

High Throughput Segmentation Techniques for Cancer Prognosis

First Progress Report

Submitted in partial fulfillment of the requirements
for the degree of

Ph.D.

by

Andrew Janowczyk

Roll No: 08405201

under the guidance of

Prof. Sharat Chandran



Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Mumbai
October 2009

1 Abstract

Presented is the first stage of the Annual progress seminar. It begins with completed work, and then proceeds to discuss the next phase of research. The stage consists of a general, robust, and efficient unsupervised segmentation algorithm, termed Hierarchical Normalized Cuts (HNCut), and demonstrates its application in precisely quantifying the presence and extent of a stained tissue marker on tissue microarrays. The high throughput of HNCut is derived from the use of a hierarchically represented data structure that bridges a novel frequency weighted mean shift (FWMS) with the normalized cuts algorithm (NCut), guided by domain knowledge. This allows HNCuts to effectively traverse a hierarchical pyramid of a very large input image at various color resolutions, efficiently and accurately segmenting the object class of interest by having the user simply annotate a few pixels belonging to the target class. Not only does the algorithm work on an individual disc in under 6 seconds, the scalability is shown to be quite extensive as the memory footprint does not grow significantly. As a demonstration of our scalability and speed, we present an application in the segmentation of large Ovarian cancer (OCa) tissue microarrays containing stained vascular makers. Using a pathologist annotated ground truth, we present both quantitative and qualitative analysis of our algorithm, using 130 samples across multiple studies, proving the superiority of our method (sensitivity, Positive Predictive Value (PPV) of 81% and 80%) versus a popular supervised learning technique, Probabilistic Boosting Trees (sensitivity, PPV of 76% and 66%), at the significant speed increase of 62% per sample. We then briefly discuss the future work that will use HNCut to pre-process data. Essentially, we aim to discover a metric that can quantify the interaction between two classes of biological entities (the lymphocytes and cancer cells), such that it may lead to a prognostic indicator.

2 Motivation and Problem Definition

The field of digital pathology brings around many new challenges that were previously not conquered. It is not uncommon to have a significant number of images to work on, or that their dimensions are extremely large. Thus, not only does a high throughput method need to be developed, but there must be a great confidence in its reproducibility as it is extremely time consuming to rerun tests. Ideally, the new tools developed to combat these difficulties should provide accurate results that could lead to a standard approach, such that the results can be shared comfortably between institutions. These tools must focus on the quantification of tasks that were previously qualitatively performed by experts, in an adaptable flexible manner.

A Tissue Microarray is an array of small samples of cells that are mounted on a microscope slide. Samples (known as discs) are then treated with (perhaps different) stains to produce reactions in a uniform environment to indicate, for example, cancer. Our region of interest, the reactive area, is a chemically stained dark brown region (Fig. 1). Finally the light brown areas are to be ignored as they are considered to be artifacts. Our

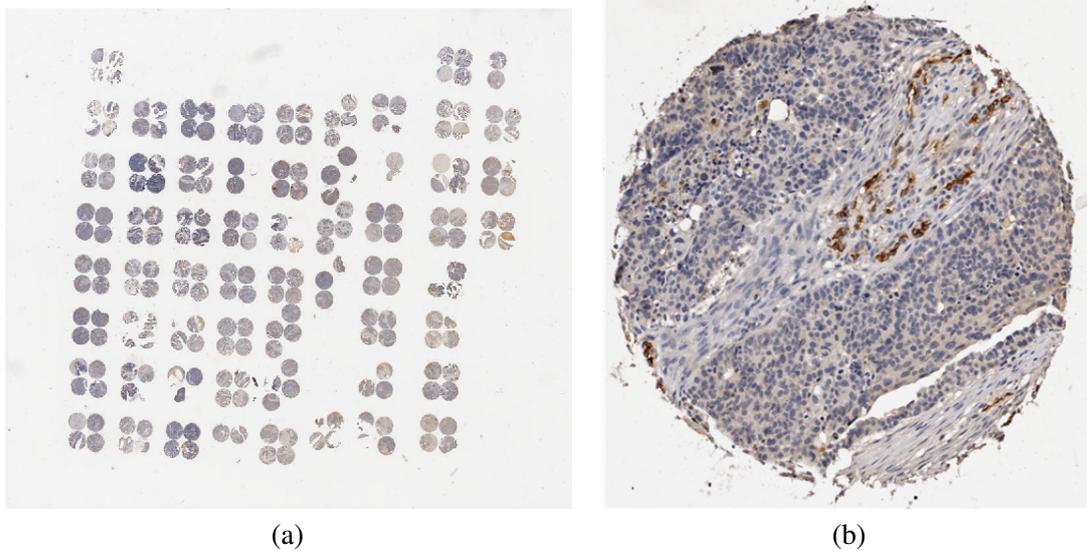


Figure 1: A zoomed out version of a TMA (a), and a single extracted cylinder (b). The TMA can contain over 500 individual discs on it, making the throughput of traditional algorithms come into question. The cylinder may contain large, perhaps disconnected portions of “brown” matter which indicate presence of a tested gene, and “light” brown artifacts where the stain steep into portions between cells. Although both the artifact and the stained region may be considered as brown, the gene indicator is indicated by the specific way clumps of brown appear in the stain.

goal then becomes the robust rapid extraction of this stained region given minimal domain knowledge from a layman. See Fig. 1 (b) for an example of a single spot(or disc), and in addition (see (a)) a glimpse at the large scale TMAs that are used.

TMAs, when scanned in, are images that are over 26 gigabytes (uncompressed) and have typical sizes of 11500 x 78000. There is research underway to push the amount of spots on a TMA to over 10,000 [1]. Each of these individual specimens will need to have the identical algorithm performed on it. Since a grid alignment technique has already been proposed by [2], its become trivial to analyze this one large TMA image as a set of smaller spots by extracting each of them as a separate image. It then becomes quite obvious how the overhead for such an operation can grow quickly to the point of infeasibility depending on the speed of the algorithm. It is infeasible from the onset for an expert to manually grade each of the samples, thus the need for these high throughput, accurate, algorithms.

As a result, our problem statement becomes the automation of these tasks, in reproducible manner. Phase I involved the goal of creating an algorithm to quantify the exact extent of a chemical stain upon TMA specimens. Phase II aims at developing a relationship between two different classes with the aim of providing a prognostic indicator to the pathologist.

3 Layout of the report

The rest of this report is organized as follows. In Section 4 we describe work progress. In Section 5 we list the associated publications. Section 6 describes a more in depth introduction, and Section ?? lists the problem statement and the novel contributions. In Section 8 we provide the methodological description of HNCut and in Section 9 we demonstrate its application in segmenting stained TVM from TMAs. Qualitative and quantitative evaluation results are presented in Section 10 with concluding remarks and future direction in Section 11. Finally, we discuss on a high level the overall application of this algorithm towards a prognosis indicator in 12.

4 Work Progress

- We have developed an algorithm to rapidly segment a target class based off of a few pixels from that class selected by a user. **(Done)**
- Using HNCut, we aim to be able to segment both lymphocytes and cancer cells from tissue samples **(Done)**
- Develop a metric associated with the organization of the lymphocytes and cancer cells **(Ongoing)**
- Attempt to associate prognosis with that metric **(Ongoing)**

5 Publications

- *Hierarchical Normalized Cuts: Unsupervised Segmentation of Vascular Biomarkers from Ovarian Cancer Tissue Microarrays* by Andrew Janowczyk, Sharat Chandran, Rajendra Singh, Dimitra Sasaroli, George Coukos, Michael D. Feldman, Anant Madabhushi was accepted as an oral paper at the 12th Medical Image Computing and Computer Assisted Intervention, 2009

6 Introduction

The field of digital pathology brings around many new challenges that were previously not conquered. It is not uncommon to have a significant number of images to work on, or that their dimensions (115,000 x 78,000 pixels) are extremely large. Thus, not only does a high throughput method need to be developed, but there must be a great confidence in its reproducibility as it is extremely time consuming to rerun tests. Ideally, the new tools developed to combat these difficulties should provide accurate results that could lead to a standard approach, such that the results can be shared comfortably between institutions. These tools must focus on the quantification of tasks that were previously qualitatively performed by experts, in an adaptable flexible manner.

6.1 Tissue Microarray

One major source of data is from tissue microarrays (TMA). A Tissue Microarray is an array of small samples of cells that are mounted on a microscope slide. Samples (known as discs or spots) are then treated with (perhaps different) stains to produce reactions in a uniform environment to indicate, for example, cancer (Fig. ??). The problem arises when a pathologist must analyze the resulting images (a typical TMA at 40x occupies about 1.5GB (compressed) of data on disk). It is incredibly laborious to analyze hundreds of samples, in a precise manner, for computing properties of the stain (e.g., “Is my cancer in a preliminary stage?”). Our goal is to come up with a high precision, robust method that, given standard stained TMAs, can produce a segmentation of cancerous regions enabling easy calculation of diagnostic properties.

There are many reasons why TMA is the leading technology for use in the pathology field, increasing the speed by which basic research can progress to clinical applications [3]. One major reason indicates that this technique allows professionals to maximize the amount of tissue that they can retain by only requiring small pieces of overall sample (0.6mm) for each of their sets, while previously whole slides of a single tissue sample had to be constructed. For the patients, this implies that they are required to give less samples, reducing the overall amount of physically discomfort.

Another reason is that each slide can be broken down into over 500 unique sections, allowing each section to be subjected to independent tests. From a scientific standpoint, this technology allows for an unprecedented amount of consistency and control. Instead of using 500 independent slides, scientists can choose to place 500 different subject’s samples on a single slide and apply a stain across all of them uniformly. This ensures that each of the samples will be treated homogeneously, exposed to the identical incubation times, temperatures and wash conditions [2]. This can not only greatly reduce the amount of inter patient disagreement, but greatly reduces the amount of overhead associated with certain testing patterns. Previously 500 different samples would have to be dealt with in perhaps 34 batches of 15, but now they can all be dealt with on a single slide. As an added bonus, the same amount of reagents required for a single large-section analysis is sufficient for a

thousand samples, making the overall process very cost efficient [2].

