



HAL
open science

From digital to computational pathology for biomarker discovery

Daniel Felipe González Obando

► **To cite this version:**

Daniel Felipe González Obando. From digital to computational pathology for biomarker discovery. Medical Imaging. Université Paris Cité, 2019. English. NNT : 2019UNIP5185 . tel-04072622

HAL Id: tel-04072622

<https://theses.hal.science/tel-04072622>

Submitted on 18 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

présentée pour obtenir le grade de

Docteur de l'Université de Paris
Spécialité: Informatique

École Doctorale Informatique, Télécommunications et Électronique (ED 130)

Institut Pasteur - Unité d'Analyse d'Images Biologiques
LIPADE - Laboratoire d'Informatique Paris Descartes

From Digital to Computational Pathology for Biomarker Discovery

Par Daniel Felipe González Obando

Présentée et soutenue publiquement le 29 novembre 2019

Devant un jury composé de :

<i>Directeurs de thèse</i>	Jean-Christophe Olivo-Marin	Directeur de recherche, Institut Pasteur
	Laurent Wendling	Professeur, Université Paris
<i>Rapporteurs</i>	Isabelle Debled-Rennesson	Professeur, Université de Lorraine
	Marc Pierrot-Deseilligny	Directeur de recherche, Institut Géographique National
<i>Examineurs</i>	Georges Stamon	Professeur, Université de Paris
	Vannary Meas-Yedid	Docteur (PhD), Institut Pasteur
	Dominique Béréziat	Maître de Conférences, Sorbonne Université
	Raphaël Marée	Docteur (PhP), Université de Liège

Résumé (français) :

L’histopathologie a pour objectif d’analyser des images de tissus biologiques pour évaluer l’état pathologique d’un organe et établir un diagnostic. L’apparition des scanners de lames à haute résolution a ouvert la voie à des nouvelles possibilités d’acquisition de très grandes images (whole slide imaging), de multiplexage de marquages, d’extraction exhaustive d’informations visuelles et d’annotations multiples à large échelle. Cette thèse propose un ensemble de méthodes algorithmiques visant à faciliter et optimiser ces différents aspects. Dans un premier temps, nous proposons une méthode de recalage multi-échelle d’images histologiques multi-marquées reposant sur les propriétés des B-splines pour modéliser, de façon continue, une image discrète. Nous proposons ensuite de nouvelles approches d’analyse morphologique sur des polygones faiblement simples, généralisés par des graphes à segments droits. Elles reposent sur le formalisme des squelettes droits (une approximation de squelettes courbes définis par des segments droits), construits à l’aide de graphes de motocyclettes. Cette structure permet de réaliser des opérations de morphologie mathématiques sur des polygones avec une complexité réduite. La précision des opérations sur des polygones bruités est obtenue en raffinant la construction des squelettes droits par ajout adaptatif de sommets. Nous avons aussi proposé un algorithme de détection de l’axe médian et montré qu’il est possible de reconstruire la forme d’origine avec une approximation arbitraire. Enfin, nous avons exploré les squelettes droits pondérés qui permettent des opérations morphologiques directionnelles. Ces approches d’analyse morphologique offrent un support consistant pour améliorer la segmentation des objets grâce à l’information contextuelle et réaliser des études liées à l’analyse spatiale des interactions entre les différentes structures d’intérêt au sein du tissu. Tous les algorithmes proposés sont optimisés pour le traitement d’images gigapixels et garantissent une reproductibilité des analyses, notamment grâce à la création du plugin Icytomine, interface entre Icy et Cytomine.

Abstract:

Histopathology aims to analyze images of biological tissues to assess the pathological condition of an organ and to provide a diagnosis. The advent of high-resolution slide scanners has opened the door to new possibilities for acquiring very large images (whole slide imaging), multiplexing stainings, exhaustive extraction of visual information and large scale annotations. This thesis proposes a set of algorithmic methods aimed at facilitating and optimizing these different aspects. First, we propose a multi-scale registration method of multi-labeled histological images based on the properties of B-splines to model, in a continuous way, a discrete image. We then propose new approaches to perform morphological analysis on weakly simple polygons generalized by straight-line graphs. They are based on the formalism of straight skeletons (an approximation of curved skeletons defined by straight segments), built with the help of motorcycle graphs. This structure makes it possible to perform mathematical morphological operations on polygons. The precision of operations on noisy polygons is obtained by refining the construction

of straight skeletons. We also propose an algorithm for computing the medial axis from straight skeletons, showing it is possible to approximate the original polygonal shape. Finally, we explore weighted straight skeletons that allow directional morphological operations. These morphological analysis approaches provide consistent support for improving the segmentation of objects through contextual information and performing studies related to the spatial analysis of interactions between different structures of interest within the tissue. All the proposed algorithms are optimized to handle gigapixel images while assuring analysis reproducibility, in particular thanks to the creation of the Icytomine plugin, an interface between Icy and Cytomine.

Mots-clés (français) : Pathologie numérique, Traitement de larges images, Recalage d'images, Analyse morphologique et spatiale, outils CAD, Géométrie algorithmique, Squelettes droits

Keywords: Digital Pathology, Large image processing, Image registration, Spatial and morphological analysis, Computer Aided Diagnosis tools, Discrete Geometry, Straight skeletons

*Je dédie cette thèse
à mes parents Alba et Julio
qui m'ont donné leur support en tout moment.*

Acknowledgments

This thesis would not have been possible without the enormous help of my advisors Jean-Christophe Olivo-Marin, Laurent Wendling and Vannary Meas-Yedid. I thank you for letting me be part of this project, your continuous feedback had great value and allowed me to go through this phase of my life and professional career. I also thank you for your positive attitude and emotional support that helped me through the toughest moments. I thank Isabelle Debled-Rennesson and Marc Pierrot-Deseilligny for closely reviewing this thesis. Similarly, I thank Georges Stamon, Dominique Béréziat, Raphael Marée and Vannary Meas-Yedid for accepting to participate as my thesis jury. In addition, I thank Georges Stamon and Jean-Christophe Olivo-Marin for receiving me on their respective laboratories at the Université de Paris and at the Institut Pasteur. Everyone at the Bioimage analysis unit at Pasteur, I am really grateful with all the advice you have given me through these years, I could not have done it without you. Of course a big thank you for everyone at LIPADE in Paris Descartes for helping me through my doctoral studies (as well as my master's). Thanks to the tumor immunology group at the Oslo university hospital for kindly providing the images which we worked with in this project. Furthermore, I thank my friends and family for being there when I needed you the most, you really are the fuel of all my work. Finally, I thank Grace for her special support and patience through this process, you really motivate me to be the best of me.

Contents

Contents	v
List of Figures	vii
List of Tables	xi
1 Introduction	3
1.1 Histology and Whole slide imaging	3
1.2 Challenges and opportunities in automated whole-slide image analysis	5
1.3 Organization of this thesis	13
I An image analysis framework for large images	15
2 Large image analysis: When data size matters	17
2.1 Icy: A software for reproducible image analysis	18
2.2 Cytomine: A software for large image management	21
2.3 Icytomine: Integrating Icy and Cytomine	23
2.4 Case of use: Using Icytomine to automatize glomeruli detection validation	31
2.5 Conclusion	35
3 Large image registration	37
3.1 Registration principles	39
3.2 Registering multi-stained large images: BigBUnwarp	40
3.3 Method evaluation	48
3.4 Conclusion and perspective	53
4 Conclusions of Part I	55
II Vectorial shape morphology	57
5 Introduction: Describing shapes as vectorial data	59
5.1 Background	61

6	Introduction to straight skeletons	65
6.1	Notation	65
6.2	From continuous to discrete skeletons	66
6.3	Existing methods for building straight skeletons	68
7	Straight skeletons based on motorcycle graphs	71
7.1	Generalization of motorcycle graphs	71
7.2	Wavefront propagation using motorcycle graphs	74
7.3	Straight skeletons from wavefront propagations	76
7.4	Performance analysis	78
7.5	Straight skeleton precision limits	79
8	Refined straight skeletons	81
8.1	Refining motorcycle graphs	81
8.2	Wavefront propagation using refined motorcycle graphs	85
8.3	Implementation considerations	85
9	Morphological operations on straight skeletons	91
9.1	Shape dilation	91
9.2	Shape erosion	93
9.3	Shape openings and closings	95
9.4	Straight Medial Axis	96
9.5	Evaluation	101
10	Shape reconstruction from Straight Medial Axes	103
10.1	Reconstructing a shape from a medial axis	103
10.2	Straight medial axis evaluation by reconstruction	105
10.3	Understanding parameters of the straight medial axis	106
11	Exploration on weighted straight skeletons	111
11.1	Introduction and State of the art	111
11.2	Ambiguities and implementation considerations	112
11.3	Current development status	115
11.4	Application opportunities on histological images	115
11.5	Conclusion	116
12	Conclusions on vectorial shape morphology	119
	Conclusions	121

List of Figures

1.1	Histology process diagram	4
1.2	Example of slide scanner (Axio Scan Z1), which is capable of load up to 100 slides to scan in a single run.	5
1.3	The different steps of an image acquisition in histology. Image adapted from [2]	6
1.4	A view of the different processes taking place in an automated histology analysis	7
1.5	Example of stain normalization	8
2.1	Icy main user interface	20
2.2	Icy protocols	20
2.3	Cytomine user interface	22
2.4	The workflow of Icytomine and the image analysis possibilities	24
2.5	Icytomine architecture diagram	25
2.6	Icytomine explorer	26
2.7	Icytomine viewer	26
2.8	Downloading views with Icytomine	27
2.9	Sending annotations to Cytomine from Icy	28
2.10	Examples of the two classes stained in Masson's Trichrome	31
2.11	Batch processing with Icytomine	33
2.12	Examples of glomerulus CNN classification	34
2.13	Examples of Mask R-CNN segmentation results	35
3.1	Image registration challenge on histological images	38
3.2	Tissue segmentation methods comparison	42
3.3	Histological tissue segmentation results	43
3.4	Examples of deformation vector field of divergent (left) and rotational (right) types.	45
3.5	Image comparison before and after registration	46
3.6	Image registration process	47
3.7	Applying the transformation model to large input images	47
3.8	Different staining techniques were used for the ANHIR challenge	49
3.9	Successful segmentation examples of tissue on virtual slides from ANHIR registration challenge.	50

LIST OF FIGURES

3.10	Unsuccessful segmentation of tissue on virtual slides from ANHIR registration challenge.	51
3.11	Successful registration examples on virtual slides from ANHIR registration challenge.	51
3.12	Unsuccessful registration examples on virtual slides from ANHIR registration challenge	52
3.13	Histogram of the Median-rTRE measured on the training data . . .	53
3.14	Successful registration example on other imaging technologies . . .	54
5.1	Comparison between raster-based and vector-based annotations . .	60
6.1	Wavefront propagation on simple polygons	67
7.1	Split events on motorcycles when propagating the wavefront	72
7.2	Examples of motorcycle graph on simple polygons	73
7.3	Simultaneous motorcycle crash handling.	74
7.4	Representation of the extended wavefront of a polygon	75
8.1	Straight skeleton compared to the continuous center line of a noisy polygon	82
8.2	Approximating a straight line wavefront to a continuous one	82
8.3	Motorcycle creation on reflex vertices at different angles with a maximum amount of motorcycles set to 4	84
8.4	Comparison of refined skeleton detail against computation time. . .	87
8.5	Comparison of refined skeleton number of input vertices against computation time.	88
9.1	Shape dilation using straight skeletons	92
9.2	Example application of dilations on histology	93
9.3	Shape erosion using straight skeletons	94
9.4	Using straight skeleton morphological operations on histological images	95
9.5	Example of shape openings and closings	95
9.6	Example of polygon smoothing by applying a sequence of morphological operations	97
9.7	Medial axis applied on the contour of a uterus to assess its behaviour with respect to the thickness of the sample	98
9.8	Example of medial axis computation	98
9.9	Example of resulting medial axis on shape taken from [174]	101
10.1	Single edge medial axis shape reconstruction	104
10.2	Multi-edge medial axis shape reconstruction	105
10.3	Normalized Hausdorff distance of the medial axis reconstruction against the original shape	106
10.4	Example shape used to explore and understand medial axis parameters.	107

