Sheet Stitching Problem Definition

Julian Schilliger
Scroll Prize Inc.
julian@scrollprize.org

Abstract

This write-up contains the problem definition of the Sheet Stitching Problem as a graph problem and a possible objective function to optimize for. This is a sub-problem we are trying to solve as a part of Vesuvius Challenge (more information: scrollprize.org).

1 Introduction

The Herculaneum papyri collection consists of a few hundred still unopened and charred scrolls that were buried by the Vesuvius eruption in 79 AD. To read the papyri, they are scanned and a 3D reconstruction is created with computed tomography. Then the scroll is digitally unrolled and the ink made visible with AI. This write-up focuses on a specific step in the ThaumatoAnakalyptor unrolling pipeline. This specific step, called "Sheet Stitching," might be encountered by other, future automatic segmentation pipelines as well.

2 ThaumatoAnakalyptor Pipeline

The pipeline of automatic scroll CT papyrus surface extraction has multiple steps. First, extract a 3D point cloud V (Figure 2) on the papyrus surface from the raw scan (Figure 1). Second, use instance segmentation to identify small patches of points belonging to the same local surface (Figure 3). Next, create a graph from the instance segmentation, connecting overlapping patches (nodes) with connection edges (Figure 4). Since there is noise in the creation of the point clouds and in connecting the patches, the graph erroneously links some patches that actually are not from the same local papyrus sheet. Defining and solving the correct objective over this graph will extract the global sheet connectivity, segmenting the entire scroll. Finally, the segmentation is meshed and virtually unwrapped.

You can find the complete automatic segmentation pipeline ThaumatoAnakalyptor at https://github.com/schillij95/ThaumatoAnakalyptor.

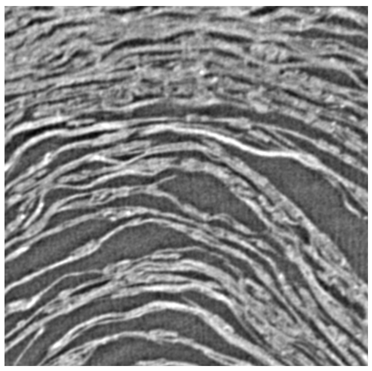
Figure 1: Small slice of CT data

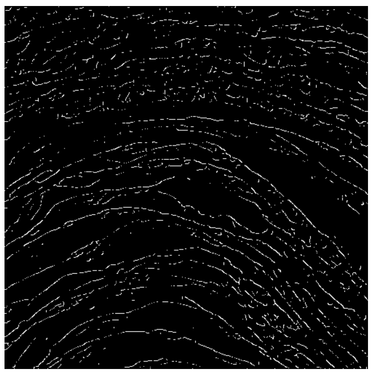
Figure 2: Extracted point cloud

3 Instance Segmentation

Point cloud instance segmentation describes the step of extracting groups of points from the point cloud which belong to the same sheet. The scroll volume V of size X, Y, Z is divided into overlapping subvolumes S of size 50, 50, 50. For $s_{xyz} \in S$, $s_{xyz} \subseteq V$ the top-right-front corner

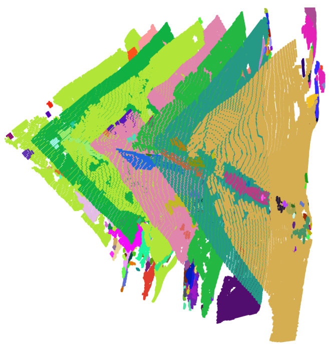

Figure 3: Predicted Instances in a subvolume

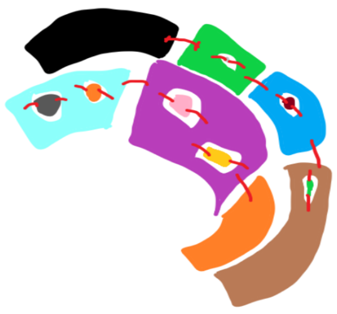

Figure 4: Sheet Stitching

is at position (x, y, z) in the scroll volume.

$$\forall s_{xyz} \in S, \quad \forall i \in \{x, y, z\}, \quad i \in \mathbb{Z}^{+} \quad \wedge$$
$$i \mod 25 \equiv 0 \quad \wedge \quad x \leq X, y \leq Y, z \leq Z$$

The instance segmentation is performed on each subvolume. For all $s_{xyz} \in S$ containing the point cloud PC_{xyz}, the instance segmentation model Mask3D creates instances set I_{xyz}. $\forall i \in I_{xyz}$: $i \subseteq PC_{xyz}$ and $\forall i, j \in I_{xyz}, i \neq j : i \cap j = \emptyset$.