A valuable case study for the application of computer aided image processing is Ovarian cancer (OCa). It is estimated ¹ that 21,650 women will be diagnosed with and 15,520 women will die of cancer of the ovary in 2008. The 5-year survival rates of these women are highly correlated to the early detection of OCa. Recent work [4] suggests that specific tumor vascular biomarkers (TVMs) may be identifiable on OCa TMAs that could have prognostic significance, helping to not only predict this survival rate but also help determine a more specific course of treatment. Biomarkers are typically discovered by staining explicitly for TVMs of interest on OCa TMAs. Precise quantification of the extent and intensity of the stain could serve as a prognostic metric reflecting risk of disease recurrence and patient survival. However, it is currently impractical in terms of both time and effort for an expert pathologist to perform this segmentation manually.

6.2 Previous Work

Previous work in this field has used the traditional computer algorithms. Typically, for database creation, a sample by sample approach is taken. Essentially, each cylinder is individually extracted from its neighbors and operated on. In order to provide a high throughput approach, only the most basic of algorithms have been applied, such as thresholding [2], [5], [6]. While this elementary approach meets the speed requirements, it is often not very robust or reproducible as compared to more sophisticated algorithms.

The next approach was to use the k -means algorithm for segmentation [2]. k -means is notoriously difficult to use since its final result depends highly on the initial inputs of the system, creating a severe problem of irreproducibility. For different initial inputs, vastly different results can be expected. Biomedical images often contain large amounts of colors fully distributed across an image and thus are especially prone to being sensitive in the initialization of k -means [7].

Learning methods are a current popular approach for data classification and segmentation because they have the attractive property that given examples of the desired output, a classifier can be trained to produce similar output given unseen input. Unfortunately, one of the challenges in constructing a supervised classifier (such as Probabilistic Boosting Trees (PBT)) is the difficulty in obtaining ground truth segmentations for classifier training of the object, or region, of interest from experts [8]. In the medical field these experts are not only expensive, but are also very busy. As a result, the amount of training data available is small and of questionable correctness. While, once trained, PBTs operate rather quickly, the training step is often burdensome. This training process must be undertaken anytime the test data changes significantly from the training data, i.e. different stains. This brings back the problem of having to create a minimal amount of training data for the new test set.

¹Cancer Facts and Figures, American Cancer Society (ACS), Atlanta, Georgia, 2008

7 Problem Statement and Novel Contributions

The major contribution of this work is a fast, novel, hierarchical unsupervised segmentation method (HNCut), which we show an application of in Ovarian TMAs. Unlike traditional clustering algorithms, we aim to extract a single cluster pertaining to the stained region, while ignoring pixels in all other clusters. In traditional algorithms, pixels are allocated to the cluster that they are least dissimilar with, as opposed to being removed. Our setup encourages cuts that confidently trim away these undesired pixels. The unsupervised aspect of HNCut is particularly desirable for image analysis applications in histopathology and TMAs where obtaining annotated samples for training a supervised classifier depends on the annotations provided by an expert and hence difficult to obtain.

Our region of interest, the reactive area, is a chemically stained dark brown region (Fig. ??). Finally the light brown areas are to be ignored as they are considered to be artifacts. Our goal then becomes the robust rapid extraction of this stained region given minimal domain knowledge from a layman.

The algorithm that we propose, Hierarchical Normalized Cuts (HNCuts), is a fusion between both clustering and cutting techniques. By using a novel frequency weighted mean shift (FWMS) to create a resolution hierarchy, followed by a domain swatch cutting technique, we present an efficient and accurate that is capable of performing the aforementioned segmentation.

The mean shift (MS) algorithm was originally presented in [9] and revised in [10] as an unsupervised technique aimed at mode discovery for use in place of k -means. MS attempts to overcome its predecessors faults by searching for the mean data point within a certain bandwidth (σ) setting. By using the steepest gradient, a fast convergence to the set of true means of the statistical data can be found [11]. The improved fast Gauss transform implementation of the mean shift algorithm allowed computation times for large images to become reasonable for use in a clinical setting [12].

The Normalized Cuts (NCut) algorithm is among the final mature descendants from a series of graph cutting techniques ranging from max cut to min cut [13]. It is a popular scheme in spite of its main drawback: the large number of calculations needed for determining the affinity matrix and the time consuming eigenvalue computation. In large sized images the computation and overhead of these border on the infeasible [14], leading to a significant amount of research in avoiding their direct calculations [15]. When it is possible to reduce the overhead, these techniques become very powerful.

A novel approach of combining both the mean shift and normalized cuts algorithms was presented [16]. Clustering the image by running the mean shifting algorithm to convergence produced class assignments to the pixels. By taking the average value of those regions and using them as the nodes in the normalized cuts algorithm, a significant speed improvement was discovered. For a 240 x 160 sized image, they could perform clustering in under 3 seconds (using a 2Ghz processor), compared to the average time of the normalized cuts

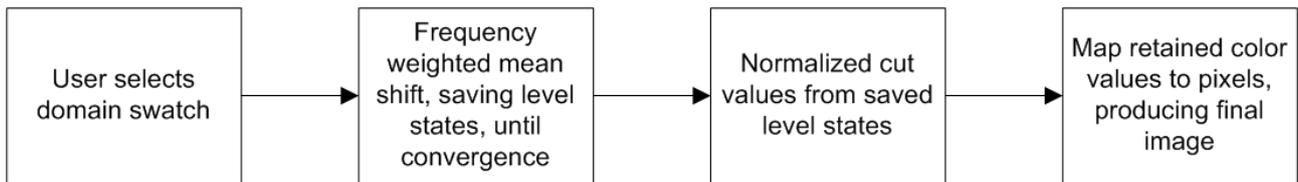


Figure 2: A simple flow chart of the HNCut process. Proceeding left to right, the user selects the domain swatch, followed by the mean shifting of the image to convergence (saving the required states along the way), following by the normalized cutting. After the normalized cutting is completed, the color values that remain in the system are mapped back to the original pixels and the final image is presented.

method at about 30 seconds. We use a similar concept, but extend it by exploiting the fact that as each iteration of the mean shift completes, the points converge. We will show how that convergence allows us to perform class extraction (which is of a slightly different purpose than [16]) on images of size 1,500 x 1,500 in under 6 seconds on a similar machine.

In summary, the work presented in this report represents important methodological and clinical contributions;

- A novel hierarchical segmentation approach that marries Frequency Weighted Mean Shift and Normalized Cuts (HNCut). HNCut not only operates on large (1.5 million or greater) images in under 10 seconds, but is easily scalable to entire TMAs. The affinity matrix can now take advantage of multiple features, and multiple color spaces efficiently across large window sizes

- Parameter insensitive segmentation for large images and the ability of HNCut to discriminate between regions with similar color values. The parameter for the Gaussian kernel in the affinity matrix of NCut is automatically computed. The parameters for the mean shift are automatically adjusted based on the variance of the output.

- Layman initialization of the system is possible, significantly reducing the difficulty of obtaining a full set of ground truths for training purposes.

- The first attempt, to our knowledge, to precisely quantify a vascular marker on OCa TMAs with the ultimate objective of creating a quantitative image based metric for OCa prognosis and survival.

Figure 2 illustrates a high level flow chart of the steps that will be presented in detail below. Figure 3 displays a typical cropped image from our dataset undergoing the HNCut procedure. The numbers presented are an example of typical output of a single 1,500 x 1,500 image.

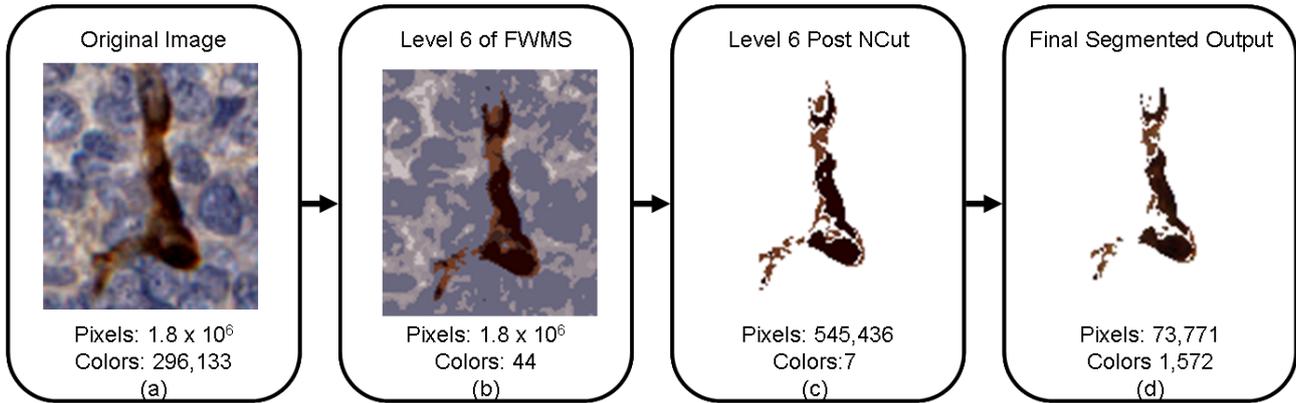


Figure 3: (a) Original sample, (b) Sample at the bottom of the pyramid, (c) Sample at the bottom of the pyramid post normalized cuts (15 cuts), (d) Final result at top level (approximately 25 cuts in total). It can be seen between (a) and (b) the significant amount of color resolution reduction, which allows the normalized cuts next in the sequence to be performed with ease. As the normalized cuts are done, while the image moves up the pyramid regaining its original color depth, we can see that various amounts of the remaining noise are removed leaving a near optimal result.

8 Detailed Description of Hierarchical Normalized Cut (HNCut)

8.1 Notation

The following notation is used consistently throughout the rest of the report. An image defined is defined as $\mathcal{C} = (C, \mathbf{f})$ where C is a 2D Cartesian grid of N pixels, $c \in C = (x, y)$ representing the Cartesian coordinates of a pixel (but may additionally be indexed linearly using c_i), and \mathbf{f} a color intensity vector associated with c , such that $\mathbf{f}(c) : \mathbb{R}^2 \rightarrow \mathbb{R}^3$. We define $\mathbf{f}_0(c)$ as the original color mapping vector of the image, written in short hand as $f_{0,i} = \mathbf{f}_0(c_i)$. Additionally, all vectors are presented in bold.

