A Saliency-Based Solution For Robust Off-Road Obstacle Detection

Pedro Santana, Magno Guedes, Luís Correia and José Barata

Abstract—Stereo-based off-road obstacle detection is a complex and still open problem. The challenges are in great extent related to computational cost and noise level. Previous work has shown that visual saliency and voting mechanisms are extremely effective in tackling these issues. This paper proposes a set of extensions to these mechanisms, to further improve the detector’s speed-accuracy trade-off as well as its robustness. The observed enhancements are due in part to the adaptive way saliency is accounted for by the detector during the image scanning procedure. Additionally, detector’s positive results are in turn used to boost the saliency map itself, thus reinforcing the analysis of relevant regions of the image. To enable detection in highly roughed terrain, the detector’s invariance to the robot’s posture is also enhanced. Experimental results show that, with the extended detector, denominated of ESaLOD, higher levels of robustness, accuracy and computational efficiency are attained.

I. INTRODUCTION

Stereo vision aggregates a set of important features for all-terrain service robots, such as general purpose capabilities, small energetic footprint, light weight, small size and low cost. However, the large amount of generated data, which is also noisy, presents a series of challenges. These, allied to the unstructured nature of off-road environments, makes the use of stereo vision for robust and fast obstacle detection a problem still open.

A typical approach for efficient obstacle detection is to assume that the terrain can be approximated by a planar model [1], [2], [3]. However, off-road environments are often highly uneven, breaking down this assumption. In this case, terrain’s surface variations for instance, can be erroneously characterised as small obstacles.

A way of relaxing the planar terrain assumption without losing computational performance is through the use of heuristics applied locally to the disparity/range image [4], [5], [3]. Another well known heuristic is traversability, which is typically defined in terms of the residual resulting from the fitting process of small planes [6], [7], [8], [9], [10], [11].

The major limitation of heuristic-based solutions is the difficulty in defining obstacles in terms of the robot’s mechanical characteristics, i.e. systematically. A way of circumventing this limitation is through the construction of Digital Elevation Maps (DEM) of the environment, upon which a detailed kinematic and dynamical model of the robot can be used for safe motion planning [12], [13], [14]. However, these solutions tend to be too computationally demanding.

An alternative is to define obstacles in terms of geometrical relationships between their composing 3-D points [15], [16], [17]. From these, the model proposed by Manduchi et al. [16] is the one providing the most exact definition of obstacle. To overcome some of its limitation, i.e. to cope with real-time constraints and noisy data, it has been recently extended [18], [19], [20].

This paper proposes a set of improvements to those extensions, in particular to saliency-based space-variant resolution [20] and voting [19] mechanisms. Experimental results will show that the improved detector, which we name ESaLOD, is more robust, accurate and computationally efficient than its predecessors.

This paper is organised as follows. In Section II the proposed model is exposed. Experimental results, obtained with a stereo vision sensor, are subsequently described in Section III, followed by conclusions and future work in Section IV.

II. PROPOSED MODEL

This section describes the obstacle detector, whose operation is modulated by visual saliency and compensated according to the estimated ground-plane. Saliency computation and ground-plane estimation processes are based in previous work [20] on off-road environments.

A. Obstacle Definition

As proposed by Manduchi et al. [16] in their model, from now on denominated Original Obstacle Detector (OOD), a 3-D point is considered obstacle if it is compatible with any other 3-D point. Two 3-D points, $p_a = (x_a, y_a, z_a)$ and $p_b = (x_b, y_b, z_b)$, are said to be compatible if,

$$H_{\text{min}} < |y_b - y_a| < H_{\text{max}} \land \left|\frac{y_b - y_a}{\|p_b - p_a\|}\right| > \sin \theta \quad (1)$$

Fig. 1. Examples of ESaLOD obstacle detection results. Red overlay on input image correspond to pixels classified as obstacle (within a 10 m range).
where $\theta$ is the minimum slope a surface must have to be considered obstacle, $H_{min}$ is the minimum height an object must have to be considered an obstacle, and $H_{max}$ is the maximum allowed height between two points to be considered compatible with each other. As mentioned, two compatible points pertain to the same obstacle. By transitivity, two points that are linked by a chain of compatible points are also part of the same obstacle.

