Fusion of Aerial Images and Sensor Data from a Ground Vehicle for Improved Semantic Mapping

Martin Persson[a,*,1] Tom Duckett[b] Achim J. Lilienthal[a]

[a] *Center for Applied Autonomous Sensor Systems, Department of Technology, Örebro University, Örebro, Sweden*

[b] *Department of Computing and Informatics, University of Lincoln, Lincoln, UK*

Abstract

This work investigates the use of semantic information to link ground level occupancy maps and aerial images. A ground level semantic map, which shows open ground and indicates the probability of cells being occupied by walls of buildings, is obtained by a mobile robot equipped with an omnidirectional camera, GPS and a laser range finder. This semantic information is used for local and global segmentation of an aerial image. The result is a map where the semantic information has been extended beyond the range of the robot sensors and predicts where the mobile robot can find buildings and potentially driveable ground.

Key words:
semantic mapping, semi-supervised learning, aerial image, mobile robot

1. Introduction

A mobile robot has a limited view of its environment. Mapping of the operational area is one way of enhancing this view for visited locations. In this work we explore the possibility of using information extracted from aerial images to further improve the mapping process. Semantic information about buildings is used as the link between the ground level information and the aerial image. The method can speed up exploration or planning in areas not yet visited by the robot.

Colour image segmentation is often used to extract information about buildings from aerial images. However, automatic detection of buildings in monocular aerial images without elevation information is hard. Buildings cannot easily be separated from other man-made structures such as driveways, tennis courts, etc. due to the resemblance in colour and shape. We show that wall estimates found by a mobile robot can compensate for the absence of elevation data.

This work builds upon previous published work. In [1] we defined a virtual sensor[2]. With an occupancy grid map and a virtual sensor learned to separate buildings from non-buildings we have a method to build a probabilistic semantic map [2]. In [3] we showed how wall estimates extracted from this probabilistic semantic map could be used for detection of buildings in aerial images by their roof outlines. To determine potential matches between the wall estimates and the roof outlines we used geo-referenced aerial images and an absolute positioning system (GPS) on-board the mobile robot. The matched lines were then used in region- and boundary-based segmentation of the aerial image for detection of buildings.

In this work we extend the approach from [3]. The extension includes global segmentation of buildings in the aerial image, the introduction of a new class for ground (which may be driveable by the robot) and the establishment of the concept and framework of the predictive map. The purpose is to detect building outlines and driveable paths faster than the mobile robot can explore the area by itself. Using this method, the robot can estimate the outline of found buildings and "see" around one or several corners without actually visiting the area. The method does not assume a

[*] Corresponding author.
Email addresses: martin.persson@tech.oru.se (Martin Persson), tduckett@lincoln.ac.uk (Tom Duckett), achim@lilienthals.de (Achim J. Lilienthal).

[1] Supported by the Swedish Defence Material Administration

[2] A virtual sensor is understood as one or several physical sensors with a dedicated signal processing unit for recognition of real world concepts.

perfectly up-to-date aerial image; buildings may be missing although they are present in the aerial image, and vice versa. It is therefore possible to use globally available [3] georeferenced images.

1.1. *Related Work*

Overhead images have been used in combination with ground vehicles in a number of applications. Oh *et al.* [4] used map data to bias a robot motion model in a Bayesian filter to areas with higher probability of robot presence. It was assumed that probable paths were known in the map. Since mobile robot trajectories are more likely to follow those paths in the map, GPS position errors due to reflections from buildings were compensated using the map priors.

Pictorial information such as aerial photos and city-maps have been used for registration of sub-maps and subsequent loop-closing in SLAM [5]. Aerial images were used by Früh and Zakhor in Monte Carlo localization of a truck during urban 3D modeling [6].

Silver *et al.* [7] discuss registration of heterogeneous data (e.g. data recorded with different sampling density) from aerial surveys and the use of these data in classification of ground surface. Cost maps are produced that can be used in long range vehicle navigation. Scrapper *et al.* [8] used heterogeneous data from, e.g., maps and aerial surveys to construct a world model with semantic labels. This model was compared with vehicle sensor views providing a fast scene interpretation.

For detection of man-made objects in aerial images, lines and edges together with elevation data are the features that are used most often. Building detection in single monocular aerial images is very hard without additional elevation data [9]. Mayer's survey [10] describes some existing systems for building detection and concludes that scale, context and 3D structure were the three most important features to consider for object extraction in aerial images. Fusion of SAR (Synthetic Aperture Radar) and aerial images has been employed for detection of building outlines [9]. The building location was established in the overhead SAR image, where walls from one side of buildings can be detected. The complete building outline was then found using edge detection in the aerial image. Parallel and perpendicular edges were considered and the method belongs to edge-only segmentation approaches. This work is similar to ours in the sense that it uses a partly found building outline to segment a building from an aerial image.

Combination of edge and region information for segmentation of aerial images has been suggested in several publications. Two papers that have influenced our work are [11] and [12]. Mueller *et al.* [11] presented a method to detect agricultural fields in satellite images. First, the most relevant edges were detected. These were then used to guide both the smoothing of the image and the following segmentation in the form of region growing. Freixenet *et al.* [12] investigated different methods for integrating region- and boundary-based segmentation, and also claim that this combination is the best approach for image segmentation.

1.2. *Outline and Overview*

The presentation of our proposed system is divided into three main parts. The first part, Section 2, concerns the estimation of walls by the mobile robot and edge detection in the aerial image. At ground level wall estimates are extracted from a probabilistic semantic map. This map is basically an occupancy map built from range data and labeled using a virtual sensor for building detection [1] mounted on the mobile robot. The second part describes matching of wall estimates from the mobile robot with the edges found in the aerial image. This procedure is described in Section 3. The third part presents the segmentation of an aerial image based on the matched lines. Section 4 deals with local segmentation to find buildings and Section 5 extends this to a global segmentation of the aerial image and also introduces the class for ground. Details of the mobile robot, the experiments performed and the results obtained are found in Section 6. Finally, the paper is concluded and suggestions for future work are given in Section 7.

2. Wall Candidates

A major problem for building detection in aerial images is to decide which of the edges in the aerial image correspond to building outlines. The idea of our approach is to match wall estimates extracted from two perspectives in order to increase the probability that a correct segmentation is achieved. In this section we describe the process of extracting wall candidates, first from the mobile robot's perspective and then from aerial images.

2.1. *Wall Candidates from Ground Perspective*

The wall candidates from the ground perspective are extracted from a semantic map acquired by a mobile robot. The semantic map we use is a probabilistic occupancy grid map with two classes: buildings and non-buildings [2]. The probabilistic semantic map is produced using an algorithm that fuses different sensor modalities. In this paper, a 2D range sensor is used to build an occupancy map, which is converted into a probabilistic semantic map using the output of a virtual sensor for building detection based on images from an omnidirectional camera.