8.2 Integrating Domain Knowledge to Guide Normalized Cuts

We define a swatch (color template) reflecting the attributes of the object of interest in the scene. Hence a user is required to define through selection $Q = \{f_{0,\alpha_1}, f_{0,\alpha_2}, \dots, f_{0,\alpha_t}\}$ where $Q \subseteq \mathbf{f}_0(c)$ and represents the colors of the objects we seek. Note that Q is trivially determined by annotating (manually) a few pixels from the object of interest on a representative image and may be easily changed based on the application. As we will describe in further detail later, Q is only used to identify which color partition to retain during the NCutting process. It is important to note that the selected Q , since it is a subset of \mathbf{f}_0 , undergoes the same operations as the rest of the pixels and colors in the image.

8.3 Frequency Weighted Mean Shift for Reducing Color Space Cardinality

The mean shift algorithm is used to detect modes in data using a density gradient estimation. By solving for when the density gradient is zero and the Hessian is negative semi-definite, we can identify local maxima. This is discussed to great lengths in [11]. We start with their fixed point iteration update $\forall j \in N$ in Mean Shift(MS) as

$$f_{k+1,j} \leftarrow \frac{\left(\sum_{i=1}^N G_{\sigma_{MS}}(f_{k,j} - f_{k,i})(f_{k,i})\right)}{\left(\sum_{i=1}^N G_{\sigma_{MS}}(f_{k,j} - f_{k,i})\right)}. \quad (1)$$

Here G is a Gaussian function, with a bandwidth parameter σ_{MS} , used to compute the kernel density estimate at data point c_j . $k \in \{1, \dots, K\}$ represents various levels of color resolution produced at each iteration. The overall computation time for Eq. 1 is $O(N^2)$. By employing the improved fast Gauss transform [12], we can reduce the computation complexity to $O(N)$ with minimal precision loss.

It becomes possible to exploit the fact that after each iteration of the MS many of the data points converge. If we consider what that convergence means mathematically, essentially two points r, s meet the requirement $|r - s| \leq \epsilon$ where ϵ is an error tolerance. We can thus rewrite the numerator of Eq. 1:

$$G_{\sigma_{MS}}(f_{k,j} - f_{k,r})(f_{k,r}) + G_{\sigma_{MS}}(f_{k,j} - f_{k,s})(f_{k,s}) + \sum_{i=1, i \neq r, s}^N G_{\sigma_{MS}}(f_{k,j} - f_{k,i})(f_{k,i}) \quad (2)$$

as:

$$2G_{\sigma_{MS}}(f_{k,j} - f_{k,r})(f_{k,r}) + \sum_{i=1, i \neq r, s}^N G_{\sigma_{MS}}(f_{k,j} - f_{k,i})(f_{k,i}) \quad (3)$$

with the c_s calculation removed (remember, $f_{k,s}$ is short hand for $\mathbf{f}_k(c_s)$). This results in one fewer computation of the Gaussian, which is by far the most expensive operation in the computation process, at the cost of a single addition. As more points converge, say r, s, t and x, y , such that all of them are less than ϵ distance away from their cluster, it becomes obvious to rewrite the computation

$$G_{\sigma_{MS}}(f_{k,j} - f_{k,r})(f_{k,r}) + G_{\sigma_{MS}}(f_{k,j} - f_{k,s})(f_{k,s}) + G_{\sigma_{MS}}(f_{k,j} - f_{k,t})(f_{k,t}) + \quad (4)$$

$$G_{\sigma_{MS}}(f_{k,j} - f_{k,x})(f_{k,x}) + G_{\sigma_{MS}}(f_{k,j} - f_{k,y})(f_{k,y}) + \sum_{i=1, i \neq r, s, t, x, y}^N G_{\sigma_{MS}}(f_{k,j} - f_{k,i})(f_{k,i})$$

as:

$$3G_{\sigma_{MS}}(f_{k,j} - f_{k,r})(f_{k,r}) + 2G_{\sigma_{MS}}(f_{k,j} - f_{k,x})(f_{k,x}) + \sum_{i=1, i \neq r, s, t, x, y}^N G_{\sigma_{MS}}(f_{k,j} - f_{k,i})(f_{k,i}) \quad (5)$$

Resulting in a massive computational efficiency improvement. 5 Gaussian computations are replaced by 2, followed by a multiplication. It is trivial to see that the denominator follows the same reduction.

As a result, we rewrite the update presented in equation Eq. 1 as a multi step update. Initially, we determine the unique values in \mathbf{f}_k under the constraint that any points $|i - j| \leq \epsilon$ are considered equivalent. We store these unique values into a vector \mathbf{u}_k . Afterward, we store the frequency of appearance of these unique values in the original vector \mathbf{f}_k as \mathbf{w}_k . It is important to note the following definition of M_k , in association with the presented property:

$$|\mathbf{w}_k| = |\mathbf{u}_k| = |\mathbf{f}_{k+1}| = M_k \quad (6)$$

$$\sum_{i=1}^{M_k} \mathbf{w}_{k,i} = N \quad (7)$$

which leads us to the update of equation Eq. 1:

$$f_{k+1,j} \leftarrow \frac{\left(\sum_{i=1}^{M_k} (w_{k,i}) G_{\sigma_{MS}}(u_{k,j} - u_{k,i})(u_{k,i}) \right)}{\left(\sum_{i=1}^{M_k} (w_{k,i}) G_{\sigma_{MS}}(u_{k,j} - u_{k,i}) \right)} \quad (8)$$

for $\forall j \in M_k$.

We term this approach the Frequency Weighted Mean Shift (FWMS). In the end our use of FWMS produces a pyramidal scene representation $\mathcal{C}_k = (C, f_k)$, where $k \in \{1, \dots, K\}$ represent K levels of the pyramid. Note that $|u_1| > |u_2| > \dots > |u_K|$. In other words, FWMS results in a series of scenes \mathcal{C}_k , all mutually aligned, but with a fewer number of colors in the lower levels of the pyramid compared to the top. It is the fewer colors at the lower levels that enable NCut to be tractable, however, the upper levels of the pyramid are needed for high quality segmentation.

This process is represented in fig. 3 as the sequence from (a) to (b). It can be seen from the snap shot, that the overall color resolution is severely reduced as the algorithm proceeds from the top level to the bottom level. In this example, the original image containing about 300,000 unique color values was reduced to a mere 44 unique values. This significantly smaller set of values allows the next steps to be almost negligible in computation time.

8.4 Normalized Cuts on Mean Shift Reduced Color Space

The creation of a hierarchical pyramid of various levels of color resolution provides the input to the NCut algorithm. By setting the vertices (V) to the basins of attraction from the FWMS (i.e. unique color values, u_k) the NCut can segment the data by representing it as a connected graph ($\mathbf{G}=(E, V)$), with edges (E) representing affinity or strength of connectivity. A cut is the processes by which the removal of edges leads to two disjointed partitions. A value of a normalized cut between two disjoint sets A and B is computed using:

$$\text{NCut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)}, \quad (9)$$

$$\text{cut}(A, B) = \sum_{c_i \in A, c_j \in B} \psi(c_i, c_j) \quad (10)$$

$$\text{assoc}(A, V) = \sum_{c_i \in A, c_k \in V} \psi(c_i, c_k) \quad (11)$$

where $i, j, k \in \{1, \dots, N\}$. Note $A \cup B = C = V$. $\psi(c_i, c_j)$ is a function used to compute an affinity measure between c_i and c_j . Partitions A, B are identified as those which yield a NCut resulting in the most intragroup similarity and the most intergroup dissimilarity. This process can be recast and solved as a generalized eigenvalue system, which is explained fully in [14]. For our purposes it is sufficient to say that the optimal partition becomes the generalized eigenvalue system described by

$$(\mathbf{D} - \mathbf{\Psi})\mathbf{y} = \lambda\mathbf{D}\mathbf{y}. \quad (12)$$

with \mathbf{D} as a diagonal matrix with $\mathbf{d}(i) = \sum_j \psi(c_i, c_j)$ on its diagonal and $\mathbf{\Psi}$ as a symmetrical matrix with $\Psi(i, j) = \psi_{c_i, c_j}$.

In the traditional NCut, the $\mathbf{\Psi}$ matrix has a spatial constraint introduced such that:

$$\psi(c_i, c_j) = \exp\left(\frac{\|f(c_i) - f(c_j)\|^2}{\sigma_1}\right) \times \begin{cases} 0 & \text{if } \|c_i - c_j\| \geq \theta \\ \exp\left(\frac{\|c_i - c_j\|^2}{\sigma_2}\right) & \text{otherwise} \end{cases} \quad (13)$$

where θ is a user selected spatial radius, and σ_1, σ_2 are user selected bandwidth parameters, and $\|\cdot\|$ represents the L2 norm. Because of this spatial constraint, the affinity matrix $\mathbf{\Psi}$ is normally sparse, making its storage and operations less burdensome. In the case of finding color values that relate to the stained area, they can appear completely disjointed, thus criteria 13 must not be used. It is trivial to see that in this case the matrix is no longer sparse, and thus much of the efficiency overhead previously experienced by the NCut algorithm is lost. By operating on our hierarchical pyramid though, we can not only regain the lost efficiency but also experience additional speed benefits.

In our implementation of NCut we are concerned with partitioning the color space and not the image pixels per se. Hence if image \mathcal{C}_k comprises M_k unique colors, $V = \{u_{k,1}, \dots, u_{k,M_k}\}$ is the set of vertices corresponding to these colors. Thus $\mathbf{\Psi} \in \mathbb{R}^{M \times M}$ and A and B are obtained as unique color vectors $A \cup B = V$ for each application of NCut on \mathcal{C}_k .

Firstly we apply NCut on lowest image resolution (K) to partition the scene into two disjoint color sets A_K and B_K , where $A_K, B_K \subset V_K$, where $V_K = \{u_{K,1}, u_{K,2}, \dots, u_{K,M_K}\}$ is the set of all unique colors present in \mathcal{C}_K . To do this compute the affinity matrix $\mathbf{\Psi}_K \in \mathbb{R}^{M_K \times M_K}$, where for any $i, j \in M_K$, $\Psi_K(i, j) = \exp\left(-\frac{\|u_i - u_j\|^2}{\sigma}\right)$, where σ is a scaling parameter. We then use color swatch Q_K to identify unique color partition. We identify which of A_K and B_K uniquely contains all colors in Q_K . Hence if $Q_K \subset A_K$ and

$Q_K \cap B_K = \emptyset$ then eliminate all colors in B_K by setting $V_K = A_K$. If $Q_K \subset B_K$ and $Q_K \cap A_K = \emptyset$, similarly eliminate A_K by setting $V_K = B_K$. However if $Q_K \cap A_K \neq \emptyset$ and $Q_K \cap B_K \neq \emptyset$ then increase $\sigma + \Delta\sigma$, recreate Ψ_K , and repeat NCut on V_K to obtain a new set of disjoint color partitions. Keep incrementing σ until Q_K is uniquely contained within either of A_K or B_K .

