Neural Networks for Real-Time, Probabilistic Obstacle Detection

Tobias Werner, Josua Bloeß, and Dominik Henrich

Lehrstuhl für Robotik und Eingebettete Systeme,
Universität Bayreuth, D-95440 Bayreuth, Germany,
tobias.werner@uni-bayreuth.de,
http://robotics.uni-bayreuth.de

Abstract. Recent research suggests intrinsically safe robots, such as through soft limbs or artificial skins, to enable close-quarter human-robot collaboration. Intrinsically safe robots allow for risk-minimized instead of collision-free path planning. Risk-minimized path planning can integrate non-binary knowledge — including obstacle probabilities, robot speed, or data age — into the choice of a robot path. In this contribution, we propose a novel approach to probabilistic obstacle detection on color images that is specifically suited for use in real-time risk-minimized path planning. Our approach enhances an existing neural network for object detection by incorporating spatial coherence via a second neural network and an optimization step inspired by simulated annealing. Finally, a bias towards false-positive obstacle detection allows us to avoid the Sleeping Person Problem for online learning. In our experiments, we show that a GPGPU implementation of our approach can process Full HD images at a soft real-time rate of 15 Hz. We conclude that our probabilistic obstacle detection is fit for use in real-time risk-minimized path planning.

Keywords: neural networks, probabilistic obstacle detection, real-time risk-minimized path planning

1 Introduction

Human-robot collaboration has prospects past traditional industrial automation, including promising use cases in small businesses, the service sector, or private homes. For such use cases, recent research advocates intrinsically safe robots, as made possible by artifical skins, force control, or soft limbs. Paths for intrinsically safe robots need not remain collision-free at all costs. Instead, real-time, risk-minimized path planning can integrate and weigh non-binary knowledge from various sources — including obstacle probabilities, sensor errors, or data age.

Our overall goal is to enable real-time, risk-minimized path planning for a robot manipulator. To this end, we monitor the robot workspace through a multi-camera network made from inexpensive, consumer-grade webcams. In this contribution, we present a novel approach to derive probabilities of obstacle presence from individual color images captured on the multi-camera network in real-time. Our approach follows a two-stage structure: In the first stage, a neural network performs per-pixel probabilistic foreground-background segmentation. In the second stage, another neural network improves obstacle detection results by incorporating spatial coherence through a gradient descent inspired by Simulated Annealing. A modified online learning behaviour avoids the infamous Sleeping Person Problem through a bias for false-positive obstacle detection.

The remainder of our contribution is structured as such: Section 2 reviews work related to our approach, including existing algorithms for probabilistic object detection. In Section 3, we discuss our approach to probabilistic obstacle detection, with one subsection for each of the two neural network stages, whereas a third subsection presents our modified learning method. Section 4 gives our experimental results and compares performance and memory readings of our implementation to related work. Section 5 concludes our contribution with a review and notes on future work.

2 Related Work

Research proposes manifold behavioural and reactional strategies for robots in the presence of a-priori unknown obstacles (e.g. speed control and path planning [6]). Each such strategy requires a specific type of environment representation (e.g. point clouds [23], voxel spaces [17]). The environment representations in turn originate from individual sensors (e.g. artificial skins [19], depth cameras [12], color cameras [7]), or from fusion of data over multiple sensors (e.g. multi-camera systems [15] that generate visual hulls [11] or photo hulls [3]).

Throughout perception and environment modeling, there are two distinct variants: binary object detection (e.g. [9]) and probabilistic object detection (e.g. [16]). The former variant is more intuitive and allows for a more efficient interpretation of resulting data in subsequent algorithms such as path planning or speed control. The latter variant enables a more comprehensive integration of sensor errors, system latencies, and risk-based assessment into path planning (e.g. [8], [1]). We therefore consider probabilistic object detection as vital to achieve our end goal of risk-minimized path planning.

Related work suggests diverse approaches to finding probability images from color images: Frame difference algorithms (e.g. [10]) compare current with previous input images and compute their absolute differences, but fail for static or abruptly stopping objects. Statistical algorithms adapt a probability density function to the background statistics of images and use these statistics to classify incoming pixels. Mixture of Gaussian Modeling (see [2]) is a popular algorithm in this category. Alternative strategies apply Sugeno and Choquet Integrals to measure the similarity between an incoming and a background image based on texture features and color (see [18]), or use self-organizing maps for background subtraction through a fuzzy function for learning (see [14]). In contrast to related approaches, our contribution does not require an extensive learning phase, and can adapt to gradual changes in the environment without the usual shortcomings of unsupervised online learning.

