A study of 2D features for 3D visual SLAM

Jose L.Muñoz
Computer Science Research Institute University of Alicante, Spain Jose.va18@gmail.com

Daniel Pastor
Computer Science Research Institute University of Alicante, Spain aebsubis@gmail.com

Pablo Gil
Computer Science Research Institute University of Alicante, Spain pablo.gil@ua.es

Santiago Puente
Computer Science Research Institute University of Alicante, Spain santiago.puente@ua.es

Miguel Cazorla
Computer Science Research Institute University of Alicante, Spain miguel.cazorla@ua.es

Abstract— The use of 2D features in computer vision has had a great impact in lots of applications. For example, the combination of those features together with 3D data has helped to solve the Simultaneous Location And Mapping (SLAM) problem in real time. Nowadays, there are several interesting feature detectors and descriptors with different characteristics: processing time, robustness against lightning conditions, changes in point of view, scale, etc., and every day it appears more and more. In this paper, a deeper study about several of these detectors and descriptors is done. Several interesting graphs where we can separate distances with respect to axis and angles have been analysed. This study helps to make decisions about which are better for a given application. The study has been done with a low cost sensor as Kinect installed on robotics arm to control the movements with accuracy. Furthermore, finally, real scenario reconstruction is shown using Kinect camera and the visual features analysed in the study.

Keywords: *visual features; 3D data; RGB-D data; SLAM.*

I. INTRODUCTION

One of the central research themes in mobile robotics is the determination of the movement performed by the robot using its sensors information, which is usually referred as egomotion [1], and also as registration. The method proposed in this work presents some improvements calculating the egomotion which can be used for automatic map building and Simultaneous Location and Mapping (SLAM) [2]. Our main goal is to perform six degrees of freedom (6DoF) visual SLAM in semi-structured environments, i.e., man-made indoor environments. Egomotion can be computed using two main approaches: point-based and feature-based. In the point-based approaches, the most widely used is the Iterative Closest Point (ICP [3]), but it does not provide good results in the presence of outliers. Using feature-based (from a RGB-D camera) and using the RANdom SAmple Consensus (RANSAC) method [4] provide a best egomotion calculation.

In this work we are interested in determining which visual features provide better results for egomotion calculation. Others works (like [5]) made a similar study, but we extend that study, using a robotic arm in order to get a ground truth, not present in the former. In order to prove the accuracy of our study, we follow a similar approach than [6] to solve the SLAM, but incorporating some improvements in the egomotion calculation.

The remainder of this paper is organized as follows: Section 2 describes the overall architecture of the platform employed to analyze the detector and descriptor methods for visual SLAM in 3D scenes. Section 3 comments, briefly, detectors and descriptors analyzed. Section 4 shows the experiments to evaluate the detectors and descriptors methods presented in Section 3. Section 5 shows examples of reconstruction using TORO and the detectors and descriptors which have provided the best results in Section 4. Finally, some conclusions are discussed.

II. EXPERIMENTAL SETUP WITH KINECT DEVICE

In order to realize the different experiments of this paper, a robotics system has been used. A 7 degrees freedom Mitsubishi PA-10 robot arm and a Kinect device mounted on the robot’s end-effector as an eye-in-hand configuration (Fig. 1) have been used to evaluate the different feature detectors and descriptors. This detectors and descriptors extract interest points of scene images and identify the appearance of these points, respectively.

Some movements have been planned to acquire some series of images from the scene of a laboratory of our University. Each serie of images have been captured to model different situations and how the movement affects the behaviour of Kinect in different ways. The series of image represent sequence of movements in different directions such as lateral movements (x-axis and y-axis) approach movements (z-axis) which generate change of scale and rotation movements in all axes. All these movements cove all different viewpoints and illuminations changes due to perspective. Thus some different distances and angles are considered. In Section 3 and 4, some examples of the evaluation of detectors and descriptors extracted from each sequence images are commented. In addition, in Section 5, an example of the 3D visual SLAM using 6 degree of freedom is shown and explained.