This definition has an intuitive geometrical interpretation (see Fig. 2). The 3-D points that are compatible with a 3-D point $p$ are all encompassed by two truncated cones $C_U$ and $C_L$ with vertex in $p$, normal to the $xz$ plane and symmetrical to each other, with an aperture angle of $(\pi - 2\theta)$ and limited by $|y| = H_{min}$ and $|y| = H_{max}$.

Checking the compatibility between all possible 3-D points is limited by real-time requirements. Fortunately, only a reduced subset of the pixels must be analysed if the following procedure is followed [16]. Let $p'$ be the projection of the 3-D point $p$ onto the image plane. Accordingly, the two truncated cones of $p$, $C_U$ and $C_L$, project onto two truncated triangles $C'_{U}$ and $C'_{L}$ in the image plane, with vertex in $p'$ (see Fig. 2). Thus, the compatible points of $p$ are limited to those whose projected pixels are encompassed in $C'_{U}$ and $C'_{L}$. Moreover, if the image is scanned from bottom to top and from left to right, it suffices to consider only the upper truncated triangle, $C'_{U}$, to efficiently detect and label all points compatible with $p$.

The truncated cone's height is given by $H_{max}f/p_z$, and its width can be approximated by $2H_{max}f \tan \theta_{max} p_z \cos u'$, where $f$ is the camera’s focal length and $u = \arctan(p_x/p_z)$.

B. Roll Compensation

All the above geometrical considerations assume that the camera is not tilted or rolled in respect to the ground-plane.

This is an obviously unbearable constraint for all-terrain robots. Small variations on the camera’s attitude could be compensated by overestimating the truncated triangle’s size [16]. However, this approximation increases the computational cost, and thus should be discarded.

Instead, a faster and more exact approximation to the problem is herein proposed. First, the dominant plane, assumed to be the ground one, is computed [20]. Then, the full 3-D point cloud is rotated in order to align the world’s reference frame, given by the normal to the computed ground-plane, with the camera’s reference frame [18], [20]. The projected truncated triangle is also rotated according to the estimated camera’s roll angle. This way, the pixels scanned in the image plane better correspond to the 3-D points that are actually encompassed by the rotated truncated cone (see Fig. 3). The tilt angle is not considered in the truncated triangle rotation due to its far less significant impact on detection accuracy.

C. Space-Variant Resolution

Despite the advantages of using a truncated triangle in order to focus the compatibility test, the computational cost of the method remains too expensive. Space-variant resolution is thus essential to further reduce the computational load. A successful extension to OOD [16] with space-variant resolution, from now on denominated of Extended Obstacle Detector (EOD) [19], can be summarised as follows.

For a given pixel $p'$ (sequentially sampled from $1/n$ of the full resolution, from the bottom of the image to its top and from left to right), its $C'_{U}$ is first scanned for compatible points with $1/m$ of the maximum resolution in a chess-like pattern, where $m > n$. If a point compatible with $p'$ is found, $C'_{U}$ is rescanned as with $1/n$ of the maximum resolution; then, finished the scanning procedure, the full resolution is recovered with the following region growing method. For each obstacle pixel $p'$, all its neighbours within a distance $d$ in the image plane, are also labelled obstacles if their corresponding 3-D points are within a distance $g$ from $p$.

Despite its considerable achievements in reducing computational cost, the EOD method [19] operates to a great
extent blindly. That is, in order to reduce computational cost, 
and are increased, and consequently the number of 
skipped pixels as well. Visual saliency is known to be an 
important asset in many search tasks and thus it is a powerful 
candidate to guide the space-variant resolution mechanism in 
an informed way. Bearing this in mind, EOD [19] has been 
extended into a Saliency based Obstacle Detector (SalOD) 
[20], which can be summarised as follows.

