0.21 |

Table 1. Acceptance rate (%), mean ($\mu$) and deviance ($\sigma$) of BDE and the Jaccard similarities for MSRA sets A and B using the proposed method (Dmap+TDHF), Dmap [16], SBOX [13], HC [5], RC [5] and SR [12]. (Best values in bold.)

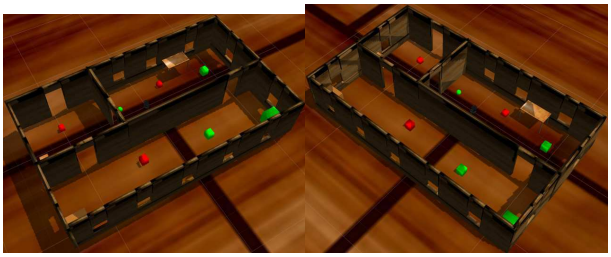


Figure 6. Images from the test environment.

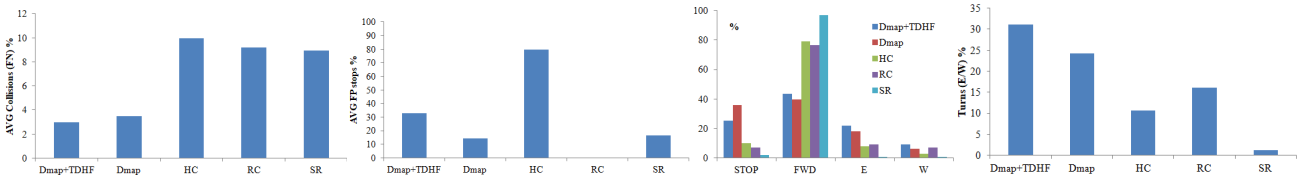
Results show that the proposed approach has a lower collision rate than the other evaluated methods, i.e., when freely browsing, the proposed method causes less bumps into obstacles.

As described above, STOP signals are generated when the methods cannot find a direction to propose at a given situation (i.e., there seems to be no "way out" at a given position and point of view). Fig. 7 (b) shows the rates (w.r.t. all performed moves) of the false STOP signals generated by the different methods, i.e., when the methods falsely observe an unavoidable obstacle in front of the camera. The proposed method has a relatively high false positive STOP rate, however, when observed in combination with the collision rate figures, the proposed method and Dmap are the better performers. Among these two, Dmap+TDHF performs with less collisions with a higher false positive rate, which translates to a better practical performance, albeit with more turns during the browsing movement of the robot.

To provide further details, Fig. 7 (c) shows the ratio of movement direction proposals of the different methods. The graphs show that overall the Dmap+TDHF and Dmap methods perform better, since they make more turns based on detections - relevant part detailed in Fig. 7 (d), which shows the aggregated right/left (E/W) turn ratios -, while the other methods make more forward (FWD) movements (they detect less obstacles) which is in accord with their higher collision rates.

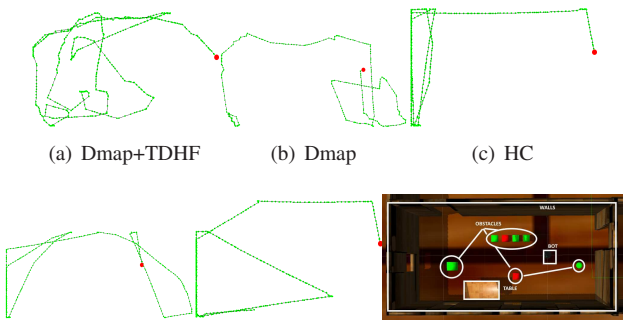
Regarding the relation between the proposed method and Dmap, the figure shows that Dmap+TDHF has a decreased forward movement ratio and increased right/left (E/W) movements, which is in accord with its lower collision rate (since it detects obstacles better, it makes more right/left turns to avoid them), which also results in a lower stopping rate (stop is signaled when an algorithm cannot find a "way out" from the current point of view).