Assuming for some value of σ , Q_K is uniquely contained in A_K , set $V = A_K$ and repeat NCut on V . We continue to perform the increasing of the σ and the cutting process until no further partitioning of color space at scale $\ell = K$ is possible. In other words, Q_K cannot be contained uniquely within the 2 color partitions for any value of σ .

Lastly, we repeat NCut and color partitioning at increasing color resolution scales. We migrate to the next higher image resolution, $\ell = K - 1$ and identify the higher color resolution swatch Q_{K-1} , identify V as the set of colors at resolution $\ell = K - 1$ but derived from the non-decomposable partition $\ell = K$. We then repeat the above process of identifying a unique non-reducible color partition at $\ell = K - 1$ that contains Q_{K-1} . This process is completed at all subsequent lower values of $\ell \in \{1, \dots, K - 2\}$. At $\ell = 1$, V contains a subset of values from u_1 , which are considered to be the chromatic values of the region of interest. Thus the final image is computed by retaining all pixels $j \in 1, \dots, N$ such that $u_{1,j} \in V$, and eliminating the others.

The hierarchical set of operations described above makes for an extremely efficient and accurate algorithm. Computing the affinity matrix and performing the normalized cut for the lowest levels of the pyramid is relatively trivial encouraging a more sophisticated definition of affinity using additional features, without penalty of significant overhead. In our case, we only had chromatic information, but this method can be extended cheaply and easily simply by adding additional features into the definition of Ψ .

9 Experiment Setup

9.1 Data Set Description

Our image database comprises of a total of seven digitized Tissue MicroArrays of ovarian cancer (OCa), in turn comprising over 500 tissue cylinders. The TMAs were obtained by sampling OCa tissue from over 100 patients and were stained for the presence of the tissue vascular marker ESM-1, resulting in vascular regions with the antibody to ESM-1 staining brown. The original TMA files, of at least 7GB each, are down-sampled and stored at 20x magnification, producing images that are approximately 1500 x 1500. An expert pathologist annotated 130 cylinders. The exact regions highlighted by him were extracted and stored separately, creating a binary map used as the ground truth.

9.2 Evaluation Description

An evaluation is presented using a macro metric. For pathologists, their concern rests mostly in the locating of regions in the sample. To account for this requirement a macro metric operates on a region level. Both the test image and ground truth are reduced to 25% of their original size. This is done to emphasize the core properties of the region and not the variable boundaries. If the intersection of the two regions is greater than 30% of the ground truth area then a true positive has taken place, other wise a false positive. The same operation is performed again, except going from the ground truth back to the output image, to determine any false negatives.

9.3 k -means

A standard k -means algorithm was performed using 10 clusters. The initial cluster centers were manually chosen, by experimentation, to provide a good overall representation of the image.

9.4 Probabilistic Boosting Trees

Although a full description is outside the scope of this report, we present a brief introduction to Probabilistic Boosting Trees (PBT). A full description and implementation details can be found in [17]. PBTs are a supervised classifier that attempt to learn a training set so that they may respond accordingly to classification queries on unseen data. Each level of the tree involves the construction of a strong classifier from multiple weak classifiers. Traditionally, an algorithm such as Adaboost is used for this purpose. The tree then expands in a binary fashion, breaking the training set into two groups. The idea is that while it may be difficult to create a single linear classifier to separate a set, perhaps multiple levels of classifiers can perform the task.

Consider Fig. 9.4. The goal is to be able to robustly separate the blue points from the red points. At the first level, towards the left, all of the pixels are bunched together. After training a hard classifier, it becomes trivial to remove the obvious cluster in the upper right hand side, but the rest of the points are too intermixed to separate easily at this stage. As a result, the intermixed group gets a hard classifier trained specifically for that purpose. For the red points that fail, another level is trained until the proper model is determined. We can see that as we move from left to right in the system, each classifier is trying to solve a smaller problem as opposed to tackling the entire system. The binary aspect of the tree also makes this algorithm quite fast during computation tests.

In our case, we've chosen to use the suggested default values for both of the PBT variables in [17] θ and ϵ (.45 and .4, respectively). Each strong Adaboost classifier was constructed using seven weak classifiers. The PBT performed seven levels of dataset separation, in the prescribed tree like manner. The training set was created by taking a 3×3 window around every $c \in C$, across all 3 color channels in HSV space, resulting in a 27 dimensional vector. 1000 random positive (stained) samples and 1000 random negative (unstained and spuriously stained) samples were selected from 25 randomly selected images, resulting in a total training vector

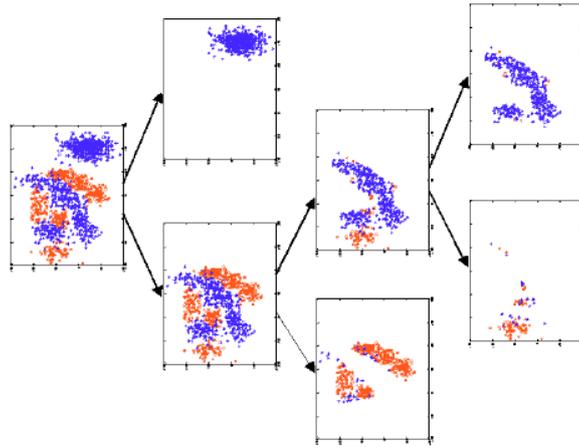


Figure 4: Illustration of PBT on a synthetic dataset of 2,000 points. Weak classifiers are likelihood classifiers on features such as position and distance to 2D lines. The first level of the tree divides the whole set into two parts. The right side mostly has blue (dark) points since they are away from the rest of the clouds. The tree expands on the parts where positive and negative samples are tangled. Figure taken from [17].

of size $27 \times 50,000$.

9.5 Hierarchical Normalized Cuts

FWMS was performed using a $\sigma_{MS} = .05$, with the number of clusters for the improved fast gauss transform set to the square root of the number of pixels in the image, as suggested by [12]. When the number of remaining clusters fell below this value, it was reset to the square root of the number of remaining clusters. NCut uses the Silverman function [18] to determine its initial σ and then increases by a factor of 10 as prescribed above. The domain knowledge is six pixels of shades of brown that we find acceptable as stain. Our affinity measure is as defined in Algorithm 2. Note, the spatial constraint is removed as to not require pixels to be adjacent to each other. While this removes the sparse property of Ψ , our algorithm is agnostic since the FWMS provides a trivial number of clusters for the NCut to analyze.

9.5.1 Implementation

There are only really two major things that need to be considered when running the algorithm, the first is the swatch that will be used, and the second is the FWMS σ parameter.

Before the creation of FWMS an in depth discussion of the Improved Fast Gauss Transform’s clustering variable selection would have been required, but in using the FWMS, this is no longer needed. As the number of clusters were increased, making the approximation more accurate, the computation time also grew steadily. Using our new implementation, the authors recommendation of the square root of the number of data points

becomes extremely fast even for large images of over 1.5 million points.

Algorithm 1 Frequency Weighted Mean Shift to Generate Color Pyramid $f_{1\dots k\dots K}$

Input: f_0 of \mathcal{C}_0

Output: $u_{0\dots K}$

```

1:  $k = 0$ 
2: while  $k < maxK$  and  $|u_k| > minColor$  and  $|f_{k-1} - f_k| > stopTresh$  do
3:   Compute the unique values of  $f_k$  and store them in  $u_k$ 
4:   Compute the frequency of appearances of  $u_k$  in  $f_k$  and store them in  $w_k$ 
5:   Generate  $f_{k+1}$  using Eq 8
6:   Generate  $w_{k+1}$  and  $u_{k+1}$  using  $f_{k+1}$ 
7:   if  $|u_{k+1}|/|u_k| > NewLevelTreshold$  then
8:      $k = k + 1$ 
9:   else
10:     $f_k = f_{k+1}$ 
11:   end if
12: end while
13: return  $u_{0\dots K}$ 

```

An algorithmic presentation for FWMS appears in Algorithm 1. The *while* loop has the stopping clauses for the function. Firstly, a test to insure that the number of levels is beneath the user selected maximum. In practice this number is rarely met and is often set as high as 20, with the average stopping level being around 12. This rapid convergence is again due to the steepest gradient approach. The next is a test to determine if there are a minimum number of values present to continue operating on. There is little reason to continue converging if there are less than 10 unique colors present. The last criteria for stopping is the one more often met, which is to determine if the system has converged enough. This is done by analyzing two adjacent levels (k and $k - 1$) and determining if their color mapping functions are within some stopping threshold. Afterwards, the algorithm proceeds as expected until the *if* statement. This is performed in order to validate that enough change has occurred in our system to merit the saving of the state. It is often, with all iterative algorithms, that saddle points are reached and thus their information isn't required to be saved.

Algorithm 2 describes the entire NCutting process, which is presented in line with the methodology section.

9.5.2 Efficiency

It is straightforward to see the efficiency improvement in the FWMS over the traditional MS, especially in the case of image segmentation. Since vision applications do not require a high level of precision, we can set our ϵ to a relatively large value. Figure 5 shows the numerical advantages to such an approach. When the initial number of points is large, after each iteration, fewer computations need to be performed. The larger ϵ is selected, the faster the algorithm will converge, on the other hand, when ϵ is selected to be extremely small the final answer approaches the unsimplified answer, although it will still obtain that answer in a much quicker

Algorithm 2 Normalized Cuts on Mean Shift Reduced Color Space

Input: $\mathbf{u}_{0,\dots,K}, \mathcal{Q}$
Output: \mathbf{u}_{final}

```

1:  $k = K$ 
2: Build  $\Psi_k$  using  $\forall_{i,j \in M_k} \psi(u_{k,i}, u_{k,j}) = e^{-\frac{\|u_i - u_j\|^2}{\sigma_1}}$ 
3: while  $k \neq 1$  do
4:    $\sigma_1 =$  initial  $\sigma$  value
5:   while  $\sigma_1 \leq \text{maxsigma}$  do
6:     Solve for  $A, B$  by using Eq. 12
7:     if  $Q_k$  is not uniquely contained in  $A$  or  $B$ , i.e. it has split then
8:       Increase  $\sigma_1$  and recalculate  $\Psi$ 
9:     else
10:       $\Psi_k = \begin{cases} \Psi_{k_i \in A}, & \text{if } Q_k \in A \\ \Psi_{k_i \in B}, & \text{if } Q_k \in B \end{cases}$ 
11:    end if
12:  end while
13:   $k = k - 1$ 
14:  Build  $\Psi_k$  using  $\forall_{i,j \in \Psi_{k-1}} \psi(u_{k,i}, u_{k,j}) = e^{-\frac{\|u_i - u_j\|^2}{\sigma_1}}$ 
15: end while
16:  $\mathbf{u}_{final} = u_{1,i}, \forall_{i \in \Psi_2}$ 
17: return  $\mathbf{u}_{final}$ 

```

fashion.