Rather than applying the compatibility test along the 
whole scan, as performed by the OOD [16], the SalOD [20] 
proposes to test a pixel , sampled from / of the full 
resolution, if:

1) consecutive pixels in the same row of have 
not been tested so far; or
2) consecutive pixels, after a pixel that has been tested 
and labelled as obstacle in the same row of , have 
not been tested so far; or
3) there is a 10% increment between the local saliency of 
and the one of its preceding scanned pixel, provided 
that both share the same row; or
4) the last scanned pixel had no computed 3-D, and hence, 
no information could be obtained from it.

Local saliency is computed by taking the maximum 
saliency from the set of pixels within the same column of 
and contained in its truncated triangle, including itself. 
This diminishes the effects of poor light conditions, which 
in some situations makes the upper part of objects to appear 
more salient than their lower part. If only the saliency of was 
used instead, many pixels in the lower part of objects 
would be inappropriately skipped.

Roughly speaking, the described process slides along rows 
for pixels unless an increase in saliency is observed, or 
an obstacle point is detected. While sliding, the compatibility 
test is not performed, and consequently computational cost is 
saved and the chances of generating false positives is reduced 
[20]. However, some additional features can be added in 
order to enhance both system’s performance and accuracy. 
These changes are herein proposed as part of an improved 
detector, the Extended SalOD (ESalOD).

First, is sampled from the full resolution input image, 
rather than from only / . As discussed later on (see 
Section II-D) this is an important feature to enable a full 
resolution algorithm. Secondly, rather than having a fixed 
, a dynamical one is used instead,

\[
n_{\text{slide}}(k) = \begin{cases} 
  k \cdot n, & \text{if } (k \leq \frac{n_{\text{max}}}{n}) \\
  n_{\text{max}}, & \text{otherwise}
\end{cases}
\]  

(2)

where, is the number of consecutive pixels tested and 
labelled as non-obstacle, since the last time a pixel was 
considered an obstacle, and is an empirically defined 
scalar. The application of this method results in skipping 
progressively more pixels as obstacles are not found. Since 
the sliding process starts with small jumps, the chances of 
failing to detect the borders of objects are reduced, which 
was one of the problems of SalOD [20].

Additionally, every time a pixel is labelled as obstacle, 
the saliency of all pixels compatible with are increased 
in 10% (empirically defined). This reinforces the presence 
of the detected obstacle, which in turn raises the chances 
of selecting other pixels of the obstacle, i.e. those whose 
saliency has been magnified, for compatibility testing. Typi-

cally, saliency is used to modulate a task-specific detector, 
e.g. (e.g. [21], [22]), as it is the case in the proposed 
method. The other way around, i.e. allowing the results 
of the detector to modulate the saliency map, is much 
more atypical. Thus, the proposed mechanism to magnify 
the saliency when compatible points are found, is itself an 
innovation to saliency computation models.

Finally, instead of analysing every row that is multiple of 
, as in SalOD [20], ESalOD instead skips rows, where 
is incremented every time an analysed row does not contain 
any obstacle pixel. Whenever an obstacle pixel is found 
is zeroed. This procedure, which mimics to some extent the 
row sliding process, is extremely useful in the reduction of 
the computation load in environments with few obstacles, or 
when obstacles are mostly in the far-field (see the effects 
of this process in Fig. 4). Since the truncated triangles for 
points in the near-field are quite large, skipping rows from 
the bottom of the image when no obstacle is found there, 
reduces considerably the computational cost.

**D. Full Resolution Operation**

Note that for the ESalOD case, and define a base 
resolution which does not limit the algorithm from using full-
resolution if necessary. This is the case for instance when 
a saliency increment is observed across two sequentially 
analysed pixels. That is, every pixel in the input image has 
non-null probability of being assessed. In this sense, the 
ESalOD is a full resolution algorithm. In opposition, and 
with the exception of the region growing mechanism, the 
base resolution of EOD [19] and SalOD [20] are only able 
to blindly reduce the amount of pixels being analysed.

**E. Voting Filter**

In addition to computational performance, both accuracy 
and robustness are likewise important. These two additional
ingredients, in the form of voting filters, are integral parts of EOD [19]. These filters intend to diminish the effects caused by artifacts introduced during the 3-D reconstruction process. In this formulation, a given point \( p \) is said to cast a number of votes equal to the number of points compatible with \( p \), and is also said to be voted by those points whose upper truncated cone include \( p \) (see Fig. 2).

