Semantic Mapping: Segmenting a Concave Structure to Convex Structures

Chahat Deep Singh
Robotics Graduate Student
University of Maryland
chahat@umd.edu

Anton Mitrokhin
CS Graduate Student
University of Maryland
anton.mitrokhin@phystech.edu

***Abstract*—Iterative closest point (ICP) is a popular algorithm employed to register two sets of curves, two sets of surfaces, or two point clouds. The Iterative Closest Point (ICP) algorithm that uses the point-toplane error metric has been shown to converge much faster than one that uses the point-to-point error metric. At each iteration of the ICP algorithm, the change of relative pose that gives the minimal point-to-plane error is usually solved using standard nonlinear least-squares methods, which are often very slow. For small relative orientation between the two input surfaces, we can approximate the nonlinear optimization problem with a linear least-squares which can be solved more efficiently. Once the 3D rigid-body transformation is finished and the model of the scene is reconstructed, we tackle the segmentation problem in a cluttered environment. Assuming the objects in the scene are convex in nature, we break down the entire concave cluttered scene into convex structures.**

***Keywords*—Convex Hull, Iterative Closest Points, Regional Convolutional Neural Network, Delaunay Triangulation**

I. Introduction

3D shape alignment is an important part of many applications. It is used for object recognition in which newly acquired shapes in the environment are fitted to model shapes in the database. The ICP technique was proposed independently by Besl and McKay [1] and Zhang [2] in two different contexts. Besl and McKay [1] developed the ICP algorithm to register partially sensed data from rigid objects with an ideal geometric model, prior to shape inspection. So this is a subsetset matching problem because each sensed point has a correspondence in the ideal model. Zhang [2] developed the ICP algorithm in the context of autonomous vehicle navigation in rugged terrain based on vision. His algorithm is used to register a sequence of sensed data in order to build a complete model of the scene and to plan a free path for navigation. So this is a subsetsubset matching problem because a fraction of data in one set does not have any correspondence in the other set.

In this paper, the authors implemented *Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration* by Low[3]. This form of ICP is often used to reconstruct 2D or 3D surfaces from different scans which reasonably low orientation difference. In the Iterative Closest Point or, in some sources, the Iterative Corresponding Point, one point cloud (vertex cloud), the reference, or target, is kept fixed, while the other one, the source, is transformed to best match the reference. The algorithm iteratively revises the transformation (combination of translation and rotation) needed to minimize an error metric, usually the distance from the source to the reference point cloud. ICP is one of the widely used algorithms in aligning three dimensional models given an initial guess of the rigid body transformation required.

Subsequently, once the 3D scene is reconstructed, each object needs to be segmented. There are many techniques- both in 2D and in 3D. Segmentation techniques like HOG feature detector, Bag of Visual Words, Textons, k-means, mean shift clustering, graph-based min-cut/max-flow technique, Support Vector Machines, Markov Random Fields, CRF and Deep Learning are commonly used for segmenting RGB images.

II. Pre-Processing

In this step, we detected the long planes i.e. the table top and the walls. Removing these before the ICP step makes it much faster.

Fig. 1. Large Planes are detected in the background

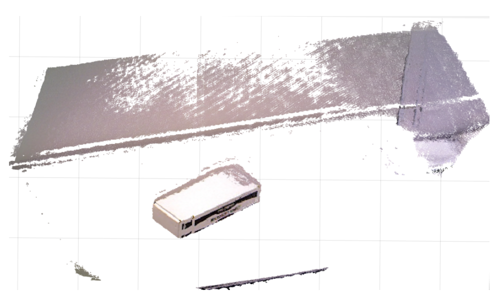

Fig. 2. Plane Removal

III. Iterative Closest Point

Iterative Closest Point (ICP) [1][2][3] is an algorithm employed to minimize the difference between two clouds of points. ICP is often used to reconstruct 2D or 3D surfaces

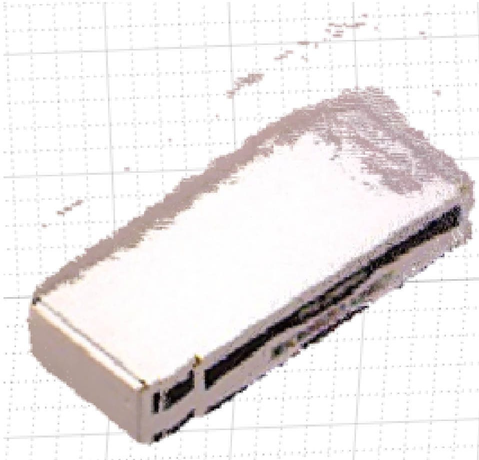

Fig. 3. Plane Removal

from different scans, to localize robots and achieve optimal path planning to co-register bone models, etc.

