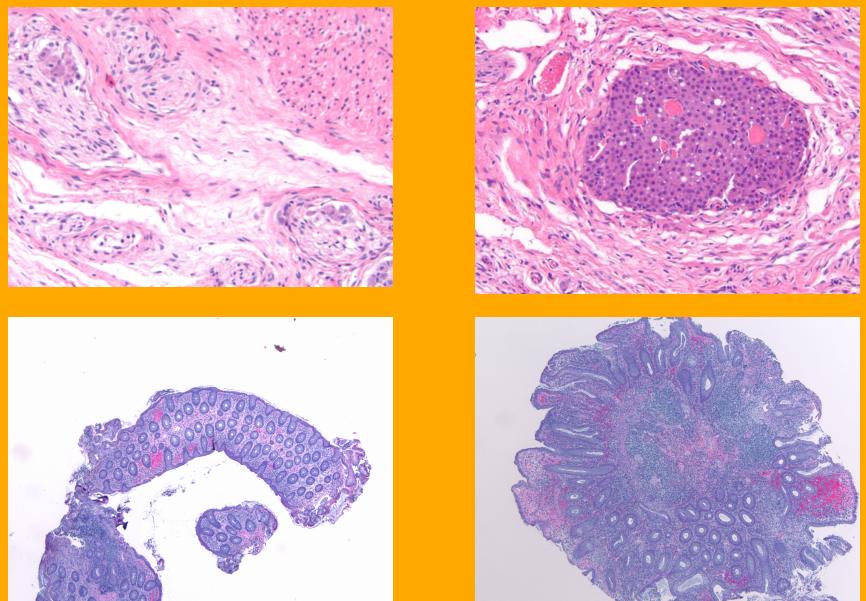


Tools for Automated Histology Image Analysis



Michael T. McCann

bimagicLab
Dept. of Biomedical Engineering
Carnegie Mellon University

Tools for Automated Histology Image Analysis

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the
Department of Biomedical Engineering

Michael T. McCann

B.S.E., Biomedical Engineering, University of Michigan, Ann Arbor, MI

Carnegie Mellon University
Pittsburgh, PA

May 2015

Acknowledgments

These last five years, I have been lucky enough to have a crowd of people supporting my research and my happiness. You are impossible to sort as “collaborators” or “friends,” since so many of you have been both.

I first need to thank my adviser, Jelena Kovačević. Jelena, you have taught me immeasurably over the years: “How to write math,” “How to write a technical paper,” “How to do a presentation,” “How to give a lecture,” and much more. At the same time, you have served as the perfect academic role model with your hard work and good cheer. Finally, your whole-hearted inclusion of all your students into your academic family has always impressed me; I will always strive to be so magnanimous.

Thank you to the rest of my thesis committee, José Moura, Ge Yang, and John Ozolek. Your feedback has shaped both this work and the way I approach engineering problems.

Thanks to my labmates, Ramu Bhagavatula, Anupama Kuruvilla, Jackie Chen, Siheng Chen, Filipe Condessa, Rohan Varma, Anuva Kulkarni, and the rest of CBI (especially my podmates, Soheil Kolouri and Shinjini Kundu). You have taught me much and contributed to every piece of this work through your constant feedback at lab meetings and beyond. Ramu, thanks for introducing me to these fascinating images and for showing me that a Ph.D. could be finished when I was still just starting. Anu, thanks for your model working hours, your willingness to make reservations, and for sharing my occasional gripes.

I have collaborated with so many people on this work. Thanks to those I advised: Leah Yingling, Anish Menon, Cheng Peng, Joshita Majumdar, Savi Medlang; this is a much bigger thesis because of you. And to those who advised me: Matt Fickus, John Ozolek, Carlos Castro, Bahram Parvin; I always learned a lot by talking with you and regret not being able to spend more time soaking up that knowledge. Matt, thanks for all those dinners and for showing me the sights in Ohio (Jungle Jim’s and giant Jesus should be a bimagicLab rite of passage). John and Carlos, given how insanely busy you both are, I was repeatedly amazed by your enthusiasm for this project, thanks!

Thanks to all the CMU staff who have guided me through life as a student with patience. Ryan Ries, you taught me to send a fax and much else. Christina Cowan, thanks for the snacks and cheerful conversation.

Angela Kang, Milda Zizyte, Felix Hutchison, and Will Levine, you have been great friends for these years, perhaps sometimes better than I have deserved. Thanks! Yomei Shaw, you make me very happy, and your diligence on your thesis these past months has greatly improved my diligence on mine. Thank you.

Jane Huggins (“The creator of the HF-difference!”), Dave Thompson, and the rest of the folks at the UM-DBI project: working with you was my introduction to the joys of research, thanks for all the faith and effort you put into me. Dave, I always enjoy your Christmas letters, no matter how late they come.

Jim McCann, you have been my academic template. Thanks for blazing the trail, big brother. John and Peg McCann, you have certainly put more work into me than anyone else. Thank you for your efforts, and thank you for instilling me with your creativity and curiosity. Dad, you got me into the fun of computers and algorithmic thinking. Mom, it is often in my life as an adult that I discover some new chore that you transparently handled for me for years. You are both impossible to thank enough.

Finally, I want to thank my funding sources, including the ARCS foundation (women of ARCS, you taught me how to write a thank you note), the NSF (through awards 0946825 and 1017278), the John and Claire Bertucci Graduate Fellowship, Philip and Marsha Dowd Teaching Fellowship, and the CMU Carnegie Institute of Technology Infrastructure Award.

Abstract

In this thesis, we present three image processing tools inspired by and designed for histology image analysis. Histology, which is the examination of biological tissue under a microscope, is a critical technique in biomedical research and clinical practice. While slide preparation and imaging is increasingly becoming automated, the analysis of the resulting histology images is not: even routine analyses still require the trained eyes of a pathologist. In our work, we aim to understand histology images as a class of signals and develop tools to aid in the automated analysis of these signals. Our first contribution is in the area of histology image normalization, where the goal is to digitally remove the variation in staining between histology images, an important preprocessing step in many histology image analysis systems. To this end, we created a new benchmark dataset with which to compare normalization methods and proposed our own. Our second contribution is a tissue segmentation method, which delineates single-tissue regions in histology images. Along with this method, we propose a new mathematical model for histology images. Our final contribution is a method for describing distributions of angles, which is useful for segmentation as well as a variety of other image processing tasks.

Contents

Acknowledgments	ii
Abstract	iii
List of Abbreviations	vii
List of Tables	viii
List of Figures	xii
1 Introduction	1
1.1 Thesis Contributions and Outline	2
1.2 List of Publications	2
I Background	4
2 Histology	5
2.1 Tissue Collection	5
2.2 Processing	5
2.3 Embedding	6
2.4 Sectioning	6
2.5 Staining	7
2.6 Visualization	8
2.7 Analysis	8
2.7.1 Clinical Practice	8
2.7.2 Research	9
2.8 Sources of Variability	10
3 Automated Histology Image Analysis	13
3.1 Preprocessing	14
3.2 Registration	15
3.2.1 Our Previous Work	16
3.3 Display and Annotation	16
3.3.1 Our Previous Work	16
3.4 Region of Interest Identification	18
3.5 Nucleus Detection	18
3.5.1 Our Previous Work	19
3.6 Cell Classification	20
3.7 Multicellular Structure Detection	20
3.8 Tissue Segmentation	21

3.9 Diagnosis	21
3.9.1 Our Previous Work	22
II Normalization and Segmentation	25
4 Normalization	26
4.1 Introduction	26
4.1.1 Previous Work	26
4.1.2 Current Work	27
4.2 Benchmark Dataset	27
4.3 Methods	28
4.3.1 Wedge Finding	28
4.3.2 Initial Explorations	29
4.3.3 Proposed Method	29
4.3.4 Stain Normalization	31
4.4 Experiments and Discussion	31
4.4.1 Stain Separation Comparison	32
4.4.2 Parameter Analysis	32
4.4.3 Stain Normalization Comparison	33
4.4.4 Normalization and Diagnosis	33
4.5 Conclusions and Future Work	36
5 Tissue Segmentation: ORTSEG	37
5.1 Introduction	37
5.1.1 Previous Work	37
5.1.2 Current Work	38
5.2 Mathematical Framework	39
5.2.1 Notation and Background	39
5.2.2 Flat Textures	39
5.2.3 Histograms of Flat Textures	40
5.2.4 Constructing Dependence-Decay Textures	40
5.2.5 Dependence Decay in Real Images	41
5.2.6 Occlusions of Dependence-Decay Textures	42
5.3 Proposed Algorithm	44
5.3.1 Local Histogram Transform	46
5.3.2 Factorization	47
5.3.3 Deconvolution	47
5.3.4 Selecting the Number of Textures	48
5.4 Experiments and Discussion	48
5.4.1 Datasets	48
5.4.2 Algorithms Tested	49
5.4.3 Experimental Setup	50
5.4.4 Discussion	50
5.5 Conclusions and Future Work	52
Appendices	55
5.A Proof of Theorem 1	55
5.B Proof of Theorem 2	57
5.C Variation of Information Results	59

6 Describing Angular Distributions with the FS-KDE	60
6.1 Introduction	60
6.1.1 Previous Work	60
6.1.2 Current Work	61
6.2 The Fourier Series Kernel Density Estimate	61
6.2.1 Derivation of the FS-KDE	61
6.2.2 Properties	63
6.2.3 Image Version	63
6.2.4 Canonicalization	64
6.2.5 Practical Considerations	66
6.3 Experiments and Discussion	70
6.3.1 Keypoint Description	70
6.3.2 Person Detection	73
6.3.3 Texture Segmentation	76
6.4 Conclusions	78
Appendices	80
6.A Proof of Theorem 3	80
7 Conclusions and Future Work	81

List of Abbreviations

CT	Computed tomography
DZI	Deep zoom image
EDISON	EDISON segmentation system [1] (see Section 5.4.2)
Efficient	Efficient Graph-Based Segmentation [2] (see Section 5.4.2)
E-only	Eosin-only
FS-KDE	Fourier series kernel density estimate
gPb-owt-ucm	gPb-owt-ucm segmentation system [3] (see Section 5.4.2)
HOG	Histogram of oriented gradients [4]
H-only	Hematoxylin-only
H&E	Hematoxylin and eosin
IHC	Immunohistochemical
JSEG	JSEG segmentation system [5] (see Section 5.4.2)
KDE	Kernel density estimate
MIL	Multiple instance learning
MR	Magnetic resonance
MRC	Multiresolution classification system [6]
MRI	Magnetic resonance imaging
NMF	Non-negative matrix factorization
Normalized	Normalized cut [7] (see Section 5.4.2)
OD	Optical density
ORTSEG	Occlusion of random texture segmenter
ORTSEG-D	Occlusion of random texture segmenter with deconvolution
PLC	Pixel-level classifier
PLC-MIL	Pixel-level classifier with multiple instance learning
ROI	Region of interest
SNR	Signal to noise ratio
SVD	Singular value decomposition
SVM	Support vector machine
WND-CHARM	WND-CHARM biomedical image classifier [8] (see Section 3.9.1)
WSI	Whole slide image

List of Tables

3.1	Colitis feature set.	22
3.2	Comparison of classification accuracy (in %), with the best accuracy for each magnification in bold.	24
4.1	Results of the stain separation comparison reported as SNR [dB] for hematoxylin and eosin, with the best result for each image and stain in bold. The proposed method improves SNR by around one dB.	32
4.2	Comparison of classification accuracy (in %) of the WND-CHARM classifier on raw, illumination-corrected, stain-varied, and stain-normalized images. The best accuracy for each magnification is shown in bold. The stain-normalized dataset gives better results than the stain-varied dataset, suggesting that staining variability does negatively impact computer-aided diagnosis.	36
5.1	Results of the segmentation comparison (Section 5.4) in terms of the Rand index mean and standard deviation, with the best result for each dataset in bold. The value one indicates perfect agreement with the ground truth.	49
5.C.1	Results of the segmentation comparison (Section 5.4) in terms of the variation of information mean and standard deviation, with the best result for each dataset in bold. The value 0 indicates perfect agreement with the ground truth.	59
6.1	Comparison of the basic ORTSEG method with versions using gradient histograms and FS-KDEs in terms of Rand index mean and standard deviation. The augmented versions improve the performance on the dead leaves dataset, where edge information is critical.	79

List of Figures

1.1	(a) A natural image of an office scene. (b) A histology image of a teratoma. In natural images, objects appear at different depths and may block each other or be out of focus. In histology images, the tissue is at a single depth. Compared to natural images, histology images are relatively smooth.	2
2.1	Block diagram of the histology process.	5
2.2	A large piece of tissue must be cut into smaller pieces and placed into cassettes for further processing.	6
2.3	(a) Paraffinized tissue is oriented and embedded into a block of paraffin. (b) Tissue blocks after embedding.	7
2.4	Importance of proper orientation during embedding. (a) Gross photograph showing an opened enteric duplication cyst (arrowhead) intimately joined to the bowel (arrow). (b) Histology image of the cyst with normal bowel (arrow) abutting cyst wall (arrowhead). This can only be observed when sectioning occurs perpendicular to the cyst wall.	7
2.5	(a) During sectioning, the tissue block is sliced into 3 to 4 μm sections, which remain connected in a ribbon. (b) Collecting a ribbon from the water bath with a glass slide.	8
2.6	(a) A staining machine which moves slides through a series of stains and rinses. (b) A single finished slide; the purple/pink color is caused by the H&E stain. (c) A set of finished slides in a slide tray.	9
2.7	(a) Visualization setup including a microscope with attached camera and desktop computer. (b) WSI of a teratoma collected as a set of tiles and stitched together into a large image.	10
2.8	Examples of important visual cues in histology. (a) In this teratoma section, color makes distinguishing cartilage (blue/gray, arrow) from bone (pink/red, arrowhead) easy, even at low magnification. (b) In this liver section of a child with mitochondrial disorder, texture is important. Under high magnification, cell borders are accentuated and cytoplasm shows tiny red granules representing abnormal mitochondria (arrow).	10
2.9	Examples of the importance of nuclear shape and distribution in colon screening. (a) In normal colon tissue, nuclei are regular in size and distribution around the colon gland (arrow). (b) In colitis, nuclei become much less regular (arrow).	11
2.10	Example of staining variability. Both images are of bone, but the color of the bone varies from pink to purple due to staining variability. Such color variations present a challenge for automated analysis systems.	12
3.1	Block diagram of a generic automated histology analysis system.	13
3.2	(a and b) Two adjacent sections from a teratoma tumor registered via a rigid transform. (c) A visualization of their difference (pink and green represent disagreement, gray indicates agreement). The tumor is about 1cm in diameter. At this scale, the agreement between the two images is good.	17
3.3	(a-c) Zoomed versions the images in Figure 3.2. At this scale, inaccuracies in the registration are apparent, indicating the need for non-rigid registration.	18

3.4	Comparison of nucleus detection methods, with black dots marking detections. (a) Using simple maxima detection. (b) Using moment filter-based detection. The moment-filter approach rejects the two objects at the center-left because they are overly oblong.	20
3.5	Example pixel-level classification results for PLC-MIL (100 \times). These results are physiologically reasonable: inflammation (black) is detected in regions of high nuclei density and not in crypts or in areas of hemorrhage, which are generally artifacts of the biopsy procedure.	24
4.1	Example stain separation showing (a) an input image of an H&E-stained slide, (b) the ground truth stain separation, and the digital stain separation results from (c) color inversion [9], (d) wedge-finding [10], and (e) our proposed method. The left half of each image shows the H-only estimate and the right half shows the E-only estimate.	27
4.2	Images collected during benchmark creation.	28
4.3	Contour plots indicating pixel density as a function of position along u_0 and u_1 , the first two singular vectors of the matrix of pixel colors for (a) H-only and (b) E-only images. The highest density of points for both stains is at the origin, which corresponds to white. If the stains were truly linear, the pixel matrix would be rank one and therefore all pixels would lie only along u_0 . For both hematoxylin and eosin this is not the case, indicating that neither stain is completely linear, however the deviation is more significant for hematoxylin.	30
4.4	Contour plots indicating pixel density as a function of position along u_0 and u_1 , the first two singular vectors of the matrix of pixels in the H&E image. Each pixel in the H&E image should be the sum of a pixel from the H-only and E-only image; however some of the density of the H&E image lies in the lower-right corner of the plot, which cannot be formed in this way.	30
4.5	Effects of wedge percentile α and eosin contrast reduction γ (solid, dashed, dotted lines corresponding to $\gamma = 0, 0.5, 1$) on (a) the H-only reconstruction, (b) the E-only reconstruction, and (c) their mean. When $\gamma = 0$, our method reduces to wedge finding [10]; setting γ to 0.5 improves both the H-only and E-only reconstruction. Setting γ to 1 improves the E-only reconstruction at the expense of the H-only reconstruction.	33
4.6	Effects of interaction amount λ for wedge percentile $\alpha = 0.3$ and eosin contrast reduction $\gamma = 1$. A wide range of λ values all improve the SNR of the H-only estimate.	34
4.7	Results of the stain normalization experiment, showing (a) the input, (b) the target, and the results using (c) color inversion [9], (d) wedge-finding [10], and (e) our proposed method. The results are similar across methods; the main difference is the depth of color in the bone tissue (long purple structure in (c)-(e), best viewed in color).	34
4.8	Examples of (a) the original colitis dataset images, (b) their illumination-corrected versions, (c) their stain-varied versions, and (d) their stain-normalized versions.	35
5.1	Examples of difficult tissue boundaries in teratoma images (black lines denote tissue boundaries as drawn by a pathologist). The lack of distinct edges between regions makes automated segmentation difficult, and even experts cannot reliably localize the boundaries at pixel level.	38
5.2	Examples of dead leaves images with different maximum template sizes and value distributions. The templates in (a) and (b) have twice the maximum diameter of those in (c). The value distributions in (a) and (c) are uniform, while the value distribution in (b) is normal.	41
5.3	(a) Example texture from the Prague set [11]. (b) Cropped and quantized version used in our dependence decay analysis.	42
5.4	Dependence decay for the image in Figure 5.3. The plot in (a) shows how the approximation error (5.4) (solid line) is bounded by the function $2.42 k^{-1/2}$ (dashed line). The regions shown in (b) are four worst-case arrangements (described in Section 5.2.5). When pixels sampled in these patterns are used to estimate the color distribution of the texture, they lead to errors of $2^{-1}, 2^{-2}, 2^{-3}$, and, 2^{-4} (decreasing error from black to light gray). The corresponding points are marked with circles in (a).	43

5.5	(a) An example of a periodic texture from the Prague set [11]. (b) Cropped and quantized version used in our dependence decay analysis. (c) Four worst-case arrangements (described in Section 5.2.5), increasing in size from black to light gray. Sampling pixels in these vertical stripes leads to very poor estimates of the true color distribution of this texture because the sample may, e.g., contain only black pixels, omitting the lighter pixels in the image.	44
5.6	Modeling an image as an occlusion of realizations of textures according to a labeling function.	45
5.7	Block diagram of the proposed algorithm.	46
5.8	Alternating least-squares method for non-negative matrix factorization.	47
5.9	Example segmentation results from ORTSEG and ORTSEG-D on a synthetic texture; Rand indexes are given in parentheses. The parametric deconvolution in ORTSEG-D forces region boundaries to be linear, improving the segmentation quality.	51
5.10	Example segmentation results for (c)-(g) the comparison methods and (h) our method on the random texture dataset; Rand indexes are given in parentheses. This dataset is challenging due its lack of meaningful edges.	52
5.11	Example segmentation results for (c)-(g) the comparison methods and (h) our method on the histology dataset; Rand indexes are given in parentheses. This is a real-world example of a dataset in which the lack of edges is challenging for many segmentation methods.	53
5.12	Example segmentation results for (c)-(g) the comparison methods and (h) our method on the Prague dataset; Rand indexes are given in parentheses. This dataset simulates segmentation of natural images. Our method is not well suited to such images because it by design does not make use of edges as evidence of boundaries.	54
6.1	Top row: A weighted set of angles, its 20-bin histogram, and the estimate based on the proposed method (FS-KDE) using 20 numbers. Bottom row: The same set of angles rotated counter-clockwise by 45° , its histogram, and FS-KDE. While the rotation distorts the histogram, it only causes a corresponding rotation in the FS-KDE.	61
6.1	Examples of the powered cosine kernel (6.2) for different orders, K . Each kernel is 2π -periodic and integrates to one. As K increases, the kernels become sharper.	62
6.2	(a) A distribution, f . (b) Two of its noisy versions, f^ϵ . (c) The distance between f and its noisy versions plotted as a function of increasing noise. (d) The additional error introduced by F_1 canonicalization of the noisy versions, along with the expectation from Theorem 3. Because this distribution has a large $ F_1 $ value compared to the noise, F_1 canonicalization makes small changes to noisy versions of f . (e) Curves indicate $\ F - F_\phi\ $ (bold line) and $\ F - F_\phi^\epsilon\ $ (thin lines) for f , two noisy versions of f , and their rotations by ϕ . Because $ F_1 $ is large compared to the noise, $\ F - \tilde{F}_\phi^\epsilon\ $ (horizontal lines) is almost as small as $\min_\phi \ F - F_\phi^\epsilon\ $	67
6.3	(a) A distribution, f . (b) Two of its noisy versions, f^ϵ . (c) The distance between f and its noisy versions plotted as a function of increasing noise. (d) The additional error introduced by F_1 canonicalization of the noisy versions, along with the expectation from Theorem 3. Because this distribution has a small $ F_1 $ value compared to the noise, F_1 canonicalization may make large changes to noisy versions of f . (e) Curves indicate $\ F - F_\phi\ $ (bold line) and $\ F - F_\phi^\epsilon\ $ (thin lines) for f , two noisy versions of f , and their rotations by ϕ . Because $ F_1 $ is small compared to the noise, $\ F - \tilde{F}_\phi^\epsilon\ $ (horizontal lines) is sometimes large.	68
6.4	Examples of the powered cosine kernel of order 64 with different levels of truncation, where ℓ is the number of non-zero coefficients. Distortion is barely noticeable even when three quarters of the F_k s are set to zero (upper-right panel).	70
6.1	Examples of (a) corresponding and (b) non-corresponding pairs of patches from the British Columbia Multi-view Stereo Correspondence Dataset [12]. Though the patches are rotated to a canonical orientation, corresponding patches still exhibit viewpoint and intensity variation.	71
6.2	Distribution of distances between descriptors for corresponding and non-corresponding patches for the FS-KDE patch descriptor on 500,000 pairs of patches from the University of British Columbia Multi-view Stereo Correspondence Dataset [12]. Lines represent 50-bin histograms.	72

6.3	Estimated chance of a patch matching error (calculated via (6.12)) for intensity, gradient histogram, and FS-KDE descriptors of varying size. The best performance (lowest error chance) is achieved by the FS-KDE descriptor of size ten. The canonical descriptors perform poorly in this experiment because the patches are already in a canonical orientation.	73
6.4	Estimated chance of a patch matching error for intensity, gradient histogram, and FS-KDE descriptors of varying size on patches with random orientation. The FS-KDE and gradient histogram descriptors cannot handle patch rotations, so have much higher error chance than in Figure 6.3. The best performance (lowest error chance) is now achieved by the FS-KDE descriptor with F_k canonicalization.	74
6.5	Accuracy on a person detection task for the intensity, gradient histogram, HOG, and FS-KDE feature extractors, plotted as a function of the feature vector length per image block. The top set of lines is for the INRIA person dataset [4], the bottom set is for the same dataset with small random rotations added. Without rotations, HOG features have the highest accuracy for most lengths and the intensity features (as expected) have the lowest. With rotations, the performance of all four methods declines, but the FS-KDE declines the least, leaving it with accuracy comparable to HOG. The accuracy of the intensity features for the rotated dataset was below chance and not plotted.	75
6.6	An example image and corresponding ground truth from the dead leaves dataset. For images like these, angular distributions are an important feature.	76
6.7	Example segmentation results for (c) the basic, (d) histogram, and (e) FS-KDE versions of ORTSEG on the random texture dataset. Rand indexes are given in parentheses. Here, color is enough to distinguish the textures and therefore all three methods perform well.	77
6.8	Example segmentation results for (c) the basic, (d) histogram, and (e) FS-KDE versions of ORTSEG on the dead leaves dataset. Rand indexes are given in parentheses. Here, color is not enough to distinguish the textures, so the basic version of ORTSEG fails.	78

Chapter 1

Introduction

Histology, which is the examination of biological tissues using a microscope, is a crucial and ubiquitous technique in medicine, serving as the gold standard diagnostic technique for many diseases. The 2010 Nation Hospital Discharge Survey [13] estimates that, of the 40 million hospital stays ending in 2010 in the United States, 1.5 million involved histology, as compared to only 250 thousand involving magnetic resonance imaging (MRI). Histology also serves as a tool in a wide range of research applications, such as studying cancer and embryogenesis.

Tissue processing for histology has become increasingly automated, drastically increasing the speed at which histology labs can produce tissue slides for viewing. Another trend is the digitization of these slides, allowing them to be viewed on a computer rather than through a microscope. Despite these changes, much of the routine analysis of tissue sections remains a painstaking, manual task that can only be completed by highly trained pathologists at a high cost per hour. There is, therefore, a niche for image analysis methods that can automate some aspects of this analysis. In the clinical domain, these methods could improve the accuracy and consistency of diagnoses and free pathologists to focus on the most difficult cases. In the research domain, these methods could complete tasks that are prohibitively time-consuming for humans, e.g., discovering new disease markers from hundreds of whole-slide images (WSIs) or precisely delineating tissues within a tumor, allowing for quantitative comparison of tumors grown under different conditions.

Given that whole-slide scanners entered the market in 2000 [14] while computer vision was (famously and over-optimistically) proposed as a summer project at MIT in 1966 [15], one might expect that the mature methods of computer vision could immediately solve the new problems arising in automated histology image analysis. This is not the case. One reason is that computer vision is difficult and there are many problems left to solve in the field, another is that histology images differ from natural images (Figure 1.1). To point out a few of many differences: generally speaking, histology images are at a single depth and thus they are always in focus and lack physical occlusions; they represent a slice of a volume of tissue; the colors in the images are created by stains engineered to give contrast to specific structures; the illumination is even; and the tissue being imaged is highly ordered. Edges are a critical cue in natural images because they often mark boundaries between objects; this is not the case in histology images, where the transitions between tissues is usually gradual.

Given the differences between histology images and natural images, our approach has been to study histology images as a unique class of images and to design image processing tools tailored to these images.

The long-term goal of our work is to develop a suite of image processing tools well suited to images of histology samples stained with hematoxylin and eosin (H&E).

We focus on this specific stain because (1) the H&E slide is the universally accepted means to examine tissue sections, (2) the vast majority of diagnoses rendered by pathologists are based solely on H&E slides without ancillary tissue testing, (3) the processing techniques are universal and standardized, and (4) the

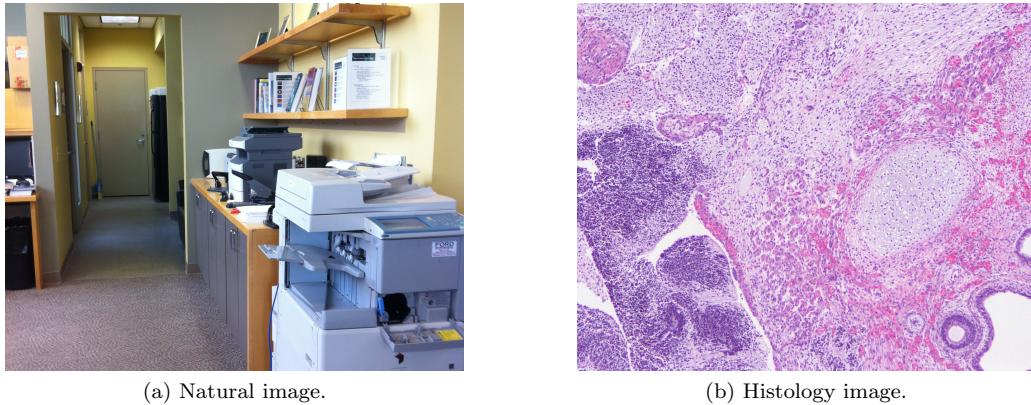


Figure 1.1: (a) A natural image of an office scene. (b) A histology image of a teratoma. In natural images, objects appear at different depths and may block each other or be out of focus. In histology images, the tissue is at a single depth. Compared to natural images, histology images are relatively smooth.

cost is low. The tools we develop will facilitate the creation of automated histology analysis systems for a plethora of tasks, increasing pathologists' efficiency, reducing cost, improving consistency of diagnosis, and enabling previously impossible studies. The new image processing tools may also prove useful outside of histology image processing, e.g. in the analysis of satellite images, which share many similarities.

1.1 Thesis Contributions and Outline

The main contributions of this thesis are

1. an algorithm and benchmark dataset for stain separation and normalization of H&E histology images (Chapter 4 and [16]);
2. a mathematical framework and algorithm for histology image segmentation based on color distributions (Chapter 5 and [17]); and
3. a rotation invariant descriptor for angular distributions (Chapter 6 and [18], submitted).

The outline of the thesis is as follows. In Part I, we give important background information on histology (Chapter 2) and automated histology systems (Chapter 3), including a literature review of current work in the field. In Part II, we present our work on histology image normalization (Chapter 4) and segmentation (Chapter 5). The discussion of segmentation leads to our work on a rotation invariant descriptor for angular distributions (Chapter 6). We conclude in Chapter 7.

1.2 List of Publications

Reviewed journal publications.

1. M. T. McCann, M. C. Fickus, and J. Kovačević, "Fourier series of kernel density estimates for rotation invariant angular distribution estimation," *IEEE Trans. Image Process.*, 2015, submitted.
2. M. T. McCann, J. A. Ozolek, C. A. Castro, B. Parvin, and J. Kovačević, "Automated histology analysis: Opportunities for signal processing," *IEEE Signal Process. Mag.*, vol. 32, no. 1, pp. 78-87, Jan. 2015.

3. R. Bhagavatula, M. T. McCann, M. Fickus, C. A. Castro, J. A. Ozolek, and J. Kovačević, “A vocabulary for the identification and delineation of teratoma tissue components in hematoxylin and eosin-stained samples,” *J. Path. Inform.*, vol. 5, p. 19, Jun. 2014.
4. M. T. McCann, D. G. Mixon, M. C. Fickus, C. A. Castro, J. A. Ozolek, and J. Kovačević, “Images as occlusions of textures: A framework for segmentation,” *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2033-2046, May 2014.

Reviewed conference publications.

1. M. T. McCann, J. Majumdar, C. Peng, C. A. Castro, and J. Kovačević, “Algorithm and benchmark dataset for stain separation in histology images,” in *Proc. IEEE Int. Conf. Image Process.*, Paris, Oct. 2014, pp. 3953-3957.
2. M. T. McCann, R. Bhagavatula, M. C. Fickus, J. A. Ozolek, and J. Kovačević, “Automated colitis detection from endoscopic biopsies as a tissue screening tool in diagnostic pathology,” in *Proc. IEEE Int. Conf. Image Process.*, Orlando, FL, Sep. 2012, pp. 2809-2812.

Part I

Background

Chapter 2

Histology

In this chapter, we present background information on histology. We describe how tissue is collected and processed as well as typical analysis tasks in both the clinical and research domains. Throughout, we focus on slides stained with the ubiquitous H&E stain and imaged with brightfield microscopy. This chapter is an expanded version of Section 2 of our histology review paper [19] and is based on standard histology texts [20, 21], as well as the years of experience in the field of pathology of Drs. John Ozolek and Carlos Castro.

Figure 2.1 presents the histology process from tissue processing to diagnosis from the perspective of the diagnostic pathologist. In the following sections, we explain each of the steps in this process. We conclude with a section on the sources of variability in histology images.

2.1 Tissue Collection

The clinical histology process begins when the treating physician, after assessing the patient by history, physical examination, and/or radiographic and laboratory studies, determines that treatment can proceed no further without histology confirmation. The treating physician then must obtain enough good-quality tissue to obtain a diagnosis. There are several possible approaches to tissue collection, including fine-needle aspiration, needle biopsy, excisional biopsy, or excision of the lesion in its entirety. The sensitivity (likelihood of getting the correct diagnosis) and specificity (likelihood of not getting the incorrect diagnosis) increase from fine-needle aspiration to excision of the entire lesion. This is because the larger biopsies preserve more cellular context and allow the pathologist to examine multiple slides from different areas of the sample.

After biopsy, the pathologist evaluates the tissue on the macroscopic scale, measuring it and recording a description of its color and characteristics. For larger tissues (e.g., tumor resections or colon resections), the tissue must be trimmed to fit into the tissue cassettes (approximately $10 \times 10 \times 3$ mm) that will contain it for the subsequent processing steps (Figure 2.2).

2.2 Processing

The next step in the histology process is tissue processing, which involves chemically and physically stabilizing the tissue. The tissue is first immersed into a fixative solution that is used to stop cells from breaking down



Figure 2.1: Block diagram of the histology process.



(a) Large tissue piece.



(b) Small tissue pieces in cassettes.

Figure 2.2: A large piece of tissue must be cut into smaller pieces and placed into cassettes for further processing.

and prevent microorganism growth. In general, tissue is fixed for a few hours (small biopsies) to about 24 hours (large biopsies). Fixation is critical because poorly fixed tissue leads to poor tissue sectioning and poor microscopic morphology.

After fixation, the tissue is physically stabilized by one of several methods (freeze drying, microwave, chemical) with the end goal of preserving the cellular morphology. The most commonly employed method involves the use of alcohols and xylene and is automated in most laboratories: First, the tissue is dehydrated, which clears the water and aqueous fixative from it. Next, the tissue is cleared of the dehydrating agent, leaving the tissue ready for paraffin infiltration. Finally, the paraffin warms in the processor until it is liquefied, infiltrates the tissue under vacuum, and then cools so that the tissue becomes firm. This process takes approximately nine hours and in many laboratories is run overnight. One of the end results of processing tissue in this manner is that the tissue section is slightly smaller than the original fresh or fixed tissue prior to processing.

2.3 Embedding

After processing, the tissue is embedded in a block of support material, as shown in Figure 2.3. To achieve this, the tissue is placed on the bottom of a mold and paraffin is poured over it. The original tissue cassette is placed over the mold and then placed onto a cooling plate to solidify the paraffin. The result is a *tissue block*, tissue that is impregnated with and surrounded by hardened paraffin. Because the tissue will be sliced parallel to the cassette, orientation of the tissue during embedding is key; see Figure 2.4 for an example.

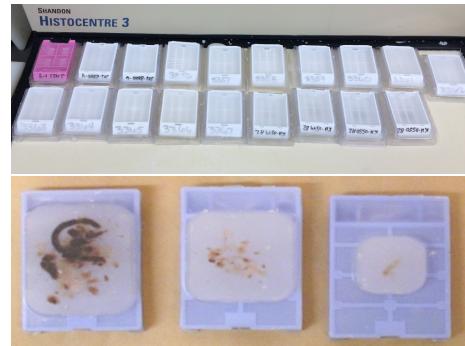
2.4 Sectioning

Sectioning is cutting thin slices of tissue and mounting them on microscope slides (Figure 2.5). The cutting is achieved with a tool called a microtome, which operates like a deli slicer. This step can be manual, semi-automated, or automated. For example, the manual rotary microtome advances the block by a set amount with each turn of the wheel and thus produces a ribbon of tissue. Most tissue sections for diagnostic purposes are cut at 3 to 4 μm thickness. Thinner sections are more difficult to obtain without damaging the tissue, while thicker sections tend to make staining dark and obscure nuclear detail.

These thinly cut sections are floated out onto a water bath heated to about 10 °C below the melting point of the paraffin. This allows dispersion of any wrinkles generated by the microtome blade at sectioning. Once the sections are cut and floated, they are placed onto 25 × 75 × 1 mm glass microscope slides.

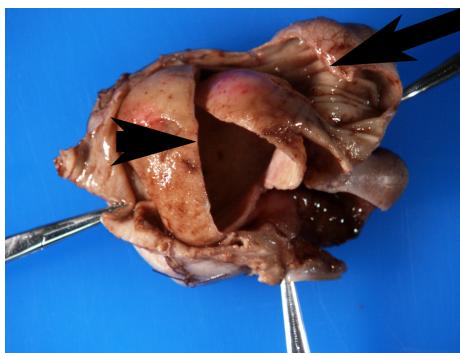


(a) Embedding.

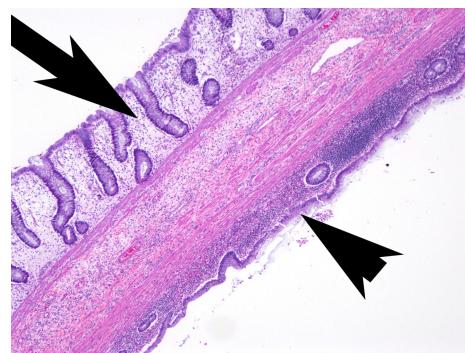


(b) Tissue blocks after embedding.

Figure 2.3: (a) Paraffinized tissue is oriented and embedded into a block of paraffin. (b) Tissue blocks after embedding.



(a) Photograph of cyst.



(b) Histology image of cyst.

Figure 2.4: Importance of proper orientation during embedding. (a) Gross photograph showing an opened enteric duplication cyst (arrowhead) intimately joined to the bowel (arrow). (b) Histology image of the cyst with normal bowel (arrow) abutting cyst wall (arrowhead). This can only be observed when sectioning occurs perpendicular to the cyst wall.

2.5 Staining

At this point, the tissue slices are nearly invisible under a light microscope and must be stained to create contrast. The most widely used stains for both diagnostic and research histology are hematoxylin and eosin. Hematoxylin stains nucleic acids and appears blue/purple, while eosin stains proteins and appears pink/red when visualized under a brightfield microscope. So for most tissues, cell nuclei are blue, while cytoplasm can vary from clear to red to purple depending on its constituents. The reason that H&E staining of tissue has persisted for decades as the primary tissue stain in diagnostic and research pathology is that these stains attach themselves to almost every cellular component, allowing for visualization of whole cells and all tissue components. Another reason is that these stains provide excellent contrast between cellular constituents by having chemical properties that produce colors at opposite ends of the visual spectrum. These color perceptions are helpful in diagnosis, though not entirely necessary, since, even in grayscale, distinctions between and within tissues can be made and diagnoses rendered.



