**Automated Weld Path Generation Using Random Sample Consensus and Iterative Closest Point Workpiece Localization with Low-Cost 3D LiDAR**

**Abstract**

Jobs performed by small to medium enterprises (SMEs) are infrequently automated due to high setup costs and lack of technical expertise needed for robot training, however productivity and worker safety can be improved in SMEs with the use automated tooling. In a traditional automated manufacturing environment, tasks such a welding or painting are accomplished through execution of pre-programmed tool motions which rely on the location and orientation of the workpiece to be fixed and known. The lack of this spatial information is typically treated through positioning of the workpiece with respect to the robot arm using jigs or fixtures which are costly in initial setup and not easily modified. Further, the resulting toolpath associated with a desired task is typically defined through manual teaching resulting in a path appropriate for an individual job. For this reason, SMEs requiring variation in part geometry or arrangement are not commonly automated. This work presents a method for automated weld path generation for a 6DOF co-bot arm using random sample consensus (RANSAC) and iterative closest point (ICP) workpiece localization from LiDAR pointclouds. Scans from a low cost 2D LiDAR mounted to the co-bot arm are used to generate 3D pointclouds of the workspace scene with the Robot Operating System (ROS). The Point Cloud Library (PCL) is used to compare the generated pointcloud with a CAD model to produce a rigid transformation to localize the workpiece. The estimated pose of the workpiece with respect to a fixed frame is used offline to generate a weld path as series of tool poses. Two example welding processes in which a cylinder or rectangular tube is joined to a flat plate and two square tubes are joined through weldment are investigated and a physical implementation of the method is demonstrated using a 2D LiDAR mounted to a 6DOF co-bot carrying a MIG welding torch.

**Introduction**

Small to medium enterprises perform manufacturing tasks associated with relatively low part volume and increased variation in assembly geometry as compared to jobs performed in large scale manufacturing environments. This type of manufacturing operation is infrequently automated due to high setup costs; however, productivity and worker safety can be improved in small to medium enterprises with the use of automated tooling. (Co-bot and Lasers too!?!?)

In a traditional automated manufacturing environment, a task such a welding or painting is accomplished through execution of pre-programmed tool motions which rely on the location and orientation of the workpiece to be fixed and known with respect to a global coordinate system. The need for spatial information is typically treated through positioning of the workpiece with respect to the robot arm using jigs or fixtures which are costly in initial setup and are not easily modified. In large scale production environments, this can be accomplished with dedicated infrastructure built into the environment such as moving jigs on assembly lines and other features available in a highly structured environment. Further, the resulting toolpath associated with a desired task is typically defined through manual teaching resulting in a path appropriate for an individual job.

For this reason, SMEs requiring variation in part geometry or arrangement are not commonly automated. This paper presents a method for automated weld path generation for a 6DOF co-bot arm using random sample consensus (RANSAC) and iterative closest point (ICP) workpiece localization. implemented using the Robot Operating System (ROS) and the Point Cloud Library (PCL). Point Cloud Library provides an open-source C++ implementation of several 3D point cloud and image processing algorithms including: filtering, feature estimation, surface reconstruction, registration, model fitting, and segmentation. [5] (object recognition also). This library is an attractive research tool because it is stable and integrates with ROS and includes example code and standard data sets that can be used for comparison repeatability in research.

This work employs a collection of these algorithms to locate, or register, a point cloud representing the workpiece in a point cloud of the working environment collected by a LIDAR scanner located on the robot. Once the known part is located with respect to a fixed frame, an automated weld path generation routine is used to plan a weld tool-path offline. A representative set of welding processes in which a cylinder or rectangular tube is joined to a flat plate through weldment is investigated and a physical implementation of the method is demonstrated using a 3D LIDAR mounted to a 6DOF co-bot carrying a MIG welding torch.

As technology advances humans and robots must adapt to remain relevant in our respective environments and we are currently seeing this in the emergence of the co-bot workcell paradigm.

A pointcloud is a list of points in 3D space representing a physical object or collection of objects [find citation], and this data is generated through measurements from a sensing device such as a LiDAR or 3D camera. These geometrical data, or features of a Pointcloud contain the locations of the boundaries of a solid object. Features may also include point normals which can be measured or inferred from the feature locations. Non-geometrical data such as color or other surface properties that are independent of the transformations between features are known as descriptors [review of point reg algs]. The two types of data contained in a pointcloud are stored separate because they are treated differently in processing algorithms such as segmentation or registration.

