2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Homography-Based Ground Plane Detection for Mobile Robot Navigation Using a Modified EM Algorithm

D. Conrad* and G. N. DeSouza**

***Abstract*— In this paper, a homography-based approach for determining the ground plane using image pairs is presented. Our approach is unique in that it uses a Modified Expectation Maximization algorithm to cluster pixels on images as belonging to one of two possible classes: ground and non-ground pixels. This classification is very useful in mobile robot navigation because, by segmenting out the ground plane, we are left with all possible objects in the scene, which can then be used to implement many mobile robot navigation algorithms such as obstacle avoidance, path planning, target following, landmark detection, etc. Specifically, we demonstrate the usefulness and robustness of our approach by applying it to a target following algorithm. As the results section shows, the proposed algorithm for ground plane detection achieves an almost perfect detection rate (over 99%) despite the relatively higher number of errors in pixel correspondence from the feature matching algorithm used: SIFT.**

I. Introduction

One of the main problems in mobile robot navigation and map building is obstacle detection [3]. To accomplish this task, many sensors have been employed to retrieve information about the robot's environment such as ultrasonic sensors (SONAR) [4] and laser scanners [6]. Even though these sensors provide means of detecting obstacles, they have a tendency to be expensive. This reason coupled with the recent increase in the availability of low cost cameras makes a vision based approach for mobile robot navigation quite attractive.

While some approaches focus on the 3D reconstruction of an entire scene [1], [10] , many others focus on just finding the ground plane [18], [17], [14], [15], [5], [19]. That is, the classification of pixels as either belonging to the ground plane or not. Some approaches rely on simple color properties of the image to detect ground planes, however they are constrained to specific environments [9]. Other approaches use a stereo camera setup to detect ground planes either from the 3D reconstruction of the scene [10] or from some disparity constraints derived from the setup of the cameras [17], [15]. While these algorithms can provide reasonable results, they require two cameras that need to be calibrated precisely. Optical flow is also used for ground plane detection since the motion fields provide a simple method of clustering the different planes. However, such methods can be computationally intense and a coarse-to-fine approach must be employed to deal with this problem [5]. There are also approaches that exist which are homography-based, such as in [18], [5], [19] that take images of two different views of a scene and use the homography constraint as a criterion for ground plane detection. Finally, like in our proposed method, some approaches resort to a probabilistic scheme to cluster the ground and non-ground pixels. In [14], for example, the RANSAC algorithm is used for robust estimation of the ground plane, but as it is well known, the RANSAC algorithm starts to fail whenever the number of outliers exceeds 40% of the total number of data points.

In this paper, we propose a modification to the Expectation Maximization (EM) algorithm [2] to create a homography-based ground plane detection algorithm. The idea is to apply the SIFT algorithm [11] to establish a correspondence between pixels from an image pair and then apply the homography constraint under a probabilistic framework as the criterion for the classification of the pixels as either ground plane or obstacle. The probabilistic framework allows for the parameters of the homography to be constantly updated while the accuracy of the classification is maximized. Our method requires a simple initialization step due to the fact that the EM needs an initial guess, however, this initialization only needs to be performed once for a mobile robot's camera configuration.

This paper is organized as follows: Section 2 provides a full description of the homography-based ground plane detection algorithm using the proposed Modified EM (or MEM). Section 3 discusses the application of the proposed MEM to mobile robot navigation. Finally, in Section 4 we present the results followed by the conclusions and future work in Section 5.

II. Homography-based Ground-plane Detection using MEM

As we have briefly explained, our goal is to perform mobile robot navigation by segmenting out the images of the objects from the ground plane. By doing so, algorithms for obstacle avoidance, target tracking (e.g. human following), landmark detection, and many others can be more easily implemented [3]. So, in order to carry out these tasks, it is required first that image pixels be clustered into at least two different sets: ground pixels and non-ground pixels. As we mentioned in Section I, many approaches have been proposed over the years for this clustering [18], [17], [14], [15], [5], [19]. In our work, we proposed a new method for ground-plane detection using the homography of the ground plane and a new unsupervised clustering approach based on the *Expectation Maximization* algorithm.