Furthermore, in this work, OpenCV (Open source Computer Vision) [7] and PCL (Point Cloud Library) [8] have been used to detect features and represent 3D information about the environment acquired from Kinect. Later, this spatial information of the environment is used to build maps, scene models for navigation and robot locations using the mobile robotic platform.

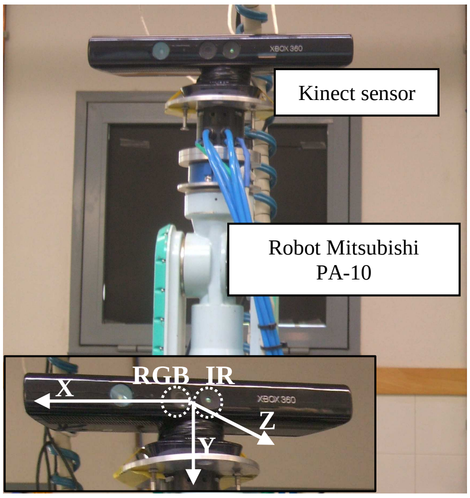

Fig. 1. System architecture to evaluate detectors and descriptors.

Detection of keypoints is a task of finding groups of pixels and/or patches. In computer vision, the keypoints detection is one of the oldest and most widely studied techniques, in which matching processes among images are required, even though changes of scale and orientation are present in the images. These matching processes allow us to align some images to construct a 3D model which can be used to estimate camera pose like in SLAM applications.

In literature, there is a lot of kind of detectors [9][10]. Early techniques use as keypoints specific locations in the images, such as corners, points of edges, etc. The keypoints are distinguished from other features in the image because they can be accurately tracked using local search techniques through a sequence of the images of a same scene acquired from a moving camera. Nowadays, some keypoint detectors have been successfully used to extract visual landmarks in visual SLAM applications. Until now, keypoints detectors have been almost always acquired from 2D cameras. Few studies about the keypoints detectors and descriptors computed from 3D cameras to build maps based on dense 3D data for SLAM have been done [11].

In this work, we have evaluated the most popular detectors in combination with the popular descriptors used in SLAM applications but this time, innovatively, using a RGB-D camera to acquire images with color and depth

A. Detectors

On the one hand, a corner/keypoint detector should satisfy a set of basic criteria which determines its quality depending on the application. On the other hand, a corner/keypoint detector should be extracted as rapidly as possible. The runtime depends on the computational cost of the method implemented (see Table I) as well as RGB-D device (i.e.: Kinect) and the computer used. All experiments were carried out on an Intel Core2 Duo T7500 2.2Ghz of CPU with 2GiB of memory RAM.

In general, it is important that the keypoints are robust in the presence of noise, they are properly located in the image and the pixel assigned is accurate and only the keypoints which represent interest point are detected and no false points. Furthermore, the method to detect must be efficient and work well with changes in viewpoint (rotations, translations, scale, perspective, etc.) (Fig. 2 and Fig 3). Among they are:

Harris [12]: This detector searches local maxima in rotationally invariant scalar derived from an auto-correlation matrix, H, called Hessian matrix. The eigenvalues and eigenvectors of H, represent the magnitude and directions gradients, respectively.

TABLE I
RUNTIME COST

Method	Type	Num. Features (average)	Num. Features (min/max)	Average Time (ms.)
Harris	DT	352	254/427	43.42
Shi-Thomasi	DT	888	735/1116	45.42
MSER	DT	222	163/266	298.97
SIFT(DoG)[*]	DT	1417	1167/1715	606.2
SURF[*]	DT	1081	904/1307	275.03
*SIFT***	DS	352	254/427	494.93
*SURF***	DS	352	254/427	58

[*] Only detector has been used
[**] DS: Descriptor has been used with the same detector: Harris
The average time and average of features correspond to image sequence of Figure 3

$$k = \lambda_1 \cdot \lambda_2 - \alpha(\lambda_1 + \lambda_2) = \det(H) - \alpha \cdot \mathrm{trace}(H) \qquad (1)$$

Harris work well with rotations and lighting changes but not with scales. This detector is known as imp-Harris or Harris-Laplace [10] when Gaussian weighting window is used to do the detector invariant to scale.

