Registration and deviation measurement for geometric learning

Scientific background

Measuring how a part differs from its reference requires placing the observed point cloud in a common coordinate system and quantifying the remaining geometric differences. Registration must distinguish pose differences from actual shape deviations. These measurements could also guide neural-network training, but a useful inspection measure is not automatically a useful training loss. Its gradients must support learning while preserving the geometric details needed for inspection.

Research question

How can rigid registration and geometric deviation measurements support reliable comparison with reference geometry, and can a selected approach serve as a loss for point-cloud reconstruction?

Your assignment

- Review point-cloud-to-point-cloud and point-cloud-to-mesh registration and comparison. Select one baseline and one registration alternative, together with complementary surface-distance and feature or dimensional measurements.
- Build a controlled benchmark with known pose changes and shape deviations. Use a bounded set of noise, sampling, or incomplete-coverage conditions to assess whether registration preserves the deviations being measured. Establish scale independently of the rigid fit.
- Select one measurement-based loss and compare it with a conventional reconstruction loss in the same modest network. Keep the architecture and data protocol fixed. Examine differentiability, gradient behavior, training stability, and computational cost.

Evaluation and expected outcome

Measure rotation and translation errors separately from the error in recovered geometric deviations; smaller errors are better. A larger measured deviation means a larger departure from the reference, not necessarily a worse measurement. Check whether improved surface agreement conceals a genuine shape difference, and treat unobserved surfaces as missing evidence rather than automatically as defects.

Distinguish the nominal inspection reference from the ground-truth geometry of the actual part, which includes genuine deviations. Train the network to reproduce that actual geometry or its known deviations, rather than rewarding replacement with an ideal part. Report reconstruction and deviation-measurement errors before and after registration.

Use object-separated training, validation, and test data. Select the model on validation data, then evaluate the frozen model on the full test set and a predefined training subset with identical settings. The outcome is a reproducible registration and measurement comparison, a focused training experiment, and recommendations on suitability and limitations. A loss that proves unsuitable can still provide a useful result through documented failure analysis.

Prerequisites

Python and PyTorch, neural-network training, linear algebra, and basic numerical optimization. Experience with 3D geometry and automatic differentiation is helpful.

Optional extension

Add one complementary measurement term for a specific geometric feature after completing the main comparison.

Contact

If you are interested in this project, please contact Prof. Pascal Cerfontaine (pascal.cerfontaine@th-koeln.de) or Kai Altwicker (kai.altwicker@th-koeln.de).

Learning geometric representations from point clouds

Scientific background

Learning representations of 3D geometry requires preserving shape details that matter for reconstruction and measurement. Independently sampled point clouds can describe the same surface, while a small geometric change can be important for assessing deviations from a reference. A useful representation should distinguish sampling differences from actual differences in geometry.

Research question

How can point-cloud-in/point-cloud-out learning preserve geometry relevant to subsequent deviation measurement?

Your assignment

- Review point-cloud learning and formulate a focused hypothesis about geometry preservation. Select a reproducible baseline and one principal approach by the end of the literature phase.
- Design controlled data containing repeated samples of the same surfaces and known differences between surfaces. Begin with complete surfaces so that missing geometry does not introduce a separate completion problem.
- Implement and compare the selected approaches. No encoder, decoder, or loss formulation is prescribed in advance; justify these choices through the literature review and research hypothesis.

Evaluation and expected outcome

Evaluate reconstruction error, consistency across repeated surface samplings, and preservation of known geometric differences. Lower reconstruction and deviation-measurement errors are better. Smaller representation differences between samples of the same surface indicate greater sampling consistency, but are useful only if different geometries remain distinguishable. Check these properties together rather than treating reconstruction alone as evidence of a useful representation.

Use object-separated training, validation, and test data. Select the model on validation data, then evaluate the frozen model on the full test set and a predefined training subset with identical settings. Explicitly document any prior research use of evaluation objects.

The outcome is a reproducible point-cloud learning study and evidence about which geometric information the selected approach preserves or loses. Image-based training and the complete inspection pipeline are outside the core scope.

Prerequisites

Python and PyTorch, neural-network training, and linear algebra. Interest in geometric learning and controlled experimental work is essential.

Optional extension

Study one additional robustness condition, such as controlled input noise, after completing the main comparison.

Contact

If you are interested in this project, please contact Prof. Pascal Cerfontaine (pascal.cerfontaine@th-koeln.de) or Kai Altwicker (kai.altwicker@th-koeln.de).

From images to reference-based dimensional inspection

Scientific background

An existing, partially developed pipeline uses Depth Anything 3 to reconstruct geometry from images. This thesis will combine the available components and complete the workflow from reconstruction to dimensional measurement against reference geometry. The work connects reconstruction, registration, feature extraction, and measurement, with consistent coordinates, scale, and measurement definitions and an assessment of the resulting errors.

Research question

How reliably can a completed pipeline based on Depth Anything 3 recover predefined geometric parameters of a part for comparison with its reference?

Your assignment

- Review the existing pipeline and its available components, identify missing connections and functionality, and select suitable geometric measurement tools. Define one object family, a fixed reference, and a short measurement recipe specifying the required features and dimensions.
- Integrate the available components and implement the missing steps to complete a reproducible workflow from images through reconstructed geometry to a structured parameter list. Define units, coordinate transforms, and failure reporting at component boundaries. Verify the required automation interfaces early.
- First test the measurement stage on reference geometry and controlled variants. Then introduce reconstructed geometry and investigate a bounded set of acquisition conditions. Reuse established components where suitable.

Evaluation and expected outcome

Compare recovered parameters with known values; smaller absolute parameter errors are better. Report successful and failed measurements, runtime, and the effects of reconstruction, scale calibration, registration, and measurement. Use known controlled changes to check whether the pipeline detects actual part deviations rather than only recovering nominal dimensions.

Establish physical scale from independent calibration information. Do not estimate scale by forcing the inspected part to match its nominal dimensions. Results on normalized CAD objects describe the chosen normalized geometry, not the original part's physical accuracy. Fix the workflow using development objects before assessing independent evaluation objects.

The outcome is an executable prototype, a documented parameter export, and an error analysis identifying usable conditions and limitations. The core uses synthetic data with known geometry. Real captures are optional and require suitable reference measurements; production certification is outside scope.

Prerequisites

Python programming, basic 3D geometry and camera models, and an interest in measurement and software integration. Experience with reconstruction tools is helpful.

Optional extension

Add guided feature selection so a user can adapt the fixed measurement recipe. Automatic discovery of arbitrary features is outside the core scope.

Contact

If you are interested in this project, please contact Prof. Pascal Cerfontaine (pascal.cerfontaine@th-koeln.de) or Kai Altwicker (kai.altwicker@th-koeln.de).