The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Can't Take my Eye off You: Attention-Driven Monocular Obstacle Detection and 3D Mapping

E. Einhorn, Ch. Schröter and H.M. Gross
Neuroinformatics and Cognitive Robotics Lab, Ilmenau University of Technology, Germany

***Abstract*— Robust and reliable obstacle detection is an important capability for mobile robots. In our previous works we have presented an approach for visual obstacle detection based on feature based monocular scene-reconstruction. Most existing feature-based approaches for visual SLAM and scene reconstruction select their features uniformly over the whole image based on visual saliency only. In this paper we present a novel attention-driven approach that guides the feature selection to image areas that provide the most information for mapping and obstacle detection. Therefore, we present an information theoretic derivation of the expected information gain that results from the selection of new image features. Additionally, we present a method for building a volumetric representation of the robots environment in terms of an occpancy voxel map. The voxel map provides top-down information that is needed for computing the expected information gain. We show that our approach for guided feature selection improves the quality of the created voxel maps and improves the obstacle detection by reducing the risk of missing obstacles.**

Keywords: ***visual attention, shape-from-motion, visual obstacle detection, EKF, voxel mapping***

I. Introduction and Related Work

Nowadays, mobile robots find their way into more and more sectors of our daily life. However, when operating in public environments, such as shopping centers or home improvement stores [1], or home environments [2] a large variety of different obstacles must be detected by an autonomous robot.

For obstacle detection most robots are equipped with sonar sensors and laser range finders. Using these sensors, most of the obstacles can be reliably detected. However, obstacles whose maximum extent is mainly located above or below the plane covered by the laser range finder and sonar sensors are difficult to perceive. Therefore, it is necessary to use additional methods for robust and reliable obstacle detection. Vision-based approaches are suitable for this purpose since they provide a large field of view and supply a large amount of information about the structure of the local surroundings.

Beside stereo vision [3] and time-of-flight cameras [4], monocular approaches are an adequate alternative for obstacle detection. The majority of such approaches use feature-based techniques that reconstruct the depth or the entire 3D position of each feature. In our previous works [5], [2], we propose such an algorithm for monocular scene reconstruction and obstacle detection. Our shape-from-motion approach uses Extended Kalman Filters (EKF) to reconstruct the 3D position of the image features in real-time in order to identify potential obstacles in the reconstructed scene.

Other authors [6], [7], [8] use similar approaches for visual SLAM. In contrast to our approach they are mainly focusing on recovering the 3D trajectory of a monocular camera, while a precise reconstruction of the scenery is less important. Our priorities are vice versa as we want to use the reconstructed scene for obstacle detection and local map building.

For feature detection and feature selection most researches apply interest operators like the Shi-Tomasi corner detector, the Harris corner detector or its scale-invariant enhancement, the Harris-Laplacian detector. These detectors provide a bottom-up feature selection scheme where the position and number of the chosen features depend on the content of the input images. However, taking top-down knowledge into account could lead to better results by choosing features in image regions that result in the largest information gain for the environment knowledge instead of choosing the features based on the information content of the images only.

In [9] and [10] such a top-down approach is presented, not for feature selection, but for feature tracking using an improved active search strategy. In [11] the authors present a visual SLAM approach for hand-held cameras that instructs the user to perform position and orientation changes of the camera to optimize the localisation. The actions and movements are chosen so as to maximize the mutual information gain between posterior states and measurements.

Another active vision approach is presented by Frintrop et al. [12], where the camera is controlled by an active gaze control module according to three behaviours for redetection of known features, tracking of features and detection of new features in unknown areas. Using a predesigned decision tree, the system switches between these behaviours depending on the number and expected location of known features.

In summary, these visual SLAM algorithms use the active vision approach basically for controlling the camera's viewing direction in a way to improve the camera's position estimates and to enhance loop closings.

In this paper, we present a different active vision approach that is focusing on feature selection. In contrast to the aforementioned publications, we use a fixed camera with a wide-angle lense whose viewing direction cannot be altered dynamically. However, instead of moving the whole camera, we can choose particular image regions that our algorithm pays more attention to. By combining bottom-up and top-down information we select features in those image regions that provide the highest information gain for the obstacle detection algorithm. By choosing new features at the right places, we can detect more obstacles, allowing us to reduce the total number of reconstructed features without increasing the risk of missing obstacles. This results in an improved performance of the whole obstacle detection algorithm.

