Mapping Techniques for AMPLE, an Autonomous Security Mobile Robot

Andras L. Majdik, Istvan Szoke, Mircea Popa, Levente Tamas, Gheorghe Lazea

Robotics Research Group, Technical University Cluj-Napoca, 400609 Cluj-Napoca, Romania (e-mail: andras.majdik, istvan.szoke, mircea.popa, levente.tamas, gheorghe.lazea}@aut.utcluj.ro)

Abstract: AMPLE is the acronym of the project called Autonomous Mapping of Polluted Environments. This paper introduces the concept of the project and presents some preliminary results of the ongoing research with the final goal of building an autonomous mobile robot that is capable of performing various tasks in unknown environments. To achieve this scope the mapping problem is an ineluctable one. This paper presents a visual mapping system which detects the same Speeded Up Robust Features (SURF) on the stereo pair images in order to obtain three dimensional point clouds at every robot location. The algorithm tracks the displacement of the identical features viewed from different positions to get back the robots positions. Also a mapping algorithm based on the laser system is presented which can detect the dynamic objects that are present in the robots field. The results of an indoor office environment experiments are shown.

Keywords: Autonomous mobile robots, Stereo vision, Lidar based mapping.

1. INTRODUCTION

This paper presents the theoretical aspects and the results obtained so far of an ongoing project, which final objective is to build a fully autonomous security mobile robot, that is capable of surveying unknown environments and detect dangerous gas leakages. The utility scenario of the project is shown on the figure 1. The robot operates in a chemistry laboratory where has to map the environment and detect an eventual accidental gas leakages. In this paper the issues of the navigation and mapping of the mobile platform are addressed.

We propose an architecture composed by two subsystems: a stereo vision and a light detection and ranging (LIDAR) system. The goal of the vision system is to perform simultaneous localization and mapping (SLAM) to achieve a high level of awareness of the robot while the LIDAR system is intended to solve the tasks of dynamic object detection and path planning. By fusing these two subsystems a highly intelligent robot will result, that is able to deal with the complex issues of the active SLAM problem.

This paper proposes the use of Speeded Up Robust Features (SURF) to detect the interest points on images and to track the displacement of these landmarks between the different robot positions. Similar visual odometer system was presented in (Konolige et al., 2007) for large scale outdoor terrains, but these use the Scale-Invariant Feature Transform (SIFT) detector. SIFT features are also used for construction of feature based maps like in (Se, 2005). The landmark detection problem is also a key issue in the simultaneous localization and mapping algorithms (Piniés, 2005).

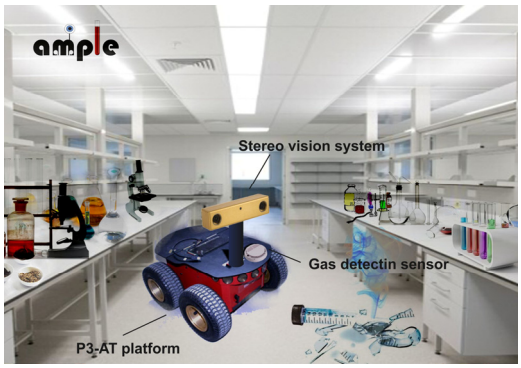

Fig. 1. Utility scenario: Performing surveillance task in chemistry laboratory.

Mostly the developed techniques for the laser based mapping were functioning in static environments, even if the actual mapping systems can handle with the noise from the odometer and noise from in the sensor data. However, if a person walks through the sensor range of the robot during mapping, the resulting map will contain evidence about an object at the corresponding location (Laugier, 2007). If multiple scans are made for the same map, and in the same time the person moved the resulting pose estimate will have serious errors. Due to this fact the localization part is also affected which can lead to the failure of the navigation system. If in environments which contain dynamic objects, a static mapping algorithm is applied, the accumulated errors

can be lead to serious failures. In this paper is described a method which handles the non-static objects from the mapped environments. The algorithm interleaves mapping and localization with a probabilistic technique to identify spurious measurements.

