

3D Point Cloud Reconstruction and Focus Stacking for Non-Destructive Inspection of Micro-Machined Features

Author

October 2024

1 Abstract

This paper presents a novel method for generating 3D point clouds from a series of 2D images, leveraging Image Processing techniques for object masking, Laplacian-based focus stacking for selecting in-focus regions, and traditional computer vision techniques for point cloud reconstruction. The proposed method allows for the reconstruction of 3D models using both machine learning and traditional methods to ensure robustness across different image sets. This hybrid approach optimizes the accuracy and completeness of the 3D reconstruction process while ensuring efficiency in memory usage and computational cost.

2 Introduction

The use of 3D point clouds has increased rapidly with the advent of technologies like virtual reality, autonomous navigation, and advanced medical imaging. Point clouds allow us to precisely represent real-world objects, which can then be used for various tasks like 3D modeling and measurements. However, creating 3D point clouds comes at a cost with the best state-of-the-art 3D stereo and depth cameras being too expensive.

In this research, we have used a normal camera to capture 2D images from various focal lengths or depths to create a near-accurate 3D replica of the object.

This work uses image processing and machine learning techniques to improve the robustness of our model. By applying Recurrent Convolutional Neural Network (R-CNN) for object segmentation and masking, along with Laplacian focus stacking for the focus-based region selection, and depth map creation for 3D point cloud generation, we have achieved highly detailed and scalable 3D reconstruction without needing any specialized hardware. The overall process was done considering the need to accurately represent micro-machined objects in 3D point clouds for various inspection and scaling purposes. This research negates the need to use Depth Cameras to inspect any deformity in micro-machined objects.

3 Methodology

The overall methodology consists of four key stages: image acquisition, object segmentation, focus stacking to extract in-focus regions, and 3D point cloud generation.

3.1 Experimental Setup

3.2 Object Masking with Mask R-CNN

In this research, we have used Mask R-CNN (Mask Region Convolutional Neural Network) for object detection and segmentation tasks. As referenced in [?], Mask RCNN can detect objects of interest within images and reproduce masks for the objects. The model used in our research is based on the ResNet-50 backbone with Feature Pyramid Networks, which ensures the ability to detect objects at various scales. In our case, we have used RCNN to detect the in-focus regions of images. In our model, RCNN is specifically used to detect the areas that are in focus and create a mask out of it.

The formulation of Mask-RCNN consists of two components: Region Proposal Network (RPN) [?] and Fully Convolutional Neural Network (FCN) [?]. RPN utilizes Regions of Interest (ROIs) where objects may exist, while the FCN generates a binary mask for each object. Given an input image I , the RPN generates proposals, each defined by bounding box coordinates (x, y, w, h) . The FCN takes these ROIs and generates pixel-wise object masks M :

$$M = FCN(P)$$

The detected objects are then masked from the input image for further processing.

3.3 Focus Stacking: A Multi-Focus Image Fusion Technique

Focus stacking, also referred to as multi-focus image fusion, is a computational photography technique used to enhance the depth of field (DoF) of an image by merging multiple images captured at varying focus levels. In scenarios like micro-machining, where certain features of the object may not be entirely in focus due to the limited DoF of traditional imaging systems, focus stacking provides a way to create a composite image with maximal sharpness across the entire object surface.

This method is particularly valuable in micro-machined feature analysis, where fine details such as edges, small depressions, or protrusions need to be captured with high precision. In focus stacking, multiple 2D images of the same scene are taken at different focal planes, and then the sharpest parts of each image are combined to generate a final image with uniform clarity across the scene. The process can be mathematically formalized as:

$$I_{\text{stacked}}(x, y) = \sum_{n=1}^N W_n(x, y) I_n(x, y)$$

where:

- $I_{\text{stacked}}(x, y)$ is the resulting composite image,
- $I_n(x, y)$ represents the intensity of pixel (x, y) in the n -th image,
- $W_n(x, y)$ is the weight assigned to the pixel (x, y) from the n -th image, which is based on its sharpness.

3.4 Variance of the Laplacian Operator for Sharpness Measurement

The Laplacian operator is a second-order differential operator used to detect edges by measuring the rate of change of pixel intensity. In focus stacking, the variance of the Laplacian is used as a sharpness metric, as it captures the rate of change in pixel intensity, which is indicative of focus. Pixels with higher variance are considered to be more in focus, while those with lower variance are out of focus.

