Multi-modal sensors path merging

Léo Baudouin[1], Youcef Mezouar[1], Omar Ait-Aider[1] and Helder Araújo[2]

[1] Institut Pascal, Université Blaise Pascal,
24 avenue des landais, 63170 Aubière, France
[2] Institute for Systems and Robotics, University of Coimbra,
3030-290 Coimbra, Portugal

Abstract. In this paper, we propose an original method to merge maps from different robots. Each map was built using a camera which can be different: perspective, fish-eye or omnidirectional. Each robot creates its own local map, the main goal is to build a global map assuming that the paths overlap each other on at least one segment of the path. The first step is to find this common part by using a correlation method. Then the rigid transformation between trajectories is computed and used to merge paths.

Keywords: Map merging, autonomous robots, multi-robots, vision, loop-closure

1 Introduction

The navigation of a multi-robot system is an important field of research due to the wide range of applications. Among the variety of robots, autonomous cars have a crucial place in a large number of scenarios where the vehicles have to navigate together. Therefore new algorithms have to be developed to manage these types of robot fleets.

A new challenge is introduced in the case of visual servoing robots in which visual odometry is built from image sequences, because the resulting trajectory and cloud point have an unknown scale factor and frame of reference. For this reason, matching maps from different sensors is a very difficult task. This paper describes a scale insensitive method to perform path matching as an approach to enable the combination of maps generated by heterogeneous sensors.

Path matching is often performed on topological maps (graphs) [9] which needs a classification (turn, straight line, ...). However, and in many cases, it is very difficult to extract correctly this kind of information given the unknown scale. At a different scale, a simple turn will be split into several lines and/or small turns. In this paper we present an approach which can deal with the issue of unknown scale factor.

1.1 Related work

Many multi-robot systems are made up of robots whose perception systems are similar. In particular all the robots use either the same type of camera or the same set of sensors (e.g. cameras and laser range finders) [7, 1]. As a result the data are similar and, in the case of the use of cameras and/or laser range finders (LRF) the detection of similar parts of the trajectories can be performed using the images and algorithms for image matching such as FABMAP [5]. Mixed/heterogeneous camera systems have been studied in [13, 3], but were tested and applied in indoor environments and image matching was performed using the SIFT [11] detector. In outdoor sequences and with different types of cameras, the points-of-view and illumination change significantly. As a result, and in general, algorithms which use SIFT or SURF are not robust enough to detect a sufficient number of inliers in images.

1.2 Contributions

The goal of our work is the development of an approach that can facilitate the determination of matches between images. This algorithm can be considered as a "bootstrap" method to image matching. The paths of the robots are pre-matched in order to guide the classical image matching based algorithms. To that end we applied a pattern recognition method to the problem of map matching and merging. The approach is based on the simple principle that the sensors must move in common areas to acquire similar data at different moments. The algorithm we developed is sensor independent so that the approach can deal with different types of cameras on a multi-robot scenario.

Our method also improves the *part-to-part* matching algorithm introduced in [4] by reducing the number of *whole-to-part* iterations. We propose a method to split the trajectory so that only relevant parts of the trajectory (in terms of information amount) are tested, e.g, succession of vehicle turns and straight lines.

2 Path Matching

The path matching process that we elaborate is built from a pattern recognition algorithm originally developed to match 2D curves [4].

Our map merging approach assumes that each robot has already estimated its trajectory and performed 3D reconstruction and, therefore, each robot has an estimate of the map [14]. We further assume that the trajectory and the 3D map were estimated up to a scale factor using structure from motion techniques [6]. Therefore each map has a different and unknown scale. The main idea is to use scale invariant information to find out the common parts between two trajectories estimated by visual odometry, GPS or other sensors. These initial estimates are then used to "bootstrap" the image and map matching (with data from vision).

The algorithm determines several potential trajectory section matches, therefore significantly facilitating the matching of the visual information by restricting the search to a reduced set of images acquired from locations situated on matched trajectory sections.

The solution proposed in this paper performs a matching between the paths using curvature as the criterion. To remove the scale factor, the curvature is resampled with respect to the integral of absolute curvature.