Shi-Tomasi [13]: It is a variation of the above detector. Its behavior is better than Harris when there are changes of scale. The smallest eigenvalue is used to consider keypoints in the image.

$$k = \min(\lambda_1, \lambda_2) \qquad (2)$$

MSER (Maximally Stable Extremal Regions) [14]: It is used to extract regions, blobs. The detector measure how stable these regions were when the intensity change. This detector is invariant to rotations, translations and scale but not perspective.

In visual SLAM is very important the tracking of the keypoints in a sequence of images. Thus, to evaluate the detectors for visual SLAM, each detection method is applied for each image acquired from sequence of movements where the viewpoint changes. The keypoints detected in one of the images, I_k, are searched in the following image in the sequence, I_{k+1} for each method. To do this, 3D indoor scenes have been used where planar and non-planar objects can appear in the environment.

Fig. 2 and 3 show the number of keypoints that do not disappear (i.e. points that not are lost, that is to say survivor points) when the camera changes its viewpoint due to movement. Therefore, when the detectors are only considered, SIFT detector detect more keypoints than any other. However, SIFT lost about 40% (displacements) 32% (rotations) of those when the initial data are compared with the final data. Otherwise, Shi-Tomasi has similar percentages (42% and 27%) but it also has a runtime of 45ms. versus 606ms. of SIFT for rotation movements.

A perfect detector should detect the same points anywhere in the image sequence and its runtime should be as small as possible to ensure the real time execution.

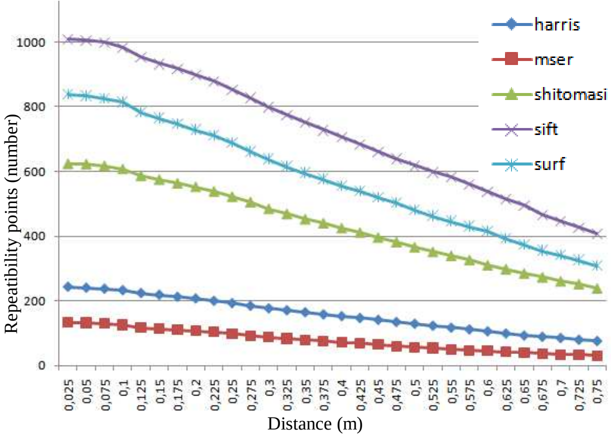

Fig. 2. Movements as path of displacements in the three axes.

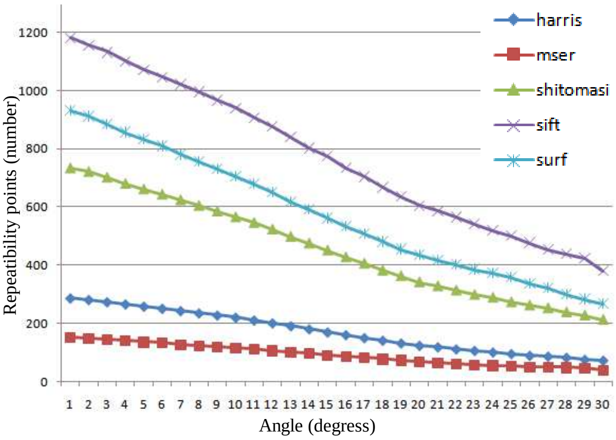

Fig. 3. Movements as path of rotations in the three axes.

B. Descriptors

SIFT (Scale-Invariant Feature Transform) [15] is based on the detector DoG (Difference of Gaussians) plus descriptor. The method uses a pyramidal structure which is obtained from the difference of the image convolved with different scale Gaussians. Later, local maxima and minimum of this pyramidal structure are used as keypoints. The orientation of this keypoints and their neighboring points in a local environment are used as descriptor. Thus, the descriptor of each keypoint is defined by a gradient orientation histogram.