Department of Electrical and Computer Engineering, University of Missouri, 349 Eng. Building West, Columbia, MO, USA *dacry2@mizzou.edu ** DeSouzaG@missouri.edu

978-1-4244-5040-4/10/$26.00 ©2010 IEEE

A. Homography Relation and Decomposition

As we know, a homography is a transformation matrix that relates the pixel coordinates of planar points as seen from two different viewing angles. That is, given the pixel correspondences on two images of a plane, the pixel coordinates of any of the planar points will satisfy the following constraint:

$$s\hat{p}_i = Hp_i \tag{1}$$

where, p_i and $\hat{p}_i$ are the homogeneous pixel coordinates of the planar points on image I and $\hat{I}$, s is a scale factor, and H is the homography of the plane between the two images. The pair of corresponding pixels p_i and $\hat{p}_i$ will be referred to as the *pixel correspondence* x_i. In [7], it is shown that H can be decomposed into:

$$H = \hat{A}(R + \frac{t}{d}n^T)A^{-1} \tag{2}$$

In this equation, A and $\hat{A}$ are the intrinsic parameters of the cameras, which would be the same in a single-camera setting. These parameters can be easily obtained using a camera calibration technique like the one described in [16]. Also, the parameters R and t are the 3x3 rotation matrix and the 3x1 translation vector describing the motion between the two cameras, while the parameters n and d are respectively the normal vector that defines the plane, and the distance between camera and plane. Together, these parameters represent a total of 10 unknown elements: three for the rotation matrix, three for the translation vector, three for the normal vector, and one for the distance value. However, since a homography only has 8 degrees of freedom, these elements are not independent and theoretically only four pairs of pixel correspondences are required to fully determine a homography.

In the next section we explain how we use the homography constraint to classify pixel correspondences as ground or non-ground.

B. Expectation Maximization Algorithm

The Expectation Maximization algorithm is a powerful method commonly used for unsupervised clustering in pattern recognition. In a typical application, observed data needs to be grouped into different clusters based on a probability density function whose parameters are unknown. In order to achieve such clustering, the EM employs two steps: 1) An expectation step, or E-Step, which computes the expected value of a likelihood function based on the current set of the parameters stored in the vector θ_t; and 2) A maximization step, or M-Step, which calculates a new parameter vector θ_{t+1} that maximizes the likelihood estimate used in the E-Step. The EM algorithm iterates between these two steps until convergence and the final clustering of the data x_i, is given by:

$$P(C_j|x_i;\theta_t) \tag{3}$$

that is, the posterior probability of the cluster C_j given the data x_i and the parameters θ_t. Since this probability density function is hard to be found directly, we resort to Bayes theorem to express (3) in terms of its prior probability. That is:

$$P(C_j|x_i;\theta_t) = \frac{P(x_i|C_j;\theta_t)P(C_j)}{\sum_k P(x_i|C_k;\theta_t)P(C_k)} \tag{4}$$

where $P(x_i|C_j;\theta_t)$ is usually easier to be inferred. Also, $P(C_j)$, the prior probability of cluster C_j, can be arbitrarily initialized (e.g. uniformly) and is subsequently refined during the iterations in the EM algorithm.

In other words, for each cluster C_j, and given an initial set of parameters θ_t, the E-Step of the algorithm estimates (3) using (4), while in the M-Step, it refines that same probability by re-calculating the next parameter vector θ_{t+1} using maximum likelihood, that is:

$$\theta_{t+1} = \arg\max_\theta \sum_i \sum_j P(C_j|x_i;\theta_t) \ln\left(P(x_i|C_j;\theta_t)P(C_j)\right) \tag{5}$$

The solution for the above maximization is given by the following expression:

$$\sum_i \sum_j P(C_j|x_i;\theta_t) \frac{d}{d\theta_j} \ln P(x_i|C_j;\theta_t) = 0 \tag{6}$$