The algorithm consists of two parts. First, a local semantic map is built using the occupancy map and the output from the virtual sensor. The virtual sensor uses the AdaBoost algorithm [13] to train a classifier that classifies close range monocular gray scale images taken by the mobile robot as buildings or non-buildings. This generic

[3] E.g. Google Earth, Microsoft Virtual Earth, and satellite images from IKONOS and its successors.

method combines different types of features such as edge orientation, gray level clustering and corners into a system with high classification rate [1].

The classification by the virtual sensor is made for planar sub-images, see Section 6.1. However, the image may also contain parts that do not belong to the detected class, e.g., an image of a building might also include some vegetation, i.e. non-building such as a tree. In order to deal with such situations, probabilities are assigned to the occupied cells that are within a sector, with an opening angle θ, representing the view of the virtual sensor. The size of the cell formations within the sector affects the probability values. These sizes are measured by the horizontal covering angles $\{\alpha_i\} = \alpha_1, \alpha_2, \ldots, \alpha_n$ of the objects within the particular view. A sector is illustrated in Figure 1.

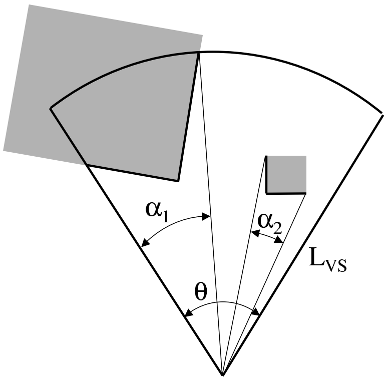


Fig. 1. Illustration of a sector with an opening angle θ and length L_{VS} representing the view of the virtual sensor. Two objects are found (the grey rectangles) within the sector and their respective sizes are represented by α_1 and α_2.

The assigned probabilities $P_i(class|\mathrm{VS}^T, \alpha_i)$ for the objects in view (the grid cells within the sector and seen from the robot) are calculated by the following expression:

$$P_i(class|\mathrm{VS}^T, \alpha_i) = \frac{1}{2} + \frac{\alpha_i}{\theta}(P(class|\mathrm{VS}^T) - \frac{1}{2}) \quad (1)$$

where $P(class|\mathrm{VS}^T)$ is the conditional probability that a view is *class* when the virtual sensor classification at time T is *class*. Thus, higher probabilities are given to larger parts of the view, assuming that larger parts are more likely to have caused the view's classification. In the current implementation $P(class|\mathrm{VS}^T)$ is a constant per class.

In the second step the local maps are used to update a global map, the probabilistic semantic map, utilizing a Bayesian method. The result is a global semantic map that distinguishes between buildings and non-buildings. An example of such a map is given in Figure 2. For more details on this approach to probabilistic semantic mapping see [2].

The lines representing probable building outlines are extracted from the probabilistic semantic map. For the line extraction an implementation by Peter Kovesi [14] was used. The parameter setting for the line extraction is described in Table 1. An example of extracted lines is given in Figure 3.

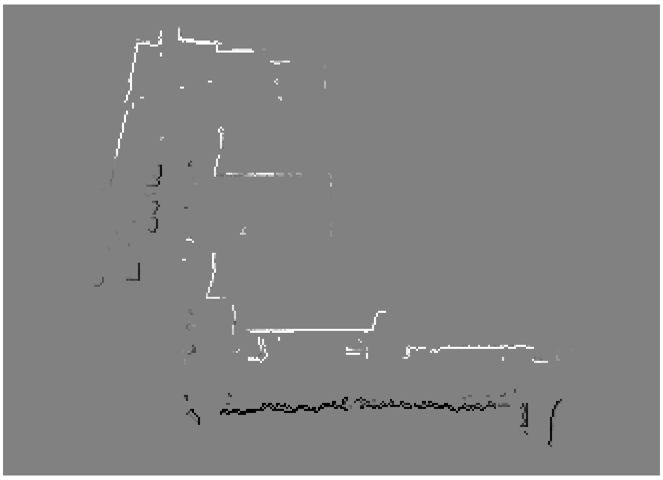


Fig. 2. An example of a probabilistic semantic map created with the approach described in the text. White cells denote high probability of walls and dark cells show outlines of non-building entities.

Name	Value	Description
TOL	2 pixels	Maximum deviation from a straight line before a segment is broken in two
ANGTOL	0.05 rad	Angle tolerance used when attempting to merge line segments
LINKRAD	2 pixels	Maximum distance between end points of line segments for segments to be eligible for linking

Table 1
Parameters used for line extraction.

2.2. *Wall Candidates in Aerial Images*

Edges extracted from an aerial image taken from a nadir view are used as potential building outlines. The edge image is a binary image from which straight lines are extracted to be used as wall candidates for the matching, see Section 3. Here, edge detection is performed separately on the three RGB-components using Canny's edge detector [15]. The resulting edge image I_e is calculated by fusing the three binary images obtained for the three colour components with a logical OR-function. Finally a thinning operation [4] is performed to remove points that occur when edges appear slightly shifted in the different components. For line extraction in I_e the same implementation and parameters as described in Section 2.1 were used. The lines extracted from the edges detected in the aerial image in Figure 3 are shown in Figure 4.

We use the colour edge detection method because it finds more edge points than gray scale edge detection. This is because edges on the border between areas that have different colours but similar intensity are not detected in gray scale versions of the same image. In a test where the two methods had the same segmentation parameters, the colour version produced 19 % more edge points resulting in 17 % more detected lines for an aerial image of size ca. 800×1300 pix-

[4] The Matlab command *bwmorph(im,'thin',Inf)* was used.

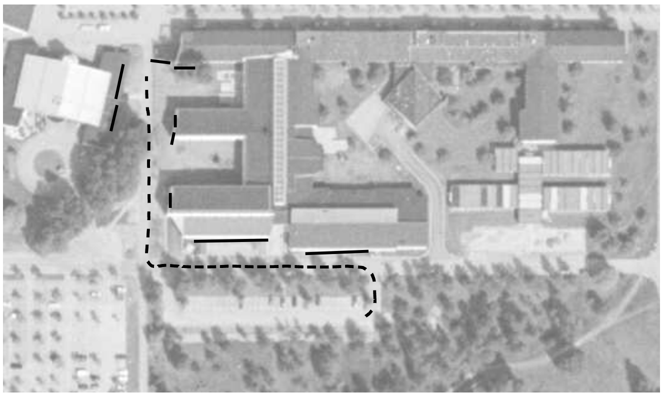


Fig. 3. The trajectory of the mobile robot (dashed), the ground level wall estimates (solid) and the used aerial image (©Örebro Community Planning Office). The semantic map in Fig. 2 covers the upper left part of this figure.