SURF (Speeded Up Robust Features) [16] is based on Hessian matrix for the detection of keypoints. In addition, similarly to SIFT, this method uses a scale space by means of pyramidal structure of image convolved with Gaussians of different size. The descriptor is composed of orientations of keypoints. These orientations are computed in the vicinity of keypoint. To do this, this neighborhood masks are convolved with 2D Haar wavelet filter in different directions and after they are summed.

III. Performance Evaluation for 3D Scenes

The descriptors define the appearance of keypoints. SIFT and SURF implements its own descriptor. However, the descriptors of SIFT and SURF can be used combined with other detectors (Fig. 2). So, each of these descriptors can be jointly used with a foresaid detectors (see section 3). In this section, the features are extracted with the detectors previously discussed and combined with the descriptors of SIFT and SURF to add information of neighborhood of the keypoint. The evaluation of these combinations is done using the criterion of repeatability. Whether, the set $\{f_1,f_2,..f_N\}$ represents the features extracted of a set of images $\{I_1,I_2,..,I_N\}$, the i-step represents what images are matched then for example, the matching among images applying step of i=2 can be defined as a set of matching from f_1-f_3, f_2-f_4 until $f_{N-2}+f_N$. Thereby, the criterion of repeatability can be made dependent of i-step, as follows:

$$r(i) = \frac{\sum_{k=1}^{N-i} \left| f_k - f_{k+i} \right| / nf_k}{N-i} \cdot 100 / \ \forall i=1 \ldots N \qquad (3)$$

where $|f_k\text{-}f_{k+i}|$ are the number of pairwise features tracked from the image 1 to the k-image with i-step. In addition, nf_k is the number of features of the k-image for each matching. Moreover, the number of samples according the step, N-i, has to be considered in percentage format.

In order to evaluate the different combination of detector-descriptor, 6 sequences of images have been acquired (Fig. 4 and 5). Displacements in the x-axis and y-axis as well as rotations in any axis define viewpoint changing among images. The displacements in z-axis and rotations in x-axis and y-axis define scale changing and perspective. On the one hand, each sequence of rotation images consists of 30 images (1 image- 1 degree) and each image has been captured with the camera following a semicircular trajectory until 30 degrees. On the other hand, each sequence of displacement images consists of 30 images (1 image - 0.025m.) so the total displacement of the camera between first and last image is 0.75m. In addition all images haven been captured without varying lighting conditions and an indoor environment. Moreover, there were no moving objects in the scene, with the exception of the robot which was programmed to move the Kinect camera and capture the images, autonomously. The images were captured at 640x480 pixels/30fps with IR and RGB sensors. They have 6.1mm and 2.9mm of focal length, respectively, according [18] so the calibration matrix K of Kinect is known. Moreover, Kinect has useful range of working distance between 0.5m. and 5m.

A. Moving as pure translation

Fig 4 shows the percentage of survivor features when the camera is moved as pure translations in one of the three axes. The movement occurs in step of 2.5cm.

