High-Level Sprout Geometry Extraction for In Vitro Angiogenesis Assays

Gio Borje
University of California, Irvine
gborje@uci.edu

Craig Steinke
University of California, Irvine
csteinke@uci.edu

ABSTRACT

We have developed an automated image analysis system for the quantitative analysis of *in vitro* angiogenesis assays. Specifically, the system is designed for fibrin gel bead sprouting assays. The quantification system provides the number of primary sprouts, average branching factor and average length for each bead in an imaged assay.

1. INTRODUCTION

Angiogenesis is a mechanism for the formation of new blood vessels from pre-existing vessels. Additionally, angiogenesis is part of a critical phase in of solid tumor growth and metastasis. For solid tumors, the avascular growth phase enables an approximate maximum size of 1–2mm in diameter; on the other hand, the vascular growth phase enables unyielding tumor expansion and metastasis [1]. Subsequently, *in vitro* angiogenesis assays are used to assess the impact of pro-angiogenic and antiangiogenic agents.

The fibrin bead assay, developed by the Hughes Lab, begins by culturing endothelial cells as a monolayer on dextran-coated Cytodex beads. The beads are embedded into fibrin gels. In particular, human umbilical vein endothelial cells (HUVEC) are used. This method induces HUVEC to recapitulate multicellular capillaries in fibrin gels. Furthermore, the method for HUVEC promotes sprouting, lumen formation and long-term stability of neovessels. The high-resolution images of beads are then captured on an IX70 Olympus microscope [2]. A sample of such an image can be seen on Figure 1.

Previously, the number of sprout counts per bead were measured by manual counting on the obtained images. Furthermore, the sprout length was measured in arbitrary units [2]. Our system relieves the laborious, manual analysis by quickly automating the detection and reliably analyzing of imaged assay features.

The system, the High-Level Sprout Geometry (HLSG) Extractor, is designed to detect features, restore features and analyze the features of imaged assays. Features for detection include the Cytodex bead and its associated multicellular capillaries. However, due to the noise and depth of the image, the HLSG Extractor uses structural inpainting to restore and broken sprout features.

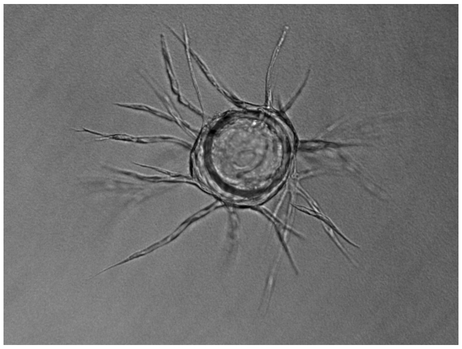


Figure 1: Single Bead Image

In addition to the High-Level Sprout Geometry (HLSG) Extractor, a driver and report generator are implemented to drive functionality on sample images and generate reports on the analyses respectively.

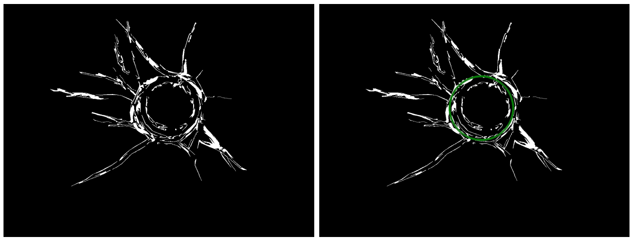
The following sections will proceed as follows. Section 2 will describe an overview of the HLSG feature detection, feature restoration and analysis mechanisms. Section 3 will describe the data structures used to represent the HLSG and its properties. Section 4 shows how the system is designed for modularity and robustness.

2. METHODOLOGY

Our system enables feature set detection, minor feature restoration and quantitative analysis which can be decomposed into four stages. The first two stages detect feature sets: beads as features and then sprouts as features. In the third stage, the system attempts to restore a few sprout features by approximating and inpainting connections between broken sprout segments as well as filling holes. Finally, the system quantitatively analyzes the imaged assays through Sholl Analysis.