10.5 Varying θ_1 parameter on the medial axis	107
10.6 Varying θ_2 parameter on the medial axis	108
10.7 Varying θ_3 parameter on the medial axis	108
11.1 Possible outcomes when edge collapses and incident edges are col- linear and with different weights	113
11.2 Ambiguity when defining vertex direction from colinear incident edges differently weighted.	114
11.3 Before, during and after a multiple split event on the wavefront . .	115
11.4 Example of a weighted skeletons	116
11.5 Applying weighted dilations to match directed complementing objects.	117

LIST OF FIGURES

List of Tables

2.1	Time to open an image for the first and re-load time	27
2.2	Classification accuracy	33
7.1	Skeletonization methods computation time comparison. This comparison was made using an binary image at multiple resolution levels, extracting a polygon from the contour of the shape at each resolution level. Tested methods were Vincent [161], Guo [162], Zhang [163], and our implementation of straight skeletons.	79
8.1	Error of the continuous wavefront approximation by straight lines at a terminal vertex	83
8.2	Comparative performance over maximum number of initial wavefront vertices (skeleton detail) on reflex input vertices.	86
8.3	Evaluating the refined straight skeleton	89
11.1	Comparison of properties in weighted and unweighted straight skeleton.	112

Introduction

Chapter 1

Introduction

In histology, the tissue of the subject (either a plant, animal or human) is inspected with a microscope for studying the biology of processes such as cancer or embryogenesis. Tissue processing has been highly automatized in recent years, reducing considerably the speed of production of slides available to visualize. Alongside, digitization of slides (acquisition of digital images of histological tissue), also known as **whole-slide imaging** (WSI), has allowed to visualize them on computers instead of using traditional microscopes. However, even with these improvements on histology imaging, there is still a lot of procedures that remain manual, requiring highly trained pathologists and raising costs of histology analyses [1]. As a consequence, there is a window of opportunity for developing automated methods that help reduce the time taken for performing these procedures by letting computers ease these tasks, and even making available other procedures that are prohibitively time-consuming for humans. In this chapter, we take a quick look at the procedures involved in histology analysis. I present procedures from tissue preparation until the different challenges that currently exist on computational pathology. Alongside this description, I present the specific research subjects involved with this thesis that aim to improve automated histology analysis. Finally, I describe how subjects in this thesis are organized.

1.1 Histology and Whole slide imaging

The main purpose of histology for pathologists is to examine tissue and provide an accurate diagnosis that will finally become a therapeutic intervention for the patient. This intervention can vary from no further action (in the case of unremarkable diagnosis), to close follow-up, medical treatment, or even surgical intervention (in the case of malignant diagnosis).

In order to provide a diagnosis practitioners follow several procedures to collect, treat and store the tissue to be analyzed (see Figure 1.1). Understanding this process helps to have a better perspective of how digital images are obtained out of tissue and the different sources of variability of the information during image acquisition.



Figure 1.1 – Histology process diagram

1.1.1 Tissue preparation

Sample acquisition

In general a clinical histology begins with the determination of the treating physician indicating that, in order to continue treatment, the patient has to go through histological confirmation. Following this, the physician has to obtain enough tissue to be able to provide a diagnosis. To **collect** enough tissue there are multiple standard protocols including fine-needle aspiration, needle biopsy, excisional biopsy, and excision of the entire lesion tissue. The sensitivity and specificity of these procedures increases as the obtained tissue increases, providing pathologists with more cellular context to provide an accurate diagnosis. After biopsy, the pathologist measures the tissue and records a detailed description of its color and characteristics. In the case of large amounts of extracted tissue, it is split into cassettes (approximately 10 by 10 by 3 millimeters) used for subsequent steps.

The extracted tissue is then **processed** to stabilize it chemically, as well as physically, to keep its microscopic morphology. During this step the tissue is immersed into a fixative solution that stop cell architecture from breaking down and stop further growth of microorganisms. Next, the tissue is physically stabilized be either freeze drying, microwave, or chemical stabilization.

The tissue processing is followed by **embedding** it on a fixated block, where the tissue gets impregnated with and surrounded by liquid paraffin which gets hardened as it gets cold. During this step the tissue has to be correctly oriented in order to obtain correct slices when it gets sectioned in thin slices.

Sectioning is the step in which tissue get cut in thin slices that are then mounted on microscope slides. This is done by using a microtome that slices the tissue on the fixated block in sections of 3 to 4 micrometers of thickness. Following this, slides pass by a *floating* of cold water in order to disperse wrinkles product of the sectioning. Then, sections are positioned on glass slides of size 25 by 75 by 1 millimeters.

At this point, tissue slices are so thin that they are now practically invisible under a light microscope. To make the tissue visible it has to go through a **staining** procedure, where immunohistochemical staining (IHS) or chemical dyeing is applied to the tissue. These dyes bind to certain components of cells and extracellular components producing a visual appearance that can be visible using a microscope. The most widely known stains for diagnostic histology are hematoxylin and eosin, which have good properties in terms of contrast between the background and the different components of the tissue, producing colors at opposite ends of the visual spectrum. Other dyes are also used according to the needs

1.2. CHALLENGES AND OPPORTUNITIES IN AUTOMATED WHOLE-SLIDE IMAGE ANALYSIS

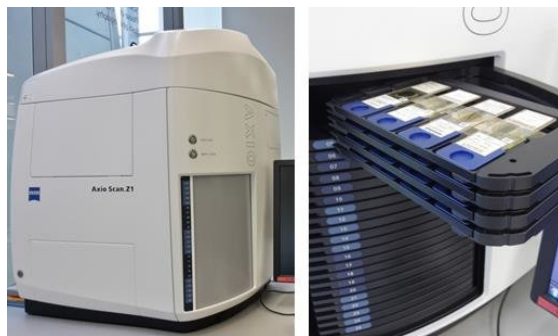


Figure 1.2 – Example of slide scanner (Axio Scan Z1), which is capable of load up to 100 slides to scan in a single run.

of the pathologists to stain specific types of cells on the tissue sample. Examples of dyes comprise Clara cell 10 protein (CC10), platelet endothelial cell adhesion molecule (CD31), prosurfactant protein C (proSPC), among several other (Some examples in Figure 3.8).

Sample visualization

With the slides already stained, they must be **digitized** in order to have a digital version of the sample and to be able to visualize the entire slide on a computer screen for further analysis. For this, microscope manufacturers provide *whole-slide scanners* able to capture high-resolution images at high speeds (See Figure 1.2). These scanners usually offer resolutions using a 20x or 40x objective reaching 0.23 to 0.25 micrometers per pixel. The image is then stored on the computer hard drive and can be opened for **visualization** anytime and as many times as needed once the slide is digitized. However, virtual slides use considerably more space than regular pictures as they hold detailed information about the tissue, contrary to regular pictures that can be compressed with a certain loss in quality. Despite this price in disk space, it offers an interesting alternative to storing the physical sample, because hard drives are relatively small and can hold multiple virtual slides. Additionally, virtual slides do not suffer from sample aging. Nowadays, despite the increasing usage of slide scanners, physical sample blocks are still kept for a period of at least 10 years. A full overview of the histology acquisition process can be seen in Figure 1.3.

1.2 Challenges and opportunities in automated whole-slide image analysis

In practice, although slide scanners are increasingly being used, many pathologists still prefer to use classical microscopes because they offer faster panning, faster focusing and an intangible sense of control of the slide. In addition, histology diagnosis has three main sources of variability [1]:

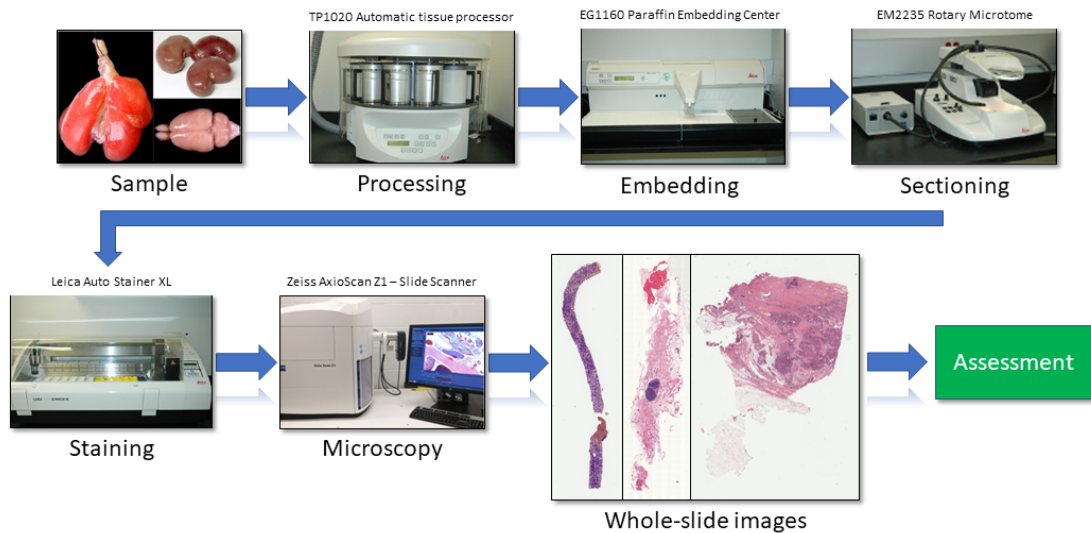


Figure 1.3 – The different steps of an image acquisition in histology. Image adapted from [2]

- **Biological:** Associated to the variations experimented among different people and the different behaviour of biological processes for each individual. This effect can generate different results on images of the same tissue of different patients.
- **Technical:** When the preparation of the tissue does not follow the same processing protocol. This can result in different results on the subsequent analyses. In some, the simple fact of using a different slide scanner of the same model and the same specification can introduce variations on the acquired images.
- **Inter-observer:** In this case the assessment by multiple pathologists (at least 2 and in some cases an odd number of specialists are requested) of the same sample can yield to different results, introducing an evaluation subjectivity.

In order to automate the analysis performed on virtual slides the previous sources of variability have to be dealt with to achieve accurate diagnoses and prognoses.

It is precisely at this point where developers of digital solutions are making efforts to improve the user experience of visualization software providing user-friendly tools. Researches are also seeking to help pathologists by providing new information extracted automatically from acquired images. Some of the research topics and challenges aiming to perform computational pathology are described in the following paragraphs (from 1.2.1 to 1.2.8). Most of these elements are taken from four main reviews in histopathological image analysis, as well as complementary articles. In [1] authors present a general perspective of all the steps involved in a histology diagnosis process organized in sample acquisition and tissue visualization. [3] introduces pre-processing, segmentation and feature extraction on

1.2. CHALLENGES AND OPPORTUNITIES IN AUTOMATED WHOLE-SLIDE IMAGE ANALYSIS

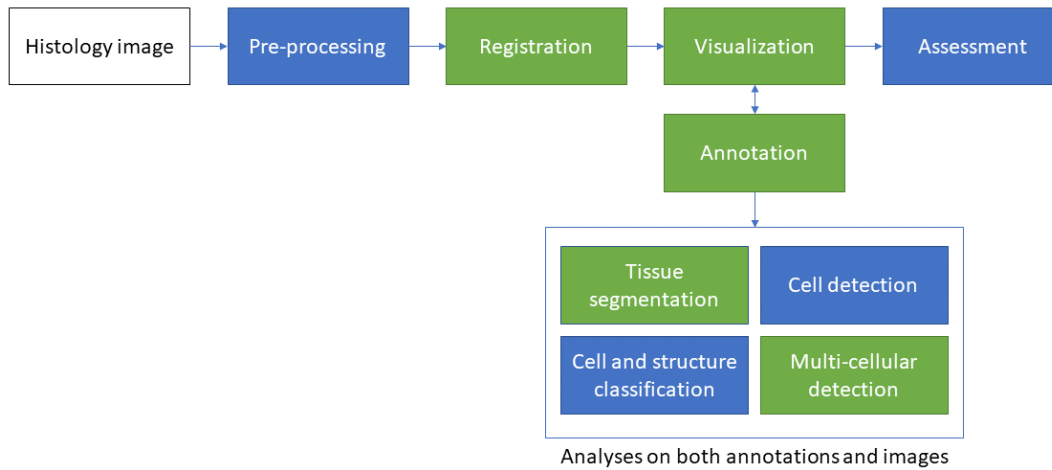


Figure 1.4 – A view of the different processes taking place in an automated histology analysis. This thesis research work is associated to some of these processes, which are colored in **green**.

histological images. The review in [4] concentrates on computation pathology, a broad term including nucleus detection, cell classification and survival statistics generation. Finally, [5] deals with acquisition quality control, feature extraction, region of interest detection and data visualization. A general view of the automated histology analysis workflow can be seen in Figure 1.4, where several processes take place in order to provide diagnoses from histology images.

