

3D Histopathology

¹Arjun Verma : IMT2017008

9th Semester Project

Project Instructor : Prof. T.K Srikanth

Project Mentor : Amit Tomar

Lighting in 3D Histopathology

Abstract. *This technical report contains a brief overview of the methodology involved in implementing a lighting mechanism for realistic 3D rendering of H&E stained slides used in histopathology. The code for the project can be found [here](#).*

1. Introduction

3D imaging is an advanced technique in the area of medical imaging. The technique has witnessed great popularity and adoption due to multiple benefits offered such as faster and accurate diagnosis. Cardiology and dentistry were two of the few fields where the technique first found its use and drove the initial growth in the market for more of such techniques. Technological advancements, over the years, such as integration of medical systems with PACS and telesurgery propelled the demand for 3D imaging among surgeons and radiologists [Bureau 2013]. This resulted in the birth of techniques such as 3D MRI and CT scans which played a huge role in the boom in popularity of 3D imaging techniques. Development of 3D imaging techniques that help in completely virtual, slides-free image visualization is also expected to drive the popularity of 3D medical imaging further in the coming years.

Another domain that we believe could be largely benefited by 3D imaging techniques is of histopathology. 3D reconstruction and examination of tissue at microscopic resolution have significant potential to enhance the study of both normal and disease processes, particularly those involving structural changes or those in which the spatial relationship of disease features is important [Roberts et al. 2012]. There have been other methods for studying tissue in 3D that have been explored but they don't provide the same benefits as those of conventional histopathological features such as staining and interpretation techniques. Because of the technical difficulties in constructing 3D tissue models, such techniques had not found their way into routine research. However, with significant advances made in the past decade, this field has now started to see light with different works exploring these techniques [Treanor 2012], [Jansen et al. 2018], [Magee et al. 2015].

2. Methodology

In this section, we discuss the distinct characteristics of the approaches involved in implementing the lighting aspect into the application. We use stacked levels of *.tiff* files as our dataset for the application. The crux of the idea implemented utilizes the concept of *multi-material transfer functions* borrowed from the work of [Igouchkine et al. 2017].

Multi-Material Transfer Functions : To understand this notion of transfer function, let us first introduce the terminology used by the authors of [Igouchkine et al. 2017]. $\vec{\rho}$ is a vector of volume samples from $\vec{\rho} = f(\vec{x})$ where \vec{x} is a position in space. \vec{L} is a set of rendering parameters, m refers to a material and $T(\vec{\rho})$ is a transfer function. The subscripts s and e refer to the values of start and end points of a ray. Generally, when a transfer function is implemented, the mapping produced is of the form $T(\vec{\rho}) = \vec{L}$, i.e, the volume values at the end of the ray are directly mapped to some rendering parameters. The authors instead suggest to look at transfer functions as a composite of two parts, a material transfer function T_{mat} and an optical transfer function T_{opt} . The material transfer function (T_{mat}) is used to map the data directly to corresponding materials by assigning some ID. T_{opt} on the other hand serves as a mapping from materials to rendering parameters. The authors further state that one can assume that all data points mapped to a single ID have homogeneous properties.

The two-part composite mappings can be briefly summarized and represented by,

$$T_{mat} : \vec{\rho} \rightarrow m \quad (1)$$

$$T_{opt} : m \rightarrow \vec{L} \quad (2)$$

Such a technique is not only beneficial for *within material* rendering but also helps in better *material boundary* rendering. Conventionally, each ray sample only focuses on the material which it currently is in. This however fails to capture the unique interactions taking place at the interfaces between materials. Optical effects such as reflection, transmission and diffusion are more prominent along material boundaries and are completely missed out on by the traditional rendering techniques. The newly defined two part definition of transfer function helps us to capture this phenomena by simply extending T_{opt} as,

$$T_{opt} : (m_s, m_e) \rightarrow \vec{L} \quad (3)$$

i.e, the mapping is now from a pair of materials at the start and end points to the assigned optical properties.

