Large Scale Egomotion and Error Analysis with Visual Features

M. Cazorla, D. Viejo, A. Hernandez, J. Nieto and E. Nebot

Abstract—Several works deal with 3D data in SLAM problem but many of them are focused on short scale maps. In this paper, we propose a method that can be used for computing the 6DoF trajectory performed by a robot from the stereo images captured during a large scale trajectory. The method transforms robust 2D features extracted from the reference stereo images to the 3D space. These 3D features are then used for obtaining the correct robot movement. Both Sift and Surf methods for feature extraction have been used. Also, a comparison between our method and the results of the ICP algorithm have been performed. We have also made a study about errors in stereo cameras.

Index Terms—Computer Vision, Mobile Robotics.

I. INTRODUCTION

During the last years, new 3D sensor devices have been developed, and computing capabilities have been improved. These improvements can be used to obtain and process a better robot environment information in the field of mobile robotics [1], [2], [3]. By now, methods for achieving tasks such as localization [4], [5], [6], navigation [7], [8] or automatic map building [9], were restricted to the two dimensional world which could be captured by the robot sensors. Nevertheless, using the new 3D sensors such as stereo cameras or 3D laser range finders it is possible to improve the representation of observed objects in order to use them into applications such as augmented reality, architecture, manufacturing process, SLAM problem, etc. Furthermore, this new dimension can be used to improve the methods and behaviors used by a robot in order to accomplish its objectives. In this way, the robots equipped with this new 3D sensors are able to move freely into a 3D space, without being confined to the ground, watching and avoiding 3D shape and volume objects.

In the mobile robot field, one of the most important issues that has to be considered is the movement performed by the robot between two consecutive poses. This information can be obtained from different sources. The most common solution consists in using the robot internal odometers to estimate its movement. Nevertheless, using odometry causes two main problems. First, odometry information always includes measurement errors, which affect the results. Second, it is possible to work with a robot without odometer sensors, or whose odometry information is quite imprecise. This is a

very common situation for robots that are able to perform six degrees of freedom (6DoF) movements into a real 3D environment. For this reason, it is necessary to obtain a movement estimation as accurate as possible. In order to compute this movement estimation, robot surroundings data grabbed by its sensors are used. This kind of solutions for robot movement estimation are known as egomotion or pose registration methods [10], [11], [12].

Using 3D information in order to get the 6DoF transformation from the robot (egomotion) is not an easy task. Although several approaches have been used (ICP [13], [14], Ransac [15], etc.) all those approaches do not work in the presence of outliers (features seen in one frame and not seen in the other). The greater the robot movement the greater the number of outliers are, and the classical methods do not provide good results. In this paper, we propose the use of visual features (like Sift [16], SURF [17]) from the 2D image together with 3D information from stereo processing. We have to deal with the 3D error of the stereo camera: 3D points close to the camera have less error than points far from it.

The paper is organized as follows: First, in section II we present an overview of the physical systems used for obtaining 3D data. Then, Section III presents the method used to get the practical error from the stereo cameras. Later, in section IV we briefly describe the feature extractors used in this study. Our approach for computing egomotion from consecutive stereo images is described in Section V. The results of our experiments are discussed in section VI. In Section VII we briefly analyze some errors present in the results. Finally, we present all the conclusions obtained during this paper and the future work in section VIII. The later is focused on obtaining a more complete 3D model that could be used for estimating robot movements. We also plan to include robot movement estimation into a global error rectification algorithm.

II. DATA ACQUISITION

The data is taken from two different cameras which are mounted on the top of a vehicle, as shown in Fig. 1. One camera is a Bumblebee XB3 from PointGrey with two different baselines, 24cm and 12cm. The other camera comes from Videre and allows different configurations for its baseline. In this case, we have set it to its maximum (60cm) in order to improve the resulting stereo data for further objects. We have acquired a set of images with the vehicle at 30-40km/h. We have driven it in a path through the University of Sydney, with a distance of 3-4kms, in a real scenario with other vehicles and pedestrians around. The frame rate is, approximately, 8Hz

Diego Viejo and Miguel Cazorla are with University of Alicante.
E-mail: {dviejo, miguel}@dccia.ua.es

Andres Hernandez, Juan Nieto and Eduardo Nebot with University of Sydney.
E-mail: aher5442@mail.usyd.edu.au, {j.nieto, nebot}@acfr.usyd.edu.au

and we have processed the sequence offline. There are more than 4000 images in total.

