Conference Paper

Surface Reconstruction and Path Planning for Industrial Inspection with a Climbing Robot

Author(s):
Breitenmoser, Andreas; Siegwart, Roland

Publication Date:
2012

Permanent Link:
https://doi.org/10.3929/ethz-a-010034943

Originally published in:
http://doi.org/10.1109/CARPI.2012.6473354

Rights / License:
In Copyright - Non-Commercial Use Permitted

This page was generated automatically upon download from the ETH Zurich Research Collection. For more information please consult the Terms of use.
Surface Reconstruction and Path Planning for Industrial Inspection with a Climbing Robot

Andreas Breitenmoser and Roland Siegwart
Autonomous Systems Laboratory (ASL), ETH Zurich, Tannenstrasse 3, 8092 Zurich, Switzerland
Email: andreas.breitenmoser@mavt.ethz.ch, rsiegwart@ethz.ch

Abstract—Inspection and maintenance tasks in the power industry often require tools to be moved smoothly in direct contact or close proximity over surfaces of curved tube-like structures. This paper addresses the problem of navigating robotic devices, namely a wheeled climbing robot carrying an inspection tool, on surfaces as they typically appear in steam chests of power plants.

The surfaces are modeled by triangle meshes. Several state-of-the-art surface reconstruction methods are evaluated, and meshes are constructed for 3D point clouds taken with different laser scanners and for tube structures varying in shape, dimension and surface property. Each mesh represents an embedded graph, and thus directly enables graph search methods to plan paths of triangle strips over the surface. Discrete path planning and continuous robot control are combined to a hybrid system, which steers the robot smoothly along the triangle strip to the target regions on the surface. The navigation approach applies to robotic surface inspection, and is verified in simulation experiments by moving a robot on triangle meshes reconstructed from real data.

I. INTRODUCTION

Scene reconstruction, robot mapping and navigation are well-studied problems in flat 2D environments. However, many real robot applications, including those of inspection and maintenance in industrial plants, require robots to move in uneven terrain or in multi-level environments, where feasible paths may pass underneath and over obstacles at the same time. Extreme examples are robots that climb on the inner or outer surfaces of curved 3D structures as they navigate in full 3D space but with the additional constraint of being attached to the surface. Such robot platforms need appropriate environment modeling and path planning techniques for their efficient and reliable operation.

The motivation of this paper is to provide a navigation solution for climbing inspection robots, such as MagneBike [1], which allows for autonomous or semi-autonomous navigation and for the inspection of target regions on a curved surface. Inspection and maintenance can guarantee the early detection of defects and the safe operation of power plants. As existing power plants are reaching the end of their designated lifespan, aging and structural damages become more and more an issue. Given the increasing need for inspection, we believe robotic agents represent assistive inspection tools with high potential for the power industry. Inspection robots may offer sensing and actuation technology for recording data at periodic inspection runs in a consistent way and over long time periods. Enhanced robot mobility, such as climbing capabilities, may result in increased information content; the inspection of unknown structures is improved, and access to structures that are inaccessible for human inspectors is provided. Furthermore, inspection robots may ease the overall process of inspection, which makes inspection operations more scalable and applicable by non-experts.

In this paper, we investigate the applicability of methods from surface reconstruction and robot path planning to the problem of robotic inspection in industrial structures. With regard to surface reconstruction, we recognize the methods that are current state-of-the-art in computer graphics and robotics, and evaluate selected methods by generating triangle meshes from noisy 3D point clouds for our industrial application. Following the surface reconstruction, we propose a triangle strip planner for planning 6-DoF paths over the reconstructed surface. A hybrid control approach, which was demonstrated for planar sequences of discrete triangular regions in [22] and [23], is applied to our scenario of path planning and control for a non-holonomic bicycle robot on a triangle mesh of 3D tube-like structures.

Robotic inspection of surfaces in 3D space has still been little investigated so far. Relevant related works include the inspection of ship hulls, e.g., with an autonomous underwater vehicle [2], the inspection of industrial piping systems, e.g., with a pole-climbing robot [3], as well as the inspection of tanks, generators and boilers as they typically appear in power plants [4], [5].

There exist plenty of surface reconstruction methods [10], [13]–[17], and it is often not clear which among those can be expected to perform well in a given environment or application. Another difficulty is that surface reconstruction methods are generally developed in computer graphics with different goals in mind. The focus is more on the user-guided generation of high accuracy meshes for 3D modeling, and
of the selected surface reconstruction methods are presented for the acquisition of 3D point cloud data. The evaluations of the selected surface reconstruction methods are presented in Section IV. Section V covers the triangle strip planner and the continuous vector field controller, which allow for the navigation on triangle meshes of curved surfaces. A final discussion and outlook conclude the paper in Section VI.