Having the general idea in place, let us now continue with the methodology. Implementing lighting using the above technique is a four step procedure : *Segmenting Image*, *Material Mapping*, *Finding Material Boundaries* and *Lighting Model*

2.1. Segmenting the Image

We first begin with the process of detection and classification of structures in the *.tiff* file. We use the popular bio-image analysis tool, **QuPath** [Bankhead et al. 2017] for the same. We begin by loading in the Level 0 of the *.tiff* file. We then set the appropriate properties for an H&E stained slide (set image type as *Brightfield (H&E)*) and encapsulate the entire image using the rectangular select tool. After doing this, you are now ready to use the *cell detection* feature from the *Analyze* menu. After running *cell detection*, use the *object data* dropdown in the *File* menu to export your detected features as a *geojson* file. Choose *export all objects* and click on all checkboxes before exporting. This will result in a

geojson file being downloaded into your system which contains information about the encapsulating polygons of the features detected. An overview of the step-by-step process has been shown in 1.

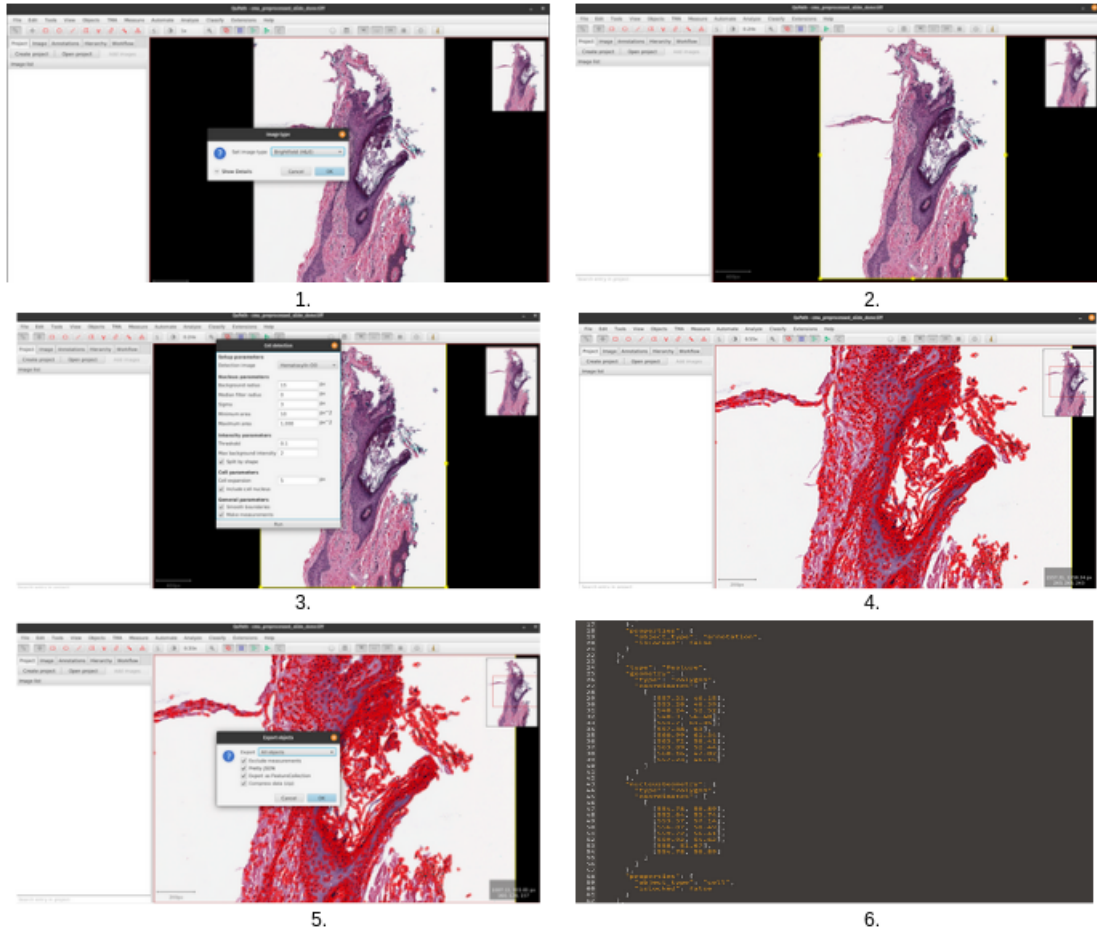


Figure 1. *Step 1* involves setting up the image properties. *Step 2* shows the yellow boundary of the rectangular select. *Step 3* shows the cell detection feature from the Analyze drop down. *Step 4* displays the output of the cell detection feature. *Step 5* shows the export process. *Step 6* shows the final polygonal coordinates obtained in the geojson.