Figure 2.5: (a) During sectioning, the the tissue block is sliced into 3 to 4 μm sections, which remain connected in a ribbon. (b) Collecting a ribbon from the water bath with a glass slide.

2.6 Visualization

Once stained, the slides must be visualized (Figure 2.7). A growing trend in pathology is to digitize slides so that pathologists can make diagnoses based solely on the digital image [22, 23]. The advent and refinement of whole-slide scanners have made rapid scanning and high-resolution WSIs commonplace. These systems are now offered by many companies and offer spatial resolutions using the $40\times$ objective of approximately $0.25\ \mu\text{m}/\text{pixel}$. Storing digital images would be an attractive alternative to storing glass slides, since glass slides take considerably more space, can be damaged or lost, and fade over time, but currently institutions must keep their glass slides and tissue blocks for at least ten years.

Despite these advantages, in most medical centers the pathologists still rely on visualization of the slide through a microscope. According to pathologists, a microscope offers faster panning, faster focusing, and an intangible sense of being closer to the tissue. This last advantage is more difficult to explain, but the sense is that the eye can capture greater detail through the microscope than from a digital image on a monitor. Even so, studies have shown no appreciable difference in diagnoses rendered by pathologists using digitized images compared to diagnoses rendered using a microscope [24–27].

2.7 Analysis

The goals of the pathologist in the clinical domain versus the research domain are different; we discuss each separately.

2.7.1 Clinical Practice

The main goal of the pathologist in a diagnostic practice is to examine tissue and render a correct diagnosis that will ultimately translate to a therapeutic intervention for the patient. The therapeutic response may range from no action, in the case of a diagnosis of *normal* or *unremarkable*, to close follow-up, local excision, medical treatment only (for a benign diagnoses), or radical chemotherapy and/or surgery (for a malignant diagnoses).

For a given slide, the pathologist analyzes a wide variety of characteristics including tissue architecture, cellular color and texture (Figure 2.8), and cellular/nuclear morphometry (Figure 2.9) to produce diagnoses. Computer algorithms may automate some of these tasks. For example, in cytopathology, automated methods for screening Pap smear slides limit the number of cytotechnologists needed to complete these screenings (though they may not yet be cost-effective [28]). In the future, we could similarly automate or semi-automate

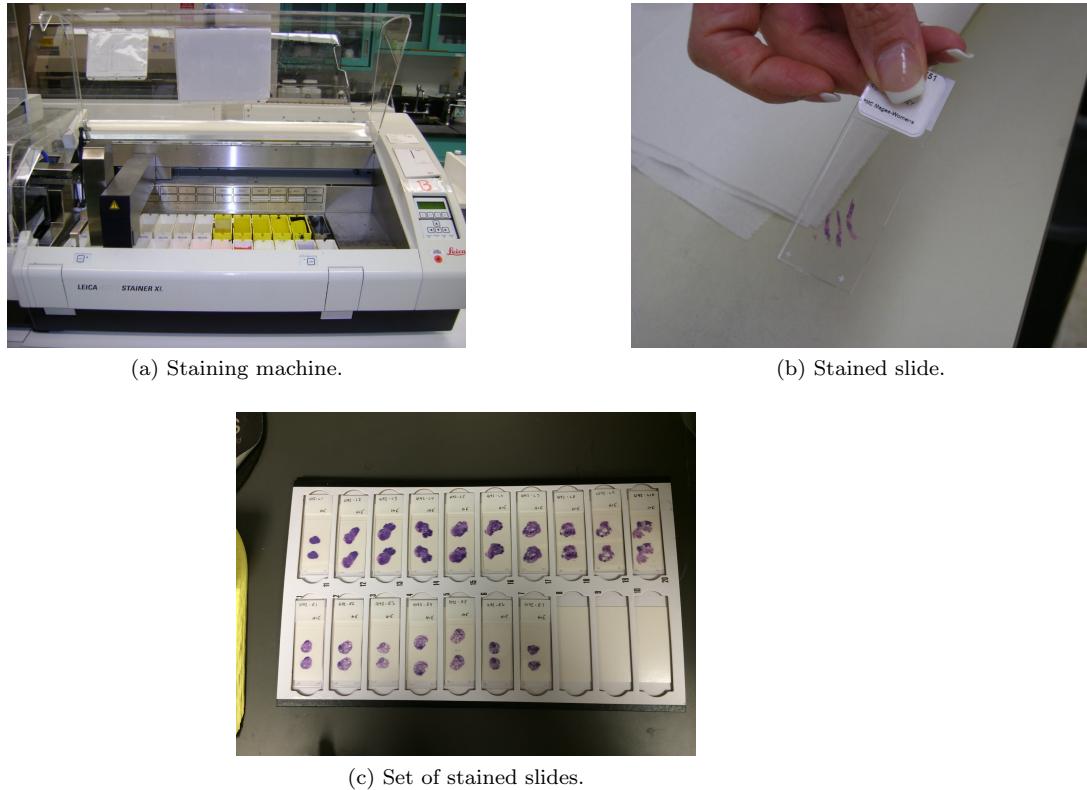


Figure 2.6: (a) A staining machine which moves slides through a series of stains and rinses. (b) A single finished slide; the purple/pink color is caused by the H&E stain. (c) A set of finished slides in a slide tray.

histology screenings, e.g., of the gastrointestinal tract. Such automation could reduce healthcare costs and provide pathologists more time for challenging cases and research.

2.7.2 Research

In the research domain, pathologists may aim to quantify differences between histology samples in terms of a variety of parameters including cellular/nuclear morphometry, amount of stroma (the connective tissue cells that support the function of cells around them) present, types of tissue present, etc. Although some basic analyses can be readily performed using available image analysis software (e.g., Adobe Photoshop, ImageJ [29], or Molecular Devices MetaMorph), most analysis in pathology remains semiquantitative: staining intensity may be rated as *low*, *moderate*, or *strong*; the amount of a certain cell type may be visually estimated as *0 to 25%*, *25 to 50%*, or *>50%* of the total population; and morphometric descriptions of cells are limited to semantic descriptions such as *larger*, *thickened*, *pleomorphic*, or *cellular*.

For some studies, this type of analysis is not accurate enough, because many biologically or clinically relevant features cannot be easily captured and processed by the human visual system. For example, given two tumors, how would a pathologist support the claim that the average nucleus size is different between them? Similarly, how can pathologists quantify complex patterns such as chromatin distribution? Automated analysis of histology images would allow researchers to perform these types of quantifications.

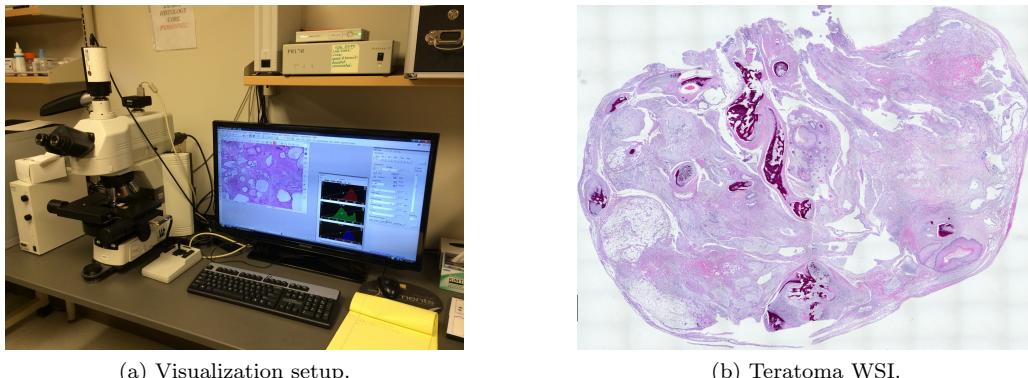


Figure 2.7: (a) Visualization setup including a microscope with attached camera and desktop computer. (b) WSI of a teratoma collected as a set of tiles and stitched together into a large image.

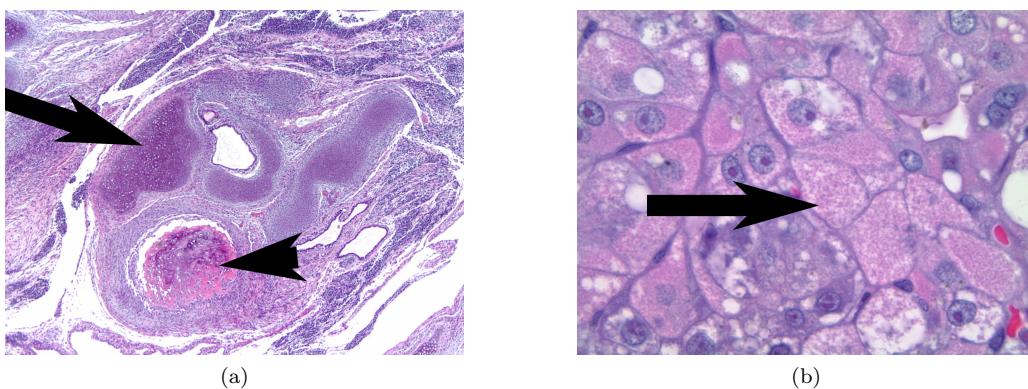


Figure 2.8: Examples of important visual cues in histology. (a) In this teratoma section, color makes distinguishing cartilage (blue/gray, arrow) from bone (pink/red, arrowhead) easy, even at low magnification. (b) In this liver section of a child with mitochondrial disorder, texture is important. Under high magnification, cell borders are accentuated and cytoplasm shows tiny red granules representing abnormal mitochondria (arrow).

2.8 Sources of Variability

There are three main sources of variability in a histology-based diagnosis: biological variability, inter-observer variability, and technical variability. Biological variability encompasses the normal variability among people and the myriad of pathological processes that can affect any tissue group. Due to biological variability, slides generated from the same tissue in different patients can look different. Interobserver variability is the variability in interpretation among pathologists looking at the same histology slide. Finally, technical variability is the variability in a slide's appearance due to how it was prepared. We focus here on the details of technical variability.

In the best case, a slide that is generated from the tissue processing process we described in this chapter will show tissue that is properly oriented, sectioned, stained, and coverslipped. Unfortunately, each of these steps can introduce variability into the final product, and differences in protocol between labs can greatly alter the appearance of even biologically similar tissue samples. Some variables such as fixation, specimen orientation in the block, and microtome sectioning are heavily dependent on human skill, and even though tissue processing, staining, and coverslipping are largely automated, they still depend on human monitoring,

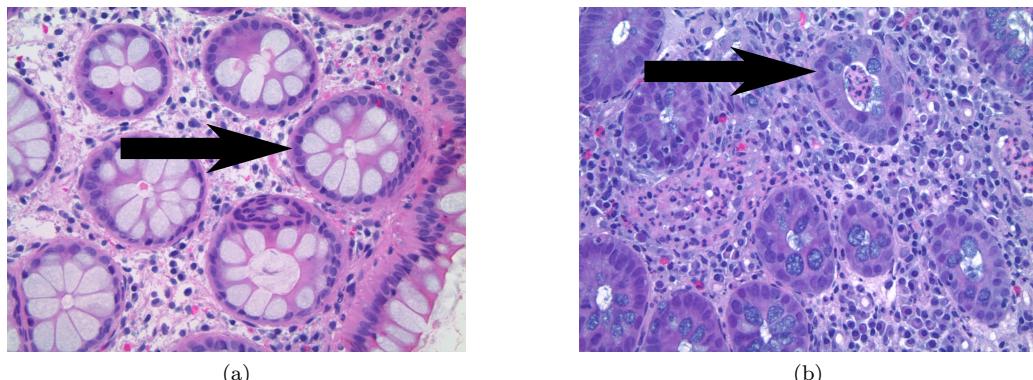
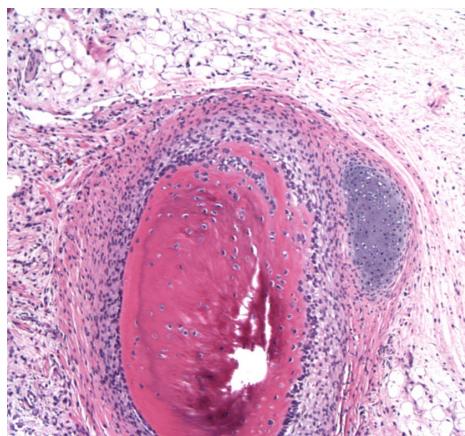


Figure 2.9: Examples of the importance of nuclear shape and distribution in colon screening. (a) In normal colon tissue, nuclei are regular in size and distribution around the colon gland (arrow). (b) In colitis, nuclei become much less regular (arrow).

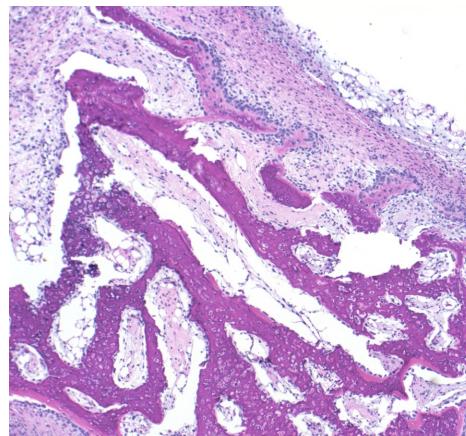
machine maintenance, and solution preparation.

For automated analysis systems, each source of variability presents a significant obstacle. Tissue that is poorly fixed will not cut well and has a blurry appearance, removing important cues such as edges. Tissue that has been dried out will be shrunken and have poor morphology and stain contrast. Sectioning artifacts are some of the most commonly encountered and can produce folds in the tissue, chatter artifacts from a dull blade (seen as alternating light and dark regions), or missing pieces of the tissue.

Staining is a critical source of variability because it produces the color and contrast on the slide (Figure 2.10). Many different formulations of H&E exist, each producing a slightly different appearance. The stain can also be applied in different manners. In progressive staining, the sample remains in the hematoxylin solution for a specified amount of time to render appropriate staining. In regressive staining, the sample remains in the hematoxylin long enough to overstain, and is then destained back to the desired contrast with an acid alcohol solution. Furthermore, both hematoxylin and eosin solutions can have their staining capabilities altered by prolonged storage, contamination by other reagents or water, precipitation, and changes in pH. Staining artifacts can include light staining with either hematoxylin or eosin, precipitated hematoxylin (seen as blue chunks under the microscope), or lack of staining with either hematoxylin, eosin, or both. These artifacts are well understood by pathologists, but may be problematic for automated histology analysis systems that rely on color.



(a) Bone appears pink.



(b) Bone appears purple.

Figure 2.10: Example of staining variability. Both images are of bone, but the color of the bone varies from pink to purple due to staining variability. Such color variations present a challenge for automated analysis systems.

Chapter 3

Automated Histology Image Analysis

In this chapter, we review the current methods and signal processing challenges in automated histology image analysis, including some of our own previous work in the area (both unpublished and from [30]). Again, we restrict the review to slides stained with the common and inexpensive H&E stain and imaged with brightfield microscopy. This focus allows us to study a more coherent group of methods; we argue that the approaches to a single task, e.g. nucleus detection, vary considerably across staining and imaging modalities. This chapter is an expanded version of Section 3 of our histology review paper [19].

We are aware of four previous reviews of automated histology analysis: (1) [31] deals with analysis of histology images including H&E images as well as fluorescence and multispectral images. It covers preprocessing, segmentation of glands, nuclei, and other subcellular components, feature extraction, dimensionality reduction, and classification. (2) [32] tackles the broader field that the authors term *computational pathology*, which includes histology as well as cytology analysis. It discusses issues of data and ground truth collection, including variation among experts and publicly available datasets, and describes automated analysis primarily from a statistical pattern recognition viewpoint. (3) [23] discusses histology WSI informatics, including quality control during image acquisition, feature extraction, region of interest (ROI) detection, and visualization. (4) [33] focuses on breast cancer histology image analysis, including both H&E and immunohistochemical (IHC) stains.

While we do not include every paper on automated histology, we aim to give the reader a sense of what has been tried for the various histology analysis tasks we present. We have given priority to recent journal papers except where necessary. Also note that we do not focus on results of or comparison between methods. This is because, at this stage, automated histology research is diffuse: most methods are tailored to private datasets and there is no consensus on what quantitative metrics should be reported. Moving toward shared datasets and metrics will be a critical step forward in the field.

This chapter is organized around a generalized block diagram of a histology analysis system (Figure 3.1). For each block in the diagram, we describe the signal processing challenge and review the current approaches in the literature.

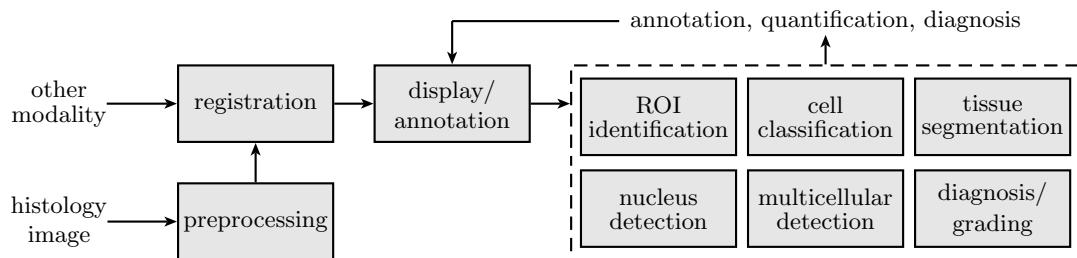


Figure 3.1: Block diagram of a generic automated histology analysis system.

3.1 Preprocessing

Histology images exhibit the same types of artifacts and noise as any digital microscopy image, as well as some novel ones introduced by tissue processing. These can affect analysis unless removed with appropriate preprocessing. A good overview of noise and artifacts in digital microscopy and methods to correct them is given in [34]; in this section we focus on the issues specific to H&E images of tissue processed as described above.

Stain normalization.

As discussed in Section 2.8, staining variations affect the appearance of histology images; such variations are problematic for automated analysis because color is a critical feature in histology. Stain normalization is the process of taking two H&E images that have staining variation between them and removing this variation. Doing so has been shown to improve histology image segmentation [35,36]. One approach to stain normalization is to use color normalization techniques from photography such as histogram equalization; e.g., in [35] the rank statistics of the input image are scaled to match those of the reference image separately in each color channel. The method has been successfully used in other automated histology work, including [37]. Such methods are especially suited for cases where the images to be normalized show approximately the same tissue, e.g., in the case of serial sections such as in [38].

An approach more specifically tailored for H&E image normalization is to first separate the image into hematoxylin-only (H-only) and eosin-only (E-only) images (sometimes called *color deconvolution*), then normalize these images separately and recombine. This approach better handles cases where, e.g., the hematoxylin stain is too intense but the eosin stain is too weak. When the stain colors are known, the color deconvolution method in [39] solves the separation. When they are not known, the problem becomes more challenging. The approach taken in [10,36,40] is to estimate the stain colors and deconvolve as before. In [10], the stain colors are estimated using the fact that all stained pixels will lie on a wedge in color space; this is a special case of the non-negative matrix factorization (NMF) problem studied elsewhere. Reference [40] uses expectation-maximization to find clusters in chromaticity space that correspond to stain colors, and [36] finds pixels stained with only one stain via supervised classification and uses their mean color to estimate the stain colors. A different approach is to estimate the single-stain images directly; e.g. [9] searches for an H-only image that removes most of the contrast from the red channel of the input, based on the assumption that most this contrast comes from the nuclei, which are stained only by hematoxylin.

These separation methods all produce qualitatively fair results, but it is unclear which comes closest to the correct separation. One of the contributions of this thesis is a new stain separation method and benchmark [16], which we discuss in depth in Chapter 4. Our benchmark indicates that the method of [10] is superior to that of [9], but we did not evaluate any of the more recent approaches.

Tissue deformation.

Because the tissue slices are very thin, they can fold over on themselves during processing, creating a tissue area that is doubly thick. The authors of [41] proposed detecting these folds by their high color saturation, and the authors of [42] devised a method for selecting a good saturation threshold for this detection, but, to our knowledge, no work addresses correcting tissue folds digitally. Additionally, differences in the water content of different tissues can cause them to pull apart when the tissue is dehydrated during processing; the result is white cracks that are not biologically meaningful. Again, we are aware of no work that addresses correcting these artifacts.

Stitching.

High-resolution WSIs are often acquired by imaging several strips or tiles separately, which then must be stitched together to create the final image. Because the offsets between the strips or tiles are known, adequate stitching results are often provided by the microscope/scanner software or can easily be achieved with simple compositing techniques. A more challenging stitching problem occurs when a large gross specimen must be

sectioned and imaged as several pieces. In [43], the authors describe software that allows stitching of high-resolution images of such tissue fragments via hand-selected control points and linear transformation. We are not aware of a work that addresses histology image stitching with automatic control point selection or explores a richer set of transforms.

3.2 Registration

There are a variety of H&E histology image registration tasks, depending on the image modalities involved. In this section, we review three common registration tasks involving H&E images. For another review of single- and multi-stain registration in histology, see [44].

H&E to H&E.

Given serial sections of a tissue sample, each stained with H&E, one aim is to register them to create a tissue volume. This 3D reconstruction can more fully show the extent of a pathological process or show relationships of tissue types to one another. This is a difficult registration task: each slice undergoes nonrigid deformation during processing and may exhibit cracking or folding artifacts. The standard approach is to register in a coarse to fine manner as in [38], where the authors register images of serial tissue slices to explore the 3D shape of cervical tumor fronts using a series of three registrations: (1) a rigid registration using a frequency domain method, (2) a polynomial registration using control points automatically selected with correlation matching on small patches, and (3) a registration consisting of unconstrained local displacements regularized by local curvature. An approach more tailored to histology is to use specific anatomical landmarks; e.g., [45] registers based on blood vessels.

H&E to another stain.

One may also want to register images of two adjacent tissue slices that use different stains; e.g. H&E to IHC stains. This task is difficult because, by design, the stains will give contrast to different structures. One way to overcome this obstacle is with the selection of an appropriate pixel-wise similarity measure, e.g. the authors of [44] register H&E images to a variety of IHC images using mutual information and the elastix software package [46]. They solve the problem of large image size by finding the transform on downsampled images and then applying it in real time only to the region of the slide being viewed. Another approach is given in [47], where distinctive landmarks such as blood vessels are segmented and used to compute the registration. Correspondences between these objects are found based on size and eccentricity and used to compute an initial rigid segmentation and subsequent non-rigid registration. An accurate non-rigid registration follows based on correspondences between groups of the same objects.

Multi-stain registration can also be completed with new laboratory techniques. In [48], the authors propose H&E staining followed by chemical destaining and IHC of the same slide, removing the need for anything but translational registration. The drawback of the technique is that it requires non-standard processing and may not work with all staining types. Furthermore, careful examination of the results of the study shows that some tissue deformation still occurs, which has also been our experience with chemical destaining [16]. Therefore, chemical destaining is unlikely to completely obviate the need for digital registration methods.

H&E to MRI.

Compared to histology, magnetic resonance imaging (MRI) has low contrast and resolution, but has the advantage of being noninvasive. Registering histology images to MR images could help train radiologists, provide better non-invasive diagnoses, and enable the development of MRI-based computer-aided diagnosis tools [49]. This task is difficult because MR images are 3D while histology images are 2D, MR images are lower resolution than histology images, and because the contrast in MR and histology images is generated in different ways. When considering MR images collected *in vivo*, registration is even more difficult because

surgical extraction and histology processing can greatly deform the tissue. In [50], histology images are registered to in vivo MR images via two intermediates, the block face photo and the ex vivo MR image, with the idea that the deformations between these intermediate stages are less drastic and therefore easier to estimate. Each step of the registration is completed using mutual information as the metric and thin plate splines to deform the image with control points initialized by hand. In [49], histology images are registered to MR slices to create a histology volume, then the MR volume is registered in 3D to the histology volume and resliced. The process is iterated until convergence.

In each of these scenarios, comparison between methods is difficult because no ground truth exists. This problem is intensified in the first two scenarios because qualitatively good registrations may actually remove true differences between the adjacent tissue slices being registered.

3.2.1 Our Previous Work

In unpublished work, we experimented with registration of WSIs of a teratoma tumor. Our approach was to detect SURF [51] control points in low resolution grayscale versions of the WSIs, find correspondences via RANSAC [52], then solve for a rigid transformation and apply it to the full resolution WSIs. This approach provided reasonable results at gross scale (Figure 3.2), was not accurate on the scale of structures like glands, much less individual nuclei (Figure 3.3). Small-scale accuracy will require non-rigid transformation to correct the deformation that tissue sections experiences during microtomy. Specifically which class of non-rigid transformations should be used and how they should be regularized remain open questions.

3.3 Display and Annotation

A fundamental problem in the display of histology images is that they can be huge, easily several GB for uncompressed WSIs. As a result, commercial slide scanners often save images in a proprietary format for which the manufacturers provide viewing software. For the researcher interested in reading the images themselves, one solution is OpenSlide [53], a C library that aims to allow slides from any vendor to be opened, manipulated, or converted to other formats. One format that OpenSlide can convert to and is a natural fit for large histology images is the Deep Zoom (DZI) format [54], which creates from a large image a tiled image pyramid, allowing real-time viewing of images of arbitrary size, even streaming over the Internet. One example of this approach can be found in [55], which involves displaying WSIs from The Cancer Genome Atlas dataset [56] on the Internet using the similar Zoomify format [57]. The system in [58] is designed specifically for viewing histology images and avoids the time-consuming calculation of image pyramids by creating the current view in real time from image tiles.

Beyond simply viewing large histology images, researchers may want to collect expert annotations for them for the purpose of training and testing their analysis algorithms. In the simplest case, the expert pathologist may label an entire image, e.g., as *normal* or *cancer*. For more fine-grained labeling, Aperio ImageScope [59], PathCore Sedeen Viewer [60], and Cytomine [61] provide freehand annotation tools for large images. The next step for histology image display and annotation is systems allowing the pathologist to interact with automated analyses, teaching and correcting the system in real time. Some efforts have been made in this area: after the automated segmentation in [62], experts can click on nuclei to indicate that they either need to be split or merged; Cytomine includes collaborative proofreading of automated cancer segmentation; and 3D-DOCTOR [63] includes automated segmentation.

3.3.1 Our Previous Work

In unpublished work, we created a web-based histology WSI viewer and annotation tool using the DZI format. This was a convenient way explore WSIs and share them with others (sending a web link rather than a large image file), but given the plethora of free and commercial solutions that do similar things, we discontinued development.

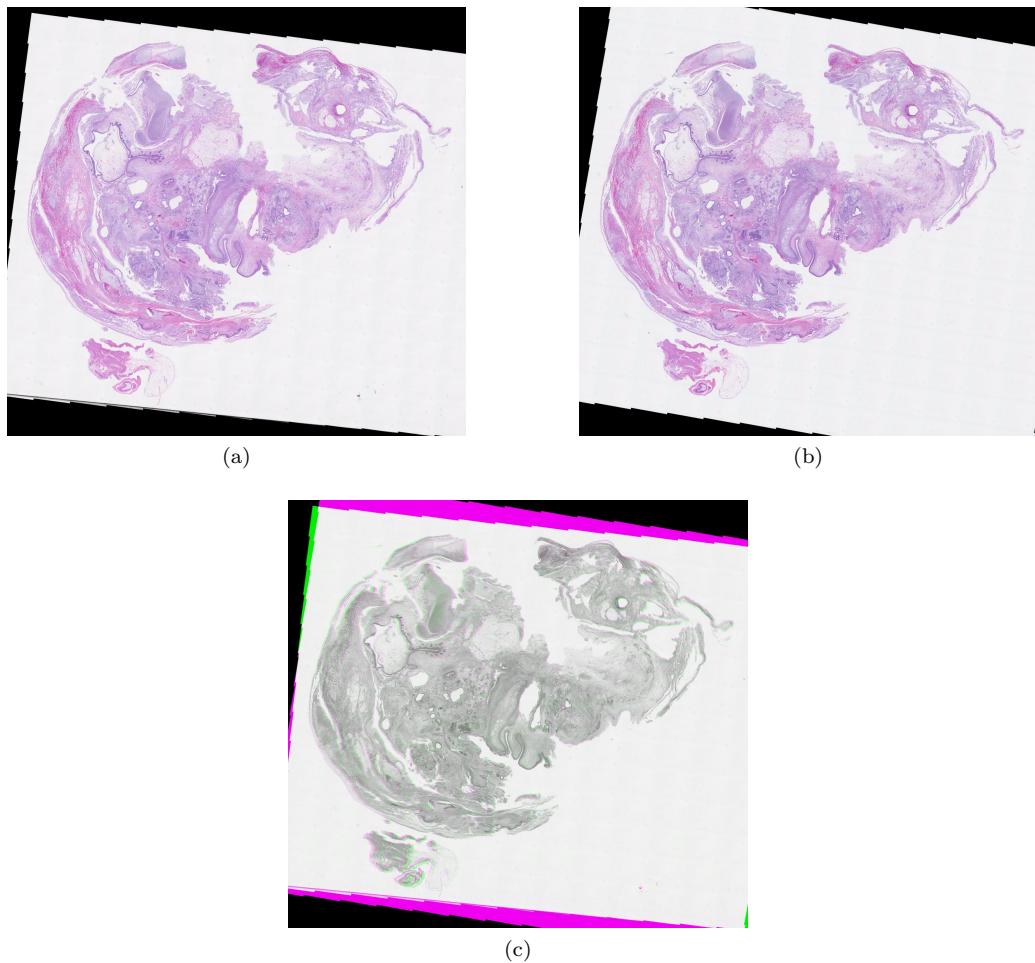


Figure 3.2: (a and b) Two adjacent sections from a teratoma tumor registered via a rigid transform. (c) A visualization of their difference (pink and green represent disagreement, gray indicates agreement). The tumor is about 1cm in diameter. At this scale, the agreement between the two images is good.

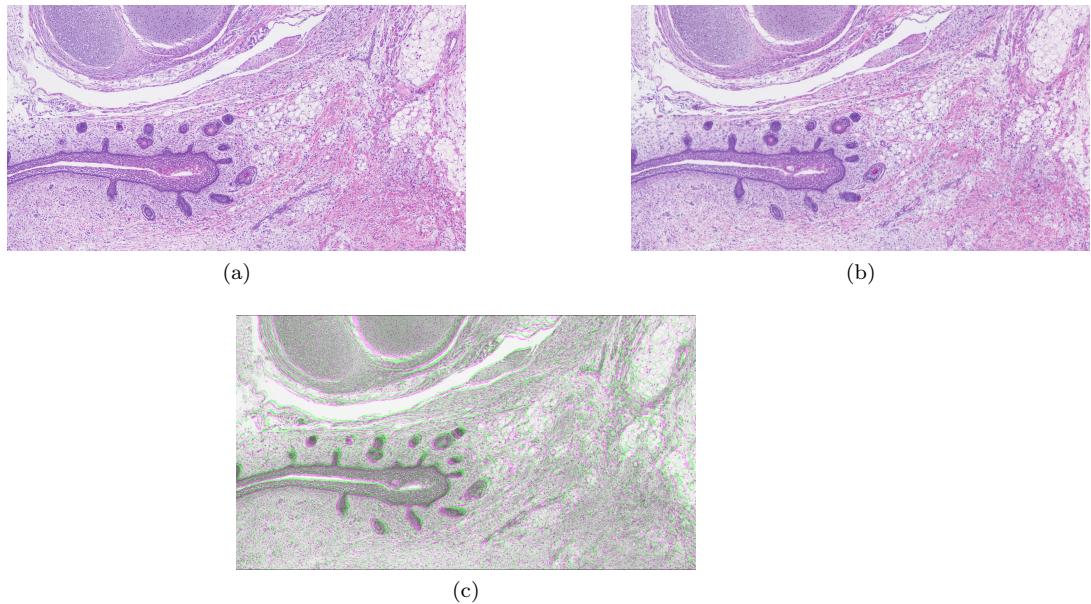


Figure 3.3: (a-c) Zoomed versions the images in Figure 3.2. At this scale, inaccuracies in the registration are apparent, indicating the need for non-rigid registration.

3.4 Region of Interest Identification

A single biopsy can generate dozens of high-resolution WSIs, however, often, only a small region of this vast quantity of tissue is diagnostically useful; these areas are known as regions of interest (ROIs). There is a need for fast computational methods that can identify these ROIs in histology WSIs. Once identified, these regions can be passed to a pathologist or to subsequent steps of a larger automated histology system. The key performance metric for these systems is false negative rate: wrongly tagging regions as ROIs when they are normal may waste time, but missing a true ROI could lead to a disease not being diagnosed.

One approach is to simply downsample the input; e.g. [64] uses a low-resolution input to extract features based on color and sparse coding of subpatches. These features are classified via support vector machine (SVM) to detect ROIs. Another approach is to process the image at multiple scales, such as in [65], where WSIs of breast cancer tissue are recursively partitioned via color clustering at increasingly fine resolutions to efficiently identify tissue vs. non-tissue and lesion vs. normal regions. Yet another approach is to detect objects such as glands and identify ROIs based on these [66]. There are generally fewer objects on a slide than pixels, so processing objects can be more efficient than processing pixels.

3.5 Nucleus Detection

Nuclei are prevalent in histology images and their size, shape, distribution, and texture are relevant for many analysis tasks such as determining tissue type, identifying inflammation, and identifying and grading cancer. Therefore, locating nuclei is a critical step in many histology analysis systems. Nucleus detection in histology images is challenging because nuclei can be tightly clustered and vary in size, shape, and color depending on their cell type. Even within a single cell type, the stain may not penetrate all nuclei equally, leaving some darkly stained and others lightly stained. Finally, human nuclei have diameters approximately equal to the thickness of the tissue slice ($5 \mu\text{m}$), meaning that, for some nuclei, only a portion appears on the slide.

Basic approaches to nucleus detection involve color clustering of the pixels; e.g. the work in [67] uses

clustering in the Lab color space to identify four subcellular components, namely nuclei, cytoplasm, neuropil, and background, and [68] uses expectation-maximization clustering to identify regions of lymphocyte nuclei, stroma, cancer nuclei, and background.

Using only local information omits the strong prior knowledge we have about nucleus size and shape. Approaches that move beyond local information include [64] and [68], which use active contours to refine their initial color segmentation, and [37], which uses a graph cut based on color and Laplacian of Gaussian features. Similarly, the authors in [62] use Laplacian of Gaussian filtering with clever scale selection to detect nucleus seed points, followed by local maximum clustering to form a rough segmentation. Another approach is to frame nucleus detection as a classification problem; e.g. in [69], an SVM is trained to detect rectangular windows containing nuclei; the features used are pixel intensities and Laplacian of Gaussian edge intensities.

Finally, several methods use heuristics to separate clustered nuclei, including the curvature-based reasoning in [37] and concavity detection in [68]. The method in [62] separates nuclei via graph cuts. These approaches achieve impressive-looking results even when nuclei are clustered, but large, hand-annotated datasets will be necessarily to quantitatively compare them. Reference [70] provides a step in this direction with a set of 58 breast cancer histology images including expert nucleus segmentations.

3.5.1 Our Previous Work

In [30], we developed a nucleus detection scheme as part of an automated colitis detection system. We now describe this scheme. From a P by Q color input image, $I : \mathbb{Z}_P \times \mathbb{Z}_Q \rightarrow \mathbb{R}^3$, we first create a nucleus mask, $1_N : \mathbb{Z}_P \times \mathbb{Z}_Q \rightarrow \{0, 1\}$, based on color clustering. This mask takes the value one at the location of pixels which have a color similar enough to a predefined nucleus color and zero elsewhere. We extract local shape information from the nucleus mask using local moment filters in an approach similar to [71]. Briefly, we begin with a windowing filter, w , that is non-negative, symmetric, and sums to one. From this window, we define the order- p, q local moment filter

$$w_{p,q}[m, n] = w[m, n] m^p n^q.$$

At location $[m, n]$, the local moment transform of the image, I , is

$$m_{p,q}(I)[m, n] = w_{p,q}[m, n] * I[m, n], \quad (3.1)$$

where $*$ denotes convolution and where we choose w to be Gaussian with a standard deviation of $1/3$ the expected radius of a nucleus (in pixels).

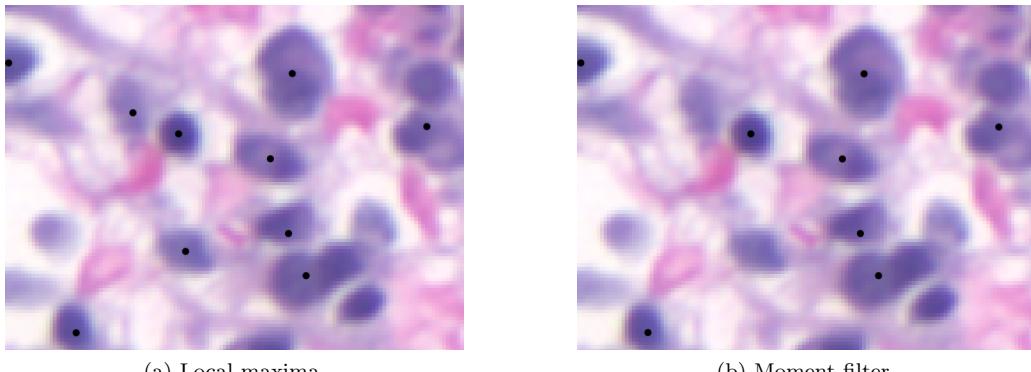
We apply (3.1) to the input image to obtain m_{00} , m_{10} , m_{01} , m_{11} , m_{20} , and m_{02} . We then convert them to central moments,

$$\begin{aligned} \bar{x} &= \frac{m_{10}}{m_{00}}, & \bar{y} &= \frac{m_{01}}{m_{00}}, \\ \mu_{20} &= m_{20} - m_{00}\bar{x}^2, & \mu_{02} &= m_{02} - m_{00}\bar{y}^2, & \mu_{11} &= m_{11} - m_{00}\bar{x}\bar{y}, \end{aligned}$$

and form the inertia matrix $J[m, n] = \begin{bmatrix} \mu_{20}[m, n] & \mu_{11}[m, n] \\ \mu_{11}[m, n] & \mu_{02}[m, n] \end{bmatrix}$. At each pixel location $[m, n]$, J describes the horizontal and vertical spread of the pixels within the support of w centered at the location $[m, n]$, calculated around the local center of mass $[m + \bar{x}, n + \bar{y}]$ rather than around $[m, n]$. From J , we calculate the local eccentricity

$$e[m, n] = \frac{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}{(\mu_{20} + \mu_{02})^2}. \quad (3.2)$$

In a departure from [71], we then use the shape information computed by the moment filters to locate nucleus centers. We take as candidates each position where m_{00} reaches a local maximum, that is, every pixel for which m_{00} is larger than at any of its eight neighbors. We then remove candidates for which m_{00} is too small (less than 0.25) or for which the eccentricity is too high (greater than 0.1). Intuitively, this scheme tags pixels that are the centers of nucleus-sized blobs, rejecting those that are too oblong to be nuclei. An example comparing this scheme to simple maxima detection is shown in Figure 3.4.