The paper is structured on a theoretical and an experimental part. In the theoretical chapters some ideas are introduced on which the developed experiments rely on.

2. THEORETICAL BACKGROUND OF THE VISUAL SYSTEM

2.1 SURF for Interest Point Detection

SURF is an abbreviation from Speeded Up Robust Features, it is a method, known in the field of computer vision to detect interest points on images. SURF was first introduced by (Bay, 2008), the advantage of this algorithm is the computational speed compared with other performing image feature detectors like Scale-Invariant Feature Transform (SIFT) presented by (Lowe, 1999), maintaining approximately the same or even better performances regarding: repeatability, distinctiveness and robustness.

Algorithm and the performances are largely discussed in (Evans, 2009) and (Bay, 2008), whereas hereby only a short overview is presented. The main reason why the algorithm decreases the computational time is because SURF uses integral images (Viola, 2001) as intermediate representation, in this way the sum of intensities over any upright, rectangular region is calculated with only four additions.

The SURF algorithm is based on the Fast-Hessian detector, which computes the determinant of the Hessian matrix:

$$H\left(p,\sigma\right)=\begin{bmatrix} L_{xx}\left(p,\sigma\right) & L_{xy}\left(p,\sigma\right) \\ L_{xy}\left(p,\sigma\right) & L_{yy}\left(p,\sigma\right) \end{bmatrix} \tag{1}$$

Where $L_{xx}\left(p,\sigma\right)$ is the convolution of the second order Gaussian derivative $\frac{\partial^2 g(\sigma)}{\partial x^2}$ of the image at point $p\left(x,y\right)$ and scale σ in the x direction. $L_{xy}\left(p,\sigma\right)$ and $L_{yy}\left(p,\sigma\right)$ are calculated similarly, these derivatives are known as Laplacian of Gaussian.

Another reason why the SURF algorithm performs fast is that it approximates the Laplacian of Gaussian with a box filter representation. The box filter allows a performance increase in time when they are computed on integral images.

2.2 Iterative Closest Point for Registration

The Iterative Closest Point algorithm also known as Iterative Corresponding Point (Rusinkiewicz, 2001) was first introduced by (Zhang, 1992) for registration of free form curves but (Besl, 1992) has proven that the method is valid also for point sets such as 3D point clouds.

The goal of the algorithm in the case of the developed experiments is to compute the rigid-body transform $\left(R,T\right)$ between two point cloud data sets, obtained at two consecutive robot positions detonated as X for the model point set respectively P for the data point set.

The method starts with an initial guess about the rigid-body transform between the two point clouds, thereafter the algorithm iteratively refines the transform by repeatedly generating pairs of corresponding points on the meshes, minimizing an error metric (Rusinkiewicz, 2001). The algorithm can be summarized as in (Bergström, 2007). The problem is to solve in a mean square sense the point matching error:

$$\min_{R,T}\sum_{k=0}^{N_p-1}\left(d\left(R*p_k+T,X\right)\right)^2 \tag{2}$$

Where the distance metric $d\left(p,X\right)$ between an individual data point p and the model points set X is:

$$d\left(p,X\right)=\min_{x\in X}\left\|x-p\right\|_2 \tag{3}$$

Let $\tilde{q}_i$ be a transformed data point q_i:

$$\tilde{q}_i=R\times q_i+T \tag{4}$$

The closest point in that yields the minimum distances to a transform point $\tilde{q}_i$ is denoted y_i such that:

$$\left\|\tilde{q}_i-y_i\right\|_2=d\left(\tilde{q}_i,X\right) \tag{5}$$

This relation is described by the closest point operator C :

$$y_i=C\left(\tilde{q}_i,X\right) \tag{6}$$

2.3 Stereo Vision Depth Estimation

This section focuses on the recovering of the 3D information from 2D images in order to estimate distances in the navigation application. The reconstruction of the third dimension from multiple images can be expressed in several ways (Demirdjian, 2002). The stereo sensor geometry and a method on how it is possible to calculate disparities form stereo images are presented.