Essentially, the algorithm steps are as follows: (a) For each point (from the whole set of vertices usually referred to as dense or a selection of pairs of vertices from each model) in the source point cloud, Match the closest point in the reference point cloud (or a selected set). (b) Estimate the combination of rotation and translation using a root mean square point to point distance metric minimization technique which will best align each source point to its match found in the previous step after weighting and rejecting outlier points. (c) Transform the source points using the obtained transformation. Iterate (re-associate the points, and so on).

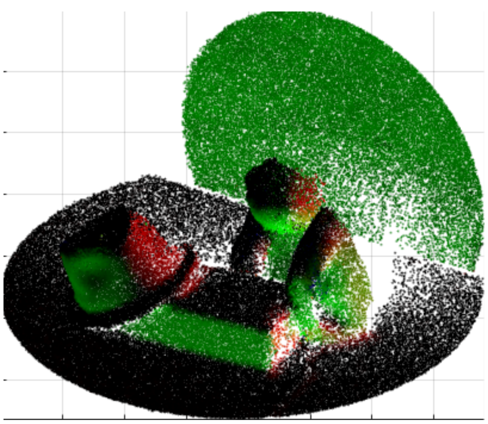

Fig. 4.

Zhang [3] proposes a modified K-D tree algorithm for efficient closest point computation. In this work a statistical method based on the distance distribution is used to deal with outliers, occlusion, appearance, and disappearance, which enables subset-subset matching.

A. Post Processing:

Remove all Point Cloud with lower Density less than threshold:

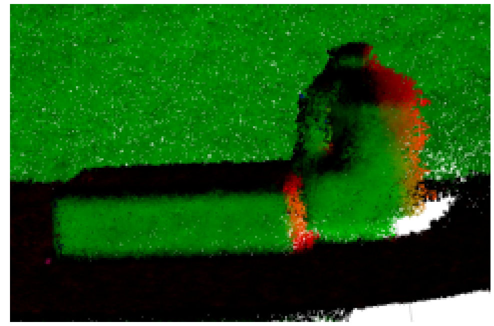

Fig. 5.

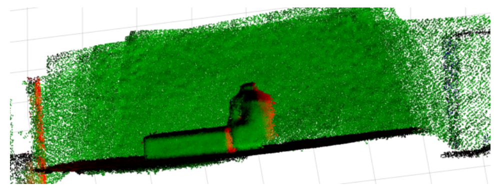

Fig. 6.

if density(idx) $<$ threshold

then PtCloud(idx) $= 0$;

IV. Segmentation

A. Convex Hull

A convex hull or a convex envelope of a set X of points in the Euclidean plane or in a Euclidean space (or affine space) is the smalled possible convex set that contains X. Alternatively, convex hull is defined as the intersection of all convex sets containing X or as all convex combinations of points in X. With the latter defination, convex hulls may be extended from Euclidean spaces to arbitrary real vector spaces.

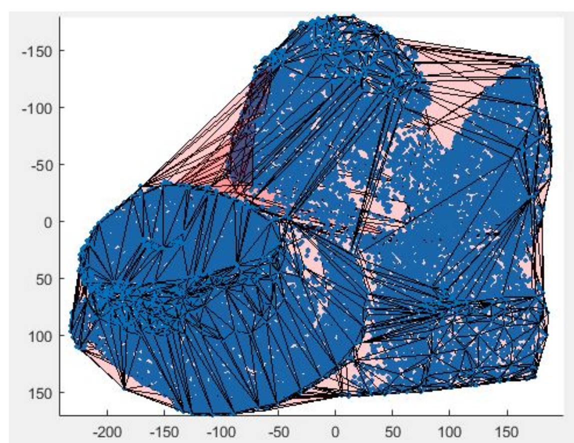

Fig. 7. Convex Hull to a Concave Structure

1) Critical Points Detection: Corners are very useful features in image matching, image representation and shape analysis. Therefore, critical points play an important role in computer vision for many applications such as object recognition, motion analysis and scene analysis. Both convex and concave points are considered important as far as locating the critical points on the boundary of the object shape is concerned. Hence, to detect convex and concave points the following steps are proposed as described next.

