Qualitative localization using vision and odometry for path following in topo-metric maps
Emmanuel Battesti, Stéphane Bazeille, David Filliat

To cite this version:
Emmanuel Battesti, Stéphane Bazeille, David Filliat. Qualitative localization using vision and odometry for path following in topo-metric maps. European Conference on Mobile Robotics (ECMR), 2011, Sweden. pp.303-308. hal-00652479

HAL Id: hal-00652479
https://hal.archives-ouvertes.fr/hal-00652479
Submitted on 15 Dec 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Abstract—We address the problem of navigation in topo-
metric maps created by using odometry data and visual loop-
closure detection. Based on our previous work [6], we present
an optimized version of our loop-closure detection algorithm
that makes it possible to create consistent topo-metric maps in
real-time while the robot is teleoperated. Using such a map, the
proposed navigation algorithm performs qualitative localization
using the same loop-closure detection framework and the
odometry data. This qualitative position is used to support
robot guidance to follow a predicted path in the topo-metric
map compensating the odometry drift. Compared to purely
visual servoing approaches for similar tasks, our path-following
algorithm is real-time, light (not more than two images per
seconds are processed), and robust as odometry is still available
to navigate even if vision information is absent for a short time.
The approach has been validated experimentally with a Pioneer
P3DX robot in indoor environments with embedded and remote
computations.

Keywords: Path following, vision, robot odometry, topo-
metric map, visual servoing.

I. INTRODUCTION

To navigate autonomously in a large environment, a robot
often requires the ability to build a map and to localize
itself using a process named Simultaneous Localization and
Mapping (SLAM). The field of SLAM can be broadly
divided into two approaches: topological and metrical. The
most common approach is the metrical SLAM in which we
traditionally use range sensors such as laser or sonars. This
mapping method is explicitly based on measured distances
and positions. The localization is geometric and clearly
corresponds to the real world. It can be done continuously
and planned navigation is accurate. The main problem is that
global geometric consistency is hard to ensure and the map
is therefore hard to build. Moreover, the sensors are usually
expensive and the computation heavy and greedy.

Now, more and more often those sensors are replaced by
cameras because they provide many advantages such as lower
price, smaller size, lighter weight, lower energy consumption
and give a richer environmental information. Using these
sensors, it is possible to recover metric information, but a
more direct way to map the environment is to use topological
approaches where the environment is modeled as a graph of
discrete locations. These maps are easy to build, suitable
for large environments and for human interactions. Their
main drawback is the lack of geometric and free space
information that only allows localization and navigation close
to previously mapped routes. In our previous work [6], we
built topological maps using visual loop-closure detection
and we used odometry data to enrich this topological map
with metric information. The choice of this second sensor
makes the mapping more accurate, reduces the computational
cost compared to purely visual solutions and also makes the
system more robust to vision failure.

Contribution: The main contribution of this paper is the
development of a new robust and light path following
algorithm combining the use of these two cheap sensors
(odometer and camera) that allows autonomous navigation
in such previously learned topo-metric maps. The approach
is qualitative and uses the feedback information given by the
vision sensor to approximately correct the odometry drift in
order to follow a path computed from the map. This path
following system can be used for delivery robots, security
robots, guide and following robots for example.

Content: In Section 2, we present a review of related
work on topological navigation. In Section 3, we recall our
previous work on topo-metric mapping and present new opti-
mizations that have been brought to this system. In Section 4,
we explain our new framework on path following navigation
using these topo-metrical maps. Finally, in Section 5, we
show experimental results and we conclude in Section 6 with
a discussion about this contribution and our future work.