Formally, let \( p \) be a 3-D point and \( p' \) its projection in the input image. Let \( A'_p \) be the set of pixels, with computed range, falling inside the upper truncated triangle of \( p' \). Let \( S'_p \) be the set of pixels whose corresponding 3-D points are encompassed by the upper truncated cone of \( p \), and thus compatible with it. Hence, \( |S'_p| \) is said to be the number of votes casted by \( p \). Let \( B'_p \) be the set of pixels, with computed range, whose upper truncated triangles encompass \( p' \). Let \( R'_p \) be the set of pixels, whose corresponding 3-D points’ upper truncated cones encompass \( p \). Hence, \( |R'_p| \) is said to be the number of times \( p \) is voted.

In this formulation, only points that cast more than \( n_{votes} \) votes and are voted by more than \( n_{voted} \) points, are considered obstacles. Thus, rather than the one-to-one mapping considered in the OOD [16], where compatibility is sufficient to define obstacle, in EOD [19] a many-to-many mapping is necessary. This naturally results in higher levels or robustness.

A careful empirical observation revealed that the and operation employed in EOD [19] is too strong, making the system’s output highly sensitive to the voting thresholds parametrisation. As a consequence, we set the ESalOD to use the logical disjunction instead. This slight change allows reducing the false positives rate by pushing further the voting thresholds, with minimum impact on the true positives. Moreover, a mechanism to normalise the number of votes associated to each point, according to the theoretically maximum number of possible votes, i.e. \( |A'_p| \) and \( |B'_p| \), is also missing in EOD [19]. The relevance of this issue stems from the fact that farther obstacles are represented by fewer pixels than closer obstacles.

Both disjunction and normalisation mechanisms are then used to determine whether point \( p \) can be accepted as an obstacle point, according to the following condition,

\[
\left( \frac{|S'_p|}{|A'_p|} > n_{votes} \right) \vee \left( \frac{|R'_p|}{|B'_p|} > n_{voted} \right)
\]

(3)

where for the context of this article both voting thresholds are identical \( v = n_{votes} = n_{voted} \).

**F. Area Filter**

Although the voting filters are extremely powerful, their operation can be better exploited when in conjunction with an area filter, as demonstrated by the experimental results (see Section III). The area filter comes into play to remove any residual noise left by the voting filters.

For the area filter application, the obstacle points are segmented in the 3-D space according to the method employed in OOD [16]. Once all obstacles are segmented, the area filter eliminates those segments, i.e. objects, whose area is below an empirically defined scalar \( a \). Formally, an obstacle point \( p \) is considered non-obstacle if \( |A_p| < (100 \cdot a)/z^2_p \), where \( A_p \) is the set of points with the label of \( p \), i.e. that correspond to the same object.

### III. Experimental Results

A set of 36 stereo 640 × 480 images, published elsewhere [20], along with its corresponding hand-labelled ground-truth (obstacle/non-obstacle), was used throughout all experiments. Images composing the dataset have been acquired with a 9 cm baseline Videre Design STOC sensor, at an approximate height of 1.5 m. Due to poor light conditions and blur induced by motion, the acquired low contrast images generate noisy 3-D point clouds. These are stringent conditions but quite realistic for outdoor robots. Small Vision System (SVS) [23] and OpenCV [24] were used for stereo and low-level computer vision routines, respectively. When nothing is said otherwise, the ESalOD model has been parametrised for the best performance, \( H_{min} = 0.1 \text{ m}, H_{max} = 0.4 \text{ m}, \theta = 40^\circ, \text{ (} n \times m \text{)} = (3 \times 6), a = 25, v = 20, n_{max} = 30, g = 0.4 \text{ m}, d = 8, r_{min} = 1 \text{ m}, r_{max} = 10 \text{ m}, n_{huepo} = 500 \). Additional parameters for saliency computation and ground plane estimation have been set to their default values [20].

The area filter, which aims at removing false positives, showed to be pivotal for the overall improvement (see Fig. 5). The Receiver Operating Characteristic (ROC) curves for different thresholds reveal that \( a = 25 \) is the one producing better results, and consequently considered for the best parametrisation.