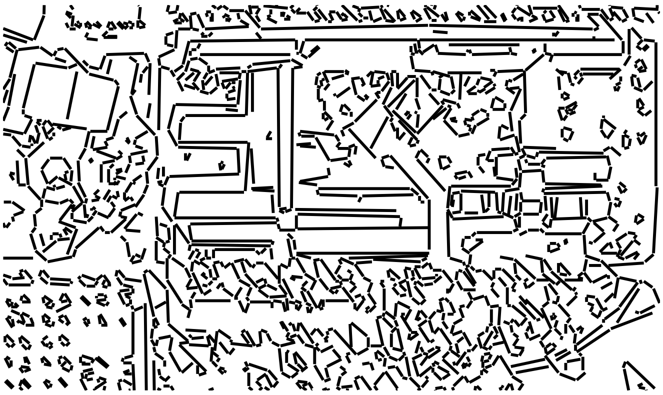


Fig. 4. The lines extracted from the edge version of the aerial image.

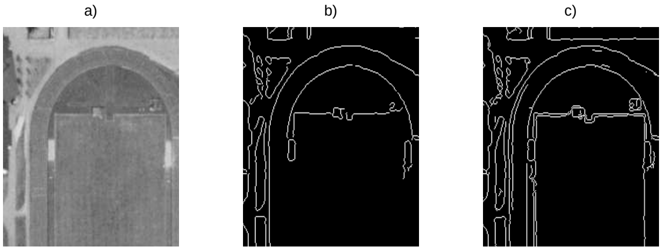


Fig. 5. Gray scale (b) and colour edge detection (c) in an aerial image (a). In the top the colour version finds edges where light green vegetation meets light gray ground, and in the lower part edges are found around the green football field where the grass meets the red running tracks.

els (400×650 m). Figure 5 gives a close-up example from that test to show the differences. The calculation time of the colour edge detection is slightly more than three times longer than ordinary gray scale edge detection. This time is still small in comparison to the routines we use for detecting lines in the edge images.

3. Matching Wall Candidates

The purpose of the wall matching step is to relate wall estimates, obtained at ground level with the mobile robot, to the edges detected in the aerial image. All wall estimates are represented as line segments. We denote a wall estimate found by the mobile robot as L_g and the N lines representing the edges found in the aerial image by L_a^i with $i \in \{1, \ldots, N\}$. Both line types are geo-referenced in the same Cartesian coordinate system.

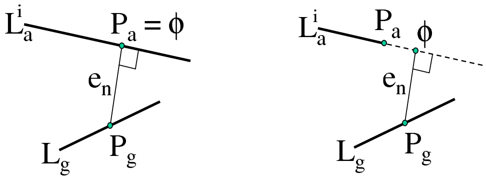


Fig. 6. Selection of characteristic points for the computation of a distance measure between two lines. The figure shows the line L_g (ground level wall candidate) with its midpoint P_g, the line L_a^i (aerial image wall candidate), and the normal to L_a^i, e_n. To the left, $P_a = \phi$ since ϕ is on L_a^i and to the right, P_a is the endpoint of L_a^i since ϕ is not on L_a^i.

The lines from both the aerial image and the semantic map may be erroneous, especially concerning the line endpoints, due to occlusion, errors in the semantic map, different sensor coverage, etc. We therefore need a measure for line-to-line distances that can handle partially occluded lines. Hence, we do not consider the length of the lines and restrict line matching to the line directions and the distance between two characteristic points, one point on each line. The line matching calculations are performed in two steps: 1) determine the two characteristic points, and 2) compute the distance measure to find the best matches.

3.1. *Finding the Closest Point*

In this section we describe how the characteristic points on the two compared lines are determined. For L_g we use the line midpoint, P_g. To cope with the possible errors described above we select the point P_a on L_a^i that is closest to P_g as the best candidate to be used in our line distance measure.

To calculate P_a, let e_n be the orthogonal line to L_a^i that intersects L_g in P_g, see Figure 6. We denote the intersection between e_n and L_a^i as ϕ where $\phi = e_n \times L_a^i$ (using homogeneous coordinates). The intersection ϕ may be outside the line segment L_a^i, see right part of Figure 6. We therefore check if ϕ is within the endpoints and if it is set $P_a = \phi$. If ϕ is not within the endpoints, then P_a is set to the closest endpoint on L_a^i.

3.2. *Distance Measure*

The calculation of the distance measure is inspired by [16], which describes geometric line matching in images for stereo matching. We have reduced the complexity in those calculations to have fewer parameters that need to be determined and to exclude the line lengths. Matching is performed using L_g's midpoint P_g, the closest point P_a on L_a^i

and the line directions θ_g and θ_a. First, a difference vector is calculated as

$$\mathbf{r}_\Delta = [P_{g_x} - P_{a_x}, P_{g_y} - P_{a_y}, \theta_g - \theta_a]^T. \tag{2}$$

Second, the similarity is measured as the Mahalanobis distance

$$d = {\mathbf{r}_\Delta}^T \mathbf{R}^{-1} \mathbf{r}_\Delta \tag{3}$$

where the diagonal covariance matrix $\mathbf{R}$ is defined as

$$\mathbf{R} = \begin{bmatrix} \sigma_{Rx}^2 & 0 & 0 \\ 0 & \sigma_{Ry}^2 & 0 \\ 0 & 0 & \sigma_{R\theta}^2 \end{bmatrix} \tag{4}$$

with $\sigma_{Rx}, \sigma_{Ry},$ and $\sigma_{R\theta}$ being the expected standard deviation of the errors between the ground-based and aerial-based wall estimates. Using Mahalanobis distance, it is only the relation between the parameters that influences the line matching. The important relation is $\sigma_{R\theta}^2/\sigma_{Rx}^2$ and usually $\sigma_{Rx}^2 = \sigma_{Ry}^2$ for symmetry reasons. Note that our distance measure is not strictly a mathematical metric, due to the method for selecting characteristic points.

4. Local Segmentation of Aerial Image

This section describes how local segmentation of the colour aerial image is performed. Generally, segmentation methods can be divided into two groups; edge- and similarity-based [17]. In our case we combine these approaches by first performing edge based segmentation for detection of closed areas and then colour segmentation based on a small training area to confirm the area's homogeneity. The following is a short description of the sequence that is performed for each line L_g:

(i) Sort the set of lines L_a based on d from Equation 3 in increasing order and set $i = 0$.
(ii) Set $i = i + 1$.
(iii) Define a start area A_{start}, 8×8 pixels square (equivalent to 4×4 m), on the side of L_a^i that is opposite to the robot (this will be in or closest to the unknown part of the occupancy grid map).
(iv) Check if A_{start} includes edge points (parts of edges in I_e). If yes, return to step 2. This check ensures that a region has a minimum width and depth.
(v) Perform edge controlled segmentation, see Section 4.1.
(vi) Perform homogeneity test, see Section 4.2.

This process is stopped, either when a region has been found or when all lines in L_a that are close enough to the present line in L_g to be considered have been checked. The "close enough" criterion could be measured by the Euclidean distance between the characteristic points P_g and P_a defined in Section 3.1. However, in the current implementation this was not activated during the experiment in order to be able to study whether other regions were found.