3 Obstacle Detection on Color Images

In this section, we present our two-stage approach to real-time, probabilistic obstacle detection on color images. See Figure 1 for an overview over both stages and their dependencies.

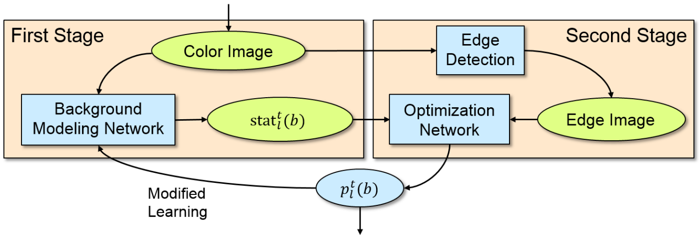

Fig. 1. Overview of the two stages in our approach. The overall input is a color image, the output is a per-pixel background probability $p^t_l(b)$.

3.1 Neural Network for Background Modeling

Our first stage utilizes a Background Modeling Neural Network (BNN) to perform a preliminary segmentation through color statistics of individual pixels. As our implementation closely follows the approach in [4], we only give a short overview over our BNN and refer to the original publication for details.

Our BNN implements a probability density function $p^t_l(\Theta_l|V)$ (PDF) for a classification $\Theta_l \in \{f, b\}$ at location l, where f stands for *foreground*, b for *background*. The BNN reconstructs $p^t_l(\Theta_l|V)$ based on $p^t_l(\Theta_l|v^t_{i,l})$ for a small number of color prototypes $\{v^t_{1,l}, ..., v^t_{k,l}\}$. The BNN then computes the PDF as

$$p^t_l(\Theta_l|v) = \frac{1}{k}\sum_{i=1}^{k} p^t_l(\Theta_l|v^t_{i,l}) \cdot \mathrm{act}^t_{i,l}(v), \qquad v \in V \tag{1}$$

where $\mathrm{act}^t_{i,l}(v)$ is the activation of color prototype v_i — a similarity measure of input values v and v_i — derived via textbook Parzen Estimation. Applying the PDF to the observed value v^t_l for timestep t gives a preliminary segmentation $\mathrm{stat}^t_l(b) = p^t_l(b|v^t_k)$. Since $\forall v, l, t : p^t_l(f|v) = 1 - p^t_l(b|v)$, each $p^t_l(\Theta_l|v^t_{i,l})$ can be stored as a single scalar value $w^t_{i,l} = p^t_l(b|v^t_{i,l})$ per pixel location l. For a sequence of input values $(v^t)_t$ over timesteps t, the BNN of each pixel adapts its weights w^t_i (and thus its PDF) according to

$$w^{t+1}_{i,l} = \mathrm{clamp}\left(\left(1 - \frac{\beta}{k}\right) \cdot w^t_{i,l} + \delta(i, i^t_{\mathrm{max},l}) \cdot \beta\right), \tag{2}$$

with the *Kronecker* δ, a learning parameter β, and $i^t_{\mathrm{max},l} := \mathrm{argmax}_i(\mathrm{act}^t_{i,l}(v^t_l))$. Color prototypes are updated if $\mathrm{act}_{i^t_{\mathrm{max},l}}(v^t_l) < \theta_{\mathrm{act}}$, i.e. if the BNN is inactive (with treshold θ_{act}) for an input v_l^t. The color prototype $v_{i,l}^t$ which minimizes the criterion function

$$\text{crit}_l^t(i) = p_l^t(f|v_{i,l}^t) + |p_l^t(b|v_{i,l}^t) - p_l^t(f|v_{i,l}^t)| = 1 - w_{i,l}^t + |2w_{i,l}^t - 1| \quad (3)$$

is then replaced by the current input v_l^t.

The above state of art approach models per-pixel statistics and performs probabilistic segmentation. However, this approach does not integrate spatial coherence, as all learning and classification is done independently per pixel.

3.2 Neural Network for Spatial Coherence

In the subsequent second stage, we apply a novel strategy to incorporate spatial coherence into probabilistic obstacle detection: Homogenous regions in the color image should remain homogenous in the final probabilistic segmentation. This gives a positive bias to spatially coherent obstacles, which corresponds to typical real world objects. To integrate spatial coherence, a preprocessing step employs a modified Sobel Operator per color channel of an incoming image to obtain edge images G_R, G_G, G_B. We then combine per-channel edge images to a final image $G = \sqrt{G_R^2 + G_G^2 + G_B^2}$ with per-pixel arithmetics. Figure 2 shows example results of edge detection.