Finally, as we mentioned earlier, $P(C_j)$ must also be refined at each iteration of the algorithm, which is done using a simple sample mean:

$$P(C_j) = \frac{1}{N}\sum_{j=1}^{N} P(C_j|x_i;\theta_t) \tag{7}$$

C. Proposed Modification to EM

Before we explain the proposed modifications to the EM algorithm, let us define how the above equations apply to the specific problem of ground plane detection. First, in this application, the observed data x_i are the pixel correspondences established between image pairs using the SIFT algorithm [11]. Also, the parameter vector θ_t consists of the unknowns in the decomposition of the homography given by (2). That is, the angles of rotation and the values of translation between the two positions of the camera, plus the direction (normal) and the distance between camera and the ground plane. Next, since (4) requires us to determine the likelihood $P(x_i|C_j;\theta_t)$, we could rearrange (1) to return a geometric error distance of the following form:

$$err_i = ||\hat{p}_i - \frac{H_{ground}p_i}{s}|| \tag{8}$$

In this case, the likelihood that a pixel correspondence belongs to the ground plane could be made inversely proportional to this distance. Unfortunately, such metric does not return a value between 0 and 1, and therefore it cannot be directly employed as our likelihood function. Instead, we must apply a decay function to this error distance so that the

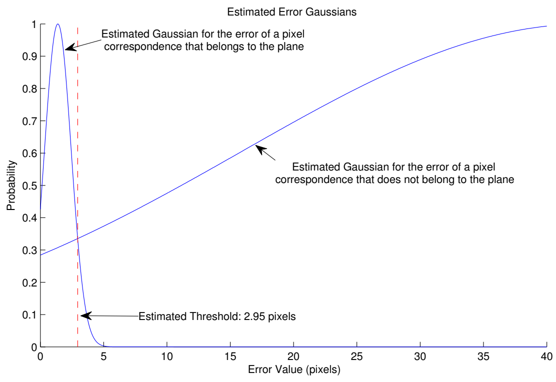

Fig. 1. Result of the empirical study for the choice for the standard deviation of the decay function

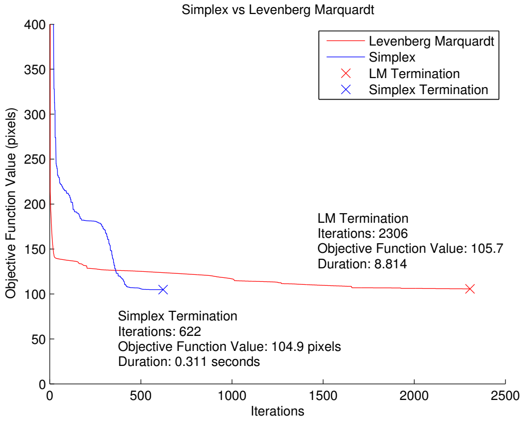

Fig. 2. Simplex algorithm vs. Levenberg-Marquardt

output becomes more appropriate for our needs. One requirement for this decay function is, of course, monotonicity, as to guarantee that no probability inversion occurs between two pixel correspondences. Here, we choose to use a standard Gaussian centered at 0 ($\mu = 0$) with standard deviation σ. The final form of our likelihood function is then given by:

$$P(x_i|C_{ground};\theta_t) = \frac{\exp(-\frac{err_i^2}{2\sigma^2})}{\sum_i \exp(-\frac{err_i^2}{2\sigma^2})} \tag{9}$$

where err_i is the geometric error distance defined by (8) for correspondence x_i assuming that x_i belongs to cluster C_{ground} and given the current parameters of the homography of the ground plane stored in θ_t. The summation in the denominator is over the entire set of pixel correspondences and is used to approximate the total number of pixels in cluster C_{ground}.

The choice for the standard deviation in the decay function was reached empirically. That is, we sought the optimal standard deviation for the decay function that simultaneously satisfies the following criteria: 1) the optimal standard deviation must be large enough so that the effect of small errors in the matching from SIFT could be minimized; and 2) the optimal standard deviation must be small enough so that pixel correspondences that do not belong to the ground plane are not given high probabilities. Figure 1 illustrates our choice of the standard deviation. In this figure, the left curve is a Gaussian that models the geometric error of a pixel correspondence that belongs to the plane, while the right curve models the errors of pixel correspondences that do not belong to the plane. Our choice of standard deviation (σ= 3) is represented by the dashed line which is the intersection of these curves.