Fig. 8 shows motion paths from the simulation for all methods (the red dot shows the constant starting position). The proposed approach enables the robot to move around more freely and cover more area. Others tend to follow longer linear paths between two collisions, while the proposed method makes more turns during its browsing, going




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

Figure 7. (a) Collision rates for the simulated environment. Values are percentages w.r.t. all the performed bot movements. (b) The rates of false STOP signals in the simulated environment for all methods. (c) The ratios of movement direction proposals of the different methods. (d) The aggregated right/left (E/W) turn ratios w.r.t. all movements for all methods.



(a) Dmap+TDHF (b) Dmap (c) HC

(d) RC (e) SR (f) The top view of the scenario.

Figure 8. Visualized movement positions/paths from a simulated scenario for all methods.

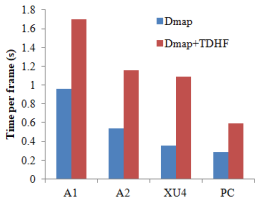



Figure 9. Time requirements of the proposed Dmap+TDHF and the Dmap approaches for the processing of a single frame on all evaluated platforms.

more around obstacles than hitting them. This supports the general conclusion that the proposed approach is better in avoiding obstacles.

Regarding computational time, we evaluated the proposed method on several platforms, and compared it with the Dmap approach which ran with 1 frame/second on 3-5 years old hardware. Our goal was to achieve at least the same performance for Dmap+TDHF on current hardware. The hardware were smartphones with Qualcomm Krait 400/MSM 8974 2.3 GHz and Exynos M1 2.60 GHz (denoted by A1 and A2); an ODROID-XU4 with 2GHz Cortex-A15 (denoted by XU4); a desktop PC with Intel Core i7 930 2.80 GHz (denoted by PC). The results are shown in Fig. 9. There are still optimization possibilities, yet the numbers show that at least a 1 frame/second speed can be achieved on current hardware. These results combined with the average 2-3% collision rate support real world applicability.

## 4. Conclusions

We presented a monocular obstacle avoidance method based on the fusion of structural and directional salient features. The intended platforms are embedded systems for autonomous vehicles as a part or a basis for visual navigation. The method does not need a priori training, is not constrained to a particular application scenario, is frame rate independent, thus usable on embedded vision systems with varying capabilities, has a low collision rate, and shows a performance level suitable for deployment.

## Acknowledgements

This work has been also supported by the Hungarian National Research, Development and Innovation Fund (NKFIA) grants nr. KH-125681 and K-120499.

## References

[1] H. Alvarez, L. Paz, J. Sturm, and D. Cremers. Collision avoidance for quadrotors with a monocular camera. In *Proc. of the 14th Intl. Symposium on Experimental Robotics*, pages 195–209, 2016.

[2] C. Benedek, X. Descombes, and J. Zerubia. Building development monitoring in multitemporal remotely sensed image pairs with stochastic birth-death dynamics. *IEEE Tr. on Pattern Analysis and Machine Intelligence*, 34(1):33–50, 2012.

[3] J. Bigun, G. H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Tr. on Pattern Analysis and Machine Intelligence*, 13(8):775–790, 1991.

[4] J. Canny. A computational approach to edge detection. *IEEE Tr. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[5] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE Tr. on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2015.

[6] A. Cherubini and F. Chaumette. Visual navigation with obstacle avoidance. In *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1503–1598, 2011.

[7] T. El-Gaaly, C. Tomaszewski, A. Valada, P. Velagapudi, B. Kannan, and P. Scerri. Visual obstacle avoidance for autonomous watercraft using smartphones. In *Proc. of Autonomous Robots and Multirobot Systems workshop (ARMS)*, 2013.

[8] K. Fergani, D. Lui, C. Scharfenberger, A. Wong, and D. A. Clausi. Hybrid structural and texture distinctiveness vector field convolution for region segmentation. *Computer Vision and Image Understanding*, 125:85–96, 2014.

[9] J. Freixenet, X. Munoz, D. Raba, J. Marti, and X. Cufi. Yet another survey on image segmentation: Region and boundary information integration. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 408–422, 2002.

[10] A. Giusti, J. Guzzi, D. Ciresan, F.-L. He, J. P. Rodriguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. D. Caro, D. Scaramuzza, and L. Gambardella. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1(2):661–667, 2015.

