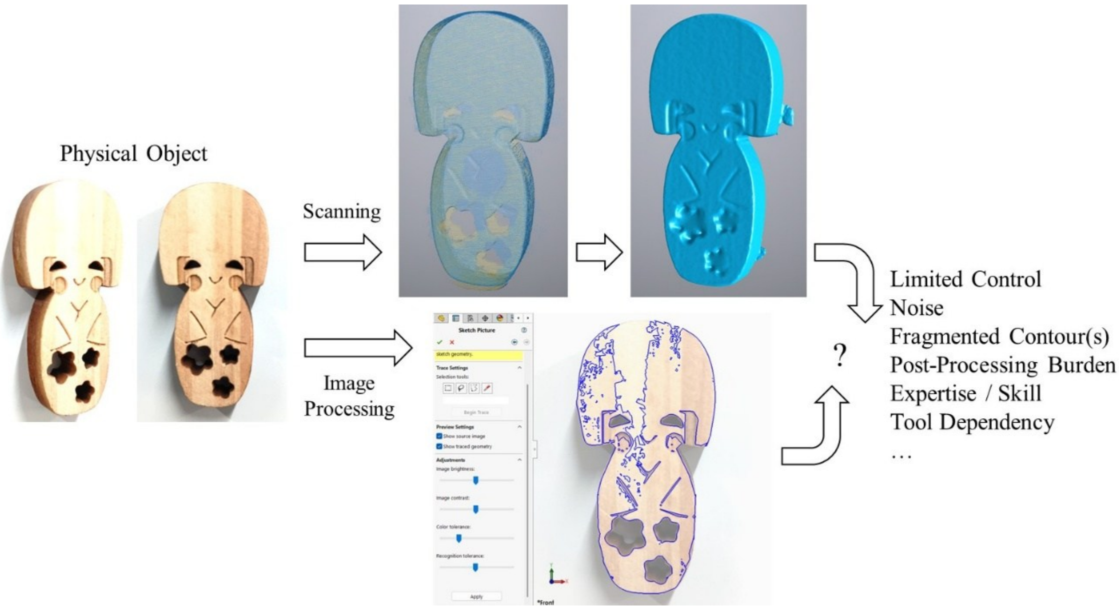
Comparing the workflow with traditional approaches

<u>(Example 2: Reverse engineering)</u>

To illustrate the framework's utility in digitization and reverse engineering, consider the physical object shown below: a Japanese wooden doll with intricate carved features. The objective is to reverse engineer this object into a usable CAD model.

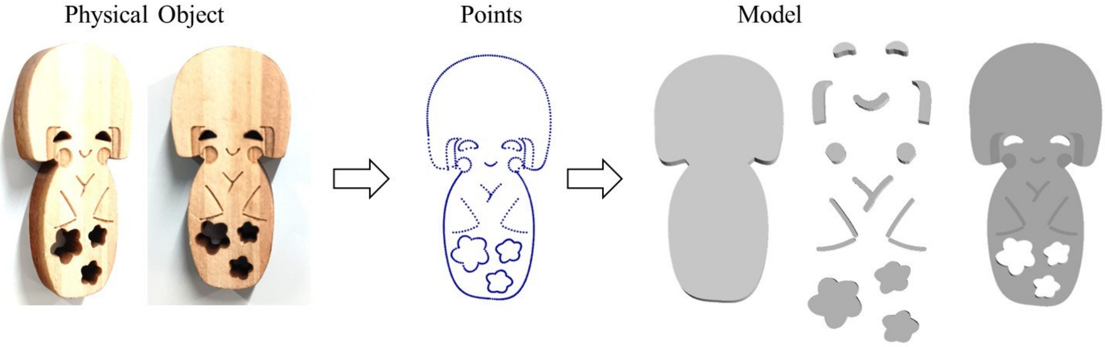
The standard industry approaches for this task are 3D scanning and automated image tracing. The following figure illustrates the results obtained from these methods.



A non-contact structured-light 3D scanner (EinScan-SP, white-LED pattern projection; point spacing 0.17–0.20 mm) was used to digitize the object. As seen in the above figure, scanning the full object yields a dense point cloud of 186,873 points, resulting in a mesh of 498,670 triangles. However, the output exhibits significant distortions, including surface bumps, irregular holefilling, and geometric deviations. These issues necessitate extensive post-processing, which requires high computational resources and specialized expertise to repair the mesh into a usable solid. Even when restricting the scan to a single front plane to reduce data size (18,017 points), the resulting point cloud remains noisy and sparse, making reliable reconstruction difficult.

The above figure also displays the results from an automated tracing algorithm available in a commercial CAD package (SolidWorks®). This method extracts paths by detecting color boundaries based on user-defined tolerances. As seen in the figure, while high-contrast features are detected reasonably well, features with nuanced textures or gradients result in fragmented, overlapping, or jagged contours. As with scanning, the extracted paths require significant manual correction—denoising, smoothing, and gap closure—before they can serve as valid geometric profiles.

In contrast, the proposed workflow provides a direct and clean pathway for reverse engineering. The user imports an image of the object into the IPCM layer and interactively traces the key features to generate structured point clouds. These points are then processed by the rendering script to generate the solid model. (Note: Detailed operational instructions for the IPCM systems and templates are available at: https://commons-repo.github.io/002-research/). The following figure shows the related results.



The comparison highlights distinct advantages of the proposed framework. Both traditional methods (scanning and automated tracing) introduce significant downstream burdens: they generate noisy, unstructured data that requires extensive post-processing, specialized software, and often expensive hardware. In contrast, the proposed workflow allows the user to define clean, noise-free geometry from the outset. This eliminates the need for expensive scanners, proprietary licensing, and complex mesh-repair tools. By removing these financial and technical barriers, the framework offers an accessible and computationally lightweight solution for reverse engineering.