These environment sensing devices are frequently used in the mobile robotics industry, and improved sensors and being developed with the increased demand [need citation] for automation in manufacturing and transportation. Widespread applications and research involving spatial data has led to the development of standard file types and storage containers for efficiently processing and transferring pointclouds [PCL citation]. The .pcd file is used for permanent storage and several pointcloud data types exist in common programming languages (C++, Python, MATLAB) for integration with various libraries (PCL, OpenCV) and software frameworks (ROS,). Although, there is an effort to standardize these data types many variants are used. Conversion between different types is common feasible because they are all equivalent representations of a collection of features and descriptors.

**Manufacturing Application**

A manufacturing task is considered, in which a weldment is performed on a work-piece resting on a welding table. The work-piece in this task typically consist of multiple components to be joined through weldment. The relative alignment of the multiple components of the work-piece is assumed to be correct within the physical constraints of the designed part prior to the automated process. In practice this alignment is set by the operator and secured using clamps or other fixtures.

Variation in surface quality and workpiece dimension and shape are likely present however these are not the focus of this process. The workpiece geometries are generally assumed to match those in the model within a working tolerance. These local model inaccuracies certainly affect the global information produced regarding the geometry and location of the weld, but these affects are minor.

Two example applications are considered. In the first of which, two square tubes are joined perpendicular to one another with a fillet weld along two of the shared edges. In the second application a round cylinder is joined to a flat plate with a fillet weld along the shared circular edge between the two components.

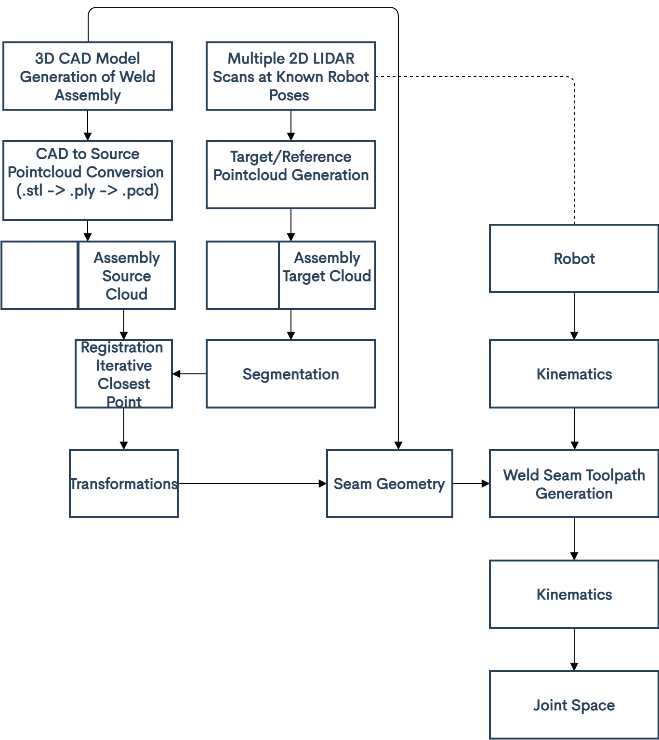


Figure Method for Automated Weld Path Generation

**Automated Weld Path Generation Approach**

The proposed approach to automated weld path generation shown in **Figure 1** consists of a model data preparation stage, a workspace sensing stage, a work-piece localization stage, followed by an offline robot path generation stage. The resulting path can be used to automate a welding process on the component in the workspace with a 6-DOF co-bot carrying a welding torch.

**Model Data Preparation Stage**

In the model data preparation stage, the geometry of the workspace and the work-piece is defined based on the prescribed application. An ideal model of the work-piece including the weldment is generated using CAD. Part models are first generated of the individual work-piece components which are then assembled to represent the work-piece. The CAD assembly representing the work-piece is converted into a pointcloud through a uniform sampling techniques to be used for work-piece registration. The pointcloud associated with the CAD model is known as the source pointcloud.

A simplified model of the workspace and environment including the welding table and the robot base is also created for simulation purposes, and the environment model is also converted into a pointcloud file.

A typical CAD application is used to generate 3D models which can be exported as .ply files or other standard file formats.

**Workspace Sensing Stage**

Prior to the sensing stage, the workpiece is placed in the robot workspace by the operator in the proper relative orientation to be joined by a weldment. The relative orientation of the parts must match that of the model to an extent and the global location of the workpieces is restricted to the usable workspace of the robot.