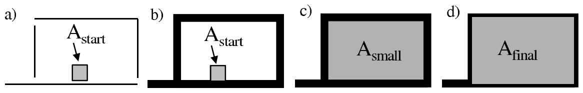


Fig. 7. Illustration of edge controlled segmentation. a) shows a small part of I_e and A_{start}. In b) I_e has been dilated and in c) A_{small} has been found. d) shows A_{final} as the dilation of A_{small}.

4.1. *Edge Controlled Segmentation*

Based on the edge image I_e constructed from the aerial image, we search for a closed area. Since there might be gaps in the edges, bottlenecks need to be found [11]. We use morphological operations, with a 3×3 structuring element, to first dilate the interesting part of the edge image in order to close gaps and then search for a closed area on the side of the matched line that is opposite to the mobile robot. When this area has been found the area is dilated in order to compensate for the previous dilation of the edge image. This procedure is illustrated in Figure 7.

4.2. *Homogeneity Test*

We use the initial starting area A_{start} as a training sample and evaluate the rest of the region based on the corresponding colour model. This means that the colour model does not gradually adapt to the growing region, but instead requires a homogeneous region on the complete roof part that is under investigation. Regions that gradually change colour or intensity, such as curved roofs, might then be partly rejected.

Gaussian Mixture Models, GMM, are popular for colour segmentation. Like Dahlkamp *et al.* [18] we tested both GMM and a model described by the mean and the covariance matrix in RGB colour space. We selected the mean/covariance model since it is faster and we noted that the mean/covariance model performs approximately equally well as the GMM in our case. A limit O_{lim} is calculated for each model so that 95% of the training sample pixels (i.e. pixels in A_{start}) have a Mahalanobis distance smaller than O_{lim}. O_{lim} is then used as the separator limit between pixels belonging to the class and the pixels that do not belong to the class.

4.3. *Alternative Methods*

Above a two step segmentation method to detect homogeneous regions surrounded by edges was presented. There exist a number of segmentation methods that could have been applied. Two alternative methods are discussed in the following. The conclusions are based on preliminary tests performed on the aerial image used in our experiment. For these tests, the parameters used in the respective algorithms were tuned manually.

The first method tested is the graph-based image segmentation, GBIS, by Felzenszwalb and Huttenlocher [19]. GBIS can adapt to the texture and can be set to reject small areas and therefore ignore small-sized disturbances such as shadows from chimneys. Due to this GBIS produces very homogeneous results. A drawback is that GBIS has a tendency to leak and continue to grow outside areas that humans would consider to be closed. Therefore, GBIS does not seem to be an option to replace both steps in our two step method, but it is an alternative to the homogeneity test. In conjunction with the edge controlled segmentation it turns out that GBIS produces similar segmentation results to the mean/covariance model.

The second method tested is a modified flood fill algorithm. The algorithm takes starting pixels from A_{start} and performs region growing limited by colour difference to the starting pixels and local gradient information. Let $\mathbf{C}$ be the mean value vector (RGB) of the starting pixels, $\mathbf{P}_i$ any pixel that has been selected to be inside the region and $\mathbf{P}_n$ a neighbouring pixel that is 4-connected with $\mathbf{P}_i$. For each $\mathbf{P}_n$ a local value g_{loc} is calculated as

$$g_{loc} = e^{-\sum_{j=r,g,b} \frac{(\mathbf{P}_n(j)-\mathbf{C}(j))^2}{\sigma_{col}^2}} e^{-\sum_{j=r,g,b} \frac{(\mathbf{P}_n(j)-\mathbf{P}_i(j))^2}{\sigma_{grad}^2}} \qquad (5)$$

The value of g_{loc} is then compared to a threshold to see if $\mathbf{P}_n$ should be included in the region or not. Due to the use of the local gradient this algorithm performs well both as a replacement for both steps and when it is used only for the homogeneity check. This modified flood fill algorithm can also leak, like GBIS, but only to areas with similar colours since $\mathbf{C}$ only depends on the starting pixels.

5. Global Segmentation of Aerial Images

In this section the view of the mobile robot is increased further. Learned colour models are used in global building segmentation within the entire aerial image. The purpose of global segmentation is to build a map that predicts different types of areas, e.g., driveable ground and buildings. We call this the predictive map, PM. When the PM includes both driveable ground and obstacles in the form of buildings it can serve as an input to a path planning algorithm.

The global segmentation of an aerial image using colour models captures all buildings with roofs in similar colours as those buildings that were detected by local segmentation. However, some colours can be very similar to ground covered by, e.g., asphalt and ground in deep shadow. Therefore it may happen that some of the detected building areas belong to a non-building class. In order to reduce these errors, we introduce an additional class, *ground*, which will compete with the building class about ambiguous pixels.

Areas of driveable ground can be extracted in different ways, e.g., vision has been used in several projects [20–22] to find driveable regions for unmanned vehicles. In this work the free space from the occupancy grid map is interpreted as ground. This free space can be considered to be driveable ground assuming that there are no negative obstacles or other features, which cannot be sensed with the horizontally mounted 2D-laser scanner and prevent the robot from driving safely. However, since we cannot guarantee that the free space in the occupancy grid map in fact corresponds to a driveable area we call the new class *ground.*

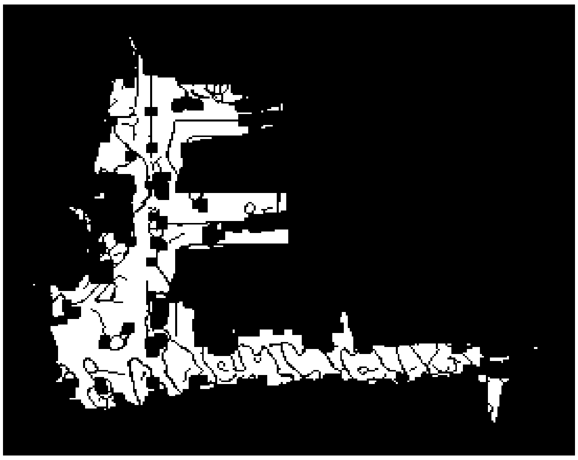


Fig. 8. The combined binary image of free points (reduced using morphological erosion with a square structuring element of size 5×5 pixels) and edges in I_e.

5.1. *Colour Models*

The segmentation of the aerial image is based on colour models. In the example in this article, models are calculated for the two classes: *building* and *ground.* To classify pixels in the aerial image we use the same procedure as for the homogeneity test in local segmentation, see Section 4.

To define the colour models for the building class, we simply use the building estimates found by local segmentation as training areas.