II. RELATED WORK

Localization is a key issue for mobile robots in en-
vvironments where a globally accurate positioning system,
such as GPS, is not available. Today, the most used sensor
to map an environment and to navigate autonomously in
a map is definitively the laser sensor, combined with a
SLAM framework it builds up a map within an unknown
environment while at the same time keeping track of the
current location (see [26] for an overview of the metrical
SLAM approaches). The position is accurate, and the map
displayed as a geometrical occupancy grid allows the robot
to explore its surrounding. In our work, we have addressed
the problem of autonomous navigation method, but focusing
on the visual sensor. As we make use of a topological
map [5], [7], [25], we have no information about obstacles,
and about free space around the robot, that is why our
navigation method has been limited to follow path that have
been already taken by the robot. The traditional method for
this kind of application is visual servoing also known as
vision-based robot control which uses feedback information
extracted from images to control the motion of the robot
[20]. Those methods generally require camera calibration
(homography, fundamental matrix, Jacobian, removal of lens
distortion [4], [9], [21], [24], [8]). Also, some approaches
make assumptions on the environment (artificial landmarks,
vertical straight lines, parallel walls) or sometimes need more
than one camera or camera of different kind (omnidirectional
for example) [4], [19], [14], [12].
In our research context, we have been interested by the
use of a perspective camera without calibration (indeed, our
method also works with omnidirectional camera [6]), and
above all without any assumption on the environment. Such
calibration free methods had been developed by [10], [11].
They are based on image features tracking, and use qual-
itative comparisons of images to control the motion. Such
methods are very interesting but they require real-time image
processing at high frame rate and are highly dependent of the
quality of image data. Tracking errors or temporary absence
of information lead quickly to system failure. Moreover,
they need lighting constancy so additional processing are
generally added to ensure the desired behavior.
To make our system more robust and accurate, and above all
lighter from a computational point of view, we enable the use
of one more cheap sensor: the odometer. Visual sensor pro-
vides a rich information and an accurate positioning system
and the combined use of odometry makes the algorithm more
robust and relieve the visual system from high frame rate
computation. Odometry allows localization for a short time in
absence of visual information, vision failure (dark or dazzle
areas, blurry image, occlusions), or important changes in the
scene that has been learned (light, people). When embedded
on small platforms, this makes it possible to remotely process
images by guiding the robot in case of network lag. As we
are not too much dependent of visual information, it is also
possible to use visual localization information only when it
is very reliable, avoiding to give position information that
would be unsure. We therefore developed a robust visual
localization system that completely banned false alarms, to
the price of giving less localization information.

III. IMPROVED TOPO-METRIC MAPPING

For the next, we will call loop-closure the event where the
robot detect a matching between the current and the reference
frame. It differs from the traditional loop-closure definition
in which we associate the loop-closure to the event where
the robot revisit an area it has not been to before a while.

A. Summary of our previous work

In [6], we have developed a fully incremental topo-metric
mapping framework. This algorithm builds in real-time topo-
metric maps of an unknown environment, with a monocular
or omnidirectional camera and the odometry gathered by
motors encoders (see Fig. 1). The system is based on an
appearance loop-closure detection method that has been
designed as a two-level decision system to ensure robust and
accurate detection. A first step detects potential loop-closure
locations when the robot comes back to a previously visited
area using appearance only. A second one verifies and selects
the best potential location using image geometry.
A Bayesian filter based on incremental bags of visual words
[16] is used to extract potential loop-closure locations that is
to say find the previous positions that are potentially close
to the current one. In the second step these locations are
verified with a 2D motion computation in the image space
(translation and rotation in image plane) based on the SIFT
[22] keypoints and we select the loop-closure which shows
the smallest translation. In order to discard outliers, the 2D
motion is computed using RANSAC, accepting the result
only if the number of matching points is above a threshold.

Fig. 1. Comparison of topo-metrical mapping and laser mapping. 1.
Raw odometry 2. Corrected odometry applying graph relaxation taking
into account the visual loop-closure (two loop-closure locations detected)
3. Ground truth trajectory (SLAM Laser). The three trajectories are shown
in the frame of the reference laser map.
With the inclusion of an odometry-based evolution model in the Bayesian filter which improves accuracy, robustness and responsiveness, and the addition of a consistent metric position estimation applying an efficient optimization algorithm at each validated loop-closure [18], our system produces a map that corresponds to the real world and only presented limited local drift. It makes it usable for global localization and planned navigation.

For the current work, the robot is teleoperated during an initial mapping phase and our algorithm is used to build a map usable later for navigation. The environment is divided into locations (defined by one or more images) that are linked by relative odometry vector. The mapping phase does not need any preprocessing, calibration, neither postprocessing or parameters adjustment and it builds incrementally its map, adding new location if no loop-closure has been found or updating a location and correcting the graph if a loop-closure has been found (see Fig. 1).