2.1 Trajectory approximation

For the matching algorithm, we need to compute the curvature at each point of the robot trajectory. The input data is the list of the 3D coordinates of the camera center obtained by visual odometry. Since the origin of the coordinate system is unknown the 3D points are not necessarily contained in the plan $\vec{x}O\vec{z}$ as usual. Therefore it is assumed that the points can be projected on a plan (i.e. the point elevation is insignificant compared to the path size). For that purpose a plan can be estimated by minimizing the distances to all 3D points. The 2D points corresponding to the 3D points are then used to define a planar map. Visual odometry generates a point cloud [12]. This is a discrete data set. To facilitate data processing spline or b-spline curves are fitted onto the visual odometry data set. This approximation allows for an easy computation of the positions, orientations, derivatives and curvatures along the robot trajectory. In addition the trajectory approximation using the splines also filters out part of sensor generated noise, while simultaneously suppressing noisy high curvature regions due to artefacts.

2.2 Whole-to-part matching

The whole-to-part pattern matching has been studied and applied to shape and letter recognition by Cui [4] which uses the curvature of the path to create a scale-invariant signature. This section summarizes this previous work.

We note $\mathbf{x}(s) : x(s), y(s)$ the point of trajectory at arc length s, and x' and x'' respectively the first and second derivatives of x with respect to s.

The curvature $\kappa(s)$ at arc length s is computed as follows:

$$\kappa(s) = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{3/2}} \tag{1}$$

The main idea is to re-sample the curvature w.r.t the integral of absolute curvature ($K(s)$).

$$K(s) = \int_{s_1}^{s_2} |\kappa(s)| \, \mathrm{d}s \tag{2}$$

Since curvature is inversely proportional to scale, the integral of absolute curvature is the same in two different scale maps, with only the arc length being different. As $K(s)$ is a monotone function, it is easy to find the correspondence between s and $K(s)$. Our algorithm uses one spline to perform this step.

Matching is performed by a zero-mean normalized cross correlation (ZNCC) [10], which is a scale insensitive measure of covariance. Taking two curves, a short and a long one, the goal is to find where the first curve is inside the second one.

As shown in figure 1, the best match between the curvature of the whole short trajectory (green) and a part of the long one (red) is found out (blue). Note that the result can be inverted (along x-axis) in order to deal with travel direction. However, and in our case, we need part-to-part matching since only several parts of both trajectories are similar.

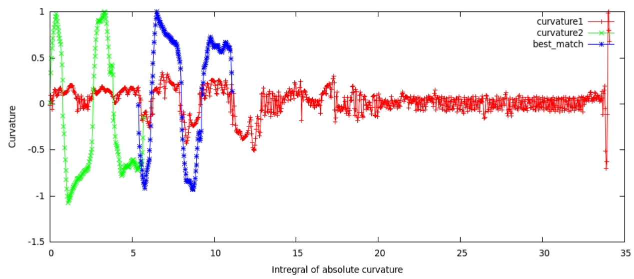

Fig. 1: Matching of resampled curvature with respect to integral of absolute curvature. A short trajectory curvature (green) is search in a long one (red). The best result is shown in blue.

2.3 Part-to-part matching

While the basic algorithm performs a *whole-to-part* matching, i.e. finds out which part of a long path is matches the small path, a *part-to-part* matching between two long paths is more complex. The reason is that only sub-parts of the second need to be chosen to find correspondences. The question is how to split the path to perform the matching (position and length of extracted part).

Our idea is to split the path in several sub-parts (called segments) where each of them has substantial information. For that purpose, a segment needs to have an important change of curvature. The first step is the fitting of G2-clothoids to the trajectory (clothoids are curves whose curvature varies linearly with arc length). Then, by using at least two clothoids, it will ensure that there is a change in the orientation of the robot, i.e. a peak of change in the graph of curvature.