II. SYSTEM OVERVIEW

A complete overview of the presented robot navigation system is shown in the schematic of Figure 1. The meshing and path planning core system includes a 3D laser scanner unit (Lidar) to capture 3D point clouds of the environment, a surface reconstruction unit (Mesher) to generate maps consisting of triangle meshes, and a path planning and control unit (MeshNavigator) to steer the robot. Further components of the system provide additional sensor input, and enable the localization of the robot (Localizer).

The navigation system is mainly based on the two subsystems for mesh generation and path planning, which are illustrated in more detail in Figure 2. The Mesher unit provides several meshing routines that are applied to reconstruct a surface from 3D point clouds (see also Section IV). 3D point clouds received from the Lidar unit may be filtered prior to the mesh generation step. After reconstruction, triangle meshes are further processed by mesh filters to condition the mesh for following applications. The input to the MeshNavigator unit consists of the 6D robot pose, the desired goal pose and the previously generated and processed triangle mesh. The implemented triangle strip planner is based on the classical A* and Dijkstra’s algorithms and plans a global path over the surface. The vector field controller builds on the continuous vector field generation method from [23] and controls the robot within the planned triangle strip toward the goal region (see Section V).

The navigation system integrates the point matcher library [7], and depends on the external libraries Eigen1 for linear algebra, Nabo2 for nearest neighbor search and OpenMesh3 for mesh representation. OpenMesh provides an efficient halfedge-based data structure for representing and operating on triangle meshes. The system furthermore offers optional interfaces to the powerful computational geometry, point cloud and mesh processing libraries CGAL4 and PCL5. For this paper, no CGAL implementations are used, but two of the meshing routines, the Poisson surface and the fast triangulation meshing routines, make use of PCL.

In our specific case, the navigation system applies to the MagneBike inspection robot [1], shown in the bottom left of Figure 3. MagneBike is a compact and lightweight climbing robot (185 × 143 × 170 mm, 3.3 kg), which consists of two magnetic wheel units in a motorbike arrangement. MagneBike’s locomotion and adhesion concepts with an active degree of freedom on the front steering wheel and permanent magnets integrated into both wheels allow to drive on ferromagnetic 3D surfaces of industrial environments, which were not designed specifically for robots. MagneBike is able to climb vertical walls, follow circumferential paths inside pipe structures and pass over step-like obstacles. The motorbike design makes turning on spot by rotating about the rear wheel possible, which further contributes to the robot’s high mobility.

MagneBike supports a rotating 3D laser scanner to record 3D point clouds from the environment. This 3D data is crucial for environment modeling as well as localization of the robot. The 6D pose of the robot is required by the Mesher unit for extending an existing mesh map (e.g., by merging several meshes) and by the MeshNavigator unit for updating the robot pose in the control loop. A possible realization of the Localizer unit takes point cloud and odometry data, and runs an iterative closest point (ICP) algorithm [7] to align single overlapping 3D point clouds, or triangle meshes respectively. A detailed

1Eigen library, http://eigen.tuxfamily.org
3OpenMesh library, http://www.openmesh.org
5Point Cloud Library, http://pointclouds.org
III. 3D LIDAR POINT CLOUDS

3D point clouds provide the basic measurement data for robot localization and mapping, modeling and visualization of the environment. In our application of robot navigation in the tube-like environments of power plants, we use small light detection and ranging (LIDAR) devices, composed of rotating 2D laser scanners, for the 3D data acquisition. For the use with compact mobile robots like MagneBike, the sensors must be low-power, lightweight and small-scale. The Hokuyo URG-04LX and UTM-30LX laser scanners represent today’s standard in this segment, and are also used on the MagneBike robots. As reported in [6], LIDARs are appropriate sensors for the inspection of industrial tube structures since dark environments or rather homogeneous metallic surfaces only provide a limited amount of natural features.

The limitations of the laser scanners for these environments are caused by limited sensor accuracy and strength. The recorded point clouds are usually noisy, anisotropic and show high variations in density. In addition, specular reflections occur on metallic surfaces, which results in increased deformations of the 3D point clouds. These reflections do not only depend on the surface itself, but also change with the type and characteristics of the laser scanners used. Figure 3 shows 3D point clouds obtained with the Hokuyo URG-04LX, Hokuyo UTM-30LX and SICK LMS-200 laser scanners, which provide varying measurement accuracies and laser strengths. A detailed analysis of LIDAR performance and its change for different range sensors and different surfaces in industrial environments can be found in our companion paper [8].