Fig. 1. Vehicle used in the experiments and detail of the cameras placed on it.

III. ERROR MODELLING

We are interested in evaluating the measurement error in depth provided by our stereo cameras. We know the theoretical errors that are provided by the manufacturer, but the real error seems to be higher than those. We have to be sure about the error, so we are going to measure it. To do that, we use the system shown in Figure 2. We have placed the two stereo cameras that we are going to use in our experiments together with a laser Sick. As the error of the laser is negligible compared to the error of the cameras, it provides a good “ground truth” for computing camera’s errors. We place a plane surface in front of the cameras and take a complete reading of the laser and the two cameras at the same time. The measurement is done from 1 to 30 meter, with intervals of less than a meter. As shown in Figure 3, the theoretical error is much less (usually three times less) than the real error. We’ll use this error information in order to improve the accuracy of the method described in the next sections.

Fig. 2. Error modelling system set up. A 2D laser is used together with our stereo cameras. The information obtained by this laser is used as “ground truth” for computing the stereo cameras real measurement error.

IV. VISUAL FEATURES

In this section, we briefly describe the two features used and compared in the paper. One of the most used visual feature is SIFT [16], a method used in computer vision to detect and describe features in an image. It performs a local pixel appearance analysis at different scales. The SIFT features are designed to be invariant to image scale and rotation. Furthermore, it obtains a descriptor for each feature that can be used

Fig. 3. Measurement error analysis. These graphics show the relation between the distance at which an object is observed and the measurement error obtained. The first two graphics come from the Bumblebee XB3 stereo camera for 12 and 24 cm. baseline configuration. The third graphic shows the error information from the Videre camera configured with a 60 cm. baseline. The real error (in blue) is compared to the theoretical error (in black) provided by the manufacturer.

for different tasks such as object recognition. SIFT algorithm is divided into two main parts. In the first one, the location of the points of interest is extracted. The image is convolved using a Gaussian filter at different standard deviations σ . Then, the difference of Gaussians (DoG) is computed as the difference between two consecutive Gaussian-convolved images. This process is repeated in order to obtain the DoG for the input image at different scales. The localization of the points of

interest starts when all DoG have been computed. A point located in a DoG is considered as a point interest if it has the maximum/minimum value compared with its 8-neighbours in the same DoG and with the 9-neighbours in the adjacent DoG at superior and inferior scale. The localization of the points of interest is then improved by interpolating nearby data, discarding low-contrast points and eliminating the edge responses. In the second part of the algorithm a descriptor vector is computed for each point of interest. Based on the image gradient around a point of interest an orientation for this point is computed. This orientation represents the starting point from where the descriptor array is computed. This is a 128-element array that holds the information about 16 histograms of 8 bins computed from the same gradient data. Finally, this descriptor vector is normalized in order to enhance invariance to changes in illumination.

On the other hand, we have analyzed the Speeded Up Robust Features (SURF [17]). This method is partly inspired by the SIFT descriptor. Instead of modifying the scale of input image, SURF algorithm uses a concept known as integral images that allows scaling up the image filter in a constant time. Furthermore, the SURF descriptor is held in a 64-element array instead of a 128 one. In order to compute this descriptor, instead of the gradient the first order Haar Wavelet distribution is used. Having the same rotation and scale invariance response, SURF is faster than SIFT in finding features and its descriptors. Furthermore, since the descriptor size is also lower, SURF-matching methods are also faster than the SIFT-based ones. Despite the speed, both kinds of feature extractors provide similar results for matching applications.

V. EGOMOTION

In order to get the egomotion of the vehicle, we present an ICP-like method to match 3D features. First, we process the base image (the image which is centered in the coordinate system of the camera) to get features (Sift and Surf in this paper, but any 2D feature can be used). Then, we match the 2D coordinates (image coordinates) of the feature with the 3D data from the stereo processing, in order to provide the feature with 3D coordinates. Fortunately, stereo cameras provide information about which 2D coordinate is associated with 3D data. Thus, matching is very simple: we look for the 3D data of a 2D coordinate (the center of a 2D feature). If there is this information, we give the 3D coordinates to the 2D feature). If not found due to, for example, lack of texture, we search around the 2D point in a certain neighborhood and provide a mean value of the 3D data found in this neighborhood.