In the next section, we give a brief overview of our monocular obstacle detection algorithm. In section III we

The research leading to these results has received funding from the European Community's Seventh Framework Programme ([FP7/2007-2013] [FP7/2007-2011]) under grant agreement #216487

describe an algorithm for building a volumetric 3D-map, that is fundamental for the main contribution of this paper, a novel attention-driven approach for feature selection, that is described in section IV. Finally, we present experimental results and conclude with an outlook for future work.

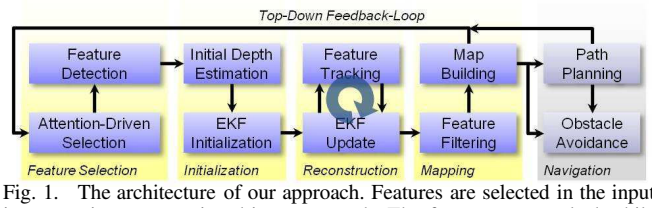

Fig. 1. The architecture of our approach. Features are selected in the input images using an attention-driven approach. The features are tracked while reconstructing their 3D positions using EKFs. The resulting point cloud is used for building a voxel map that is used for navigation tasks and provides information for the attention-driven feature selection.

II. Monocular Scene Reconstruction

As stated before, we use a single calibrated camera that is mounted in front of the robot. During the robot's locomotion, the camera is capturing a sequence of images $I_{1:t} = (I_1, \ldots, I_t)$ that are rectified immediately according to the intrinsic camera parameters in order to correct the lens distortions. Using the robot's odometry, we can obtain the position and the orientation of the camera and therefore its projection matrix for each captured image. This preprocessing step yields different two-dimensional views I_t of a scene and the projection matrices $\mathbf{P}_t$ of the corresponding ideal pinhole camera. The projection matrices fully describe the camera including its position and orientation in the world coordinate frame. For scene reconstruction we use a feature based approach. Distinctive image points (image features) x'_j are detected in the preprocessed input images and tracked over the acquired image sequence while the 3D positions $\mathbf{x}_j$ of these features are estimated using EKFs (see Fig. 1).

Since we require a dense reconstruction of the scene for obstacle detection, we have to cope with a large number of features, which cannot be handled by a full covariance SLAM approach in real-time. Therefore, we use one EKF per feature to recover the scene structure. Each feature j is associated with a state vector $\mathbf{x}_j$ that represents its 3D position, and a corresponding covariance matrix $\mathbf{\Sigma}_j$.

A. Feature Selection and State Initialization

For selecting appropriate features in the captured images we use a feature detector guided by our attention-driven selection approach that is described in section IV in detail.

For newly selected features, the 3D positions, i.e. their corresponding EKF states, must be initialized using a suitable a priori estimate. For this purpose various methods have been proposed in related literature. A simple method for initializing the features is to choose their depths in a way that the height of the initial feature position is zero, i.e. the features are initialized on the ground plane. This kind of initialization has certain advantages when used for obstacle detection since false positive detections are reduced. Using this method, we achieved good results for obstacle detection, although it leads to high initial model errors, since many points are initialized at too large depths.

In [5] we introduced a more complex method which uses a classic multi-baseline stereo approach for initializing new features. The approach uses the images that were captured *before* the features were first detected and therefore can immediately obtain valid depth estimates for newly detected features. Hence, such an hybrid approach can react quickly to obstacles that suddenly appear in front of the robot.

B. Feature Tracking

While the robot is moving, previously selected image features are tracked in subsequent frames. In previous works, we experimented with different tracking algorithms. Besides SIFT descriptors, we applied bipartite graph matching in [5] and a guided search using image patches combined with an advanced descriptor update scheme in [2]. Since the focus of this paper is on feature selection, we use the well known Kanade-Lucas-Tomasi tracker so that our results can be reproduced easier.

After the features are tracked, their 3D positions are updated using the common EKF update equations leading to a more precise reconstruction of the scenery. This step is straightforward and is described in [7] and [5] in more detail.

III. Map Building and Obstacle Detection

For obstacle detection, we perform the described monocular scene reconstruction for 200-300 salient features of the scene simultaneously. Before the reconstructed features are used to build a representation of the environment, they have to undergo some post-processing where unreliable estimates are removed. From all features that were tracked in the current frame, we only use those that meet the following two criteria: First of all, the covariance $\mathbf{\Sigma}_i$ of the estimated 3D point must be below a certain threshold [2]. Besides, the estimated distance to the camera must be smaller than 3 m. This removes most virtual features that arise where the boundaries of foreground and background objects intersect in the image. These features do not correspond to a single 3D point in the scene and cannot be estimated properly.