Using the Cornucopia library [2], a sequence of clothoids can be fitted to a list of points. However scale is still a problem. If the scale factor is too small, Cornucopia will determine a very small number of clothoids (approximation too far away from reality), and if the scale is too big, the number of clothoids will be very high. Since these algorithms are not scale invariant, we first need to rescale the cloud of points to an exploitable scale.

Empirically we found a scale value for which the fitting yields good results. Using the camera frame rate or/and distance between points, we rescale the input path so that this new scale is obtained even if it differs significantly from its real value. A result can be seen in figure 2.

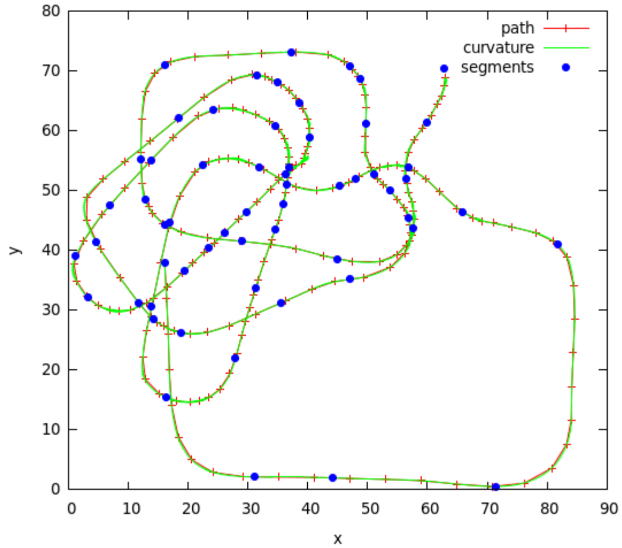

Fig. 2: Partitioning one path affected by severe drifting. Each blue dot is one end of segment (clothoid).

3 Algorithms and Implementation

The algorithm can be divided into two steps, namely approximation and matching (see algorithm 1). Segmentation can be executed on all trajectories, offline to accelerate the process, but only the shortest trajectory segmentation will be used during the matching process.

We take N consecutive segments in the list of clothoids. This allows the determination of the start and end arc lengths which is then used in the part-to-part method. To have a good amount of information, N is varied between two limits $N1$ and $N2$. The minimal $N1$ is, as shown before, 2 segments. The value for $N2$ depends on the maximum size of the common trajectory. In our specific implementation we empirically fixed this value to 12, so that the path included a large number of turns. ZNCC score and length of the segments are then used to compute the final score to be compared during the matching.

4 Experiments

4.1 Trajectory matching

Experiments were performed with image sequences from IPDS [8]. The images were taken by a perspective camera, a fish-eye camera (on the vehicle) and also by an omnidirectional camera attached to a pole fixed on the top of our VipaLab mobile robot.

Algorithm 1 Path matching

c1 ← longest curve
c2 ← shortest curve
(B)Spline approximation (c1 & c2)
Segmentation with clothoids (c2)
for N = N1 to N2 **do**
  p ← take 'N' consecutive segments of c2
  Perform whole-to-part matching (between c1 & p)
  Keep 'k' best matches
end for