After a 3D point cloud is captured, the point cloud is augmented in a pre-processing step. The Mesher unit provides different filters to improve the point clouds, including equalization of varying point densities, removal of outlier points, and subsampling and trimming in order to reduce the point cloud data. Augmenting the point cloud by point normals is a common requirement for many mesh generation techniques. Especially implicit surface reconstruction methods rely on robust normal estimation for computing signed distance functions. Point normals are estimated by fitting local planes to the point neighborhoods. A principal component analysis (PCA) on the scatter matrix of the point neighbors estimates the point normal as the eigenvector of the scatter matrix with the smallest corresponding eigenvalue. The resulting normal estimates need to be adjusted to obtain consistent normal orientations of neighboring points. Single point clouds recorded from rotating laser scanners include some inherent additional information about the organization of the point cloud. The points must be visible for the sensor, the point cloud origin coincides with the sensor position, and the recorded data is associated with a particular position of the moving sensor. The point normals can be oriented by using viewpoint information. Alternatively, for multiple point clouds and no available additional sensor information, globally consistent orientations of the normals are obtained by propagating the normal orientations over a minimum spanning tree [10].

IV. TRIANGLE MESHES FOR SURFACE REPRESENTATION

While 3D point clouds can directly be used as representations of a surface and path planning on 3D point clouds is possible [9], we take here an alternative approach of reconstructing the surface. The reconstructed triangle mesh provides an embedded graph, which serves as an ideal representation when it comes to path planning. Moreover, the presented robot navigation method is particularly intended for inspection tasks. A mesh offers additional advantages since it is a convenient data structure for mapping the found defects, for logging the already covered trajectories, and for visualization purposes in the interaction with human inspectors.

Surface meshes are constructed by the mesher routines in the Mesher unit, using different state-of-the-art surface reconstruction methods. In order to explore the suitability of
a wide spectrum of methods for their practical use in an industrial setting, local sensor-centric as well as global, explicit as well as implicit algorithms are investigated. Examples of triangle meshes, generated from the point clouds of the industrial structures in Figure 3, are shown in Figure 4.

A. Irregular triangular meshes

The irregular triangulation mesher (ITM) routine is the most simple surface reconstruction method evaluated in this paper. It follows the idea introduced in [11] for mesh generation for robots in rough terrain. ITM is a local explicit approach; it is sensor-centric, i.e., it works on a single laser scan with additional viewpoint information. Points are explicitly connected by triangles through Delaunay triangulation. First, the 3D point cloud is transformed into spherical coordinates and a 2D Delaunay triangulation is applied to the two angular coordinates $\phi$ and $\theta$, omitting the range component. The connectivity among the points is established in this first step. The resulting triangles are close to equilateral as the Delaunay triangulation maximizes the minimal angle of each triangle. In a second step, range information is added back to generate the final triangle mesh in 3D space. The established connectivity from the first step remains valid. The periodicity in the angle of the spherical coordinates is retained in the generated mesh by duplication and identification of $(\phi, \theta)$ coordinate pairs when the 2D Delaunay triangulation is performed. ITM interpolates the point cloud, i.e., it keeps the original points of the point cloud as vertices of the mesh and thus exactly represents the real input data. The mesh generation works for non-uniform point densities but is sensitive to noisy data. Typically, the generated meshes contain holes and artifacts, such as incorrect triangle connections at the point cloud boundary or at occluded regions. Meshes obtained by ITM require post-processing in order to be usable for robot navigation. The Mesher unit provides a mesh filter based on suggestions in [11] for the removal of incorrect artifact triangles.

B. Fast triangulated surfaces

The fast triangulation mesher (FTM) routine implements the explicit surface reconstruction method presented in [12]. The method is based on a greedy algorithm that works under the principle of incremental surface growing: a start triangle is created and new triangles are formed and added explicitly until all points in the point cloud are included in triangles in the mesh, or no more valid triangles can be constructed. The triangulation method partially relies on the triangulation algorithm presented in [13]. First, the k-nearest neighbors of a single point in the point cloud are selected. In a second step, the points in this neighborhood are projected onto a plane, which is fitted to the points in the neighborhood. Finally, new triangles are built by connecting the points through edges, while visibility, and maximum and minimum angle criteria set constraints on the triangle creation. FTM, similar to ITM, is a fast surface reconstruction method that works for non-uniform point densities and provides near real-time performance. It offers improved robustness against noise but still results in much rougher meshes than the implicit methods. The incremental nature of FTM promises the straightforward extension of a mesh by point clouds from new laser scans.