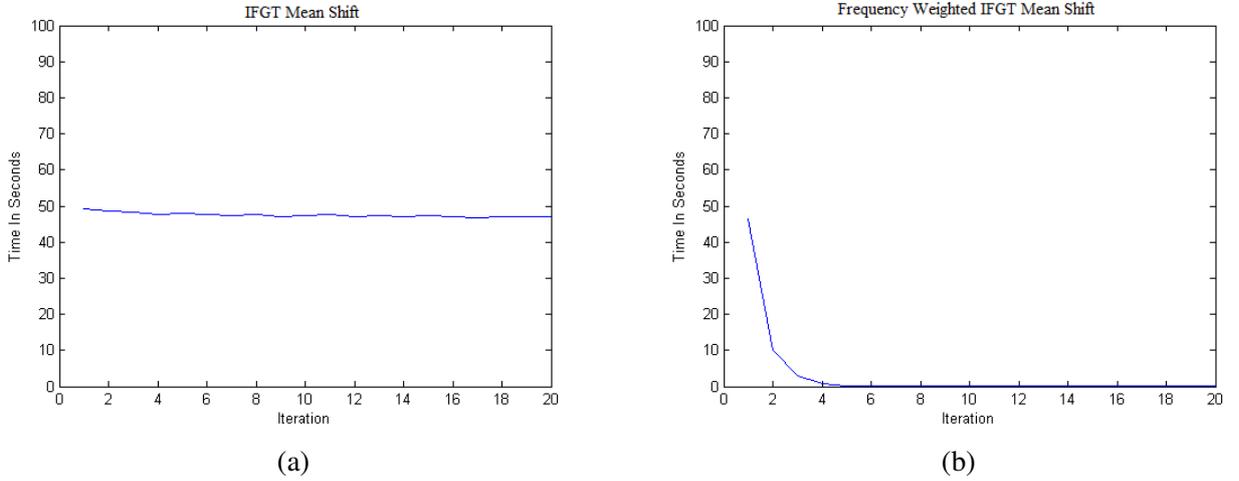


Figure 5: Two graphs showing the typical time for each iteration of the mean shift as convergence of the points occurs. The original Improved Fast Gauss Mean shift (IFGT) (a) has constant time for each iteration. The benefits of the Frequency Weighted Mean Shift (FWMS) algorithm (b) become apparent as iteration times become trivial as the data points converge.

The easiest way to apply this ϵ requirement into the algorithm is to simply choose a level of desired precision and then perform a rounding operation. Afterwards, the procedure of locating unique values and computing their frequencies is as simple as producing a histogram of the data values with each unique value being its own

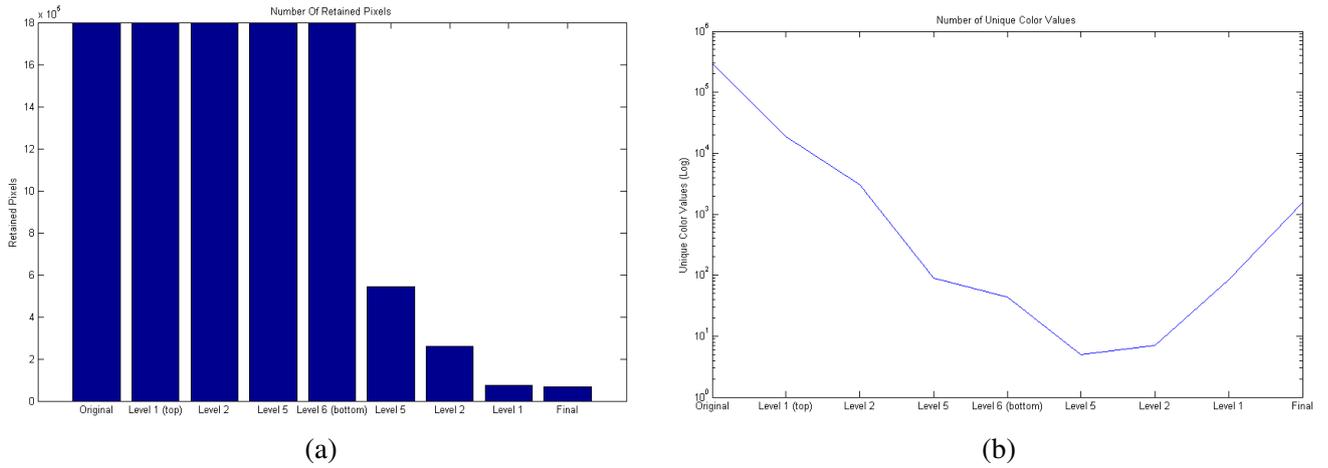


Figure 6: The HNCut algorithm in both graph (a) and (b) proceeds from left to right. The number of retained pixels remains constant during the FWMS stage, indicated by the bar graph (a), while the number of unique colors (i.e. color resolution) is reduced, indicated by the line graph (b). As the normalized cuts are performed, the number of pixels is reduced significantly, until the final image, in original resolution, is presented. Note how graph (b) is presented on a logarithmic scale.

bin. This is a significant benefit, as the production of histograms is not only well studied but easily converted into a parallel computing problem.

A clear numerical motivation for our work can be viewed in fig 6. Up until the bottom level of the pyramid, only FWMS is taking place, reducing the cardinality of the color space. During this process no pixels are being pruned. Once at the bottom level the normalized cuts beings and the cutting away of undesirable pixels take place. After the completion on the bottom level, we move up to the next level $K - 1$. At this point, the previously retained colors are mapped to their parent level, resulting in an image of slightly better color resolution. This proceeds until conclusion, which results in an image at the original color resolution, all be it a small subset of it, with all of the extraneous pixels removed.

Starting off at the top level and expecting to reasonably cut out the proper 1,500 color values from the 300,000 colors is difficult at best. If we can reduce this number down to 50, and cut away all but a few colors, not only do we remove 95% of the pixels present in the overall image, but we can make those cuts with confidence because we have solid domain knowledge information backing up the decision.

9.5.3 Swatch Selection

The results produced by the algorithm are obviously highly dependent upon the swatch used. It is quite difficult to pick the exact perfect swatch from the beginning thus a more organized approach is usually taken in its creation. It is best to start of with one or two sample data points from the desired class and run the algorithm. Upon examining the result, it becomes easy to visually determine which chromatic values were improperly cut

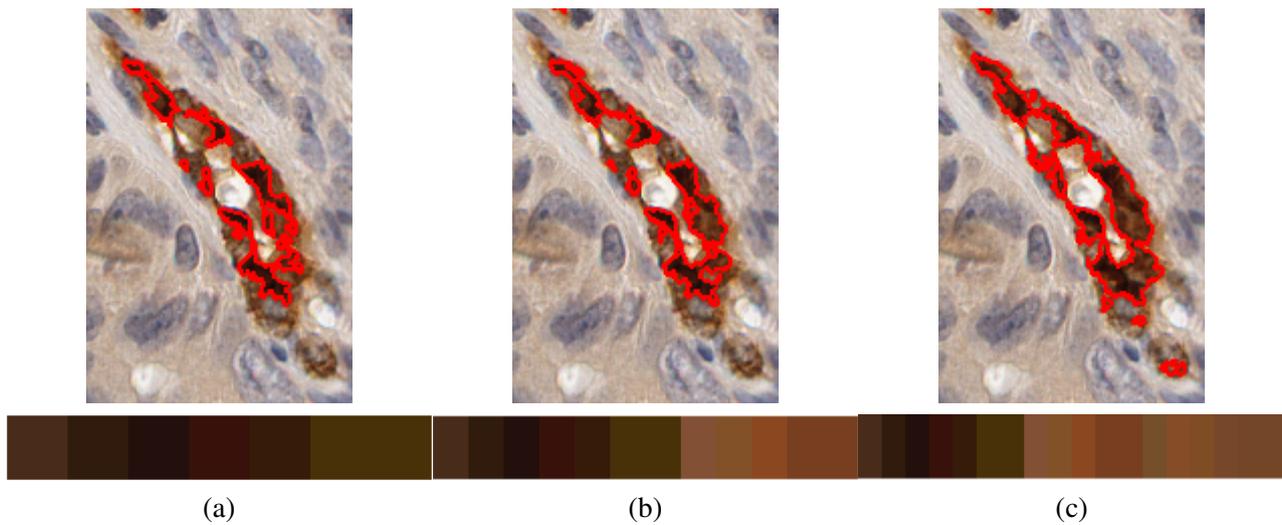


Figure 7: The above shows the variance in output given some sample patches. The first column (a) was created using 7 selected pixels. The next column (b) contains the same values as (a) with the addition of another 5 values. The final column (c) has 18 values selected from the original image. The red line encapsulates the results of the segmentation algorithm. We can see that the first set of results are fairly good, but as we add in additional samples from the desired class, the results improve.

away. As a result, the user can add those values into the swatch and re-run the algorithm again. Since each run of the algorithm takes less than 6 seconds, this hardship is considered to be acceptable.

For completeness we present the following results in Figure 7. A small patch was randomly selected from the desired class by a layman. The resulting segmentation was overlaid using a red boundary on the original image. Afterwards a few more sampled pixels were added to the swatch, and the results recomputed. Finally, a few more were added and the results are near optimal. From the presented image swatches beneath the images, we can see that when the user selects dark pixels as their color template, the segmentation focuses on the darker aspects of the stain. When the final swatch was used, which is an accurate representation of the entire class, the results approached the correct result. It is interesting to note how easy it is to modify the patch based off of the output. Looking at column (a), it would be trivial for a layman to determine which areas of the target class they did not sample from and include those in the swatch. This iterative process can be done until the layman sees that the results of algorithm match the exact desired output. This is to say that there may be times when only the darker stain is required, and the light stain is to be ignored. This domain swatch techniques allows for the exact specification of such a target. As a result, the question of the reproducibility of the output of the algorithm should be put to rest, by providing this mechanism, getting very similar results from scratch, without the retention of the swatch, becomes a simple iterative task. Once the domain swatch is selected, it can safely be used for the rest of the images in the TMA set.