In the sensing stage a sweeping motion of the arm is performed, and the workpiece and environment are scanned with the 2D LiDAR mounted on linkS of the robot. Multiple 2D lidar scans are measured along with corresponding sensor poses at (linkS). As the scanning stage continues the data are transformed from the sensor frame linkS to the base frame link0 through the robot forward kinematics and accumulated into a 3D pointcloud with respect to the base frame. This process produces sparse data sets with redundant points. Therefore, the scans are filtered and downsampled [verify] to improve results and reduce the resource requirements of storage and processing. The resulting pointcloud contains an image of the work-piece and fixtures as well as the top of the welding table and the background. The pointcloud associated with the LiDAR scan is known as the reference or target cloud.

The robot and sensor are operated simultaneously using ROS, and the resulting pointcloud is saved as a .pcd file which can be processed using the Point Cloud Library or converted to a .ply polygon file.

**Work-Piece Localization Stage**

In the work-piece localization stage, the **source** pointcloud derived from the CAD model is compared to the **reference** pointcloud acquired in the sensing stage. The rigid transformation between the two clouds is found using the iterative closest point (ICP) algorithm. The result gives the pose of the workpiece with respect to the fixed frame on the robot which can be used to determine the required location of the weld seam in a global sense. This strategy is effective if correct correspondences between pointclouds can be determined and ICP converges.

ICP will converge slowly or fail to converge if outliers are present in the input data sets or if the initial transformation used for the search contains significant angular error [7]. To account for these known issues, the scene cloud is prepared with series three different filters to reduce the data to a pointcloud representing the workpiece alone before searching for the rigid transformation with the Iterative Closest Point algorithm. Bounding box, Voxel Filter, and RASAC segmentation

The scene pointcloud collected in the sensing stage contains objects other than the work-piece, such as the welding table and clamps or fixtures.. If the outliers are removed from the reference cloud so that the two clouds contain roughly the same image the transformation can be reliably found with an appropriate initial transformation. Similarly the convergence is The initial transformation used in the search significantly affects the performance due to local minimum or maxima in objective function[need citation].

Different methods have been shown for reducing or down sampling images while still retaining the useful features [].

The geometry and location of the desired weld seam in the workspace is required for offline generation of an appropriate toolpath. This information is determined by measuring the pose of the two individual workpieces with respect to a frame fixed to the robot base. Once this is information is known the appropriate toolpath can be generated. Determination of the poses of the individual workpieces represents to majority of this work and the method is described in detail.

The primary challenge in the localization stage is determining the appropriate

**Implementation**

Iterative Closets Point represents a general class of algorithms in which involves association solving and transformation minimization between pointclouds is used to represent one pointcloud in the coordinate frame of the other.

The objective of the original ICP algorithm is to find a rigid transformation, with which the reference cloud is in the best alignment with the source pointcloud set. This is done by iteratively registering the reference cloud with the source cloud by applying a rotation matrix R and a translation vector *t.*

This process begins with an initial rigid transformation, and then repeats the above two steps until the clouds have reached convergence. “However, it has been proved that this method is locally convergent, which means that the algorithm is easily failed when the rotation angle between two point sets is large. For this reason, a good initial transformation is so important that it guarantees that the algorithm converges to the global minimum finally” [7].

What steps and why we chose them.

Prefiltering Stage with RANSAC prior to ICP

Do not get very deep into the steps of ICP (main steps) – Point-to-point, kd-tree searching, and SVD. All for registration

Section discussing the process of getting the CAD to cloud

Sampling

Possible sampling methods in PCL:

1. index space sampling (taking every n-th point)
2. Uniform sub-sampling in the input 3D space
3. Sampling in the space of local surface normal

Correspondence Matching

1. kd-tree rapid matching \*\*
2. PCL depends on FLANN (open source library for fast (approximate) nearest neighbor searches)

Rejection (Filtering)

1. Distance Based \* (Max correspondence Distance)
2. Median Distance based
3. Duplicate target matches
4. RANSAC-based rejection \*\*

Alignment

1. Point-to-point \*\*

Termination Criteria

1. Max iterations
2. Absolute transformation threshold
3. Relative transformation threshold
4. Maximum number of similar iterations
5. Relative mean square
6. Absolute mean square error

Rigid transformation composed of rotation R and translation t

Using point-to-point error metric instead of point-to-plane

**Correspondence Matching and Alignment**

Let be the CAD pointcloud set for to be aligned with the source cloud set for where , and where each point corresponds to the point with the same index. The mean square objective function to be minimized is