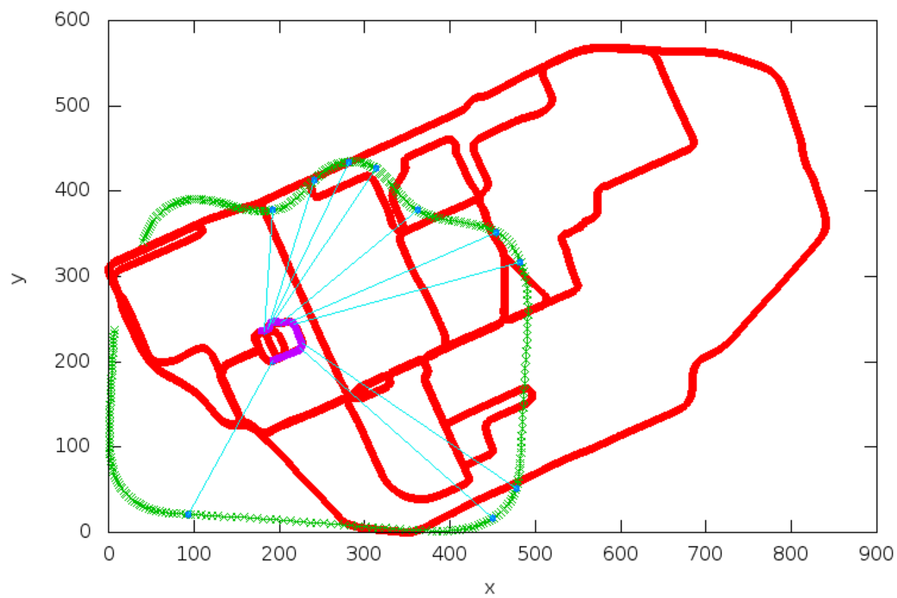

Fig. 3: Result of *part-to-part* matching between a trajectory estimated by visual odometry (green) and a GPS ground-truth (red), the purple segment is the common part found with the rigid transformation applied.

Figure 3 presents the results of our approach. Two sets of data were used: one obtained by GPS and the other obtained using an image sequence acquired by a perspective camera. Both sets of data correspond to similar trajectories, whose paths were described by a vehicle in our campus.

The image-based data set is affected by a small drift along the trajectory and the scale is unknown.

4.2 Path merging

Using path matching, a number of rigid transformations are computed and scored, only best results are kept. Knowing the rigid transformation between

Fig. 4: Path merging from two paths with different scale and frame of reference. Purple and blue parts are common segments found. Orange lines shows the rigid transformation. The result of the merging is shown on the second figure.

segments in two different paths, maps can be combined to create a global one. The example on figure 4 shows the result of a path merging. Input paths are trajectories from the campus (red part), and from Pavin, our experimental platform (green). Only a part of the maps are common, a single loop around Pavin. The rigid transformation is then applied on the green trajectory in order to place correctly the path over the campus.

4.3 Loop closure

One of the results of our work is the possibility of loop closure detection when we apply trajectory matching of a path with itself. Self-matching yields important information on locations already visited by the robot. This can then be used in bundle adjustment [6] to improve the robustness of SLAM 3D reconstruction.

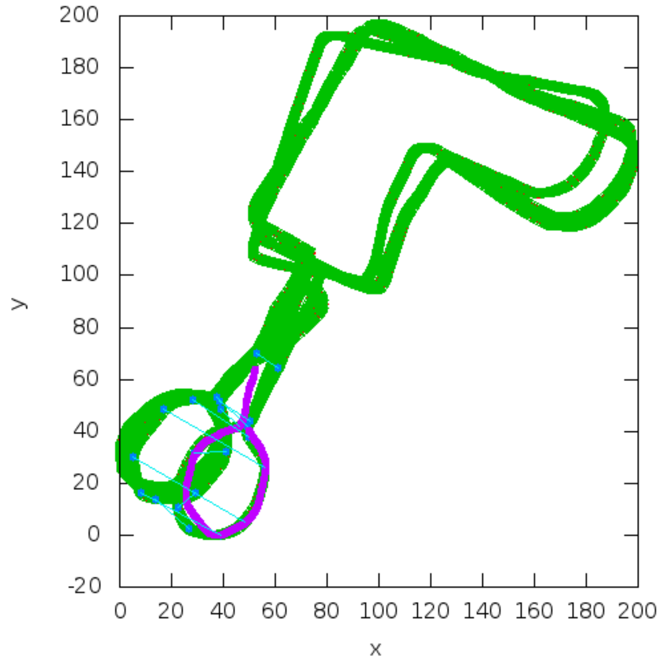

Fig. 5: One loop closure on the New-college dataset

In this case the algorithm needs to prevent the trivial self-matching case (same part) by removing the segment used for searching from the original trajectory. By keeping the k best matches, a list of potential loop-closures can be created. Since it is not possible to determine whether or not this set of potential matches are correct, an additional sensor, namely vision, has to be used to confirm or reject the hypotheses.