10 Results

10.1 Qualitative

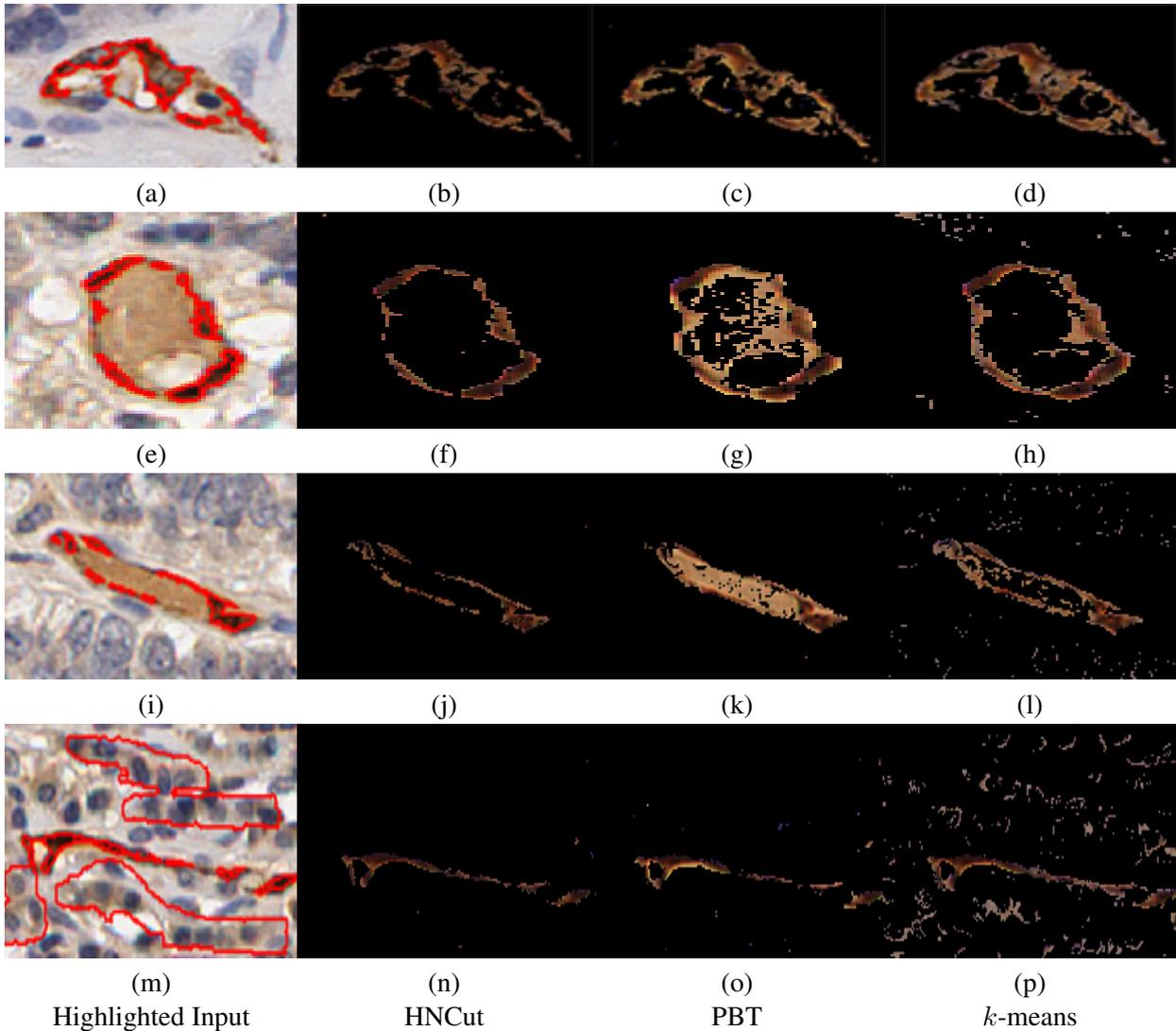


Figure 8: Selected examples are presented. The original input, with the annotated ground truth in red, is presented on the left, followed by HNCut, PBT and k -means.

A subset of results are presented in Fig. 8. The first column represents the original input image, with the boundary of the ground truth highlighted by the pathologist labeled in red. The first row illustrates a case where all of the algorithms performed as expected. The second and third rows illustrate times where the HNCut algorithm performs optimally while both the PBT and k -means extract many miscellaneous false positives. The final row is used to illustrate a scenario where false negatives occur for all three methods. The middle region is correctly segmented in all algorithms, while the three other regions are incorrectly rejected. The stain in those regions is only barely visible to an expert. Lastly, we can see that although k -means does the best in attempting

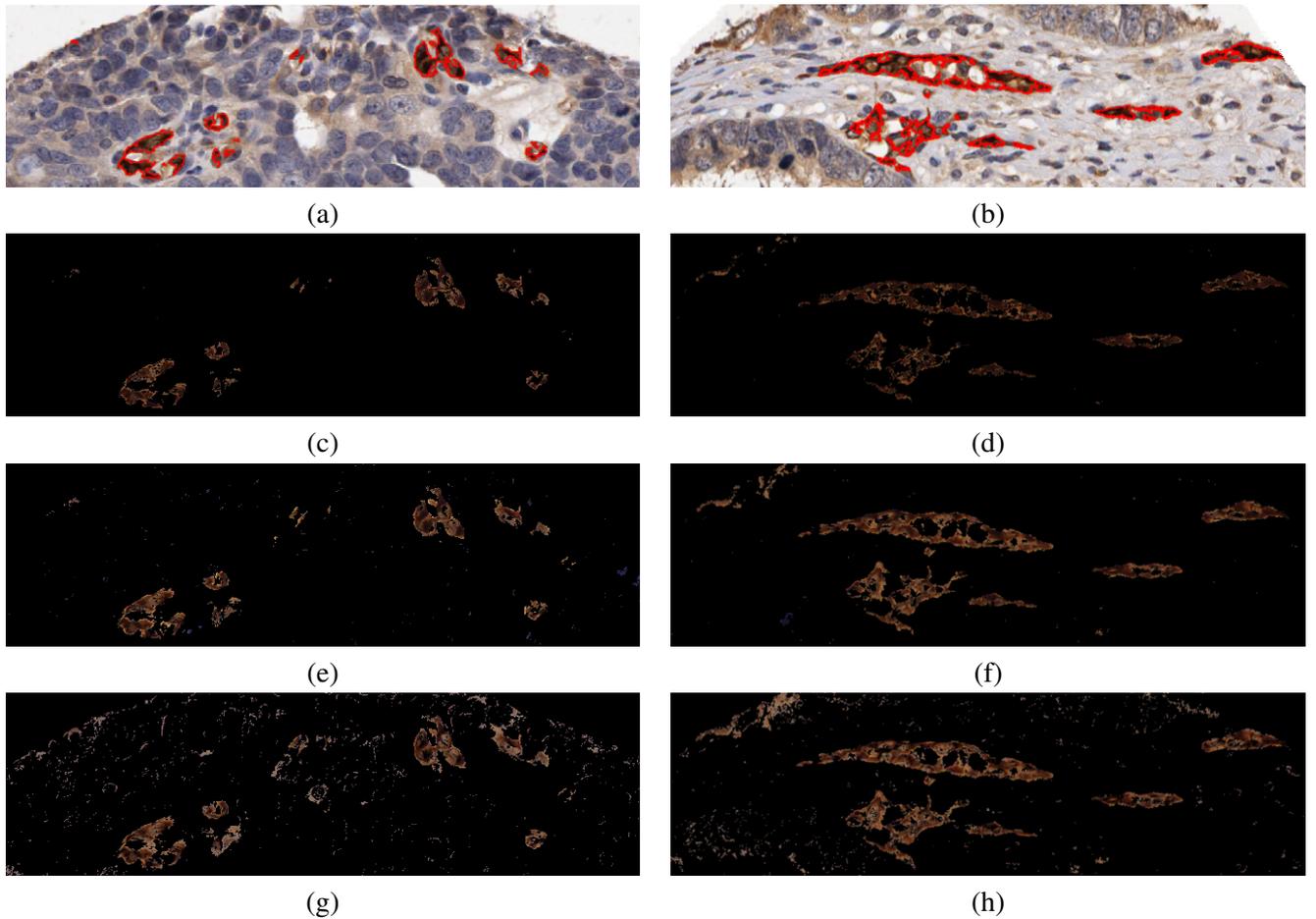


Figure 9: Two bands across selected TMA spots are presented. The original input, with the annotated ground truth in red, is presented on the top, followed by HNCut, PBT and k -means.

to detect those regions, the same setup results in many false positives in other images. This is a result of the k -means requiring all pixels to be assigned to a cluster, filling the stain cluster as it the most similar out of the candidates. Both PBT and HNCut determine in most instances that these same values are simply too different to be included as stain.

10.2 Quantitative

In the case of the PBT, 50 cross validation iterations took place resulting in an average Area under the curve (AUC) of .9296 with a standard deviation of .0039. The probability returned by the PBT was converted into a strong classifier by taking the upper Otsu threshold [19].

Figure 10 quantitatively illustrates that HNCut provides the best balance between sensitivity and specificity, providing 50% less false positives (FP) than PBT and 86% less FP than k -means. All of the algorithms seem to do very well in the specificity category, but this is a result of the vast number of true negative pixels (TN)

	False Negative	True Positive	False Positive
HNCut	$378.8 \pm 108.5\sigma$ (18.1%)	$1711.3 \pm 108.5\sigma$ (81.2%)	$717.7 \pm 141.4\sigma$
PBT (97% confidence)	$423.3 \pm 50.3\sigma$ (20.2%)	$1648.7 \pm 50.3\sigma$ (78.6%)	$1218.8 \pm 314.8\sigma$
PBT (99% confidence)	$761.7 \pm 228.5\sigma$ (36.3%)	$1310.3 \pm 228.5\sigma$ (62.4%)	$541.2 \pm 251.1\sigma$
PBT trained with HNCut (97% confidence)	$338.2 \pm 41.8\sigma$ (16.1%)	$1733.8 \pm 41.8\sigma$ (82.6%)	$700.4 \pm 126.5\sigma$
PBT trained with HNCut (99% confidence)	$559.5 \pm 72.7\sigma$ (26.7%)	$1512.5 \pm 72.7\sigma$ (72.1%)	$500.5 \pm 32.5\sigma$
<i>k</i> -means	to come	to come	to come

Table 1: Performance measures of Tested Classification Algorithms on Ovarian TMA data. Each algorithm was run 10 times with different initial conditions and their mean values are presented with the standard deviation next to them in parenthesis.