2.1 Bead Extraction

Bead extraction is a two-step process. To reduce noise, the system first smooths the image using a Gaussian blur. Second, circles in the image are detected using the circular Hough Transform. The circles detected correspond to the beads in the assay. Subsequently, the origin and radius of



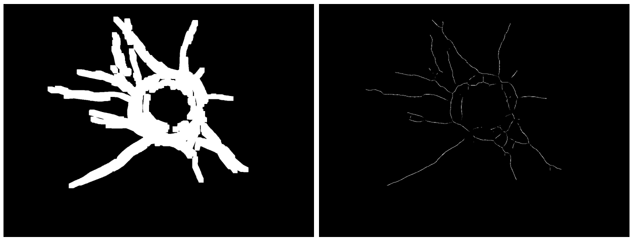
(a) Edges (b) Bead Detection

Figure 2: Bead Extraction Outline

the bead is obtained.

In order to avoid false bead detections, we enforce that the minimum distance between the origin of every pair of detected circles be four times the average bead radius i.e. each detected bead should at least be a bead apart.

2.2 Sprout Extraction



(a) Dilated (b) Skeleton

Figure 3: Sprout Extraction Outline

Sprout extraction depends on bead extraction because the beads must be masked before sprout extraction occurs to separate beads from sprouts. We mask the beads given the geometry of the circles. Due to the disconnectivity of sprouts in the assay, we begin by obtaining all sprout segments and represent them as line segments. We distinguish the end points of each line segment for connectivity. The start point, S, is the end point on the line segment such that it is closer to the origin of the line segment's closest bead. The end point, E, is the end point on the line segment such that it is farther from the origin of the line segment's closest bead.

To determine which line segments belong to the same sprout, we use Euclidean distance between their start and end points enforced by the constraint that the end point must by closer to the origin of its closest's bead than the target start point. For example, two line segments are part of the same sprout if the distance between the start an end points are within a specified distance parameter, d.

2.3 Sprout Restoration

There are two primary sources of degradation that we have discovered in the imaged assays: disjoint sprout segments and false capillary lumen detections i.e. holes. We will discuss our approaches to both problems independently.

The disjoint sprout segments are caused by the three-dimensional nature of the sprouts moving in and out of focus of the microscope.

False capillary lumen detections are artifacts of the Canny edge-detection algorithm. By the morphology of these structures, we simply find the contours which enclose small holes and perform inpainting by polygon approximation using the Ramer-Douglas-Peucker algorithm.

2.4 Sholl Analysis

Sholl Analysis is a quantitative method for quantitatively analyzing morphological characteristics of neurons [4]. Briefly, Sholl Analysis consists of counting the number of foreground to background crossings given concentric circles of a parameterized radius.

Algorithm 1 Sholl Analysis

```
procedure SHOLL ANALYSIS(image, origin, radius)
    minRadius ← radius
    maxRadius ← min {image.width - origin.x, image.height - origin.y}
    maxRadius ← min {maxRadius, origin.x, origin.y}
    crossings ← []
    for all r ∈ [minRadius, maxRadius] do
        crossings[r] ← 0
        for all pixel ∈ ConcentricCircle(origin, radius) do
            if crossing detected at pixel then
                crossings[r] ← crossings[r] + 1
            end if
        end for
    end for
    return crossings
end procedure
```

3. DATA STRUCTURES

The following data structures are used to implement the HLSG Extractor.

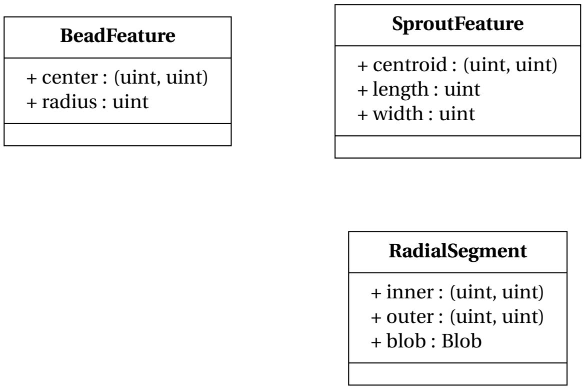


Figure 4: HLSG Extractor Features Class Diagram

3.1 Bead Feature

A bead feature is an abstraction of the Cytodex bead coated with endothelial cells in the assay. The geometry of the bead is intuitively circular; subsequently the geometry can is described by the descriptor in Figure 4.