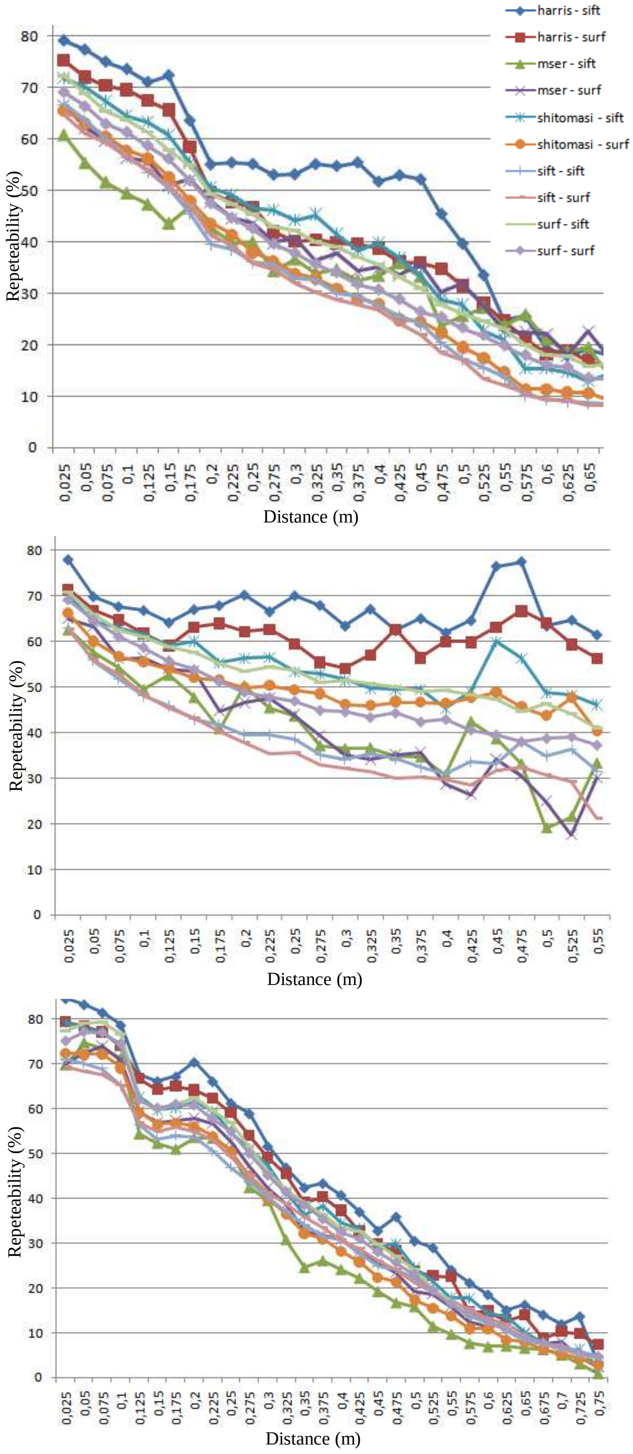

Fig. 4. a) x-displacement. b) y-displacement. c) z-displacement.

In general, Harris with SIFT or SURF are the best combinations in terms of repeatability for all the tests of movements. Both provide the highest percentage of survivor points. The following best combinations are ShiTomasi-SIFT and SURF-SIFT.

The comparison of repeatability shows instability when there are fewer samples to average. This case occurs when i-step is quite high, e.g. i>20 for rotations (Fig. 5).

B. Rolling as a pure rotation

Fig 5 shows the percentage average of survivor features when the camera is moved as pure rotations in one of the three axes. The movement occurs in step 1 degree.