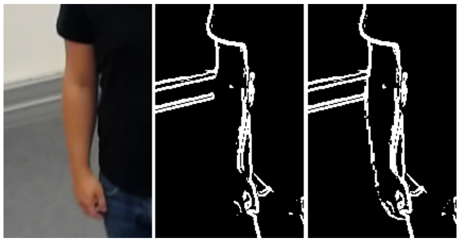

Fig. 2. Left to right: Input color image, output of a basic Sobel Filter, output of our modified Sobel Filter with improved edge detection.

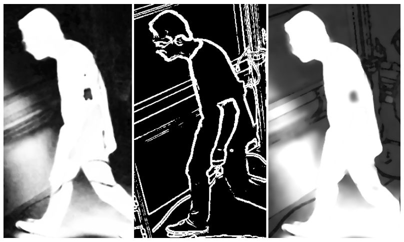

Fig. 3. Left to right: per-pixel probabilistic segmentation after first stage, edge image, spatially optimized segmentation with reduced artifacts.

Subsequently, we formulate a neural network that encodes our edge image. In particular, let network N be a set of neurons, let $s^t : N \to [0,1]$ be the state of a neuron at timestep t, let $N_{\text{stat}} \subset N$ be the subset of all static neurons ($n \in N_{\text{stat}} \implies \exists c \forall t : s^t(n) = c$), let $N_{\text{var}} = N \setminus N_{\text{stat}}$ be the subset of all variable neurons and let $l : N_{\text{var}} \to L$ be the location function that maps neurons to their corresponding location in the pixel space. The resulting network represents an energy function which we now optimize with a deterministic variant of Simulated Annealing to maintain spatial coherence in the final probabilistic segmentation. Deterministic annealing is an iterative technique that changes the states of variable neurons N_{var} to converge towards a local minimum of an energy function. Static neurons N_{stat} serve as constant input constraints. The general energy function in our approach has the form

$$E(N) = \sum_{n \in N_{\mathrm{var}}} \sum_{m \in N} D_{(m,n)} |s(n) - s(m)|^2, \tag{4}$$

where $D_{(m,n)} \geq 0$ encodes the relation of two neurons m and n. Given a monotonically decreasing temperature function decrease(T), a virtual force $F : [0,1] \times \mathbb{R}^2 \to [0,1],\ F(s,f,T) = s + f \cdot T$, influences the state of a neuron. As f is directly proportional to the energy gradient of neuron n, we can reformulate our algorithm as a local gradient descent. This deterministic approach is susceptible to get trapped in local minima. However, we have found a good-guess initialization stat$^t_l(b)$ to be sufficient, whereas a stochastic F (e.g. as in [13]) was not necessary. Algorithm 1 provides pseudo code for the annealing process.

Algorithm 1 Deterministic Annealing

```
procedure ANNEALING($E, N$)                              ▷ Energy function $E$ and network $N$
    $T \leftarrow T_0$                                   ▷ Initialize temperature with $T_0$
    for $k \leftarrow 0$ to $k_{\max}$ do
        for all $n \in N_{\mathrm{var}}$ do
            $f \leftarrow \sum_{m \in N} D_{(m,n)} |s(n) - s(m)|$        ▷ Compute ”force”
            $s(n) \leftarrow F(s(n), f, T)$                ▷ Update the state of $n$
        end for
        $T \leftarrow$ decrease$(T)$
    end for
end procedure
```

The specific energy function we want to minimize has the form

$$E(N) = w_{\mathrm{stat}} \cdot E_{\mathrm{stat}}(N) + w_{\mathrm{spat}} \cdot E_{\mathrm{spat}}(N), \tag{5}$$

where E_{stat} and E_{spat} respectively encode statistical segmentation and spatial information of the edge image. Weights w_{stat} and w_{spat} control individual energy influence on optimization. Energy for statistical segmentation calculates as

$$E_{\mathrm{stat}}(N) = \sum_{n \in N_{\mathrm{var}}} |s(n) - S(\mathrm{l}(n))|^2, \tag{6}$$

while spatial energy is more involved,

$$E_{\mathrm{spat}}(N) = \sum_{n \in N_{\mathrm{var}}} \sum_{m \in N} D_{\mathrm{edge}}(m,n) \cdot |s(n) - s(m)|^2, \tag{7}$$

where the edge image influences homogenity in the final result through

$$D_{\mathrm{edge}}(m,n) = \begin{cases} 1, & \text{if } m \in V(n) \text{ and } \min\left(G(\mathrm{l}(n)), G(\mathrm{l}(m))\right) < \theta_{\mathrm{edge}}, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Here, θ_{edge} is a threshold for binarizing G, and $V(n)$ returns all coupled neighbors of a neuron n. The latter may be decoupled through edges in the color image. Figure 3 shows the influence of spatial coherence on overall obstacle detection.