The features that pass the above filters are inserted into a three-dimensional volumetric model of the environment. Similar to 2D occupancy grid maps, we partition the robot's surrounding three-dimensional volume $V = \{v_i\}$ into disjoint cube-shaped 3D cells (voxels) v_i. Each voxel v_i is associated with an occupancy value $p(v_i)$ which specifies the probability of the volume covered by the voxel being occupied by an obstacle. Similar to 2D occupancy grid maps a voxel is either fully occupied or free, partially occupied voxels are not considered. The voxel map is modeled as a Markov Random Field (MRF) of order 0, where the state of each individual voxel can be estimated as an independent random variable.

At the beginning, all voxels are initialized with a probability of 0.5 to indicate that nothing is known about the robot's environment. After a new frame I_t has been processed, the voxel map is updated using the estimated features in a similar way as laser and sonar range scans are inserted into a 2D occupancy grid map. We use each reconstructed 3D point as a single measurement that is described by the tuple $z_j = (\mathbf{x}_j, \mathbf{\Sigma}_j, \mathbf{P}_t)$ consiting of the estimated 3D position $\mathbf{x}_j$ of the feature, its corresponding error covariance matrix $\mathbf{\Sigma}_j$ and the current camera projection matrix $\mathbf{P}_t$. For each measurement z_j the new occupancy probability $p(v_i|z_{1:j})$ of each voxel v_i can be updated recursively from its previous

value $p(v_i|z_{1:j-1})$ (which only takes previous measurements $z_{1:j-1}$ into account) using Bayes rule [13] as follows:

$$p(v_i|z_{1:j}) = 1 - \left[1 + \frac{p(v_i|z_{1:t-1})}{1-p(v_i|z_{1:t-1})} \frac{p(v_i|z_j)}{1-p(v_i|z_j)}\right]^{-1} \quad (1)$$

where $p(v_i|z_j)$ denotes the *inverse sensor model*, which we will describe in the following. Each measurement $z_j = (\mathbf{x}_j, \mathbf{\Sigma}_j, \mathbf{P}_t)$ is considered to be a range measurement along the viewing ray $\mathrm{r}(\mathbf{P}_t, \mathbf{x}_j)$ that is defined by the position of the camera center taken from the camera's projection matrix $\mathbf{P}_t$ and the estimated 3D position $\mathbf{x}_j$ of the feature. The feature's position is estimated during the scene reconstruction in terms of a trivariate normal probability distribution that is defined by the mean value $\mathbf{x}_j$ and the corresponding covariance matrix $\mathbf{\Sigma}_j$. As seen in Fig. 2 the position uncertainty of a reconstructed point is largest in the depth direction, i.e. along the ray, while the uncertainty orthogonal to the ray is minor and smaller than the width of a voxel. Therefore, it is a sufficiently good approximation to update only the voxels along the ray when inserting the measurement into the voxel map.

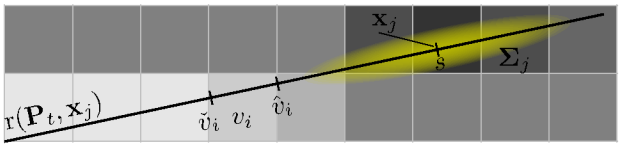

Fig. 2. Each feature x_j is inserted into the voxel map by updating the voxels along the ray $\mathrm{r}(\mathbf{P}_t, \mathbf{x}_j)$ according to an inverse sensor model by taking the error covariance matrix $\mathbf{P}_t$ of the reconstructed feature into account.

This allows to apply a one-dimensional sensor model where we take the marginal probability of the trivariate normal distribution along the ray into account. On this ray s denotes the distance of the reconstructed point $\mathbf{x}_i$ to the camera which is located at its origin. Let us first assume that our measurement is free from errors. Then we can update the voxels along the ray according to the ideal sensor model, where the occupancy value for the voxel that contains the estimated point $\mathbf{x}_j$ is set to $p_{occ} = 1.0$ since it is occupied by the surface of an object where the feature is located on. The occupancy values of voxels that are located on the ray in the line-of-sight between the camera and the estimated 3D point are set to $p_{free} = 0.0$, since these voxels are free - otherwise the feature had not been visible to the camera. The state of voxels that lie 'behind' the estimated feature position is unknown since they can not be observed. The characteristic of this ideal sensor model $p_{ideal}(v_i|z_j)$ is shown in Fig. 3 and can be described formally as follows:

$$p_{ideal}(v_i|z_j) = \begin{cases} p_{free} & \check{v}_i < s \\ p_{occ} & \check{v}_i \leq s \leq \hat{v}_i \\ 0.5 & \text{otherwise} \end{cases} \quad (2)$$

where $\check{v}_i$ and $\hat{v}_i$ denote the starting and end position of the area that is covered by the voxel v_i along the ray $\mathrm{r}(\mathbf{P}_t, \mathbf{x}_j)$, while s denotes the position of the reconstructed point $\mathbf{x}_j$ along the same ray.