To extract colour models that represent the different ground areas we combine the occupancy grid map and the edge version of the aerial image, I_e. The free cells in the occupancy grid map define the regions in I_e that represent ground. This combination can be done either under the assumption that the navigation is precise giving a perfect registration or by reduction of the area of free cells with the estimated size of the navigation error. We used the latter approach and reduced the area of free cells in the occupancy grid map by morphological erosion with a square structuring element of size 5×5 pixels to compensate for errors up to 1 m in all directions. An example of the combination of the occupancy map and I_e is shown in Figure 8. Next, edge controlled segmentation of that region is performed, as described in Section 4.1, to find the different ground areas. The largest areas[5] point out samples in the aerial image that are used to train mean/covariance models, the same type of colour models as in Section 4.2.

The combination of the result from the local building segmentation (an example can be found in Figure 11) and

[5] The limit was set to 50 pixels (12.5 m^2) in order to avoid small areas that could represent movable objects such as cars and small trucks.

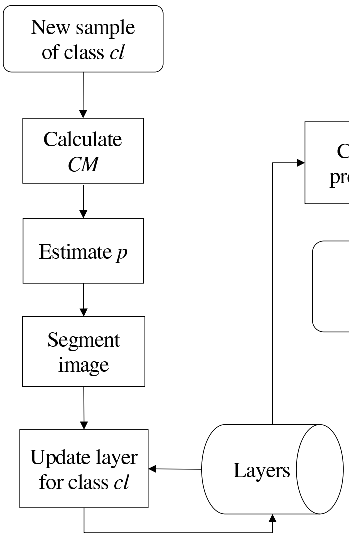


Fig. 9. Flow chart of the process for calculating the predictive map.

the ground information from the occupancy grid map are referred to as the *local information*. Note that both the local building segmentation and the ground information extracted from the occupancy grid map result from direct observation by the mobile robot.

5.2. *The Predictive Map*

The PM is designed to handle multiclass problems and updating this map can be performed incrementally. The PM is a grid map of the same size as the aerial image that is segmented. For each of the n classes, a separate layer l_i, with $i \in \{1, \ldots, n\}$, is used to store the accumulated segmentation results. These layers also have the same size as the aerial image. The colour models used to segment the aerial image are two-class models (building or non-building, ground or non-ground, etc.) and the classifiers are therefore binary classifiers.

To calculate the predictive map incrementally two main steps are performed; 1) the aerial image is segmented when a new colour model is available and 2) the predictive map is recalculated using the result from the latest segmentation. Figure 9 shows a flow chart of the updating process. This is adapted to work also in an on-line situation and is explained in the following. When a *New sample* belonging to class *cl* is available a new colour model *CM* is calculated. Based on the quality of *CM*, a measure $p, 0 \leq p \leq 1$ should be estimated[6]. Then the aerial image is segmented using the new model and the result is multiplied with p and stored in a temporary layer. The old layer, l_{cl}, is fused with the temporary layer using a max function[7].

The predictive map is based on voting from separate layers l_i for the n classes, one layer for each class considered. In this example $n = 2$; one building layer and one ground layer. The voting is a comparison of the layers cell by cell. In those grid cells where the values are similar, the cells are set to *unknown*. To evaluate the similarity between cells buffer zones are introduced in a voting process. The buffer zones are collected in the off-diagonal elements of a matrix $\mathbf{C}$. The off-diagonal elements, $c_{ij} \geq 0, i \neq j, i = \{1, 2, \ldots, n\}, j = \{1, 2, \ldots, n\}$, are then used for the classification of cells pm^{xy} in PM, where pm^{xy} denotes cell (x, y). Introducing the buffer zones defined in $\mathbf{C}$ in the voting process makes it possible to adjust the sensitivity of the voting individually for all classes. The voting is performed using IF-THEN rules biased with c_{ij}:

$$\text{IF } l_i^{xy} > l_j^{xy} + c_{ij} \quad \forall\, j \neq i \text{ THEN } pm^{xy} = class_i \tag{6}$$

where l_i^{xy} denotes cell (x, y) in layer i. If the condition cannot be fulfilled due to conflicting information, pm^{xy} is set to *unknown*. If $c_{ij} = 0$ the rules in Equation 6 will turn into ordinary voting where the largest value wins and where ties give *unknown*.

During the experiments presented in this article $\mathbf{C}$ was set to

$$\mathbf{C} = \begin{bmatrix} - & 0.1 \\ 0.1 & - \end{bmatrix} \tag{7}$$

(the values of the diagonal elements are not used).

All in all, the PM contains information about $n + 2$ categories. First there are the n different classes, then the *unknown* cells due to ambiguous class values and finally the unexplored cells that represent the remaining pixels that cannot be explained by any of the learned colour models.

5.3. *Combination of Local and Global Segmentation*

The approach described above results in two sets of information. The first is the local information that has been confirmed by the mobile robot and the second is stored in the PM. Where these sets overlap they can be fused into one final estimate. Since the local information has been confirmed by the mobile robot it is reasonable to let the local information have precedence over the PM by giving it a higher probability p. Fusion of the PM and the local information can use the same method (with the exception of segmentation) as the updates of the PM described in the previous section.

6. Experiments

6.1. *Data Collection*

The above algorithms were implemented in Matlab [23] for evaluation and currently work off-line. Data were collected with a mobile robot, a Pioneer P3-AT from ActivMedia, equipped with differential GPS (NovAtel ProPak-G2*Plus*), a horizontally mounted laser range scanner (SICK

[6] Estimation of the parameter p is still an unsolved issue for future work. In our experiments we used $p = 0.7$.

[7] Another possibility to fuse the layers would be to use a Bayesian method.

LMS 200), cameras and odometry. The robot is equipped with two different types of cameras, an ordinary camera mounted on a PT-head and an omni-directional camera. The omni-directional camera gives a 360° view of the surroundings in one single shot. The camera itself is a standard consumer-grade SLR digital camera (Canon EOS350D, 8 megapixels). On top of the lens, a curved mirror from 0-360.com is mounted. From each omni-image 8 (every 45°) planar views or sub-images, with a horizontal field-of-view of 56°, were computed. These sub-images are the input to the virtual sensor. The images were taken approximately every 1.5 m along the robot trajectory and were stored together with the corresponding robot pose. The trajectory of the mobile robot is shown in Figure 3. Since the ground where the robot was driven during the experiment is flat, inertial sensors were not needed. This can be confirmed by visual inspection of the resulting occupancy map in Figure 10.

6.2. *Tests of the Local Segmentation*

The occupancy map shown in Figure 10 was used for the experiment. This map was built from data measured by the laser range scanner (with 180 degrees field of view) and positioning data obtained from fusion of odometry and DGPS. The grid cell size was 0.5 m, the range of the data was limited to 40 m and the map was built using the known poses and a standard Bayes update equation as described in [24]. Even though this 2D map works well in our experiments (with exception of the hedge/building mix-up described in Section 6.3), one should note that a fixed horizontally mounted 2D laser is limited for detection of building outlines, especially in cases when the terrain is not flat. Alternative methods suitable for capturing large objects in outdoor environments are 3D laser [25], vertically mounted laser range scanner [6] or (motion) stereo vision [26].