3.2 Sprout Feature

A sprout feature is an abstraction of the blood vessels that develop through angiogenesis from the designed bead. Subsequently, sprout feature extraction is dependent upon bead descriptors. The sprout is actually comprised of a set of pixel segments because of the possibility that a sprout is disconnected.

3.3 Radial Line Segment

Due to the disconnectivity of sprouts, individual sprout segments are represented by a radially defined line segment. That is, we distinguish the end points from its radial distance from the origin of its corresponding bead. Given a line segment, we say that and end point is the *inner point* if it is radially closer than its complementary end point; otherwise, we call the end point the *outer point*.

In addition to the distinguishable end points, a radial line segment is a line fit onto a corresponding blob of pixels which can be considered a sprout segment.

3.4 Driver

The Driver is responsible for parsing input from the user and emulating the encoded actions as functions of the HLSG Extractor. That is, the Driver acts similar to a REPL (Read-Eval-Print-Loop) that reads input from the user, evaluates the input and prints the corresponding output in a loop. The set of commands available to the user is outlined Table 1.

4. SYSTEM ARCHITECTURE

The system requires Python version 2.7x with the SimpleCV package. The architecture of the system is based on our methodology for quantitatively analyzing *in vitro* angiogenesis. The system, however, incorporates modules for driving batch processes as well as a Read-Eval-Print-Loop (REPL) for console interaction. Finally, a module incorporated for generating CSV reports of the analysis. The sequence diagram for the system components are shown in Figure 5.

The REPL module controls the interaction between the user and the system. Commands available in the REPL are shown in Table 1.

5. RESULTS

Display a comparison table with human counts.

6. DISCUSSION

Although the methodology is similar to AngioQuant, our method operates on two-ridge structures [3].

Command	Output	Description
extract [file]	HLSG of file	Extracts the HLSG of the given file.
extract [files]	HLSG of files	Extacts the HLSGs of the given files.
exit	Goodbye	Exits the system.

Table 1: Commands

Human Sprout Counts	HLSG Sprout Counts
0	0
0	0
0	0
0	0

Table 2: Result Comparison

7. REFERENCES

[1] Robert S. Kerbel. Tumor angiogenesis: past, present and the near future. *Carcinogenesis*, 21(3):505–515, 1999.

[2] Martin N. Nakatsu, Richard C.A. Sainson, Jason N. Aoto, Kevin L. Taylor, Mark Aitkenhead, Sofía Pérez del Pulgar, Philip M. Carpenter, and Christopher C.W. Hughes. Angiogenic sprouting and capillary lumen formation modeled by human umbilical vein endothelial cells (huvec) in fibrin gels: the role of fibroblasts and angiopoietin-1. *Microvascular Research*, 66(2):102 – 112, 2003.

[3] A. Niemisto, V. Dunmire, O. Yli-Harja, Wei Zhang, and I. Shmulevich. Robust quantification of in vitro angiogenesis through image analysis. *Medical Imaging, IEEE Transactions on*, 24(4):549–553, 2005.

[4] D. A. Sholl. Dendritic organization in the neurons of the visual and motor cortices of the cat. *J Anat.*, 87(4):387–406, 1953.

APPENDIX

A. PSEUDO CODE

This section outlines the pseudo-code for the Driver and HLSGExtractor operations.

A.1 Driver

The following pseudo-code outlines the Driver which reads input from the user, evaluates the input as a command, prints the output as a consequence of executing the command and then repeats this sequence of operations.

Note that the driver executes while the running flag is true. Consequently, the REPL is responsible for setting this flag false.

A.1.1 Sprout Extractor

Given an imaged assay and a set of bead features, the algorithm proceeds by masking the beads from the image.