Since we do not have a homography to describe the non-ground pixels, the only question remaining is how to assign probabilities when assuming the pixels belong to cluster $C_{non-ground}$. To solve this problem, we look at the output of the decay function not only as an indication of the likelihood that a pixel correspondence belongs to the ground plane, but also as a indication that it does not belong to the ground plane and therefore belongs to cluster $C_{non-ground}$. To use the decay function in the second way, we take the complement of the decay function. In other words, we replace the $\exp(-\frac{err_i^2}{2\sigma^2})$ terms with $\left(1-\exp(-\frac{err_i^2}{2\sigma^2})\right)$ in both the numerator and denominator in (9). It should be noted here that this is not the same as assigning a probability that is the complement of $P(x_i|C_{ground};\theta_t)$. Instead, we are simply taking the complement of the decay function.

Now that we have defined these equations for our application, we can explain why it is necessary for us to modify the EM algorithm. As stated previously, after all of the posterior probabilities have been computed, the next step of the algorithm is to calculate the new parameter vector θ_{t+1} based on maximum likelihood. In order to do this, we refer to (6) which is the desired solution for θ_{t+1}, but involves taking the derivative with respect to each parameter in the vector θ_t. For our case, it would be impractical to factor (8) (2) and (9) into (6). For that reason, we propose an alternative solution to the M-Step.

In this modified M-Step we choose to use an optimization algorithm to estimate the parameters in θ_{t+1} instead of solving for them analytically. The objective function of this minimization is the summation of the geometric error in (8) for all of the correspondences with respect to the homography of the ground plane. The problem with this approach is the fact that we cannot simply use all the detected pixel correspondences, otherwise the optimization will find values of θ_{t+1} that satisfy all of the pixels, whether they form a plane or not. Therefore, in order for this optimization to be successful, we need a criterion to minimize the effect of pixel correspondences that do not belong to the ground plane. We do this by calculating the geometric error for a pixel correspondence and weighting it with the posterior probability found for the pixel correspondence. This idea creates a step that is similar to (5), which is the goal of the original M-Step.

For the optimization that replaces the maximization step, we investigated two algorithms: Levenberg-Marquardt [12] and the Simplex method [8]. The reason that these algorithms

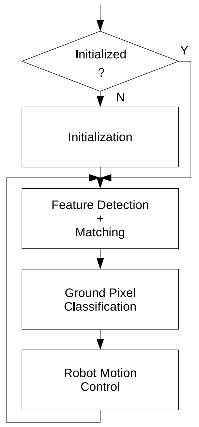

Fig. 3. Flowchart of the Target Follower Navigation Algorithm

are attractive for our MEM is because they do not require an explicit gradient or hessian. In order to help make the decision on which algorithm to use, we performed a comparison which is described in the next section.

D. Levenberg-Marquardt vs. Simplex

We compared the performance of the two algorithms by creating a test where we ran each optimization on 256 real pixel correspondences that belonged to a plane. Each optimization was provided with the same initial guess for θ_t and the same termination criteria. Both optimizations were ran until convergence and the output can be seen in Fig. 2.

As shown in the figure, both algorithms quickly decrease the objective function value within a few iterations. It can be seen that even though the Levenberg-Marquardt at first reaches a lower objective function value early in the iterations, it takes a longer time to converge to a final solution. The Simplex optimization however surpasses it, reaching approximately the same ending objective function value. Not only does the Simplex method converge after fewer iterations, but also it takes much less time to run a single iteration. For these reasons, the decision was made to use the Simplex algorithm for our modified M-Step.

III. Navigation Algorithm

As we mentioned earlier, the proposed homography-based method for ground plane detection together with the Modified EM algorithm can be very useful in many aspects of mobile robot navigation. In this section, we provide one example by employing our method in a simple target tracking and following algorithm. To keep this algorithm simple and focus on the performance of the classification itself, we assume that the mobile robot only sees one object on the ground. This object is the one that the robot needs to follow.