(a) Local maxima.

(b) Moment filter.

Figure 3.4: Comparison of nucleus detection methods, with black dots marking detections. (a) Using simple maxima detection. (b) Using moment filter-based detection. The moment-filter approach rejects the two objects at the center-left because they are overly oblong.

3.6 Cell Classification

Most histology images contain cells of several types; moving beyond nucleus detection to cell classification can provide valuable diagnostic information. For example, the method in [72] classifies cells as centroblasts or normal, which is useful for cancer grading. It uses color and Fourier-based texture features with quadratic discriminant analysis as the classifier. The nuclei found in [68] are classified as belonging to lymphocytes or other based on their color.

Other work focuses on detecting mitotic cells. Notably, [73] presents a contest dataset for this task which comprises 50 images collected on each of two different slide scanners as well as a multispectral scanner. The highest-scoring approach at the time of the contest was based on a deep convolutional neural network [74].

3.7 Multicellular Structure Detection

Cells in histology images are not solitary, rather they are part of organized structures (e.g., glands, acini). Detecting and analyzing these structures is a unique challenge in histology image analysis. Because there are many types of multicellular structures, these tasks tend to be more varied and application-driven than the ones we have discussed before, hence we give only a few examples here.

Glands are a multicellular structure common to many tissue types (e.g. salivary, breast, prostate, pancreatic, sinonasal, gastrointestinal tissues), and changes in their morphology can be an important indicator of disease. In histology images, most glands appear as clear areas surrounded by cells. The method in [66] leverages the clear areas to find seed pixels and then uses region growing to segment the glands. Going further, [66] classifies glands as malignant or normal based on their size with a Markov random field to impose spatial smoothness. In [75], a graph is built on top of a colon tissue image with nodes corresponding to either nucleus or non-nucleus objects. The colon glands can then be described by subgraphs around a user-selected point at their center. Matching these subgraphs to reference ones from healthy or diseased glands allows classification of the input image.

Some approaches to segmenting multicellular structures treat them like nuclei, e.g. [76] segments lymphoid follicles, which are organized groups of lymphoid cells, using active contours. Detected regions are split based on curvature and false detections are further trimmed based on color.

3.8 Tissue Segmentation

Tissues are organized groups of cells. Identifying them in an image is important for diagnosis or giving context to subsequent analyses. The wide variety of tissues and the complexity of their appearances makes this a challenging problem. The method in [77] uses local pixel intensities as features and is able to segment bone, cartilage, and fat tissue in teratoma tumor images. We presented a segmentation method [17] inspired by the lack of edges in histology images. It uses local color histograms rather than edge-based features and outperforms generic methods for tissue segmentation. This paper forms the basis of Chapter 5 of this thesis.

Other methods rely on histology-specific features. The authors in [78] showed that segmentations seeking homogeneity of objects such as cells and crypts, rather than simply pixel homogeneity, perform well on colon tissue. We developed a method [79] that classifies pixels based on local features such as nucleus density and color.

Some tissue-level analyses amount to diagnoses; e.g. in [80], small subregions of a prostate tissue image are classified as normal, stroma, or prostatic adenocarcinoma. Normal regions are those around glands, which are simple to segment because of their white centers. Stroma and cancer regions are distinguished using Haralick texture features. Another group [81] used a graph-based methodology to segment regions of cancer in colon images, where features were based on the frequency of co-occurrence of nodes.

Tissue segmentation is one of the main contributions of this thesis, we discuss our approach in Chapter 5.

3.9 Diagnosis

Automated diagnosis is essentially image classification: given a histology image, what disease does it represent? As such, most approaches follow the paradigm of feature extraction followed by classification. The authors in [82] address the problem of classifying subtypes of renal tumor in expert-selected ROIs. They use Fourier shape descriptors extracted from binary masks of nuclei, cytoplasm, and unstained regions as features and a series of SVM classifiers arranged in a directed acyclic graph to distinguish between three types of renal cell carcinoma and one benign tumor. Based on which shape descriptors are most distinguishing during classification, the authors can identify which shapes are indicative of each tissue type. In [83], hand-selected ROIs of breast tissue are classified as normal, *in situ* cancer, or invasive cancer. They use generic features including local binary patterns, cooccurrence matrix statistics, and curvelet coefficient statistics. Classification proceeds in two stages, first a random subspace ensemble of SVMs, then a random subspace ensemble of neural networks. Each stage may either classify an image or reject it as too difficult. Images rejected from the first stage move on to the second, while images rejected from the second stage are viewed by a human expert. The method achieves accuracy of over 99% with a 1.94% rejection rate. The recent conference paper [84] moves away from hand-designed features by learning features from the data using sparse representation.

Grading is diagnosing the severity of a disease. Established grading scales are useful guides for algorithm development, and automated grading promises increased repeatability over human grading. The method in [64] grades breast cancer by analyzing the size, shape, and texture of the nuclei inside an ROI using a Bayesian classifier. Another approach is to include some notion of cellular components without specifically segmenting nuclei. In, [85] greyscale thresholding and morphology are used to find blobs that are then classified based on intensity and size into three different nucleus types and stroma. The image is then segmented into areas of high and low nucleus density. Features including nucleus count, nucleus spacing, and tubule count are extracted for the high density areas only. The images are finally classified into grade one, two, or three with a quadratic classifier. In [67], Haralick features are extracted from cytoplasm and neuropil regions and used to determine the differentiation level of neuroblastoma tumors. The classification step is an ensemble of seven different classifiers and dimensionality reduction methods combined via weighted voting. The method handles WSIs by splitting them into small tiles that are processed in parallel as well as first classifying a downsampled version of the tile and using higher resolution data only if the classification certainty is low.

Finally, generic features can be used. In [86], prostate cancer is detected and graded based on color

and color co-occurrence features and a random forest classifier. In [87], two different lung cancer subtypes are differentiated in images of tissue microarrays and hand-selected ROIs from full slides. The red and blue channels of the image are histogram-stretched to enhance the H&E contrast, and then Haralick and densitometric (e.g. mean pixel value, pixel center of mass, etc.) features are used in a boosting decision tree, achieving accuracy over 90%.

3.9.1 Our Previous Work

In [30], we explored the use of local features (based on our previous work [79]) combined with multiple-instance learning (MIL) [88] for an automated colitis detection system. We describe that system here.

Features.

We use color to assign each pixel in the input image to one of four object types: nuclei, red blood cells, background tissue, and empty slide. Since each of these objects has a distinct color under H&E staining, we use our prior knowledge to assign a set of color values to each object (shades of blue for nuclei, red for red blood cells, pink for background, and white for empty slide). For each image, we adjust these color values by running five iterations of k-means clustering. This step helps account for illumination and staining intensity variations. We then label pixels according to their nearest cluster (Euclidean distance in RGB space).

As described in Section 3.5.1, we use the pixel labels to create a nucleus mask, 1_N , a binary version of the input image that takes a value of one wherever nuclei are present. Similarly, we create masks for red blood cells and background tissue, 1_R and 1_B , respectively. We also use the nucleus center detections from Section 3.5.1 to form a new mask, 1_C , which takes the value one for those pixels that have been labeled as nucleus centers. We proceed to use these masks and an averaging filter, \hat{w} (Gaussian with standard deviation of nine times the expected nucleus radius) to create the feature set we list in Table 3.1, where \cdot and $\overline{\cdot}$ represent pointwise multiplication and division, respectively. These six features form a length-ten (because there are two color features) feature vector for each pixel in the image.

Feature name	Expression
Nucleus density	$\hat{w} * 1_C$
Average nucleus size	$\frac{[\hat{w} * (m_{00} \cdot 1_C)]}{[\hat{w} * 1_C]}$
Average nucleus eccentricity	$\frac{[\hat{w} * (e \cdot 1_C)]}{[\hat{w} * 1_C]}$
Average nucleus color	$\frac{[\hat{w} * (I \cdot 1_N)]}{[\hat{w} * 1_N]}$
RBC coverage	$\hat{w} * 1_R$
Average background color	$\frac{[\hat{w} * (I \cdot 1_B)]}{[\hat{w} * 1_B]}$

Table 3.1: Colitis feature set.

Pixel-level classifier.

After feature extraction, we classify pixels using an artificial neural network (MATLAB Neural Network Toolbox). The network has ten input nodes, no hidden layers, and one output node, with a bias connected only to the output node; activation functions are linear throughout. For training, we throw away all nontissue pixels, then label the remaining pixels according to the image they come from. We call this version of the system the *pixel-level classifier (PLC)*.

Since our goal is to label images rather than pixels, we need a way to convert the pixel-level labels into image-level labels. We accomplish this by selecting a threshold and labeling any image with a number

of inflamed pixels greater than the threshold as *colitis*. The threshold is selected from the training data so as to give the least training error at the image level.

Multiple-instance learning.

The PLC training we just described overlooks an important aspect of the colitis problem: even in an image labeled *colitis*, some pixels actually belong to normal tissue. To understand how to adapt our PLC training to this problem, we look at the classification task in the multiple-instance learning (MIL) framework [88].

Let X be a set of N digital images, $X = \{x_i\}_{i=1}^N$. Each x_i contains multiple instances (pixels) $\{x_{i,j}\}_{j=1}^{P \times Q}$, where P and Q denote the height and width of each image, respectively. Each pixel $x_{i,j}$ has a corresponding label, $y_{i,j}$ where $y_{i,j} \in \{0, 1\}$ (in our case, 0 denotes *normal* and 1 *colitis*), but only the overall label for each image, $\{y_i\}_{i=1}^N$ with $y_i \in \{0, 1\}$, is known during training. For each image, the relationship between y_i and $\{y_{i,j}\}_{j=1}^{P \times Q}$ is given by

$$y_i = \begin{cases} 1, & \text{there exists a } j \text{ such that } y_{i,j} = 1; \\ 0, & \text{otherwise.} \end{cases}$$

In other words, a pathologist will label (diagnose) an image as *colitis* if any region in the image exhibits inflammation.

The goal is to create an image-level classifier function d such that $d(x_*) = y_*$ for any unseen image, x_* , with true label y_* . This is accomplished by learning a pixel-level classifier function p such that $p(x_{*,j}) = y_{*,j}$ for any unseen pixel, $x_{*,j}$, with true label $y_{*,j}$, such that

$$d(x_*) = \begin{cases} 1, & \text{there exists a } j \text{ such that } p(x_{*,j}) = 1; \\ 0, & \text{otherwise.} \end{cases}$$

We thus see that some of the training labels for the PLC training described earlier were incorrect: some *colitis* pixels should have been labeled as *normal*. To address this issue, we first use the basic PLC to identify the top 50% most inflamed pixels in each colitis image and create a new training set with only these pixels labeled as *colitis*, and retrain the classifier. We call the retrained classifier the *PLC-MIL*.

Experiments and discussion.

We compared the performance of the PLC and the PLC-MIL to two freely available biological image classifiers. The first is WND-CHARM [8], which extracts a large number generic image-level features, then classifies images with a nearest neighbor algorithm. We used the expanded color feature set. The second is the multiresolution classification system (MRC) [6], which decomposes the image into subspaces via a multiresolution transform, extracts features and classifies within each subspace separately, then combines the subspace decisions via weighted voting. We used two levels of a wavelet packet decomposition and the expanded Haralick texture feature set.

Feature extraction for WND-CHARM required a powerful desktop computer (2x Intel Xeon E7540 2.00GHz and 64GB RAM), while the PLC and MRC features were computed on a laptop (Intel Core i7 2.67 GHz and 4 GB of RAM). For a single image, feature extraction required 30 seconds for the PLC, 5 seconds for the MRC, and 140 minutes for WND-CHRM. Each classifier was evaluated at each magnification level ($40\times$, $100\times$, $200\times$, and $400\times$) with a leave-one-out cross-validation over twenty normal and twenty colitis images.

Table 3.2 summarizes the results. They show that no classifier is consistently better than the others at all scales. For all four classifiers, accuracy is generally higher at lower magnifications and decreases as magnification increases. While the PLC classifier performs well, the PLC-MIL classifier improves its performance at high magnification. Notably, the PLC-MIL performs comparably to both the MRC and WND-CHRM, with the advantage of simple, physiologically meaningful features and pixel-level labels (see Figure 3.5).

Magnification	WND-CHRM	MRC	PLC	PLC-MIL
40×	100.0	95.0	97.5	97.5
100×	97.5	90.0	90.0	87.5
200×	87.5	85.0	77.5	90.0
400×	77.5	85.0	80.0	85.0

Table 3.2: Comparison of classification accuracy (in %), with the best accuracy for each magnification in bold.

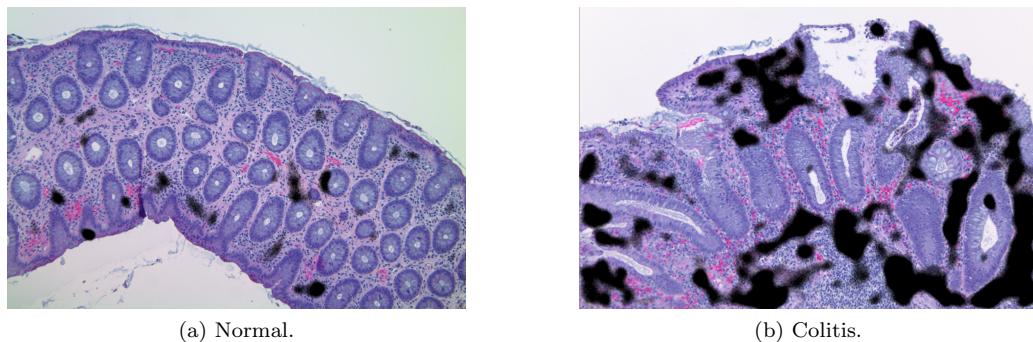


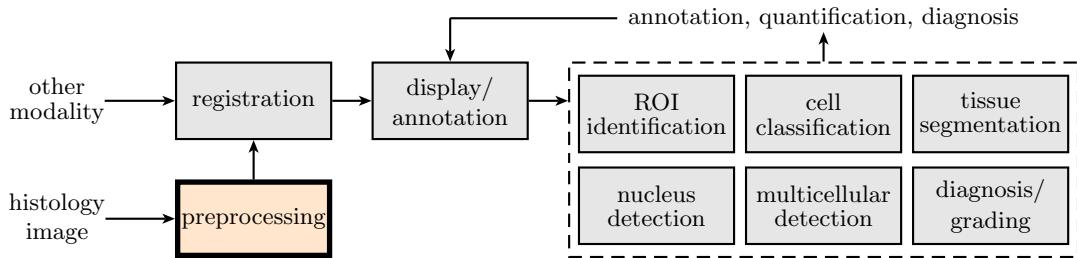
Figure 3.5: Example pixel-level classification results for PLC-MIL (100×). These results are physiologically reasonable: inflammation (black) is detected in regions of high nuclei density and not in crypts or in areas of hemorrhage, which are generally artifacts of the biopsy procedure.

Part II

Normalization and Segmentation

Chapter 4

Normalization



As noted in Section 2.8, variations in H&E stain intensity and color are a challenge for automated histology analysis systems. In this chapter, which is based on our work published in [16], we present a method to separate and normalize the stains on images of H&E histology slides, as well as a new benchmark for stain separation. Our experiments show that our method achieves more accurate stain separation than two comparison methods and that this improvement in separation accuracy leads to improved normalization. In an unpublished experiment, we show that normalization improves the performance of an automated colitis detection system.

4.1 Introduction

Automated histology analysis systems would benefit from accurate stain normalization techniques. Variations in stain color and intensity between slides may arise from differences in staining protocol, differences between stain brands, or the shelf life of stains. This problem is complicated by the fact that H&E images have two stains which must be normalized separately, meaning that standard color correction techniques [89] do not apply. Therefore, stain separation is an important first step in stain normalization. To address this need, this work presents a new algorithm and benchmark dataset for digital stain separation.

4.1.1 Previous Work

Throughout this chapter, we compare to the stain separation methods of [9] and [10]. For example results from these methods as well as our proposed method, see Figure 4.1.

In [9], stain separation is based on the idea that the contrast in the red channel comes from hematoxylin. Therefore, the method finds the E-only image by adding an inverted and scaled version of the image's red channel to its red, green, and blue channels, searching for the scaling in each channel that reduces its variance the most. The result is an image with constant red channel and low variance in the green and blue channels. These changes make the image mostly red and lacking in nuclei, so this is the E-only estimate. The H-only estimate is found by subtracting the E-only estimate from the input and re-white balancing. Throughout the chapter, we will refer to this method as the *color inversion* method.

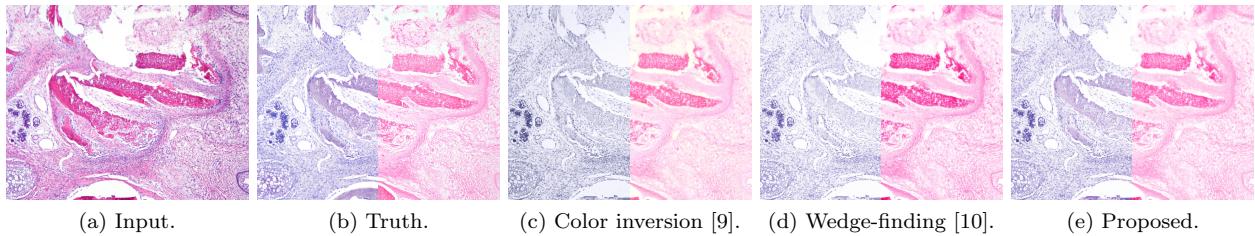


Figure 4.1: Example stain separation showing (a) an input image of an H&E-stained slide, (b) the ground truth stain separation, and the digital stain separation results from (c) color inversion [9], (d) wedge-finding [10], and (e) our proposed method. The left half of each image shows the H-only estimate and the right half shows the E-only estimate.

We, instead, build on the work in [10]; it is explained in detail in Section 4.3.1. The idea is that the color of each pixel in the H&E image is a positive linear combination of the two unknown stain vectors. This means that all the pixels of the image lie inside a wedge with its apex at the color white and its edges determined by the color of hematoxylin and eosin; for this reason we call this approach *wedge finding*. Wedge finding is a special case of NMF, but rather than rely on generic algorithms, the authors propose a geometric solution.

4.1.2 Current Work

Our method improves on the state of the art by adjusting the contrast of its eosin-only estimate and including a notion of stain interaction.

The main contributions of this chapter are

1. a new *benchmark dataset*, generated via chemical destaining, containing ground truth images for stain separation and
2. a new *stain separation algorithm* for H&E histology images.

The outline of the chapter is as follows: In Section 4.2, we describe our benchmark dataset. In Section 4.3, we describe the wedge-find method [10] and our method, which builds off of it. In Section 4.4, we describe our experiments and discuss their results. We conclude in Section 4.5.

4.2 Benchmark Dataset

A key obstacle in advancing the state of the art in digital stain separation is the lack of a publicly available benchmark dataset. In [9], a benchmark is created via a process of staining and destaining a single slide; but it is only used for qualitative evaluation of the result and is not, to our knowledge, available online.

Following a procedure similar to that of [9], we have created our own stain separation benchmark using chemical destaining according to standard histological practice [90]. Our goal was to generate images of the same slide under three staining conditions: H-only, E-only, and H&E. Thus, the H-only and E-only images can serve as the ground truth for separation of the H&E image.

Our procedure was as follows: (1) image an unstained slide; (2) apply eosin stain and image; (3) destain the slide totally and image; (4) apply hematoxylin stain and image; and (5) apply eosin stain and image. At each imaging step, we photographed the slide at the same three locations, giving us three images of the same slide under blank, E-only, destained, H-only, and H&E conditions (Figure 4.2). (We did not capture more images because they are essentially redundant as we don't expect the stain to vary spatially on the slide). The blank and destained images serve as controls: since both show no staining, we can be certain that the destaining process was successful. We provide all of these images in the reproducible research compendium to this work [91].



Figure 4.2: Images collected during benchmark creation.

4.3 Methods

In this section, we review the details of the wedge-finding method of [10], describe our proposed method, which extends it, and explain how we turn a stain separation technique into a stain normalization technique.

4.3.1 Wedge Finding

Let f_{HE} be an $M \times N$ RGB image of an H&E-stained slide with the value $[1, 1, 1]^T$ corresponding to white and $[0, 0, 0]^T$ to black. Making the assumption that the stains follow the Beer-Lambert law [92], we can write the absorbance of the H&E image at each pixel x as a weighted sum of the unknown absorbances of the hematoxylin and eosin stains,

$$f_{\text{HE}}^{\text{OD}}(x) = -\log(f_{\text{HE}}(x)) = \varphi_H a_H(x) + \varphi_E a_E(x), \quad (4.1)$$

where φ_H and φ_E are unit-length 3×1 vectors giving the color of hematoxylin and eosin and $a_H(x)$ and $a_E(x)$ are positive scalars for all x . We refer to $f_{\text{HE}}^{\text{OD}}$ as the optical density (OD) space version of f_{HE} , because high values correspond to high density of stain on the slide (dark colors).

Stacking all of the pixels of $f_{\text{HE}}^{\text{OD}}$ together into a matrix, we can rewrite (4.1) as

$$\begin{bmatrix} f_{\text{HE}}^{\text{OD}} \\ [3 \times MN] \end{bmatrix} = \Phi_{\text{HE}} \begin{bmatrix} a_{\text{HE}} \\ [3 \times 2] [2 \times MN] \end{bmatrix},$$

where $\Phi_{\text{HE}} = [\varphi_H \quad \varphi_E]$ and $a_{\text{HE}} = [a_H^T \quad a_E^T]^T$.

Performing stain separation on $f_{\text{HE}}^{\text{OD}}$ then amounts to an NMF problem where the goal is to find Φ_{HE} and a_{HE} . To do this, we use singular value decomposition (SVD) to project the OD space image onto a plane,

$$\begin{bmatrix} f_{\text{HE}}^{\text{OD}} \\ [3 \times MN] \end{bmatrix} \approx \begin{bmatrix} \Phi \\ [3 \times 2] [2 \times MN] \end{bmatrix} \begin{bmatrix} a \\ [2 \times 1] \end{bmatrix} = \begin{bmatrix} \varphi_0 & \varphi_1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}.$$

To move from the arbitrary orthogonal basis Φ to an estimate of the H&E basis $\hat{\Phi}_{\text{HE}}$, we assume that some pixels exist that are stained with hematoxylin only and that some pixels exist that are stained with eosin only; therefore all the pixels should exist inside a wedge, the extreme members of which correspond to pure hematoxylin and pure eosin. The extreme pixels are those that form the maximum and minimum angle with the φ_0 axis,

$$x^\Delta = \arg \max_x \arctan \left(\frac{a_1(x)}{a_0(x)} \right) \quad \text{and} \quad x^\nabla = \arg \min_x \arctan \left(\frac{a_1(x)}{a_0(x)} \right),$$

and $\hat{\varphi}_H$ and $\hat{\varphi}_E$ are their projections back into OD space,

$$\hat{\varphi}_H = \Phi \begin{bmatrix} a_0(x^\Delta) \\ a_1(x^\Delta) \end{bmatrix} \quad \text{and} \quad \hat{\varphi}_E = \Phi \begin{bmatrix} a_0(x^\nabla) \\ a_1(x^\nabla) \end{bmatrix}.$$

To complete the stain separation, we find weights, $\hat{a}_{\text{HE}} = \begin{bmatrix} \hat{a}_{\text{H}}^T & \hat{a}_{\text{E}}^T \end{bmatrix}^T = (\hat{\Phi}_{\text{HE}})_{\text{left}}^{-1} f_{\text{HE}}^{OD}$, and project the input H&E image separately onto the hematoxylin and eosin vectors, so $\hat{f}_{\text{H}}^{OD} = \hat{\varphi}_{\text{H}} \hat{a}_{\text{H}}$ and $\hat{f}_{\text{E}}^{OD} = \hat{\varphi}_{\text{E}} \hat{a}_{\text{E}}$. Finally we convert back from OD space to RGB space and reshape the results into the single-stain images \hat{f}_{H} and \hat{f}_{E} .

In practice, we replace the minimum and maximum with the α th and $(100 - \alpha)$ th percentile, which provides robustness against outlier pixels (e.g. dirt on the slide). We also exclude nearly white pixels from these calculations because estimation of x^Δ and x^∇ becomes unstable when $\|a\|$ is small. Finally, we may need to swap the identities of $\hat{\varphi}_{\text{H}}$ and $\hat{\varphi}_{\text{E}}$, since the method described so far does not impose that the hematoxylin stain should be blue and the eosin stain pink. This is easily achieved by, e.g., comparing the red values of the estimated stain vectors and assigning $\hat{\varphi}_{\text{E}}$ to be the one that is more red.

4.3.2 Initial Explorations

The collection of the benchmark dataset allows us to evaluate some of the assumptions of the image formation model, (4.1):

Linearity of stains.

We first test that the stains are only of a single color, no matter their concentration; i.e. that pixels having only been stained with one stain lie on a straight line in OD space. For example, this assumes that the eosin stain is always a shade of pink and does not become more yellow at high concentrations. We test the assumption computationally by converting the single-stain images we collected into OD space and checking that the pixels lie on a single line. Deviation from this line suggests that the stain absorbs different frequencies of light as it increases in concentration. Our results (Figure 4.3) show that both stains are mostly linear, with slightly more deviation for hematoxylin.

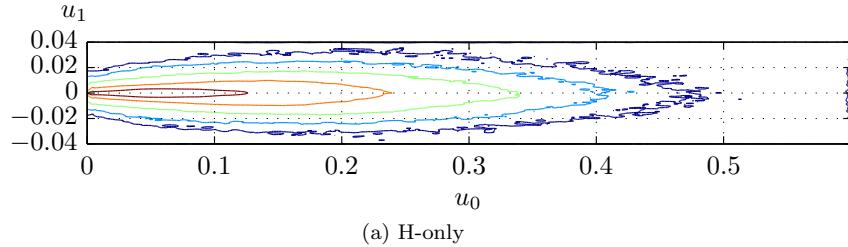
Addition of stains.

We then test whether each pixel in the H&E image is a positive linear combination of the stain vectors in OD space, as stated in (4.1). Our approach is to compare the color histograms of the H-only, E-only, and H&E images, as shown in Figure 4.4. If (4.1) held, then each point in the H&E image would be a sum of one point in the H-only and one point in the E-only image; however the histograms show a large number of points that cannot have been formed in this way. This suggests that either there is an interaction between H and E when they are mixed that has the effect of strengthening the E stain, or that the E-only image is not representative of the eosin staining in the final H&E image.

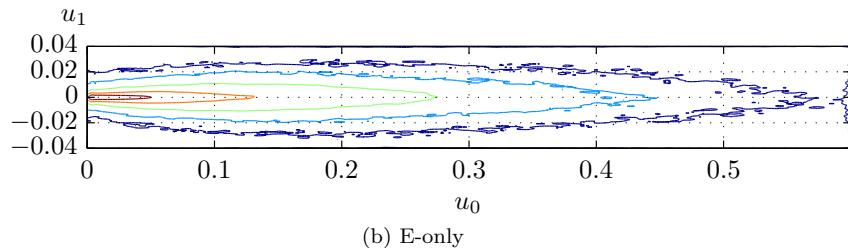
4.3.3 Proposed Method

Our proposed method uses two observations from our benchmark dataset to build on the wedge-finding method presented above. First, real E-only images exhibit lower contrast when compared to digitally separated E-only images using the wedge-finding method (compare the right half of Figure 4.1b and Figure 4.1d to see this). Second, the optical density of some regions of the H&E image is significantly higher than the sum of the optical densities of the corresponding H-only and E-only image. This points to an interaction between the stains beyond summing in the OD space. Our proposed method is motivated by these observations.

Contrast of Eosin. Our first improvement to the wedge-finding method is to reduce the contrast of the E-only image. To better quantify the observation that real E-only images have low contrast, we examined the order statistics of the norms of the pixel values of our ground truth single-stain images in OD space. We found that the order statistics change smoothly and that for each pair of H- and E-only images, the order statistics of the two single-stain images were similar except for a multiplicative constant, $\|f_{\text{E}}^{OD}(x_i)\| \approx C \|f_{\text{H}}^{OD}(y_i)\|$, where x_i denotes the pixel location of the i th smallest element in f_{E}^{OD} and y_i denotes the pixel location of the i th smallest element in f_{H}^{OD} . Performing the same investigation on H- and E-only images created



(a) H-only



(b) E-only

Figure 4.3: Contour plots indicating pixel density as a function of position along u_0 and u_1 , the first two singular vectors of the matrix of pixel colors for (a) H-only and (b) E-only images. The highest density of points for both stains is at the origin, which corresponds to white. If the stains were truly linear, the pixel matrix would be rank one and therefore all pixels would lie only along u_0 . For both hematoxylin and eosin this is not the case, indicating that neither stain is completely linear, however the deviation is more significant for hematoxylin.

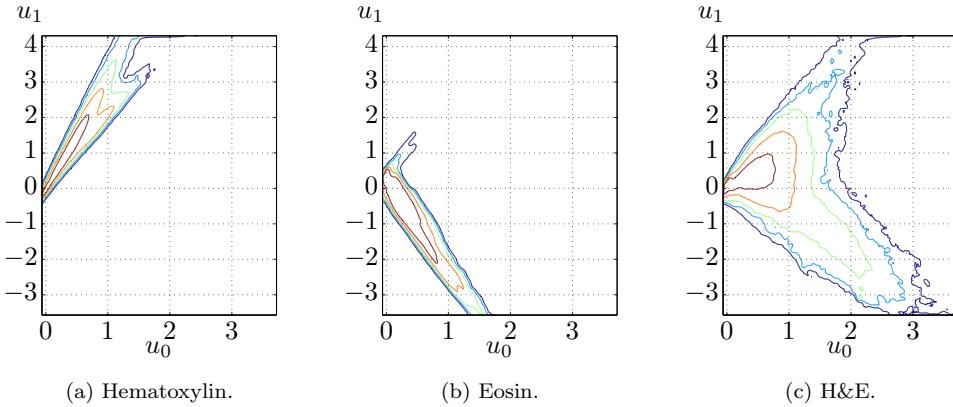


Figure 4.4: Contour plots indicating pixel density as a function of position along u_0 and u_1 , the first two singular vectors of the matrix of pixels in the H&E image. Each pixel in the H&E image should be the sum of a pixel from the H-only and E-only image; however some of the density of the H&E image lies in the lower-right corner of the plot, which cannot be formed in this way.

digitally using the wedge-finding method, we noted significant deviation from the smooth behavior for the E-only estimate.

Therefore, starting with \hat{a}_{HE} found via the wedge-finding method, we estimate C as the median of $\hat{a}_{\text{E}}(x_i)/\hat{a}_{\text{H}}(y_i)$. To correct the contrast of the E-only image, we let

$$\hat{a}_{\text{E}}^*(x_i) = (1 - \gamma)\hat{a}_{\text{E}}(x_i) + \gamma C \hat{a}_{\text{H}}(y_i),$$

where γ controls the degree of smoothing: when $\gamma = 0$, the contrast reduction has no effect; when $\gamma = 1$, the eosin coefficients are shifted so their order statistics are exactly the same as the hematoxylin coefficients times a constant. We can then compute a new stain separation

$$\hat{f}_{\text{E}}^{OD} = \hat{\varphi}_{\text{E}} \hat{a}_{\text{E}}^*, \quad \hat{f}_{\text{H}}^{OD} = \hat{f}_{\text{HE}}^{OD} - \hat{f}_{\text{E}}^{OD}.$$

Stain Interaction. The method described in the previous section results in improved estimates of the E-only image and, when γ is low, it also improves the H-only estimate by adding some pink which is indeed present in the ground truth (Figure 4.1b). When γ is high, the H-only estimate begins to suffer because it contains too much pink. According to (4.1), however, whenever the E-only estimate improves, the H-only estimate should too, because they exactly sum to the H&E image. We therefore extend (4.1) by adding an interaction term,

$$f_{\text{HE}}^{OD}(x) = \varphi_{\text{H}} a_{\text{H}}(x) + \varphi_{\text{E}} a_{\text{E}}(x) + I(x).$$

In general, $I(x)$ could have a complicated dependence on $a_{\text{H}}(x)$, $a_{\text{E}}(x)$, and the underlying tissue, making it hard to calculate directly. Instead, we assume that interaction occurs only in those places where the contrast of the eosin coefficients is adjusted. We let some fraction λ of the change in eosin be due to that interaction and the rest be due to pinkness within the H-only image, giving our final stain separation expression,

$$\hat{f}_{\text{E}}^{OD} = \hat{\varphi}_{\text{E}} \hat{a}_{\text{E}}^*, \quad \hat{f}_{\text{H}}^{OD} = (1 - \lambda)(\hat{f}_{\text{HE}}^{OD} - \hat{f}_{\text{E}}^{OD}) + \lambda \hat{\varphi}_{\text{H}} \hat{a}_{\text{H}}.$$

When λ is high, most of the adjustment in the E-only image is assumed to be due to the interaction term, so \hat{f}_{H}^{OD} is unaffected; when λ is low, the eosin adjustment is assumed to be due to pinkness in the H-only image, so \hat{f}_{H}^{OD} changes a lot.

4.3.4 Stain Normalization

After stain separation, we can normalize images using an approach similar to that of [10]. Given two H&E images, our goal is to normalize the staining of the input image to match the target image. To do this, we stain separate both input and target images and represent each (OD space) pixel in the single-stain estimates as a scalar weight times a unit-length stain vector. For the color inversion technique [9], we find the the stain vectors and weights via 1D SVD on the single-stain images. For the wedge-finding method [10], the stain vectors are $\hat{\varphi}_{\text{H}}$ and $\hat{\varphi}_{\text{E}}$ and the weights are \hat{a}_{H} and \hat{a}_{E} . For our method, we also use $\hat{\varphi}_{\text{H}}$ and $\hat{\varphi}_{\text{E}}$ as the stain vectors, but use \hat{a}_{E}^* for the eosin weights and $\|\hat{f}_{\text{H}}^{OD}(x)\|$ for the hematoxylin weight at each pixel x . To adjust for variations in stain intensity, we independently scale the input eosin and hematoxylin weights so that their 95th percentile values match the 95th percentile values of the corresponding weights from the target image. To adjust for variations in stain color, we replace the input stain vectors with those of the target. We complete the normalization process by recombining the single-stain images of the input.

4.4 Experiments and Discussion

We validate our method with four experiments: (1) We compare our stain separation method to the color inversion [9] and wedge finding [10] methods quantitatively, using our newly created benchmark dataset (described in Section 4.2). (2) We again use the benchmark dataset to explore the effects of our method's three parameters. (3) We compare the performance of the same three methods on a stain normalization task. (4) We explore whether stain normalization improves automated diagnosis. We now describe these experiments and discuss their results in detail.

	Method					
	color inversion [9]		wedge finding [10]		proposed	
	H	E	H	E	H	E
Image 1	26.13	25.54	26.07	25.34	27.31	26.42
Image 2	23.39	22.89	23.29	21.54	23.56	23.04
Image 3	25.20	26.04	28.48	24.89	29.56	27.32
Mean	24.91	24.82	25.95	23.92	26.81	25.59

Table 4.1: Results of the stain separation comparison reported as SNR [dB] for hematoxylin and eosin, with the best result for each image and stain in bold. The proposed method improves SNR by around one dB.

4.4.1 Stain Separation Comparison

For each of the three methods, we performed a stain separation on each of the three images H&E images in our dataset. We set parameters at typical values: for the wedge finding, $\alpha = 0.5$; for our method, $\alpha = 0.5, \gamma = .75, \lambda = 0.5$. We explore parameter selection in depth in the next section; here our goal was to compare the methods with default parameters, since in the real use case for stain separation no ground truth exists with which to tune parameters. To quantitatively evaluate the results, we use affine registration to align each ground truth H- or E-only image with the corresponding H&E image. Then, for each digital separation result, we blur the ground truth image and the result with a Gaussian ($\sigma = 15\text{px}$) and calculate the signal to noise ratio (SNR); this blurring is intended to reduce the effect of small misalignments between the two images being compared, it does not affect the trend in the results.

Figure 4.1 shows the qualitative results on a single image and Table 4.1 shows the results in terms of SNR. Both our method and the color inversion method result in low contrast for the E-only estimate; but our method additionally captures some of the pink tone in the H-only estimate, especially in the bone tissue (long regions in the center of the image). Quantitatively, our proposed method outperforms the others for each image and stain.

4.4.2 Parameter Analysis

In order for our method to be really useful, its strong performance must hold over a wide range of parameters. To explore whether this is true, we performed a parameter analysis over the wedge percentile α , the eosin contrast reduction γ , and the interaction amount λ for the first image in our dataset (first row of Table 4.1).

We began by investigating α and γ with λ set to zero (Figure 4.5). Focusing on the $\gamma = 0$ case (solid lines), we see that α does not affect performance much unless it is too small; a setting of 0.5 or slightly less is optimal for this image, but larger values also work. This is intuitive because setting α to be very small would allow outliers to greatly affect the estimation of the hematoxylin and eosin vectors. Increasing γ (from solid to dashed to dotted lines) always improves the estimate for the E-only image, but at the cost of the H-only estimate for high values. Intuitively, the H-only image gets too pink when γ is high. The mean SNR shows that $\gamma = 1$ (dotted line) is still a good choice for all values of α .

Next, we explore how increasing the interaction amount λ can improve the estimate of the H-only image. Starting at the point that gives highest mean SNR ($\alpha = 0.3, \gamma = 1$) in the previous experiment, we vary λ (Figure 4.6). For any λ between 0 and 0.8, the SNR of the H-only estimate improves, with a maximum improvement of .47dB at $\lambda = 0.4$. Taken together, these parameter studies show that the proposed method is robust with respect to parameter choices.