A simple representation of a stereo camera system is given on figure 2, where C represents the centres of the lenses for the

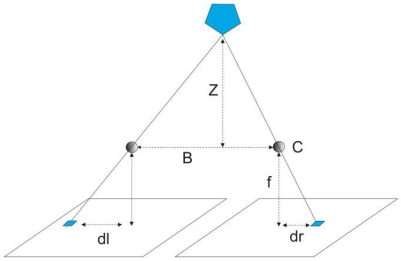

Fig. 2. Stereo geometry of the vision system.

right and left camera, f is the focal length, Z is depth in the coordinate system, and B is the baseline. By simple geometrical deduction the depth information Z of an object (blue pentagon) can be obtained in the following way:

$$Z = \frac{f \times B}{d} \tag{7}$$

Where *d* represents the disparity, the disparity can be defined as the difference between the coordinates (*dl* and *dr*) of the same feature in the left and right image.

The correspondence between images can be established in several ways. One of the fastest methods to compute disparities form stereo images is based on correlation analyses. Using this method the depth is computed at each pixel, a grey level around the pixel in the left image is correlated with the corresponding pixel in the right image. The disparity of the best match from the correspondence is determined using the sum of absolute differences (SAD) (Demirdjian, 2001), for a given square mask *m* is given by:

$$\min_{d=d_{\min}}^{d_{\max}} \sum_{i=-\frac{m}{2}}^{\frac{m}{2}} \sum_{j=-\frac{m}{2}}^{\frac{m}{2}} \left| R(x+i, y+j) - L(x+i+d, y+j) \right| \tag{8}$$

Where d_{min} and d_{max} are the minimum and maximum disparities, m is the mask size in which the search is done and R and L are the right and left images (row and column pixels). Once the disparity is determined the depth Z can be computed using (7) and knowing the camera constants (baseline, focal length).

Also the sum of squared differences (SSD) can be used which has the similar form, but the computational time for this is greater compared with the SAD. The disparity space is a projective space, and the noise in the disparity space is isotropic (Demirdjian, 2002).

2.4 Proposed Algorithm for Visual Odometer System

To accomplish a fast visual odometer system the advantages of the presented methods are combined. The drawback of the ICP is that it is too slow when it is computed for a great number of points. Therefore the SURF algorithm is used to detect only a few points, those points which are the optimum ones from the point of view of repeatability, distinctiveness and robustness.

In figure 3 the flow diagram of the algorithm proposed in this paper is presented. Instead of computing the real XYZ coordinate for every pixel using the Sum of Absolute Differences or other stereo correspondence algorithms, which are even more time consuming, in the first step of our method the SURF features are detected in both left and right images. For the common features from the previous step, the real XYZ coordinates are calculated for every feature considering the displacement in the right and left image, knowing the stereo sensor geometry and the camera constants (focal length, baseline). By choosing the right threshold for the features match the number of points considered can be limited.

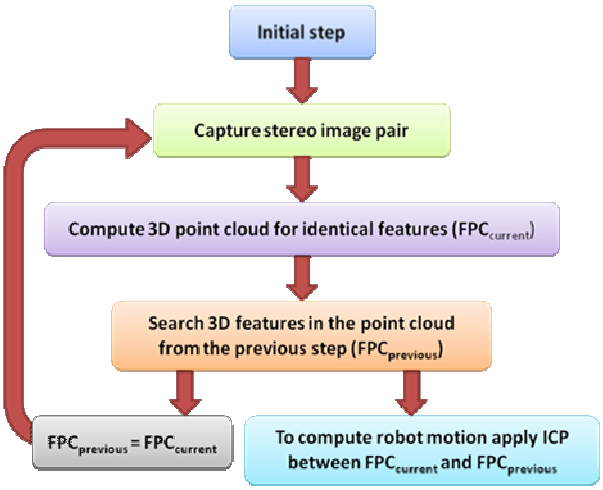

Fig. 3. The proposed algorithms flow diagram.