However, in practice the positions of the features are error-prone and given as probability distributions as stated before. Therefore, we apply an inverse sensor model that takes the Gaussian error distribution into account. While most researchers use an approximated sensor model, we derived our inverse model analytically. One can verify that

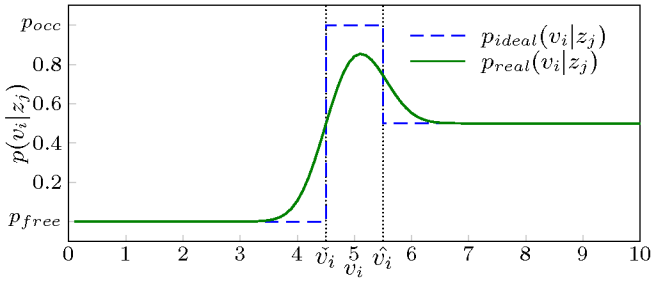

Fig. 3. Occupancy probabilities for an ideal sensor and our Gaussian sensor model.

the real sensor model can be obtained by convoluting the above ideal sensor model with a Gaussian $\mathcal{N}(0, \sigma^2)$. Taking the discretization of the voxels into account, the real sensor model can be described by:

$$\begin{aligned} p_{real}(v_i|z_j) &= p_{ideal}(v_i|s) * \mathcal{N}(0, \sigma^2) \\ &= \mathrm{F}\left(\check{v}_i, s, \sigma^2\right) - \frac{1}{2}\mathrm{F}\left(\hat{v}_i, s, \sigma^2\right) \end{aligned} \quad (3)$$

where $\mathrm{F}\left(x, \mu, \sigma^2\right) = \int_{-\infty}^{x} \mathcal{N}(\mu, \sigma^2)$ is the cumulative normal distribution. The variance σ^2 is taken from the error covariance matrix $\mathbf{\Sigma}_i$ and expresses the position uncertainty of the reconstructed point in depth direction.

The resulting voxel representation that is built using the above update rules can finally be used for path planning and obstacle avoidance. However, most of the existing navigation algorithms still operate on 2D occupancy grid maps. Therefore, the 3D voxel representation can be transformed into such a 2D occupancy grid map by choosing the maximal occupancy value of all voxels located in the column above each grid cell.

IV. Attention-driven Feature Selection

As stated before, most approaches select their features uniformely in the whole image using feature detectors. These detectors choose the strongest features in the input images, i.e. pixels that yield the strongest response of a certain interest operator. Therefore, the position of the selected features depend on the image content only. In this section we use a more biologically inspired attention-driven approach that chooses new features in those image areas that are most relevant for obstacle detection and for preventing a collision. This procedure is similar to humans and animals who usually turn their gaze to areas where the obstacle situations are unclear in order to use monocular and stereo cues that help them to ascertain if they can move on safely. Particularly, close obstacles are watched carefully to observe their precise position and to avoid a potential collision.

In order to achieve the desired behaviour, our approach first computes an attention map that has the same dimensions as each image of the captured sequence. For all pixels $x' \in I_t$ of the input image I_t the attention map $A = \{a(x')|x' \in I_t\}$ contains values $a(x') \in \mathbb{R}$ which indicate the importance of selecting new features in a certain region of the input image.

The actual feature selection is performed by a standard feature detector like the Shi-Tomasi detector in a region of interest $R \subset I_t$ which is a subset of the input image. Since most standard feature detectors rely on the rectangular shape of the input images we use a rectangular region of interest

R with a fixed size. In our experiments the extent of R is set to $^1/_4$ of the complete image. Applying the feature detector in the region of interest makes a big difference compared to detecting features in the whole image, as the detector will select the strongest features in a local image region only. These features usually do not belong to the globally strongest features and would not have been selected if the feature detector was applied on the whole image. Hence, by choosing the position of the region of interest R we can control the location of newly selected features. The position of R is chosen in a way to maximize the sum of the attention values that are covered by the region:

$$R = \underset{R' \subset I_t}{\operatorname{argmax}} \sum_{x' \in R'} a(x') \tag{4}$$

Adding new features in this region therefore maximizes the gain for the whole approach.