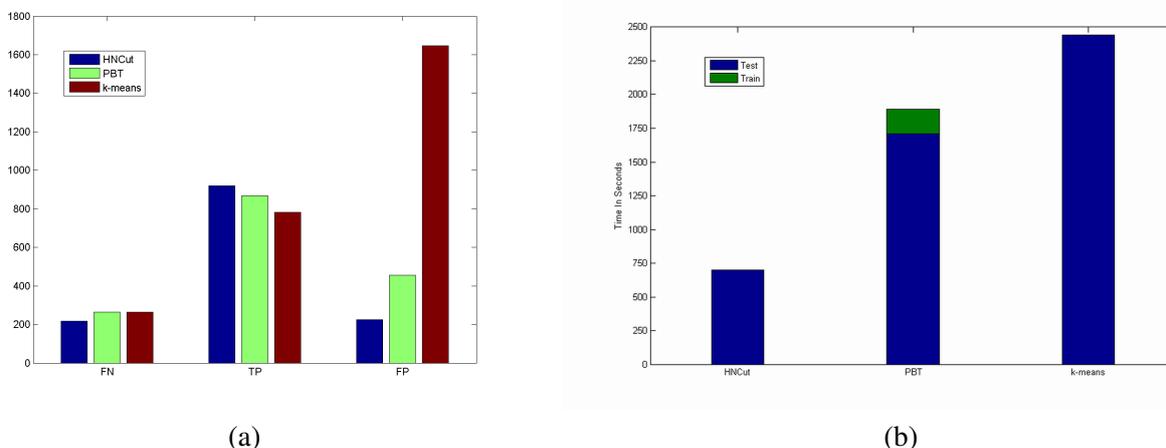


Figure 10: (a) False Negatives (FN), True Positives (TP) and False positives (FP) presented for the three different algorithms.. From this visualization it is apparent that HNCut outperforms each of the other algorithms. (b) A comparison of the run times of the different algorithms. We can see that HNCut performs the fastest out of those tested, motivating its high throughput compatibilities. Interestingly if both PBT and HNCut were start at the same time, after PBT has completed its training stage, HNCut would already be 25% done with our dataset associated with the majority of the sample.

The reproducibility of these algorithms is presented in Table 1. Each of the algorithms was run 10 times with different initial conditions and then graded. The standard deviation (σ) is presented in parentheses next to the means. We can see, as expected, that the *k*-means algorithm depends highly on the initial conditions. Interestingly, randomly generating the training set for the PBT from the ground truths provided by the expert seems to lead to a larger variance in the data. This can be as a result of human error in the marking of the ground truths, or the selection of pixels that do not truly mimic the rest of the desired class.

10.3 Run Time Comparison

In order to completely motivate our highthrough put approach a simple comparison of the runtimes needs to be considered. Figure 10 presents a graphical representation of the results. While PBT's training time of 181 seconds would account for 25% of HNCuts 643 second run time, it is normally averaged across all samples. Thus the more samples that are tested, the cheaper it becomes to train the system. Irregardless, even upon excluding the training time for PBT, HNCut still performs significantly faster. The average of 16 seconds per sample by PBT is easily overshadowed by the runtime of 6 seconds per sample by HNCut. This results in a speed increase of 37%. Across thousands of samples, it is quite apparent how much of a significant time saver HNCut becomes.

10.4 Turing Test

An interesting question that we wished to investigate revolved around a Turing style question. Could the PBT differentiate between the HNCut and the ground truth? This is to say, is there any difference between performing the training operations for the PBT using the HNCut as the ground truth versus an expertly annotated training set. To examine this question, we performed 10 iterations of the training/testing procedure using the HNCut output. The results are as presented in Table 1. It can be seen that by training the PBT with the output from the HNCut, the results actually fair better, with a tighter standard deviation. In a sense, we can say, that the PBT is incapable of determining if its training input comes from an expert or the HNCut algorithm, essentially letting us say that our HNCut algorithm can pass this Turing test.

11 Discussion

The presence of a large number of false positives merit an additional note. Since the stain is a representation of the amount of a gene present, the intensity (i.e. darkness) of the stain is proportional to the quantity of the biomarker. As a result, the stain will vary greatly in intensity of color across not only all areas of the spot but also across all spots themselves. Hopefully it has become obvious to the reader why the selection of k -means initial clusters becomes challenging.

In most cases, the rim's of the discs tend to have noise significantly darker than most of the rest of the image. Thankfully, pathologists rarely consider results on the boundary of the spots, making the removal of such artifacts in post processing a trivial function of distance from the disc center, which we consider a post-processing step.

Another issue, which is a more difficult, is presented in Figure 11. In the situation where the disc is not well formed, from either tissue tearing or an absence of cells, there is the possibility for large scale pooling of false positive stain. The chromatic qualities of the stain are too close to that of the artifact, and differ highly

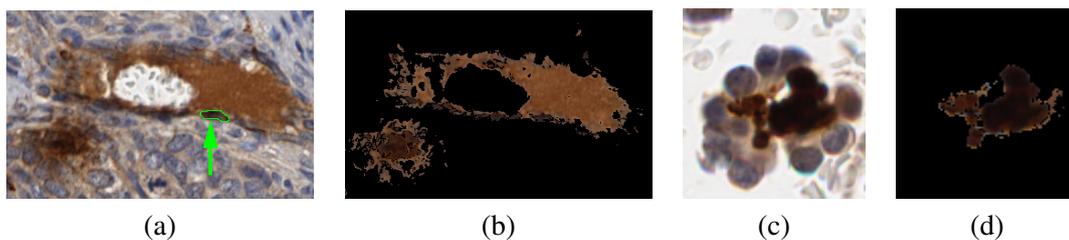


Figure 11: An example of stain artifact that results in either false positives or false negatives. In the original image (a), a circle and arrow denote a vessel that is correctly stained but nestled inside of an artifact area, preventing the use of size thresholding. On (b) it can be seen that our algorithm successfully maintains the true positive at the cost of some false positives. An example of a psammoma (c) and its associated segmentation (d). While visually these objects can be excluded from the segmentation because of their shape characteristics, chromatically they are no different than regions that stain darkly.

between images, making the automatic removal very uncertain. It was initially thought that since these areas tend to be very large and irregular compared to the desired region, they would be trivial to filter out. Upon further inspection by the pathologist, it came to light that there are often true positives that are adjacent to the area in question. This results in a trade off, either the entire area can be deemed as artifact and a performance loss is experienced on the true positives side, or we can retain the artifact and gain false positives.

Psammomas (refer to Figure 11) are calcified material within the center of a laminated whorl of elongated fibroblastic cells. Unfortunately, psammomas are exactly the same in color and style as the true positives, making it difficult even for an inexperienced human to classify. As a result of the severe similarity we have to accept the false positive rates that are associated with them.

All in all, the false positive rates have to be taken with a grain of salt. While the algorithms function as they're intended, providing a proper segmentation based on the chromatic information, the difficulty of the situation is inherent in this specific biomedical application.

The FWMS's σ is the only parameter that may not be trivially understood. This σ 's value depends highly on the dataset. For example, a σ value that works well with an image stored in integer format (i.e. values between $[0,255]$), will not be suitable for an image stored in double format (i.e. values between $[0,1]$). As a result, a suitable bandwidth parameter must be found. Typically for double formatted values, a good starting place is .05 as a σ value. In the final result, if the user determines that the result is cut too aggressively the sigma value should be increased. The opposite is also true, thus if the pixels aren't sufficiently pruned, the σ value should be decreased.

In figure 12, the importance of selecting the correct σ and domain patch become apparent. In the case where the σ value is too large, the FWMS clumps together inappropriate pixels. As a result, they can never be pruned away as shown in (a). The highlighted blue section is dark enough in color that it becomes associated with the stain due to the large bandwidth selection. On the other hand, when a proper domain swatch is selected, that

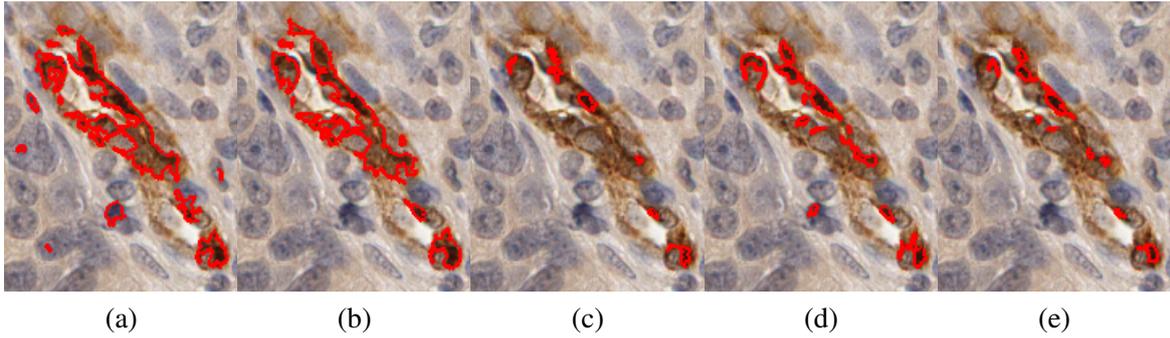


Figure 12: (a) & (b) above shows the segmentation output as a result of selecting two different σ values. The left image (a) was created using an grossly inappropriately selected value of .3. From it we can see that since the bandwidth parameter is too large, colors that are un-similar tend to get clustered with the stained region, in this image visualized by the blue area additionally circled in red. (b) shows the result of a segmentation with a small σ value of .01. The algorithm rarely experiences unacceptable segmentations in the case of too small of a σ value if the domain swatch is properly defined. (c),(d) are again presented with σ values of .01 and .3 respectively, except an improper domain swatch was selected. To compare (e) is the poorly selected domain swatch with a σ value of .05.

truly resembles the desired target class, almost any σ value becomes acceptable, as shown with the extremely small σ value of .01 in (b). Since it becomes impossible to prune away values that are similar to the domain swatch, even when a small σ value is selected, most appropriate values will be retained. Unfortunately, in the case where a poor color template is created, as in (c),(d) and (e), the results suffer greatly. It can be seen that, even though the results are poor, the large σ value still retains values of blue that should have been pruned away. It also becomes evident that even selecting an appropriate σ value doesn't improve the results if the domain swatch is fundamentally flawed.

