Software Requirements Specification (SRS) Document

3-D Point Cloud Variation Measurement (Five Fingers Innovative Solutions) under Mr. Anil Kumar Upadhyay and Mrs. Chitralekha Upadhyay - Harshika Jain, Shivam Nayak, Sonu Guru, Subodh Sondkar

# Brief Problem Statement

We live in a world of 3 space dimensions, but most of the pictorial information we have, be it photos or videos, are 2-D. We have gotten accustomed to getting information in 2-D format. All automation in the field of processing photos and videos to get useful data has only been for 2-D. There is a need for automation in processing of 3-D photos and videos.

For instance, say you want to calculate the work done in a mine, but there is a geographical constraint that you cannot go to the mine and physically check the progress. If you ask the contractor for proof of progress, she may easily show you the progress from such an angle, that it looks more than it is. When you have a 3-D model of the site, there won’t be any deception because the data we have is in the dimension of the space we live in.

# System Requirements

For object identification, we shall have to use opencv. Opencv is a library in c++ and python. For extraction, removal, boundary detection, and variation measurement, we can use MLX, a python library for Meshlab, or Cloudcompare scripting.

# Users Profile

As 3-D scanners are expensive, and 3-D videos are not easily available, our early adopters will mostly be companies involved in 3-D printing, and research teams involved in 3-D model development and analysis.

Rich organisations who want to track progress of on-field jobs automatically can use the system to track changes in topography of objects (for example- land). Doctors and hospitals can use it to track the properties (for example- size) of deformities on the skin, or about tumours inside the body.

If the system reaches to common people inexpensively, they can also use it to track the deformities on objects, though this is far fetched. For example, we can use to analyse damage to objects based on their deformities.

# Feature Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Use Case Name | Description | Release |
| 1 | Object Identification (Photo) | Given an STL image of an object (without noise) and another STL image (with noise) containing one or more instances of that object, being able to recognise that object with accuracy. | R1 |
| 2 | Point Cloud Extraction (Photo) | Given an STL image of an object and another STL image, being able to extract the point cloud(s) of that object. | R1 |
| 3 | Noise Removal (Photo) | Given an STL image of an object and another STL image, being able to remove noise from the point cloud of the object extracted from it. | R1 |
| 4 | Object Identification (Video) | Given an STL image of an object and an STL video, being able to recognise instances of the object throughout the video. | R1 |
| 5 | Point Cloud Extraction (Video) | Given an STL image of an object and an STL video, being able to extract the point cloud throughout the video. | R1 |
| 6 | Noise Removal (Video) | Given the point cloud of an object extracted from a video, being able to remove noise from the point cloud extracted from the video. | R1 |
| 7 | Variation Boundary Detection | Given the point cloud of an object extracted from a video, being able to detect the variation boundaries. | R2 |
| 8 | Variation Measurement | Given the point cloud of an object extracted from a video, being able to calculate the changes in the properties of the point cloud. | R2 |