For computing the attention map, different measures and objectives O_i can be taken into account. For computing the final attention values $a(x')$ the weighted sum of the attention values $o_i(x')$ of all objectives is used:

$$a(x') = \sum_i w_i o_i(x') \tag{5}$$

Currently, we use two different objectives - an "Obstacle Uncertainty" objective and an "Inhibition of Return" objective. In order to allow the objectives to compute their attention value based on the current scene reconstruction and using information from the navigator about the planned path we implemented a feedback-loop as seen in Figure 1. Hence, the feedback-loop provides top-down information that is used to guide the feature detector.

A. Obstacle Uncertainty - Objective: This objective is used to focus the feature selection to areas where the obstacle situation is unclear and where more observations are necessary. As measure for the uncertainty we use the entropy of the voxel map that was described in the previous section. The entropy is known from information theory and defined as: $H(X) = -\sum p(x) \log_2 p(x)$. The entropy $H(v_i)$ of a single voxel v_i is given as binary entropy function:

$$H(v_i) = -p(v_i) \log p(v_i) - [1 - p(v_i)] \log[1 - p(v_i)] \tag{6}$$

It is maximal when the voxel is initialized with 0.5 and nothing is known about that part of the environment. With additional observations using the reconstructed features the entropy decreases and will finally converge near zero after the voxel has been explored and is classified as either free or occupied. Obviously, each measurement z_j decreases the expected entropy $H(v_i)$ of the voxel and leads to an expected information gain that can be expressed by the mutual information:

$$I(v_i; z_j) = H(v_i) - H(v_i|z_j) \tag{7}$$

where $H(v_i|z_j)$ is the entropy of the voxel v_i *after* inserting the measurement z_j according to Eq. (1) in section III.

If a pixel x'_j is selected as new feature in the input image, the resulting measurement z_j will affect the occupancy probability and hence the entropy of each voxel v_i along the ray $r(P, x'_j)$ in different ways, depending on whether the reconstructed point $\mathbf{x}_j$ is located inside, behind or in front of the voxel v_i. Unfortunately, the location $\mathbf{x}_j$ of the point is still unknown when the corresponding pixel x'_j is selected as feature. However, using the occupancy probabilities we can compute the probability for the point $\mathbf{x}_j$ to be located in a certain voxel along the ray $r(P, x'_j)$. If e.g. the occupancy value of a voxel v_n on the ray is near 1.0 while the values of the previous voxels $v_0, \ldots, v_{n-1}$ on the ray are near 0.0, the point will most likely be located in v_n. In general, the probability for the point $\mathbf{x}_j$ to be located in v_n is:

$$p(\mathbf{x}_j \in v_n) = p(v_n) \prod_{i=0}^{n-1} 1 - p(v_i) \tag{8}$$

while the probability for the point to be located in any voxel behind v_n is:

$$p(\mathbf{x}_j \succ v_n) = \prod_{i=0}^{n} 1 - p(v_i) \tag{9}$$

In the first case, the voxel is assumed to be occupied, its occupancy probability is increased and its entropy changes to $H(v_i|occ)$. In the latter case, the voxel is assumed to be free, the occupancy probability is decreased and the entropy changes to $H(v_i|free)$. Taking these considerations into account, we can predict the expected information gain $I(v_i; x'_j)$ for each voxel v_i along the ray $r(P, x'_j)$ after selecting the feature x'_j:

$$\begin{aligned} I(v_i; x'_j) = H(v_i) \quad &- \quad p(\mathbf{x}_j \in v_n) H(v_i|occ) \\ &- \quad p(\mathbf{x}_j \succ v_n) H(v_i|free) \end{aligned} \tag{10}$$

For computing the entropies $H(v_i|occ)$ and $H(v_i|free)$ we simulate updating the voxel as occupied and free using Eq. (1). In order to simplify the computation we use the ideal sensor model here which is sufficient for this purpose. To approximate the real sensor model coarsely the models parameters are chosen to $p_{occ} = 0.8$ and $p_{free} = 0.2$.

In Figure 4 the information gain for a single voxel is plotted against the occupancy probability of the voxel according to Eq. (10) using the ideal sensor model with different values for p_{occ} and p_{free}. In all graphs the information gain drops at both ends. Hence, updating voxels that are already identified as free or occupied with a high certainty does not lead to a significant information gain. Surprisingly, the function of the information gain may have a local minimum for occupancy values near 0.5 depending on the chosen parameters p_{occ} and p_{free}. This is a result of the characteristics of the the binary entropy function with its steep slope for probabilities near 0 and 1. Small changes in the probability lead to large information gains for these values. The occupancy update function counteracts this tendency. Here, the change in the probability for cells with a occupancy probability near 0.5 is dominant compared to those with probabilities near 0 or 1. For less reliable