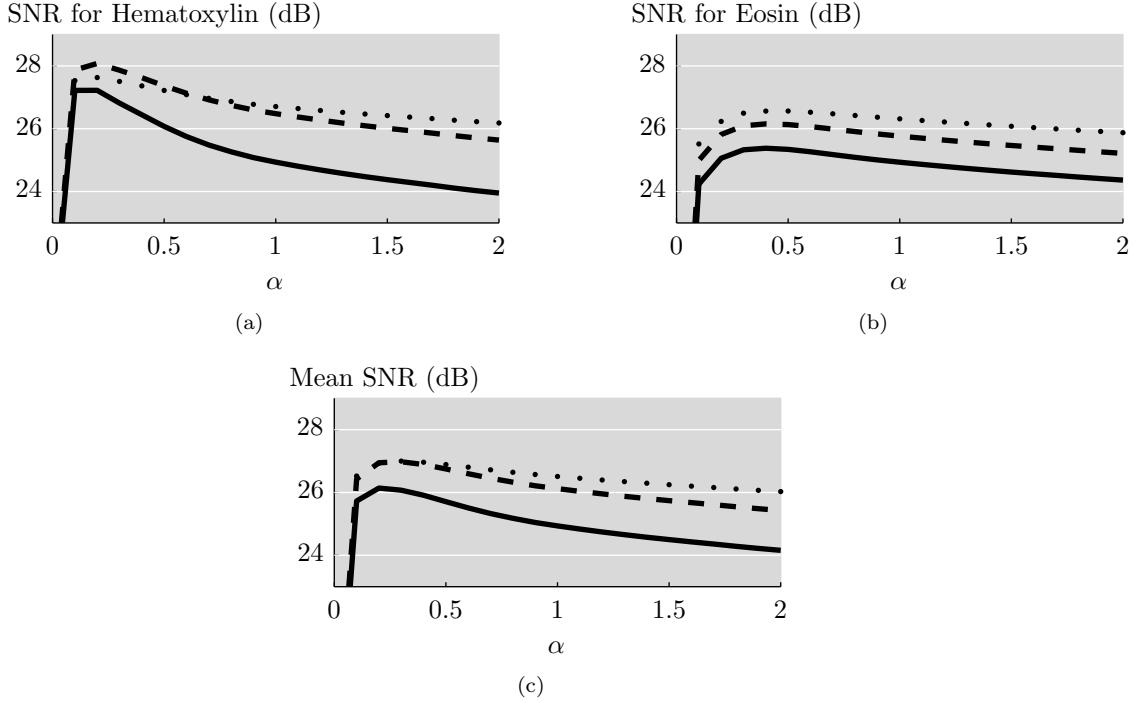


Figure 4.5: Effects of wedge percentile α and eosin contrast reduction γ (solid, dashed, dotted lines corresponding to $\gamma = 0, 0.5, 1$) on (a) the H-only reconstruction, (b) the E-only reconstruction, and (c) their mean. When $\gamma = 0$, our method reduces to wedge finding [10]; setting γ to 0.5 improves both the H-only and E-only reconstruction. Setting γ to 1 improves the E-only reconstruction at the expense of the H-only reconstruction.

4.4.3 Stain Normalization Comparison

We now compare the performance of the stain separation methods on a normalization task. For each method, we used the process described in Section 4.3.4 to normalize an input image (Figure 4.1a) to a target image (Figure 4.7b). Qualitatively, the normalization results (Figure 4.7) are similar among the wedge-finding, color inversion, and our proposed method.

Quantifying the results of this experiment is difficult because simply subtracting the normalized and target images yields no useful information, since the images are of different tissue areas. We therefore seek an image similarity metric that compares the colors occurring in each image without accounting for their location or frequency. To accomplish this, we generate a signature (along the lines of [93]) for the normalized and target image by clustering their pixels using k-means in the Lab color space, which we selected for its perceptual uniformity. We then calculate an optimal linear assignment (with the MATLAB code [94]) between the sets of cluster centers. We use the total cost of the optimal assignment as our measure of distance. Using ten cluster centers, we find the total costs of assignment to be 122.4, 96.3, and 51.1 for color inversion, wedge-finding, and our method, respectively. Although this experiment is small and the proper distance metric is an open question, these results at least suggest that our method's improved stain separation performance translates into more accurate stain normalization as well.

4.4.4 Normalization and Diagnosis

In a final experiment (unpublished), we aim to determine whether or not stain normalization is actually a useful preprocessing step in a larger histology image analysis system. To do this, we repeat the colitis

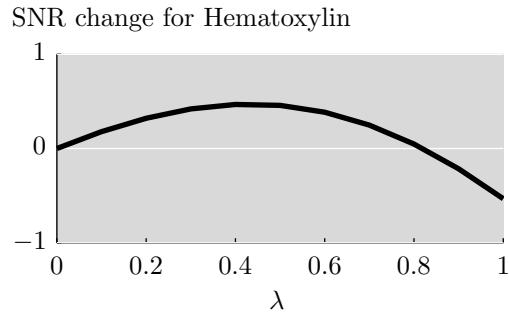


Figure 4.6: Effects of interaction amount λ for wedge percentile $\alpha = 0.3$ and eosin contrast reduction $\gamma = 1$. A wide range of λ values all improve the SNR of the H-only estimate.

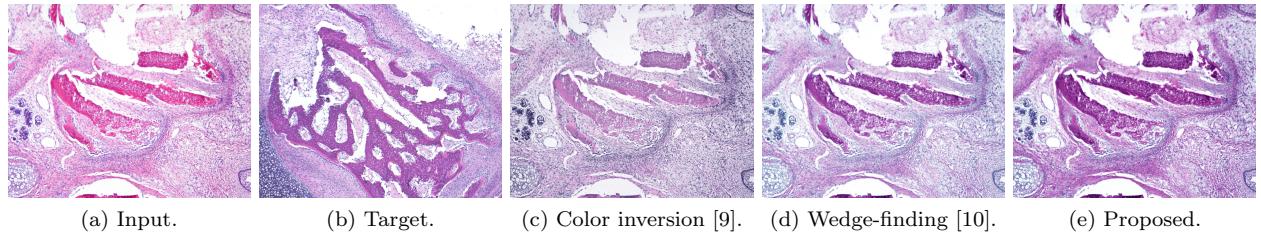


Figure 4.7: Results of the stain normalization experiment, showing (a) the input, (b) the target, and the results using (c) color inversion [9], (d) wedge-finding [10], and (e) our proposed method. The results are similar across methods; the main difference is the depth of color in the bone tissue (long purple structure in (c)-(e), best viewed in color).

detection experiment described in Section 3.9.1 using the WND-CHRM classifier on three new versions of the colitis dataset: (1) The *illumination-corrected dataset* has had its illumination corrected via linear regression. (2) The *stain-varied dataset* has had its stain colors and intensities jittered to simulate stain variability between labs. (3) The *stain-normalized dataset* has had its stains normalized according to the method described in this chapter to the median stain colors and intensities of the set. We compare these to the *raw dataset* which was used in Section 3.9.1. See Figure 4.8 for sample images from each of these four datasets.

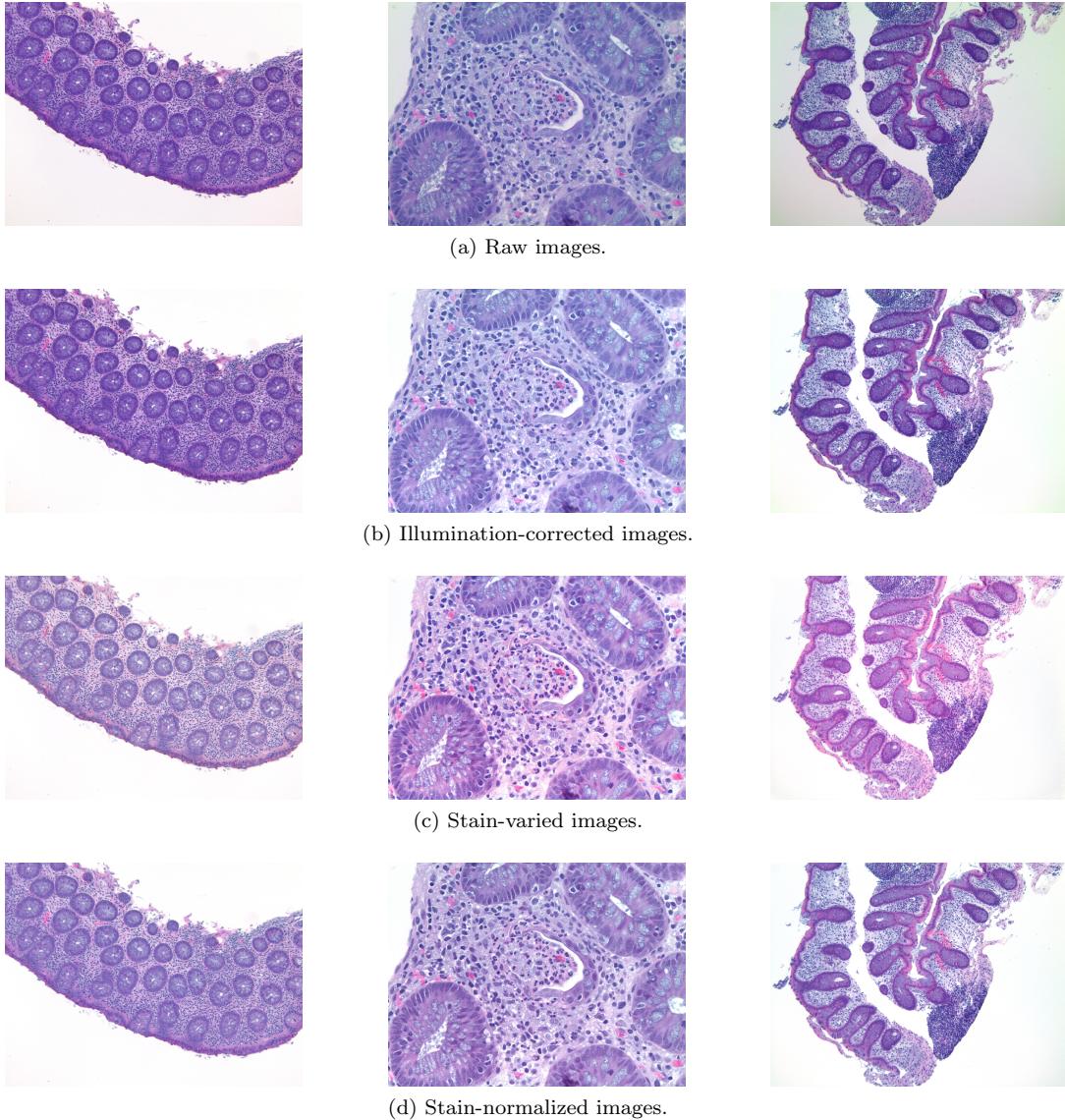


Figure 4.8: Examples of (a) the original colitis dataset images, (b) their illumination-corrected versions, (c) their stain-varied versions, and (d) their stain-normalized versions.

The results of our normalization and diagnosis experiment are given in Table 4.2. Accuracy for the raw, illumination-corrected, and stain-normalized datasets is similar, which makes sense because the raw images were all stained and imaged by the same lab with well-controlled lighting conditions, meaning that correcting for illumination and staining variation has little effect. On the other hand, the simulated staining

variability in the stain-varied dataset is large enough to negatively impact the accuracy of the classifier. These results do suggest that stain normalization can improve automated analysis accuracy, but a larger experiment involving images collected from multiple labs is necessary to show whether these results will hold for real-world staining variation.

Magnification	Dataset			
	raw	illumination	varied	normalized
40×	100.0	100.0	97.5	97.5
100×	100.0	82.5	87.5	95.0
200×	85.0	95.0	82.5	90.0
400×	85.0	72.5	67.5	90.0

Table 4.2: Comparison of classification accuracy (in %) of the WND-CHARM classifier on raw, illumination-corrected, stain-varied, and stain-normalized images. The best accuracy for each magnification is shown in bold. The stain-normalized dataset gives better results than the stain-varied dataset, suggesting that staining variability does negatively impact computer-aided diagnosis.

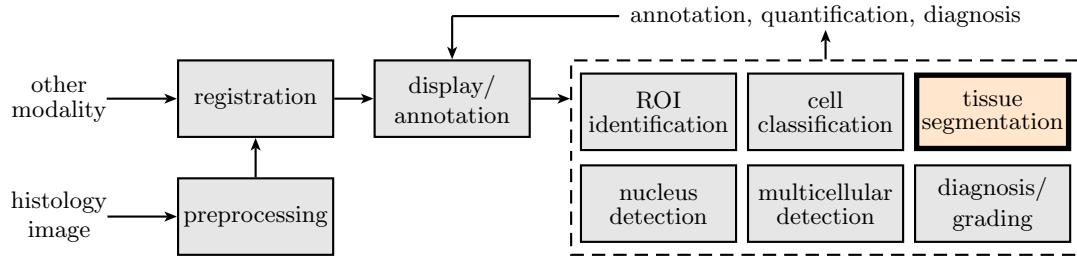
4.5 Conclusions and Future Work

In this chapter, we have presented a new algorithm and benchmark dataset for digital H&E stain separation. The benchmark provides a way to quantitatively evaluate digital stain separations via images of chemically destained slides. We showed that our algorithm outperforms the currently available methods on all test images and is robust to parameter selection. We also showed that this improved separation leads to improved stain normalization.

In our future work, we will explore more principled priors on the E-only contrast, perhaps by finding a parametric description of the order statistics. We would also like to continue to improve the image formation model. One approach would be to explore more complex lighting models including reflection in addition to absorbance; another would be to study the interaction term more closely. All of this work would benefit from chemical experiments including a wider variety of stain brands, staining procedures, lighting conditions, and tissues.

Chapter 5

Tissue Segmentation: ORTSEG



Tissue segmentation is a fundamental part of many automated histology systems, but generic image segmentation techniques often fail to produce meaningful results on histology images. In this chapter, which is based on our work published in [17], we present a new model for histology images that helps explain why they are challenging to segment, and then present a new segmentation method based on this model. Results on synthetic texture mosaics and real histology images show the promise of the method.

5.1 Introduction

Image segmentation is the process of locating the boundaries between visually distinct regions in an image, thereby partitioning the pixels. Applications of image segmentation are numerous: from remote sensing [95] and video processing [96] to non-destructive testing [97]. Within the biomedical field, segmentation is used with diverse imaging modalities such as MRI [98], both light microscopy [99–101] and electron microscopy [102], ultrasound [103], and many others to identify regions at all scales from organelles to organisms.

5.1.1 Previous Work

There exists a wide variety of classic approaches to generic image segmentation, including graph cuts [7], active contours [104], level sets [105], Gabor filtering and clustering [106], random fields [107], watersheds [108], region growing [109], and mean shift [110]. In addition, there also exist engineered segmentation systems, such as BlobWorld [111], JSEG [5], EDISON [1], and CTex [112]. Broadly, these methods vary in how the segmentation regions are parametrized and whether edge or region information is used.

Even with this plethora of methods to try, working on a new segmentation problem is not trivial. Each existing method makes assumptions about the images it aims to segment. When these assumptions are met, the method works. When they are not, it fails. Because many of these assumptions are implicit, selecting a method to use on a new segmentation problem involves educated guessing. When no suitable method can be found, a new method is designed. This is especially true for biomedical imaging, where the diversity of imaging modalities and analysis objectives means that new algorithms are often designed for each specific application. For example, the active shape models in [113] aim to segment the complex

shapes in computed tomography (CT) scans, the active masks in [100] work for the punctate appearance of fluorescence microscopy images, and the multiple model approach in [103] handles the problems of speckle and motion in ultrasound images of the heart.

Here, we were inspired by the problem of segmenting tissues in bright field microscopy images of H&E-stained slices of teratoma tumors (see Figure 5.1 for examples; more details on this dataset can be found in [114]). Previous work in our lab [115] began exploring models for these complicated tissues and proposed a simple supervised segmentation method for them. In a separate line of work [79], we proposed image features based on the local density and color of cellular structures such as nuclei. Other works that address segmentation of histology images are reviewed in Section 3.8.

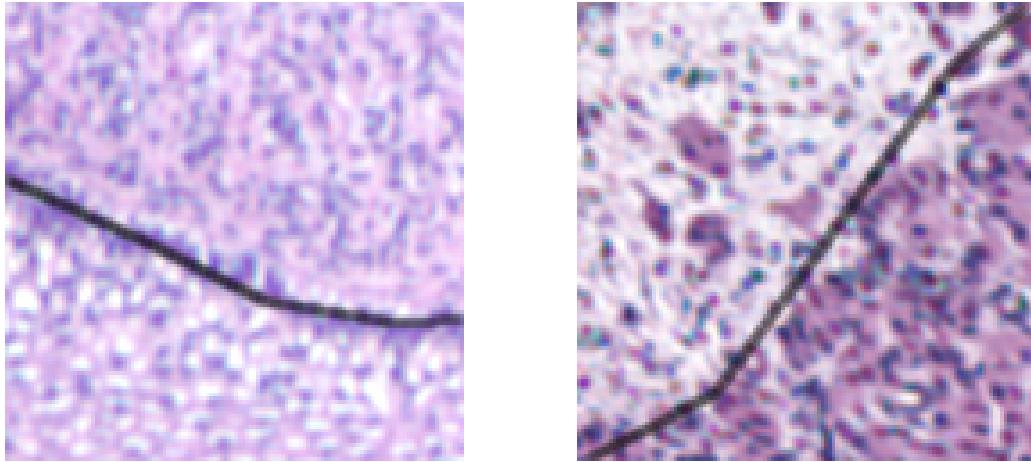


Figure 5.1: Examples of difficult tissue boundaries in teratoma images (black lines denote tissue boundaries as drawn by a pathologist). The lack of distinct edges between regions makes automated segmentation difficult, and even experts cannot reliably localize the boundaries at pixel level.

5.1.2 Current Work

Our working hypothesis is that segmenting histology images is difficult because tissues are complicated, while tissue boundaries are sometimes subtle and not marked by edges. To formalize this idea, we propose a new model for images formed from multiple tissues. Working from this model, we design a new unsupervised segmentation method based on local histograms that is well suited to a broad class of images with poorly defined boundaries. We expect this class to include images such as aerial photos of grass and trees (“Where is the edge of the forest?”), photographs of crowds (“Where is the edge of the crowd?”), and, as is our focus here, histology images (“Where is the edge of the adipose tissue?”).

The main contributions of this chapter are as follows:

1. We propose a *mathematical framework* for image segmentation which models images as occlusions of textures. Given an image formed according to this model, we prove that its local value histograms will approximately be convex combinations of the value distributions of its component textures.
2. Based on this result, we present a new *algorithmic framework* for image segmentation based on histogram factorization and deconvolution.

The outline of the chapter is as follows: In Section 5.2, we introduce local histograms, occlusions, and necessary notation. We then describe our proposed model and present the main theorem. In Section 5.3, we propose an algorithmic framework for image segmentation based on this theorem and describe one implementation of it. We present experimental results of a comparison of our algorithm to other segmentation algorithms in Section 5.4 and conclude in Section 5.5.

5.2 Mathematical Framework

In this section, we introduce necessary notation, then build the texture model as follows: We first define dependence decay, which describes how well a texture can be characterized by its local histograms, show how to construct textures with a known dependence decay, and explore dependence decay in real textures. Based on this, we propose modeling images as occlusions of textures. Under this model, we present our main result, which describes how the local histograms of this class of images are mixtures of the value distributions of their component textures.

5.2.1 Notation and Background

Let f be an image, $f : X \rightarrow V$, where X is a discrete set of pixel locations and V is a discrete set of pixel values. For example, for a 512×512 8-bit grayscale image, $X = \mathbb{Z}_{512} \times \mathbb{Z}_{512}$ (we use \mathbb{Z}_K to denote the set of integers modulo K) and $V = \mathbb{Z}_{256}$, while for a stack of ten 1600×1200 8-bit RGB color images, $X = \mathbb{Z}_{1600} \times \mathbb{Z}_{1200} \times \mathbb{Z}_{10}$ and $V = \mathbb{Z}_{256} \times \mathbb{Z}_{256} \times \mathbb{Z}_{256}$. Clearly, the elements of V can be vectors or scalars; since the set is always discrete, however, without loss of generality we treat it as a set of integers $0, 1, \dots, |V| - 1$.

We now define the local histogram and occlusion operators (our definitions are a slight adjustment of those presented in [115]). Define first the indicator function,

$$1_A(x) = \begin{cases} 1, & \text{for } x \in A; \\ 0, & \text{for } x \notin A. \end{cases}$$

Then, the *local histogram* transform of an image f is

$$L_w f(x, v) = \sum_{x' \in X} 1_{\{v\}}(f(x')) w(x - x'), \quad (5.1)$$

for $x \in X$ and $v \in V$, and where w is an averaging filter that sums to one. For example, when w is a constant averaging filter of size 3×3 , the local histogram transform $L_w f(x, v)$ gives the fraction of the pixels in the 3×3 neighborhood around x that have value v .

We call a function $\sigma : X \rightarrow \{0, 1, \dots, N - 1\}$ a *labeling function*. We define the *occlusion* of a set of images $\{f_n\}_{n=0}^{N-1}$ with respect to a labeling function σ as

$$O_\sigma \{f_n\}_{n=0}^{N-1}(x) = \sum_{n=0}^{N-1} 1_{\{n\}}(\sigma(x)) f_n(x) \quad (5.2)$$

for $x \in X$. In other words, an occlusion of a set of images is itself an image that matches one of its component images at each pixel; which image “shows through” at each pixel is governed by σ . Note that we use the term *occlusion* because we view the set of images as a stack with one image blocking, or occluding, the others at each pixel.

5.2.2 Flat Textures

We now define a texture as we use it in our work; note that it is simply a random-valued image and that many other texture definitions exist.

Definition 1 (Texture). *A texture $F = \{F(x)\}_{x \in X}$ is a V -valued random field indexed by pixel locations, X .*

In other words, for each $x \in X$, $F(x)$ is a V -valued random variable; one realization of a texture is an image. Associated with each of these random variables is a probability mass function, $p_{F(x)} : V \rightarrow [0, 1]$; the probability that $F(x)$ is equal to v is $p_{F(x)}(v)$. While each $F(x)$ may have its own unique probability mass function, we find it useful to consider only those textures for which these functions are the same, which we call *flat textures*.

Definition 2 (Flat texture). A texture F is flat when $p_{F(x_i)}(v) = p_{F(x_j)}(v)$, for all $x_i, x_j \in X$ and $v \in V$.

In other words, a flat texture is one where the distribution of values is the same at every pixel, and we can therefore refer to the value distribution of the texture, $p_F(v)$, without dependence on x . Note that this does not imply that the random variables $F(x_i)$ are independent, only that they have the same distribution. Flat textures may have complex and spatially varying patterns of dependence; we will further explore this dependence in the following sections.

Flatness is not a highly restrictive condition. In fact, it is a property of most real-world textures. For example, take F to be the texture for 128×128 color images of grass. Intuitively, we would assign one color distribution to F , with peaks for brown, yellow, and green. We would not say that grass images are more likely to be brown in the bottom right corner or green at the top. The fact that colors do not appear more or less frequently in specific areas of the image means that this texture is flat.

5.2.3 Histograms of Flat Textures

Many textures can be distinguished based solely on their value distribution, $p_F(v)$. For example, consider a grass texture and a sky texture; $p_{\text{grass}}(v)$ would have a peak around green, while $p_{\text{sky}}(v)$ would have peaks around blue and white. If our goal is to classify whether an image is of grass or sky, we could build a value histogram from the pixels in the image and compare it to our model of $p_{\text{grass}}(v)$ and $p_{\text{sky}}(v)$.

We now consider more closely the relationship between the histograms of a realization of a flat texture and that texture's value distribution, $p_F(v)$. Let

$$\hat{p}_f^{\{x_i\}_{i=0}^{k-1}}(v) = \frac{1}{k} \sum_{i=0}^{k-1} 1_{\{v\}}(f(x_i)) \quad (5.3)$$

be a histogram built from k pixels at the unique locations x_0, x_1, \dots, x_{k-1} of the image f . If k is large enough, we expect that $\hat{p}_f^{\{x_i\}}(v)$ will estimate $p_F(v)$; intuitively, sampling f over a small region will give a worse estimate than sampling f over a large one. The quality of the estimate will be determined by the number and position of the pixels and the way the statistical dependence between pixels of F changes over space. Moreover, for some textures, such an estimate may never be good. We quantify how quickly these estimates improve by introducing the notion of *dependence decay*.

Definition 3 (Dependence decay). A flat texture F exhibits dependence decay with complexity a if, with probability $1 - \delta$ and for any $v \in V$ and unique $x_0, x_1, \dots, x_{k-1} \in X$,

$$|\hat{p}_f^{\{x_i\}_{i=0}^{k-1}}(v) - p_F(v)| \leq ak^{-1/2},$$

with $0 < a$ and $0 < \delta \ll 1$.

The value δ is a small positive number so that $1 - \delta$ is our standard of high probability. If the complexity, a , is small, then F is simple in the sense that even histograms formed from a small number of pixels approximate p_F well. (Note: using this definition, a texture may have multiple complexities, a_0, a_1, \dots ; when we refer to the complexity of texture, we generally mean the smallest a satisfying Definition 3.)

5.2.4 Constructing Dependence-Decay Textures

Though any texture can exhibit dependence decay with a very loose bound (for example, select a such that $a|X|^{-1/2} > 1$), we focus on the class of textures for which the dependence between pixels decays sufficiently so that this bound is useful, that is, $ak^{-1/2} \ll 1$ even when $k \ll |X|$; informally, we will refer to these as *dependence-decay textures*. At this point, it is natural to wonder whether there exist any dependence-decay textures. To find the answer, we explore a property of textures we call *persistence length* and show that when a texture's persistence length is short enough, it is dependence-decay.

Definition 4 (Persistence length). A texture F has a persistence length ϵ if, for any two pixel locations, x_i and x_j , the random variables $F(x_i)$ and $F(x_j)$ are independent whenever $\|x_i - x_j\|_1 > \epsilon$.

Persistence length is another way of thinking about the scale of a texture. For example, consider a texture F that generates images of colored pebbles. If we know that the value of a sample of this texture at pixel location x_i is blue, we can expect the value at pixel location x_j to be blue as well when x_i and x_j are close; when these locations are far enough apart, however, $f(x_i)$ no longer gives any predictive information about $f(x_j)$. The distance at which $f(x_i)$ stops being helpful in predicting $f(x_j)$ is F 's persistence length. In this example, it is the diameter of the pebbles.

Persistence length is related to dependence decay, since statistically independent samples are the best ones with which to estimate a texture's value distribution. We make this relationship explicit with a theorem; the proof is given in Appendix 5.A.

Theorem 1 (Persistence length and dependence decay). A flat texture with persistence length ϵ exhibits dependence decay with complexity $a = (-2A_\epsilon \log \frac{1}{2}(1 - \sqrt[4]{1-\delta}))^{1/2}$, with $A_\epsilon = 2\epsilon^2 + 2\epsilon + 1$.

If we could create textures with a given persistence length, this theorem would allow us to establish their dependence decay properties as well. One straightforward way of doing this is to use a dead leaves model; that is, generate images by layering templates of random size, shape, position, and value on top of each other (see Figure 5.2 for examples and [116] for a more thorough description). Textures defined in this way have a persistence length equal to the ℓ^1 -diameter of their largest template, since pixels separated by more than this distance cannot belong to the same template and are therefore independent.

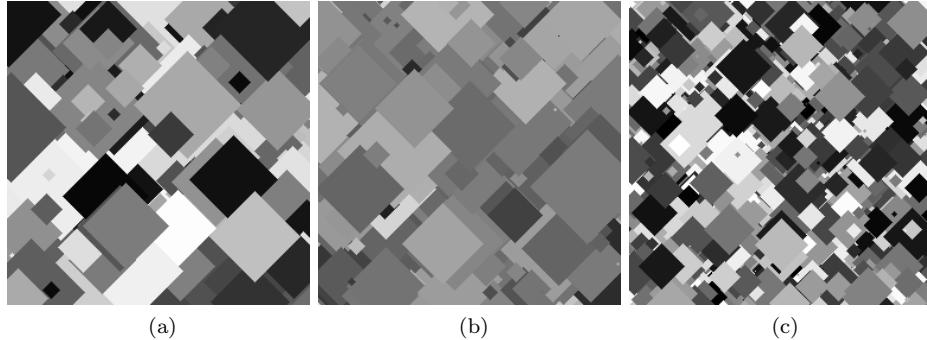


Figure 5.2: Examples of dead leaves images with different maximum template sizes and value distributions. The templates in (a) and (b) have twice the maximum diameter of those in (c). The value distributions in (a) and (c) are uniform, while the value distribution in (b) is normal.

We also note that Theorem 1 will overestimate the complexity, a , of many textures. This is because the concept of persistence length only considers complete independence. In a real texture, the dependence between pixel values may decay rapidly but only reach true independence over long distances (or never). In this case, the persistence length of the texture will be large and the complexity from Theorem 1 will also be large, but estimates of p_F will actually improve quickly as the sample size increases.

5.2.5 Dependence Decay in Real Images

We now discuss how the above theory applies to real images. As our example textures, we take grayscale images from the Prague Texture Segmentation Benchmark [11]. For each image in the dataset, we define a corresponding F that assigns equal probability to this image and all its circular translates. This means $p_F(v)$ is equal to the global histogram of the image. To show that F is a dependence-decay texture we must set a probability δ (in our experiment, $\delta = 0.05$), then for each $k = 1, 2, \dots$, find the arrangement of points,

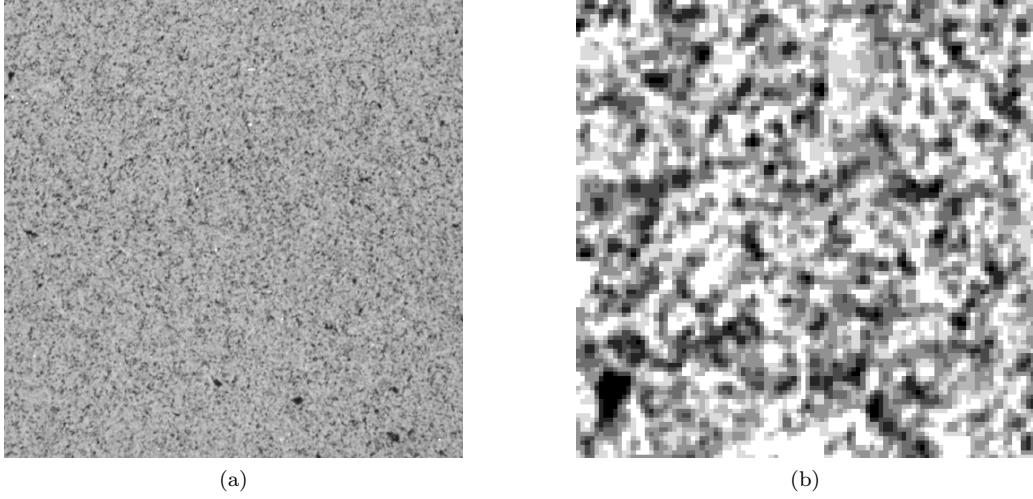


Figure 5.3: (a) Example texture from the Prague set [11]. (b) Cropped and quantized version used in our dependence decay analysis.

x_0, x_1, \dots, x_{k-1} , that maximizes $|\hat{p}_f^{\{x_i\}}(v) - p_F(v)|$ in at least a fraction δ of the realizations of F . If this maximum error,

$$E(k) = \max_{x_0, x_1, \dots, x_{k-1}} |\hat{p}_f^{\{x_i\}_{i=0}^{k-1}}(v) - p_F(v)| \quad (5.4)$$

can be bounded usefully by a function $ak^{-1/2}$, then F is a dependence decay texture.

In practice, we cannot check every possible arrangement of k pixels even for small k , so we make the simplifying assumption that, in the worst case, the x_i will be eight-connected and perform a greedy search for worst-case arrangements. This assumption is reasonable for natural images because adjacent pixels are nearly always more correlated than those that are far apart. To further speed up the computation, we crop the image to a size of 100×100 pixels and quantize it to eight values (Figure 5.3).

Our experiments show that many of these images do exhibit dependence decay, see Figure 5.4 for an example. This supports the claim that many real-world textures exhibit dependence decay. For the periodic textures in the dataset, the complexity a is large because pixels are highly correlated, see Figure 5.5 for an example. If the search were not constrained to contiguous regions of pixels, the complexity would be even higher. This is because when the offset between samples matches the period, the samples are completely dependent on each other. We will see in the next section that the high a value associated with periodic textures means that they are not usefully governed by our main result, Theorem 2. This essentially says that periodic textures can be combined into pathological images that are impossible to segment. We will stress here, however, that periodic textures can still be segmented by the algorithm we present in Section 5.3, because the worst case error only happens with very specific corner cases that are unlikely to occur in practice.

5.2.6 Occlusions of Dependence-Decay Textures

We now come to the main result of the chapter. We propose modeling images as occlusions of realizations of textures, $O_\sigma\{f_n\}_{n=0}^{N-1}$, where the component textures, F_0, F_1, \dots, F_{N-1} , and the labeling function, σ , are not known. Under this model, segmentation is equivalent to recovering σ .

Figure 5.6 illustrates modeling a natural image as such an occlusion. Starting with an original image as in Figure 5.6a, imagine there exist three textures, F_{sky} , F_{window} , and F_{brick} , each generating images of the corresponding texture, as in Figure 5.6b. We can then use a labeling function σ (Figure 5.6c) to create a new image $f_{\text{lighthouse}} = O_\sigma\{f_{\text{sky}}, f_{\text{window}}, f_{\text{brick}}\}$ (Figure 5.6d).

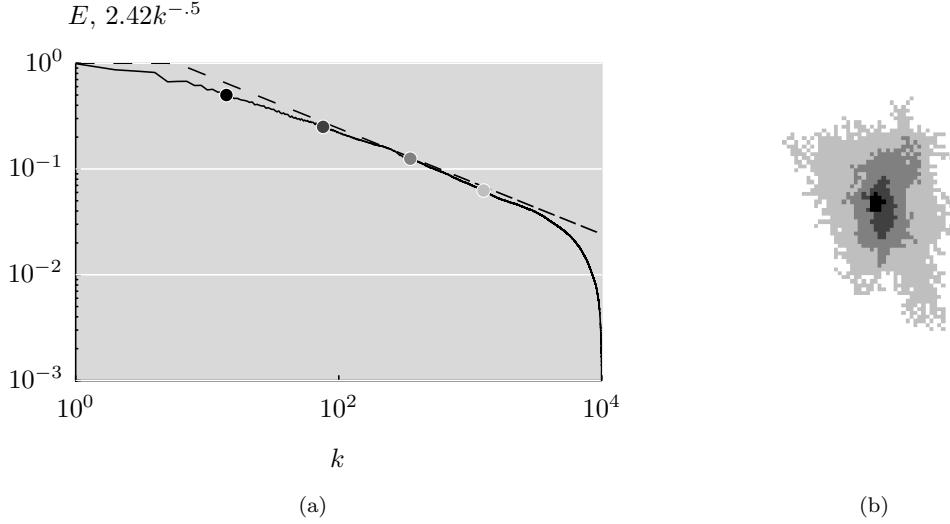


Figure 5.4: Dependence decay for the image in Figure 5.3. The plot in (a) shows how the approximation error (5.4) (solid line) is bounded by the function $2.42 k^{-1/2}$ (dashed line). The regions shown in (b) are four worst-case arrangements (described in Section 5.2.5). When pixels sampled in these patterns are used to estimate the color distribution of the texture, they lead to errors of 2^{-1} , 2^{-2} , 2^{-3} , and, 2^{-4} (decreasing error from black to light gray). The corresponding points are marked with circles in (a).

If we use occlusions of textures as our model for images, then we can define segmentation as the problem of taking an image and recovering the labeling function that was used to generate it. If the component textures of the image are dependence-decay, then a reasonable approach is to examine the local histograms of f , since these histograms well approximate the true distribution of values in each texture. For example, in Figure 5.6a, local histograms taken from the upper right corner will approximate p_{sky} , while those from the lower left corner will approximate p_{brick} . Local histograms taken around the window will be mixtures of p_{brick} and p_{window} . The following theorem formalizes this process; the proof is given in Appendix 5.B.

Theorem 2 (Local histograms of occlusions are mixtures of component value distributions). *Given a set of dependence-decay textures $\{F_0, F_1, \dots, F_{N-1}\}$ with complexities a_0, a_1, \dots, a_{N-1} , a constant averaging window w , and a labeling function σ , then with probability $(1 - \delta)^N$, and for any $x \in X$ and $v \in V$,*

$$\begin{aligned} & \left| L_w O_\sigma \{f_n\}_{n=0}^{N-1}(x, v) - \sum_{n=0}^{N-1} [w * 1_{\{n\}}(\sigma)](x) p_{F_n}(v) \right| \\ & \leq a \left(\frac{|w|}{N} \right)^{-1/2}, \end{aligned} \quad (5.5)$$

with $|w| = \sum_{x \in X} 1_{(0,1]}(w(x))$, $a = \max\{a_0, a_1, \dots, a_{N-1}\}$, and δ the threshold for high probability selected in Definition 3.

Here we use the notation $(0, 1]$ to indicate the interval $\{x \mid 0 < x \leq 1\}$.

In other words, if an image is formed from the occlusion of realizations of several dependence-decay textures, then each of its local histograms will be approximately a convex combination of the true value distributions of those textures. The amount that each texture contributes to a histogram is proportional to how much of that texture exists in the neighborhood of the local histogram. This theorem offers a key insight about how to do segmentation because it relates the local histograms of f , which we can calculate, to the labeling function σ , which we want to discover.