The occupied cells in this map (marked in black) were labeled by the virtual sensor giving the semantic map presented in Figure 2. The semantic map contains two classes: buildings (values above 0.5) and non-buildings (values below 0.5). From this semantic map we extracted the grid cells with a high probability of being a building[8] (above 0.9) and converted them to the lines L_g^M presented in Figure 3. Matching these lines with the lines extracted from the aerial image L_a^N, see Figure 4, was then performed. Finally, based on the best line matches segmentation was performed as described in Section 4.

The three parameters in $\mathbf{R}$ (Equation 4) were set to $\sigma_{Rx} = 1$ m, $\sigma_{Ry} = 1$ m, and $\sigma_{R\theta} = 0.2$ rad. The first two parameters reflect a possible error of 2 pixels between the robot position and the aerial image, and the third parameter allows, for example, each endpoint of a 10 pixel long line to be shifted one pixel (parallel edges in the aerial image do not always result in parallel lines, see roof outline in Figure 4). In the tests described in the following paragraph it will be shown that the matching result is not sensitive to small changes of these parameters.

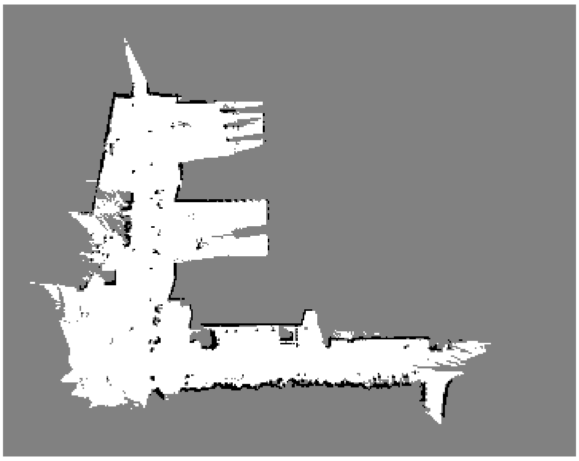


Fig. 10. Occupancy map used to build the semantic map in Fig. 2.

We have performed two different types of tests. The tests are defined in Table 2. *Tests 1-3* are the nominal cases when the collected data are used as they are. These tests intend to show the influence of a changed relation between σ_{Rx}, σ_{Ry} and $\sigma_{R\theta}$ by varying $\sigma_{R\theta}$. In *Test 2* $\sigma_{R\theta}$ is decreased by a factor of 2 and in *Test 3* $\sigma_{R\theta}$ is increased by a factor of 2. In *Tests 4* and *5* additional uncertainty (in addition to the uncertainty already present in L_g^M and L_a^N) was introduced. This uncertainty is in the form of Gaussian noise added to the midpoints (σ_x and σ_y) and directions (σ_θ) of L_g^M and evaluated in Monte Carlo simulation with 20 runs.

Test	σ_x [m]	σ_y [m]	σ_θ [rad]	$\sigma_{R\theta}$ [rad]	N_{run}
1	0	0	0	0.2	1
2	0	0	0	0.1	1
3	0	0	0	0.4	1
4	1	1	0.1	0.2	20
5	2	2	0.2	0.2	20

Table 2
Definition of the parameters used in the different tests.

6.3. *Result of Local Segmentation*

The local segmentation has a limited range and the ground truth area can grow outside of this range without affecting the resulting segmentation, e.g. by including new buildings that are not seen by the robot. A traditional quality measure of true positive rate is therefore not suitable for these tests, since a true positive rate depends on the size of the ground truth area. Instead, the *positive predictive value*, *PPV* or *precision*, has been used as the quality measure. *PPV* is calculated as

$$PPV = \frac{TP}{TP + FP} \tag{8}$$

[8] The limit 0.9 was chosen with respect to the probabilities used in the process of building the semantic map [2]. With this limit at least two positive building readings are needed for a single cell to be used in L_g^M.

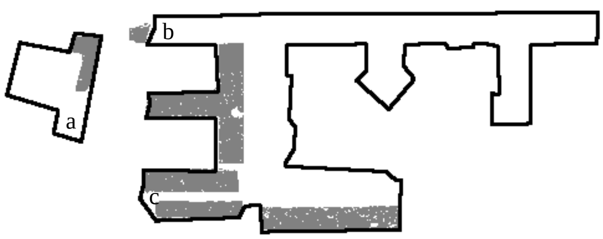


Fig. 11. The result of the local segmentation of the aerial image using the wall estimates shown in Figure 3. The ground truth building outlines are drawn in black.

where TP are the number of true positives and FP are the number of false positives.

The results of *Test 1* show a high positive predictive value of 96.5%, see Table 3. The resulting segmentation is presented in Figure 11. Three deviations from an ideal result can be noted. At a and b tree tops were obstructing the wall edges in the aerial image and therefore the area opposite to these walls was not detected as a building and a gap between two regions appears at c due to a wall visible in the aerial image. Finally, a false area, to the left of b, originates from an error in the semantic map where a low hedge in front of a building was marked as building because the building was the dominating object in the camera view.

The results of *Test 1-3* are very similar, indicating that the algorithm in this case was not specifically sensitive to the changes in $\sigma_{R\theta}$. In *Test 4* and *5* the scenario of *Test 1* was repeated using a Monte Carlo simulation with introduced pose uncertainty. These results are presented in Table 3. One can note that the difference between the nominal case and *Test 4* is very small. In *Test 5* where the additional uncertainties are higher, the positive predictive value has decreased slightly.

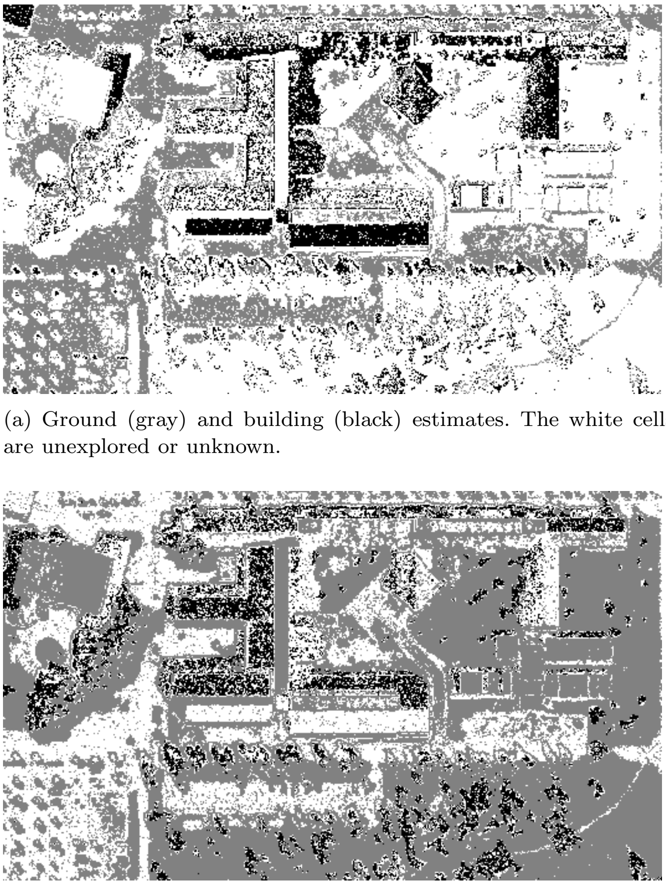
Test	PPV [%]
1	96.5
2	97.0
3	96.5
4	96.8 ± 0.2
5	95.9 ± 1.7

Table 3
Results for the tests defined in Table 2. The results of Test 4 and 5 are presented with the corresponding standard deviation computed from the 20 Monte Carlo simulation runs.