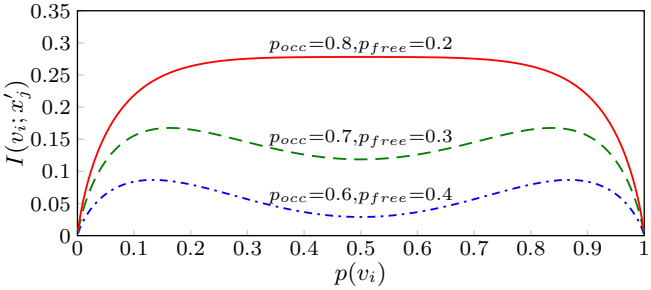

Fig. 4. Information gain $I(v_i; x'_j)$ of a single voxel that is plot against the occupancy probability of the voxel using different values p_{occ} and p_{free} of the ideal sensor model.

sensor models the binary entropy function will dominate the characteristics of the information gain and updating voxels whose occupancy state is completely unknown is expected to yield a smaller information gain compared to voxels where at least some information is already available and where additional observations can ascertain their real states. However, when using values $p_{occ} \geq 0.8$ and $p_{free} \leq 0.2$ for the ideal sensor model this effect disappears.

As stated before, Eq. (10) yields the information gain of a single voxel along the ray after selecting a new feature x'_j. In order to obtain the total information gain for selecting the feature x'_j the gains of all voxels along the ray have to be accumulated. This can be implemented efficiently using ray casting. Putting all together, we get the final attention function for the *obstacle uncertainty objective*:

$$o_1(x') = \sum_i u(v_i) I(v_i; x') \tag{11}$$

In this equation we added an additional weight function $u(v_i)$ that can be used to control the importance of a each voxel. For obstacle avoidance e.g. the occupancy states of voxels along and near the path that was planned by the navigator are more important than voxels that are far away. Additionally, we use higher weights for voxels near the robot than for distant voxels.

B. Inhibition of Return - Objective: The second objective we apply is an inhibitory objective that implements a so called Inhibition of Return. It is required to avoid the attention getting stuck at the same image region while other parts of the image are never chosen for selecting new features. The objective manages an activity map $M = \{m(x') | x' \in I_t\}$ that has the same size as the attention map and the input image. This activity map keeps track of the image regions $R_{t-1}, R_{t-2}, \ldots$ previously selected for feature selection according to Eq. (4). Therefore, each element $m_t(x')$ of the current activity map M_t is updated as follows:

$$m_t(x') = \eta m_{t-1}(x') + \beta \delta(x') \text{ , with } \delta(x') = \begin{cases} 1 & x' \in R_{t-1} \\ 0 & \text{otherwise} \end{cases}$$

Here, η denotes a decay rate that decreases the previous activation $m_{t-1}(x')$. It is chosen to $\eta = 0.95$ in our experiments. The parameter β denotes some activation parameter that adds activation to all elements in the activity map that correspond to the image region R_{t-1} chosen for feature selection in the previous iteration. Reasonable values for this parameter are $\beta = 0.1, \ldots, 0.2$. Finally, the attention value of this objective can be easily defined as: $o_2(x') = -m(x')$, where the activation that accumulates the positions of the previously selected regions has an inhibitory influence on the overall attention $a(x')$ in Eq. (5).

Using the Inhibition of Return Objective together with the Obstacle Uncertainty Objective results in a movement of the region of interest used for feature detection similar to the saccade-like movement of the eyes of vertebrates, allowing to cover the whole field of view while concentrating on the most interesting parts of the environment.

V. Results

In our experiments we used a 1/4" CCD fire-wire camera for image acquisition. The camera is mounted in front of our robot at a height of 1.15 m and tilted by 35° towards the ground.