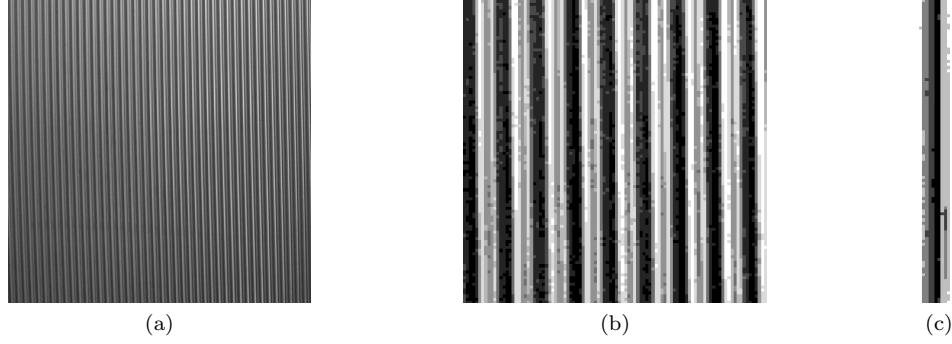


Figure 5.5: (a) An example of a periodic texture from the Prague set [11]. (b) Cropped and quantized version used in our dependence decay analysis. (c) Four worst-case arrangements (described in Section 5.2.5), increasing in size from black to light gray. Sampling pixels in these vertical stripes leads to very poor estimates of the true color distribution of this texture because the sample may, e.g., contain only black pixels, omitting the lighter pixels in the image.

5.3 Proposed Algorithm

Based on our discussion, the task of segmentation can now be seen as finding the labeling function σ using local histograms of f , which we can calculate. Thus, based on Theorem 2, the task of segmenting an image modeled as occlusions of realizations of textures, $f = O_\sigma\{f_n\}_{n=0}^{N-1}$, into N regions can be approached as an optimization

$$\arg \min_{\sigma} \left\| L_w f - \sum_{n=0}^{N-1} [w * 1_{\{n\}}(\sigma)] \hat{p}_f^{X_n} \right\|, \quad (5.6)$$

where $X_n = \{x \mid \sigma(x) = n\}$ is the set of pixels belonging to the n th region and $\hat{p}_f^{X_n}(v)$ is a histogram formed from the pixels in X_n as defined in (5.3). This optimization finds a labeling function, σ , that splits the image into N regions. These regions are such that for each $x \in X$, the local histogram, $L_w f(x, v)$, is approximated by a convex combination of the value histograms of those regions, $\hat{p}_f^{X_n}(v)$, with the weights of this convex combination determined by the amount of each label in the neighborhood of the histogram, $[w * 1_{\{n\}}](x)$.

Our approach is philosophically similar to the method in [117], which uses an active contour to separate the image into a foreground and a background region that have maximally distinct color distributions. This method differs from ours in two key ways: First, our method does not find maximally distinct distributions, rather it finds distributions that are consistent with the local histograms taken inside their corresponding region. Second, our method does not parametrize the boundary between regions as a level set, allowing it to more naturally handle segmentation of more than two regions. In short, though both methods focus on color distributions rather than edges, they actually optimize different quantities in different ways.

Rather than solve (5.6) directly, we approximate it via variable splitting [118, 119] and relaxation [120] as

$$\begin{aligned} & \arg \min_{\substack{\sigma, p_0, p_1, \dots, p_{N-1} \\ \alpha_0, \alpha_1, \dots, \alpha_{N-1}}} \left\| L_w f - \sum_{n=0}^{N-1} \alpha_n p_n \right\| + \\ & \quad \sum_{n=0}^{N-1} \lambda_n \|w * 1_{\{n\}}(\sigma) - \alpha_n\| + \\ & \quad \sum_{n=0}^{N-1} \mu_n \|\hat{p}_f^{X_n} - p_n\|, \end{aligned} \quad (5.7)$$

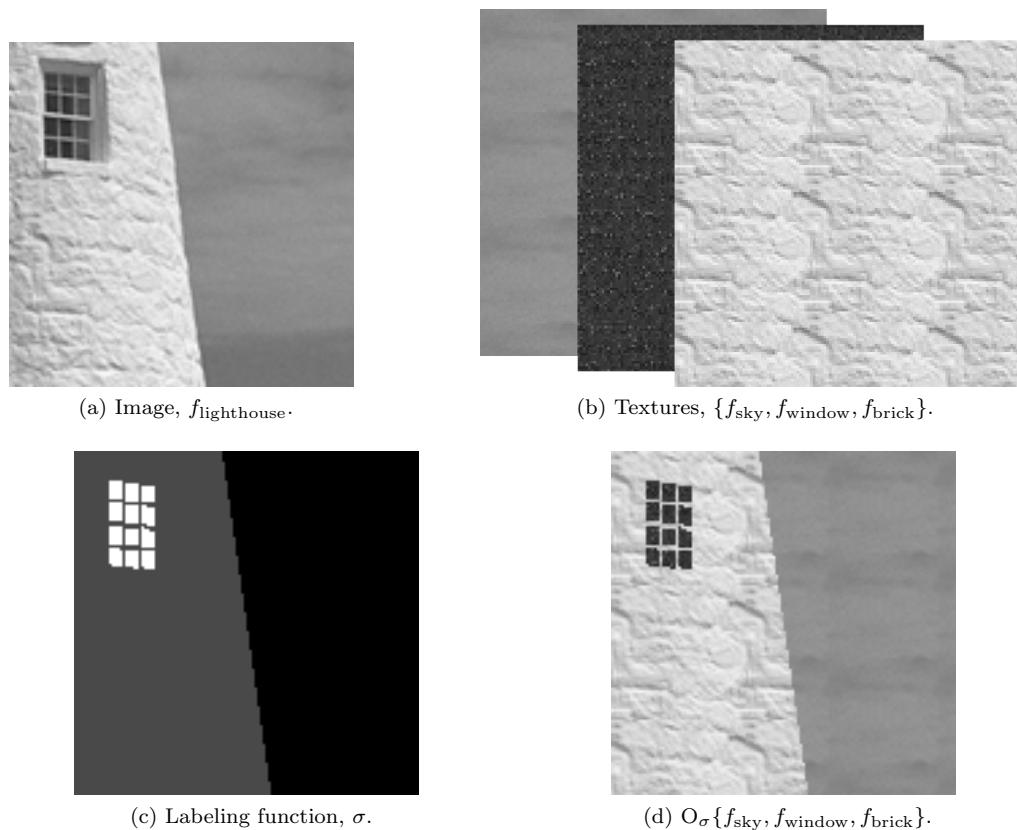


Figure 5.6: Modeling an image as an occlusion of realizations of textures according to a labeling function.

subject to

$$\sum_{v \in V} p_n(v) = 1, \quad \sum_{n=0}^{N-1} \alpha_n(x) = 1, \quad p_n(v), \alpha_n(x) \geq 0,$$

where $\{\lambda_n\}$ and $\{\mu_n\}$ are Lagrange multipliers. The first term of the optimization finds histograms p_0, p_1, \dots, p_{N-1} that can be combined with weights $\alpha_0, \alpha_1, \dots, \alpha_{N-1}$ to create the local histograms of f . The second term finds an indicator function for each region in the image $1_{\{n\}}(\sigma(x))$, such that the blurred version of this indicator is similar to the corresponding weight image α_n . The third term keeps the histograms p_0, p_1, \dots, p_{N-1} close to the empirical histograms of the regions $\hat{p}_f^{X_n}$.

The benefit of working with (5.7) is that the third term is straightforward and the first two terms are well-studied problems, NMF [121, 122] and image deconvolution (deblurring) [89, 123], respectively. To find a local minimum of (5.7), we draw on the wide variety of existing approaches for each of these subproblems to iteratively minimize each of the three terms.

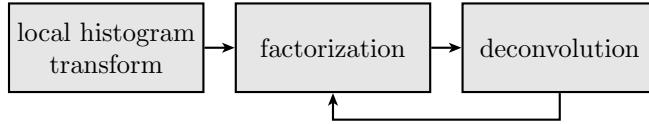


Figure 5.7: Block diagram of the proposed algorithm.

Figure 5.7 shows the block diagram of our proposed algorithm and the following pseudocode gives more details:

```

1: procedure  $[\sigma] = \text{SEGMENT}(f)$ 
2:   compute the local histogram transform of  $f$ 
3:   factor the histograms to initialize  $\{\alpha_n\}$  and  $\{p_n\}$ 
4:   while not converged do
5:     deconvolve  $\{\alpha_n\}$  to estimate  $\sigma$ 
6:     use  $\sigma$  to compute  $\hat{p}_f^{X_n}$ 
7:     set  $p_n = \hat{p}_f^{X_n}$ 
8:     use  $p_n$  and the local histograms to calculate  $\{\alpha_n\}$ 
9:   end while
10: end procedure
  
```

The following subsections present the details of one implementation of each of these steps; others are possible.

5.3.1 Local Histogram Transform

Implementation of the local histogram transform is straightforward based on its definition: each level of the local histogram can be computed as a single convolution. The size and shape of the window w over which local histograms are computed is an important parameter of the method. In general, when w has small support, the resulting segmentation will have better-localized boundaries, but may be noisy. Making w larger smooths the boundaries but also makes the segmentation less susceptible to noise.

The main challenge in computing the local histogram transform is memory, since an image can have 24 bits of color and 2 million pixels, meaning that the full local histogram of the image would have $2^{24} \times 2 \times 10^6 \approx 3.4 \times 10^{13}$ values in it. Luckily, real-world images rarely make meaningful use of this color depth. We can therefore safely quantize the image down to tens of values. The result is a new image, $f' : X \rightarrow V'$ with $|V'| \ll |V|$. The specific quantization method can be chosen to fit the problem; we use k-means clustering.

The quantization step can also be understood as a way to control the value bandwidth of the local histograms. Using empirical histograms rather than, for example, kernel density estimation implicitly assumes

```

1: procedure  $[H, W] = \text{FACTOR\_ALS}(A)$ 
2:   initialize  $H$  with random values
3:   while not converged do
4:      $W = (HH^T)^{-1}HA$ 
5:     set negative values in  $W$  to zero
6:      $H = (W^TW)^{-1}W^TA$ 
7:     set negative values in  $H$  to zero
8:     normalize  $H$  and  $W$ 
9:   end while
10: end procedure

```

Figure 5.8: Alternating least-squares method for non-negative matrix factorization.

that $p_F(v_i)$ and $p_F(v_j)$ are unrelated, no matter how similar the values v_i and v_j are. This assumption is more accurate for a well-quantized image than one with 24 bits of color.

5.3.2 Factorization

To compute the initial factorization of the local histograms of f' , we arrange them into a matrix, A , of size $|V'| \times |X|$. Our goal is to express A as $A = HW$, where H is a $|V'| \times N$ matrix of histograms and W is an $N \times |X|$ matrix of weights. We frame this as the optimization

$$\begin{aligned} \arg \min_{H,W} \quad & ||A - HW|| \\ \text{subject to} \quad & H_{ij} \geq 0, W_{ij} \geq 0, \quad \text{for all } i,j, \\ & \sum_i H_{ij} = 1, \sum_j W_{ij} = 1, \end{aligned}$$

which we solve with a variant of the alternating least-squares method [124], described in Figure 5.8.

To increase the robustness of the algorithm, we repeat it multiple times with different random initializations of H and keep the result with the lowest error. To reduce the computational cost, we factor only a random subset of the histograms in each iteration.

We used alternating least-squares for its simplicity, but repeat that our framework is flexible: any NMF method can be used. Other approaches in the literature may be faster or allow for the inclusion of priors. For example, encouraging W to be sparse would correspond to the assumption that each point in the image is only near a small number (one or two) texture regions. For more discussion of approaches to the NMF problem, see [122].

5.3.3 Deconvolution

Each row of the matrix W from the factorization step can be reshaped into an image of weights α_n that represents a blurred version of one level of the labeling function, $w * 1_{\{n\}}(\sigma)$. To recover σ from these images, we must deconvolve them, that is, solve

$$\arg \min_{\sigma} \sum_{n=0}^{N-1} ||w * 1_{\{n\}}(\sigma) - \alpha_n||. \quad (5.8)$$

The problem in (5.8) is a sum of N deconvolution problems with the added complexity that the image to be recovered in each of the problems, $1_{\{n\}}(\sigma)$, is $\{0, 1\}$ -valued. In general, existing deconvolution methods can be adapted to this problem by relaxing σ to be a set of real-valued images, $\{\hat{\sigma}_n\}$ with the constraint that $\sum_{n=0}^{N-1} \hat{\sigma}_n(x) = 1$. Then the joint deconvolution can be solved by working on each $\hat{\sigma}_n$ in turn and renormalizing every few iterations. Then, σ can be recovered from $\{\hat{\sigma}_n\}$ via $\sigma(x) = \arg \max_n \hat{\sigma}_n(x)$.

The design freedom our framework offers is especially valuable in the deconvolution step, as there exist numerous deconvolution methods that allow the use of prior knowledge about the image to be recovered. If we can describe the shape of the regions with a few parameters, parametric deconvolution can achieve excellent results. In less constrained cases, many looser priors may be used. For example, total variation regularization [125] promotes solutions with a blocky appearance.

In this work, we explore two deconvolution methods: The first method is to assume that $\hat{\sigma}_n = \alpha_n$ for all n . This means that the label at a given point corresponds to the histogram with the largest weight at that point, so $\sigma(x) = \arg \max_n \alpha_n(x)$. This method, while basic, is useful because it is fast and requires no particular knowledge about the properties of the labels we want to recover. The downside is that the resulting labelings tend to be smooth.

The second method is parametric deconvolution. We assume that the regions to be labeled, $1_{\{n\}}(\sigma)$, are the Voronoi cells of three seed pixel locations. We then deconvolve using gradient descent on the seed pixel locations. This method is useful only on images that fit this model; we include it as a proof of concept for parametric deconvolution.

5.3.4 Selecting the Number of Textures

In what we have presented so far, the number of textures N must be known beforehand. While this may be sufficient in certain applications, we may also want a method for selecting an appropriate N automatically from the input image. Selecting the number of textures is essentially the same problem as selecting the number clusters for any unsupervised clustering algorithm. There are numerous approaches to this problem; for an overview see Section 3.3 in [126]. In general these methods replace the parameter N with one or more new parameters that control the trade-off between having tight clusters and having a small number of clusters.

In our work, we have experimented with a simple method in which we set a parameter $r < 1$ that controls how much we expect the model error to decrease when the number of textures increases by one. We begin by setting $N = 1$, running the whole algorithm, and measuring the model error as the value of the norm in (5.6). We then increase N by one and repeat this process. As soon as the error at N_{i+1} fails to be smaller than r times the error at N_i , we select N_i as the final number of textures.

5.4 Experiments and Discussion

We compared our segmentation method to five other methods from the literature on synthetic test images and on real histology images; results are given in Table 5.1 and example segmentations in Figures 5.9, 5.10, 5.11, and 5.12. We now describe the test data, the specifics of these experiments, and each of the comparison methods and discuss the results of the comparison. The test data and MATLAB code for our proposed algorithm is available in the reproducible research compendium for this work, [127].

5.4.1 Datasets

The three datasets we used for testing are as follows:

The *random texture* dataset is a synthetic dataset designed to illustrate the difficulty of segmenting edgeless images. To generate these images, we choose three seed pixel locations at random and let the ground truth regions be the Voronoi cells formed by these seeds. For each region, we select a color distribution at random and select each pixel independently from this distribution (since each pixel is independent of its neighbors, these images have a persistence length of zero). The resulting images can be segmented by eye, but lack distinct edges between regions. At the same time, the high frequency content within regions can be mistaken for edges. We created 25 such images. The number of unique colors in each image is eight and the size of each image is 128×128 .

The *histology* dataset consists of color images of H&E-stained tissue (for more details on these images, see [114]). The ground truth was drawn by an expert pathologist via visual inspection. From a set of 36

Method	Dataset		
	random texture	histology	Prague
Normalized	0.785 ± 0.122	0.736 ± 0.176	0.812 ± 0.064
JSEG	0.943 ± 0.067	0.751 ± 0.141	0.839 ± 0.056
EDISON	0.697 ± 0.084	0.798 ± 0.148	0.784 ± 0.077
Efficient	0.566 ± 0.189	0.619 ± 0.129	0.823 ± 0.070
gPb-owt-ucm	0.707 ± 0.206	0.787 ± 0.130	0.835 ± 0.062
ORTSEG	0.989 ± 0.002	0.830 ± 0.144	0.766 ± 0.073
ORTSEG-D	0.992 ± 0.004		

Table 5.1: Results of the segmentation comparison (Section 5.4) in terms of the Rand index mean and standard deviation, with the best result for each dataset in bold. The value one indicates perfect agreement with the ground truth.

images comprising more than 20 tissue types, we manually selected 36 128×128 subimages such that each contained only two tissues. The aim of this process was to include only subimages that had very accurate ground truth labeling. This dataset is relatively small because only trained pathologists can reliably label histology images and creating pixel-wise labels is very time-consuming; we hope to expand the dataset in the future. In the histology dataset, some regions have clearly defined boundaries while many do not, we expect that our edgeless approach should work well.

The *Prague* dataset contains the grayscale mosaics from the Prague Texture Segmentation Benchmark [11]. These images were designed to be similar to the natural images in datasets like the Berkeley Segmentation Dataset and Benchmark [3], while avoiding the ambiguity in the ground truth that comes with human labeling. We take this dataset to be indicative of performance on natural image segmentation; it is *not* representative of the class of edgeless images for which our algorithm is designed. The dataset contains twenty images of size 512×512 with the number of textures per image varying from three to twelve.

5.4.2 Algorithms Tested

We implemented two versions of the algorithm described in Section 5.3:

ORTSEG (Occlusion of Random Texture SEGmenter) uses the maximum weight deconvolution method and therefore includes no strong prior information about the region boundaries. The parameters are the number of textures N (or the automatic selection parameter r), the size of the window $|w|$, and the number of colors to quantize to, $|V'|$.

ORTSEG-D uses parametric deconvolution. Since it makes strong assumptions on the region boundaries, it is only useful on the random texture dataset. The parameters are the same as those for ORTSEG.

We compare our two algorithms to five others:

Normalized Cut [7] forms a graph from an image by using pixels as nodes and assigning edge weights based on distance and color similarity. It segments by recursively finding graph cuts that maximize the similarity within and dissimilarity between regions. The only parameter is the number of regions.

JSEG [5] quantizes the colors in an image and then searches for segmentations that produce regions of uniform color distribution via region growing. JSEG does not rely on edge information, and is therefore the most similar to ORTSEG of these comparison methods. The parameters are the region merging threshold, the quantization threshold, and the number of scales.

EDISON [1] uses mean shift to cluster pixels in terms of their color and location. Edge information is included as a weight during the mean shift computation.

The parameters are the spatial bandwidth, the range bandwidth, and the minimum region size.

Efficient Graph-Based Segmentation [2] is another graph-based algorithm. Pixels are used as nodes and edges are assigned between adjacent pixels based on difference in intensity. It groups pixels into regions when there is no evidence of a boundary between them. The parameters are the smoothing σ , scale k , and the minimum region size.

gPb-owt-ucm [3] detects local edges with a combination of color and texture features. Global information is combined with the local edges via spectral clustering to detect contours in the image. Finally, a watershed transform finds regions from the contours. As of its publication in 2010, this method was the top performer on the Berkeley Segmentation Dataset [128], which is a large and well-annotated dataset for benchmarking natural image segmentation methods. The only parameter is the edge strength required to create a segment, k .

Note that we do not compare to our previous work [115] because it presented a supervised segmentation method, while ORTSEG and all the comparison methods are unsupervised.

5.4.3 Experimental Setup

The comparison was performed as a leave-one-out cross validation. For each method and dataset, each image was sequestered in turn while the others were used to perform a parameter sweep over the parameters mentioned in Section 5.4.2 with the goal of allowing each method to achieve its maximum performance. Specifically, we set up the sweep according the principles: (1) When the authors of the method specified an operating range for a parameter, we swept between its minimum and maximum value. When they did not, we swept between values that produced opposite kinds of bad results (e.g. too coarse and too fine segmentations) to ensure that the optimal value was between our end points. (2) When the correct value for a parameter (e.g. number of regions) was known for an entire dataset, this value was the only one used. (3) When methods had comparable parameters (e.g. a window size), comparable values were swept. (4) The total number of parameter settings to sweep did not exceed 150 for any method.

The performance of the method was then evaluated on the sequestered image using the parameters that resulting in the best average performance on the training set. The results in Table 5.1 represent the average performance when this process was repeated over all images.

As our performance metric, we chose the Rand index [129, 130], calculated as

$$\text{PR}(\sigma_{\text{test}}, \sigma_{\text{truth}}) = \frac{1}{\binom{N}{2}} \sum_{\substack{x,y \\ x < y}} c(x, y)d(x, y) + (1 - c(x, y))(1 - d(x, y)),$$

where

$$\begin{aligned} c(x, y) &= 1_{\{\sigma_{\text{test}}(x)\}}(\sigma_{\text{test}}(y)), \\ d(x, y) &= 1_{\{\sigma_{\text{truth}}(x)\}}(\sigma_{\text{truth}}(y)). \end{aligned}$$

The Rand index gives the fraction of pairs of pixels that are either grouped into a single region in both the ground truth and the test image or are in different regions in both the ground truth and the test image. It therefore ranges from zero to one, with one being perfect agreement between the test image and the ground truth. We selected it because it naturally handles multi-region unsupervised segmentation and avoids degenerating when a method greatly over- or under-segments an image. There are a wide variety of other, similarly good performance metrics for segmentation algorithms; see Section 3 of [131] for a good overview. In addition to the Rand index, we give results in terms of the variation of information [132] in Appendix 5.C. The ranking of methods is largely unchanged between metrics and the best-performing method for each dataset does not change.

5.4.4 Discussion

On the random texture dataset (Figure 5.10), ORTSEG is clearly superior to the comparison methods, which shows that it excels when segmenting truly edgeless images. ORTSEG-D performs slightly better, giving

essentially perfect segmentations (Figure 5.9), because it leverages prior knowledge about how the region boundaries were formed. JSEG has the next best performance, which makes sense because it does not rely on edge information. We believe that ORTSEG outperforms JSEG in this case because ORTSEG optimizes its labels over the whole image rather than in a region merging scheme. The other methods have a difficult time with the random texture dataset, likely due to their reliance on edge information.

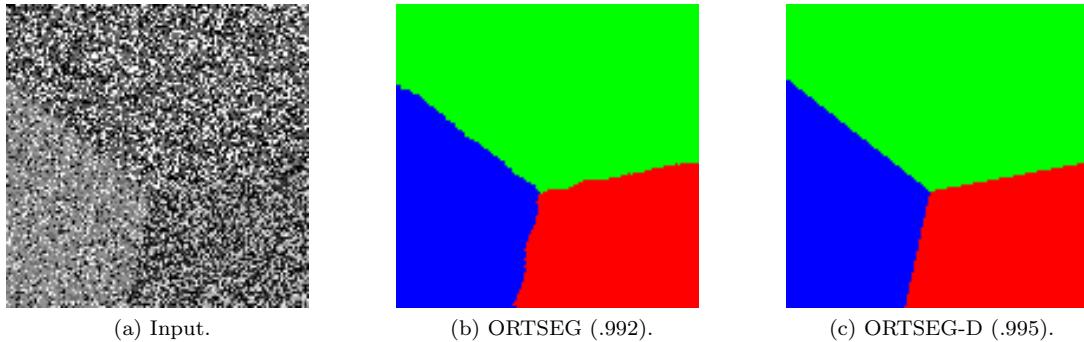


Figure 5.9: Example segmentation results from ORTSEG and ORTSEG-D on a synthetic texture; Rand indexes are given in parentheses. The parametric deconvolution in ORTSEG-D forces region boundaries to be linear, improving the segmentation quality.

On the histology dataset (Figure 5.11), ORTSEG also outperforms the comparison methods, though the next best method is within a standard deviation. On this dataset, ORTSEG selects a relatively large window w (25×25 , as opposed to 9×9 on the random textures), resulting in smoother boundaries compared to e.g. EDISON. This is an asset because the tissues generally have smooth boundaries. We believe that ORTSEG performs worse on the histology images than on the random textures because the histology images are much more complex. The relatively stronger performance of the comparison methods makes sense because at least some of the images in this dataset do have sharp edges.

On the Prague dataset (Figure 5.12), ORTSEG does not outperform the comparison methods. We believe that ORTSEG's relatively poor performance is because, by design, it does not detect the sharp edges between regions, while other methods do (e.g. note the sharp edges that Normalized Cut finds). Surprisingly, JSEG has the best performance on this dataset. We have a few ideas why. First, this may be because the seeded region growing approach is good at finding the correct number of regions in the image. Second, while JSEG does not explicitly look for edges, it uses high J values as evidence of boundaries, which may let it leverage the edges in these images to some degree. Finally, JSEG uses information at a variety of scales while our implementation of ORTSEG does not, which may help in the Prague dataset where the scale of the textures is large relative to the complexity of the boundaries.

Parameter selection.

Here we make a few observations about the parameters selected for ORTSEG during the cross validation. For the random texture dataset, the typical window size selected was 9×9 , while the quantization was set to 8 colors and the number of textures was set to 3. This relatively small window size makes sense: the textures in images are small scale (neighboring pixels are statistically independent) so only a small window is needed to recognize them and has the advantage of localizing the boundaries better than a larger window. The performance on this dataset is not very sensitive to scale, however, as window sizes from 5×5 to 25×25 generally give Rand indexes that vary less than .01. In the histology dataset, the typical window size selected was 25×25 and the typical quantization selected was 50 colors. The number of textures was set to two. Though the histology images are the same size as the random images, the increased scale of the textures means that a larger window is necessary, reducing the ability of the algorithm to accurately localize boundaries. Again, results were not very sensitive to changes in window size. Changes to the

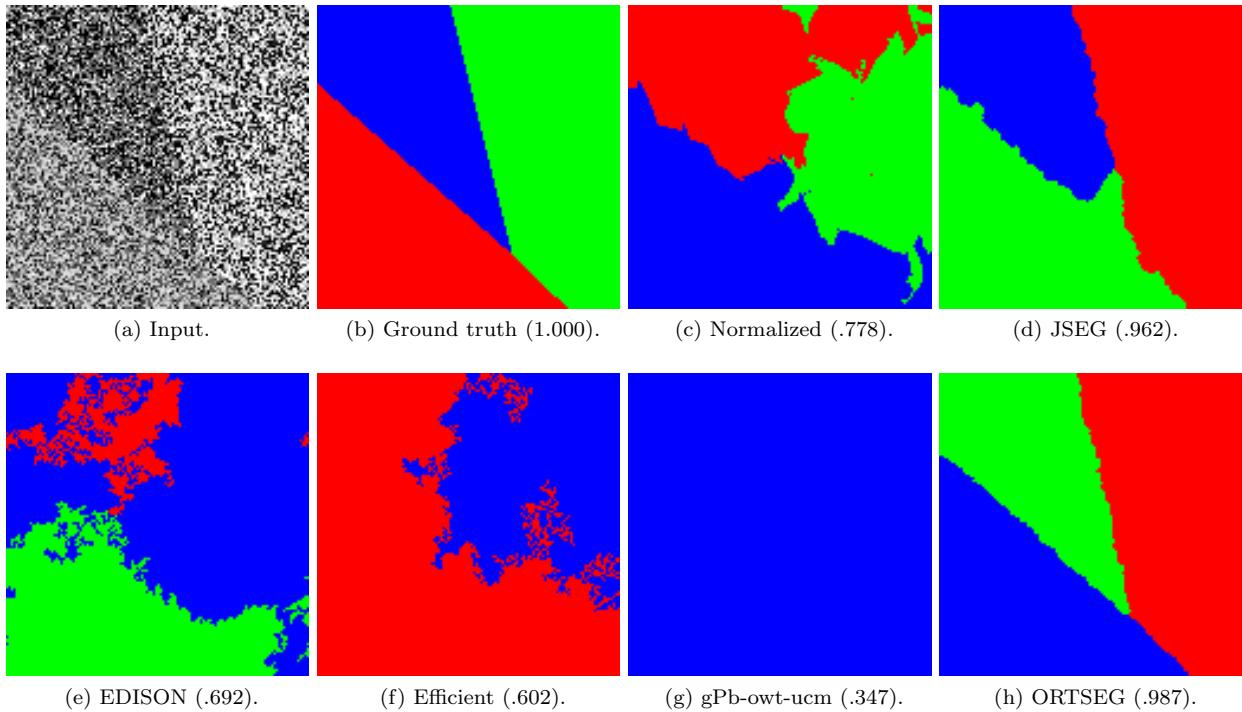


Figure 5.10: Example segmentation results for (c)-(g) the comparison methods and (h) our method on the random texture dataset; Rand indexes are given in parentheses. This dataset is challenging due its lack of meaningful edges.

number of quantized colors did not make much difference once the number of colors was greater than ten. The performance of ORTSEG on the Prague dataset was not strong enough to draw any conclusions about parameters.

Complexity and performance.

The computation complexity of ORTSEG is dependent on the particular implementation of the individual steps. The local histogram transform is a filtering operation and will therefore scale well with input image size; this is also true of many deconvolution algorithms. The factorization step as we described it involves inverting a matrix that grows in size with the square of the number of pixels in the image, however if the number of textures remains low and no texture region is very small, then in practice a random subset of pixels can be used for factorization, reducing the complexity of this step.

For our non-optimized MATLAB implementation, run times per image on a Windows 7 laptop with an Intel Core i7-620M processor for the random texture, histology, and Prague datasets are about 1.5 seconds, 2.5 seconds, and 22 seconds, respectively. On the same computer, segmenting a color 1600×1200 image into five regions (result not shown) takes about 30 seconds. Selecting the number of textures automatically as described in Section 5.3.4 increases these times multiplicatively because it simply runs the algorithm repeatedly with different values of N .

5.5 Conclusions and Future Work

Inspired by the difficult task of segmenting histology images, we proposed a new mathematical and algorithmic framework for image segmentation. We began with the idea of textures as random fields and

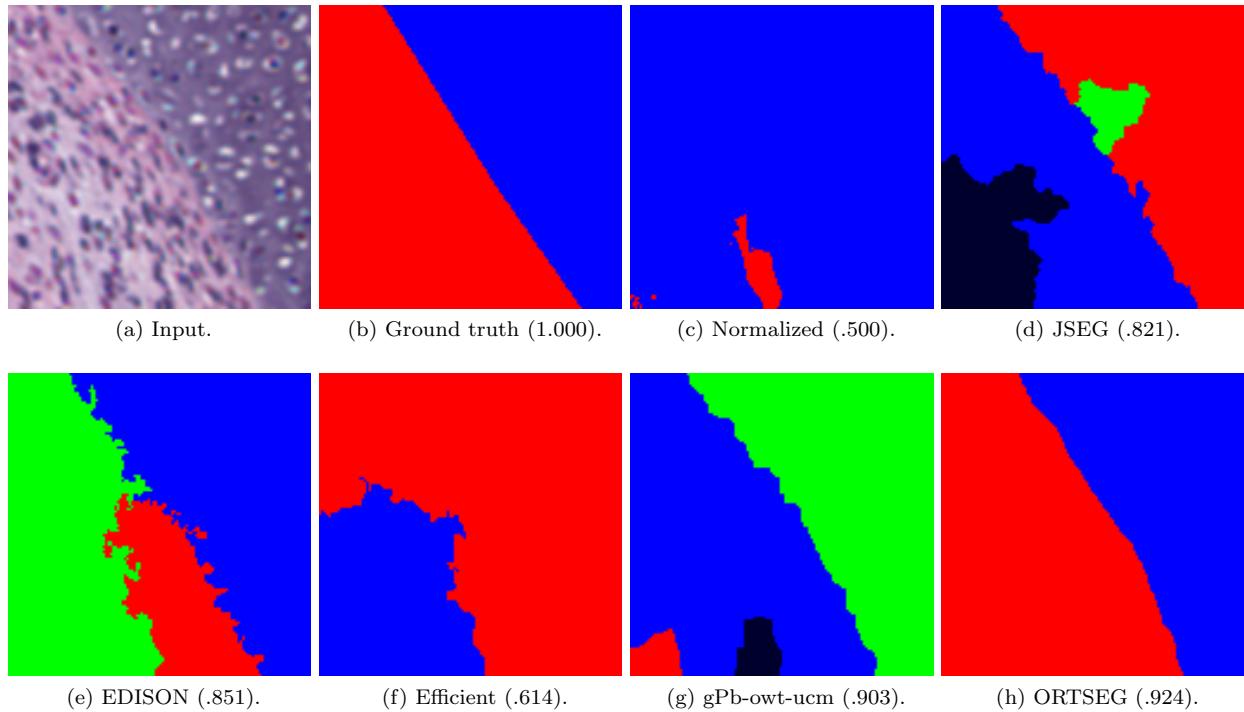


Figure 5.11: Example segmentation results for (c)-(g) the comparison methods and (h) our method on the histology dataset; Rand indexes are given in parentheses. This is a real-world example of a dataset in which the lack of edges is challenging for many segmentation methods.

explored how the dependence between pixels in these textures varies spatially by introducing the notions of dependence decay and persistence length. We then modeled images as occlusions of textures and showed in Theorem 2 that the local histograms of these images are convex combinations of the value distributions of their component textures. Based on this theorem, we proposed a segmentation framework that first discovers the value distributions of the component textures of an image, finds the contribution of each texture to every local histogram of the image, then deconvolves the contributions to recover a segmentation.

We presented one implementation of this framework, ORTSEG, and compared it to five segmentation methods from the literature on three datasets. ORTSEG outperformed the other methods on images of random textures and real histology images, indicating that it handles the difficult class of edgeless images and has real applicability to histology image segmentation.

In the future, we aim to expand our mathematical model by studying histograms of filter responses rather than simply colors. This would allow for a richer characterization of textures and has interesting algorithmic ramifications. We will also explore other implementations of the presented algorithmic framework, including calculating histograms over multiple scales and testing a wider variety of deconvolution approaches.

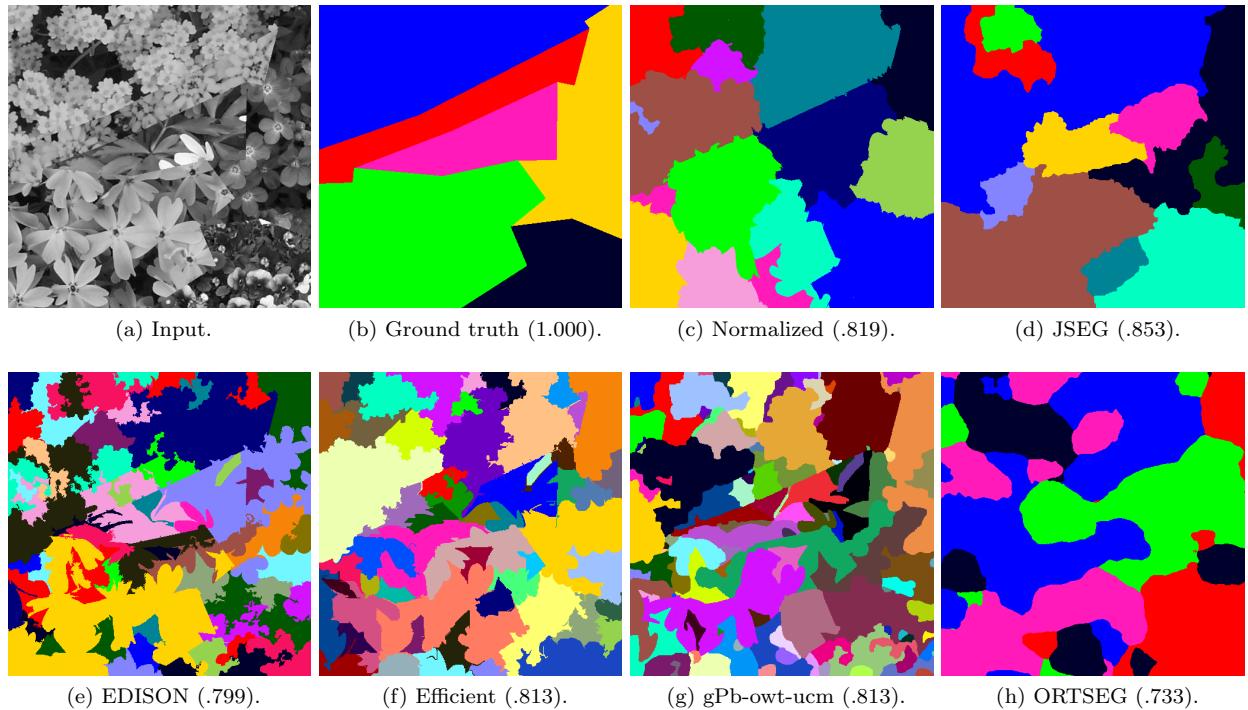


Figure 5.12: Example segmentation results for (c)-(g) the comparison methods and (h) our method on the Prague dataset; Rand indexes are given in parentheses. This dataset simulates segmentation of natural images. Our method is not well suited to such images because it by design does not make use of edges as evidence of boundaries.

Appendix

5.A Proof of Theorem 1

Proof. Let F be a flat texture with value distribution p_F and persistence length ϵ . Given any set of N_0 unique pixel locations, $\Theta = x_0, x_1, \dots, x_{N_0-1}$, we aim to partition that set into subsets such that the minimum distance between two locations in a subset is larger than ϵ . We call the locations within ϵ of a location, x_i , its neighbors. We sort the locations in order of decreasing number of neighbors. Let the largest number of neighbors of any location be A_0 . We build a subset by first selecting the location with the largest number of neighbors and then selecting the location with the next largest number of neighbors among all those that are not neighbors of the first location. We continue this process with each new location selected such that it is not a neighbor of any previously selected location. We end this process after forming a subset of size $K_0 = \left\lfloor \frac{N_0}{A_0} \right\rfloor$. This is possible because each addition to the subset can only preclude a maximum of A_0 locations from further consideration. Additionally, each location with A_0 neighbors must either be in this set or be a neighbor of a location in this set. If this is not the case, then there are A_0 locations not in the set, which leads to $N_0 \geq K_0 + A_0$, which is a contradiction.