The synergistic effect between both area and voting filters can be appreciated in Fig. 6, which illustrates the ROC curves of the several tested models when varying the voting thresholds. The curves also clearly show the proposed ESalOD model outperforming its preceding models in both accuracy and robustness. In fact, the curves show that ESalOD reduces in \( \approx 70\% \) the FPR of OOD, with only \( \approx 10\% \) of reduction in terms of TPR.

Finally, Fig. 7 shows that the added complexity of ESalOD, which endows it with higher levels of accuracy and robustness than its predecessors EOD [19] and SalOD [20], while performing in full resolution, does not compromise its performance. An additional experiment showed that the saliency-based slide mechanism is responsible for 17% saving of computation time. On a Centrino Dual Core 2.0 GHz CPU running Linux, stereo, saliency, and ground-plane estimate computation, take in average, 40 ms, 43 ms and 54 ms, respectively.

The original obstacle detector (OOD) performs on average 110 times slower than ESalOD. The timing information reported by Manduchi et. al [16] varies between 0.67 s and 4 s (after conversion to 640×480 images). This is rather different from the 40 s (on average) obtained with our implementation of their algorithm. We believe this is mostly due to the fact that most of our images cover larger areas of the near-field. In these situations the upper truncated triangle is quite large and consequently expensive to analyse. This happens because
Our images were taken from a lower height, and many of them with large tilt angles pointing downwards. Note that all tested algorithms are built from the same backbone, i.e. our OOD implementation. This reduces to the minimum any bias that could bias any of the models. Fig. 8 illustrates the product of ESalOD, with its best parametrisation, on three typical images.

IV. CONCLUSIONS

A saliency-based obstacle detector for all-terrain environments, denominated of ESalOD, was presented. Experimental results showed that the detector is more robust, accurate and computationally efficient than the models on which it is based on, i.e. OOD [16], EOD [19] and SalOD [20].

The obtained improvements are due to a set of factors. First, the adaptive (progressive) way saliency is accounted for during the detection process allows the system to better skip uninteresting pixels without failing to detect obstacles’ boundaries. Second, the fact that the detector’s positive results are used to boost the saliency map itself, which is innovative in terms of saliency models, reinforces the analysis of relevant regions of the image. This helps for instance on dealing with local saliency failures. The improvements in terms of detection invariance to robot’s posture have shown to be essential for a proper operation in highly roughed terrain. Finally, the way 3-D points are allowed to vote on each other, normalised by the distance, increases robustness considerably. All these results contribute to the evidence that, visual saliency and voting mechanisms are key to enable both parsimonious and robust all-terrain obstacle detection.

Quantisation for extensive use of look-up tables is known to reduce considerably detection computation time [18].

The advantages of introducing such technique, which is complementary to the ones proposed in this paper, will be subject of future work. Special attention will be given to the unavoidably loss of accuracy.

The saliency map is used to guide the application of the obstacle detector, which is in turn modulating the former as obstacles are detected. This mutual influence, which is innovative, will be pushed further in subsequent work. In particular, the output of the obstacle detector will be used to determine which feature maps should contribute more to the saliency map [25], and as a consequence improve the correlation between saliency and obstacle presence.

V. ACKNOWLEDGEMENTS

This work was partially supported by FCT/MCTES grant No. SFRH/BD/27305/2006.

REFERENCES

Fig. 7. Computation time comparison. Each bar corresponds to the average computation time over all images in the dataset, for a given configuration. Error bars correspond to their standard deviation. All configurations operate on resolution \( (3 \times 6) \). EOD runs with votes in their best parametrization, i.e. \( v = 7 \), space-variant resolution, and without its additional morphological filters \([19]\). ESalOD runs both with voting and space-variant resolution mechanism on, parametrised for the best, i.e. \( v = 20 \) and \( a = 25 \). SalOD runs with space-variant resolution and differently from its original implementation \([20]\), for the sake of a comparison, with voting mechanisms configured as for the ESalOD case. Ground-plane compensation is turned on in all cases, despite not originally considered in EOD.