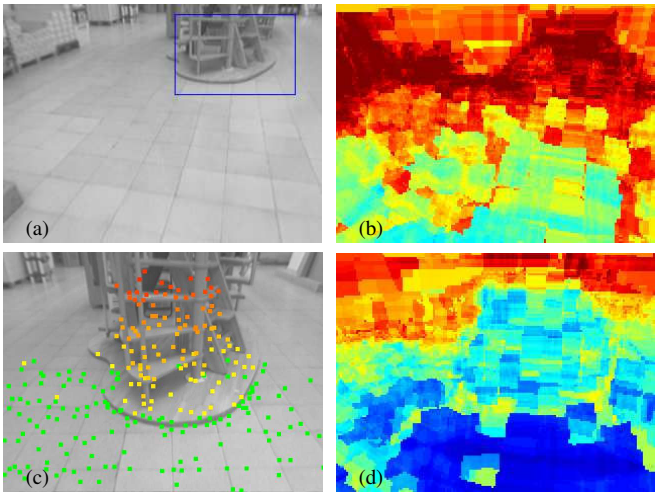

Fig. 5. **(a)** Input image as seen by the robot's front camera. The region that is used for feature selection is marked by the blue rectangle. **(b)** The information gain for each pixel of the upper left image, where red color indicates high values and blue corresponds to low values. **(c)** Input image taken a few frames later. The reconstructed features are shown as dots, where the height is coded by different colors (green: $<$ 0.10 m, yellow-red: 0.10 m - 1.15 m) **(d)** Information gain for each pixel of the lower left image

Fig. 5a shows an image of the scene as seen by the front camera. In Fig. 5b the expected information gain is shown for each pixel of the image. High values are drawn using red colors and indicate high benefits for selecting new features in these regions. As seen in Fig. 5b the highest information gain can be achieved by selecting new features in the upper part of the image especially near the obstacle shown in Fig. 5a. According to Eq. (4) our approach selects the features in this region as indicated by the blue rectangle in Fig. 5a. After the robot has approached the obstacle, our algorithm for scene reconstruction has estimated the 3D positions of the selected features as seen in Fig. 5c. The reconstructed points are shown as colored dots, where the color indicates the estimated height of each feature. Since the obstacle has now been discovered, adding new features in this area would results in minor information gain as denoted by the blue and yellow colors in Fig. 5d. Instead, new features will now be selected in the image regions around the detected obstacle in order to discover these unknown parts of the environment.

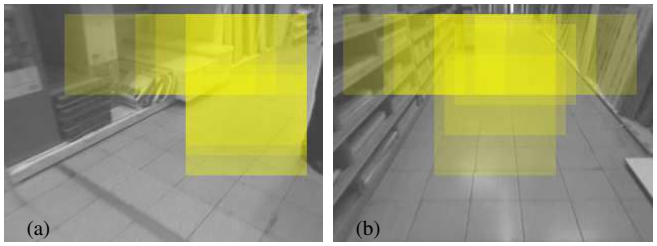

Fig. 6. Image regions that were used for feature selection during the last 10 frames are shown as transparent rectangles. Areas where more attention was paid to are more opaque. Images were taken while **(a)** driving around a right hand bend and **(b)** driving along a narrow corridor.

Similar desired behaviors of our approach are shown in Fig. 6 where we tried to visualize the saccade-like movement of the region of interest. The regions that were used for

feature detection during the last 10 frames are shown as transparent yellow rectangles. Image areas that were used more often are more opaque than areas where less attention was paid to. Fig. 6a was taken while the robot was turning around a right hand bend. Here, our approach guides the feature selection to the upper right image areas that newly became visible to the camera in order to discover those parts of the environment not being observed before. This is important for local path planning since the robot must react quickly to obstacles that suddenly appear behind corners. Fig. 6b shows an image where the robot is moving along a corridor. Here, most features are selected on distant objects in the upper parts of the images. This is reasonable since these features remain visible over more frames of the captured image sequence compared to foreground features that move out of the field of view very quickly due to the robots forward motion. Additionally, the foreground objects have already been discovered by previous measurements.

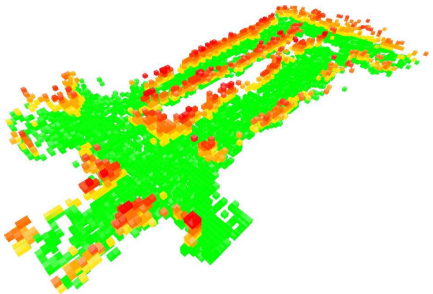

Fig. 7. 3D voxel map that was created while driving through the test environment. Each place was visited only once. The colors of the voxels indicate their heights as in Fig. 5c.

Fig. 7 shows a 3D voxel map that was created using the algorithms presented in this paper. Voxels that were estimated as occupied are shown using different colors, where the color again codes the height of each voxel. For visualization purposes only, we used a laser based SLAM approach for correcting the odometry before creating the maps printed on this page. In our final application the robots odometry is sufficient for mapping since we are only interested in the robots local environment for obstacle detection, where slight odometry errors can be neglect.

In Fig. 8 two occupancy maps are shown that were created from voxel maps of the same test environment. While creating these maps we reduced the number of features that are tracked per frame to 50-100 in order to show the advantages of our proposed guided feature selection. The map in Fig. 8a was created by selecting the features according to the attention-driven approach presented in this paper while Fig. 8b shows a map that was created using features that were detected uniformely in each image.