2.2. Material Mapping

We now begin our procedure of material mapping. We do this by utilizing matplotlib's *polygon* and *path* objects. *Polygon's contains_points* feature helps us find out if a particular set of points are within a polygon or not. We use this to make two grids of size equal to the image's pixel space. One grid (pixel length \times pixel width matrix) represents all the points within all nuclei boundaries (*NucleusMask*). The other grid represents all the points within all cytoplasm boundaries (*CytoplasmMask*). In a grid, if a point is within a polygon boundary it is marked *True* else *False*. Ideally, one could now iterate through the matrices in an $\mathcal{O}(n^2)$ manner to assign IDs' to each point in the pixel space (which is actually $\mathcal{O}(n^3)$ due to an outer reading loop.) This procedure is however

very time-consuming as even the small dataset that we are working with contains around 6000 polygons. This number would only increase with more refined datasets. Hence, to improve upon this, another strategy was utilized. We maintain three separate grids (all initialized to *False*) for the three separate features (*NucleusGrid*, *CytoplasmGrid*, *OtherGrid*). To identify, what ID a grid point in the pixel space would have, we perform the following operations,

$$NucleusGrid = NucleusGrid \oplus NucleusMask \quad (4)$$

$$CytoplasmGrid = CytoplasmGrid \oplus NucleusMask \oplus CytoplasmMask \quad (5)$$

$$OtherGrid = OtherGrid \oplus CytoplasmMask \quad (6)$$

Following this, if the points in *NucleusGrid*, *CytoplasmGrid* are *True*, they are marked as a **Nucleus** (ID : 253) or **Cytoplasm** (ID : 254) respectively. When the points in *OtherGrid* are *False*, they are marked as **Other** (ID : 255). A simple construction of the XOR Truth Table (shown in 2) would reveal why the operations have been performed the way they have been performed. This new optimized procedure is an $\mathcal{O}(1)$ procedure ($\mathcal{O}(n)$ considering the outer loop). The *.tiff* file is read in using the PIL library and the assigned material IDs' are written into the opacity channel at each pixel. The material IDs' are selected as 253, 254 and 255 to overcome OpenGL's certain in-built normalization constraints.

Nucleus Grid	Nucleus Mask	Final Value
F	T	T (needed)
F	F	F

For Nucleus

Other Grid	Cytoplasm Mask	Final Value
F	T	T
F	F	F (needed)

For Other

Cytoplasm Grid	Nucleus Mask	Cytoplasm Mask	Final Value
F	F	T	T (needed)
F	T	F	T (never)
F	F	F	F
F	T	T	F

For Cytoplasm

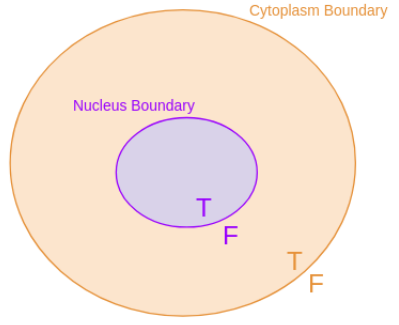


Figure 2. All the points within a polygonal boundary are marked True and outside it False. The construction of the XOR truth tables for each of the possible materials has been shown.

2.3. Finding Material Boundaries

With the above two procedures ready, our next step will take place in the rendering pipeline. An overview of our per-ray segment operations are shown in 3. During the ray casting procedure, at each step length we now check for the material IDs' of the starting and ending position of the ray segment. If both the endpoints have the same material ID, we treat it as a homogeneous component and assign appropriate optical properties for color composition. However, when the endpoints show different segment IDs', this means that the ray is now transitioning within materials and we need to assign different optical properties for these material boundaries. One of the key challenges here is to figure out where does the boundary lie between materials. We use the technique employed by [Hadwiger et al. 2005] to obtain these boundary points. Once a ray segment containing an intersection has been detected (the endpoints have different material IDs'), the next stage determines an accurate intersection position using an iterative bisection procedure. In one iteration, we first compute an approximate intersection position assuming a linear field within the segment. Given the sample values $\vec{\rho}$ at positions \vec{x} for the near and far ends of the segment, the new sample position is,

$$x_{new} = (x_{far} - x_{near}) \cdot \frac{m_{iso} - \rho_{near}}{\rho_{far} - \rho_{near}} + x_{near} \quad (7)$$

where m_{iso} is the isovalue for material boundary taken to be the average of the two material IDs' in our case. After computing x_{new} , the value f_{new} is fetched at this point and compared to the isovalue m_{iso} . Depending on the result, we update the ray segment with either the front or the back sub-segment. If the new point lies in front of the isosurface (e.g. $f_{new} > m_{iso}$), we set x_{near} to x_{new} , otherwise we set x_{far} to x_{new} and repeat. [Hadwiger et al. 2005] have found empirically that a fixed number of *four* iteration steps is enough for high-quality intersection positions.