2) Concave and Convex Points Detection: This section presents the details of the proposed approach that combines boundary information and skeleton of the original image in order to detect convex and concave points separately. We find a set of concave points and convex points. Let S be a subspace of a discrete n-dimensional space. A function $f : S \to R$ is discretely convex if $\forall x, y \in S$ and $\forall a \in (0, 1)$:

$$\alpha f(x) + (1-\alpha)f(y) \geq min_{u \in N(z)}\ f(u)$$

where $N(z) = u \in S : ||u - z|| < 1, z = \alpha x + (1-\alpha)y$ and $||u|| = max_{1 \leq i \leq n|u_i|}$. For every other value, the points will be concave.

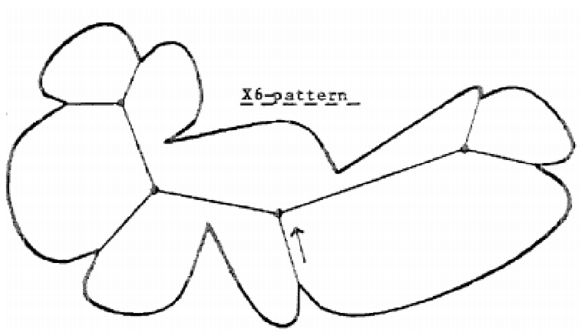

Fig. 8. Concave to convex Decomposition

For any non-complex concave structure, like in fig. 8, the algorithm mentioned in [5] proves that decomposition into n convex structure can be done in polynomial time. Thus, we try to find the best planes so as to cut the concave structure into a set of convex structures.

B. Boundary Detection

Now, rather than finding the critical point or *voxel*, we propose to perform a triangulation with each
edge length $<$ threshold$_1$. In fig. 9, we can see a concave boundary or triangulation (with small edge length) on the same structure.

Now, rather than finding concave points as mentiooned in section IV.2

Alternatively, for object segmentation, an object detector was trained using a *Faster Regional Convolutional Neural Network (R-CNN)*. Faster R-CNN is an extension of the R-CNN and Fast R-CNN object detection techniques. All three of these techniques use convolutional neural networks (CNN). The difference between them is the way each algorithm selects regions in order to process and the way those regions are classified. R-CNN and Fast R-CNN use a region proposal algorithm as a pre-processing step before running the CNN. The proposal algorithms are typically techniques such as EdgeBoxes or Selective Search , which are independent of the CNN. In the case of Fast R-CNN, the use of these techniques becomes the processing bottleneck compared to running the CNN. Faster R-CNN addresses this issue by implementing

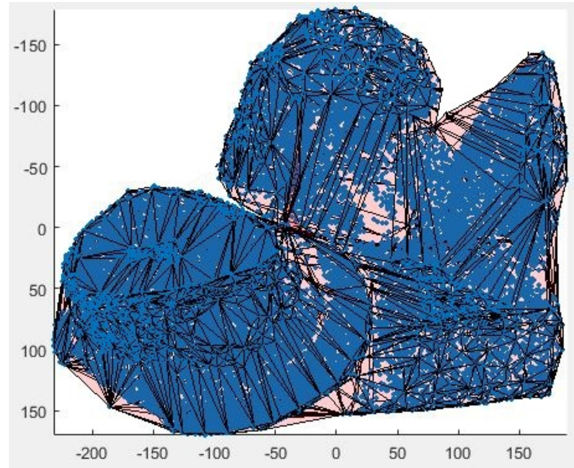

Fig. 9. Boundary detection using triangulation

the region proposal mechanism using the CNN and thereby making region proposal a part of the CNN training and prediction steps.

V. SEMANTIC MAPPING

Based on the number of pixels, one can ask about total size of each object and maximum volume possible. One can talk about dominating colors. Also, about which one of the object most resembles a sphere.