Although the same number of features was used for creating both maps, the right map in Fig. 8 contains several voxels whose occupancy probability is unknown though they have been visible to the camera. Additionally, some voxels were erroneously estimated as occupied. However, the left map that was created using our guided feature selection approach contains significantly less errors and less voxels whose obstacle situation is unclear. These results show that the guided feature selection can improve the map-building and the detection of obstacles by reducing the uncertainty

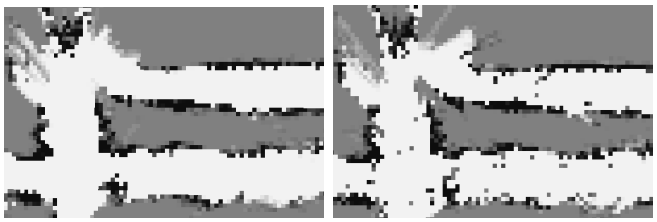

Fig. 8. Occupancy maps that were created from voxel maps. **(left)** Map created using the proposed attention-driven feature selection approach. **(right)** Map created by selecting the features uniformely in each image.

in the created map and therefore by decreasing the risk of missing an obstacle. A video of the approach can be found here: `http://www.youtube.com/user/neurobTV`

VI. Conclusion and Future Work

In this paper we have presented a method for creating volumetric voxel maps using 3D points that were obtained from monocular scene reconstruction. We use the created maps for navigational tasks like obstacle detection. Additionally, these maps provide top-down information for an attention-driven feature selection scheme. Our approach selects new features in those image regions that maximize the information gain. As a result, the created maps contain less uncertainties, and more obstacles can be detected without increasing the number of reconstructed features and therefore without increasing the runtime of the whole algorithm.

Although this approach was developed to robustly handle the obstacle detection problem, it is very flexible and can be modified depending on the purposes the scene reconstruction is used for. With small changes in the objectives presented in section IV or by adding new objectives it can be adapted easily to create a dense scene reconstruction for building precise 3D models or to improve visual odometry.

References

[1] H.-M.Gross, et. al., "TOOMAS: Interactive Shopping Guide Robots in Everyday Use - Final Implementation and Experiences from Long-term Field Trials," in *Proc. of IROS*, 2009, pp. 2005–2012.

[2] E. Einhorn, C. Schröter, and H.-M. Gross, "Monocular Scene Reconstruction for Reliable Obstacle Detection and Robot Navigation," in *Proc. of the 4th ECMR*, 2009, pp. 156–161.

[3] P. Foggia, J. Jolion, A. Limongiello, and M. Vento, "Stereo Vision for Obstacle Detection: A Grap-Based Approach," *LNCS Graph-Based Representations in Pattern Recognition*, vol. 4538, pp. 37–48, 2007.

[4] T. Schamm, S. Vacek, J. Schröder, J. Zöllner, and R. Dillmann, "Obstacle detection with a Photonic Mixing Device-camera in autonomous vehicles," *Int. Journ. of Int. Systems Technologies and Applications*, vol. 5, pp. 315–324, 2008.

[5] E. Einhorn, C. Schröter, H.-J. Böhme, and H.-M. Gross, "A Hybrid Kalman Filter Based Algorithm for Real-time Visual Obstacle Detection," in *Proc. of the 3rd ECMR*, 2007, pp. 156–161.

[6] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Trans. on PAMI*, vol. 29, no. 6, pp. 1052–1067, 2007.

[7] J. Civera, A. Davison, and J. Montiel, "Inverse Depth Parametrization for Monocular SLAM," *IEEE Trans. on Robotics*, pp. 932–945, 2008.

[8] E. Eade and T. Drummond, "Unified Loop Closing and Recovery for Real Time Monocular SLAM," in *Proc. of the BMVC*, 2008.

[9] A. Davison, "Active Search for Real-Time Vision," in *ICCV*, 2005.

[10] M. Chli and A. Davison, "Active Matching," in *Proc. ECCV*, 2008.

[11] T. Vidal-Calleja, A. Davison, J. Andrade-Cetto, and D. Murray, "Active Control for Single Camera SLAM," in *Proc. of IEEE Int. Conf. on Robotics and Automation, ICRA*, 2006, pp. 1930–1936.

[12] S. Frintrop and P. Jensfelt, "Attentional Landmarks and Active Gaze Control for Visual SLAM," in *IEEE Transactions on Robotics, special Issue on Visual SLAM*, vol. 24, no. 5, 2008.

[13] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.