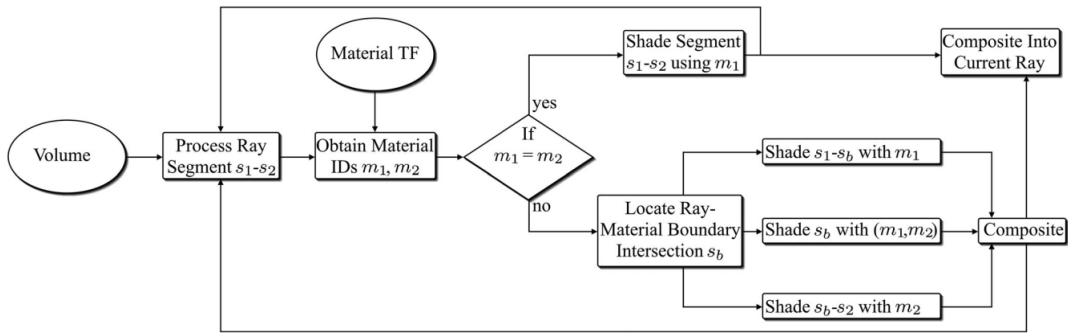


Figure 3. An overview of our ray step iterations. The method employed by [Hadwiger et al. 2005] helps us in locating the ray-material boundary intersections.

2.4. Lighting Model

After obtaining the boundary intersection points (if any), we are now ready for the composition step in our ray-caster. In each ray step, we first add the composition for the starting point of the ray segment followed by the composition for the intersecting point (if any). The color to be composed is obtained using a Blinn-Phong shading model. The equations utilized in the computation are a slight variation of the equations found in [Schulze 2018]. For the process of normal computation required in the shading technique, we make use of the material IDs' assigned to get an estimate of the gradient. The different optical properties mapped to the different material IDs' are given in 4.

Material	Ambient Coeff	Diffuse Coeff	Specular Coeff	Exponent
Nucleus	0.1	0.1	1.0	5
Nucleus - Cytoplasm	0.1	0.35	0.85	5
Cytoplasm	0.1	0.6	0.7	5
Cytoplasm - Other	0.1	0.65	0.35	3
Other	0.1	0.7	0.0	1

Figure 4. The final coefficients have been chosen for each of the materials after extensive experimentation. The Nucleus is set to depict a metal like object, the cytoplasm as a plasticity object and the surrounding areas as a rough diffusive surface.

3. Output and Additional Features

3.1. Final Rendering

To render the final output, first run the application. Then load the volume and click on the best resolution button. Move over to the Lighting tab next and select *Enable Lighting*. The effects of lighting can be seen in 5

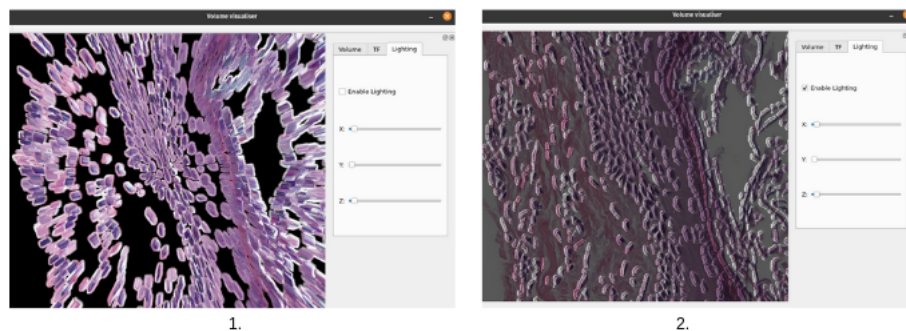


Figure 5. The figure on the left shows the slide without the lighting and the figure on the right shows the effect of lighting

3.2. Lighting Position and Material Opacities

One can also play around with the position of the light using the sliders in the Lighting tab. A user could also play around with the opacities for each of the material to get better visual effects. Setting the opacity for Nucleus as 1, Cytoplasm as 0.4 and Other as 0 gives us a better visualization with specular highlights within the cytoplasm. Refer to [6](#) to see the same.