1.2.1 Pre-processing

As with microscopy imaging, histology images present different types of artifacts and noise, some related to general microscopy and some specific to whole-slide imaging. Since these artifacts on the image can alter their analysis, they have to be removed before any image analysis is performed.

Stain normalization

As mentioned before variability on captured images can alter analysis results. One possible source of variability is the staining variations caused by tissue preparation or digitization conditions, which may affect the appearance of tissue on virtual slides. To solve this, the intensities of the pixels must be normalized with respect to a reference intensity spectrum, which is taken from another reference image. In general two images are (the reference image and the image to normalize), and the images are adjusted to fall on the same intensity spectrum of the reference image (See Figure 1.5). The simplest approach is to use color normalization techniques usually used in photography (histogram equalization for example), which can be automatized to handle specific types of staining techniques [6, 7]. This approach

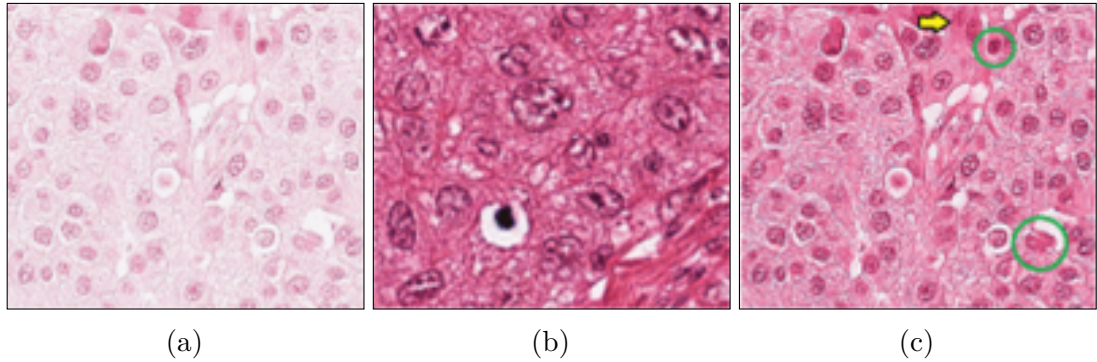


Figure 1.5 – Example of stain normalization. These images stained using the same protocol (hematoxylin and eosin) have been acquired with two different scanners (Aperio and Hammamatsu), introducing a variability on the intensities of acquired images. Here, the source image 1.5a is modified in order to adjust its color intensities with respect to the target image 1.5b, obtaining intensities on the same color spectrum (1.5c) [13].

has been successfully applied on serial sections that have been stained with the same dyes [8]. Another way to achieve stain normalization is by performing a color deconvolution, in which the colors of the dyes in the image are separated, normalized and finally recombined. This approach is useful when one of the dyes has a low contrast with respect to the others. Furthermore, color deconvolution can be performed when a single dye needs to be extracted for further analysis, to detect cell nuclei for example [9]. One of the main issues with this approach is that, due to lack of staining protocol standardization, the color profile of the dyes must be learned before deconvolving the images [10], which works fairly well on classic staining techniques such as hematoxylin and eosin (H&E). However, when the color profiles have not been learned, the process is more challenging because it has to be estimated. For this, non-negative matrix factorization [11] or expectation-maximization [12] is used to retrieve to stain standard colors to find clusters in chromaticity space corresponding to each stain color. A lot of research is currently being done to improve results of these procedures, specially on slides that have been damaged by slide aging or tissue-folding. This process is crucial for the rest of analyses done on histology images, specially on registration and classification methods because they work by linking color intensities on images.

Tile and tissue stitching

Since the objective of the microscope camera has a limited field of view of the tissue, multiple slightly overlapping images are captured (called tiles). Then, these tiles are put together accordingly to create a high-resolution whole-slide image (hence the name *whole-slide scanner*). Approaches like [14] provide a user interface to select corresponding points on the images to be stitched. This is ideal when the imaged areas are too big or if they are selected manually. However, most

slide scanners take care of selecting the grid of images to capture and their stitching, achieving adequate results. As result of this process, images can achieve an important size in pixel count, and consequently in file size. This becomes an important issue because images cannot be processed and stored entirely in memory. Methods had to adapt in order to handle these images, either by down-sampling images (reducing image quality) or by working by patches (which introduces a loss of global knowledge). This issue will be covered in this thesis when developing a integrated framework to analyze whole slide images in Chapter 2. In addition, handling large images is also presented in Chapter 3 of this thesis when storing transformed image after computing the transformation model that registers large histology images.

1.2.2 Registration

When more than one virtual slide is captured from the same tissue (creating a 3D stack of slides), they usually present a misalignment due to the tissue staining and fixation on the glass. Hence, the need for registering these images in order to analyze these images as a collection of the same tissue block. In addition, an important challenge of registering histology images is that they are usually large once they have been stitched (going up to 500k by 500k pixels in size), and the file size per image can attain several Gigabytes. Most of the methods associated to images registration have their beginnings on computer vision systems, where data collected from different sources has to be aligned on a common space to create a richer representation of a certain area (e.g. [15, 16]). There exist multiple ways to register histological images according to the acquisition protocol used on them. If the contiguous slides are stained using the same dyes, then classical image registration techniques are applied using either rigid or non-rigid registration techniques. If, on the other hand, slides are stained using different dyes or have been acquired using different imaging techniques (e.g. registering types H&E with SHG - second harmonic generation), multi-modal registration techniques have to be applied in order to correlate the parts of the tissue across the two staining protocols/imaging technique. On this subject, Chapter 3 presents a non-rigid registration technique for multi-stained histology images based on a pre-segmentation of the tissue in the image and a multi-resolution level deformation estimation. All this is handled taking into account that these images have an important size in pixel count and in file size.

One issue that has no known solution to this day on digital histology is tissue folding. This happens when the tissue is fixated on the glass but some wrinkles remain on the floated section, when the cover glass is put in place this wrinkle is flattened creating a fold on the tissue that becomes evident (by intensity saturation) when it is digitized. On the other hand, detection of these areas has been addressed by analyzing color saturation [17, 5]

1.2.3 Visualization and annotation

One of the main problems related to whole-slide images is their size, reaching up to several gigabytes per image. This poses an important limitation in terms of processing because most times the image can not be entirely held in RAM memory and limits the amount of analysis that can be done at once. Additionally, proprietary scanner manufacturers have come up with their own image formats adjusted to their needs, raising further issues in terms of compatibility. In research, solutions like *OpenSlide*) [18] have adapted readers for most popular formats and have adopted an open format called DZI (Deep zoom image. Other initiative to provide cross compatible formats is *Bioformats* [19], which provides readers for most proprietary formats and uses its OME-TIFF as its open standard format allowing not only to store large images in a resolution pyramid, but also allows to add any kind of additional metadata to the images. More recently, some researchers developed real-time viewers that allow access to images through an Internet connection [20] (e.g. Cytomine, ImageScope, Seden Viewer, etc). Some of the viewers previously mentioned have also allowed users to add annotations on the images to identify specific structures on them such as cancerous tissue or cell clusters. This is crucial on the image analysis workflow since this makes it possible for experts to create ground truth maps that can help construct classification models. For example, to detect cancerous versus healthy tissue by creating a collaborative annotation features.

Taking into account that virtual slides not only hold the scanned tissue, but also a large background area, it is important to identify and provide only relevant areas of the image for the pathologist to provide an accurate diagnosis. Some approaches take a subsampled version of the image and identify *regions of interest* (ROIs) based on the colors present on the image [21]. Usually, these regions are classified using a combination of decomposing the image in small low-resolution patches and the usage of a support vector machine to classify and filter useful patches [22]. Another way of creating regions of interest on the images is to search the image at multiple resolution levels, starting from the lowest resolution and refining ROIs as higher resolution images are used. In this case the regions of interest can be identified using color clustering, refining the clusters when increasing the resolution level [23]. The output of this procedure allows pathologist to save time and concentrate only on useful areas of the image, allowing for faster analyses.

Although several large image visualization tools have surfaced in recent years, very few tools to analyze these large images have been developed. In Chapter 2 we address this subject and provide tools (Icytomine) to interconnect visualization tools (Cytomine [20]) with image analysis tools (Icy [24]) to expand possibilities for analysis automation (Chapter 2). This requirement for an analysis framework is important because the rest of the challenges and research opportunities presented in this chapter involve high interaction between pathologists, visualization tools and analysis software.

1.2.4 Cell detection

One of the most attractive promises of digital pathology, if not the most attractive, is the possibility of automated cell identification. This can be a very challenging task since virtual slides contain an enormous amount of cells, and many times they do not conserve the same properties across the image (cell shape, size, presence of nucleus, etc.). Most works in this domain have opted to base the detection on nuclei detection, which can itself be a complicated task since nuclei can be tightly clustered and also vary in size ($\approx 5\mu m$) and shape depending on their cell type. Most basic approaches use color clustering (by expectation-maximization for example) to identify different types of elements in the image and take the cluster for cell nuclei to identify nuclei areas [25, 26]. After this step, nuclei segmentation shape can be refined by using active-contours [21, 26] or graph-cut based on color and intensity difference features [7]. Other methods use heuristics to separate clustered nuclei, such as curvature-based reasoning [27] or concavity [26]. These methods offer very accurate results that allow further analysis of cell behavior and reach accurate diagnosis. The challenge in this area is the validation given that a large hand annotated datasets are needed to compare these methods and provide a better insight on their performance.

1.2.5 Cell and structure classification

Knowing the type of cells and specific structures present on tissue samples is of great value for diagnosis. For example, [28] uses classification to differentiate centroblasts from normal cells in order to grade cancer. The authors use Fourier-based textures as classifier. Other works focus on mitotic cell detection as the challenge in [29]. The best known method as of the writing of this document is [30], where deep convolutional network AlexNet and UNet architectures on color normalized images are used in combination with a blue ratio measure to identify potential mitosis, achieving results with up to 97% in accuracy.

1.2.6 Multi-cellular structure detection

Cells express themselves, in general, as part of organized groups when looking at histology images (e.g. glands, acini). Detecting and understanding their behaviour can be of great help. Due to the high-variability of cells across types of multi-cellular structures these tasks are usually adapted to specific needs. In [31] graphs are created on top of colon tissue images aided by user annotations. Matching these graphs to reference ones allows to classify glands as healthy or diseased according to their structural configuration. Other methods aim to segment multi-cellular structures by treating them as if they had a nucleus. For example, [32] detects lymphoid follicles by segmenting groups of lymphoid cells, using active contours and splitting results based on curvature and further color filtering to remove false positives.