### B. Optimization of the loop-closure detection algorithm

Since we will use this framework for real-time path-following navigation, we brought some optimizations to the approach, notably to improve the performances of the visual localization module:

- We have improved the performances of the algorithm by replacing SIFT [22] keypoints by STAR [3] keypoints. It has greatly divided the keypoints extraction time (more than 20 times), but it has decreased the number of keypoints and their quality. We have compensated this quality loss by using a new validation strategy less restrictive on the number of extracted keypoints.
- We improved the accuracy of the prediction step of the Bayesian filter which is used to extract potential loop-closure locations. Our first version was only using the probability at the previous time-step to predict the new one. In the new version, the Bayesian filter takes into account several previous time steps and the evolution model is applied to the odometry displacements corresponding to these time-steps. The predictions are lastly merged using the max operator to give the final prediction. This step reduces the influence of the map discretization on the quality of the prediction and makes more accurate the extracted potential locations.
- We simplified the validation stage by modifying the geometric model of image transform and by thresholding using all the parameters extracted from 2D motion computation in the image space (translation, rotation and scale). The computation of an homography using four couples of matching points through RANSAC[17] has been replaced by a simpler computation of a 2D motion using two couples of points through RANSAC. Homography was already a simplified version of the real transform but as we work on images with very close viewpoint when closing loops, it could be again simplified to speed up the computation.
- A new navigation mode (described in the next section) has been created to perform path following navigation using the loop-closure detection framework. Compared to the mapping mode, the incremental part of the system that adds new words in the dictionary, new locations in the graph and that relays the topo-metrical map is disabled. It therefore enables qualitative visual localization in the topo-metric map.

It is important to note that, during mapping, loop-closure are only accepted and integrated in the map if the robot comes back very close to a previously visited location. A loop-closure is therefore accepted only if the two images show enough matching points, and if the computed rotation, translation and scale between them are below some threshold [6]. We will see below that this definition has been relaxed for the navigation mode by disabling the translation threshold.

Table I presents some computation time and loop-closure detection system (LCDS) before and after optimization.

### Table I

<table>
<thead>
<tr>
<th>Images</th>
<th>Museum</th>
<th>Gosia</th>
<th>Lab (Fig. 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance (m)</td>
<td>112</td>
<td>38</td>
<td>82</td>
</tr>
<tr>
<td>Distance (m)</td>
<td>82</td>
<td>98</td>
<td>350</td>
</tr>
<tr>
<td>LCD Truth</td>
<td>14</td>
<td>25</td>
<td>9</td>
</tr>
<tr>
<td>Old LCDS [5]</td>
<td>15</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td>Missed LC</td>
<td>7 %</td>
<td>12 %</td>
<td>20 %</td>
</tr>
<tr>
<td>False LC</td>
<td>0 %</td>
<td>5 %</td>
<td>0 %</td>
</tr>
<tr>
<td>CPU Time</td>
<td>42s</td>
<td>70s</td>
<td>210s</td>
</tr>
<tr>
<td>CPU Time/image</td>
<td>0.37s</td>
<td>0.41s</td>
<td>0.5s</td>
</tr>
<tr>
<td>New LCDS</td>
<td>13</td>
<td>26</td>
<td>8</td>
</tr>
<tr>
<td>Missed LC</td>
<td>7 %</td>
<td>0 %</td>
<td>22 %</td>
</tr>
<tr>
<td>False LC</td>
<td>0 %</td>
<td>3.84 %</td>
<td>11 %</td>
</tr>
<tr>
<td>CPU Time</td>
<td>2.16s</td>
<td>2.57s</td>
<td>5.56s</td>
</tr>
<tr>
<td>CPU Time/image</td>
<td>0.019s</td>
<td>0.015s</td>
<td>0.016s</td>
</tr>
</tbody>
</table>

### Compared results of our visual loop-closure detection system (LCDS) before and after optimization.
C. Software overview

The system has been developed using Urbi[1] an open-source software platform to control robots. It includes a C++ component library called Uobject to describe motors, sensors and algorithms. We also use urbiscript to glue the components together using embedded parallel and event-driven semantics. Figure 2 presents the whole description of the architecture of our mapping, localization and path following system. It is composed of five different components including a viewer to supervise robot behavior. Two components have been tested remotely in particular our visual SLAM algorithm.

IV. QUALITATIVE NAVIGATION SYSTEM

The newly developed navigation mode is based on a qualitative position estimate that combines odometry with the visual information provided by loop-closure detection.