C. Meshing with RBF- and MLS-based distance functions

The mesher routines we describe next are global implicit surface reconstruction methods based on differently defined signed distance functions. The extended marching cubes mesher (ECM) routine implements two common choices of signed distance functions: the radial basis function (RBF) [14] and moving least-squares (MLS) [15]. Once the distance function computation has completed, the triangle mesh is retrieved by the marching cubes algorithm. We use an extended marching cubes implementation [16], together with triharmonic or piecewise polynomial RBFs and MLS kernels with Gaussian or Wendland weight functions, in our ECM mesher routine.

In contrast to explicit methods, implicit methods generate a mesh as a global approximation of an input point cloud. Whereas RBF-based methods compute an approximation globally, MLS-based methods achieve the approximation by local computation. Different selections of the kernel functions differ in their local or global support, and determine the overall smoothness of the resulting signed distance function and the generated triangle mesh. Implicit methods reconstruct watertight or closed surfaces. In the context of robot navigation, this is advantage as well as disadvantage: noisy regions of the point cloud, regions of highly irregular density or with missed parts can widely be recovered, but approximation may incorrectly fill in a hole or opening that is truly present in the
real surface. Therefore, implicit methods as well require post-
processing of the generated triangle mesh. Mesh filters may
use prior knowledge of the surface or additional information
extracted from the augmented point cloud in order to decide
if parts of the mesh need to be cut.

D. Poisson surface reconstruction

The Poisson surface mesher (PSM) routine represents a
global implicit method, which is based on the Poisson sur-
face reconstruction algorithm introduced in [17]. The Poisson
surface reconstruction algorithm is one of the most recent and
best performing surface reconstruction algorithms in computer
graphics literature to date.

The formulation of PSM relies on the indicator function,
a function that distinguishes inside from outside space in a
3D point cloud recorded from a surface of an object. The
gradient of such an indicator function can be considered
as a vector field that takes non-zero values solely in the
proximity of the real surface. The indicator function is found
as the function whose gradient approximates the vector field
computed from the sample points. By applying the divergence
operator, a Poisson equation results, which can be solved
efficiently by conventional Poisson solvers [17]. PSM’s basis
functions are compactly supported and the implicit function is
inherently constrained at all spatial points without addition of
off-surface constraints. Similar to ECM routines, PSM depends
on augmented point clouds with consistently oriented point
normals, is apart from that robust against noisy data, and
results in the generation of watertight meshes, including all
the associated benefits and drawbacks discussed above.

V. PATH PLANNING ON SURFACE MESHES

Once the surface has been reconstructed, the generated
triangle mesh is passed on to the MeshNavigator unit for path
planning. The mesh serves as representation or map of the
environment and provides a graph embedding on the surface,
which enables robot path planning. The mesh is usually filtered
prior to planning by a post-processing step in the Mesher unit
in order to improve navigability. For this paper, we limit the
post-processing step mainly to mesh simplification. We apply
the QSlim mesh simplification algorithm [18] to obtain a mesh
with triangles that are reasonably scaled for navigation.

Tasks of a climbing inspection robot like MagneBike are
versatile: they span from exploration and map building to
surface coverage and exhaustive search for defects, to the
analysis and repair of defects at specific points in the structure.
The underlying problem is always to plan a path on the
surface; however, depending on which of the tasks needs to be
performed, the focus is either on planning a transitioning path
to a goal region, or planning a covering path that moves the
robot over the complete area. We address both of the problems
by planning triangle strip paths over the surface (see Figure 5).

A. Triangle strip path planning

The triangle strip planner implements a graph search method
to plan an initial discrete path from a start to a goal position
on the mesh by either connecting the triangle centroids or the
edge centers of adjacent triangles. This results in a connected
sequence of triangles, or a triangle strip. We use standard
graph search methods like A* and Dijkstra’s algorithms to
connect the triangles to a strip, and hence generate a discrete
initial path from a start to a goal point. Weights of traversing
cost can further be derived from local mesh properties and be
included in the graph.