1.2.7 Tissue segmentation and morphological analysis

In addition to identify multi-cellular structures, segmenting different tissues in histological images provides context to subsequent analyses. However, the variety of tissues and their appearance makes this task very challenging. In [33] local pixel intensities are used as features to discriminate bone, cartilage and fat tissue in teratoma tumor images. Furthermore, in [34] local color histograms are used instead of edge-based features to detect different types of tissue. Some authors concentrate on object homogeneity to detect cells and crypts, instead of just relying on pixel homogeneity [35]. Another group achieved cancer region segmentation by graph-based classification where features were taken from node co-occurrence frequencies [36]. A recent method focuses on interactive polygonal segmentation of images from hand-picked square ROIs [37]. Their approach uses recurrent neural networks to automatically refine segmentation with each interaction with the user. Another approach is to perform spatial analysis on the different types of tissue found on the acquired sample. For example, in [38] authors aim to classify tumor cell, normal cell and lymphocyte regions on histological images based on their cytological profile. They apply spatial point statistics on cells to retrieve the profile at different locations in the tissue. Performing profile analysis and providing spatial statistics can provide improvements on subsequent diagnosis of the tissue.

In this thesis, we take advantage of the way annotations are stored in most systems (polygonal and vectorial shapes) to make this kind of analysis faster and as accurate as possible. I dedicate Part II to present tools to work with vectorial ROIs to provide morphological descriptors and extend the knowledge these ROIs can provide to image analysis. Examples of these tools are straight medial axis (Section 9.4), iso-contours through annotation erosion and dilation(Sections 9.1 and 9.2) from which tissue can be assessed.

1.2.8 Assessment

The last challenge in digital histopathology is the assessment (either diagnosis or prognosis), where an input histology image is classified to describe it as containing malign or benign tumor tissue. Assessment can be achieved by grading the tissue on images to evaluate the severity of the disease. Most methods tend to extract a set of features to classify them afterwards. In [39] authors use Fourier shape descriptors (nuclei, cytoplasm and unstained) to classify benign or renal tumors on hand-selected ROIs, distinguishing three different types of renal tumor, by using SVM classifiers. Authors in [40] use generic features (local binary patterns, co-occurrence matrix statistics, curvelet coefficient statistics) to classify tissue as normal, in situ cancer and invasive cancer. They claim to achieve 99% of accuracy by applying a two stage classification involving a series of SVM classifiers, followed by a set of neural networks. Their method also has the feature of rejecting ROIs (with a rate of 1.94%) that result too difficult to classify so that humans can assess it. Examples of grading can be seen in [21] where breast cancer is graded according to the size, and texture of the nuclei inside detected ROIs using a Bayesian clas-

sifier. Other approaches such as [41] use morphology to find blobs that are then classified using different pixel features. In addition, in [42] random forests are used to classify prostate cancer taking into account color and co-occurrence features. More recently, this task has benefited of the high advance of deep neural network approaches, helping to improve the model learning to classify cancer stages [43].

1.3 Organization of this thesis

As seen in the previous section, automatizing digital pathology is a laborious task and involves research in multiple domains besides simple image processing, such as graph theory or automatic classification methods, among many other. In this thesis I aim to provide solutions that allow for large image analysis in histopathology, bringing all advances on whole slide imaging into the world of image analysis.

Two parts divide this document; First, in Part I, I concentrate my efforts to develop a base framework that enables automatizing histopathology analysis by integrating remote large image storage with local image analysis software. First, in Chapter 2 a large image analysis framework is introduced to provide both fast access to remote images (stored in Cytomine), and accessibility to powerful image analysis methods already available on our local software solution (Icy). Here, large image processing can be achieved at multiple resolution levels and with the possibility to perform patch processing on images. Second, a pre-processing method is presented to register large multi-stained histological images in Chapter 3. The second part starts by remarking that most annotations stored on histology systems are saved in vectorial format, aiming to save space on disk and to save time on analysis. However, some of the classic methods to perform morphology on images become prohibitive in time with large images. For this, an effort was made to facilitate morphology analysis on large images, taking advantage of the vectorial nature of annotations and proposing a set of tools to alleviate this issue. First, some notions are presented on morphology and skeletonization of shapes (Chapter 6). Following this subject, in Chapter 8 refined straight skeletons are introduced in order to approximate continuous skeletons. With these structures it is then possible to introduce a set of interesting morphology operators that can be used to perform shape analysis, as well as shape smoothing (Chapter 9). Additionally, a simplified version of the straight skeleton is presented in this chapter with the name of straight medial axis, which keeps the essential morphological information of shapes using less data. Some applications are presented for each method presented in this part with their corresponding evaluation. Furthermore, as an exploratory work, some insights are discussed on weighted straight skeletons, which allow dilations and erosions to be redirected into specific angles to help improve tissue segmentation (Chapter 11). Finally, in Chapter 12 some conclusions on the work done during my doctoral studies are presented alongside with some perspective for future possible developments.

Part I

An image analysis framework for large images

Chapter 2

Large image analysis: When data size matters

One of the most noticeable features when new optic techniques are developed is the size of acquired images. In general when a new generation of microscope becomes public, the size of the images is increased by at least a factor of two. This increase in size can be reflected in two different ways:

- **The spatial resolution:** The pixel size becomes smaller. Thus, the pixel count of the image of the same sample increases exponentially.
- **The field of view:** The extent of the image acquired covers a larger area. In this case, the pixel size is not changed, but the image size is larger because more elements are included in the acquired image.

This is excellent for image analysis as more details become visible on the image, providing more information in terms of sample context or sample resolution, that can be used to improve image analysis processes. However, increasing the amount of information on images comes with some complications:

- Processing images becomes slower as more information has to go through calculations involved in the method. This can be translated into more delays to obtain results. Some methods may even become time-prohibitive due to the amount of processing needed.
- Storage becomes a major constraint for these images as data stores have to allow for multiple image storage. Transferring these images between different storage medias can also take a long time due to their size. Furthermore, processing larger images means that more space is used in RAM memory, limiting some methods to the maximum amount of data machines can handle at once.

In addition to handling large images, it is also important for most users to be able to identify structures on these images, hence the need for annotations. In histology analysis this is crucial because many of the analyses performed on the

images are concentrated on specific regions of interest (ROIs). Following the same rationale of large images, annotations must be treated with special care when using large images in order to avoid performance issues. It is common to find the usage of bitmap masks to store annotations on most common image analysis software (ImageJ [44] or Icy [24] for example). However, in the case of large images this becomes impractical because bitmap masks end up taking a lot of space in disk when stored. Using bitmap masks also means that the whole image has to be traversed at least once to compute any statistic on the images, making this approach extremely expensive in time. Instead of that, large image processing systems take advantage of vectorial annotations, where only the border of the annotations is stored as a set of points, joined by straight lines, creating a light description of the annotation shape.

In this chapter, I concentrate on providing a framework that deals with the issues described above by connecting two technologies well known by their capabilities to analyze images and storing large images with their corresponding annotations. Starting with a description of the technologies (Icy and Cytomine [20]) used to develop our solution Icytomine. Then, I introduce our solution that enables image analysis on large histological images. I also present how our solution allows analysis automation by providing batch processing. In addition, we present a use case where Icytomine has been used successfully for automating validation of glomeruli segmentation. Finally, some perspectives are provided with some future developments for this framework.

2.1 Icy: A software for reproducible image analysis

Bioimage informatics makes use of the computation and mathematical sciences to extract comprehensive information from imaging data sets. This can be achieved by using multiple algorithms and methods developed in interconnected fields such as computer vision, signal and image processing, computation modeling, optics, biophysics and computer science. An immediate challenge arises when developing software coming from diverse fields:

- Multiple groups work on their own side for a common goal, and keeping the software quality is a challenge as members of each group integrates their work on an complete solution.
- The methods developed by one team can be useful for different problems out of the scope of the original solution developers. Making these methods reachable for other teams is a particular challenge as teams evolve on their goals and projects.
- Documentation of published methods is vital for understanding and improving them. Having access to all the related documentation allows to understand the preconditions and possible outcomes when trying to reproduce

experiments. In addition, this allows for method feedback and detection of possible performance improvements.

In order to achieve reproducible research, it becomes necessary to have access to all related information (code, data and parameters) used to produce the original publication. To answer this need, the bioimage analysis unit at the Institut Pasteur has developed a framework that allows users not only to publish their software in the form of plugin, but also it allows users to share the details behind the software and the means for users to test it the same way developers do when they first published their solutions. For this, Icy [24] provides means to develop, test and share research projects. In this section the main features of Icy are presented.

2.1.1 Main features

Although Icy has evolved over the years (it was first published in 2012), it has kept several core features almost unchanged since they hold together the rest of contents available for the platform. Some of them are presented here:

Graphical user interface

Icy has been developed to be as easy as possible to use so that final users can interact with all the parts of the software, including image viewers (2-D and 3-D), handling regions of interest, access to the image histogram and to the plugin access platform, which is linked to the online plugin repository (See Figure 2.1).

Protocols

In order to enable reproducible research for large amounts of data, it is necessary to have an environment that allows to apply repetitively the same procedure to each input element. For this, Icy provides an easy way to combine multiple plugins and perform batch procedures. The most common tasks are file batch processing and sequence batch processing, which take all the selected files or open image sequences, and apply a given task for each element. This feature can be very useful when evaluating procedures as it can be automated, allowing for reproducible testing. For example, when a new method has to be tested with multiple images to check its robustness, a protocol can be designed to apply the same method (with the same parameters) to all the images in the test image set. In addition, protocols can be saved to disk and shared with the community, which make them very useful to communicate the workflow of a specific method since protocols are presented as interconnected boxes in the editor provided with Icy (See Figure 2.2). Hence, reproducible research is achieved not only by allowing to repeat the same procedure multiple times with stable results, but also by being able to deploy the same application scenario in multiple locations.

CHAPTER 2. LARGE IMAGE ANALYSIS: WHEN DATA SIZE MATTERS

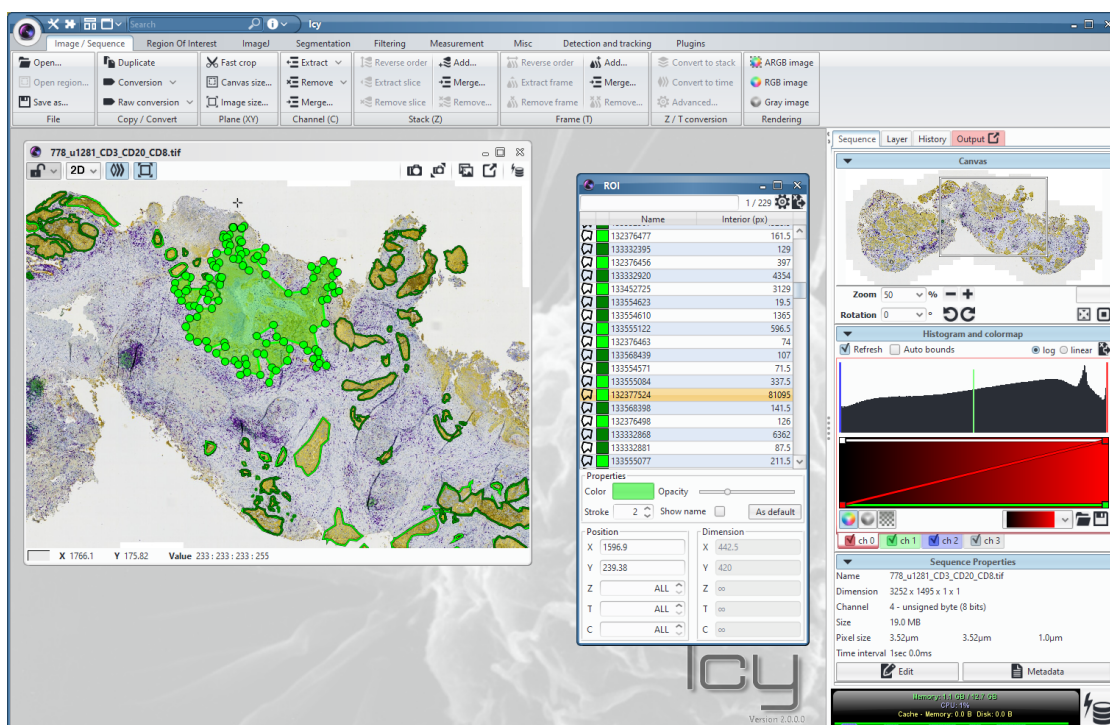


Figure 2.1 – Icy main user interface. A ribbon on top offers convenient access to most frequently used plugins. Each open image is presented on the workspace to the right of Icy. All details relative to the active image are presented on the right panel, which includes a listing of all the regions of interest (ROIs) associated to the image.