To obtain the translation and rotation between the feature point clouds for the consecutive steps the ICP algorithm is applied. Further on this algorithm will be referred as Surf ICP.

3. THEORETICAL BACKGROUND OF THE LASER SYSTEM

Gaussian mixture models are widely used in data mining, pattern recognition, machine learning and statistical analysis. In order to build complex probability distributions, mixture models can also be used to cluster data. The problem of finding clusters in a set of data points can be resolved by a non-probabilistic technique, the K-means algorithm.

Expectation-Maximization algorithm is a powerful method for finding maximum likelihood solutions for models with latent variables (Dempster, 1977).

3.1 K-means Clustering Another Subsection

The main usage of the method created by (Lloyd, 1982) is to identify groups, or clusters of data points in a multidimensional space. The inputs of the method are a data set $\{x_1,.....,x_N\}$ consisting of N observations of a random D-dimensional Euclidean variable x, and k the number of clusters which we would like to obtain.

In common words a cluster can be described as comprising a group of data points whose distances between the points are less than the distances between the points outside from the cluster.

We note μ_k as the representation of the centres of the clusters. Finding an optimal solution to the data set clustering problem is not straightforward. A relative good solution has been found when the sum of the squares of the distances of each data point to its closest centre is a minimum. For each data point x_n is introduced a corresponding set of binary

indicator variables $r_{nk} \in \{0,1\}$, where $k = 1,....K$ describing which of the K clusters the data point x_n is assigned to.

Each point can be assigned just to one cluster, this can be described as follows if $r_{nk}=1 \wedge r_{nj}=0$, then $j \neq k$, this is called the 1-of-K coding scheme.

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \left\| x_n - \mu_k \right\|^2 \quad (9)$$

The objective function (9) represents the sum of the squares of the distances of each data point to its assigned vector μ_k. The solution is optimal when the values found for r_{nk} and μ_k are minimizing J.

These values can be obtained developing an iterative procedure, it involves two successive steps in which r_{nk} and μ_k are optimized. In the beginning initial values are chosen for μ_k , J is minimized with respect to r_{nk} , keeping the μ_k fixed. In the following step J is minimized with respect to μ_k, r_{nk} is kept fixed.

These two stages correspond to E (expectation) and M (maximization) steps of the EM algorithm, which steps are repeated until convergence.

3.2 Expectation-Maximization algorithm

The goal of the EM algorithm is to find maximum likelihood solutions for models having latent variables. The algorithm is summarized in Table 1.

Table 1. Description of the EM algorithm

1. Chose initial values for Θ^{old}
2. **E step** Evaluate $p(Z \mid X, \Theta^{old})$
3. **M step** Evaluate Θ^{new} , where

$$\Theta^{new} = \arg\max_{\Theta} \Phi(\Theta, \Theta^{old})$$

we know that,

$$\Phi(\Theta, \Theta^{old}) = \sum_{Z} p(Z \mid X, \Theta^{old}) \ln p(X, Z \mid \Theta).$$

4. Check the convergence of the parameter values, if the convergence is not achieved

4. EXPERIMENTS

In this chapter are presented the experiments, which were carried out in the laboratory building of the Technical University of Cluj-Napoca and are based on the theory presented above. For all experiments a Pioneer 3 All-Terrain mobile robot was used, made by Mobile Robots Inc. equipped with a Bumblebee2 stereo camera, made by Point Grey Research Inc and a SICK 200 Laser Measurement Sensor.

Fig. 4. Left: SURF features on the right images captured at the initial position; Right: SURF features on the right images captured after robot motion.

4.1 Obtaining Point Clouds form Stereo Images

To demonstrate the advantages of the algorithm proposed a case study on real images is shown and the performances are presented.