Due to the errors presented in Section III, we have decided to select features close to the camera, because the longer the distance to the camera, the greater the 3D error. Thus, only features with a Z distance below a threshold are selected to match between two consecutive sets. We have to take into account the movement between two consecutive frames, in order to select a set of features in both frames which intersect and have enough number of features to match. If movements are limited to, for example, 1 meter we select features from 1

to 2 meters in the first frame and from 0 to 1 in the second one. If there are not enough matches, we expand the limits from 1 to 3 meters and from 0 to 2, and so on to find a minimal number of matches or to reach a long distance (10 or 20 meters, depending on the baseline).

Once we have found matches between two consecutive frames, we apply an ICP-like algorithm to find the 3D transformation between frames. ICP is a classical algorithm to match two 3D point sets, but it can not find a good alignment in the presence of outliers. For long movements, ICP does not give good results, because there are a lot of outliers. In our case, where movements are from 0.5 to 4 meters, ICP is not a good approach. But using features like Sift or Surf, with an additional information, i.e. descriptors which are robust to brightness change and point of view change, are good enough for this task. So we use descriptors to find matches, instead of using Euclidean distance like the original ICP.

We have found several problems in our approach. We selected only the closest points in order to avoid the error (Z error) from the furthest points. The distance mean error (X, Y, Z) between two consecutive frames is below 1cm, but the angle error is not acceptable. This angular error comes from the fact that small angular errors in closest points yield bigger angular mean errors. We have used closest points to get the translation term (T_x , T_y , T_z) and all the points for angular term (R_x , R_y , R_z). In Figure 4 we show some results. In this figure we have used the Sift features.

VI. RESULTS

The result of applying the proposed method to our data set, taken at the University of Sydney, is shown in Figure 5. It corresponds to the first part of the sequence. Three different resulting trajectories are shown. In two of them the proposed method was applied using Sift and Surf features and descriptors. Red and blue lines correspond to the trajectories computed in this way. The green line was obtained by applying ICP algorithm directly over each two consecutive 3D stereo sets. The ICP result is used to compare the results of our proposal with a classical 3D egomotion approach. Sift based egomotion seems to provide the best results, compared to Surf's and the ICP's, as it fits almost correctly the real trajectory followed by the robot.

Once we have computed a trajectory using the method proposed in this paper, we can use it in order to represent all the observations within a common reference system. This reconstruction makes up a 3D map of the environment of the robot. For the experiment explained before, Figure 6 shows the resulting 3D map for the trajectory obtained from the University of Sydney with the Sift based approach explained in this paper. It is a free 3D view of the computed map. Robot trajectory is represented with a red line. All the 3D points from the scenes taken by the robot at each pose are stored together using an occupancy grid. This kind of maps can be used for 3D localization or volumetric obstacle avoidance in 3D. The trajectories obtained with the other methods (Surf and ICP) don't provide clear results due to the error committed and therefore are not shown.

Fig. 4. Two examples of the matching method. Left: initial matching (without applying any algorithm) Right: final matching, result of applying our method). The black lines indicate matches. Top: first experiment: only 51 out of 850 possible matches were used. The distance error was 2,5mm and the execution time was 169ms. Bottom: second experiment: only 22 out of 860 matches were used. Distance error of 6,2mm and execution time of 144ms.

Fig. 6. Reconstructed 3D map from computed trajectory (red line). All the 3D scenes observed by the robot during its trajectory are stored in a reference frame.

Fig. 5. Results from the experiment carried out at the University of Sydney. Three different trajectories are placed on top of an aerial picture of the environment. The result of applying our proposed method with Sift features is represented by the red line, Surf based result is represented in blue and ICP egomotion algorithm result is shown in green.

VII. LARGE SCALE ERRORS

The previous method presents a good egomotion estimation in small paths. However, there still is a residual error, which makes the final trajectory unreadable after 100 meters. It comes from the fact that we use a least squares approach to find the transformation between two consecutive frames. However, the least square method is only valid if errors are isotropic (see Figure 7). If not, a different (and not so obvious) approach must be followed. When using a 3D laser in large

paths, errors are very low compared with stereo. So, if the previous method is applied in conjunction with 3D laser data, the results could be acceptable. However, with stereo data [18] this error produce a reconstruction like the one shown in Figure 9. The transformation found at the egomotion step has a rotation error (around X axis) and some minor translation errors. This provides an additive error which produces a hill where an almost plane path must be. In very large paths we have detected a “tornado-like” effect, with loops around the X axis.

Another source of error is present when our vehicle is stopped, like in a traffic light. If another vehicle is in front of us, and as this is closer than other objects, our algorithm uses 3D information from this vehicle, which is moving like our vehicle, with the result of a bad egomotion. A better study of this situation must be done.