If the correct correspondences are known, the correct relative rotation/translation can be calculated in closed form. The close form implementation from PCL consists of using singular value decomposition (SVD). The process of using SVD to begin minimizing distances between corresponding point requires the center of mass be calculate for each point set where, and

So, th

For the reference point set and the target point set, respectively. The idea is to subtract the corresponding centers of mass from every point in the two point sets before calculating the transformation. The resulting point sets are given as:

Let

denote the singular value decomposition (SVD) of W by:

where U, V are unitary, and are the singular values of W

Transformation from source to reference of part1

Transformation from source to reference of part2

CODE QUESTION

Set TransofrmationEpsilon

Set EuclidianFitnessEpsilon

**ICP – Iterative Closest Point –**

* “Reference” cloud ( Target ) is fixed
* “Source” is transformed to match reference
* The data association (step1) is the most expensive part. It can be done in different ways.
  + Closest Point (kd-Trees)
  + Normal Shooting
  + Closest Compatible Point
  + Projection Based
* The error metric can also be calculated in different ways.
  + Point-Point
  + Point-Plane
* There are also other variants.
  + Point subsets
  + Weighting correspondences
  + Data association
  + Outlier pair rejection
* Here is the basic ICP algorithm:
  1. For each point in the **source**, find the closest point in the **reference**.
  2. Estimate a transformation **T** using root mean squared minimization to best align each **source** point to its **corresponding** reference point found in step 1.
  3. Transform the **source** points using **T** from step 2.
  4. Back to step 1 (re-associate).

**Results**

“PCL incorporates methods for the initial alignment of point clouds using a variety of local shape feature descriptors as well as for refining initial alignments using different variants of the well-known Iterative Closest Point (ICP) algorithm.” [6]

Registraction (ICP or TEASER): The registration stage is required for the alignment of two or more pointcloud sets by finding the relative position and orientation between views in a global coordinate frame, iteratively aligning the point clouds such that processing stages such as segmentation can be applied. In this paper, registration of two pointclouds is carried out by the classical variant of Iterative Closest Point (ICP).

“Due to the non-convexity of the optimization, ICPbased approaches require initialization with a rough initial transformation in order to increase the chance of ending up with a successful alignment. Good initialization also speeds up their convergence.” [6]

General Notes

* This approach provides a benefit over classical online seam tracking in that it does not require measuring the joint itself.
* It is important to distinguish between ‘assembly variations’ and ‘form variations’
* This process could be extended and used to check for unacceptable workpiece geometries and or relative placement.
* The ability to locate objects in the working environment could also be used to assist the operator in workpiece placement as in [] where a laser projector is used to provide a visual guide for manual placement of the workpiece prior to weldment.
* The group at ‘Fraunhofer’ then uses the transformations to update the reference models…
* Rajaraman Dawson-Haggerty Shimada Bourne use a ‘laser projector’ to help the operator in ‘placement’ of the workpeice. ‘robot guided placement procedure’ + ‘3d sensing for preplanned tool paths’

**Bibliography**

Numbered references are tied to notes, please do not change.

[1] Segal, A. V., Haehnel, D., and Thrun, S., 2010, “Generalized-ICP,” Robotics: Science and Systems.

[4] Zhang, J., Yao, Y., & Deng, B. (2021). Fast and Robust Iterative Closest Point. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[5] Rusu, R. B., and Cousins, S., 2011, “3D Is Here: Point Cloud Library (PCL),” Proceedings - IEEE International Conference on Robotics and Automation.

[6] Zhao, H., Anwer, N., et. Al., 2016, “Registration with the Point Cloud Library PCL,” IEEE Int. Conf. Intell. Robot. Syst.

[7] Du, S., Xu, Y., Wan, T., Hu, H., Zhang, S., Xu, G., and Zhang, X., 2017, “Robust Iterative Closest Point Algorithm Based on Global Reference Point for Rotation Invariant Registration,” PLoS One.

Detection of Assembly Variations for Automatic Program Adaptation in Robotic Welding Systems

Alexander Kuss, Ulrich Schneider, Thomas Dietz Fraunhofer

Automated Planning of Robotic MAG Welding Based on Adaptive Gap Model

Alexander Kuss, Ulrich Schneider, Thomas Dietz Fraunhofer

Generalized-ICP

Aleksandr V. Segal, Dirk Haehnel, Sebastian Thrun

Automated workpiece localization for robotic welding

Rajaraman Dawson-Haggerty Shimada Bourne

A method for registration of 3D shapes

Paul Besl, Neil McKay