A. The navigation mode

The navigation mode of the algorithm presented in this paper requires a topo-metrical map, and the knowledge of the robot starting position in the map. A path to reach a goal from the starting position is computed as a list of nodes using Dijkstra algorithm [13], taking into account the orientation of the robot in each node. In order to follow the computed path, the robot position is continuously computed using odometry and visual loop-closure detections. In this mode, loop-closure detection is less restrictive than in the mapping mode, as loop-closure are accepted whatever the translation between images is. This translation is used to estimate an approximate position which is used to guide the robot. This use of a less restrictive loop-closure validation makes it possible to benefit from much more position correction than in the mapping mode, even if these corrections are less accurate. This limited precision is however not a problem as only localization is performed and the map quality is therefore not impacted. This navigation mode requires that the trajectory is obstacle free because obstacle avoidance is not currently included in our model.

B. Qualitative localization using vision and odometry

The visual loop-closure detection framework verifies at each recorded image if the robot is in an already visited location or not. When a loop-closure is detected, the simple matching between images does not permit to estimate precisely the robot position relatively to the image in the map as the scale factor is unknown when computing the camera displacement. Moreover, for small displacement and particular environment configuration, there is an ambiguity because a lateral translation in images can be caused either by a robot translation or by a rotation. For these reasons, we prefer to estimate a qualitative position, by assuming that the image movement is caused only by a rotation of the robot.

Therefore, when a loop-closure is detected the parameters extracted during the validation of potentials loop-closure locations are used to estimate a qualitative direction. Among the three parameters (translation, rotation and scale), we only use the x-axis translation in pixels between the two matching images to compute the angle between the current robot direction and the direction recorded in the map. An rough camera calibration (only based on image size and camera vision field) make it possible to convert this translation in pixels into an angle (Figure 3). The position of the robot is therefore computed as the position of the loop-closure node but taking into account the deviation in direction.

If no loop-closure is detected between places corresponding to an image acquisitions, the position is computed as the previous loop-closure location position to which the relative odometry recorded since this point in time is added. This makes it possible to produce a continuous position estimate which is corrected when a loop-closure is detected.

C. Servoing system for path following

To control robot motion we have used the strategy proposed by [15]. In order to reach a goal in the topo-metric map, we first compute a sequence of nodes using Dijkstra algorithm. The path linking this sequence of nodes is then discretized each centimeter to form the global path that

Fig. 2. Diagram of the developed system. Each box represents an uobject.

Fig. 3. Illustration of the qualitative visual localization. A loop-closure is detected between image 9 (right) and image 45 (left) with 33 pixels of x-axis translation. The computed corresponding angle of robot rotation is 4.73 degrees.
should be followed to reach the goal. This global path is only computed one time. When an image is acquired, the position is updated using visual information, and a local path to join or to follow the desired trajectory is computed between the position and the global path. The local path is a line between the position and a point of the global path situated at 40 cm in front of the robot to which we add the global path after this point.

Given the local path, each time the robot moves, the position and the global path. The local path is a line between the position and a point of the global path situated at 40 cm in front of the robot to which we add the global path after this point.

Given the local path, each time the robot moves, the position and a point of the global path situated at more than 20 cm of the robot is selected as a target. A heading direction error between this point and the robot position is computed and used to estimate the rotation speed by using a PID controller. As the robot translation speed is set to a constant, the servoing system adjusts the velocity of each wheel to correct the heading error and to follow at best the local path (see Fig. 5).

While this guidance strategy is quite standard, it should be noted that the interplay between this strategy and the qualitative localization method has the effect of guiding the robot to actively close loops during movement. Indeed, without the qualitative localization, the robot would be guided by the odometry only and the drift induced would lead the robot far from the map nodes, thus preventing from visual loop closure detection. With this strategy, each time the robot deviate from the predicted path, the qualitative position correction lead to a local path that guides the robot back on the global path, thus enforcing future loop-closure and position correction.

V. RESULTS AND DISCUSSION

To validate our method, experimentation have been done in an indoor environment using a Pioneer P3DX mobile robot mounted with a Canon VC-C50i camera with a wide angle lens. During the showed experiment all the code was embedded on the robot except the viewing system. The image processing rate was 1 image each 50 cm or 10 degrees. To give an accurate idea of what the system is able to do, we have launched in parallel with our mapping and path following system the laser SLAM positioning system Karto [2]. It gives a reference trajectory in a laser map during the learning and the path following phases. Figure 4 (left) shows an experiment where the trajectory used for mapping (in green) has been replayed using our system (trajectory in pink). The odometry recorded during the path following run (in blue) shows the drift that has been compensated by the visual localization system and that would have led inescapably to wall collision without these compensations. Figure 4 (right) illustrates the effect of our qualitative localization approach during the same experiment. The pink circles correspond to the locations where loop-closures have been detected during path following. The pink line in the circle is the translation computed between the loop-closing images that is used for the qualitative position estimation. We can see that our guidance framework lead to a high loop-closure detection rate (around 60% here) and that the path following behavior is very smooth with sometimes 5 images without direction correction.