In our specific application, using HNCut on 500 discs, about 10 of them failed to converge properly, resulting in very poor segmentations. Interestingly, these 10 images all had little to no stain present. By computing the variance of the color pixels that are in the segmented output against the domain swatch, we can assess the performance of HNCuts and make relevant adjustments in an unsupervised manner. For instance, if the variance is larger than desired, adjusting σ_{FWMS} to a smaller value will produce new output that is more similar to the domain swatch. In the case of poor output, the experienced variance is typically of an order of magnitude greater than the variance in the correctly segmented images. This fact makes the determination of needed adjustment rather trivial. Thus we can claim, all in all, except for the specification of the swatch, the scheme is completely unsupervised. In our ten images the automatic adjustment of the σ_{FWMS} took place, as described above, to a narrower setting resulting in appropriate results in all of the experiments we ran.

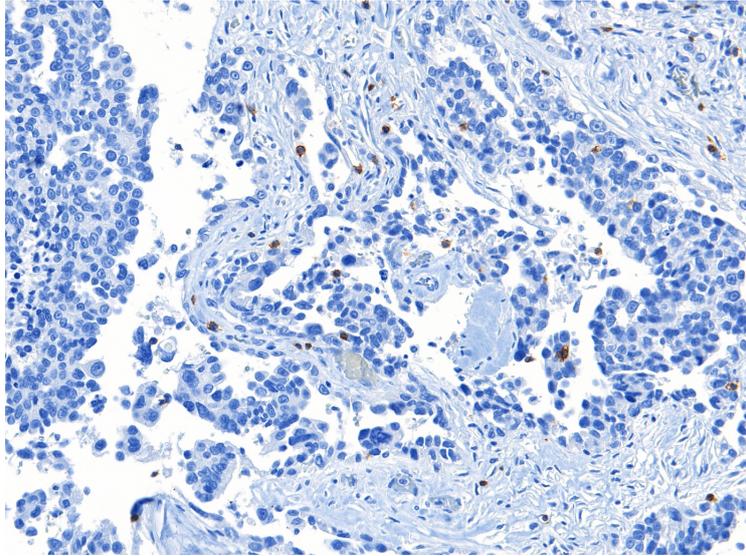


Figure 13: Original image with cancer cells stained in blue and lymphocytes stained in brown.

11.1 Whole TMA Processing

Due to the manner in which HNCut operates on the image, it is possible to perform a segmentation on the entire TMA, all 26GB, at one instance. After the generation of \mathbf{u}_1 , the individual pixels are only analyzed again at the final step. It is well known that extracting unique values from a large vector, even one of 7GB in length, is rather straight forward and fast. Typically, the number of unique color values of a TMA will be on a similar scale as the number of unique colors in the individual images that make it up. Thus, once that processing step is performed, the algorithm operates with the same efficiency, regardless of the original image size. Afterwards, upon the completion of the final NCut in HNCut, the only re-interaction with the pixels takes place, within which a determination if the pixel's color value belongs to the final \mathbf{u}_{final} . This computation must take place regardless if the large TMAs broken into individual images, and is not only trivial to perform but also can easily take advantage of a parallel computing environment.

12 Future Work

Current research is suggesting that there may be a link between lymphocyte arrangement amongst tumor cells and prognosis. These Tumor Infiltrating Lymphocytes (TILs) are fairly easy to stain for. Currently, we are able to use the HNCut algorithm to segment images such as Fig. 13 and producing two sub images from it as shown in Fig. 14, with the lymphocytes and cancer cells isolated.

The goal is to create a metric that analyses the relationship between the sub images. Afterwards, using a significant amount of clinical data, a link between various arrangements should appear. This link should be in

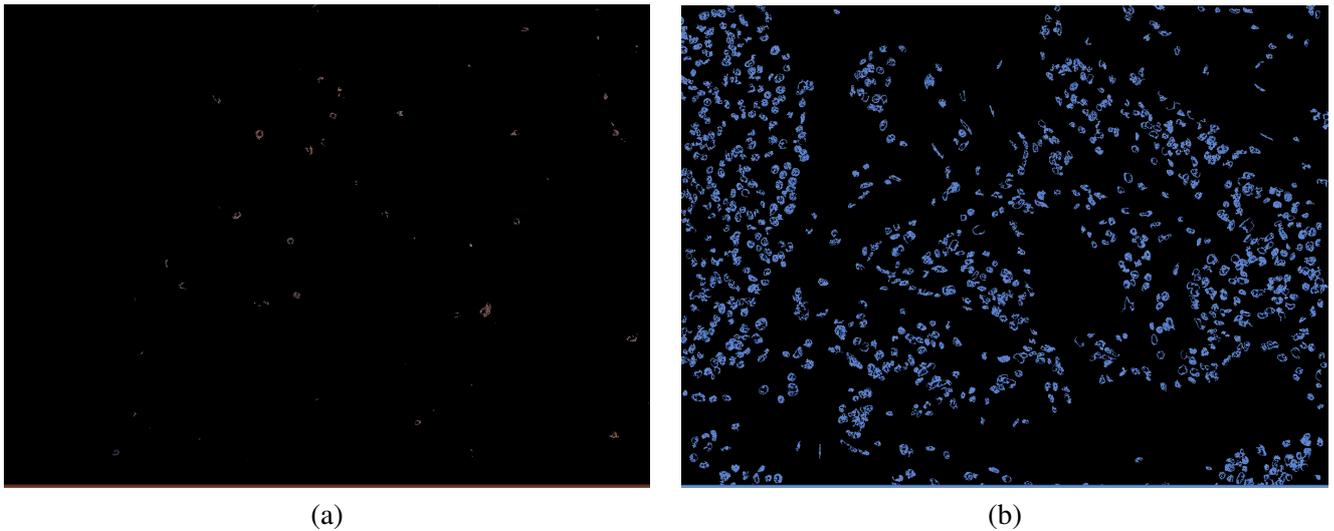


Figure 14: The HNCut provided segmentation version of the previous figure. Panel (a) illustrates where the lymphocytes are located, while (b) locates the cancer cells.

the form of a prognostic indicator, with the larger goal of helping oncologists make more informed decisions for treatment.

References

- [1] H. Rui and M.J. LeBaron. Creating tissue microarrays by cutting-edge matrix assembly. *Expert Rev Med Devices*, 2(6):673–680, Nov 2005. 2
- [2] H. Vrolijk, W. Sloos, W. Mesker, P. Franken, R. Fodde, H. Morreau, and H. Tanke. Automated acquisition of stained tissue microarrays for high-throughput evaluation of molecular targets. *Journal Of Molecular Diagnostics*, 5(3), 2003. 2, 4, 5
- [3] S. Ronald and S. Guido. Tissue microarray (tma) applications: implications for molecular medicine. *Expert reviews in molecular medicine*, 5(26):1–12, 2003. 4
- [4] R. Buckanovich, D. Sasaroli, and et al. Tumor vascular proteins as biomarkers in ovarian cancer. *Journal Of Clinical Oncology*, pages 852–861, March 2007. 5
- [5] J. Wu, J. Dong, and H. Zhou. Image quantification of high-throughput tissue microarray. In A. Manduca and A. A. Amini, editors, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6143 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 509–520, March 2006. 5

- [6] Andrew Rabinovich, S. Krajewski, M. Krajewska, A. Shabaik, S. M. Hewitt, Serge J. Belongie, J. C. Reed, and Jeffrey H. Price. Framework for parsing, visualizing and scoring tissue microarray images. *IEEE Transactions on Information Technology in Biomedicine*, 10(2):209–219, 2006. 5
- [7] W. Zhong, G. Altun, R. Harrison, P.C. Tai, and Y. Pan. Improved k-means clustering algorithm for exploring local protein sequence motifs representing common structural property. *NanoBioscience, IEEE Transactions on*, 4(3):255–265, Sept. 2005. 5
- [8] Gustavo Carneiro, Bogdan Georgescu, Sara Good, and Dorin Comaniciu. Detection and measurement of fetal anatomies from ultrasound images using a constrained probabilistic boosting tree. *IEEE Trans. Med. Imaging*, 27(9):1342–1355, 2008. 5
- [9] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32–40, Jan 1975. 6
- [10] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Trans, PAMI*, 24(5):603–619, May 2002. 6
- [11] Yizong Cheng. Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(8):790–799, Aug. 1995. 6, 9
- [12] C. Yang, R. Duraiswami, N.A. Gumerov, and L. Davis. Improved fast gauss transform and efficient kernel density estimation. *IEEE ICCV*, 1:664–671, Oct. 2003. 6, 9, 14
- [13] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(11):1101–1113, Nov. 1993. 6
- [14] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. PAMI*, 22(8):888–905, Aug. 2000. 6, 11
- [15] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Trans. PAMI*, 29(11):1944–1957, 2007. 6
- [16] Yuzhong Wang, Jie Yang, and Ningsong Peng. Unsupervised color-texture segmentation based on soft criterion with adaptive mean-shift clustering. *Pattern Recogn. Lett.*, 27(5):386–392, 2006. 6, 7
- [17] Zhuowen Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1589–1596, Washington, DC, USA, 2005. IEEE Computer Society. 13, 14

- [18] B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall, 1986. 14
- [19] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, January 1979. 20
- [20] S. Doyle, A. Madabhushi, M. Feldman, and J. Tomaszewski. A boosting cascade for automated detection of prostate cancer from digitized histology. In *MICCAI (2)*, pages 504–511, 2006.
- [21] James Monaco, John E. Tomaszewski, Michael D. Feldman, Mehdi Moradi, Parvin Mousav, Alexander Boag, Chris Davidson, Purang Abolmaesumi, and Anant Madabhushi. Probabilistic pairwise markov models: Application to prostate cancer detection. *Society of Photo-Optical Instrumentation Engineers (SPIE) 2009*.