B. Controlling a robot onto smooth paths

In a second step, a continuous vector field is computed
within each triangle face of the strip, such that a smooth
path over the mesh is obtained by following the vector field’s
gradient.

Graph-based planners, as the ones described above, perform
well in terms of finding an optimal global path. However, the
resulting paths are discrete and often it is desired to have
more continuous trajectories and control laws that are directly
applicable to the actual robot. An interesting direction is to
take advantage of both discrete and continuous algorithms by
formulating a hybrid system.

Given the discrete path along the triangle strip, a vector
field for steering the robot from triangle to triangle can be
computed by interpolation within each triangle face following
the concepts in [22], [23]. For each triangle, a set of three
base vectors, \( \mathbf{v}_{q_i}, \mathbf{v}_{q_j} \) and \( \mathbf{v}_{q_k} \), one for each triangle vertex,
is chosen, such that the vectors are guaranteed to always
point along the triangle strip toward the goal without having
positive projections to the lateral outward normals of the
strip. Following [23], it is possible to adjust the single base
vectors by additional triangle split operations, such that a fully
continuous vector field over the whole triangle sequence is
guaranteed.

Fig. 5. Planned triangle strip paths on industrial structures. Left: A triangle
strip is planned from a start to a goal region on the simplified triangle mesh;
the mesh from the bottom right of Figure 4 (generated by PSM) was reduced
from 25'000 to 1’000 faces by QSlim. A close-up (bottom left) shows the
discrete A* path connecting the triangle centroids, the computed vector field
and the simulated smooth path of the robot along the triangle strip. Right: Six
transitioning paths (top right) all starting at the original scan location guide
the robot to different target regions. A covering path (bottom right) composed
of several triangle strips in sequence covers parts of the surface. Note that
the covering here is semi-autonomous; new targets to cover are successively
specified by a human operator.
Given the partial triangle areas $A_i$, $A_j$, and $A_k$, the vector field $\mathbf{u}(q) = (A_i \mathbf{v}_{q_i} + A_j \mathbf{v}_{q_j} + A_k \mathbf{v}_{q_k})/(A_i + A_j + A_k)$ is created in each triangle of the triangle sequence from the start to the goal. For a holonomic point robot, a simple proportional controller can be derived by $\dot{q} = K \mathbf{u}(q)$. The case of nonholonomic robots, such as MagneBike, is handled by feedback linearization, as further explained in [23].

In our case of vector field generation over 2D manifolds, the vector field is fully continuous in the triangle strip unfolded to a common plane, and only exhibits discontinuities in the relative orientation of adjacent triangle faces. The discontinuities are inherent to the discrete approximation of the continuous curved surface by the triangle mesh. A possibility to achieve continuity in orientation could be found in interpolating between the orientations of two adjacent triangle faces.

VI. DISCUSSION, OUTLOOK AND CONCLUSION

This paper has presented a method for navigating inspection robots on curved surfaces. The Mesher unit models the surface as triangle mesh and the MeshNavigator unit applies a triangle strip planner to plan discrete paths from start to target regions as a sequence of triangles. Feedback control is directly added to the planning procedure in constructing a continuous vector field. The hybrid approach seems promising as it incorporates several favorable elements for robotic inspection, such as mesh representation of the environment, efficient discrete planners, and robustness through integrated feedback control.

Several state-of-the-art surface reconstruction methods have been applied to 3D point clouds recorded from industrial tube-like structures. Whereas explicit methods require effective filtering and outlier removal methods in order to create a navigable mesh, implicit methods rely on robust normal estimation and means for detecting missed real features through too strong approximation and smoothing.

Further steps are to be taken to address the current limitations and open problems of the navigation approach. Concerning the Mesher unit, a particular meshing method needs to be further developed and adjusted to the specific application. Furthermore, support for mesh updates must be included. The triangle strip path planning needs to incorporate additional constraints present in the motion planning problem. Curvature constraints could be taken into account on the discrete level of triangle strip planning. The planning of covering triangle strips must be automated. Another direction is toward obstacle negotiation and avoidance, including 3D collision checks against the triangle mesh. We assumed the robots move on well-shaped curved surfaces and there exists enough clearance in the tube interior. For robust operation, obstacles need to be detected and handled by additional controllers in full 3D, while the robot is moving on the 2D manifold.

ACKNOWLEDGMENT

This research is supported by ALSTOM. The authors would like to thank Dario Fenner and Markus Bühler for great technical support, and Roland Moser for providing his advice with regard to the industrial inspection of power plants.

REFERENCES