In this experiment a stereo pair picture was taken (size of 640x480), after this the robot moved forward 0.3 meter and rotated 10 degrees, when a another stereo pair was captured. A point cloud was computed in both positions with the Sum of Absolute Differences (SAD) algorithm. SAD is one of the fastest stereo correspondence algorithm, but it is affected by errors.

If the sum of absolute differences algorithm presented in the theoretical chapter is applied on a stereo image pair a disparity map is obtained also known as depth map where the big disparities mean closer objects and low values far objects.

Using the stereo sensor geometry, (7) and knowing the camera constant parameters, the desired maximum and minimum detectable distances can be set. As the environment in which the experiments were done is an office the following distances were chosen:

$$range_{\min} = 0.5(m) \Leftrightarrow disparity_{\max} = 60 \quad (10)$$

$$range_{\max} = 3(m) \Leftrightarrow disparity_{\max} = 10 \quad (11)$$

If the disparity map is computed with (8) the depths, the real world Z coordinates can be easily switched back. After this step the real world X and Y are computed for each Z:

$$X = \frac{u \times Z}{f}, u = centerCol - col \quad (12)$$

$$Y = \frac{v \times Z}{f}, v = centerRow - row \quad (13)$$

The result of the algorithm is a 3D point cloud. The generated point cloud from the stereo pair of which the right image is presented in figure 4, left is shown in figure 5, left. After some filtering the flower is detectable and parts of the blue chair, but the 3D map obtained is badly corrupted by false points. A similar point cloud was obtained for the other robot position. To detect the displacement between the two point

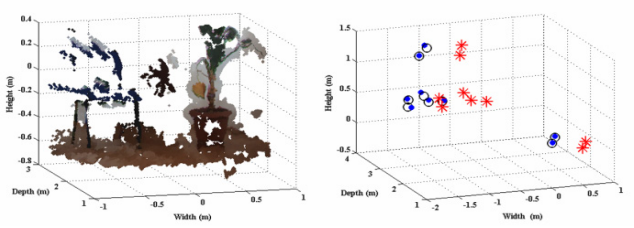

Fig. 5. Left: Initial point cloud; Right: Displacement between the interest point clouds.

clouds the ICP algorithm presented in the previous chapter was applied on the point clouds.

In each experiment the model in the ICP algorithm was considered to be the point cloud from the first robot position and the data point set was considered the points obtained from the photo pair in the second position. The transformations obtained in this way are very precise, but time consuming.

The ICP algorithm is slow if it is applied for a big number of 3D points. By an effort to reduce the dimensions of the data sets the odometer data is used. The point cloud dimension is reduced by subtracting those points that are predicted by the odometer to be outliers. After the point clouds are reduced the ICP algorithm is applied, the results are significant, in which case 67% of the points are eliminated and the time needed to run the ICP is only 31% of the initial case, where no subtraction was done. Still a computational time needed is too long (37 sec), so another solution was developed.

The algorithm was presented in the previous chapter and it was named Surf ICP. In the first step the mutual interest points are detected in both left and right images, the results are shown on the right image in figure 4, left. On the right picture in this figure are visible the same image features detected on the next image set taken after the robot moved (second robot position). The interest points are plotted with markers that have different forms and colours in order to be easily detectable. It can be concluded that the features are detected in accurately and right positions on both image sets. The threshold of the feature match was set to detect only the best nine features which are present on all of the images taken. After the surf feature match is done the real XYZ coordinate is computed in each position from the left and right images knowing the camera constants and the stereo geometry.

These much reduced point clouds are plotted in figure 5, right: with red stars are the features from the initial stereo pair and with black dots the transformations of them after applying the displacement and rotation matrixes obtained with the ICP algorithm. The black circles are the Surf features with the real XYZ coordinates obtained from the stereo pair after robot motion. It can be concluded that in this way the algorithm is much faster and much more precise. To calculate the surf features and the matching of them takes approximately another 0.05 seconds.