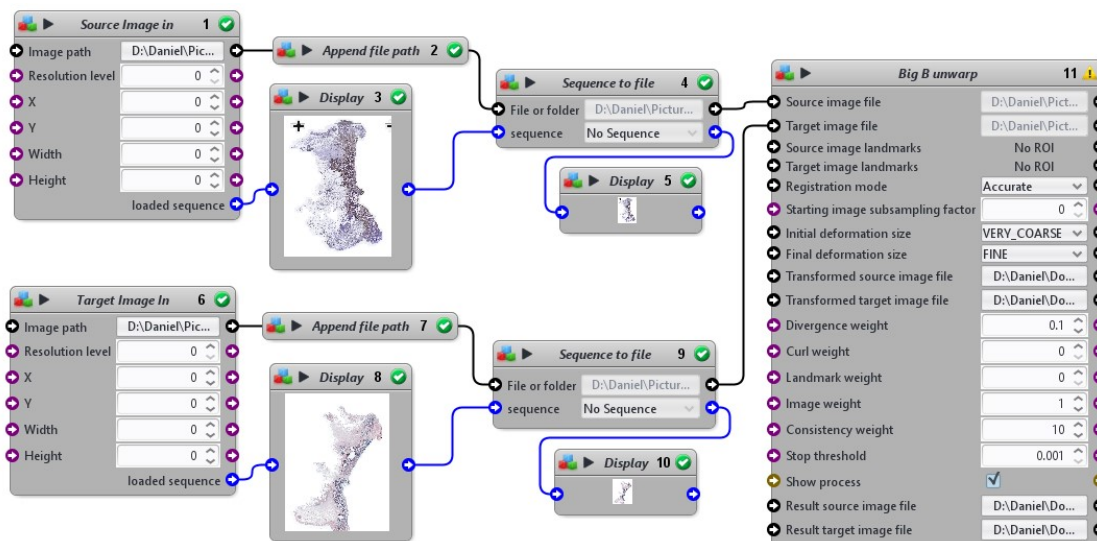


Figure 2.2 – Protocols allow to graphically program the different steps of an image analysis procedure. This image shows an example of protocol for registration of two images loaded from files.

Error report

Aiming to enhance the experience with Icy, developers have provided the error/bug reporting system. When users identify an unexpected behavior on the software, they can send a bug report directly to the plugin developers with a trace of the problems. This results in a collaborative debugging experience and makes plugins more robust by expanding the testing power of users.

Online community

One of the main goals of Icy is to make the results of research projects as available as possible, and for this an online community has been established since its creation. Developers can share their advances on their projects (a central plugin repository). And at the same users are giving feedback on available plugins. All this effort converges on useful plugins, with higher quality, and highly available for users.

A simple API for developers

In addition to the ability to read more than 100 file formats, Icy provides a comprehensible data structure that allows to handle images with a generic access to pixels in any format, to direct memory access to linear data buffers. The API is also conceived to handle parallel processing taking advantage of modern computer architectures. To accelerate the development process Icy provides a library (called *EzPlug*) that assists developers to create the graphical user interfaces of their plugins by automatically generating input fields corresponding to the plugin inputs. This also allows to expose plugins as protocol blocks, enabling graphical workflow programming.

2.1.2 Icy limitations

Although Icy provides a large collection of image analysis methods ready to be used out of the box, it is not capable of handling images that are larger than $2^{32} - 1$ pixels of size (approximately an planar image of 46300 by 46300 pixels). With this limitation, Icy becomes not very suitable for handling whole slide images coming from modern slide scanners. Later in this chapter Icytamine will be presented as a solution for this issue, allowing to interact between large image stores and Icy, providing image analysis on virtual slides while always keeping in mind local data size limits.

2.2 Cytomine: A software for large image management

In the field of digital pathology there is a strong need for efficient tools to create atlases, diagnose and collaborate on virtual slides. Although there have been several

CHAPTER 2. LARGE IMAGE ANALYSIS: WHEN DATA SIZE MATTERS

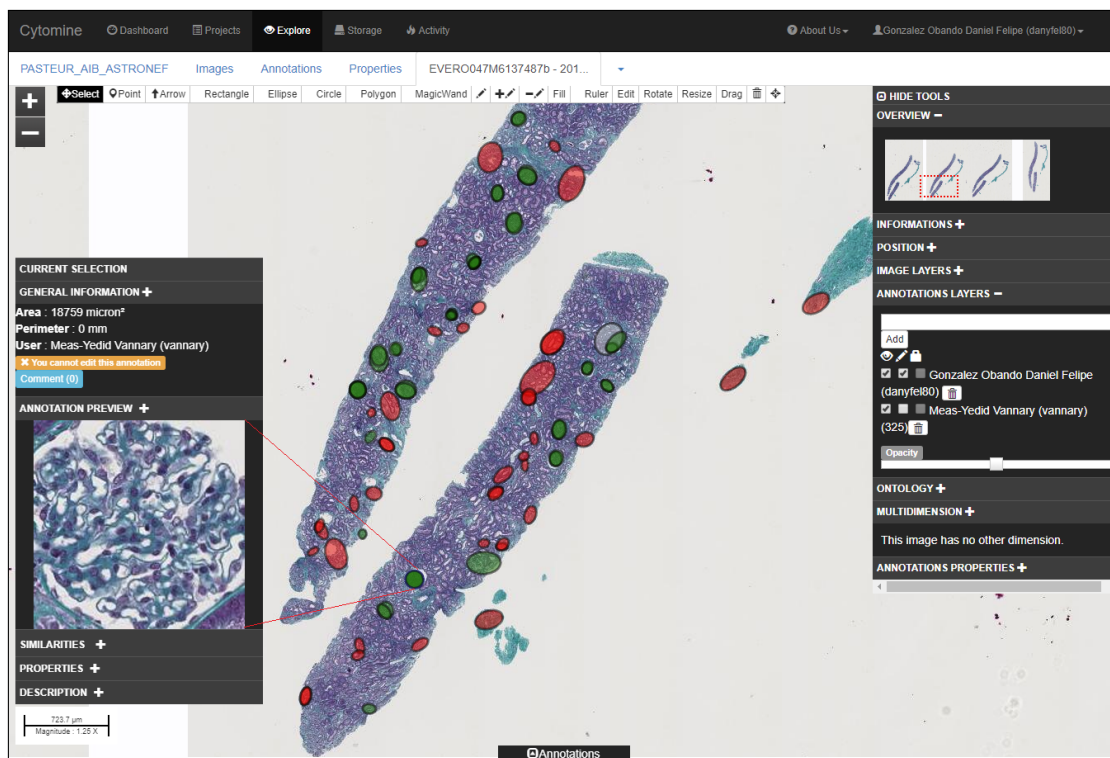


Figure 2.3 – Cytomine: A web application for whole slide image collaborative management

attempts to provide tools for virtual microscopy and tele-pathology [45, 46, 47, 48], they have not been developed to be fully collaborative or are vendor-dependant. As a consequence this limits the collaboration potential and the compatibility that web technologies provide in our days. With these issues in mind our collaborators from the Université de Liege developed Cytomine, a web application that is able to handle whole slide image storage, visualization, virtual slide annotation creation and review [20]. Their solution provides an easy way to organize virtual slides by projects. They also provide a real-time slide viewer that allows users not only to see the virtual slide at any desired resolution, but also annotations made by the image author or any of the project collaborators. Using specific ontologies, structures identified on the image can be easily annotated and classified, and then they can be used to compute different statistics about these structures (See Figure 2.3).

In order to organize images stored on the server, Cytomine uses the following scheme:

- **Users:** Every user has a user name with a specific role and privileges on the platform (e.g. basic users and administrators)
- **Projects:** Every image has to be associated to a project in order to be accessed by users. A project has an author and a set of administrators that handle the access to the project data. In addition, images associated to the same project use the same ontology for their annotations.

- **Images:** Each image is identified uniquely but can be associated to multiple projects. On each associated project, an image associates its own set of both annotations and allowed users.
- **Annotations:** Analogous to the regions of interest in Icy, annotations allow users to highlight an area of the image as a polygonal geometry. Every annotation is associated to an image and an author user and can have a particular label associated to it from the project ontology.
- **Ontologies:** These are the terms that can be associated to the different structures annotated on images of a given project. Although each project has a specific ontology associate, two projects can use the same ontology for their annotations.

In addition, Cytomine makes an effort on securing the access to images since these can be directly linked to real human patients. Users can only have access to projects and images that have been previously authorized by the system administrator or the project manager.

Furthermore, Cytomine conveniently allows users to easily find the content they are looking for by using search filters on each of the items previously described.

In order to access stored images efficiently at any resolution level, a resolution pyramid is first created when an image is uploaded. This pyramid is then stored and can be accessed through the provided REST API of Cytomine. User annotations also efficiently stored in the form of polygonal sets. This allows to handle thousands of annotations per image without incurring in great disk space consumption because only the coordinates of the control points of the polygons are saved in the repositories. Again, these annotations are available through the REST API of Cytomine.

2.2.1 Cytomine limitations

Despite being able to share virtual slide projects and allowing users to add annotations on images, Cytomine falls short when images need to be processed mostly due to the need for creating new image pyramids, which becomes expensive in terms of needed data space. Furthermore, automatizing annotation creation is not easy to achieve directly on Cytomine. Limitation which is circumvented with our solution presented in the next section.

2.3 Icytomine: Integrating Icy and Cytomine

Following the need to interact with large images to perform image processing tasks, and acknowledging the capabilities of Cytomine for image access and of Icy for image analysis, an initial implementation to integrate both technologies in [49]. From this idea and in collaboration with the authors, a new Icy plugin was developed to connect Cytomine with Icy easing data transfer and access related

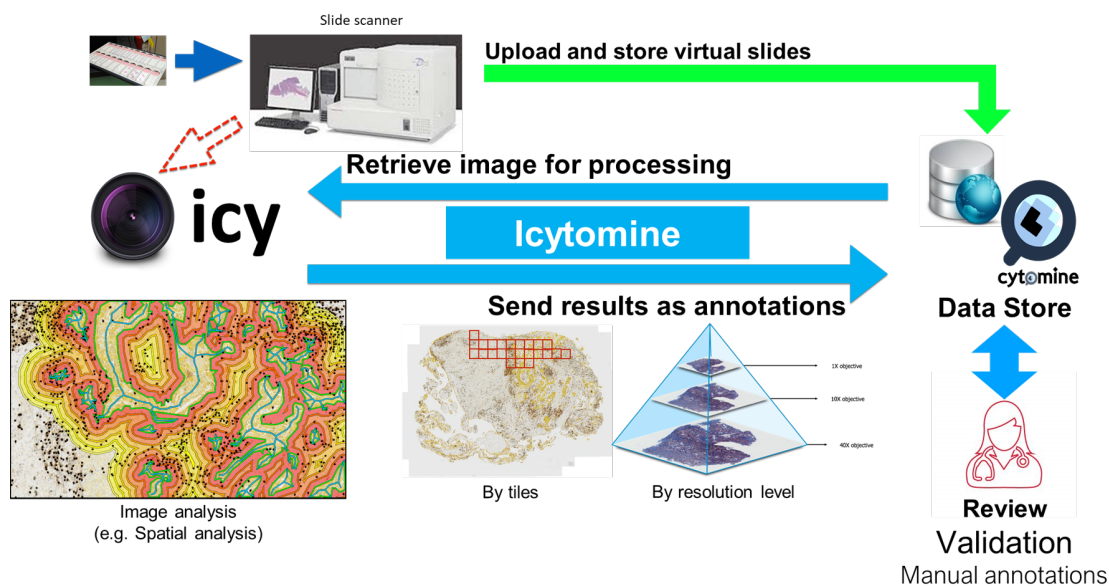


Figure 2.4 – The workflow of Icytomine and the image analysis possibilities, transferring images, analysis results, and manual annotations between Cytomine servers and Icy clients.

to large image analysis. Icytomine, our solution, has been developed with the aim of simplify the data transfer and provide a simple way to view data stored on Cytomine servers directly into Icy. The sections in this chapter will describe the software architecture behind Icytomine and show how this plugin helps image analysts to handle multiple large images and obtain relevant results in less time. A simplified schema of Icytomine showing the interactions made with Icy and Cytomine to handle large images and annotations is available in Figure 2.4.