Figure 6 shows another experiment illustrating the purpose of Dijkstra algorithm. The replayed trajectory (in pink) to go from the first node to the last node of the map is avoiding the large loop executed during map construction as a shortcut is notescapably to wall collision without these compensations. The replayed trajectory (in pink) shows an experiment where the trajectory used for mapping (in green) has been replayed using our system (trajectory in pink). The odometry recorded during the path following run (in blue) shows the drift that has been compensated by the visual localization system and that would have led inescapably to wall collision without these compensations.
add to our system sonar data to obtain a map giving an
robot is spinning around. This will be used to retrieve the
by applying the loop-closure detection framework while the
Future work will deal with localization after kidnapping
lighten again the on-board computer charge.
point of view, it can also be used as a remote process to
occlusion or lag of the vision system response. From this
the system is robust to lighting change, furniture moved,
As odometry is used to complement the visual information,
or directional) and is suitable for toy robots as it just needs
it can use any kind of camera (omnidirectional, wide angle
and really few computer resources to achieve the navigation
task. It is real-time and does not need any precise camera cal-
ibration or parameters adjustment. With minor adaptations,
it can use any kind of camera (omnidirectional, wide angle or
directional) and is suitable for toy robots as it just needs
cheap sensors and small computation performances.
As odometry is used to complement the visual information,
the system is robust to lighting change, furniture moved,
people crossing, blurry image or even from temporary sensor
occlusion or lag of the vision system response. From this
point of view, it can also be used as a remote process to
lighten again the on-board computer charge.
Future work will deal with localization after kidnapping
by applying the loop-closure detection framework while the
robot is spinning around. This will be used to retrieve the
direction to follow a path after a kidnapping. We will also
add to our system sonar data to obtain a map giving an

VI. CONCLUSION AND FUTURE WORK

In this paper, we have addressed the problem of navigation
in topo-metric maps by using visual loop-closure detection.
The presented algorithm uses the vision system and the
odometric data for qualitative localization in the topo-metric
map in order to guide the robot to follow a path already taken
during a learning phase. The qualitative visual localization
is computed and sent to a servoing system that compensates
the odometry drift to ensure we are always on the learned
trajectory. This system can be seen as an active loop-
closure detection framework as we are forcing by controlled
guidance to close loops. The system only needs two sensors
and really few computer resources to achieve the navigation
task. It is real-time and does not need any precise camera cal-
ibration or parameters adjustment. With minor adaptations,
it can use any kind of camera (omnidirectional, wide angle or
directional) and is suitable for toy robots as it just needs
cheap sensors and small computation performances.

As odometry is used to complement the visual information,
the system is robust to lighting change, furniture moved,
people crossing, blurry image or even from temporary sensor
occlusion or lag of the vision system response. From this
point of view, it can also be used as a remote process to
lighten again the on-board computer charge.

Future work will deal with localization after kidnapping
by applying the loop-closure detection framework while the
robot is spinning around. This will be used to retrieve the
direction to follow a path after a kidnapping. We will also
add to our system sonar data to obtain a map giving an

REFERENCES

exemmas for realtime feature detection and matching,” in European
[5] S. Bazeille and D. Filliat, “Combining odometry and visual loop-
closure detection for consistent topo-metrical mapping,” International
and robot odometry,” in Proceedings of the International Conference
on Robotics and Automation (ICRA11), 2011.
maps for hybrid metric-topological slam,” Robotics and Autonomous
appearance based topological map,” in IEEE International Conference
[10] Z. Chen and T. Birchfield, “Qualitative vision-based mobile robot
visual path following exploiting visual loop-closure detection,” in
using a virtual vehicle approach,” IEEE Transactions on Automatic
localization and mapping,” in IEEE International Conference on
paradigm for model fitting with applications to image analysis and
automated cartography,” Communications of the ACM, vol. 24, no. 6,
parameterization for efficiently computing maximum likelihood maps
using gradient descent,” in Proceedings of Robotics: Science and
1996.
in Proc. of the International Conference on Pattern Recognition, 2002.
[22] D. Lowe, “Distinctive image feature from scale-invariant keypoint,”
International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110,
2004.
tree,” in Accepted for oral presentation at CVPR 2006, 2006.