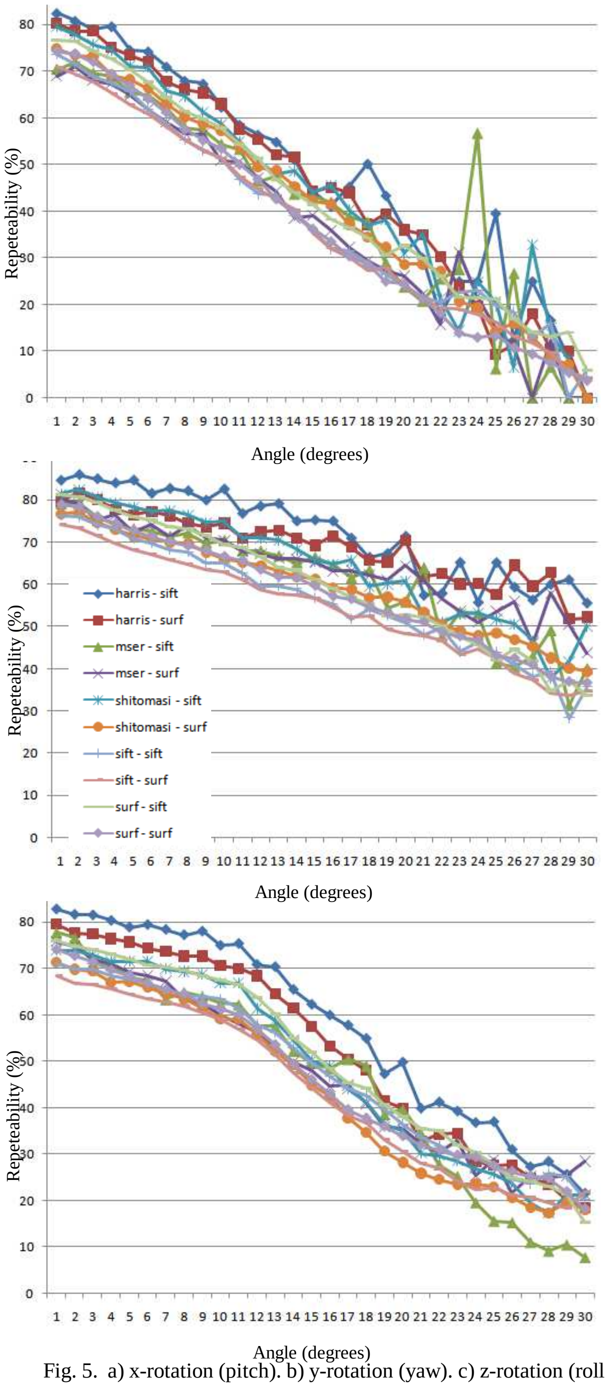

Fig. 5. a) x-rotation (pitch). b) y-rotation (yaw). c) z-rotation (roll).

The methodology to test the detectors and descriptors, previously, commented in the section is as follows:

Algorithm 1: Test detectors-descriptors

```
-For k=0 to N/∀ N robot positions
    a. The robot is moved to k-position /$T_k$∈{Transformation of displacement or rotation}
    b. The images, $I_k$, from Kinect are acquired/K is matrix of calibration of Kinect and where $I_k$ is a depth map from IR, $z_i$, and colour map from RGB, $c_i$ /∀ pixel-i □ $I_k$.
```

c. The features as keypoints-descriptors, f_k, are extracted from I_k.

-In order to compare the features, f_k, which are maintained throughout the sequence of the images, I_k/ k=1…N when T_k is applied to the robot, this is done:

a. The matching between images is done using the descriptors. Thus the matched features f_k' and f_{k-1}' are obtained.
b. The matched features f_k' of I_k are reprojected to I_{k-1} as $f_k' = T_k^{-1} \cdot f_{k-1}$
c. The reprojection error is used to check the matching among features. This is computed as the Euclidean distance, $|f_k' - f_k| < d_{max}$.
d. The percentage of survivors features are computed according Eq(3)

IV. Results in 3D Visual SLAM

To evaluate our system, we have used Shi-Tomasi's detector, because the percentage of survivor features is similar to SIFT and SURF detectors and it also is much faster than those (13 and 6 times faster than SIFT and SURF, respectively). Moreover, Shi-Tomasi is as fast as Harris detector and it provides more features than that (2.5 times).

Investing the influence of descriptors combinated with the detectors, we observed that SIFT combined with Harris or ShiTomasi provides good results (see Section 3). However, these combinations are slow compared with the same detectors with SURF when the mapping must be performed in real time with Kinect. Furthermore, Harris or ShiTomasi detectors with SURF provide a combination very stable in terms of repeatibility when the image sequences are very different (e.g. large displacements and rotations).

SLAM with both combinations, Harris-SURF and ShiTomasi-SURF have been made. We have preferred to choose ShiTomasi-SURF instead of Harris-SURF because it ensures a greater set of features for each image. This way, the computing of pose is more reliable and we avoid the positioning loss during the mapping in real-time.