Removing the locations in the first subset from consideration, let N_1 be the number of remaining locations and A_1 be the maximum number of neighbors among them. We know that $A_1 \leq A_0 - 1$ because, similarly to the reasoning above, each point that had A_0 neighbors belongs to the first subset or lost at least one neighbor to the first subset. We now follow the same procedure as above, creating a subset of size $K_1 = \left\lfloor \frac{N_1}{A_1} \right\rfloor$. We continue the process until all points have been placed in subsets. The same argument as above gives that $A_{i+1} \leq A_i - 1$ and therefore,

$$\begin{aligned} K_{i+1} &= \left\lfloor \frac{N_{i+1}}{A_{i+1}} \right\rfloor = \left\lfloor \frac{N_i - K_i}{A_{i+1}} \right\rfloor = \left\lfloor \frac{N_i - \left\lfloor \frac{N_i}{A_i} \right\rfloor}{A_{i+1}} \right\rfloor \\ &\geq \left\lfloor \frac{N_i - \frac{N_i}{A_i}}{A_{i+1}} \right\rfloor \geq \left\lfloor \frac{N_i - \frac{N_i}{A_i}}{A_i - 1} \right\rfloor \\ &= \left\lfloor \frac{A_i N_i - N_i}{A_i (A_i - 1)} \right\rfloor = \left\lfloor \frac{N_i}{A_i} \right\rfloor = K_i. \end{aligned}$$

The above implies that $K_i \geq K_0 = \left\lfloor \frac{N_0}{A_0} \right\rfloor$. Finally, since the largest possible value for A_0 is $A_\epsilon = 2\epsilon^2 + 2\epsilon + 1$, which is the number of locations in the ℓ^1 -ball with radius ϵ , we can define K where $K_i \geq K = \left\lfloor \frac{N_0}{A_\epsilon} \right\rfloor$. Thus we have partitioned the N_0 locations of the original set Θ into subsets where all members are separated by a distance greater than ϵ , and we have shown that each subset has at least K members.

Let $Y_i = \{y_0, y_1, \dots, y_{K_i-1}\}$ be the i th subset. By the definition of persistence length and flatness, the random variables $F(y_0), F(y_1), \dots, F(y_{K_i-1})$ are i.i.d. (persistence length gives independence, flatness gives identical distribution) with distribution p_F . Let the empirical distribution of $F(y_0), F(y_1), \dots, F(y_{K_i-1})$

be

$$\hat{p}_F^{Y_i}(v) = \frac{1}{K_i} \sum_{k=0}^{K_i-1} 1_{\{v\}}(F(y_k)).$$

We can select an arbitrary ordering on the set of values and define the cumulative value distribution and the empirical cumulative value distribution of F as

$$P_F(v) = \sum_{w=0}^v p_F(w)$$

and

$$\hat{P}_F^{Y_i}(v) = \frac{1}{K_i} \sum_{k=0}^{K_i-1} 1_{[0,v]}(F(y_k)),$$

respectively.

Then, by the Dvoretzky-Kiefer-Wolfowitz inequality [133],

$$\Pr(\max |\hat{P}_F^{Y_i}(v) - P_F(v)| > \alpha) \leq 2e^{-2K_i\alpha^2}.$$

To return to the (non-cumulative) color distributions we note that

$$\begin{aligned} & |\hat{p}_F^{Y_i} - p_F|(v) \\ &= |(\hat{P}_F^{Y_i}(v) - \hat{P}_F^{Y_i}(v-1)) - (P_F(v) - P_F(v-1))| \\ &\leq |(\hat{P}_F^{Y_i}(v) - P_F(v)) - (\hat{P}_F^{Y_i}(v-1) - P_F(v-1))| \\ &\leq 2 \max |\hat{P}_F^{Y_i}(v) - P_F(v)|, \end{aligned}$$

so

$$\Pr(\max |\hat{p}_F^{Y_i}(v) - p_F(v)| > \alpha) \leq 2e^{-K_i\alpha^2/2}.$$

Finally, letting $\alpha = \sqrt{\frac{-2 \log \frac{\gamma}{2}}{K_i}}$, gives

$$\max |\hat{p}_F^{Y_i}(v) - p_F(v)| \leq \sqrt{\frac{-2 \log \frac{\gamma}{2}}{K_i}} \leq \sqrt{\frac{-2 \log \frac{\gamma}{2}}{K}} \quad (5.9)$$

with probability at least $1 - \gamma$.

We now examine the histogram formed from all the locations in the original set Θ rather than a single subset. If we have a total of I subsets, then

$$\begin{aligned} \hat{p}_F^\Theta(v) &= \frac{1}{N_0} \sum_{n=0}^{N_0-1} 1_{\{v\}}(F(x_n)) \\ &= \frac{1}{N_0} \sum_{i=0}^{I-1} \sum_{y \in Y_i} 1_{\{v\}}(F(y)) = \frac{1}{N_0} \sum_{i=0}^{I-1} K_i \hat{p}_F^{Y_i}(v) \end{aligned}$$

and therefore

$$\begin{aligned}
& \max |\hat{p}_F^\Theta(v) - p_F(v)| \\
&= \max \left| \frac{1}{N_0} \sum_{i=0}^{I-1} K_i \hat{p}_F^{Y_i}(v) - p_F(v) \right| \\
&= \max \left| \frac{1}{N_0} \sum_{i=0}^{I-1} K_i \hat{p}_F^{Y_i}(v) - \frac{1}{N_0} \sum_{i=0}^{I-1} K_i p_F(v) \right| \\
&= \max \frac{1}{N_0} \sum_{i=0}^{I-1} K_i \left| \hat{p}_F^{Y_i}(v) - p_F(v) \right| \\
&\leq \frac{1}{N_0} \sum_{i=0}^{I-1} K_i \sqrt{\frac{-2 \log \frac{\gamma}{2}}{K}} \\
&= \sqrt{\frac{-2 \log \frac{\gamma}{2}}{K}} = \sqrt{\frac{-2 \log \frac{\gamma}{2}}{\left\lfloor \frac{N_0}{A_\epsilon} \right\rfloor}} \leq \sqrt{\frac{-2 A_\epsilon \log \frac{\gamma}{2}}{N_0}}
\end{aligned}$$

with probability at least $(1 - \gamma)^I$, because each of the I inequalities from (5.9) must hold simultaneously. Given that the minimum subset size is $\left\lfloor \frac{N_0}{A_\epsilon} \right\rfloor$, we know that $I \leq A_\epsilon$, and thus this probability is at least $(1 - \gamma)^{A_\epsilon}$. We finally set $\gamma = 1 - \sqrt[A_\epsilon]{1 - \delta}$. The result is

$$\max |\hat{p}_F^\Theta(v) - p_F(v)| \leq \sqrt{\frac{-2 A_\epsilon \log \frac{1 - \sqrt[A_\epsilon]{1 - \delta}}{2}}{N_0}}$$

with probability at least $(1 - \delta)$. This means that F has dependence decay complexity $a = (-2 A_\epsilon \log \frac{1}{2} (1 - \sqrt[A_\epsilon]{1 - \delta}))^{1/2}$, proving the theorem. \square

5.B Proof of Theorem 2

Proof. Start with the first term of the left-hand side of (5.5),

$$\begin{aligned}
L_w O_\sigma \{f_n\}_{n=0}^{N-1}(x, v) &\stackrel{(a)}{=} L_w \sum_{n=0}^{N-1} 1_{\{n\}}(\sigma(x)) f_n(x) \\
&\stackrel{(b)}{=} \sum_{x' \in X} 1_{\{v\}} \left(\sum_{n=0}^{N-1} 1_{\{n\}}(\sigma(x')) f_n(x') \right) w(x - x') \\
&\stackrel{(c)}{=} \sum_{n=0}^{N-1} \sum_{x' \in X} 1_{\{n\}}(\sigma(x')) 1_{\{v\}}(f_n(x')) w(x - x'),
\end{aligned} \tag{5.10}$$

where (a) follows from the definition of occlusion (5.2), (b) from the definition of the local histogram transform (5.1), and (c) because, for a given v and n , the indicator functions are only nonzero when $\sigma(x') = n$ and $f_n(x') = v$.

We now manipulate (5.10) so that it becomes, for each n , a histogram sampled from f_n . Using the fact

that w is a constant filter (that is, it takes only the value $1/|w|$ or 0), (5.10) becomes

$$\begin{aligned} & \sum_{n=0}^{N-1} \frac{\sum_{x' \in X} 1_{\{n\}}(\sigma(x')) 1_{\{v\}}(f_n(x')) 1_{(0,1]}(w(x - x'))}{|w|} \stackrel{(a)}{=} \\ & \sum_{n=0}^{N-1} \frac{|\Theta_n(x)| \sum_{x' \in X} 1_{\{n\}}(\sigma(x')) 1_{\{v\}}(f_n(x')) 1_{(0,1]}(w(x - x'))}{|\Theta_n(x)|} \\ & \stackrel{(b)}{=} \sum_{n=0}^{N-1} \frac{|\Theta_n(x)|}{|w|} \hat{p}_{f_n}^{\Theta_n(x)}(v), \end{aligned} \quad (5.11)$$

where $\Theta_n(x) = \{y \mid \sigma(y) = n \text{ and } w(x - y) > 0\}$. The equality (a) is a multiplication by one and (b) uses the definition of a histogram, (5.3).

Turning our attention to the second term of the left-hand side of (5.5), we have

$$\begin{aligned} & \sum_{n=0}^{N-1} [w * 1_{\{n\}}(\sigma)](x) p_{F_n}(v) \\ & \stackrel{(a)}{=} \sum_{n=0}^{N-1} \sum_{x' \in X} w(x - x') 1_{\{n\}}(\sigma(x')) p_{F_n}(v) \\ & \stackrel{(b)}{=} \sum_{n=0}^{N-1} \frac{\sum_{x' \in X} 1_{(0,1]}(w(x - x')) 1_{\{n\}}(\sigma(x'))}{|w|} p_{F_n}(v) \\ & \stackrel{(c)}{=} \sum_{n=0}^{N-1} \frac{|\Theta_n(x)|}{|w|} p_{F_n}(v), \end{aligned} \quad (5.12)$$

where (a) follows from the definition of convolution, (b) from w being a constant filter, and (c) from defining $\Theta_n(x)$ as above.

Subtracting (5.12) from (5.11), we have

$$\begin{aligned} & \left| \sum_{n=0}^{N-1} \frac{|\Theta_n(x)|}{|w|} (\hat{p}_{f_n}^{\Theta_n(x)}(v) - p_{F_n}(v)) \right| \\ & \stackrel{(a)}{\leq} \sum_{n=0}^{N-1} \frac{|\Theta_n(x)|}{|w|} |\hat{p}_{f_n}^{\Theta_n(x)}(v) - p_{F_n}(v)| \\ & \stackrel{(b)}{\leq} \sum_{n=0}^{N-1} \frac{|\Theta_n(x)|}{|w|} a_n |\Theta_n(x)|^{-1/2}, \end{aligned} \quad (5.13)$$

where (a) follows from the triangle inequality and positivity of $|\Theta_n(x)|$ and $|w|$ and (b) from the dependence-decay property of $\{F_0, F_1, \dots, F_{N-1}\}$. The probability of (b) holding for each n is $1 - \delta$, thus the probability of it holding for all of them simultaneously is $(1 - \delta)^N$.

We aim to bound (5.13) by its maximum. To do this, we first replace a_n with $a = \max\{a_0, a_1, \dots, a_{N-1}\}$. We then maximize the quantity over $|\Theta_0|, |\Theta_1|, \dots, |\Theta_{N-1}|$ with the constraint that $|\Theta_0| + |\Theta_1| + \dots + |\Theta_{N-1}| = |w|$. This maximum occurs at $|\Theta_0| = |\Theta_1| = \dots = |\Theta_{N-1}| = |w|/N$, meaning that (5.13) is bounded by

$$\sum_{n=0}^{N-1} \frac{1}{N} a \left(\frac{|w|}{N} \right)^{-1/2} = a \left(\frac{|w|}{N} \right)^{-1/2},$$

thus proving the theorem. \square

5.C Variation of Information Results

Method	Dataset		
	random texture	histology	Prague
Normalized	0.819 ± 0.354	0.640 ± 0.336	1.632 ± 0.218
JSEG	0.294 ± 0.188	0.618 ± 0.269	1.327 ± 0.237
EDISON	1.317 ± 0.341	0.509 ± 0.228	1.573 ± 0.392
Efficient	0.918 ± 0.210	0.756 ± 0.252	1.485 ± 0.429
gPb-owt-ucm	0.590 ± 0.285	0.578 ± 0.200	1.545 ± 0.400
ORTSEG	0.097 ± 0.015	0.452 ± 0.268	1.472 ± 0.367
ORTSEG-D	0.065 ± 0.031		

Table 5.C.1: Results of the segmentation comparison (Section 5.4) in terms of the variation of information mean and standard deviation, with the best result for each dataset in bold. The value 0 indicates perfect agreement with the ground truth.

Chapter 6

Describing Angular Distributions with the FS-KDE

The segmentation method of Chapter 5 describes textures by their distribution of colors. What about textures that have the same colors arranged in different patterns? In this chapter, we were inspired by this question to create a descriptor for distributions of angles. Distributions of angular quantities, such as gradients, are a central part of several state-of-the-art image processing algorithms, but these distributions are usually described via histograms, and therefore lack rotation invariance due to binning artifacts. Like histograms, our method allows angular distributions to be represented by a finite number of values, but removes binning artifacts. We discuss how our method can produce a rotation invariant descriptor and then show that it compares favorably to gradient histograms for patch matching, person detection, and texture segmentation.

6.1 Introduction

Central to the tissue segmentation approach of Chapter 5 was the idea that a tissue (or, in fact, any texture), could be described by its distribution of colors. But there is more to texture than color: a checkerboard and pattern of black and white stripes have the same distribution of colors, but they are certainly different textures. One way to address this problem is to include the distribution of edges in the texture description. Checkerboards have both horizontal and vertical edges, while black and white stripes have only horizontal (or only vertical) edges. The focus of this chapter is to explore how to efficiently represent distributions of oriented edges; our representation is useful in a variety of image processing applications including segmentation.

6.1.1 Previous Work

In image processing, distributions of angular quantities are usually described via histograms. These angular histograms are a key component of many of the most successful algorithms for a variety of image processing tasks. For example, SIFT [134], along with some of its variants including GLOH [135], SIFT+GC [136], and CSIFT [137] (but not SURF [51] or PCA-SIFT [138]), use histograms of local gradient angles to form a keypoint descriptor. SIFT descriptors are widely used, with applications including medical image registration [139], human activity analysis [140], and object recognition [141]. HOG [4] and its extensions, such as part-based models [142], calculate local gradient histograms at every point in an image and are useful in human [143] and object [144] detection.

Despite their widespread use in vision, histograms have a fundamental weakness when estimating distributions of angular quantities because they rely on binning and are therefore not invariant to rotation. That is, a rotation of the input angles results in a rotation of the histogram plus distortion; see Figure 6.1.

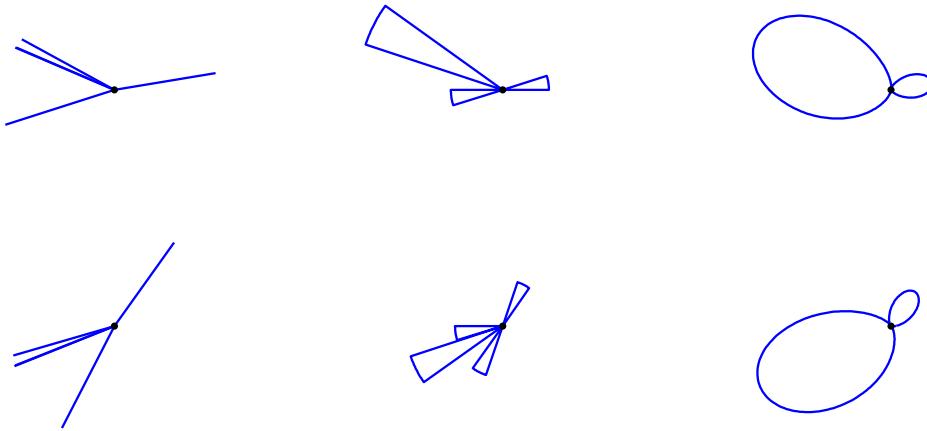


Figure 6.1: Top row: A weighted set of angles, its 20-bin histogram, and the estimate based on the proposed method (FS-KDE) using 20 numbers. Bottom row: The same set of angles rotated counter-clockwise by 45° , its histogram, and FS-KDE. While the rotation distorts the histogram, it only causes a corresponding rotation in the FS-KDE.

This problem affects even methods that attempt to be invariant to rotation. For example, SIFT [134] builds histograms with respect to a dominant angle, but this dominant angle is itself estimated from an angular histogram. The radial gradients computed in RIFF [145] are invariant to rotation, but are collected into angular histograms which are not.

6.1.2 Current Work

In this chapter, we present a new method for estimating the distribution of a set of angular quantities without resorting to binning. Our new estimate can be used as a replacement for histograms in any algorithm where rotation invariance is beneficial.

The outline of the chapter is as follows. In Section 6.2, we develop the theory of our method including its derivation, properties, and extensions. In Section 6.3, we provide experiments in three domains that support the efficacy of the method as a tool in image processing. We conclude in Section 6.4.

6.2 The Fourier Series Kernel Density Estimate

We call our method for estimating and representing angular distributions the *Fourier Series Kernel Density Estimate* (FS-KDE); we now describe its properties, canonicalization, image version, and implementation details.

6.2.1 Derivation of the FS-KDE

The FS-KDE builds on the idea of kernel density estimates (KDEs) [146]. A KDE builds an estimate of a continuous distribution from its samples by putting a lump (kernel) of density at the location of each sample. KDEs create smooth estimates and, under certain assumptions, converge to the correct distribution with fewer samples than histograms [146]. However, KDEs are not as useful as histograms in image processing, because (1) evaluating the KDE at a point requires all of the samples to be stored in memory and (2) there is no straightforward way to compute distances between KDEs. Histograms alleviate these problems, but are not well suited to estimating angular distributions because they rely on binning, which, as Figure 6.1 shows, causes distortion when the input sample is rotated. The key insight of the FS-KDE is that with correct

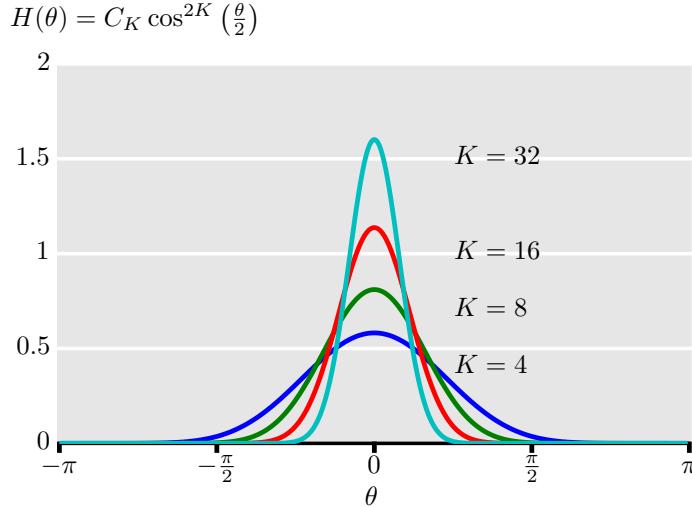


Figure 6.1: Examples of the powered cosine kernel (6.2) for different orders, K . Each kernel is 2π -periodic and integrates to one. As K increases, the kernels become sharper.

kernel selection, the KDE of a set of angles will be band-limited, meaning that it can be exactly represented by a finite number of its Fourier series coefficients. As our subsequent discussion will show, storing the KDE in terms of its Fourier series coefficients addresses both the aforementioned limitations of KDEs without binning.

Given an angle-weight pair, (Θ, W) , consisting of a set of angles, $\Theta = \{\theta_0, \theta_1, \dots, \theta_{N-1}\}$, and a set of positive scalar weights, $W = \{w_0, w_1, \dots, w_{N-1}\}$, we form a KDE, $f : [-\pi, \pi] \rightarrow \mathbb{R}$, of their distribution as a sum of kernels,

$$f(\theta) = \frac{1}{N} \sum_{n=0}^{N-1} w_n H(\theta - \theta_n), \quad (6.1)$$

where the kernel, $H(\theta)$, is a positive function that integrates to one.¹ For example, the angle-weight pair might come from the angles and magnitudes of the gradients in an image.

We now select a specific kernel,

$$H(\theta) = C_K \cos^{2K} \left(\frac{\theta}{2} \right), \quad (6.2)$$

where C_K is a normalizing constant and K , which we call the *order*, controls the width of the kernel (Figure 6.1). We pick this kernel because it is well localized in the angular domain, it has a controllable width, and because it results in a KDE that is bandlimited.

We can rearrange (6.1) to reveal the Fourier series coefficients of the KDE, f ,

¹For greater flexibility, we do not require f to integrate to one; we thus use the term *distribution* loosely.

$$\begin{aligned}
f(\theta) &= \frac{1}{N} \sum_{n=0}^{N-1} w_n C_K \cos^{2K} \left(\frac{\theta - \theta_n}{2} \right) \\
&\stackrel{(a)}{=} \frac{C_K}{N} \sum_{n=0}^{N-1} w_n \left(\frac{e^{\frac{j}{2}(\theta-\theta_n)} + e^{-\frac{j}{2}(\theta-\theta_n)}}{2} \right)^{2K} \\
&\stackrel{(b)}{=} \frac{C_K}{2^{2K} N} \sum_{n=0}^{N-1} w_n \sum_{p=0}^{2K} \binom{2K}{p} e^{\frac{j}{2}p(\theta-\theta_n)} e^{-\frac{j}{2}(2K-p)(\theta-\theta_n)} \\
&\stackrel{(c)}{=} \sum_{p=0}^{2K} \frac{C_K}{2^{2K} N} \binom{2K}{p} \sum_{n=0}^{N-1} w_n e^{j(K-p)\theta_n} e^{j(p-K)\theta} \\
&\stackrel{(d)}{=} \sum_{k=-K}^K \underbrace{\left(\frac{C_K}{2^{2K} N} \binom{2K}{K+k} \sum_{n=0}^{N-1} w_n e^{-jk\theta_n} \right)}_{F_k} e^{jk\theta} \tag{6.3}
\end{aligned}$$

where (a) follows from Euler's formula; (b) from the binomial theorem; (c) from an interchange of finite sums and combining the exponents; and (d) from the substitution $k = p - K$. Equation (6.3) is the expression of f in terms of its Fourier series coefficients, F_k . We denote the relationship between $f(\theta)$ and F_k as $f(\theta) \xrightarrow{FS} F_k$. From (6.3), we see that f is bandlimited: it has $2K + 1$ non-zero Fourier series coefficients. We also see that F_{-k} is the complex conjugate of F_k , so, in practice, only $K + 1$ complex values must be computed and stored to represent f . Thus, an FS-KDE of order K takes the same amount of storage as a histogram with $2(K + 1)$ bins, which addresses the first limitation of KDEs discussed in Section 6.2.1.

6.2.2 Properties

We now discuss some useful properties of the FS-KDE. First, since F is simply the Fourier series representation of f , we can leverage all of the properties of the Fourier series [147]. Of specific interest here are linearity, $\alpha f(\theta) + \beta g(\theta) \xrightarrow{FS} \alpha F_k + \beta G_k$, and Parseval's equality, $\|f\|^2 = 2\pi\|F\|^2$. Together, these mean that the distance between two FS-KDEs, $\|f - g\|^2$, can be computed as the finite sum $2\pi\|F - G\|^2$, which addresses the second limitation of KDEs discussed in Section 6.2.1.

The FS-KDE is rotation invariant in the sense that a rotation of the angles in the angle-weight pair results in a corresponding rotation in the FS-KDE. To be more precise, begin with an angle-weight pair (Θ, W) . Form its rotation, (Θ_ϕ, W) , where $\Theta_\phi = \{\theta_0 + \phi, \theta_1 + \phi, \dots, \theta_{N-1} + \phi\}$. If F is the FS-KDE for (Θ, W) and F_ϕ is the FS-KDE for (Θ_ϕ, W) , then from (6.3),

$$\begin{aligned}
F_{\phi,k} &= \frac{C_K}{2^{2K} N} \binom{2K}{K+k} \sum_{n=0}^{N-1} w_n e^{-jk(\theta_n+\phi)} \\
&= e^{-jk\phi} \frac{C_K}{2^{2K} N} \binom{2K}{K+k} \sum_{n=0}^{N-1} w_n e^{-jk\theta_n} \\
&= e^{-jk\phi} F_k. \tag{6.4}
\end{aligned}$$

By the shift in time property of Fourier series, (6.4) means that f_ϕ is equal to f circularly shifted by ϕ . Thus, a rotation in the input angles has caused a corresponding rotation in the KDE.

6.2.3 Image Version

A common approach in object detection is to estimate local angular distributions for every location in an image. For example, this is the approach of deformable parts models [142]. In this section, we describe

efficient computation of local FS-KDE estimates on images via linear filtering.

Let $(\Theta(x), W(x))$ be a weighted angular image, where $\Theta : X \rightarrow [-\pi, \pi]$ is an image of angles and $W : X \rightarrow \mathbb{R}$ is a corresponding image of weights, where X is a discrete set of pixel locations (e.g. $\mathbb{Z}_{1200} \times \mathbb{Z}_{1600}$). For example, $(\Theta(x), W(x))$ may be formed from computing the gradient of a traditional image. (Note that we write the argument x in $(\Theta(x), W(x))$ to make a distinction between the weighted angular image and the angle-weight pair introduced in Section 6.2.1.)

We aim to compute a KDE around each point $x \in X$, with a neighborhood defined by h , a positive window function with $\|h\|_1 = 1$. We define the FS-KDE for $(\Theta(x), W(x))$ at location a x and angle θ as

$$f(x, \theta) = \sum_{y \in X} W(y) C_K \cos^{2K} \left(\frac{\theta - \Theta(y)}{2} \right) h(x - y),$$

where, as before, K controls the sharpness of the kernel. Then, following a similar procedure as in Section 6.2.1, we arrive at an expression for $f(x, \theta)$ in terms of its Fourier series coefficients,

$$f(x, \theta) = \sum_{k=-K}^K F_k(x) e^{jk\theta},$$

where

$$F_k(x) = \frac{C_K}{2^{2K}} \binom{2K}{K+k} (We^{-jk\Theta} * h)(x),$$

$*$ denotes convolution, and $We^{-jk\Theta}$ is computed pointwise. This means that local FS-KDEs of order K can be computed via $K + 1$ complex filtering operations.

6.2.4 Canonicalization

We showed in Section 6.2.2 that the FS-KDE is rotation invariant in the sense that a rotation of the input angles causes a corresponding rotation in the density estimate. We may, however, also desire that a rotation of the input angles cause no change at all to the estimated distribution. This would be useful if, e.g., FS-KDEs are being used as point descriptors in an image matching application. We can achieve this by rotating FS-KDEs to a standard, or *canonical*, position, such that all FS-KDEs that are rotations of each other end up with the same canonical version. In this section, we present two methods of achieving this canonicalization.

F_1 Canonicalization

A natural way of canonicalizing an angle-weight pair, (Θ, W) , is to rotate the angles such that their mean is equal to zero. One way to define the angular mean is to assign to each angle θ_n a complex number, $z_n = w_n e^{j\theta_n}$ with modulus w_n and argument θ_n , and then sum these numbers and take the argument of the result, $\bar{\theta} = \arg \left(\sum_{n=0}^{N-1} z_n \right)$, where we use $\arg(z)$ to denote the argument of the complex number z . Then, the canonical angle-weight pair is $(\tilde{\Theta}, \tilde{W})$, where $\tilde{\Theta} = \Theta - \bar{\theta}$, which is the rotation of Θ by $-\bar{\theta}$.

From (6.3) we see that the argument of the first Fourier series coefficient of the corresponding FS-KDE, $\arg(F_1)$, is equal to $-\bar{\theta}$. As a result, canonicalizing the angle-weight pair causes F_1 to be real because $\bar{\theta} = 0$. Using this fact, we can directly canonicalize an FS-KDE, f , without having to know the angle-weight pair that it came from. We define the canonical version of f as

$$\tilde{f}(\theta) \xrightarrow{FS} \tilde{F}_k = e^{-jk\arg(F_1)} F_k, \quad (6.5)$$

which is the rotation of f that makes F_1 real.

We now show that this canonicalization has the property that rotating a set of angles does not change its canonical FS-KDE. In other words, all rotations of an angle-weight pair have the same canonical FS-KDE, \tilde{f} .

Lemma 1. Let (Θ, W) be an angle-weight pair, let (Θ_ϕ, W) be its rotation by ϕ , and let F and F_ϕ be the corresponding FS-KDEs. Then,

$$\tilde{F}_{\phi,k} = \tilde{F}_k$$

Proof. Beginning with the FS-KDE (6.3), we have

$$\begin{aligned}\arg(F_{\phi,1}) &= \arg\left(\frac{C_K}{2^{2K}} \binom{2K}{K+1} \sum_{n=0}^{N-1} w_n e^{-j(\theta_n + \phi)}\right) \\ &= \arg\left(e^{-jk\phi} \frac{C_K}{2^{2K}} \binom{2K}{K+1} \sum_{n=0}^{N-1} w_n e^{-j\theta_n}\right) \\ &= \arg(F_1) - \phi.\end{aligned}$$

We know from Section 6.2.2 that $F_{\phi,k} = e^{-jk\phi} F_k$, and thus using the definition of canonicalization (6.5),

$$\begin{aligned}\tilde{F}_{\phi,k} &= e^{-jk\arg(F_{\phi,1})} F_{\phi,k} \\ &= e^{-j(\arg(F_1) - \phi)k} e^{-jk\phi} F_k \\ &= e^{-jk\arg(F_1)} F_k = \tilde{F}_k.\end{aligned}$$

□

Stability of F_1 Canonicalization

Now that we have shown that F_1 canonicalization aligns distributions that are exact rotations of each other, we study its effect on distributions that are noisy rotations of each other. Intuitively, a good canonicalization will give similar canonical versions to all FS-KDEs that are noisy rotations of each other; We call this property *stability*. Conversely, a bad canonicalization might amplify small amounts of noise, assigning similar FS-KDEs very different canonical versions. The following theorem states that the stability of F_1 canonicalization is related to the magnitude of the first Fourier series coefficient of the distribution that is being canonicalized, $|F_1|$. We leave the proof of the theorem to Appendix 6.A.

Theorem 3 (Stability of F_1 Canonicalization). *Let (Θ, W) be an angle-weight pair. Without loss of generality, assume $\bar{\theta} = 0$ and $\sum_{n=0}^{N-1} w_n = N$. Let $(\Theta^\epsilon, \alpha W^\epsilon)$ be its noisy version such that $w_n^\epsilon e^{j\theta_n^\epsilon} = w_n e^{j\theta_n} + \epsilon_n$ where the ϵ_n are drawn according to the complex normal distribution with mean zero and standard deviation σ/\sqrt{N} (i.e., $\Im(\epsilon_n) \sim \mathcal{N}(0, \sigma^2/N)$ and $\Re(\epsilon_n) \sim \mathcal{N}(0, \sigma^2/N)$), and α scales W^ϵ such that $\sum_{n=0}^{N-1} \alpha w_n^\epsilon = N$. Let f and f^ϵ be the K th-order FS-KDEs of (Θ, W) and $(\Theta^\epsilon, W^\epsilon)$, respectively. Then*

$$E \left[\|F^\epsilon - \tilde{F}^\epsilon\| \right] \leq E \left[\sqrt{\sum_{k=-K}^K \left(2B_k N \sin \left(\frac{k}{2} \arctan \left(\frac{B_1 \epsilon}{|F_1| + B_1 v} \right) \right) \right)^2} \right], \quad (6.6)$$

with $\epsilon \sim \mathcal{N}(0, \sigma^2)$, $v \sim \mathcal{N}(0, \sigma^2)$, and B_k being the coefficient from (6.3), $B_k = \frac{C_K}{2^{2K} N} \binom{2K}{K+k}$.

To make use of Theorem 3, we note that the distance between a canonical distribution and its canonical noisy version is bounded by the distance due to noise and the distance due to canonicalizing the noisy version, i.e. $\|F - \tilde{F}^\epsilon\| \leq \|F - F^\epsilon\| + \|F^\epsilon - \tilde{F}^\epsilon\|$. The theorem lets us calculate the expected value of $\|F^\epsilon - \tilde{F}^\epsilon\|$ only as a function of $|F_1|$ relative to the variance of the noise, σ , without having to know f or f^ϵ . Notably, $E \left[\|F^\epsilon - \tilde{F}^\epsilon\| \right]$ approaches zero as $|F_1|$ grows relative to the noise. Because the norm is always non-negative, its expected value approaching zero implies that its variance is also approaching zero. This means that as noise gets smaller, $\|F - \tilde{F}^\epsilon\| \approx \|F - F^\epsilon\| \approx 0$, which is what we set out to show.

We illustrate this with a simulation (Figures 6.2 and 6.3). We first generate two random distributions, one with a large $|F_1|$ and one with a small $|F_1|$. For each of these distributions, we generate noisy versions for

a range of noise levels and calculate $\|F - F^\epsilon\|$ and $\|F^\epsilon - \tilde{F}^\epsilon\|$. Comparing Figures 6.2 and 6.3, we see that $\|F^\epsilon - \tilde{F}^\epsilon\|$ is expected to be smaller in Figure 6.2, where $|F_1|$ is large. For comparison, we plot the distance caused by rotating these distributions, $\|F - F_\phi\|$ (Figure 6.2(e) and 6.3(e)). In Figure 6.2, the distance caused by rotation is larger than the distance due to canonicalizing the noisy versions, but in Figure 6.3, the distance due to canonicalization is significant compared to the rotation distance.

As a concrete example, take a patch matching application, such as we describe in Section 6.3.1. If distributions in a dataset are randomly rotated and $|F_1|$ for each patch is large relative to the expected noise, it makes sense to canonicalize the distributions before matching because much of the distance between corresponding patches will come from their rotation, which canonicalization will remove. If, on the other hand, patches in the dataset are not rotated, canonicalization will hurt performance because $\|F^\epsilon - \tilde{F}^\epsilon\|$ will increase the distance between matching patches. As $|F_1|$ for the patches shrinks relative to the noise, canonicalization becomes increasingly unstable. This is because when $|F_1|$ is small, a small amount of noise can greatly affect $\arg(F_1)$. This situation can arise in two ways. The first is when all weights are small, meaning the distribution being calculated is essentially zero; instability is no problem in this case because rotation has no effect on FS-KDEs that are nearly zero. The second is when the distribution has symmetry, e.g., when Θ contains only angles only at zero and π . Such cases may arise in practice, leading us to explore a generalization of F_1 canonicalization that can remove these symmetries.

F_k Canonicalization

We can easily generalize the idea in (6.5) to rotating f such that any one of the F_k s is real,

$$\tilde{f}^\ell(\theta) \xrightarrow{FS} \tilde{F}_k^\ell = e^{-jk\arg(F_\ell)/\ell} F_k.$$

For any choice, $1 \leq \ell \leq K$, we can show that rotating the input set of angles does not affect the canonical version, using the same steps as for F_1 canonicalization.

The benefit of using $\ell > 1$ is that for angular distributions with a certain kind of symmetry, $|F_1|$ may be small (and thus F_1 canonicalization will not be robust to noise), while, e.g., $|F_2|$ may be large, meaning F_2 canonicalization will be robust to noise. The trade-off is that if $|F_1|$ and $|F_2|$ are of similar size, F_1 canonicalization will be more robust to noise. (To see this, note that F_2 canonicalization is just another mean subtraction, except that the mean is calculated by first doubling all the angles in Θ . This doubling can remove unwanted symmetry, but it also amplifies noise.)

In our experiments, we leverage this in the following way: When what is important is pairwise distances between FS-KDEs, then we can define a canonical distance,

$$\|f - g\|_{\text{canonical}} = 2\pi \min_{\phi} \|F - e^{-jk\phi} G\|.$$

Finding this distance requires an optimization over ϕ , so is not appropriate when many pairwise distances must be computed. A reasonable approximation, however, is

$$\|f - g\|_{F_k \text{ canonical}} = \min_{1 \leq \ell \leq K} \|\tilde{f}^\ell - \tilde{g}^\ell\|, \quad (6.7)$$

which only requires the calculation of K distances.

6.2.5 Practical Considerations

In this section, we explore a few practical considerations that must be taken into account when implementing the FS-KDE algorithm, including calculation of C_K , a normal approximation to (6.3), and creating approximate FS-KDEs via truncation.

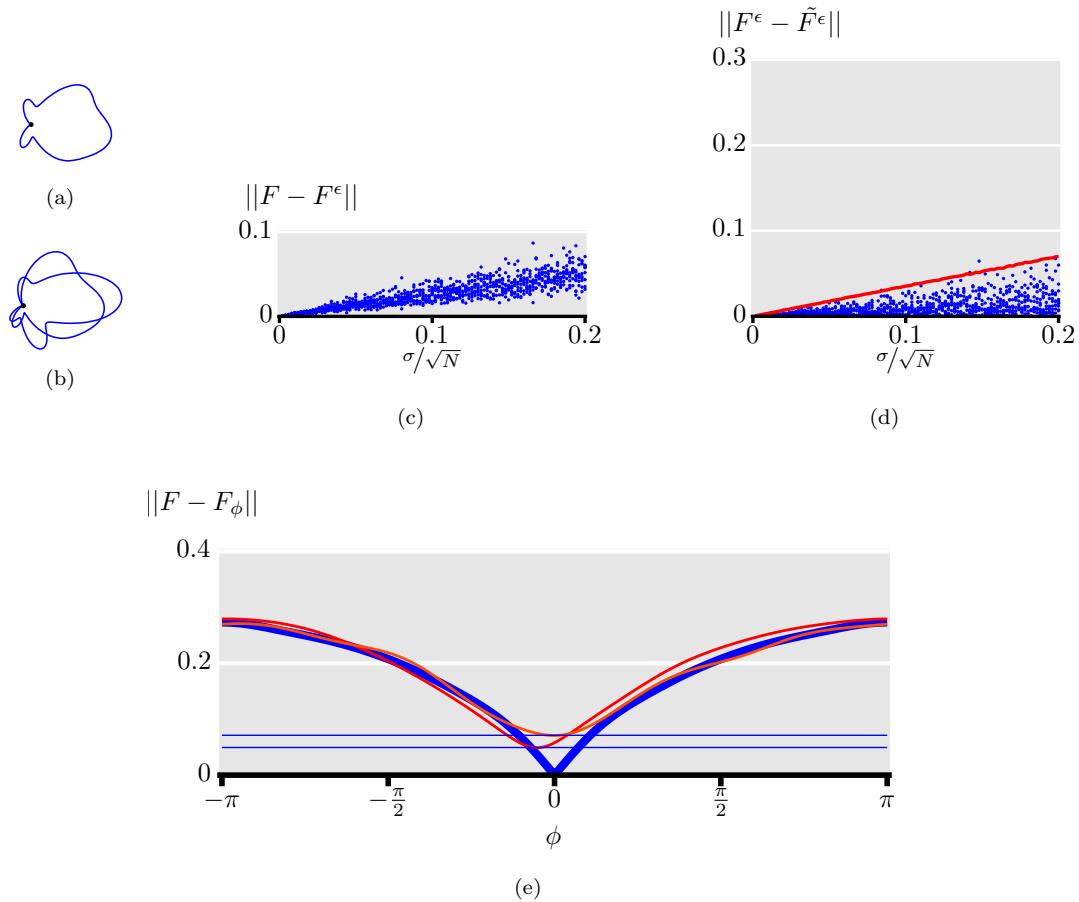


Figure 6.2: (a) A distribution, f . (b) Two of its noisy versions, f^ϵ . (c) The distance between f and its noisy versions plotted as a function of increasing noise. (d) The additional error introduced by F_1 canonicalization of the noisy versions, along with the expectation from Theorem 3. Because this distribution has a large $|F_1|$ value compared to the noise, F_1 canonicalization makes small changes to noisy versions of f . (e) Curves indicate $\|F - F_\phi\|$ (bold line) and $\|F - F_\phi^\epsilon\|$ (thin lines) for f , two noisy versions of f , and their rotations by ϕ . Because $|F_1|$ is large compared to the noise, $\|F - \tilde{F}_\phi^\epsilon\|$ (horizontal lines) is almost as small as $\min_\phi \|F - F_\phi^\epsilon\|$.