2.3.1 Core manager library

In order to provide means to view large images stored on servers, as well as analyzing and annotating them, a core library has been developed to handle connections to Cytomine server instances and allows for optimized image and annotation transfers in both ways server-to-client and client-to-server. Furthermore, since Cytomine has been developed to hold sensible information about real human patients, security is an important issue. Our core library takes care of transmitting the right user credentials to the servers when they are needed and makes sure to provide access only to the features authorized by the service administrator. An image of the general architecture of Icytomine is presented in Figure 2.5.

The core library is divided in the following sub-packages:

- **Connection:** The code in this package is in charge of keeping connections and handling authentications on remote Cytomine servers, providing secure and stable access to server data.

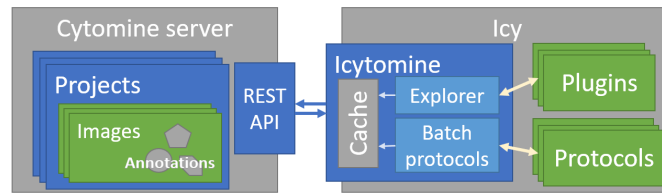


Figure 2.5 – Icytomine architecture diagram. The different parts of the Icytomine plugins are presented in this schema.

- **Image transfer:** Elements in this package handle image data transfer from remote Cytomine servers into local Icy clients.
- **Annotation transfer:** Files in this package allow the transfer between remote servers and Icy clients (both directions) of polygonal data and meta-data related to annotations associated to the images stored on the server.
- **Cache:** Since excessive data transfer can deteriorate the performance and fluidity of the viewer and other analysis on images stored on servers, it is necessary to have a cache memory that keeps local data that is constantly requested to avoid redundant server request that could impact the general plugin performance.

Taking advantage of the API provided by Cytomine, the core manager library has been implemented to allow other packages to have a fast, reliable and easy access to the data stored on the server as well as providing the possibility to send data to the server from Icy clients.

2.3.2 Project explorer and image viewer

With the core manager implemented, it is now possible to take advantage of the features in Cytomine applied to image analysis on Icy. The first feature in Icytomine is allowing users to explore projects and images available to the user on a given server. For this, an *easy-to-use* interface has been implemented: Icytomine explorer, with three columns enabling users to explore available projects (left), images(center), and provide users with details and previews of the selected elements (right). An example of the explorer is available in Figure 2.6. In addition to this interface, a simple connection interface is provided so that users can choose which credentials and which server they want to access.

Having enabled users to select projects and images, the second feature to be implemented is the possibility to visualize selected large images at different resolution levels without major delays (making strong usage of cache memory). In addition to presenting images to the user, the viewer is also able to show the annotations associated to the image being presented (also using cache memory to optimize server requests). Developing this viewer provides users with an interesting tool to search on the images for locations of interest and focus their analyses. A preview of the viewer can be seen on Figure 2.7.

CHAPTER 2. LARGE IMAGE ANALYSIS: WHEN DATA SIZE MATTERS

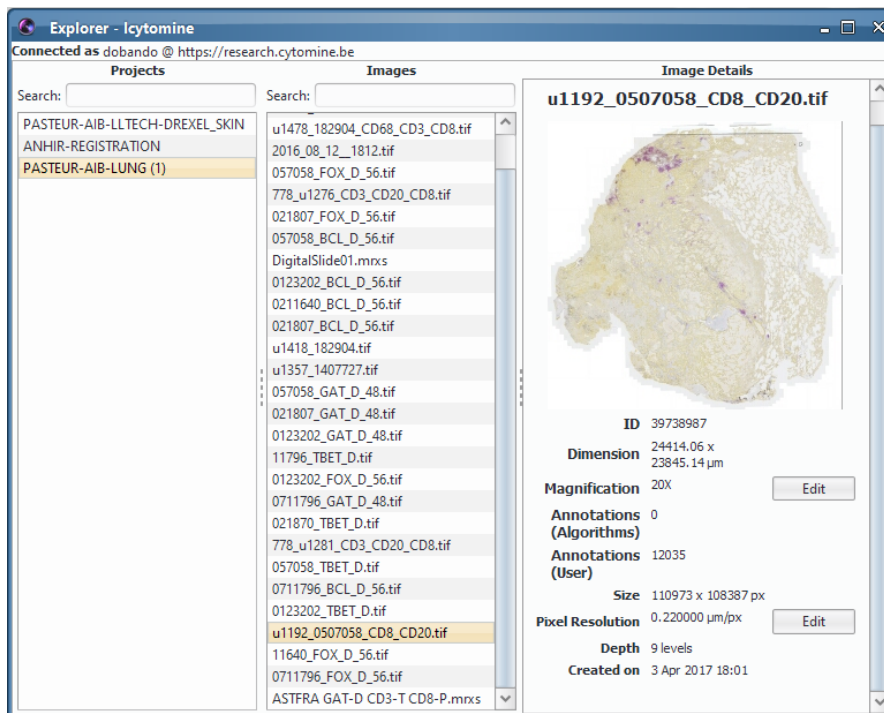


Figure 2.6 – The Icytomine explorer enables users to glance at the different projects and images available on a given Cytomine server directly from Icy.

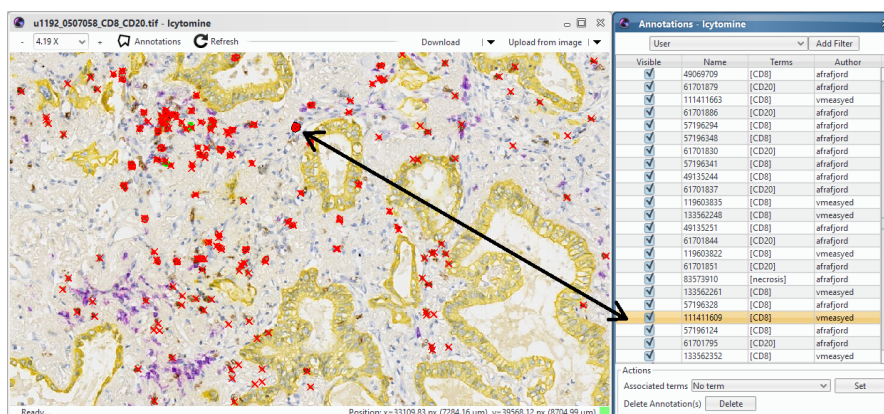


Figure 2.7 – The Icytomine viewer allows users to spot regions of interest or annotated areas on the image to perform further analysis.

2.3. ICYTOMINE: INTEGRATING ICY AND CYTOMINE

Image	Width (px)	Height (px)	Number of annotations	first load time/ reload time (msecs)
ima1	111,145	99,589	0	161 / <1
ima2	50,603	29,326	35	177 / 1
ima3	134,182	84,480	424	262 / 1
ima4	27,635	34,757	1,925	615 / 1
ima5	110,973	108,387	11,964	2,761 / 3

Table 2.1 – Time to open an image for the first and re-load time (in parentheses). Server was located at a distance of 300 km. When images are reloaded, cached images and annotations greatly reduces loading time.



Figure 2.8 – The Icytomine viewer allows users to download the current viewport and select the download resolution.

Table 2.1 presents the time it takes to open images and the time it take to re-load them once they are stored in cache. This evaluations were performed loading virtual slides of different sizes from a remote server located at a distance of 300 kilometers from the requesting machine under average request load conditions. As the table shows, times are much lower when images are cached. This is one of the factor keys to ensure a fluid visualization of the image and the associated annotations.

In addition to allow viewing the image at different resolutions, Icytomine allows users to download the current viewer viewport as a local sequence. This not only creates a sequence which users can work with on Icy, but also annotations are downloaded and included on the local sequence so that users can locally access and interact with annotations visible on the viewer. Moreover, users have the possibility to choose the download resolution of the image presented on the viewport. This really helps the speed of the analysis workflow since users can use the viewer to efficiently spot areas of interest on the image and then they can download them with more detail than the image presented on the viewer for further processing. A preview of the download dialog is presented on Figure 2.8

While allowing users to see annotations on images is a useful feature, it can

CHAPTER 2. LARGE IMAGE ANALYSIS: WHEN DATA SIZE MATTERS

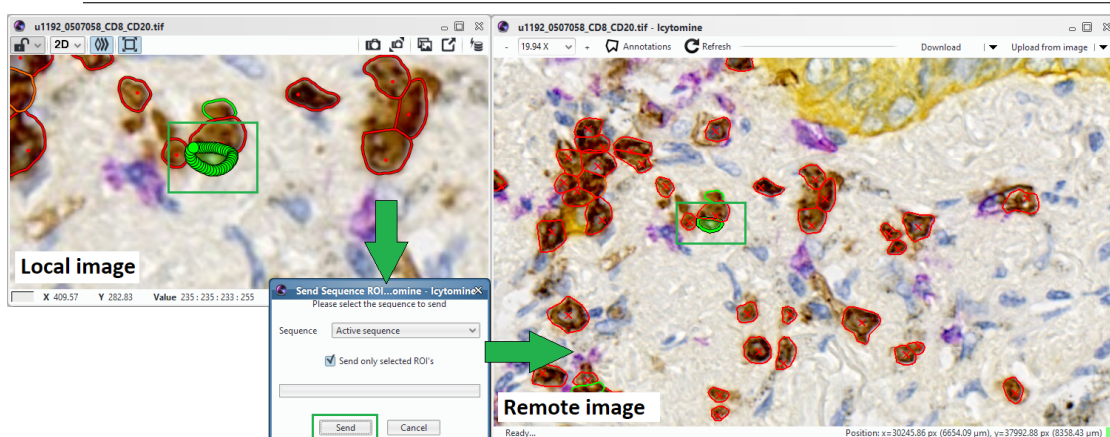


Figure 2.9 – Annotations made on Icy (result of some analysis) can be sent to the server using the Icytomine viewer.

be improved by allowing users to filter them according certain criteria. This is precisely one of the features that our viewer comes with. Users choose which annotations should be visible according to their associated label (term) or to their owner, and then download the image to Icy (See right panel on Figure 2.7). From this same panel users can also change annotation labels and delete annotations from the server, which comes in handy on heavy annotation transfers.

Moreover, after downloaded images are processed on Icy, the result is generally a set of regions of interest (ROI). Icytomine also offers the feature of sending these ROIs back to the server (Figure 2.9), with the possibility to associate them with a given label. This is particularly helpful when segmentation is performed on the images by allowing to store results remotely.

All of these features were implemented as a single interactive plugin known as Icytomine Explorer and is currently available as a downloadable plugin for Icy. Although these features are very practical for exploring and analyzing large images, it is meant to be used with one image at a time per plugin instance. This limits the productivity when handling automatically more than one image. For this, and following the protocol philosophy present on Icy, I created batch protocols to enable large image batch analysis. In the next section this feature is presented in more detail.