In this section, we present the results and the accuracy of our SLAM algorithm when ShiTomasi-SURF is chosen. Algorithm 2 presents the method followed to solve the SLAM. First, we detect the features in two consecutive images and calculate the descriptors. Then, only the best features are selected. Selecting the best features follows a elimination criteria based on: Euclidean distance between descriptors must be below a given threshold; a feature only must be matched with one feature in the other image, if not, the feature is not selected; if a feature is matched with other feature, the other feature must be matched only with the former one; given two features matched, the matching must follow a similar projection that the whole set (outlier rejection). From these features, we calculate the whole set of possible transformations, using all the possible transformations between all the features in both images. Each transformation is evaluated using a Euclidean distance between the feature coordinates in one image and the reprojection (applying the transformation being evaluated) of the features coordinates from the other image to the current. The transformation with less total Euclidean distance is selected. The transformation provides the egomotion done by the robot between the two consecutive poses. However, this transformation has an error which must be reduced using a SLAM method.

In order to solve the SLAM, we apply the Tree-based netwORk Optimizer (TORO) method [17]. This method uses a tree structure to define and efficiently update local regions at each iteration by applying a variant of stochastic gradient descent. The nodes of the structure are the robot poses and the edges between nodes are the transformations obtained in the last step. TORO, once applied, provides the corrections of the errors accumulated in the path. Figure 7 shows an example of reconstruction. At the left part we can observe the reconstruction without the SLAM correction and right part after SLAM was applied.

Algorithm 2: SLAM

```
-To detect the features for each two images ($f_{k-1}$ and $f_k$)
-To search the matching between consecutive images (pairwise image):
    a. According to the criteria commented in section 4, the features
       are filtered and voted.
    b. The best n=15 filtered features are chosen.
-To compute the transformation between consecutive images (pairwise
image) with filtered features:
    a. The r-combinations from the given set of n=15 filtered features
       are computed by: C(n,r)=n!/r!(n-r)! where r=7.
    b. The transformation T for each C(n,r) is calculated.
-To make the matching between images, the whole set of
transformations T for each set of 7 points is evaluated as follows:
    a. $f_k' = T_k^{-1} \cdot f_{k-1}$ is done for each T of C(n,r).
    b. The reprojection error, $|f_k' - f_k|$ is used to check the T.
    c. The T which minimizes the Euclidean distance between $f_k$'
       and $f_k$ is chosen.
-To build the graph for SLAM, each edge is created as a transformation
T for these matched features.
-TORO is computed to optimize the alignment of transformations
among images
```

V. Conclusions and future work

In this work we have presented a study of different visual detectors and descriptors, showing its validity for the egomotion calculation. The study uses a robotic arm in order to get a ground truth. The use of the robotic arm allows us to separate errors in the 6 degrees of freedom: translation and rotation, with three main axes each.

Several detector and descriptors have been tested, not only with their accuracy but also their computation time. The number of features returned is also an important key factor, as less number of them could provide bad matching. As a result of the study, the combination of ShiTomasi detector together with SURF descriptor provides the best results.

We have used that combination for building a complete SLAM application, in which we use the matching process with those features as the way for egomotion calculation. Results of the SLAM show the validity of our method.

As future work, we plan to continue incorporating new detectors and descriptors, in order to obtain the limitations of each of them. We also plan to speed up the