The goal of this loop-closure pre-match is to reduce the number of tests in traditional image to image matching.

Figure 5 presents a possible loop-closure on the NewCollege dataset. Our algorithm was applied to the path provided, which was obtained by visual odometry. For each part-to-part matching several potential hypotheses were generated and sorted according to the matching score.

The O-shaped segment can also be found in several other loops as well as in all L-shaped corners.

4.4 Side results

As shown before, we can match GPS maps and vision based maps. Using a trajectory built with a robot equipped with a GPS or a map scaled by an operator, we can determine the exact scale of the 3D reconstruction.

5 Conclusions and future work

In this paper we propose a new approach to part-to-part matching of robot trajectories by modifying already existing pattern recognition methods. This information can be helpful for merging robot-generated maps if the trajectories share common parts. In addition our method can be used to generate hypotheses for loop-closure determination. The hypotheses can be sorted based on a matching score. Since true and false matches can not be determined based only on trajectory shape, image matching or visual appearance matching is necessary. Matching of visual appearances, however, needs only to be applied to images taken from matched trajectory segments (pre-determined possible matches and/or loop-closures).

This algorithm can be improved by applying filters to remove extremely high curvature points (GPS signal lost, GPS signal reflected on building, visual odometry error, . . .) which generates incoherent results in matching process.

Acknowledgment

This work was supported by a help from the government managed by the *'Agence Nationale de la Recherche'* for the program *'Investissements d'Avenir'* in the project LabEx IMobS[3] (ANR7107LABX716701), a help from the *'Union Européenne'* to the *'Programme Compétitivité Régionale et Emploi'* (200772013 - FEDER - *Région Auvergne*) and a help from the *'Région Auvergne'*.

References

1. R. Aragues. *Distributed Algorithms on Robotic Nectwoks for Coordination in Perception Tasks.* PhD thesis, Universidad de Zaragoza, 2011.
2. I. Baran, J. Lehtinen, and J. Popovic. Sketching Clothoid Splines Using Shortest Paths. *Comput. Graph. Forum*, 29(2):655–664, 2010.
3. Y. Bastanlar. *Structure-from-Motion for Systems with Perspective and Omnidirectional Cameras.* PhD thesis, Middle East Technical University, 2009.
4. M. Cui, J. Femiani, J. Hu, P. Wonka, and A. Razdan. Curve matching for open 2D curves. *Pattern Recognition Letters*, 30(1):1–10, 2009.
5. M. Cummins and P. Newman. Appearance-only SLAM at Large Scale with FAB-MAP 2.0. *The International Journal of Robotics Research*, November 2010.
6. R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press, 2003.
7. K. Konolige, D. Fox, B. Limketkai, J. Ko, and B. Stewart. Map merging for distributed robot navigation. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 212–217. Ieee, 2003.
8. H. Korrapati, J. Courbon, S. Alizon, and F. Marmoiton. "The Institut Pascal Data Sets": un jeu de données en extérieur, multicapteurs et datées avec réalité terrain, données d'étalonnage et outils logiciels. *ORASIS*, June 2013.

9. H. Korrapati, Y. Mezouar, and P. Martinet. Efficient Topological Mapping with Image Sequence Partitioning. In *European Conference on Mobile Robots, ECMR11, Örebro, Sweden*, September 2011.
10. J. P. Lewis. Fast Normalized Cross-Correlation. volume 1995, pages 120–123. Citeseer, 1995.
11. David G. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.
12. David Nistér, Oleg Naroditsky, and James R. Bergen. Visual Odometry. In *Computer Vision and Pattern Recognition*, pages 652–659, 2004.
13. L. Puig, J.J. Guerrero, and P. Sturm. Matching of omnidirectional and perspective images using the hybrid fundamental matrix. In *Proceedings of the Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras, Marseille, France*, 2008.
14. Eric Royer, Maxime Lhuillier, Michel Dhome, and Jean-Marc Lavest. Monocular Vision for Mobile Robot Localization and Autonomous Navigation. *International Journal of Computer Vision*, 74(3):237–260, 2007.