3.3 Conditional Learning

Online learning in the background subtraction stage incurs the Sleeping Person Problem, where a long-term static obstacle fades into background [20]. Notably, the update rule in Equation 2 increases $p_l^t(b|v_{i,l}^t)$ for a color prototype $v_{i,l}^t$ without regards to pixel classification. To solve this problem, we substitute the learning rate β with a modified learning rate $\tilde{\beta} = \beta \cdot p_l^t(b|v_l^t)$ that incorporates the results of the final image segmentation. A new update rule follows,

$$w_{i,l}^{t+1} = \text{clamp}\left(1 - \frac{\tilde{\beta}}{k} \cdot w_{i,l}^t + \delta(i, i_{\text{max},l}^t) \cdot \tilde{\beta}\right), \tag{9}$$

where the influence of a pixel on learning directly depends on its probability to classify as background. Consequently we also have to revise the replacement of color prototypes: We only replace a prototype if the respective pixel exceeds a background probability ($p_l^t(b|v_l^t) < \theta_{\text{replace}}$). This modification prohibits learning from adapting to sudden background changes, but preserves background statistics for pixels occluded by foreground objects (see Figure 4).

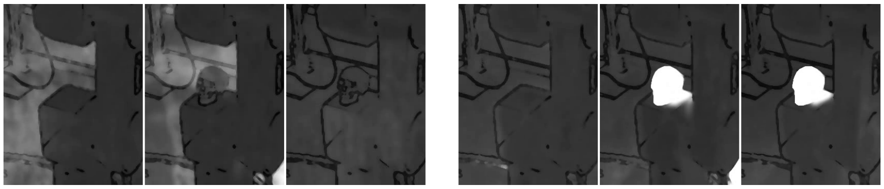

Fig. 4. Three frames from an experiment sequence. An object is introduced between the first two frames and remains static. Left: Traditional learning, the object fades to background. Right: Conditional learning, the object remains foreground. The robot manipulator is known a-priori and can be suppressed by post-processing.

4 Experiments

We have evaluated our approach through a variety of video sequences from an example robot workcell. Over all video sequences, the following parameter choices gave best results: $\beta = 0.1$, neuron activation threshold $\theta = 0.9$, $w_{\text{init}} = 0.1$, prototype count $k = 10$, annealing steps $k_{\text{max}} = 20$, $w_{\text{stat}} = 0.8$, $w_{\text{spat}} = 0.4$, $\theta_{\text{edge}} = 0.1$, and $T_0 = 1.0$.

At implementation level, we use a GPU for efficient parallelization over pixels of the input images. Further parallelization would be possible (e.g. per prototype), but we already achieve full load on our target NVIDIA GTX1070 GPU. Figure 5 provides timings for our experiments. Notably, we maintain a steady 15 Hz update rate over all our test sequences. The memory footprint of both neural networks does not exceed 1 GB of RAM and remains well within GPU limits. For further reference, Table 1 shows benchmarks of related approaches on their respective hardware.

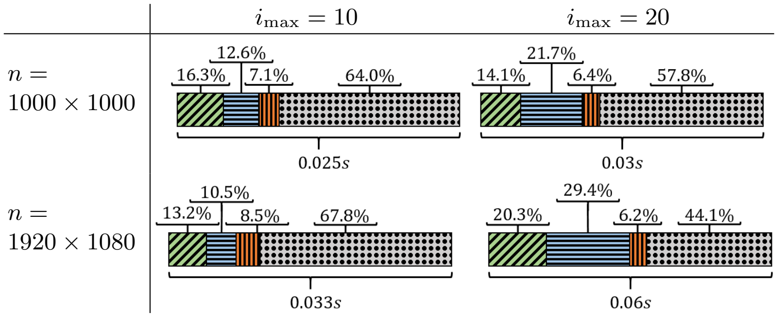

Fig. 5. Experimental timings at different image resolutions. From left to right per bar: BNN learning and classification, optimization, Sobel filter, and system overhead.

Algorithm	Framerate	Image Resolution	Hardware
[5]	10.5fps	720 × 576	GTS 250 (2009)
[2]	11fps	240 × 360	Intel Pentium Duo (2010, no GPU)
our contribution	15fps	1920 × 1080	GTX 1070 (2016)

Table 1. Runtime comparison. Here, algorithm [5] is a GPU variant of [4], the BNN we used, while [2] is a CPU variant of a Mixture of Gaussian model.