VIII. CONCLUSION

In this paper we have presented a method for large scale robot 6DoF movement estimation using stereo images. We have also performed a comparison between Sift and Surf feature extractors. Although Sift is slower than Surf, it provides better results for our method. It is not clear where these bad results come from when using Surf, since previous works show how both methods provide almost the same results. In this way, a deeper study must be done. We have also demonstrated how the trajectory computed with our method improves the results obtained with the classical ICP algorithm, commonly used for computing 6DoF egomotion for two 3D set of points.

On the other hand, we have made a study about errors in this framework. First, real errors from stereo system have been calculated, to be used in the matching process. Second, some

Fig. 7. Pose error due to anisotropic error. Least square method tries to fit features with different errors and the obtained result has an error of rotation (and some translation), which is accumulated.

Fig. 8. Cenital view of the reconstructed trajectory. This result is not so bad, but in large distances (like the end of the path, to the right) a straight path is obtained as a curve.

Fig. 9. Robot View of the reconstructed trajectory. The trajectory should be almost a horizontal line (without elevation) in this part, but due to the accumulative errors (see Figure 7) is like a hill.

situations which must be taken into account in real scenarios have been described.

As future work, we plan to minimize the effect of anisotropic errors by reprojecting 3d locations of the matched features on the image plane. Then we can use the most consistent euclidean distance between features in the image plane instead of 3D euclidean distance. We want also to merge

information from both cameras in order to improve the final result.

Besides, we are planning to model dynamic objects in the environment (like pedestrians, other vehicles and so on), because that can yield (and effectively does) a great source of errors.

ACKNOWLEDGMENT

This work has been supported by grant JC08-00077 from Ministerio de Ciencia e Innovación of the Spanish Government. We would like to thank Rio Tinto Mine Automation Center for their support in this work.

REFERENCES

- [1] J. Weingarten, G. Gruener, and R. Siegwart, "A Fast and Robust 3D Feature Extraction Algorithm for Structured Environment Reconstruction," in *None*, 2003.
- [2] D. Schroter and M. Beetz, "Acquiring models of rectangular 3d objects for robot maps," in *Proc. IEEE International Conference on Robotics and Automation ICRA '04*, vol. 4, pp. 3759–3764, Apr 26–May 1, 2004.
- [3] C.-T. Kuo and S.-C. Cheng, "3d model retrieval using principal plane analysis and dynamic programming," *Pattern Recogn.*, vol. 40, no. 2, pp. 742–755, 2007.
- [4] W. Burgard, D. Fox, and S. Thrun, "Active mobile robot localization," Tech. Rep. IAI-TR-97-3, 25, 1997.
- [5] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *IEEE International Conference on Robotics and Automation (ICRA99)*, May 1999.
- [6] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2001.
- [7] A. Tsatsanis, K. Valavanis, and N. Tsourveloudis, "Mobile robot navigation using sonar and range measurements from uncalibrated cameras," in *Proc. 14th Mediterranean Conference on Control and Automation MED '06*, pp. 1–7, June 2006.
- [8] G. L. Mariottini and D. Prattichizzo, "Image-based visual servoing with central catadioptric cameras," *Int. J. Rob. Res.*, vol. 27, no. 1, pp. 41–56, 2008.
- [9] S. Thrun, D. Hahnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker, "A system for volumetric robotic mapping of abandoned mines," in *Proc. IEEE International Conference on Robotics and Automation ICRA '03*, vol. 3, pp. 4270–4275, 14–19 Sept. 2003.
- [10] M. Agrawal, "A lie algebraic approach for consistent pose registration for general euclidean motion," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1891–1897, Oct. 2006.
- [11] O. Koch and 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, 17–22 June 2007.
- [12] R. Goecke, A. Asthana, N. Pettersson, and L. Petersson, "Visual vehicle egomotion estimation using the fourier-mellin transform," in *Proc. IEEE Intelligent Vehicles Symposium*, pp. 450–455, 13–15 June 2007.
- [13] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *1991 Proceedings., IEEE International Conference on Robotics and Automation, 1991.* (G. Medioni, ed.), pp. 2724–2729 vol.3, 1991.
- [14] P. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [15] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.
- [17] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008. Similarity Matching in Computer Vision and Multimedia.
- [18] G. Dubbelman and F. Groen, "Bias reduction for stereo based motion estimation with applications to large scale visual odometry," in *Proc. of Computer Vision and Pattern Recognition*, pp. 2222–2229, 2009.