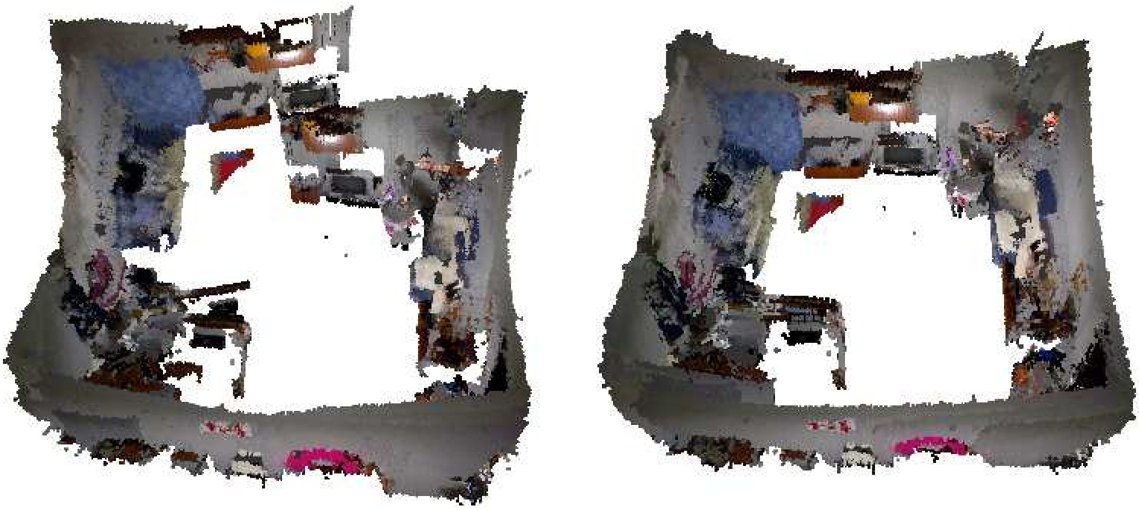

Fig. 7. a) Egomotion of frame to frame tracking (no loop closures). b) Result after optimization with TORO (loop closures)

complete SLAM problem, using some promising GPU implementation of some detectors.

VI. Acknowledgment(s)

These authors want to express their gratitude to Spanish Ministry of Science and Technology (MYCIT) and the Research and Innovation Vice-president Office of the University of Alicante for their financial support through the projects DPI2009-07144 and GRE10-16, respectively.

VII. References

[1] O. Koch, S. Teller, Wide-area egomotion estimation from known 3d structure, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR '07, pp. 1-8.

[2] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, M. Csorba, A solution to the simultaneous localization and map building (slam) problem, Robotics and Automation, IEEE Transactions on 17 (2001) 229–241.

[3] P. Besl, N. McKay, A method for registration of 3-d shapes, IEEE Trans. on Pattern Analysis and Machine Intelligence 14 (1992) 239–256.

[4] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (1981) 381–395.

[5] A. Gil, O. Mozos, M. Ballesta, O. Reinoso, "A comparative evaluation of interest points detectors and local descriptors for visual SLAM," Machine Vision and Applications Journal, 2010, vol. 21, no. 6, pp. 905-920.

[6] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. The International Journal of Robotics Research. In press. 2012

[7] OpenCV web site: http://opencv.willowgarage.com/wiki/

[8] PCL web site: http://pointclouds.org/

[9] K. Mikolajczyk, C. Schmid, "A Performance Evaluation of Local Descriptors," IEEE Trans. on Pattern Analysis and Machine Intelligence, 2005, vol. 27, no.10, pp. 1615-1630.

[10] K. Mikolaiczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T Kadir, L. Van Gool, "A Comparison of Affine Region Detectors," Int. J. of Computer Vision , 2006, vol. 65, no.1, pp. 43-72.

[11] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, W. Burgard., "An Evaluation of the RGB-D SLAM," In Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), Minnesota, 2012.

[12] C. Harris, M. Stephens, "A Combined Corner and Edge Detector," in Proc. 4th Alvey Vision Conf, Manchester, 1988, pp. 189-192.

[13] J. Shi and C. Tomasi, "Good Features to Track," in Proc IEEE Conf. on Computer Vision on Pattern Recognition (CVPR), Seattle, 1994, pp. 593-600.

[14] J. Matas, O. Chum, M. Urban, T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," Image and Vision Computing, 2004, vol. 22, no. 10, pp. 761-767.

[15] D. Lowe. "Distinctive Image Features from Scale-Invariant keypoinys," Int. J. of Computer Vision, 2004, vol. 60, pp. 91-110

[16] H. Bay, A. Ess, T. Tuytelaars, L.V. Gool, "SURF: Speeded up robust features," Computer Vision and Image Understanding, 2008, vol. 110, no. 3, pp. 346-359

[17] G. Grisetti, C. Stachmiss, W. Burgard, "Non-linear Contraint Network Optimization for Efficient Map Learning," IEEE Trans. on Intelligent Transportation Systems, 2009, vol. 10, no. 3, pp. 428-439.

[18] J. Smisek, "3D Camera Calibration," MSc. Thesis. Czech Technnical Univesity, 2011, Prague (Czech).