6.4. *Result of Global Segmentation*

The result of the global segmentation is shown in Fig. 12 and 13. Visual inspection of the result shown illustrates the potential of our approach. The PM based on ground colour models from regions in Figure 8 and building colour models from the regions in Figure 11 is presented in Figures 12(a) (cells classified as ground and buildings) and 12(b) (unexplored and unknown cells).

Compared with the aerial image in Figure 3 the result is promising. One can now follow the outline of the main building and most of the paths, including paved paths, roads and beaten tracks, have been found. The main problem experienced during the work is caused by shadowed ground areas that look very similar to dark roofs resulting in the major part of the *unknown* cells.



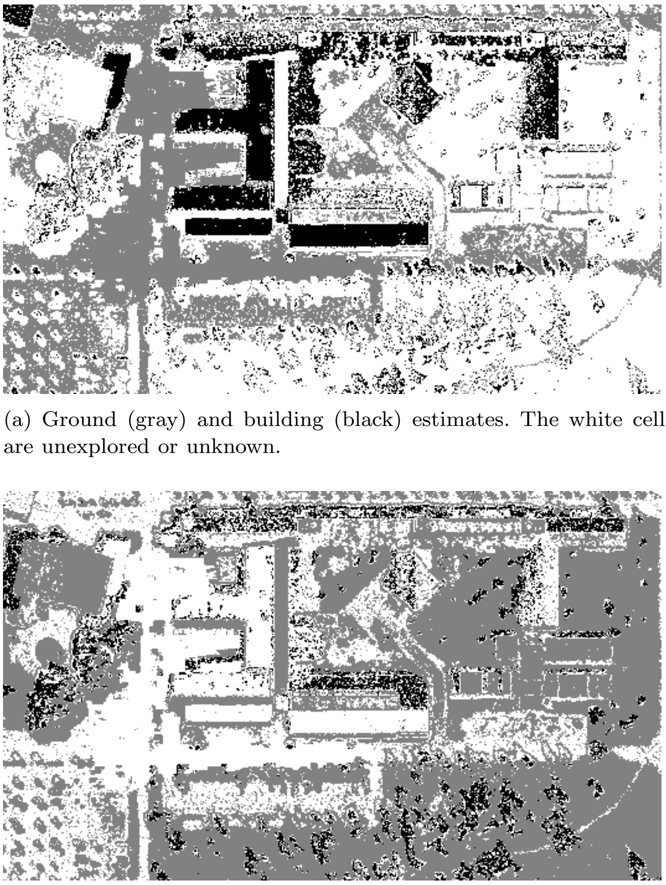
(a) Ground (gray) and building (black) estimates. The white cells are unexplored or unknown.

(b) Ties or unknown cells (black), not classified cells (gray), and classified cells (white).

Fig. 12. The result of the global segmentation of the aerial image (see Section 5) using both ground and building models.

If areas representing the unknown cells have already been classified by the mobile robot, as in Figures 10 and 11, that result has precedence over the PM. The final result is therefore obtained when the PM is combined with the free areas and the buildings found by local segmentation. For these pixels we set $p = 0.9$, performed another update of the PM (using the second step described in Section 5.2) and got the resulting map shown in Figure 13.

A formal evaluation of the ground class is hard to perform. Ground truth for buildings can be manually extracted from the aerial image, but it is hard to specify in detail the area that belongs to ground. Based on the ground truth of buildings and an approximation of the ground truth of ground as the non-building cells, statistics of the result are presented in Table 4. In the table all values in the right column, where the results from the combined PM and local information are shown, are better than those in the middle column (only PM).



(a) Ground (gray) and building (black) estimates. The white cells are unexplored or unknown.

(b) Ties or unknown cells (black), not classified cells (gray), and classified cells (white).

Fig. 13. The PM combined with the local information (see Section 5.3).

Since the PPV depends on the actual presence of the different classes in the aerial image, normalized values are also presented. The normalized values are calculated as

$$PPV_{norm} = \frac{TP_{cl}}{TP_{cl} + FP_{cl}\frac{GT_{cl}}{NGT_{cl}}} \tag{9}$$

where TP_{cl} and FP_{cl} are the numbers of true and false positives of class cl respectively. GT_{cl} is the number of ground truth cells of class cl and NGT_{cl} (non ground truth) is the difference between the total number of cells in PM and GT_{cl}. The area covered by buildings is smaller than the ground area giving an increase in the normalized PPV for buildings and a decrease for ground compared to the nominal PPV.

7. Conclusions and Future Work

This paper discusses how aerial images can be used to extend the observation range of a mobile robot. A virtual sensor for building detection on a mobile robot is used to build a ground level semantic map. This map is used in a process for building detection in aerial images. The benefit from the extended range of the robot's view can clearly be noted in the presented example.

Descriptions	PM (Fig. 12) [%]	PM + local (Fig. 13) [%]
PPV buildings (norm)	66.6 (88.6)	73.0 (91.3)
PPV ground (norm)	96.8 (88.6)	97.3 (90.4)
Building cells	12.3	13.8
Ground cells	21.7	25.8
Unclassified cells	55.5	52.4
Unknown cells (ties)	10.5	8.1

Table 4
Results of the evaluation of the two predictive maps displayed in Fig. 12 and 13. The last four rows show the actual proportions of the cells in the two predictive maps.

In the local segmentation step it can be hard to extract a complete building outline due to factors such as different roof materials, different roof inclinations and additions on the roof, specifically when the robot has only seen a small portion of the building outline. The global segmentation is a powerful extension. Even though the roof structure in the example is quite complicated, the outline of a large building could be extracted based on the limited view of the mobile robot, which had only seen a minor part of surrounding walls.

7.1. *Discussion*

Oh *et al.* [4] assumed that probable paths were known in a map and used this to bias a robot motion model towards areas with higher probability of robot presence. Using the approach suggested in this article these areas could be automatically found from aerial images.

With the presented method, changes in the environment compared to an aerial image that is not perfectly up-to-date are handled automatically. Assume that a building, present in the aerial image, has been removed after the image was taken. It may therefore be classified as a building in the PM if it had a roof colour similar to a building already detected by the mobile robot. When the robot approaches the area where the building was situated, the building will not be detected. If the mobile robot classifies the area as ground, the PM will turn into *unknown* (of course depending on c_{ij} and p), not only for that specific area but also globally, with the exception of areas where local information exists.