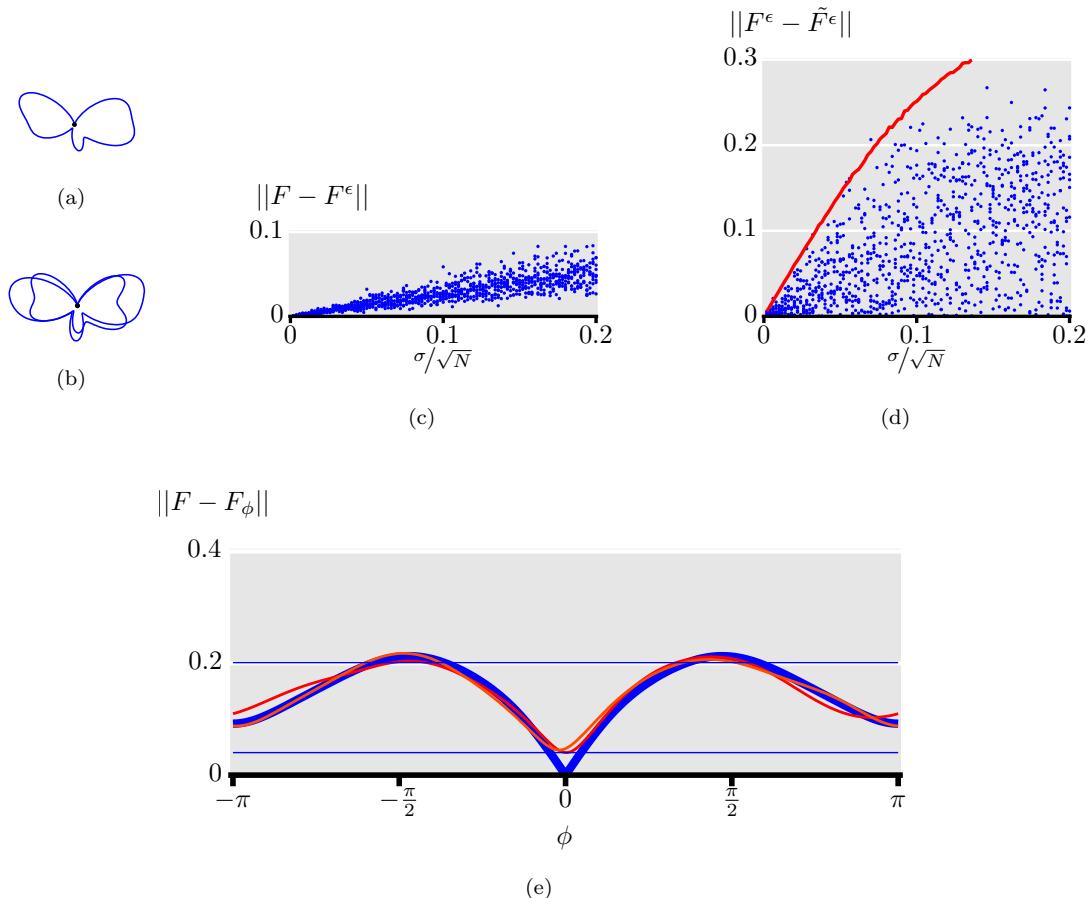


Figure 6.3: (a) A distribution, f . (b) Two of its noisy versions, f^ϵ . (c) The distance between f and its noisy versions plotted as a function of increasing noise. (d) The additional error introduced by F_1 canonicalization of the noisy versions, along with the expectation from Theorem 3. Because this distribution has a small $|F_1|$ value compared to the noise, F_1 canonicalization may make large changes to noisy versions of f . (e) Curves indicate $\|F - F_\phi\|$ (bold line) and $\|F - F_\phi^\epsilon\|$ (thin lines) for f , two noisy versions of f , and their rotations by ϕ . Because $|F_1|$ is small compared to the noise, $\|F - \tilde{F}_\phi^\epsilon\|$ (horizontal lines) is sometimes large.

Normalization

We have not yet calculated C_K , the normalizing constant for the kernel H in (6.2). We do this via a specific selection of (Θ, W) and (6.3), giving

$$\begin{aligned} \int_0^{2\pi} H(\theta) d\theta &\stackrel{(a)}{=} \int_0^{2\pi} f(\theta)|_{\Theta=\{0\}, W=\{1\}} d\theta \\ &\stackrel{(b)}{=} \int_0^{2\pi} \sum_{k=-K}^K \left(\frac{C_K}{2^{2K}} \binom{2K}{K+k} \right) e^{jk\theta} d\theta \\ &\stackrel{(c)}{=} \int_0^{2\pi} \frac{C_K}{2^{2K}} \binom{2K}{K} d\theta = \frac{C_K}{2^{2K-1}} \binom{2K}{K} \pi, \end{aligned}$$

where (a) follows from the definition of a KDE (6.1), (b) from the expression of an FS-KDE in terms of its Fourier series coefficients (6.3), and (c) from the symmetry of $e^{jk\theta}$ for $k \neq 0$. So to make the kernel, H , integrate to one, we set

$$C_K = \frac{2^{2K-1}}{\binom{2K}{K} \pi}. \quad (6.8)$$

Thus, the formula for Fourier series coefficients of the FS-KDE is

$$F_k = \frac{(K!)^2}{2\pi N(K-k)!(K+k)!} \sum_{n=0}^{N-1} w_n e^{-jk\theta_n}. \quad (6.9)$$

Normal Approximation

For large K s, the binomial coefficients in (6.3) and (6.8) can be replaced with a normal approximation,

$$\binom{n}{k} \approx \frac{1}{(1/2)^n \sqrt{n\pi/2}} e^{-\frac{((k-n)/2)^2}{n/2}},$$

giving an approximate version of (6.9),

$$F_k = \frac{1}{2\pi N} e^{-k^2/K} \sum_{n=0}^{N-1} w_n e^{-jk\theta_n}. \quad (6.10)$$

This approximation saves computation as compared to (6.9) and also reveals that the F_k s decay exponentially. The quality of the normal approximation improves as K increases; in our implementation we switch from (6.9) to (6.10) when $2K \geq 80$.

Truncation

In our current formulation, the bandwidth of the kernel density estimate is controlled by K , which also governs how many Fourier series terms are nonzero. Careful inspection of Figure 6.1 reveals that the kernels do not sharpen quickly as K increases: sharp kernels require a large K and therefore a long descriptor. One way to achieve sharp kernels with a shorter descriptor is through truncation. The approximation (6.10) reveals that when K is large, the decaying exponential term will cause $|F_k|$ to be very small for k near K . In fact,

$$\max_{\Theta} \frac{|F_k|}{|F_0|} = e^{-k^2/K}.$$

Thus, for a fixed K and a small truncation threshold ϵ , we create a truncated FS-KDE, \hat{F} , according to

$$\hat{F}_k = \begin{cases} 0 & \text{if } e^{-k^2/K} < \epsilon, \\ F_k & \text{otherwise.} \end{cases} \quad (6.11)$$

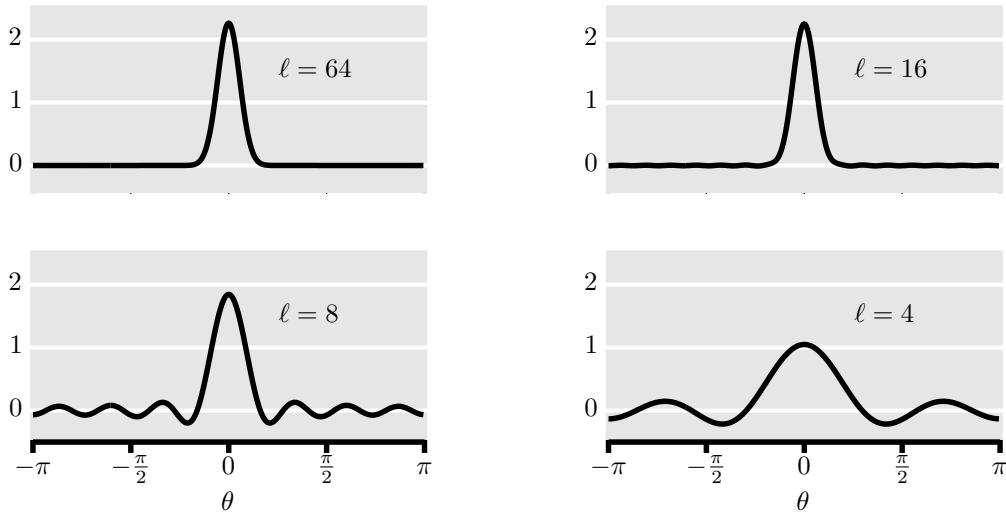


Figure 6.4: Examples of the powered cosine kernel of order 64 with different levels of truncation, where ℓ is the number of non-zero coefficients. Distortion is barely noticeable even when three quarters of the F_k s are set to zero (upper-right panel).

In the angular domain, truncation introduces distortion into the kernel, but this distortion is slight even when many coefficients are truncated (see Figure 6.4). We provide MATLAB code for the FS-KDE in the reproducible research compendium for this work, [148].

6.3 Experiments and Discussion

We now present experiments in keypoint description, person detection, and texture segmentation that show the promise of the FS-KDE as a tool in image processing.

6.3.1 Keypoint Description

A typical approach to image registration involves selecting keypoints from the images to be matched, finding pairs of corresponding keypoints, and solving for the transform based on the location of these pairs. One way to find correspondences between keypoints is to calculate a keypoint descriptor from the pixels around each keypoint. When two keypoints correspond, the distance between their descriptors should be low; when they do not, it should be high. A good keypoint descriptor should be highly discriminative while simultaneously being invariant to the transform that the registration aims to reverse.

We evaluate the performance of the FS-KDE as a keypoint descriptor using the University of British Columbia Multi-view Stereo Correspondence Dataset [12]. This dataset was constructed by extracting image patches around difference of Gaussian interest points in many images of the same few scenes (Statue of Liberty, Yosemite National Park, and Notre-Dame Cathedral). Depth maps of the scenes were used to determine which interest points match in 3D space, resulting in lists of corresponding and non-corresponding image patches (see Figure 6.1). The patches are 64×64 greyscale images; in our experiments we crop them to a circular region with a diameter of 60 pixels to avoid artifacts when rotating the patches.

We compare three simple keypoint descriptors: (1) The *raw intensity* descriptor is formed by concatenating the pixel values of the patch into a vector. It has dimension equal to the number of pixels in the patch, 2828. (2) The *gradient histogram* descriptor is formed by computing the image gradient at each pixel

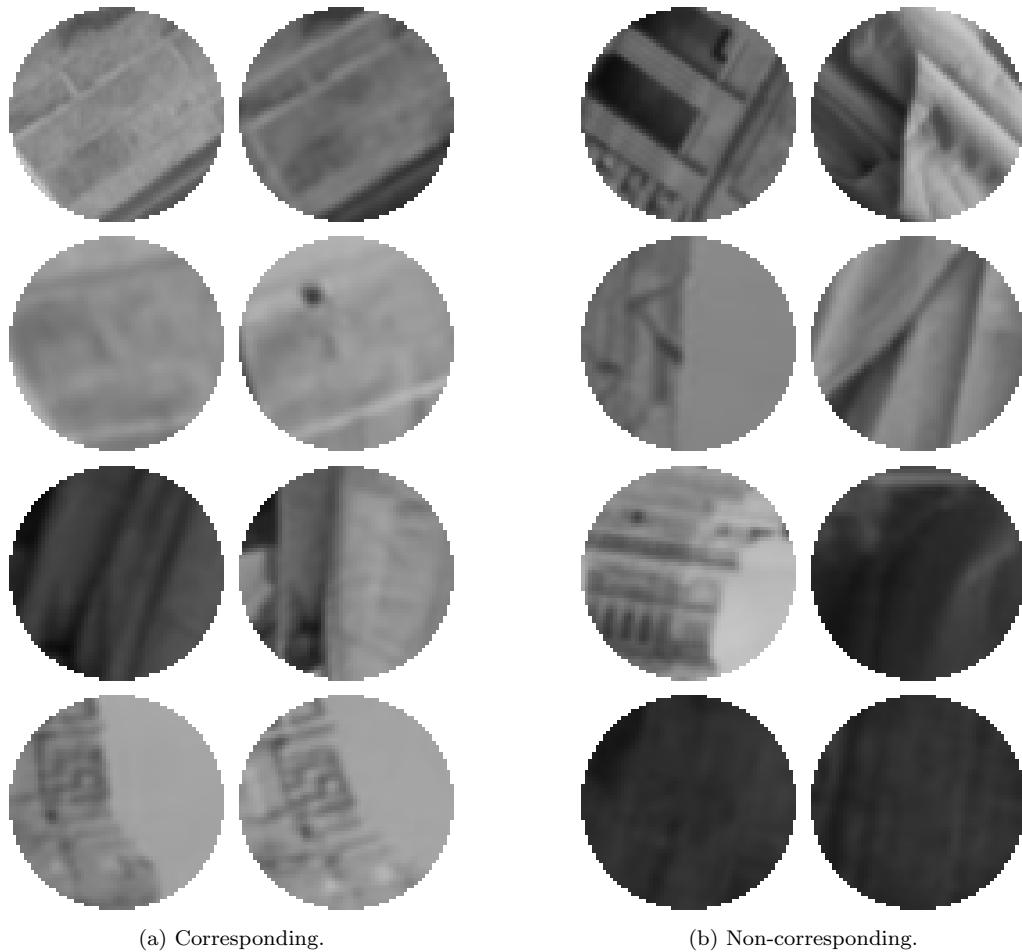


Figure 6.1: Examples of (a) corresponding and (b) non-corresponding pairs of patches from the British Columbia Multi-view Stereo Correspondence Dataset [12]. Though the patches are rotated to a canonical orientation, corresponding patches still exhibit viewpoint and intensity variation.

in the patch and forming a histogram of the gradient angles weighted by the norm of the gradient. The dimension of the descriptor is equal to the number of histogram bins; we vary it between 4 and 32. (3) The *FS-KDE* descriptor is formed by computing the image gradient at each pixel and computing an FS-KDE of the gradient angles weighted by the norm of the gradient. We truncate the descriptors according to (6.11) ($\epsilon = 1 \times 10^{-5}$) and we vary the descriptor length between 4 and 32.

Additionally, we compare three canonical versions of these descriptors: (1) The *canonical gradient histogram*, which is the same as the gradient histogram but with its bins rotated so that the first bin has the largest value. (2) The F_1 *canonical FS-KDE*, which follows the canonicalization procedure from (6.5). (3) The F_k *canonical FS-KDE*, which follows the canonicalization procedure from (6.7).

We use each of these methods to compute descriptors for 250,000 pairs of corresponding patches and 250,000 pairs of non-corresponding patches. Again, a good descriptor will give small distances between corresponding patches and large distances between non-corresponding patches. Figure 6.2 shows the distributions of these distances for the FS-KDE descriptor with dimension equal to sixteen. To quantify the performance of each descriptor, we estimate its probability of matching error, E . That is, if a corresponding and non-corresponding pair are selected at random, what is the probability that the descriptor distance between the non-corresponding pair is smaller than the descriptor distance between the corresponding pair? Given $C = \{c_0, c_1, \dots, c_{M-1}\}$ and $D = \{d_0, d_1, \dots, d_{N-1}\}$, lists of descriptor distances between corresponding and non-corresponding pairs with lengths M and N , respectively,

$$E = \sum_{i=0}^{M-1} \left(\frac{1}{M} \sum_{j=0}^{N-1} \frac{1}{N} \mathbf{1}_{d_j \leq c_i} \right), \quad (6.12)$$

where $\mathbf{1}_P$ is an indicator function equal to one when P is true and zero otherwise.

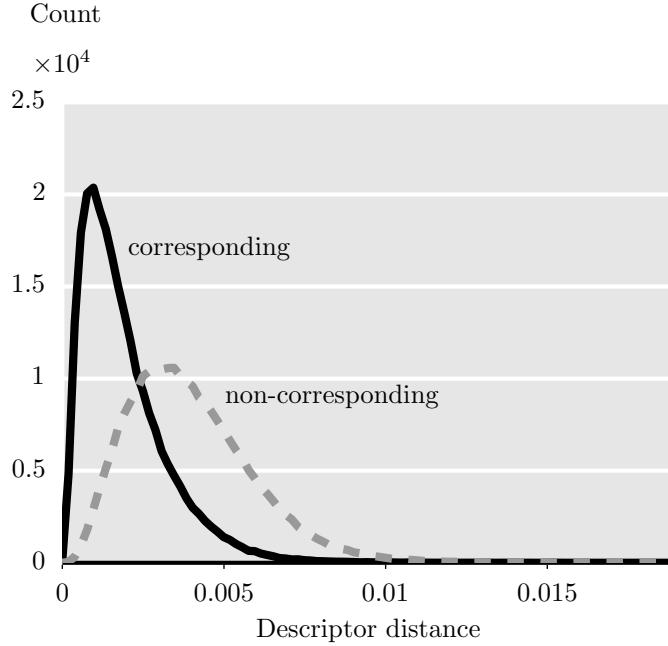


Figure 6.2: Distribution of distances between descriptors for corresponding and non-corresponding patches for the FS-KDE patch descriptor on 500,000 pairs of patches from the University of British Columbia Multi-view Stereo Correspondence Dataset [12]. Lines represent 50-bin histograms.

Figure 6.3 shows the results of our comparison of keypoint descriptors in terms of error probability, E . The FS-KDE shows the best overall performance, giving lower error chance for all descriptor sizes larger

than four and also achieving the lowest error chance of any of the descriptors (.168 at descriptor length 10). Its performance declines as the descriptor length increases, which is consistent with the idea that the descriptor becomes overly specific when it is long, increasing distances between corresponding patches and therefore increasing the error chance. The error chance of the gradient histogram descriptor also increases with descriptor size, for the same reason. We attribute the gap in performance between the FS-KDE and gradient histogram descriptors to the FS-KDE’s smooth handling of small rotations: Even though the patches in the dataset are rotated to a canonical orientation, small rotations do exist between corresponding patches (see Figure 6.1 for examples), which could distort the gradient histograms. The canonical versions of the FS-KDE and histograms perform uniformly worse than their non-canonical counterparts, which is as expected since no canonicalization should be necessary for this dataset. In this case, canonicalization will decrease the distance between non-corresponding patches more than it does for corresponding patches, increasing the error chance.

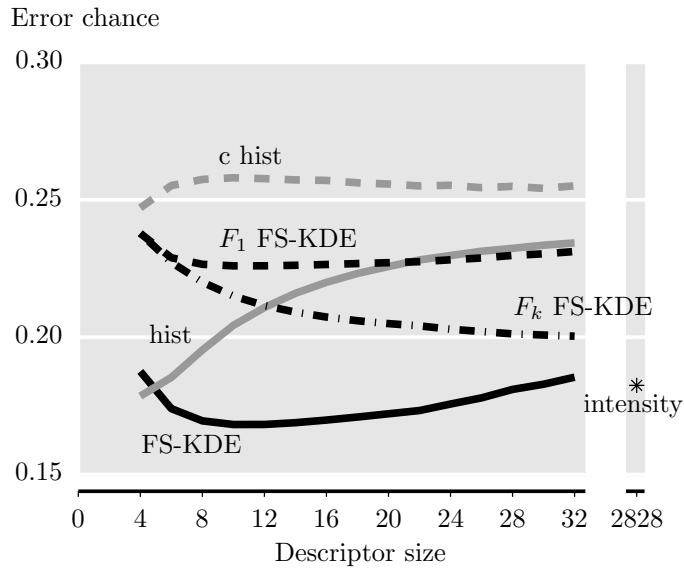


Figure 6.3: Estimated chance of a patch matching error (calculated via (6.12)) for intensity, gradient histogram, and FS-KDE descriptors of varying size. The best performance (lowest error chance) is achieved by the FS-KDE descriptor of size ten. The canonical descriptors perform poorly in this experiment because the patches are already in a canonical orientation.

In a separate experiment, we randomly rotated each patch in the dataset and ran the same comparison (Figure 6.4). As expected, this greatly increased the error rate for the histogram and FS-KDE descriptors (recall that the FS-KDE is a rotation invariant transform, but that the descriptor it creates is not rotation invariant). The canonical versions of these descriptors were mostly unaffected by the change, since they are rotation invariant descriptors. Both canonical versions of the FS-KDE were superior to the canonical gradient histogram, which we hypothesize is due to the robustness to noise of the FS-KDE canonicalization. The F_k canonicalization was better than the F_1 canonicalization, which suggests that symmetries of the type discussed in Section 6.2.4, which break the F_1 canonicalization, exist in this dataset.

6.3.2 Person Detection

We now evaluate the usefulness of the FS-KDE as a feature in a person detection application. A typical approach to person detection (or, in general, object detection), is to train a classifier on features which consist of distributions of angles. To preserve some spatial information, these distributions are calculated

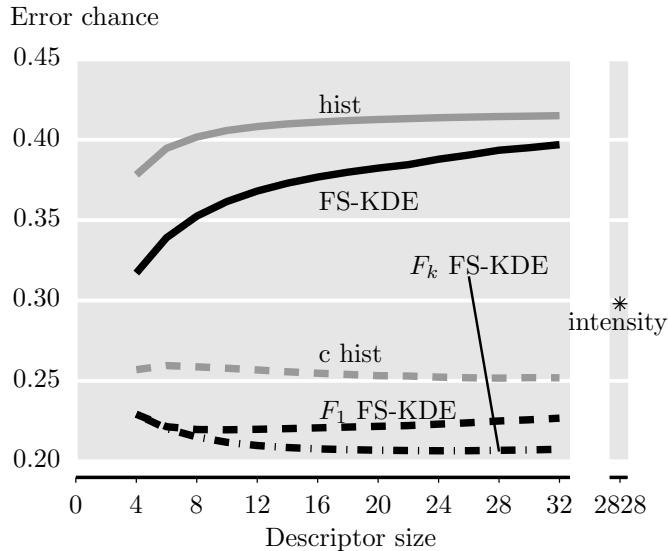


Figure 6.4: Estimated chance of a patch matching error for intensity, gradient histogram, and FS-KDE descriptors of varying size on patches with random orientation. The FS-KDE and gradient histogram descriptors cannot handle patch rotations, so have much higher error chance than in Figure 6.3. The best performance (lowest error chance) is now achieved by the FS-KDE descriptor with F_k canonicalization.

in a few windows of the input, e.g., upper left, upper right, lower left, lower right, and then concatenated together to form the final feature vector.

We use the INRIA person dataset [4] to do a comparison of feature detectors for human detection. This dataset is intended for supervised classification of images as containing a person (positive image) and not containing a person (negative image). It includes a training set of 2,416 positive and 1,218 negative images and a testing set of 1,126 positive and 453 negative images. For all images, we use the center 64×128 pixels for feature extraction.

We compare four feature extractors: (1) The *raw intensity* descriptor simply uses the pixels of the image as features and therefore has length 8192. (2) The *gradient histogram* descriptor separates the image into 8×8 -pixel blocks and computes a histogram of gradients inside each block, concatenating these histograms into a feature vector. We vary the number of bins per block from four to sixty-four. (3) The *HOG* descriptor, originally described in [4], also forms gradient histograms from blocks of the input image, but includes an additional block normalization step that can increase the feature's illumination invariance. We use the implementation in [149] and vary the number of orientations per block from four to sixty-four. (4) The *FS-KDE* descriptor forms truncated ($\epsilon = 1 \times 10^{-5}$) FS-KDEs for 8×8 -pixel blocks of the input. We vary the descriptor length per block from four to sixty-four.

We use each of these methods to extract features from the training and test sets. For each set of features, we train a linear SVM classifier (in MATLAB) on the training set, then use it to classify each image in the test set as negative or positive. In a separate experiment, we create new testing and training datasets by rotating each image between 15 and -15 degrees uniformly at random. This rotation is not enough that canonicalization is necessary, but is meant to test the robustness of the features to small rotations. We report accuracy, the number of correctly classified images in the testing set divided by the total number of images in the testing set, from both experiments in Figure 6.5.

For the unrotated set, the HOG features have the highest accuracy, except when the feature vector size is very small (four numbers per image block, resulting in 256 numbers per image). We consider HOG features to be state of the art for this problem, so their strong performance is not surprising; we suspect that the increase in performance over the gradient histogram and FS-KDE features comes from the normalization

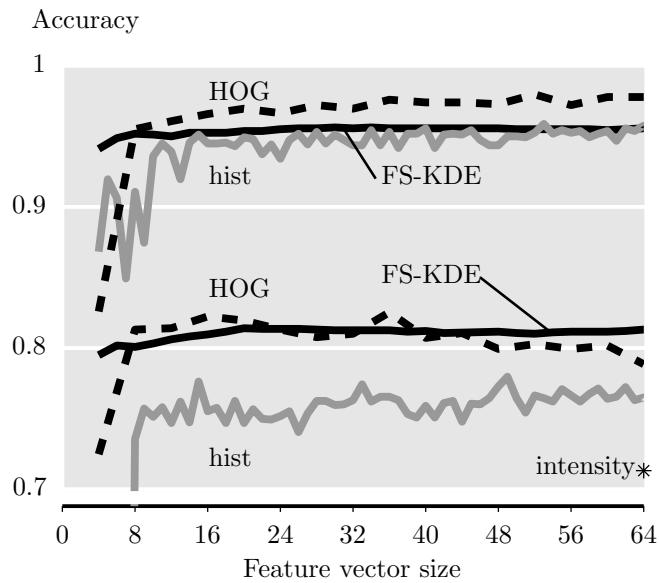


Figure 6.5: Accuracy on a person detection task for the intensity, gradient histogram, HOG, and FS-KDE feature extractors, plotted as a function of the feature vector length per image block. The top set of lines is for the INRIA person dataset [4], the bottom set is for the same dataset with small random rotations added. Without rotations, HOG features have the highest accuracy for most lengths and the intensity features (as expected) have the lowest. With rotations, the performance of all four methods declines, but the FS-KDE declines the least, leaving it with accuracy comparable to HOG. The accuracy of the intensity features for the rotated dataset was below chance and not plotted.

scheme used in HOG, which gives it an invariance to illumination changes missing in the other methods. The low accuracy of the intensity features is also as expected, given that greyscale intensity is not a reliable way to distinguish people from background clutter. The accuracy of the histogram gradient and FS-KDE features are similar, except that the accuracy of the FS-KDE features is more stable as the feature vector size changes. We attribute this to the lack of binning effects in the FS-KDE. We also note that the decline in performance as descriptor length increases seen in Figure 6.3 is not evident here because we use an SVM as opposed to simply calculating distances.

When a small amount (± 15 degrees) of rotation is added to the images in the dataset, the accuracy of all the feature sets decreases, but the decrease is smallest for the FS-KDE features. The rotation makes intensity features worse than chance (not plotted in Figure 6.5) because these features have no invariance to rotation. We suspect that binning artifacts (as discussed in Figure 6.1) explain the relatively larger decrease in accuracy for the gradient histogram and HOG features, because they both rely on gradient histograms. This experiment shows that the smooth kernel used in the FS-KDE provides greater invariance to small rotations than the binning employed by histograms, resulting in higher accuracy in the person detection task.

6.3.3 Texture Segmentation

Distributions of angles are also useful as texture features. In this experiment, we extend ORTSEG (Chapter 5) to include distributions of angles as well. We compare the effectiveness of histograms versus FS-KDEs to capture these distributions of angles.

We compare the methods on the *random texture* dataset from Section 5.4 plus an additional synthetic dataset, which we refer to as *dead leaves*. The dead leaves dataset (Figure 6.6) images each comprise three textures with the same color distributions but differently oriented edges. To create this dataset, we first pick three seed locations at random and use them to partition the image into the three Voronoi regions. We then generate the image via a dead leaves procedure: we sequentially place shapes of random color into the image at random locations until every pixel is covered. Depending on which of the three regions a shape lands in, it is selected to be either a vertical bar, horizontal bar, or diagonal bar. We select the ground truth label of each pixel to correspond to the shape that covered it most recently.

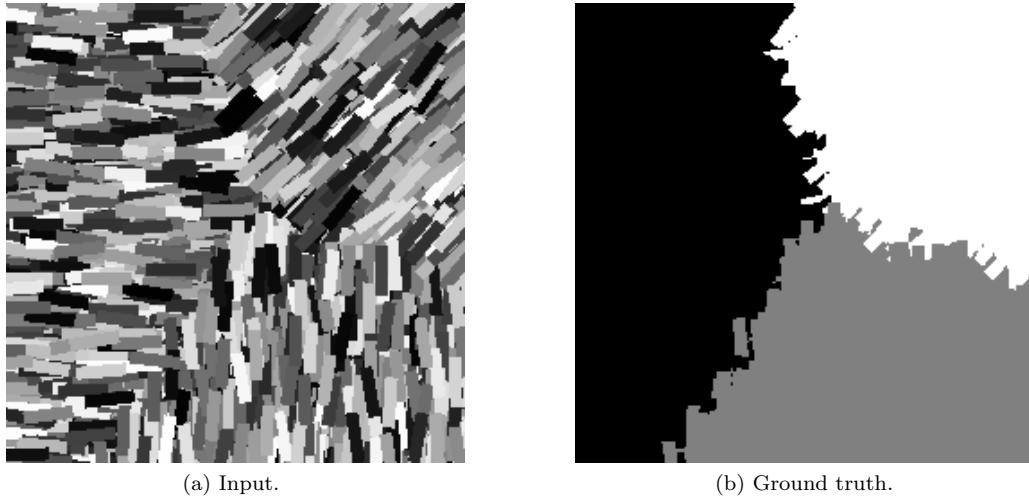


Figure 6.6: An example image and corresponding ground truth from the dead leaves dataset. For images like these, angular distributions are an important feature.

We compare three methods on this dataset. (1) *ORTSEG* is the original segmentation system described in Chapter 5, which relies only on color histograms. (2) *ORTSEG-hist* uses both color histograms and local

gradient histograms, with the number of bins selected from training between eight and forty. (3) *ORTSEG-FS-KDE* uses color histograms and local FS-KDEs of gradients, with no canonicalization, and with the number of complex coefficients selected from training between four and twenty. We do not evaluate canonical versions of these methods because canonicalization will make the angular distributions in the different texture regions of the dead leaves images match, resulting in low segmentation accuracy. The choice of whether or not to canonicalize for segmentation depends on whether textures that match except for their orientation are intended to be grouped together or not.

The experiment is structured exactly as in Section 5.4; in short, it is a leave-one-out cross validation. The results are reported in terms of Rand index [129, 130], which measures the fraction of pairs of pixels that are either in the same region in both the segmentation result and ground truth or in different regions in both the segmentation result and ground truth. It therefore ranges between zero and one, with one being perfect agreement with the ground truth.

The results of the segmentation experiment are given in Table 6.1. The three methods perform equally well on the random texture dataset (Figure 6.7), which makes sense because color information alone is enough to distinguish the textures. On the dead leaves dataset (Figure 6.8), the basic ORTSEG method, which relies only on color, cannot distinguish the textures at all and thus performs poorly. The gradient histogram and FS-KDE versions of ORTSEG improve performance by including edge information. That performance increase is most pronounced for the FS-KDE version. We attribute this difference to the FS-KDE’s smooth kernel giving better robustness to small variations in gradient angle as compared to histograms. These results serve as a proof of concept for the efficacy of the FS-KDE for including gradient information into the segmentation method ORTSEG.

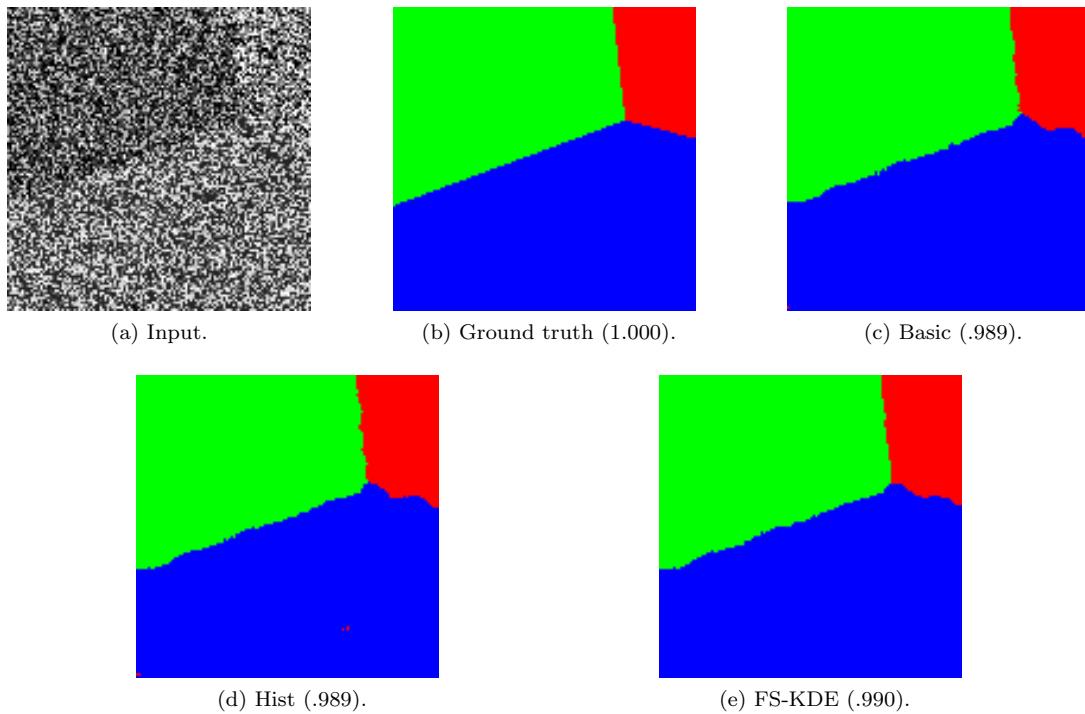


Figure 6.7: Example segmentation results for (c) the basic, (d) histogram, and (e) FS-KDE versions of ORTSEG on the random texture dataset. Rand indexes are given in parentheses. Here, color is enough to distinguish the textures and therefore all three methods perform well.

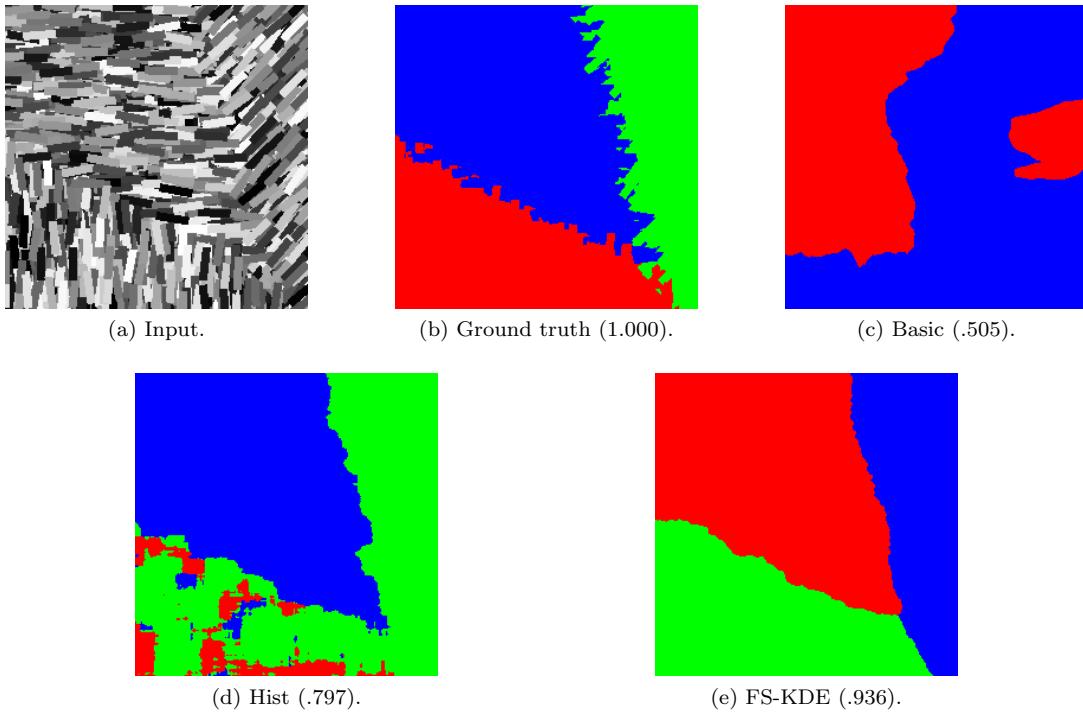


Figure 6.8: Example segmentation results for (c) the basic, (d) histogram, and (e) FS-KDE versions of ORTSEG on the dead leaves dataset. Rand indexes are given in parentheses. Here, color is not enough to distinguish the textures, so the basic version of ORTSEG fails.