[11] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of the 4th Alvey Vision Conf.*, pages 147–151, 1988.

[12] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[13] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Tr. on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 2002.

[14] A. Kovacs and T. Szirányi. Harris function based active contour external force for image segmentation. *Pattern Recognition Letters*, 33(9):1180–1187, 2012.

[15] L. Kovács. Single image visual obstacle avoidance for low power mobile sensing. In *Proc. of Advanced Concepts for Intelligent Vision Systems (ACIVS)*, pages 261–272, 2015.

[16] L. Kovács. Visual monocular obstacle avoidance for small unmanned vehicles. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops (12th Embedded Vision Workshop)*, pages 877–884, 2016.

[17] L. Kovács and T. Szirányi. Focus area extraction by blind deconvolution for defining regions of interest. *IEEE Tr. on Pattern Analysis and Machine Intelligence*, 29(6):1080–1085, 2007.

[18] J.-O. Lee, K.-H. Lee, S.-H. Park, S.-G. Im, and J. Park. Obstacle avoidance for small UAVs using monocular vision. *Aircraft Engineering and Aerospace Technology*, 83(6):397–406, 2011.

[19] S. Lenser and M. Veloso. Visual sonar: Fast obstacle avoidance using monocular vision. In *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[20] I. Lenz, M. Gemici, and A. Saxena. Low-power parallel algorithms for single image based obstacle avoidance in aerial robots. In *Proc. of IEEE Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 772–779, 2012.

[21] M. Levandowsky and D. Winter. Distance between sets. *Nature*, 234:34–35, 1971.

[22] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[23] A. Manno-Kovacs. Direction selective vector field convolution for contour detection. In *Proc. of IEEE Intl. Conf. on Image Processing (ICIP)*, pages 4722–4726, 2014. "Top10%".

[24] A. Manno-Kovacs and T. Szirányi. Orientation-selective building detection in aerial images. *ISPRS J. of Photogrammetry and Remote Sensing*, 108:94–112, 2015.

[25] R. Mester. Orientation estimation: Conventional techniques and a new non-differential approach. In *Proc. 10th European Signal Processing Conference*, pages 921–924, 2000.

[26] J. Michels, A. Saxena, and A. Y. Ng. High speed obstacle avoidance using monocular vision and reinforcement learning. In *Proc. of the 21st Intl. Conf. on Machine Learning (ICML)*, pages 593–600, 2005.

[27] A. Oh, A. Kosaka, and A. Kak. Vision-based navigation of mobile robot with obstacle avoidance by single camera vision and ultrasonic sensing. In *Proc. of IEEE Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 704–711, 1997.

[28] E. Ohn-Bar and M. M. Trivedi. Are all objects equal? Deep spatio-temporal importance prediction in driving videos. *Pattern Recognition (in press)*, 2016.

[29] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Tr. on Systems, Man and Cybernetics*, 9(1):62–66, 1979.

[30] P. Perona. Orientation diffusions. *IEEE Tr. on Image Processing*, 7(3):457–467, 1998.

[31] C. Scharfenberger, A. Wong, K. Fergani, J. S. Zelek, D. Clausi, et al. Statistical textural distinctiveness for salient region detection in natural images. In *Proc. of IEEE Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, pages 979–986, 2013.

[32] K. Souhila and A. Karim. Optical flow based robot obstacle avoidance. *Intl. Journal of Advanced Robotic Systems*, 4(1):13–16, 2007.

[33] C. Spiteri, A. Shaukat, and Y. Gao. Structure augmented monocular saliency for planetary rovers. *Robotics and Autonomous Systems*, 88:1–10, 2017.

[34] C. N. Viet and I. Marshall. Vision-based obstacle avoidance for a small, low-cost robot. In *Proc. of IEEE Intl. Conf. on Advanced Robotics (ICAR)*, 2007.

[35] S. Yi, D. Labate, G. R. Easley, and H. Krim. A shearlet approach to edge analysis and detection. *IEEE Tr. on Image Processing*, 18(5):929–941, 2009.

[36] I. Zingman, D. Saupe, and K. Lambers. A morphological approach for distinguishing texture and individual features in images. *Pattern Recognition Letters*, 47:129–138, 2014.