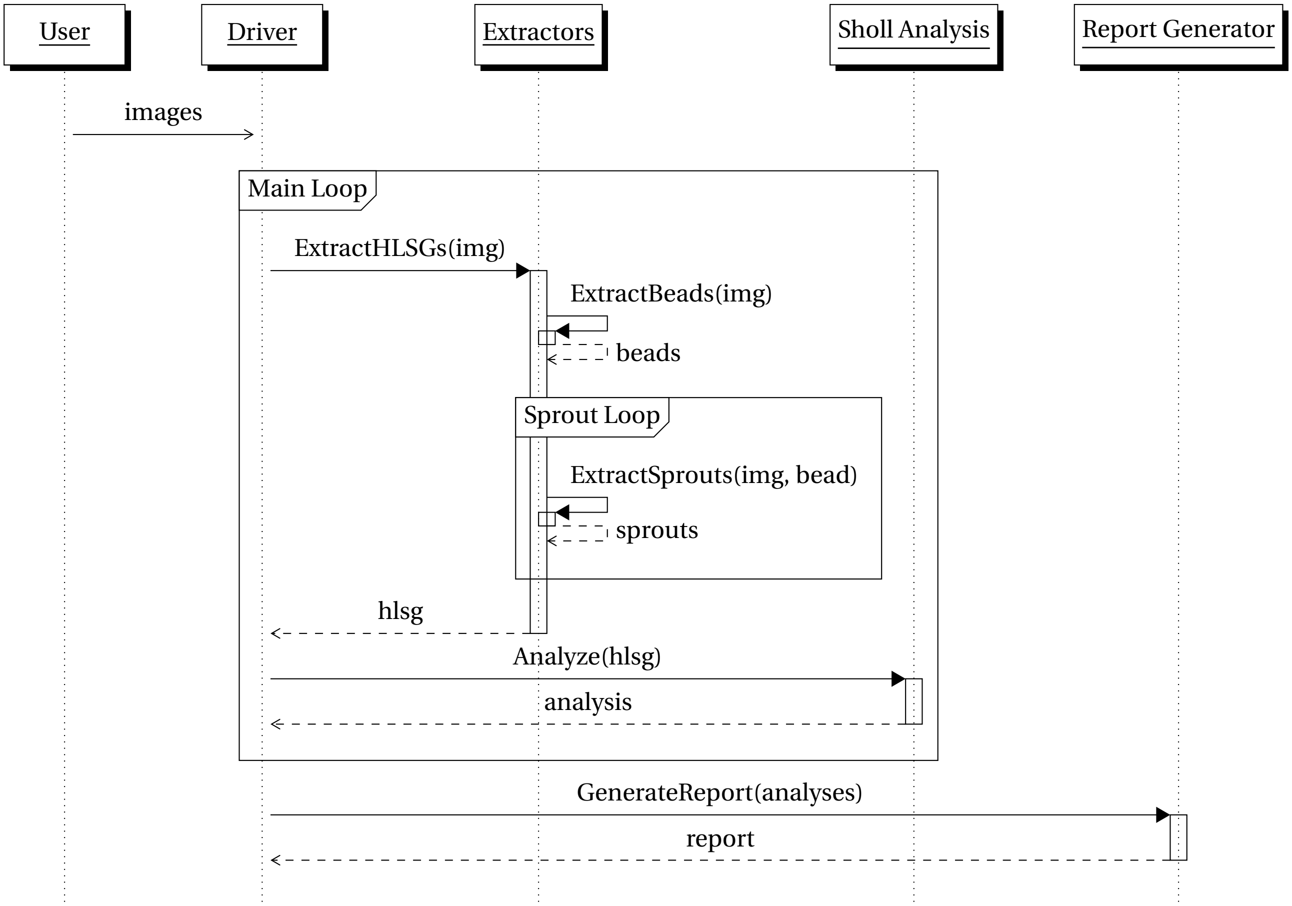


Figure 5: High-Level Architecture

Algorithm 2 Driver

```
procedure DRIVER
    running ← True
    while running do
        input ← read_input()
        command ← parse(input)
        output ← HLSGExtractor.execute(command)
        print(output)
    end while
end procedure
```

Next, a segmentation strategy is used to separate individual sprouts from the collection of globally detected sprouts. Finally, the segmentation strategy yields the detected feature set of sprouts.

A.1.2 HLSG Extractor

Algorithm 3 Sprout Extraction

```
procedure EXTRACTSPROUTS(img, beads)
    maskedImg ← maskBeads(img, beads)
    strategy ← SegmentStrategy(maskedImg,beads)
    sprouts ← strategy.segment()
    return sprouts
end procedure
```

Algorithm 4 HLSG Extraction

```
procedure EXTRACTHLSGS(img)
    beads ← ExtractBeads(img)
    sprouts ← ExtractSprouts(img, beads)
    hlsgs ← MapSproutsToBeads(sprouts, beads)
    return hlsgs
end procedure
```