6.4 Conclusions

In this work, we presented the FS-KDE, a method that computes a kernel density estimate of a set of angular quantities and represents it in terms of its Fourier series coefficients. We argue that the properties of the FS-KDE make it a useful tool in the myriad vision applications where gradient histograms are traditionally used.

In Section 6.2, we show that our specific choice of kernel ensures that the Fourier series is bandlimited, meaning that the continuous kernel density estimate is represented by a finite number of coefficients. We next describe the properties of the FS-KDE, most importantly that it is immune to binning artifacts: when the input angles rotate, the FS-KDE rotates by the same amount without distortion. We go on to describe two methods for canonicalizing the FS-KDE, i.e., aligning all FS-KDEs that are rotations of each to a single position, and prove a theorem about the robustness of one of these methods to noise. Finally, we present a version of the FS-KDE that works at every location in an image via linear filtering and discuss practical considerations for implementing these methods.

In Section 6.3, we compare FS-KDEs to histograms in the contexts of patch matching, person detection, and texture segmentation. The patch matching experiment shows that FS-KDE-based descriptors give a lower chance of making a patch matching error than histogram-based descriptors; it also shows that canonicalization improves the matching performance of the FS-KDE when patches are randomly rotated. The person detection experiment shows that FS-KDE features provide higher person detection accuracy than histogram features, especially when a small amount of random rotation is added to the dataset. Finally, the segmentation experiment suggests that the FS-KDE is better way to capture texture information than histograms in the context of texture segmentation. Taken together, these experiments provide strong proof of concept for the efficacy of the FS-KDE as a tool for describing distributions of angles in image processing

	Dataset	
Method	random texture	dead leaves
basic	0.989 \pm 0.002	0.551 \pm 0.087
hist	0.988 \pm 0.002	0.702 \pm 0.161
FS-KDE	0.989 \pm 0.002	0.907 \pm 0.094

Table 6.1: Comparison of the basic ORTSEG method with versions using gradient histograms and FS-KDEs in terms of Rand index mean and standard deviation. The augmented versions improve the performance on the dead leaves dataset, where edge information is critical.

applications.

Appendix

6.A Proof of Theorem 3

Proof.

$$\begin{aligned}
\|F^\epsilon - \tilde{F}^\epsilon\| &\stackrel{(a)}{=} \sqrt{\sum_{k=-K}^K |F_k^\epsilon - e^{-jk \arg(F_1^\epsilon)} F_k^\epsilon|^2} \\
&\stackrel{(b)}{=} \sqrt{\sum_{k=-K}^K (|F_k^\epsilon| |1 - e^{-jk \arg(F_1^\epsilon)}|)^2} \\
&\stackrel{(c)}{=} \sqrt{\sum_{k=-K}^K \left(|F_k^\epsilon| 2 \sin\left(\frac{k}{2} \arg(F_1^\epsilon)\right) \right)^2},
\end{aligned} \tag{6.13}$$

where (a) follows from the definition of norm, (b) from factoring, and (c) from Euler's formula and the fact that $|e^{j\theta}x| = |x|$ for all x .

In order to find $\arg(F_1^\epsilon)$, we note that $\arg(F_1^\epsilon) = \arg(F_1^\epsilon/\alpha)$ and that

$$F_1^\epsilon/\alpha = B_1 \sum_{n=0}^{N-1} w_n e^{-jk\theta_n} + \epsilon_n = F_1 + B_1 \sum_{n=0}^{N-1} \epsilon_n$$

by assumption and the definition of the FS-KDE (6.3). Therefore

$$\begin{aligned}
\arg(F_1^\epsilon) &= \arctan\left(\frac{\Im(F_1^\epsilon/\alpha)}{\Re(F_1^\epsilon/\alpha)}\right) \\
&= \arctan\left(\frac{B_1 \sum_{n=0}^{N-1} \Im(\epsilon_n)}{|F_1| + B_1 \sum_{n=0}^{N-1} \Re(\epsilon_n)}\right),
\end{aligned} \tag{6.14}$$

because F_1 is real by assumption.

To bound $|F_k^\epsilon|$, we use the fact that $|B_k \sum_{n=0}^{N-1} w_n e^{-jk\theta_n}| \leq |B_k \sum_{n=0}^{N-1} w_n|$ for any choice of (Θ, W) , meaning that

$$|F_k^\epsilon| \leq B_k N. \tag{6.15}$$

To finish the proof, we replace the sums in (6.14) with new random variables ϵ and v , the distributions of which we know because of the noise model assumed in the proof. \square

Chapter 7

Conclusions and Future Work

In this thesis, we have presented three tools for automated histology image analysis. Throughout the work, our guiding principle has been to develop tools rather than to address specific applications. We believe the variety of histology image analysis applications is much larger than the variability among H&E histology images, and therefore these tools will find wide applicability.

In Chapter 4, we presented a new benchmark dataset for H&E stain separation and used it to develop improvements to an existing stain separation method, suggesting that the currently accepted model of H&E image formation is incomplete. The next step for this work is to propose a new image formation model which can provide a solid theoretical basis for further development of stain separation methods.

In Chapter 5, we developed a model of histology images as occlusions of random textures. We used this model to develop a new unsupervised segmentation method, ORTSEG, and showed its efficacy with an experiment on a small two-tissue segmentation dataset. In the future, we would like to explore a multiscale and multiresolution version of ORTSEG that would allow simultaneous segmentation of structures of different sizes, e.g., nuclei and glands.

In Chapter 6, we developed a new descriptor of angular distributions, which we called the FS-KDE. We described its properties and showed that it compares favorably to angular histograms in several image processing applications. One future direction for the work would be to generalize the concept from the powered cosine kernel to a larger family of bandlimited kernels, which might lead to the discovery of kernels with better angular localization properties (i.e., a sharper lump using fewer coefficients). Another would be to include image gradients into the texture theory developed for ORTSEG. Finally, we need more experimentation to show that using angular distributions inside ORTSEG can improve segmentation in real-world images and especially histology images.

Bibliography

- [1] C. M. Christoudias, B. Georgescu, and P. Meer, “Synergism in low level vision,” in *Proc. IEEE Int. Conf. Pattern Recogn.*, vol. 4, Québec City, Aug. 2002, pp. 150–155.
- [2] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, 2004.
- [3] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.
- [4] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn.*, vol. 1, 2005, pp. 886–893.
- [5] Y. Deng and B. S. Manjunath, “Unsupervised segmentation of color-texture regions in images and video,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 800–810, Aug. 2001.
- [6] A. Chebira, J. A. Ozolek, C. A. Castro, W. G. Jenkinson, M. Gore, R. Bhagavatula, I. Khaimovich, S. E. Ormon, C. S. Navara, M. Sukhwani, K. E. Orwig, A. Ben-Yehudah, G. Schatten, G. K. Rohde, and J. Kovačević, “Multiresolution identification of germ layer components in teratomas derived from human and nonhuman primate embryonic stem cells,” in *Proc. IEEE Int. Symp. Biomed. Imag.*, Paris, May 2008, pp. 979–982.
- [7] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.
- [8] L. Shamir, N. Orlov, D. M. Eckley, T. Macura, J. Johnston, and I. G. Goldberg, “WND-CHARM: Multi-purpose image classification using compound image transforms,” *Pattern Recogn. Lett.*, vol. 29, pp. 1684–1693, 2008.
- [9] P. Tadrous, “Digital stain separation for histological images,” *J. Microsc.*, vol. 240, no. 2, pp. 164–172, Nov. 2010.
- [10] M. Macenko, M. Niethammer, J. S. Marron, D. Borland, J. T. Woosley, X. Guan, C. Schmitt, and N. E. Thomas, “A method for normalizing histology slides for quantitative analysis,” in *Proc. IEEE Int. Symp. Biomed. Imag.*, Boston, MA, Jun. 2009, pp. 1107–1110.
- [11] M. Haindl and S. Mikes, “Texture segmentation benchmark,” in *Proc. IEEE Int. Conf. Pattern Recogn.*, Dec. 2008, pp. 1–4.
- [12] S. Winder, G. Hua, and M. Brown, “Picking the best DAISY,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn.*, Miami Beach, FL, Jun. 2009, pp. 178–185.
- [13] National Center for Health Statistics. (2010) National hospital discharge survey, 2010. [Online]. Available: http://www.cdc.gov/nchs/nhds/nhds_questionnaires.htm
- [14] L. Pantanowitz, “Digital images and the future of digital pathology,” *J. Pathol. Inform.*, vol. 1, p. 15, 2010.

- [15] S. Papert. (1966) The summer vision project. [Online]. Available: <http://hdl.handle.net/1721.1/6125>
- [16] M. T. McCann, J. Majumdar, C. Peng, C. A. Castro, and J. Kovačević, “Algorithm and benchmark dataset for stain separation in histology images,” in *Proc. IEEE Int. Conf. Image Process.*, Paris, Oct. 2014, pp. 3953–3957.
- [17] M. T. McCann, D. G. Mixon, M. C. Fickus, C. A. Castro, J. A. Ozolek, and J. Kovačević, “Images as occlusions of textures: A framework for segmentation,” *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2033–2046, May 2014.
- [18] M. T. McCann, M. C. Fickus, and J. Kovačević, “Fourier series of kernel density estimates for rotation invariant angular distribution estimation,” *IEEE Trans. Image Process.*, 2015, submitted.
- [19] M. T. McCann, J. A. Ozolek, C. A. Castro, B. Parvin, and J. Kovačević, “Automated histology analysis: Opportunities for signal processing,” *IEEE Signal Process. Mag.*, vol. 32, no. 1, pp. 78–87, Jan. 2015.
- [20] K. S. Suvarna, C. Layton, and J. D. Bancroft, *Bancroft’s Theory and Practice of Histological Techniques*, 7th ed. London, UK: Churchill Livingstone, Nov. 2012.
- [21] D. J. Cook, *Cellular Pathology: An Introduction to Techniques and Applications*, 2nd ed. Oxfordshire, UK: Scion Publishing Ltd., Jun. 2006.
- [22] T. C. Cornish, R. E. Swapp, and K. J. Kaplan, “Whole-slide imaging: Routine pathologic diagnosis.” *Adv. Anat. Pathol.*, vol. 19, no. 3, pp. 152–159, May 2012.
- [23] S. Kothari, J. H. Phan, T. H. Stokes, and M. D. Wang, “Pathology imaging informatics for quantitative analysis of whole-slide images.” *J. Am. Med. Inform. Assoc.*, vol. 20, no. 6, pp. 1099–1108, Aug. 2013.
- [24] T. W. Bauer, L. Schoenfeld, R. J. Slaw, L. Yerian, Z. Sun, and W. H. Henricks, “Validation of whole slide imaging for primary diagnosis in surgical pathology.” *Arch. Pathol. Lab. Med.*, vol. 137, no. 4, pp. 518–524, Apr. 2013.
- [25] S. Krishnamurthy, K. Mathews, S. McClure, M. Murray, M. Gilcrease, C. Albaracin, J. Spinosa, B. Chang, J. Ho, J. Holt, A. Cohen, D. Giri, K. Garg, R. L. Bassett, and K. Liang, “Multi-institutional comparison of whole slide digital imaging and optical microscopy for interpretation of hematoxylin-eosin-stained breast tissue sections.” *Arch. Pathol. Lab. Med.*, vol. 137, no. 12, pp. 1733–1739, Dec. 2013.
- [26] L. Pantanowitz, J. H. Sinard, W. H. Henricks, L. A. Fatheree, A. B. Carter, L. Contis, B. A. Beckwith, A. J. Evans, A. Lal, A. V. Parwani, and College of American Pathologists Pathology and Laboratory Quality Center, “Validating whole slide imaging for diagnostic purposes in pathology: Guideline from the college of american pathologists pathology and laboratory quality center.” *Arch. Pathol. Lab. Med.*, vol. 137, no. 12, pp. 1710–1722, Dec. 2013.
- [27] S. Al-Janabi, A. Huisman, P. G. J. Nikkels, F. J. W. ten Kate, and P. J. van Diest, “Whole slide images for primary diagnostics of paediatric pathology specimens: A feasibility study,” *J. Clin. Pathol.*, vol. 66, no. 3, pp. 218–223, Mar. 2013.
- [28] E. Bengtsson and P. Malm, “Screening for Cervical Cancer Using Automated Analysis of PAP-Smears,” *Comput. Math. Methods Med.*, vol. 2014, p. 842037, Mar. 2014.
- [29] W. S. Rasband. (1997-2015) ImageJ. U. S. National Institutes of Health, Bethesda, MD. [Online]. Available: <http://imagej.nih.gov/ij>
- [30] M. T. McCann, R. Bhagavatula, M. C. Fickus, J. A. Ozolek, and J. Kovačević, “Automated colitis detection from endoscopic biopsies as a tissue screening tool in diagnostic pathology,” in *Proc. IEEE Int. Conf. Image Process.*, Orlando, FL, Sep. 2012, pp. 2809–2812.

- [31] M. Gurcan, L. Boucheron, A. Can, A. Madabhushi, N. Rajpoot, and B. Yener, "Histopathological image analysis: A review," *IEEE Rev. Biomed. Eng.*, vol. 2, pp. 147–171, Apr. 2009.
- [32] T. J. Fuchs and J. M. Buhmann, "Computational pathology: challenges and promises for tissue analysis." *Comput. Med. Imaging Graph.*, vol. 35, no. 7-8, pp. 515–530, Oct. 2011.
- [33] M. Veta, J. Pluim, P. van Diest, and M. Viergever, "Breast cancer histopathology image analysis: A review," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 5, pp. 1400–1411, May 2014.
- [34] M. W. Davidson. (2014, Feb.) Introduction to optical microscopy, digital imaging, and photomicrography. [Online]. Available: <http://micro.magnet.fsu.edu/primer/index.html>
- [35] S. Kothari, J. H. Phan, R. A. Moffitt, T. H. Stokes, S. E. Hassberger, Q. Chaudry, A. N. Young, and M. D. Wang, "Automatic batch-invariant color segmentation of histological cancer images," in *Proc. IEEE Int. Symp. Biomed. Imag.*, Chicago, IL, Mar. 2011, pp. 657–660.
- [36] A. M. Khan, N. Rajpoot, D. Treanor, and D. Magee, "A Non-Linear mapping approach to stain normalisation in digital histopathology images using image-specific colour deconvolution," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 6, pp. 1729–1738, Jun. 2014.
- [37] H. Chang, J. Han, A. Borowsky, L. Loss, J. W. Gray, P. T. Spellman, and B. Parvin, "Invariant delineation of nuclear architecture in glioblastoma multiforme for clinical and molecular association." *IEEE Trans. Med. Imag.*, vol. 32, no. 4, pp. 670–682, Apr. 2013.
- [38] U. D. Braumann, J. P. Kuska, J. Einenkel, L. C. Horn, M. Loffler, and M. Hockel, "Three-dimensional reconstruction and quantification of cervical carcinoma invasion fronts from histological serial sections," *IEEE Trans. Med. Imag.*, vol. 24, no. 10, pp. 1286–1307, Oct. 2005.
- [39] A. C. Ruifrok and D. A. Johnston, "Quantification of histochemical staining by color deconvolution." *Anal. Quant. Cytol. Histol.*, vol. 23, no. 4, pp. 291–299, Aug. 2001.
- [40] M. Gavrilovic, J. C. Azar, J. Lindblad, C. Wahlby, E. Bengtsson, C. Busch, and I. B. Carlstrom, "Blind color decomposition of histological images," *IEEE Trans. Med. Imag.*, vol. 32, no. 6, pp. 983–994, Jun. 2013.
- [41] P. A. Bautista and Y. Yagi, "Improving the visualization and detection of tissue folds in whole slide images through color enhancement." *J. Pathol. Inform.*, vol. 1, p. 25, Nov. 2010.
- [42] S. Kothari, J. H. Phan, and M. D. Wang, "Eliminating tissue-fold artifacts in histopathological whole-slide images for improved image-based prediction of cancer grade." *J. Pathol. Inform.*, vol. 4, p. 22, Aug. 2013.
- [43] J. Chappelow, J. E. Tomaszewski, M. Feldman, N. Shih, and A. Madabhushi, "HistoStitcher(©): An interactive program for accurate and rapid reconstruction of digitized whole histological sections from tissue fragments." *Comput. Med. Imaging Graph.*, vol. 35, no. 7-8, pp. 557–567, Mar. 2011.
- [44] D. Mueller, D. Vossen, and B. Hulskens, "Real-time deformable registration of multi-modal whole slides for digital pathology." *Comput. Med. Imaging Graph.*, vol. 35, no. 7-8, pp. 542–556, Jun. 2011.
- [45] M. Schwier, T. Böhler, H. K. Hahn, U. Dahmen, and O. Dirsch, "Registration of histological whole slide images guided by vessel structures," *J. Pathol. Inform.*, vol. 4(Suppl), p. S10, 2013.
- [46] S. Klein, M. Staring, K. Murphy, M. A. Viergever, and J. P. Pluim, "elastix: A toolbox for intensity-based medical image registration." *IEEE Trans. Med. Imag.*, vol. 29, no. 1, pp. 196–205, Jan. 2010.
- [47] L. Cooper, O. Sertel, J. Kong, G. Lozanski, K. Huang, and M. Gurcan, "Feature-based registration of histopathology images with different stains: an application for computerized follicular lymphoma prognosis." *Comput. Methods Programs Biomed.*, vol. 96, no. 3, pp. 182–192, Dec. 2009.

- [48] H. O. Helin, M. E. Lundin, M. Laakso, J. Lundin, H. J. Helin, and J. Isola, "Virtual microscopy in prostate histopathology: simultaneous viewing of biopsies stained sequentially with hematoxylin and eosin, and alpha-methylacyl-coenzyme a racemase/p63 immunohistochemistry." *J. Urol.*, vol. 175, no. 2, pp. 495–499, Feb. 2006.
- [49] G. Xiao, B. N. Bloch, J. Chappelow, E. M. Genega, N. M. Rofsky, R. E. Lenkinski, J. Tomaszewski, M. D. Feldman, M. Rosen, and A. Madabhushi, "Determining histology-MRI slice correspondences for defining MRI-based disease signatures of prostate cancer." *Comput. Med. Imaging Graph.*, vol. 35, no. 7-8, pp. 568–578, Jan. 2011.
- [50] H. Park, M. R. Piert, A. Khan, R. Shah, H. Hussain, J. Siddiqui, T. L. Chenevert, and C. R. Meyer, "Registration methodology for histological sections and in vivo imaging of human prostate." *Acad. Radiol.*, vol. 15, no. 8, pp. 1027–1039, Aug. 2008.
- [51] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image. Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.
- [52] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.
- [53] A. Goode, B. Gilbert, J. Harkes, D. Jukic, and M. Satyanarayanan, "OpenSlide: A vendor-neutral software foundation for digital pathology." *J. Pathol. Inform.*, vol. 4, p. 27, Sep. 2013.
- [54] Microsoft. (2015) Deep zoom file format overview. [Online]. Available: <https://msdn.microsoft.com/en-us/library/cc645077>
- [55] H. Chang, G. Fontenay, J. Han, G. Cong, F. Baehner, J. Gray, P. Spellman, and B. Parvin, "Morphometric analysis of TCGA glioblastoma multiforme," *BMC Bioinformatics*, vol. 12, p. 484, Dec. 2011.
- [56] TCGA Research Network. (2014, Aug.) The cancer genome atlas. [Online]. Available: <http://cancergenome.nih.gov/>
- [57] Zoomify, Inc. (1999-2015) Zoomify. [Online]. Available: <http://www.zoomify.com/>
- [58] W.-K. Jeong, J. Schneider, S. Turney, B. Faulkner-Jones, D. Meyer, R. Westermann, R. Reid, J. Lichterman, and H. Pfister, "Interactive histology of large-scale biomedical image stacks," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 1386–1395, Nov. 2010.
- [59] Leica Biosystems. (2015) Imagescope. [Online]. Available: <http://www.leicabiosystems.com/pathology-imaging/aperio-epathology/integrate/imagescope/>
- [60] PathCore. (2015) The sedein viewer project. [Online]. Available: <http://www.pathcore.ca/sedein/>
- [61] R. Marée, B. Stévens, L. Rollus, N. Rocks, X. M. Lopez, I. Salmon, D. Cataldo, and L. Wehenkel, "A rich internet application for remote visualization and collaborative annotation of digital slides in histology and cytology," *Diagn. Pathol.*, vol. 8(Suppl 1), p. S26, Sep. 2013.
- [62] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam, "Improved automatic detection and segmentation of cell nuclei in histopathology images." *IEEE Trans. Biomed. Eng.*, vol. 57, no. 4, pp. 841–852, Apr. 2010.
- [63] (2015) 3D-DOCTOR. [Online]. Available: <http://www.ablesw.com/3d-doctor/>
- [64] C.-H. H. Huang, A. Veillard, L. Roux, N. Loménie, and D. Racoceanu, "Time-efficient sparse analysis of histopathological whole slide images." *Comput. Med. Imaging Graph.*, vol. 35, no. 7-8, pp. 579–591, Dec. 2011.

- [65] V. Roullier, O. Lézoray, V.-T. Ta, and A. Elmoataz, “Multi-resolution graph-based analysis of histopathological whole slide images: application to mitotic cell extraction and visualization.” *Comput. Med. Imaging Graph.*, vol. 35, no. 7-8, pp. 603–615, May 2011.
- [66] J. P. Monaco, J. E. Tomaszewski, M. D. Feldman, I. Hagemann, M. Moradi, P. Mousavi, A. Boag, C. Davidson, P. Abolmaesumi, and A. Madabhushi, “High-throughput detection of prostate cancer in histological sections using probabilistic pairwise Markov models.” *Med. Image Anal.*, vol. 14, no. 4, pp. 617–629, Aug. 2010.
- [67] J. Kong, O. Sertel, H. Shimada, K. L. Boyer, J. H. Saltz, and M. N. Gurcan, “Computer-aided evaluation of neuroblastoma on whole-slide histology images: Classifying grade of neuroblastic differentiation,” *Pattern Recogn.*, vol. 42, no. 6, pp. 1080–1092, Jun. 2009.
- [68] H. Fatakdawala, J. Xu, A. Basavanhally, G. Bhanot, S. Ganesan, M. Feldman, J. E. Tomaszewski, and A. Madabhushi, “Expectation-maximization-driven geodesic active contour with overlap resolution (EMaGACOR): application to lymphocyte segmentation on breast cancer histopathology.” *IEEE Trans. Biomed. Eng.*, vol. 57, no. 7, pp. 1676–1689, Jul. 2010.
- [69] J. Han, T. Breckon, D. Randell, and G. Landini, “The application of support vector machine classification to detect cell nuclei for automated microscopy,” *Mach. Vis. Appl.*, vol. 23, no. 1, pp. 15–24, Jun. 2012.
- [70] E. D. Gelasca, B. Obara, D. Fedorov, K. Kvilekval, and B. S. Manjunath, “A biosegmentation benchmark for evaluation of bioimage analysis methods,” *BMC Bioinformatics*, vol. 10, p. 368, Nov. 2009.
- [71] M. Sühling, M. Arigovindan, P. Hunziker, and M. Unser, “Multiresolution moment filters: Theory and applications,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 484–495, Apr. 2004.
- [72] K. Belkacem-Boussaid, M. Pennell, G. Lozanski, A. Shana’ah, and M. Gurcan, “Computer-aided classification of centroblast cells in follicular lymphoma.” *Anal. Quant. Cytol. Histol.*, vol. 32, no. 5, pp. 254–260, Oct. 2010.
- [73] L. Roux, D. Racoceanu, N. Loménie, M. Kulikova, H. Irshad, J. Klossa, F. Capron, C. Genestie, G. Le Naour, and M. N. Gurcan, “Mitosis detection in breast cancer histological images An ICPR 2012 contest.” *J. Pathol. Inform.*, vol. 4, p. 8, May 2013.
- [74] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, “Mitosis detection in breast cancer histology images with deep neural networks,” *Med. Image Comput. Comput. Assist. Interv.*, vol. 16(Pt 2), pp. 411–418, 2013.
- [75] E. Ozdemir and C. Gunduz-Demir, “A hybrid classification model for digital pathology using structural and statistical pattern recognition,” *IEEE Trans. Med. Imag.*, vol. 32, no. 2, pp. 474–483, Feb. 2013.
- [76] K. Belkacem-Boussaid, S. Samsi, G. Lozanski, and M. N. Gurcan, “Automatic detection of follicular regions in H&E images using iterative shape index,” *Comput. Med. Imaging Graph.*, vol. 35, no. 7-8, pp. 592–602, Oct. 2011.
- [77] C. Chen, J. A. Ozolek, W. Wang, and G. K. Rohde, “A general system for automatic biomedical image segmentation using intensity neighborhoods,” *Int. J. Biomed. Imag.*, vol. 2011, Mar. 2011.
- [78] A. B. Tosun, M. Kandemir, C. Sokmensuer, and C. Gunduz-Demir, “Object-oriented texture analysis for the unsupervised segmentation of biopsy images for cancer detection,” *Pattern Recogn.*, vol. 42, no. 6, pp. 1104–1112, Jun. 2009.
- [79] R. Bhagavatula, M. T. McCann, M. C. Fickus, C. A. Castro, J. A. Ozolek, and J. Kovačević, “A vocabulary for the identification of teratoma tissue in H&E-stained samples,” *J. Pathol. Inform.*, vol. 5, p. 19, Jun. 2014.

- [80] J. Diamond, N. H. Anderson, P. H. Bartels, R. Montironi, and P. W. Hamilton, "The use of morphological characteristics and texture analysis in the identification of tissue composition in prostatic neoplasia," *Hum. Pathol.*, vol. 35, no. 9, pp. 1121–1131, Sep. 2004.
- [81] A. Cagri, A. Burak, C. Aykanat, C. Sokmensuer, and C. Gunduz-Demir, "Multilevel segmentation of histopathological images using cooccurrence of tissue objects." *IEEE Trans. Biomed. Eng.*, vol. 59, no. 6, pp. 1681–1690, Jun. 2012.
- [82] S. Kothari, J. Phan, A. Young, and M. Wang, "Histological image classification using biologically interpretable shape-based features," *BMC Med. Imag.*, vol. 13, no. 9, Mar. 2013.
- [83] Y. Zhang, B. Zhang, F. Coenen, and W. Lu, "Breast cancer diagnosis from biopsy images with highly reliable random subspace classifier ensembles," *Mach. Vis. Appl.*, vol. 24, no. 7, pp. 1405–1420, Oct. 2013.
- [84] H. Chang, N. Nayak, P. Spellman, and B. Parvin, "Characterization of tissue histopathology via predictive sparse decomposition and spatial pyramid matching," in *Proc. Int. Conf. Med. Image Comput. Computer-Assisted Intervention*, Nagoya, Sep. 2013, vol. 8150, pp. 91–98.
- [85] S. Petushi, F. Garcia, M. Haber, C. Katsinis, and A. Tozeren, "Large-scale computations on histology images reveal grade-differentiating parameters for breast cancer," *BMC Med. Imag.*, vol. 6, p. 14, Oct. 2006.
- [86] M. D. DiFranco, G. O'Hurley, E. W. Kay, R. W. Watson, and P. Cunningham, "Ensemble based system for whole-slide prostate cancer probability mapping using color texture features." *Comput. Med. Imaging Graph.*, vol. 35, no. 7-8, pp. 629–645, Jan. 2011.
- [87] C.-W. Wang and C.-P. Yu, "Automated morphological classification of lung cancer subtypes using H&E tissue images," *Mach. Vis. Appl.*, vol. 24, no. 7, pp. 1383–1391, Oct. 2013.
- [88] T. D. Dietterich, R. H. Lathrop, and T. Lozano-Perez, "Solving the Multiple-Instance problem with Axis-Parallel rectangles," *Artif. Intell.*, vol. 89, pp. 31–71, 1997.
- [89] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Englewood Cliffs, NJ: Prentice Hall, 2002.
- [90] G. W. Gill and D. M. Wood, "Biological microtechniques," *DAKO Connection*, vol. 14, pp. 144–150, 2010.
- [91] M. T. McCann, J. Majumdar, C. Peng, C. A. Castro, and J. Kovačević. (2014, Jan.) Algorithm and benchmark dataset for stain separation in histology images. [Online]. Available: http://www.jelena.ece.cmu.edu/repository/rr/14_McCannMPCK/14_McCannMPCK.html
- [92] P. Atkins and J. d. Paula, *Physical Chemistry, 9th Edition*, 9th ed. New York, NY: W. H. Freeman, Dec. 2009.
- [93] C. Beecks, M. Uysal, and T. Seidl, "A comparative study of similarity measures for content-based multimedia retrieval," in *Proc (IEEE Int. Conf. Multimed. Expo.)*, Suntec City, 2010, pp. 1552–1557.
- [94] Y. Cao. (2014, Jan.) Hungarian algorithm for linear assignment problems (v2.3). [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/20652-hungarian-algorithm-for-linear-assignment-problems-v2-3>
- [95] D. R. Li, G. Zhang, Z. Wu, and L. Yi, "An edge embedded marker-based watershed algorithm for high spatial resolution remote sensing image segmentation," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2781–2787, Oct. 2010.

- [96] N. Jacobson, Y.-L. Lee, V. Mahadevan, N. M. Vasconcelos, and T. Q. Nguyen, “A novel approach to FRUC using discriminant saliency and frame segmentation,” *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2924–2934, Nov. 2010.
- [97] H. Ayasso and A. Mohammad-Djafari, “Joint NDT image restoration and segmentation using Gauss-Markov-Potts prior models and variational Bayesian computation,” *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2265–2277, Sep. 2010.
- [98] A. Gooya, H. Liao, K. Matsumiya, K. Masamune, Y. Masutani, and T. Dohi, “A variational method for geometric regularization of vascular segmentation in medical images,” *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1295–1312, Aug. 2008.
- [99] C. Zanella, M. Campana, B. Rizzi, C. Melani, G. Sanguinetti, P. Bourgine, K. Mikula, N. Peyriéras, and A. Sarti, “Cells segmentation from 3-D confocal images of early zebrafish embryogenesis,” *IEEE Trans. Image Process.*, vol. 19, no. 3, pp. 770–781, Mar. 2010.
- [100] G. Srinivasa, M. C. Fickus, Y. Guo, A. D. Linstedt, and J. Kovačević, “Active mask segmentation of fluorescence microscope images,” *IEEE Trans. Image Process.*, vol. 18, no. 8, pp. 1817–1829, Aug. 2009.
- [101] K.-C. J. Chen, Y. Yu, R. Li, H.-C. Lee, G. Yang, and J. Kovačević, “Adaptive active-mask image segmentation for quantitative characterization of mitochondrial morphology,” in *Proc. IEEE Int. Conf. Image Process.*, Orlando, FL, Sep. 2012, pp. 2033–2036.
- [102] D. B. Chklovskii, S. Vitaladevuni, and L. K. Scheffer, “Semi-automated reconstruction of neural circuits using electron microscopy,” *Curr. Opin. Neurobiol.*, vol. 20, no. 5, pp. 667–675, Oct. 2010.
- [103] J. Nascimento and J. Marques, “Robust shape tracking with multiple models in ultrasound images,” *IEEE Trans. Image Process.*, vol. 17, no. 3, pp. 392–406, 2008.
- [104] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1988.
- [105] R. Malladi, J. A. Sethian, and B. Vemuri, “Shape modeling with front propagation: A level set approach,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 2, pp. 158–175, Feb. 1995.
- [106] A. Jain and F. Farrokhnia, “Unsupervised texture segmentation using Gabor filters,” *Pattern Recogn.*, vol. 24, pp. 1167–1186, 1991.
- [107] H. Derin and W. S. Cole, “Segmentation of textured images using Gibbs random fields,” *Computer Vis. Graph. Image Process.*, vol. 35, no. 1, pp. 72–98, Jul. 1986.
- [108] L. Vincent and P. Soille, “Watersheds in digital spaces: An efficient algorithm based on immersion simulations,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, Jun. 1991.
- [109] R. Adams and L. Bischof, “Seeded region growing,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 641–647, 1994.
- [110] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
- [111] C. Carson, S. Belongie, H. Greenspan, and J. Malik, “Blobworld: Image segmentation using Expectation-Maximization and its application to image querying,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 1026–1038, 1999.
- [112] D. E. Ilea and P. F. Whelan, “CTex—An adaptive unsupervised segmentation algorithm based on color-texture coherence,” *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1926–1939, Oct. 2008.

- [113] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam, “Use of active shape models for locating structures in medical images,” *Image Vis. Comput.*, vol. 12, no. 6, pp. 355–365, Jul. 1994.
- [114] J. A. Ozolek and C. A. Castro, *Teratomas Derived from Embryonic Stem Cells as Models for Embryonic Development, Disease, and Tumorigenesis*. InTech, 2011, ch. 13.
- [115] M. L. Massar, R. Bhagavatula, M. C. Fickus, and J. Kovačević, “Local histograms and image occlusion models,” *Appl. Comput. Harmon. Anal.*, vol. 34, no. 3, pp. 469–487, May 2013.
- [116] A. B. Lee, D. Mumford, and J. Huang, “Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model,” *Int. J. Comput. Vis.*, vol. 41, pp. 35–59, 2001.
- [117] O. Michailovich, Y. Rathi, and A. Tannenbaum, “Image segmentation using active contours driven by the bhattacharyya gradient flow,” *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2787–2801, Nov. 2007.
- [118] R. Courant, “Variational methods for the solution of problems of equilibrium and vibrations,” *Bull. Am. Math. Soc.*, vol. 49, no. 1, pp. 1–24, Jan. 1943.
- [119] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, “Fast image recovery using variable splitting and constrained optimization,” *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–2356, Sep. 2010.
- [120] M. L. Fisher, “An applications oriented guide to lagrangian relaxation,” *Interfaces*, vol. 15, no. 2, pp. 10–21, Mar. 1985.
- [121] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.
- [122] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, “Algorithms and applications for approximate nonnegative matrix factorization,” *Comput. Stat. Data Anal.*, vol. 52, no. 1, pp. 155–173, Sep. 2007.
- [123] P. Sarder and A. Nehorai, “Deconvolution methods for 3-D fluorescence microscopy images,” *IEEE Signal Process. Mag., sp. iss. Molec. Cellular Bioimaging*, vol. 23, no. 3, pp. 32–45, May 2006.
- [124] P. Paatero and U. Tapper, “Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values,” *Environmetrics*, vol. 5, no. 2, pp. 111–126, Jun. 1994.
- [125] L. I. Rudin, Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D*, no. 1–4, pp. 259–268, Nov. 1992.
- [126] A. K. Jain, “Data clustering: 50 years beyond k-means,” *Pattern Recogn. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.
- [127] M. T. McCann, D. G. Mixon, M. Fickus, C. A. Castro, J. A. Ozolek, and J. Kovačević. (2013) Images as occlusions of textures: A framework for segmentation. [Online]. Available: http://www.jelena.ece.cmu.edu/repository/rr/13_McCannMFCOK/13_McCannMFCOK.html
- [128] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, Vancouver, July 2001, pp. 416–423.
- [129] W. M. Rand, “Objective criteria for the evaluation of clustering methods,” *J. Am. Stat. Assoc.*, vol. 66, no. 336, pp. 846–850, Dec. 1971.

- [130] R. Unnikrishnan, C. Pantofaru, and M. H. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 929–944, Jun. 2007.
- [131] D. E. Ilea and P. F. Whelan, "Image segmentation based on the integration of colour-texture descriptors—A review," *Pattern Recogn.*, vol. 44, no. 10-11, pp. 2479–2501, Oct. 2011.
- [132] M. Meilă, "Comparing clusterings by the variation of information," in *Learning Theory and Kernel Machines*, ser. Lecture Notes in Computer Science, B. Schölkopf and M. Warmuth, Eds. Springer Berlin Heidelberg, 2003, vol. 2777, pp. 173–187.
- [133] A. Dvoretzky, J. Kiefer, and J. Wolfowitz, "Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator," *Ann. Math. Stat.*, vol. 27, no. 3, pp. 642–669, Sep. 1956.
- [134] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, Kerkyra, 1999, pp. 1150–1157 vol.2.
- [135] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.
- [136] E. Mortensen, H. Deng, and L. Shapiro, "A SIFT descriptor with global context," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn.*, vol. 1, San Diego, CA, Jun. 2005, pp. 184–190 vol. 1.
- [137] A. Abdel-Hakim and A. Farag, "CSIFT: A SIFT descriptor with color invariant characteristics," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn.*, vol. 2, New York, NY, 2006, pp. 1978–1983.
- [138] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn.*, vol. 2, Washington, DC, Jun. 2004, pp. II–506 – II–513 Vol.2.
- [139] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable medical image registration: A survey," *IEEE Trans. Med. Imag.*, vol. 32, no. 7, pp. 1153–1190, Jul. 2013.
- [140] J. Aggarwal and M. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, no. 3, pp. 16:1–16:43, Apr. 2011.
- [141] A. Ramanan and M. Niranjan, "A review of codebook models in patch-based visual object recognition," *J. Signal Process. Syst.*, vol. 68, no. 3, pp. 333–352, Sep. 2012.
- [142] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
- [143] D. Geronimo, A. Lopez, A. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1239–1258, Jul. 2010.
- [144] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
- [145] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod, "Fast computation of rotation-invariant image features by an approximate radial gradient transform," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 2970–2982, Aug. 2013.
- [146] L. Wasserman, *All of Statistics: A Concise Course in Statistical Inference*. New York: Springer, Sep. 2004.

- [147] M. Vetterli, J. Kovačević, and V. K. Goyal, *Foundations of Signal Processing*. Cambridge: Cambridge University Press, 2014, <http://www.fourierandwavelets.org/>.
- [148] M. T. McCann, M. Fickus, and J. Kovačević. (2015) Fourier series of kernel density estimates for rotation invariant angular distribution estimation. [Online]. Available: http://www.jelena.ece.cmu.edu/repository/rr/15_McCannFK/15_McCannFK.html
- [149] A. Vedaldi and B. Fulkerson. (2008) VLFeat: An open and portable library of computer vision algorithms. [Online]. Available: <http://www.vlfeat.org/>