The Laplacian $L(x, y)$ at any pixel position (x, y) is calculated as the sum of the second derivatives of the intensity function $I(x, y)$ with respect to x and y :

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Once the Laplacian is calculated for each pixel, the variance of the Laplacian, $\text{Var}(L)$, provides a measure of how sharply the pixel changes relative to its neighbors:

$$\text{Var}(L) = \frac{1}{N} \sum_{i=1}^N (L(x_i, y_i) - \mu)^2$$

where:

- N is the total number of pixels in a neighborhood window,
- μ is the mean Laplacian value within the window.

Pixels with high variance represent edges and sharp transitions, thus indicating that they are in focus. For each pixel position (x, y) , the Laplacian value is computed as:

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

The variance $\text{Var}(L)$ at each pixel position determines the sharpness:

$$\text{Var}(L) = \frac{1}{N} \sum_{i=1}^N (L(x_i, y_i) - \mu)^2$$

For each set of images, the algorithm selects the pixel with the highest variance in the Laplacian, indicating that this pixel is in focus. This generates a composite "stacked" image where only the sharpest regions of each image are retained.

3.5 Depth Map Creation

Once the stacked image is generated, a depth map is created based on the indices of the selected sharpest pixels. For every selected in-focus region, a corresponding depth value is assigned based on the index of the image from which it was taken. This assumes that different images correspond to different heights or distances from the object.

Let i represent the index of the image, and let z_i represent the distance (or layer depth) of image i from the camera. The depth map D is generated as follows:

$$D(x, y) = i \cdot d$$

where d is the distance between layers (or focus planes).

3.6 Point Cloud Generation

Once the depth map is created, it is combined with the image's pixel data to create a point cloud. Each pixel (x, y) in the image is associated with a 3D point (X, Y, Z) , where X and Y are derived from the pixel coordinates, and Z is taken from the depth map:

$$X = x \cdot s_{xy}, \quad Y = y \cdot s_{xy}, \quad Z = D(x, y) \cdot s_z$$

where s_{xy} and s_z are scaling factors for the X , Y , and Z dimensions, respectively. The corresponding color of each point is taken from the pixel value in the original image.

3.7 Image Processing for Edge Detection

In cases where Mask R-CNN does not produce a valid mask, we employ traditional image processing techniques such as Canny edge detection and morphological transformations to generate a mask. The Canny edge detection algorithm detects strong gradients in the image to find edges, while dilation and morphological closing operations refine the mask.

Canny edge detection is mathematically described as follows:

$$G(x, y) = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$$

where $G(x, y)$ represents the gradient magnitude at position (x, y) .

3.8 Voxel Downsampling for Point Cloud Optimization

To improve performance and memory usage, the generated point cloud is downsampled using voxel grid filtering. The point cloud is divided into a 3D grid, and points falling into the same voxel are averaged into a single point.

The voxel grid filter reduces the number of points while preserving the geometric structure of the object.

3.9 Real-Time Visualization with Open3D

For interactive visualization, Open3D is employed to display the point cloud in real time. After generating the 3D points and color information, the points are passed to Open3D for rendering. Voxel downsampling is applied to optimize memory usage:

$$P_{\text{cloud}} = (X, Y, Z, C)$$

where C represents the RGB color values of each point.

4 Results and Discussion

The proposed method was evaluated on micro-machined components using a standard camera for image acquisition. The experiments focused on validating the dimensional accuracy and point cloud quality.

4.1 Performance Metrics

The accuracy of the point clouds was evaluated using root mean square (RMS) error for dimensional measurements:

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N (D_{\text{measured}} - D_{\text{true}})^2}$$

where D_{measured} and D_{true} represent the measured and true dimensions, respectively.

4.2 Memory and Processing Optimization

In order to optimize memory usage, voxel downsampling was applied to reduce the number of points in the cloud, while retaining the essential geometric details. This step significantly improved performance on low-memory systems.

5 Conclusion

This paper demonstrates an efficient and robust approach to 3D point cloud generation from 2D images, providing a low-cost alternative to traditional 3D scanning hardware. By integrating Mask R-CNN segmentation, focus stacking, and Open3D visualization, we achieved high accuracy in micro-machined feature inspection. The method can be further extended to other domains, such as medical imaging and industrial quality control, where 3D information is crucial but specialized hardware is impractical.