The algorithm consists of four steps, which are illustrated in Figure 3. These steps are: 1) Initialization; 2) Feature Detection and Matching; 3) Ground Pixel Classification; and 4) Robot Motion Control. Details about each of these steps are provided in the following sections.

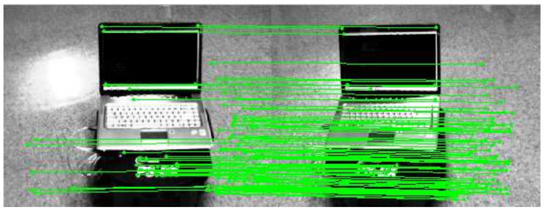

Fig. 4. Example of Sift Output

A. Initialization

Since the homography constraint could be applied to any plane in the environment, it is necessary that the algorithm be initialized with respect to the desired plane, in this case, the ground plane. This initialization is simple, but important since the Expectation Maximization algorithm is sensitive to the initial guess, and the overall quality of the clustering depends on it. In order to carry out such initialization, the algorithm must be provided with two images of the ground plane, with no objects. The initialization step then finds pixel correspondences between those two images and runs the MEM algorithm using a posterior probability set to one for all pixel correspondences. It is important to stress the fact that this initialization only needs to be done once for any robot, that is, for any configuration of the cameras.

B. Feature Detection and Matching

Once the robot has been initialized, the detection of the ground plane starts by the capture of two images. These images are run against each other using the SIFT algorithm [11], which extracts a large number of pixel correspondences, both on the ground plane as well as on possible objects in the scene. A sample of the output from the SIFT algorithm can be seen in Figure 4. It is this output of pixel correspondences that is passed onto the MEM algorithm for clustering.

C. Ground Pixel Classification

The MEM algorithm is the step of our navigation algorithm where pixel classification occurs. As mentioned before, this pixel classification is the main step of the navigation algorithm and is what allows an object in front of the robot to be detected and tracked. As *Algorithm 1* shows, the MEM takes as input the pixel correspondences detected by SIFT, and outputs two clusters: ground pixels and non-ground pixels. Our implementation relies on two cameras, however the algorithm can be used in a single camera setting. The difference between these two settings comes when creating the guess for the MEM. For a single camera setting some method for obtaining a guess for the transformation of the robot between frames would need to be incorporated such as using dead reckoning or decomposition of the fundamental matrix. By using two cameras, we are able to eliminate the need for this step, and are able to use the same guesses for every pair of frames. The MEM step remains the same for either setting in that it takes an initial guess and clusters the pixel correspondences. An example of a clustering produced by the MEM can be seen in Figure 6 where black squares

Algorithm 1 MEM Algorithm

```
Input: Pixel Correspondences
Output: Classified Pixels
  while(!termination_criteria_met)
    Calculate_Probabilities()
    Update_Homography_Parameters()
    Update_Class_Probabilities()
  end
  Calculate_Probabilities()
  Assign_Clustering()
```

represent ground pixels and the blue circles represent the target object pixels.

D. Robot Motion Control

Once the MEM step returns the two clusters of pixel correspondences, that information can be used for target tracking and following. Since we are interested in following the target, the pixels that we actually use for navigation are the pixels on the target object and not the pixels on the ground plane. In order to track the target object, the algorithm aims to keep the target object in the center of the field of view of the camera on the robot. Any deviation of the target object from the center requires a modification of the heading of the mobile robot. This adjustment of the heading should result in the target object being centered again in the camera's view.

The calculation of the adjustment heading is done by the following steps. First, we back-project all the pixels of the target object into the 3D space in front of the robot. Since we do not have depth information, we simply take the rays departing from the camera and project them onto the horizontal axis. The accumulation of intersection points between the rays and the horizontal axis is what is used for the actual control of the robot.