Table 2. The comparison of the results

Test parameter	ICP	Odo ICP	Surf ICP
Time (sec)	118	37	0.17
Nr. Points	73186	24013	9
Time win (%)	---	31.36	0.14
Eliminated points (%)	---	67.18	99.98

The comparison of the results is summarized in Table 2. The advantage of the developed algorithm is that only a limited number of points are taken into account, in this case nine, the time is significantly reduced; only 0.14 % of the initial time is needed. Selecting only the best points from the repeatability, distinctiveness and robustness point of view the computational time can be reduced by the order of scales.

4.2 Visual System for Building Feature Based Map

The odometer error is a cumulative one and is dependent on the environment in which the test is done (especially the rotations of the robot is dependent on the surface nature). This is displayed in figure 6, the blue crosses represents the positions of the robot based on the wheel odometer. The little gray dots are a horizontal cut of the office in which the experiment was done, the cut was obtained by a laser in other applications, and in this plot is just a supporting mark to understand the path of the robot. The L shape objects are the desks in the office and in the middle is a column. It can be observed that even after a short motion the odometer error is getting significant and is accumulating the error. In conclusion the positions of the robot based on the odometer sensors are not real and it is not reliable, because the robot did not hit the desks in the office. To develop any robot navigation algorithm the error of the robot position must be corrected. In this paper a visual odometer system was developed.

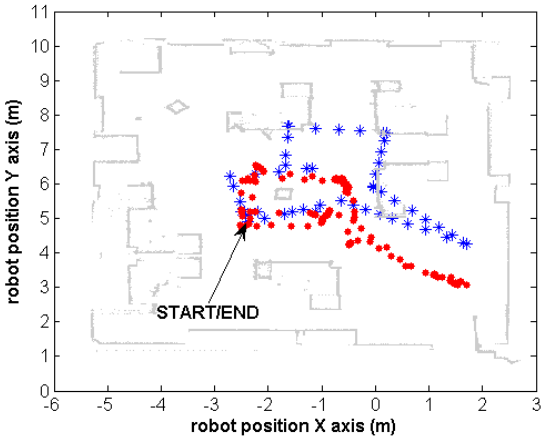

Fig. 6. Robot positions obtained with the visual odometer system developed (red dots) and the wheel odometer (blue crosses).

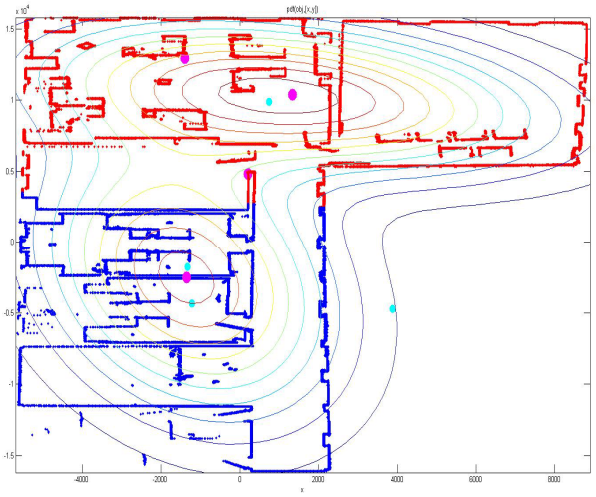

Fig. 7. Two clusters were created for the whole map.

The architecture of the proposed visual odometer system is presented in figure 3. The idea is to search for the same SURF features on the both images of the stereo pair. From the stereo displacement of them the features real world 3D coordinates can be calculated and a feature point cloud can be generated at every robot position.

As in the previous example the SURF landmarks from the current point cloud are searched in the point cloud computed in the previous step. For the found landmarks the Iterative Closest Point algorithm is applied to get the displacement between the two data sets. If the algorithm is applied not only for two positions (as in the first experiment) but for the whole robot path the position of the robot is corrected. This is valid from the second observation position, because the algorithm needs an initial set to detect the first feature point cloud. The corrected positions can be seen in figure 6; they are plotted with red dots. It can be concluded that the position of the robot is corrected.