2.3.3 Automating image analysis

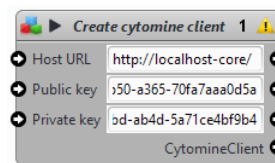
With the means to import remote large images at multiple resolution levels along with their respective annotations as ROIs, it is now possible to seek for automating these tasks and add automatic image analysis capabilities with Icytomine. Icy comes with a development tool set to create processing blocks that can be interconnected to form more complex analysis workflows, known as protocols. With protocols, developers can create workflows that make use not only of blocks developed by themselves, but also the blocks that Icy's large community has made

2.3. ICYTOMINE: INTEGRATING ICY AND CYTOMINE

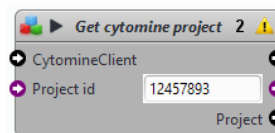
available to other users. With Icytamine I try to provide means to allow for creating protocols that provide access to data on large images (images as well as annotations), and enable automated analysis on these images. For this end, I have developed a set of protocols that expose the features of the core manager and allow to for batch processing of large images stored on Cytomine servers. Developed protocols can be classified into two categories. First, the functional protocols, providing a single result. And second, the batch protocols, which allow to loop over elements stored on the server.

The functional protocols developed for Icytamine to provide single results are the following:

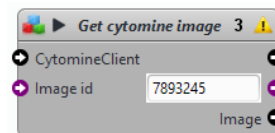
Create Cytomine connection: Allows to specify the credentials to access to a target Cytomine server



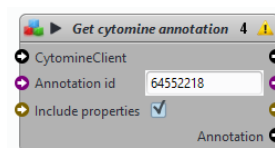
Get project: Retrieves the information of a project by its identifier



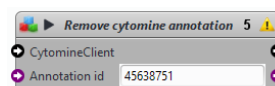
Get image: Retrieves the information of an image by its identifier



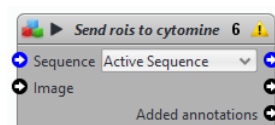
Get annotation: Retrieves the information related to an annotation by its identifier



Remove annotation: Removes an image annotation from the server given its identifier



Send annotations: Registers an image annotation on the server given its geometry and associated term



CHAPTER 2. LARGE IMAGE ANALYSIS: WHEN DATA SIZE MATTERS

The batch protocols developed for Icytomine to provide loop functionalities on elements on servers are the following:

Project loop: Given a project, this block iterates through all the images of the project one at a time

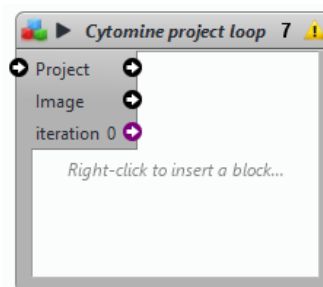
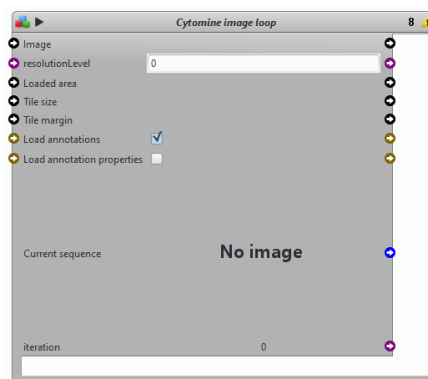
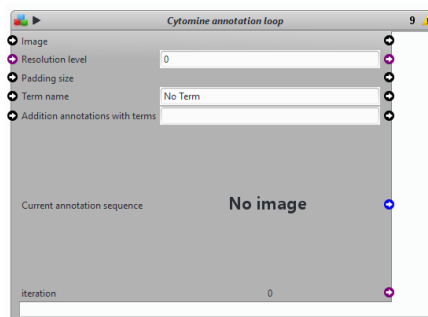


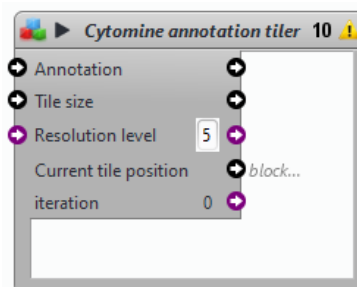
Image loop: Given an image, this block iterates through either the entire image or a specific sub region by patches downloading one at a time. If no patch size is given, the entire requested scope is retrieved at once



Annotation loop: This block behaves like the image loop, but instead of iterating through patches, it iterates through the annotations of a given image. A set of terms can be used as filter of the loop block, retrieving only images from annotations associated with the given terms



Annotation tiler: This block takes an annotation and iterates the image by patches retrieving those which intersect with the annotation. This is practical to avoid patches without useful content



2.4 Case of use: Using Icytamine to automatize glomeruli detection validation

As an example application making use of Icytamine, we presented a method where Icytamine helps in the process of glomeruli detection on kidney histological images by automating classification validation [50]. Glomerulus is a network of capillaries essential to the blood filtration by kidney. Accurate detection of glomeruli is the first important step in many tasks such as diagnostic, prognostic, or assessment of the kidney quality graft. Due to biological, tissue sample preparation and set-up acquisition variability, the glomerulus appearance could vary in terms of color, size and texture, making automatic recognition of glomeruli challenging as shown in Figure 2.10.

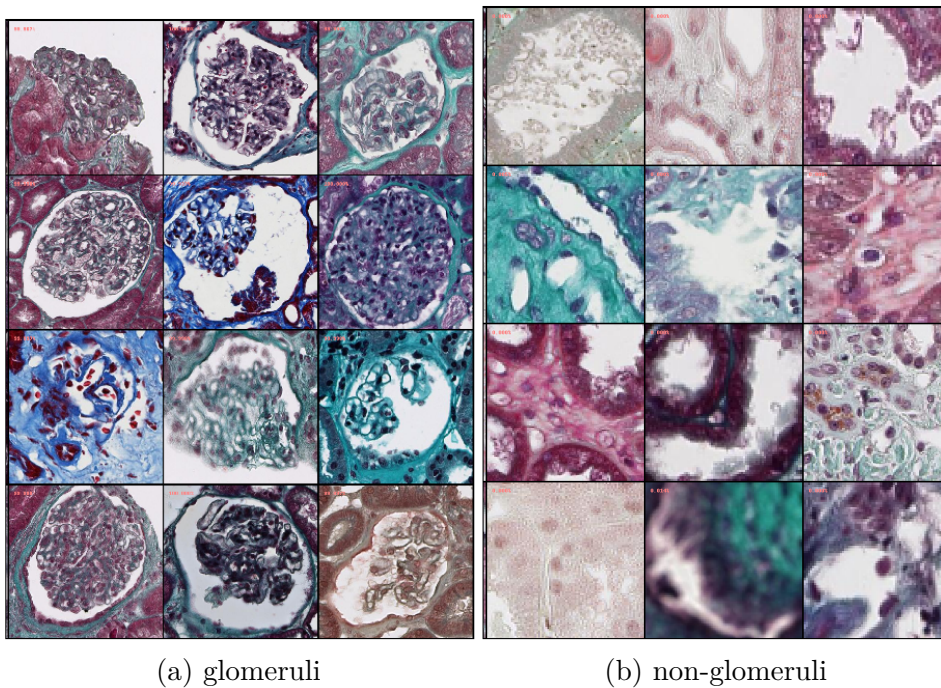


Figure 2.10 – Examples of the two classes stained in Masson's Trichrome

2.4.1 Previous works

Many studies define specific features, such as color, texture, histogram of oriented gradient (HOG) descriptors to characterize the glomerulus. However, the performance of feature-based patch-wise classification was not sufficient for a reliable classification or segmentation of glomeruli. Recently, many authors have explored the Deep Learning approach. Gadermayr [51, 52] proposed two different CNN cascades for segmentation applications with sparse objects. The authors in [53] described the development of a deep learning model that identifies and classifies non-sclerosed and sclerosed glomeruli in WSI of kidney frozen section biopsies. This model extends a convolutional neural network (CNN) pre-trained on a large

database of digital images. For more information, a comprehensive review can be found in [53].

2.4.2 Automatic glomerulus candidate generation

Our dataset contains 100 slides stained by Masson’s trichrome from 7 research centers and digitized with 20X objective by using a Nanozoomer scanner at $0.452\mu m$ pixel resolution, either Aperio scanner at $0.504\mu m$ pixel resolution or Zeiss AxioScan with $0.44\mu m$ pixel resolution.

Icy can provide glomeruli candidate patches as training set: 5857 glomeruli, 7047 non-glomeruli; test set: 595 glomeruli, 792 non-glomeruli.

As the shape is the solely invariance feature of the glomerulus appearance, its extraction is based on the ellipse fitting [54] of two kinds of feature regions: i) lumen extraction (when lumen is visible): extracted by a simple threshold that separate tissue from the background, ii) color texture segmentation (for other cases): the graph based region merging algorithm presented by Felzenszwalb and Huttenlocher [55] generating superpixels. However, this step generates also non-glomerular regions that should be removed from the final detection and could be done with a classification step. CNN appears as a good solution for its high versatility and efficiency.

2.4.3 Glomerulus CNN classification

Based on previous works, we chose the classical approach of fine-tuning the state-of-the-art architectures InceptionV3 [56], Resnet50 [57], as well as VGG16 [58]. VGG is characterized by its straightforward construction, with consecutive convolution and spatial reduction blocks which sum up to a large number of parameters i.e. 140M, making it computationally costly. Both ResNet and Inception are based on micro-modules which introduce different pathways in the propagation of information, with the benefit of having dramatically less parameters, $\approx 24M$ each, while also bringing improved performance.

All architectures were pre-trained on ImageNet and a fully-connected layer of 1024 neurons was added on top as a classifier along with one last neuron as output for binary classification. Fine-tuning was performed by minimizing the cross-entropy loss using stochastic gradient descent on mini-batches of 64 patches with a learning rate of 0.001 and momentum of 0.9. An average of 110 epochs (learning passes on the full training set) was sufficient to reach a stable result for all the architectures.

The architectures were fed with patches of their respective expected input, i.e. 299x299 pixels for InceptionV3 and 244x244 for ResNet50 and VGG16.

Validating classification

This framework uses image processing to create candidate patches and need a weak interaction from the user to validate these patches as glomerulus or not.

2.4. CASE OF USE: USING ICYTOMINE TO AUTOMATIZE GLOMERULI DETECTION VALIDATION

This can be done either with the Cytomine web proofreading tool or the Icytomine viewer, to re-classify manually candidates. Figure 2.11 shows the Icy protocol that eases the batch processing task. The best results are obtained with the Inception network (See Table 2.2). In addition, some classification examples are presented in Figure 2.12.

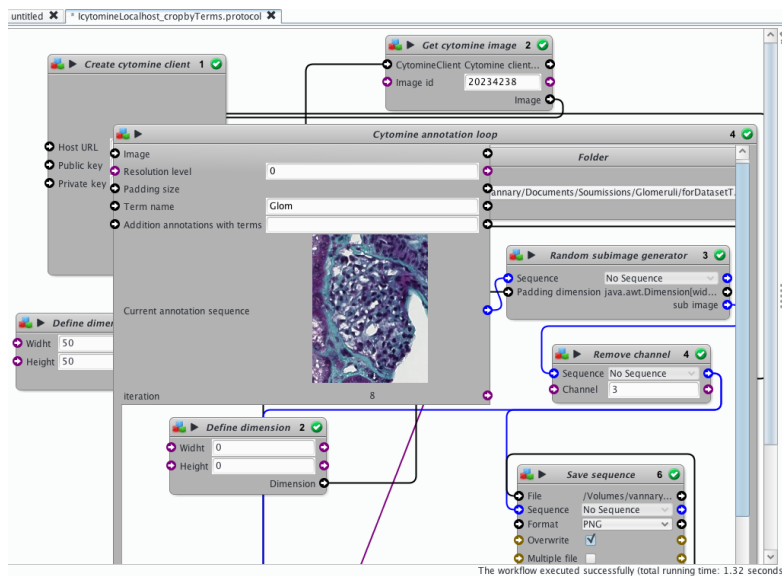


Figure 2.11 – Batch processing with Icytomine. An image can be processed by tiles that are configured before the batch processing. Extraction of glomeruli within an image of a project.