IV. Experimental Results

Our target tracking and following algorithm was implemented and tested on a HP Pavilion dv6 running Intel(R) Core(TM)2 Duo CPU @ 2.0GHz. In order to improve the performance of the navigation algorithm, we ran the MEM step only once for every four image pairs collected. That is, the MEM step of the algorithm was used on the first pair of images to estimate the probabilities and the parameter vector as described in section II-B. After that, the same parameter vector and ground class prior probability, $P(C_{ground})$, were preserved during the classification of the pixels for the next three collected image pairs. Upon collection of the fourth image pair, the MEM was ran again to obtain a new parameter vector and probabilities. The assumption was that the environment does not change drastically within four frames, so the MEM algorithm could rely on the same parameters of the first pair.

For the actual tracking and following algorithm, we used two P3DX mobile robots from Mobile Robots Inc (Fig. 5). One of the mobile robots served as our target object, which ran a program that allows the robot to "wander" through the hallway, while avoiding collisions using its on board SONAR ring. Meanwhile, the second robot ran the proposed tracking and following algorithm. The experiment consisted of capturing 6 trials of the robot running all the way down the hallway while tracking and following the other robot. This equated to 1000 frames and over 400,000 pixels for classification by the proposed algorithm.

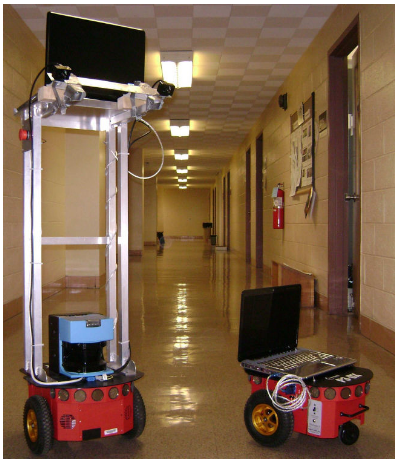

Fig. 5. Mobile Robots used for testing

	Total number of pixels Classified	Correct Classification Percentage	Incorrect classification from SIFT	Incorrect classification from MEM alone
Ground	88,145	99.62%	270 (0.3%)	71 (0.08%)
Non-ground	7,930	99.4%	24 (0.3%)	22 (0.3%)
Total	96,075	99.6%	294 (0.3%)	93 (0.1%)

TABLE I

Statistics of the Classification

A. Qualitative Results

The statistics collected for the experiment above are summarized in Table I. Since we did not have ground truth available, the ground truth had to be manually obtained. For this reason, we randomly sampled 20% of the 1,000 frames collected from the six sequences. The table summarizes the statistics of the 20% of the samples.

As it can be noticed from the table, 96,075 pixel correspondences were collected, with 88,145 of these pixels being classified as ground pixels, and 7,930 as target object pixels. Also, the algorithm returned 341 ground pixels that happened to be misclassified as object pixels. At the same time, it also returned 46 object pixels that were misclassified as ground pixels. However, not all of these misclassifications were due to the proposed algorithm alone. That is, 270 of ground pixels were misclassified because the SIFT algorithm was not able to find a correct match between the pair of images. Similarly, 24 of the target object pixels were misclassified for the same reason. In the end, only 71 of the ground pixels and 22 of the target object pixels were misclassified by the

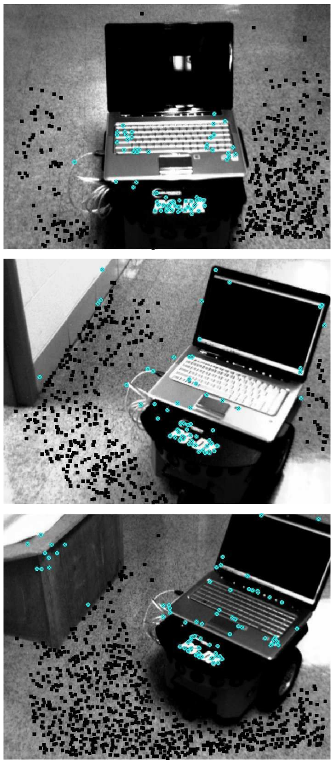

Fig. 6. Sample images from test sequences. Black squares are pixels classified as ground. Blue circles are pixels classified as non-ground.