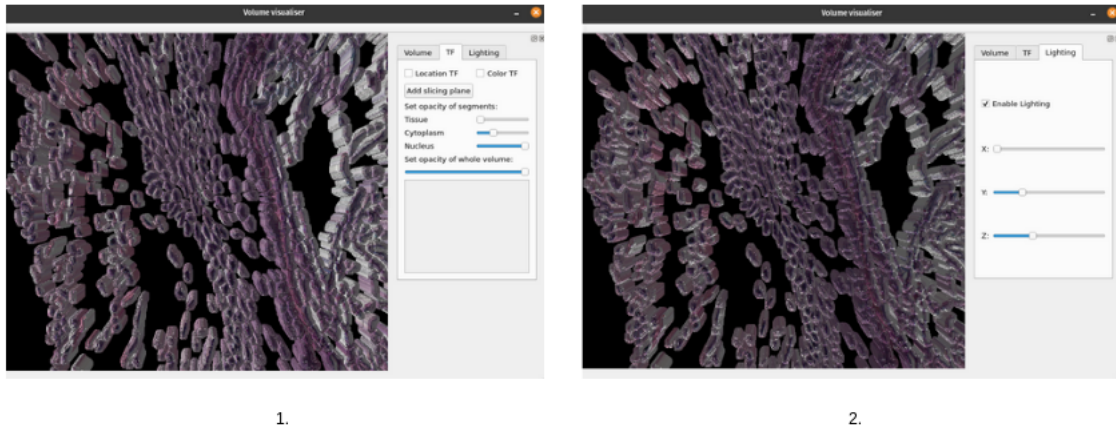


Figure 6. The figure on the left shows the effect of different opacities while the figure on the right shows a change in light position.

References

- [Bankhead et al. 2017] Bankhead, P., Loughrey, M., Fernandez, J., Dombrowski, Y., Mcart, D., Dunne, P., Mcquaid, S., Gray, R., Murray, L., Coleman, H., James, J., Salto-Tellez, M., and Hamilton, P. (2017). Qupath: Open source software for digital pathology image analysis. *Scientific Reports*, 7.
- [Bureau 2013] Bureau, B. (2013). <https://www.biospectrumasia.com/news/53/5582/tech-drive-demand-for-3d-medical-imaging.html>. Biospectrum Bureau, news edition.
- [Hadwiger et al. 2005] Hadwiger, M., Sigg, C., Scharsach, H., Bühler, K., and Gross, M. (2005). Real-time ray-casting and advanced shading of discrete isosurfaces. *Comput. Graph. Forum*, 24:303–312.
- [Igouchkine et al. 2017] Igouchkine, O., Zhang, Y., and Ma, K.-L. (2017). Multi-material volume rendering with a physically-based surface reflection model. *IEEE Transactions on Visualization and Computer Graphics*, PP:1–1.
- [Jansen et al. 2018] Jansen, I., Lucas, M., Savci-Heijink, C., Meijer, S., Marquering, H., Bruin, D., and Zondervan, P. (2018). Histopathology: ditch the slides, because digital and 3d are on show. *World Journal of Urology*, 36.
- [Magee et al. 2015] Magee, D., Song, Y., Gilbert, S., Roberts, N., Wijayathunga, N., Wilcox, R., Bulpitt, A., and Treanor, D. (2015). Histopathology in 3D: From three-dimensional reconstruction to multi-stain and multi-modal analysis. *Journal of Pathology Informatics*, 6(1):6.
- [Roberts et al. 2012] Roberts, N., Magee, D., Song, Y., Brabazon, K., Shires, M., Crellin, D., Orsi, N. M., Quirke, R., Quirke, P., and Treanor, D. (2012). Toward routine use of 3d histopathology as a research tool. *The American Journal of Pathology*, 180(5):1835–1842.
- [Schulze 2018] Schulze, J. (2018). <https://cseweb.ucsd.edu/classes/wi18/cse167-a/lec5.pdf>. CSE 167, slides edition.
- [Treanor 2012] Treanor, D. (2012). <https://www.virtualpathology.leeds.ac.uk/research/3d/>. University of Leeds, news edition.