If the 3D features point clouds are registered in a general map at every position of the robot, the feature based map of the entire environment is built. Feature based map can be used to solve the kidnapped robot problem in which the robot must recognize the initially unknown position of the robot or other simultaneous localization and mapping related issues.

4.3 The Light Detection and Ranging Mapping System

Two algorithms are inter-processed in order to obtain a highly accurate map. First the clusters are built (figure 7) from the acquired data set using the K-means algorithm. The obtained parameter values are passed to the general EM algorithm which calculates more exactly the parameters of the input distribution.

On figure 8 is presented the created clusters for the part from the whole map. Using the two algorithms presented, two clusters were obtained for the input from the upper left corner of the whole map. The cluster represented with yellow is the basis cluster, the other one, represented with red will be the

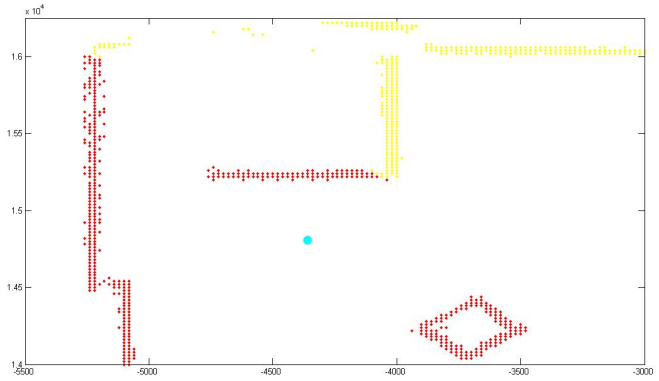

Fig. 8. The yellow cluster is the static one, the red one will be the input for the modified EM algorithm.

input for the modified EM algorithm which will detect the spurious measurements from the map.

On figure 9 in the upper plot is described the clustering of the most left part of the whole office room which will be clustered in separate objects. Using the *kmeans* algorithm the obtained clusters are not separated in an optimal way for our objective in this work. The points seen in the left of the plot represent the left wall of the room, points which should be in the same cluster, representing the same object in the map.

In the second plot of figure 9 the mean values for each cluster is used as initial parameters for the Expectation-Maximization algorithm, which results can be seen in the lower plot. The obtained clusters are optimally separated, representing good determined objects. The crossed points represent the left wall, the upper the wall from the north of the room, and the remaining points are part from an office desk.

The described procedure is effectuated for each part of the office map, in this way all the objects from the 2D laser map will be correctly separated and applying the dynamic objects detection algorithm, the dynamic objects will be more easily detected and removed from the map in order to obtain a correct prior static map.

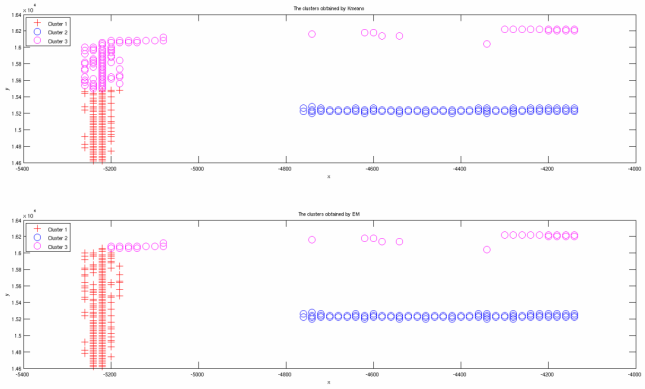

Fig. 9. The difference between the obtained clusters after applying the 2 algorithms.

5. CONCLUSIONS AND FURTHER WORK

The paper contributes with an algorithm that detects the common SURF features on the image pairs and considers them landmarks of the environment. The system tracks the displacement of these landmarks between the different robot positions. The advantage of the developed visual odometer system is in decreasing the computational complexity by selecting only the best points on the images at every position from the point of view of repeatability, distinctiveness and robustness. By reducing the dimension of the data sets implicitly computational time is reduced.