What about the other way around? Assume that a new building is erected and this is not yet reflected in the aerial image. If the wall matching indicates an edge as a wall this can of course introduce errors. However, there are several cases where it would not be a problem. When the area is cluttered, e.g., a forest, several close edges will be found and no segmentation is therefore performed. The same result is obtained if the building is erected in a smooth area, for example an open field, since there are no edges to be found. The result of these cases is that the building will only be present in the probabilistic semantic map in the form of a possible wall.

7.2. *Future Work*

We believe that the accuracy of the PM could be further improved by using a measure of the colour model quality to assign a value to the parameter p for each model. Also the probabilities from the semantic map where the ground wall estimates are extracted and the certainty of the virtual sensor could be used in the calculation of p.

We further expect that shadow detection, which merges shadowed areas with corresponding areas in the sun, can reduce the number of false positives and decrease unknown areas caused by ties.

Experiments where the PM is used to direct exploration of unknown areas should be performed. At the same time it should be investigated whether post-processing of the PM, e.g., with filters taking neighbouring cells into account, can improve the results.

Multi-line matching, in comparison to the single line matching used, can relax the need for accurate localisation of the mobile robot. An example of successful matching between ground readings and aerial image for localization is given in [6] and for matching of building outlines in [27].

8. Acknowledgments

The authors gratefully acknowledge the anonymous reviewers for their valuable comments and Christoffer Valgren for the implementation of the modified flood fill algorithm.

References

[1] M. Persson, T. Duckett, A. Lilienthal, Virtual sensor for human concepts – building detection by an outdoor mobile robot, Robotics and Autonomous Systems 55 (5) (2007) 383–390.

[2] M. Persson, T. Duckett, C. Valgren, A. Lilienthal, Probabilistic semantic mapping with a virtual sensor for building/nature detection, in: The 7th IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA2007, Jacksonville, FL, 2007.

[3] M. Persson, T. Duckett, A. Lilienthal, Improved mapping and image segmentation by using semantic information to link aerial images and ground-level information, in: The 13th International Conference on Advanced Robotics, ICAR, Jeju, Korea, 2007, pp. 924–929.

[4] S. M. Oh, S. Tariq, B. N. Walker, F. Dellaert, Map-based priors for localization, in: IEEE/RSJ 2004 International Conference on Intelligent Robotics and Systems, Sendai, Japan, 2004, pp. 2179–2184.

[5] C. Chen, H. Wang, Large-scale loop-closing with pictorial matching, in: Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, Florida, 2006, pp. 1194–1199.

[6] C. Früh, A. Zakhor, An automated mathod for large-scale, ground-based city model acquisition, International Journal of Computer Vision 60 (1) (2004) 5–24.

[7] D. Silver, B. Sofman, N. Vandapel, J. A. Bagnell, A. Stentz, Experimental analysis of overhead data processing to support long range navigation, in: Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 2006, pp. 2443–2450.

[8] C. Scrapper, A. Takeuchi, T. Chang, T. H. Hong, M. Shneier, Using a priori data for prediction and object recognition in an autonomous mobile vehicle, in: G. R. Gerhart, C. M. Shoemaker, D. W. Gage (Eds.), Unmanned Ground Vehicle Technology V. Edited by Gerhart, Grant R.; Shoemaker, Charles M.; Gage, Douglas W. Proceedings of the SPIE, Volume 5083, 2003, pp. 414–418.

[9] F. Tupin, M. Roux, Detection of building outlines based on the fusion of SAR and optical features, ISPRS Journal of Photogrammetry & Remote Sensing 58 (2003) 71–82.

[10] H. Mayer, Automatic object extraction from aerial imagery – a survey focusing on buildings, Computer vision and image understanding 74 (2) (1999) 138–149.

[11] M. Mueller, K. Segl, H. Kaufmann, Edge- and region-based segmentation technique for the extraction of large, man-made objects in high-resolution satellite imagery, Pattern Recognition 37 (2004) 1621–1628.

[12] J. Freixenet, X. Munoz, D. Raba, J. Marti, X. Cufi, Yet another survey on image segmentation: Region and boundary information integration, in: European Conference on Computer Vision, Vol. III, Copenhagen, Denmark, 2002, pp. 408–422.

[13] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1) (1997) 119–139.

[14] P. D. Kovesi, MATLAB and Octave functions for computer vision and image processing, School of Computer Science & Software Engineering, The University of Western Australia, last checked Nov 2007. Available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/> (2000).

[15] J. Canny, A computational approach for edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (2) (1986) 279–98.

[16] J. Guerrero, C. Sagüés, Robust line matching and estimate of homographies simultaneously, in: Pattern Recognition and Image Analysis: First Iberian Conference, IbPRIA 2003, Puerto de Andratx, Mallorca, Spain, 2003, pp. 297–307.

[17] R. C. Gonzales, R. E. Woods, Digital Image Processing, Prentice-Hall, 2002.

[18] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, G. Bradski, Self-supervised monocular road detection in desert terrain, in: Proceedings of Robotics: Science and Systems, Cambridge, USA, 2006.

[19] P. F. Felzenszwalb, D. P. Huttenlocher, Efficient graph-based image segmentation, International Journal of Computer Vision 59 (2) (2004) 167–181.

[20] Y. Guo, V. Gerasimov, G. Poulton, Vision-based drivable surface detection in autonomous ground vehicles, in: Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 2006, pp. 3273–3278.

[21] D. Song, H. N. Lee, J. Yi, A. Levandowski, Vision-based motion planning for an autonomous motorcycle on ill-structured road, in: Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 2006, pp. 3279–3286.

[22] I. Ulrich, I. Nourbakhsh, Appearance-based obstacle detection with monocular color vision, in: AAAI National Conference on Artificial Intelligence, Austin, TX, 2000, pp. 866–871.

[23] The MathWorks, Matlab 7.0, including Image Processing Toolbox 5.0, <http://www.mathworks.com>.

[24] S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Henning, T. Hofmann, M. Krell, T. Schmidt, Map learning and high-speed navigation in RHINO, in: D. Kortenkamp, R. P. Bonasso, R. Murphy (Eds.), Artificial intelligence and mobile robots: case studies of successful robot systems, AAAI Press / The MIT Press, 1998, pp. 21–52.

[25] H. Surmann, K. Lingemann, A. Nüchter, J. Hertzberg, A 3D laser range finder for autonomous mobile robots, in: Proceedings

of the 32nd ISR (International Symposium on Robotics), 2001, pp. 153 – 158.

[26] J. Huber, V. Graefe, Motion stereo for mobile robots, IEEE Transactions on Industrial Electronics 41 (4) (1994) 378–383.

[27] J. R. Beveridge, E. M. Riseman, How easy is matching 2D line models using local search?, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (6) (1997) 564–579.