For an average scroll volume V, this process creates around 10 million instances.

4 Translation to Graph

Let's number the instance patches by integer p:

$$I_{xyz} = \{i_p \mid p = 0, 1, 2, \ldots, n-1\}$$

All nodes can now be defined by:

$$N = \{(x, y, z, p) \mid s_{xyz} \in S,\, p \in I_{xyz}\}$$

Edges connect nodes from neighboring subvolumes:

$$E = \{((x_0, y_0, z_0, p_0), (x_1, y_1, z_1, p_1)) \mid$$
$$|x_0 - x_1|, |y_0 - y_1|, |z_0 - z_1| \leq 25\}$$

Let's define the directed, weighted, winding angle graph as $G = (N, E, w, k)$.

We define the winding angle difference function $k : E \rightarrow \mathbb{R}$ as an estimation for the angular rotation that results of traversing the papyrus surface from the first to the second node of the edge with respect to a center axis (called umbilicus).

The estimation of the winding angle difference k is accurate up to a multiple $o \in \mathbb{Z}$ of 360 to the actual winding angle difference k^{*}, since the position of the instance with respect to the umbilicus is known:

$$k^{*}(e) = k(e) + o \cdot 360$$

k is anti-symmetric:

$$k((n_0, n_1)) = -k((n_1, n_0))$$

The weight function $w : E \rightarrow \mathbb{R}^{+}$ estimates if the assignment $k(e)$ is true.

w is symmetric:

$$w((n_0, n_1)) = w((n_1, n_0))$$

The resulting graph will have edges with conflicting information about the winding angle difference within the graph.

5 Sheet Stitching

We seek to find a winding angle assignment function $f : N \to \mathbb{R}$, which assigns each node a winding angle.

Subsection 5.1 contains an example of what type of score function could be used to track your solutions performance. So far this is an open area of research.

Another method to verify a good solution is inspecting the shape of the winding angle histogram of your solution (Subsection 5.2).

5.1 Desired Graph Node Assignment Characteristics

A good measure of the accuracy of the found solution $\tilde{f}$ could be to optimize for:

$$f^* = \arg\max_{f \in F} \sum_{e \in E} w(e) \cdot c(e)$$

where

$$c((n_0, n_1)) = \begin{cases} 1, & \text{if } f(n_1) - f(n_0) = k((n_0, n_1)) \\ 0, & \text{otherwise} \end{cases}$$

Which translates to the least number of wrongly estimated edge winding angle differences.

5.2 Winding Angle Histogram

The Min and Max values in Figure 5 are the f^* minimum and maximum winding angle in degrees to a possible solution of the example graph. The histogram should be in a form close to a triangle. This is because the innermost wraps contain less surface area (and therefore nodes) than the outermost wraps.

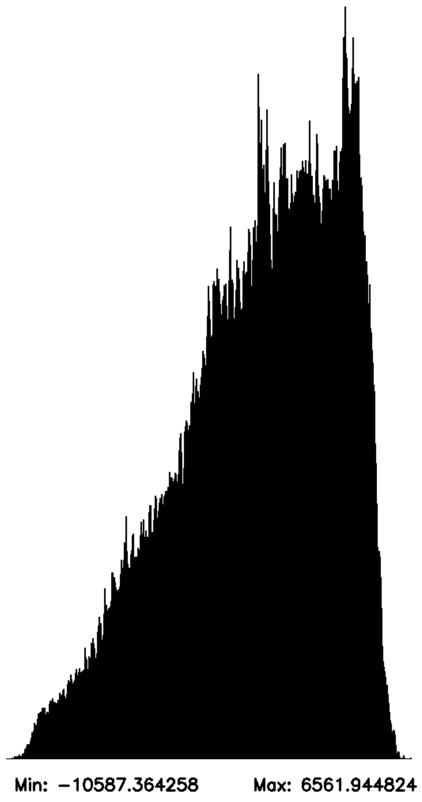

Figure 5: Histogram of Solution.