MEM alone. Overall, the MEM algorithm achieves a total correct classification rate of 99.6%. Some samples from the sequences can be seen in Figure 6.

V. Conclusion and Future Work

In this paper we introduced a Modified Expectation Maximization algorithm that can be employed in a novel approach to homography-based ground plane detection. We have shown that this approach provides very accurate means for classifying image pixels as either belonging to objects or the ground plane in a scene. The algorithm was tested using a Simplex optimization algorithm, which out performed the Levenberg-Marquardt, but in the future other possible optimization algorithms can be studied – e.g. particle swarms. Also, a simple target following navigation algorithm was developed as a proof of concept and test case for the MEM algorithm. In the future, this algorithm will be improved in order to detect multiple planes for indoor navigation in hallways. Other applications of our MEM method for ground plane detection include its use in an outdoor setting, where the ground plane to be detected is not necessarily smooth as in the case of indoor navigation. Figure 7 shows one such example where the algorithm is being used for target tracking and geolocation from airborne video [13].

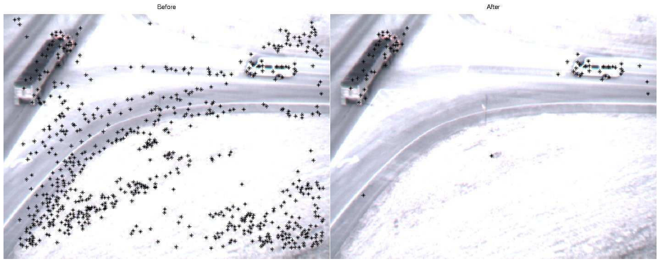

Fig. 7. Before image of all pixels and after image after ground plane pixels are segmented out

References

[1] Anoop Cherian, Vassilios Morellas, and Nikolaos Papanikolopoulos. Accurate 3d ground plane estimation from a single image. In *IEEE International Conference on Robotics and Automation*, 2009.

[2] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 1977.

[3] G.N. DeSouza and A.C. Kak. Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[4] A Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 1987.

[5] Young geun Kim and Hakil Kim. Layered ground floor detection for vision-based mobile robot navigation. In *IEEE International Conference on Robotics and Automation*, 2004.

[6] J. Hancock, M. Hebert, and C. Thorpe. Laser intensity-based obstacle detection. In *International Conference on Intelligent Robots and Systems*, 1998.

[7] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision Second Edition*. Cambridge University Press, 2004.

[8] J.C. Lagarias, J.A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal of Optimization*, 1998.

[9] Scott Lenser and Manuela Veloso. Visual sonar: Fast obstacle avoidance using monocular vision. In *IEEE International Conference on Intelligent Robots and Systems*, 2003.

[10] Paolo Lombardi, Michele Zanin, and Stefano Messelodi. Unified stereovision for ground, road, and obstacle detection. In *IEEE Intelligent Vehicles Symposium*, 2005.

[11] D.G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, 1999.

[12] Donald Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 1963.

[13] Kyung min Han and Guilherme DeSouza. Instantaneous geo-location of multiple targets from monocular airborne video. In *IEEE International Geoscience and Remote Sensing Symposium*, 2009.

[14] F. Mufti, R. Mahony, and J. Heinzmann. Spatio-temporal ransac for robust estimation of ground plane in video range images for automotive applications. In *IEEE Conference on Intelligent Transportation Systems*, 2008.

[15] Tilman Wekel, Olaf Kroll-Peters, and Sahin Albayrak. Vision based obstacle detection for wheeled robots. In *International Conference on Control, Automation and Systems*, 2008.

[16] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[17] Jun Zhao, Jasyantha Katupitiya, and James Ward. Global correlation based ground plane estimation using v-disparity image. In *IEEE International Conference on Robotics and Automation*, 2007.

[18] Jin Zhou and Baoxin Li. Homography-based ground detection for a mobile robot platform using a single camera. In *IEEE International Conference on Robotics and Automation*, 2006.

[19] Jin Zhou and Baoxin Li. Robust ground plane detection with normalized homography in monocular sequences from a robot platform. In *IEEE International Conference on Image Processing*, 2006.