Further work is intended to be done regarding the robustness of the algorithm developed. The authors intend to develop an adaptive method to select a fixed number of features at every position. In some cases it is possible that the algorithm is not convergent if between two images sets the difference is to big the common feature cannot be detected. In these situations only the wheel odometer can be used.

Regarding the LIDAR mapping system future work is intended to be done after clustering the whole map. The clusters will be compared and the most alike ones will be classified, in this way the computing time will be efficiently reduced. The next step will be to create cells in each cluster, cells which will have dynamic size, depending of the object which will undertake in it. This latter step is necessary to apply the Markov Decision Problem algorithm to obtain more efficacies in the path planning process.

In order to accomplish the proposed objectives of the AMPLE project the authors intend to interleave the visual mapping system and the laser path planning system will solve the active simultaneous localization and mapping problem, which is the core issue of the AMPLE project from the mobile robotic perspective and will lead to a fully autonomous mobile platform, capable of navigating in an unknown environment.

6. ACKNOWLEDGMENT

This work was done within the PRODOC (Project of Doctoral Studies Development in Advanced Technologies) project at the Technical University of Cluj-Napoca, supported by European Social Fund, contract No.: POSDRU/6/1.5/S/5.

REFERENCES

Bay, H. et al. (2008), SURF: Speeded Up Robust Features, *Computer Vision and Image Understanding*, Vol. 110, No. 3, pp. 346--359

Bergström, P. (2007), A method suitable for reiterated matching of surfaces, *Conference on Geometric Design and Computing*, San Antonio, United States

Besl, P. J., McKay, N. D. (1992), A Method for Registration of 3-D Shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 14, pp. 239-256,

Demirdjian, D., Darrell, T. (2002), Using multiple-hypothesis disparity maps and image velocity for 3-D motion estimation, *International Journal of Computer Vision*, Vol. 47, pp. 1-3, Springer

Demirdjian, D., Darrell, T. (2001), Motion Estimation from Disparity Images, *International Conference on Computer Vision*, vol. I, pp. 213-218

Dempster; A. P., Laird, N. M., Rubin, D. B. (1977), Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal of the Royal Statistical Society*, Vol. 39, No.1, pp. 1-38

Evans, C. (2009), Notes on the OpenSURF library, Technical report nr. CSTR-09-001, University of Bristol

Konolige, K., Agrawal, M., Sola, J. (2007), Large Scale Visual Odometry for Rough Terrain, *Proc. International Symposium on Research in Robotic*

Laugier, C., Chatila, R. (2007), Autonomous Navigation in Dynamic Environments, *Springer tracts in Advanced Robotics* 35

Lloyd, S. P. (1982), Least squares quantization in PCM, *IEEE Transactions on Information Theory,* 28(2), 129-137

Lowe, D. G. (1999), Object recognition from local scale-invariant features, *International Conference on Computer Vision,* pp. 1150-115

Piniés, P., Tardos, J. D. (2008), Large Scale SLAM Building Conditionally Independent Local Maps: Application to Monocular Vision, *IEEE Transactions on Robotics*, pp. 11094--1106

Rusinkiewicz, S., Levoy, M. (2001), Efficient Variants of the ICP Algorithm. Efficient Variants of the ICP Algorithm, *International Conference On 3-D Digital Imaging And Modeling*

Se, S. L. (2005), Vision-based global localization and mapping for mobile robots, *IEEE Transactions on Robotics*, pp. 364—375

Viola, P., Jones, M. (2001), Rapid object detection using a boosted cascade of simple features, *Computer Vision and Pattern Recognition*, ISSN: 1063-6919

Zhang, Z. (1992), Iterative Point Matching for Registration of Free-form Curves, Institut national de recherche en informatique et automatique, Rapports de Recherche No. 1658