5 Conclusion

In the preceding, we have contributed a novel approach to probabilistic obstacle detection on color images. In contrast to existing approaches, we consider spatial coherence through an edge image and we introduce conditional learning to avoid the Sleeping Person Problem. Experiments show that our approach notably exceeds existing approaches in performance and quality. Our implementation can process Full HD color images at a 15 Hz rate, and thus is suited for soft real-time applications, including risk-based online path planning for robot manipulators. In future work, we plan to integrate probabilistic segmentation results into a probabilistic variant of a 3D environment reconstruction [21], which in turn is a suitable input for our risk-minimized path planner [22].

6 Acknowledgements

This work has partly been supported by the Deutsche Forschungsgemeinschaft (DFG) under grant agreement He2696/11 SIMERO.

References

1. L. Blackmore et al.: *A probabilistic approach to optimal robust path planning with obstacles*, American Control Conference, 2006.
2. T. Bouwmans, F. El Baf, B. Vachon: *Statistical background modeling for foreground detection: a survey*, Handbook of Pattern Recognition and Computer Vision, vol. 4, World Scientific Publishing, 2010.
3. A. Broadhurst, R. Cipolla: *A Statistical Consistency Check for the Space Carving Algorithm*, BMVC, 2000.
4. D. Culibrk et al.: *Neural Network Approach to Background Modeling for Video Object Segmentation*, IEEE Transactions on Neural Networks, 2007.
5. D. Culibrk, V. Crnojevi: *GPU-Based Complex-Background Segmentation Using Neural Networks*, The Irish Machine Vision and Image Processing, 2010.
6. T. Gecks: *Sensorbasierte, echtzeitfähige Online-Bahnplanung für die Mensch-Roboter-Koexistenz*, PhD thesis, Universität Bayreuth, 2011.
7. S. Kuhn: *Multi-view reconstruction in-between known environments*, Technical report, Univ. Bayreuth, 2010.
8. B. Lacevic, P. Rocco: *Towards a complete safe path planning for robotic manipulators*, Intelligent Robots and Systems, 2010.
9. A. Ladikos, S. Benhimane, N. Navab: *Efficient Visual Hull Computation for Real-Time 3D Reconstruction using CUDA*, CVPR Workshops, 2008.
10. A. Lai, N. Yung: *A fast and accurate scoreboard algorithm for estimating stationary backgrounds in an image sequence*, Symposium on Circuits and Systems, 1998.
11. A. Laurentini: *The Visual Hull Concept for Silhouette-Based Image Understanding*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994.
12. C. Lenz et al.: *Fusing multiple kinects to survey shared human-robot workspaces*, Technical Report TUM-I1214, Technische Universität München, 2012.
13. M. Locatelli: *Simulated Annealing Algorithms for Continuous Global Optimization: Convergence Conditions*, Journal of Optimization Theory and Applications, 2000.
14. L. Maddalena et al.: *A fuzzy spatial coherence-based approach to background foreground separation for moving object detection*, Neural Comput. 19 (2), 2010.
15. A. Ober-Gecks, M. Hänel, T. Werner, D. Henrich: *Fast multi-camera reconstruction and surveillance with human tracking and optimized camera configurations*, International Symposium on Robotics, 2014.
16. J. Salvador, J. R. Casas: *Shape from Probability Maps with Image-Adapted Voxelization*, Workshop on Multi-camera and Multi-modal Sensor Fusion, 2008.
17. D. Stengel, T. Wiedemann, B. Vogel-Heuser: *Efficient 3d voxel reconstruction of human shape within robotic work cells*, Mechatronics and Automation, 2012.
18. M. Sugeno: *Theory of fuzzy integrals and its applications*, Ph.D. thesis, Tokyo Institute of Technology, 1974.
19. J. Ulmen, M. R. Cutkosky: *A robust, low-cost and low-noise artificial skin for human-friendly robots*, ICRA, 2010.
20. K. Toyama et al.: *Wallflower: principles and practice of background maintenance*, IEEE International Conference on Computer Vision, 1999.
21. T. Werner, D. Henrich: *Efficient and Precise Multi-Camera Reconstruction*, International Conference on Distributed Smart Cameras, 2014.
22. T. Werner, D. Henrich, D. Riedelbauch: *Design and Evaluation of a Multi-Agent Software Architecture for Risk-Minimized Path Planning in Human-Robot Workcells*, Zweiter Kongress Montage Handhabung Industrieroboter, 2017.
23. K. M. Wurm et al.: *OctoMap: A Probabilistic, Flexible, and Compact 3D Map Representation for Robotic Systems*, Conference on Robotics and Automation, 2010