Method	Accuracy
VGG [58]	0.94375
ResNet50 [57]	0.97188
InceptionV3 [56]	0.975

Table 2.2 – Classification accuracy

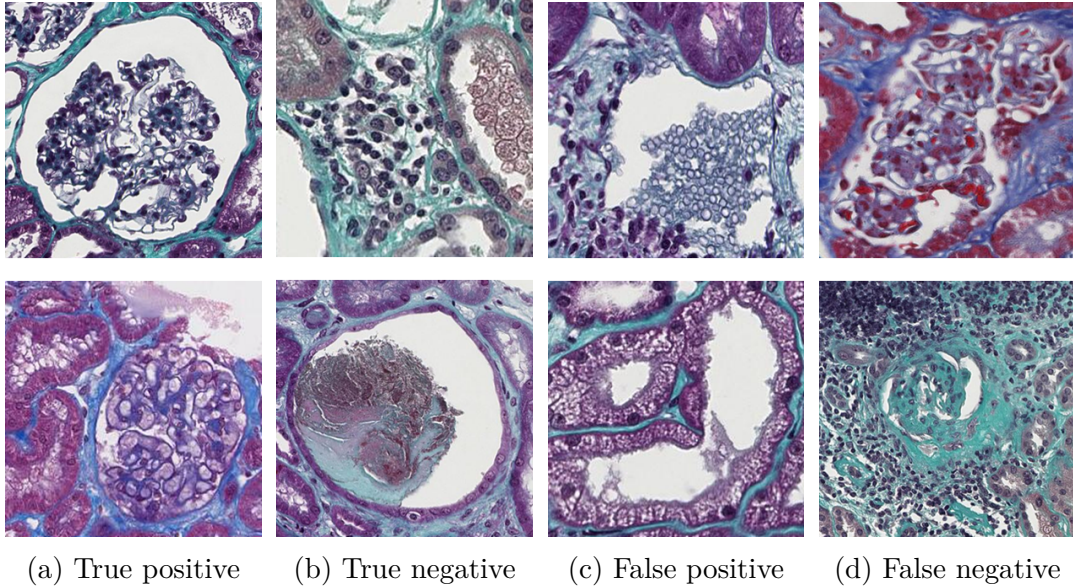


Figure 2.12 – Examples of glomerulus CNN classification results.

2.4.4 Glomerulus segmentation by Mask R-CNN

To go further, we tested Mask R-CNN [59], which is a state-of-the-art instance segmentation architecture, able to achieve pixel accuracy. For the presented work we used the Keras & Tensorflow implementation by [60], with a ResNet50 backbone, also pre-trained on ImageNet.

After fine-tuning the network, we obtained a score of 78% of the mean average precision (mAP). However, the more interesting observations emerged from the visual assessment of the predicted segmentation. We notice that this approach is able to produce a more precise segmentation of the glomeruli borders, while also being robust to the outliers (erroneous ground truth) in the automatic annotation. Studying the cases with null intersection over union (IoU) between the ground truth and prediction, we noticed that Mask R-CNN was able to discover non-annotated positives (both in train and validation set, which is a good indicator for the lack of over-fitting). We also observed that this approach is adapted for separating adjacent glomeruli (See Figure 2.13).

From these results we deduce that our automatic training set generation method is well suited also for segmentation and what is more, Mask R-CNN can be applied as a last step in refining the annotations.

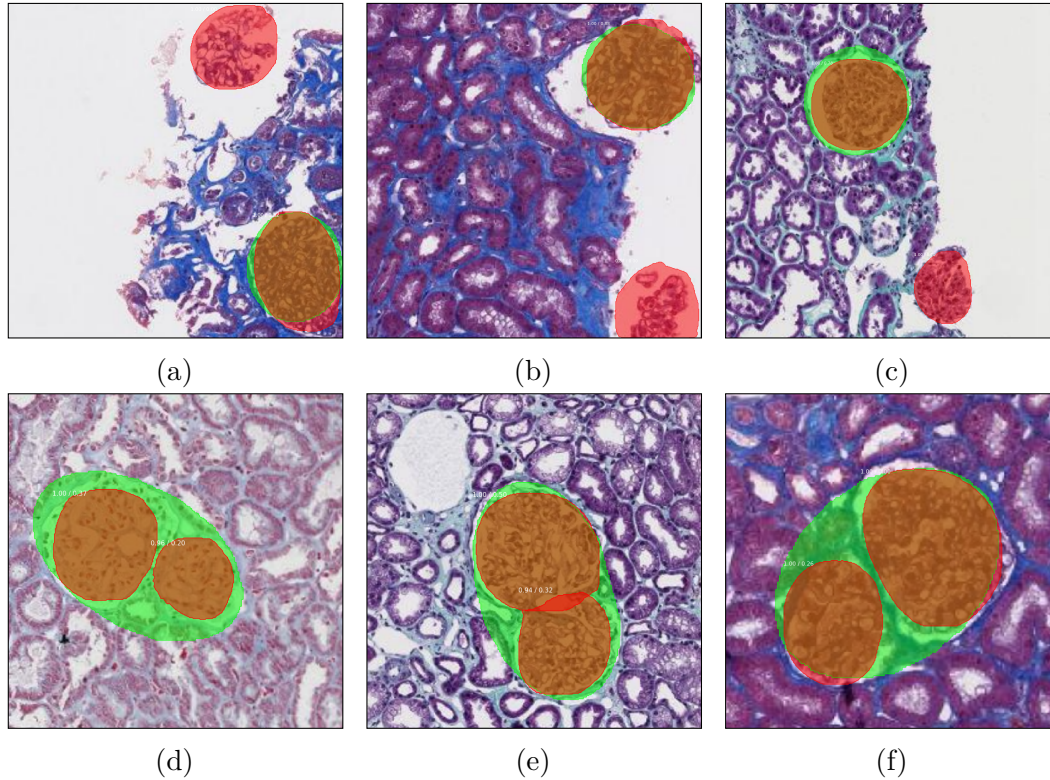


Figure 2.13 – Examples of Mask R-CNN segmentation results (ground truth in green, prediction in red): glomeruli detection on sample borders in (a)(b)(c), separation of adjacent glomeruli in (d)(e)(f)

2.5 Conclusion

Handling whole slide images can be a complex task in terms of performing image analysis while traversing the image. Icytamine assesses this problem by providing an effective solution to the needs of Icy users looking to have access to large planar images stored remotely. It allows users to explore and view images stored on remote servers with a minimized latency, giving as a result a high performance tool for image exploration and annotation. In addition, the protocols developed for automatization enable users to create their own image analysis workflows including access to high resolution virtual slides without much effort. It is important to mention that Icytamine development was performed using high use of unitary tests to allow a stable incremental development. Although developing with unitary tests can take time, it allows to prevent side effects on existing features when new ones were added. In perspective, the Icytamine viewer could be integrated to the Icy core as a solution to access remote images. Furthermore, a side project should be created to extend Icytamine with plugins that provide general predefined image analysis features using Icytamine, helping to further improve analysts productivity. Since Icytamine is actually a proxy layer between Icy and remote image servers, it could provide extended access not only to Cytomine servers, but

CHAPTER 2. LARGE IMAGE ANALYSIS: WHEN DATA SIZE MATTERS

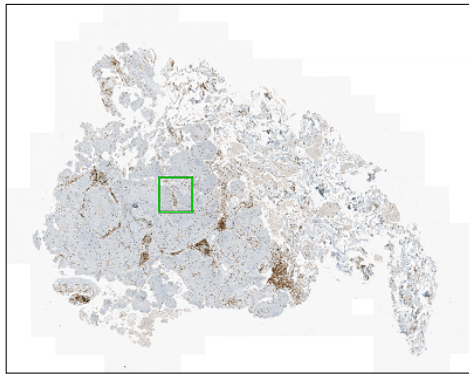
also to other large image servers that hold a similar system data organization (i.e. users, projects, images and annotations). Finally, as a consequence of Icytome's strong and effective usage of memory caching, Icy has been developing lately its own caching system for the classic 2D and 3D viewer which will provide an even faster access to the already efficient sequence viewer.

Chapter 3

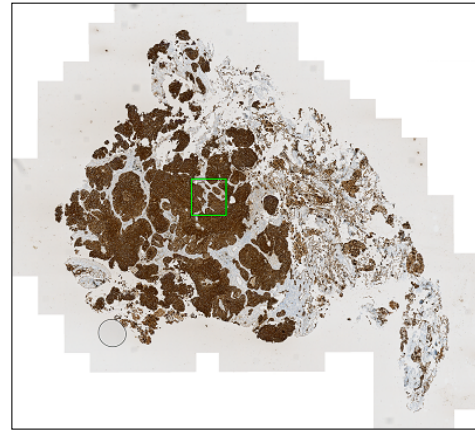
Large image registration

In histopathology it is necessary to use several dyes and sometimes several imaging modalities for the observation of tissue samples on serial sections, in order to reveal different structures of the organ under study. However, the number of chemical or immunohistochemical dyes is limited (up to 12) on a single section [61]. Additionally, the more dyes present in the image the more complex the stain separation gets, making color decomposition not suitable for these cases. Despite the possibilities offered by this technique, integrating information coming from different slides of the same tissue becomes a challenge. Indeed, the preparation of the tissue (fixation, embedding, cutting and staining), and digitization of the sample can generate variations on the images of the tissue for each section at the same locations. This change becomes an important limitation when automatically analyzing these images. A popular technique for handling this limitation is performing an image registration, a transformation applied to each digitized section of the tissue sample, that aligns each section from the sample with respect to a given target section. This type of treatment is very popular and is commonly applied when the information in the images to be registered show similar information. Unfortunately, this is not the case when the dyes used for staining vary from section to section. Conventional registration algorithms seek to register images by associating the intensities of the source image with the same intensities on the target image, which is not the case in images with different stains (see Figure 3.1).

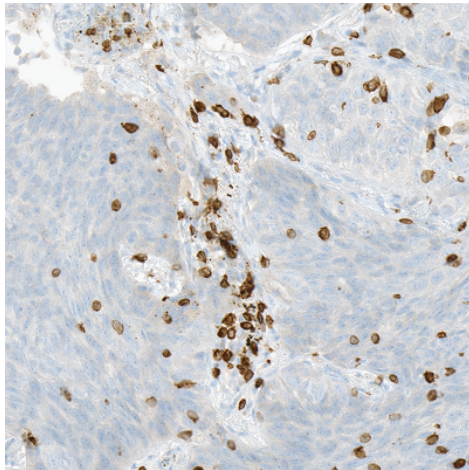
In this chapter a method for registration of multi-staining histological images is presented. More precisely, the work presented focuses on registering contiguous pairs of slides marked differently (labeling immune cells on one slide and cancerous tissue on the other), which does not allow a direct relation on the intensities on both images. These images take a considerable size up to 90000 by 90000 pixels per image, making registration on these images prohibitive in time if handled as classic methods do (i.e. by performing registration of the image at full resolution). First, a quick state of the art on existing registration techniques is presented. Second, the method proposed for registration is presented highlighting in particular two main issues: *i*) registering images with weak common information, and *ii*) handling transformations on large histological images. The images used for this



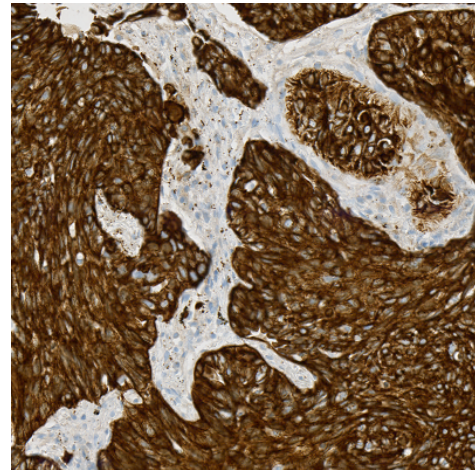
(a) Biopsy labeling immune cells (in brown)



(b) Biopsy labeling cancerous tissue (in brown)



(c) Detail of labeled immune cells



(d) Detail of labeled cancerous tissue

Figure 3.1 – Images with multiple stains are particularly challenging to register due to the variation of position, shape and intensities of structures present in the tissue of the scanned slide series.

work come from non-small cell lung cancer biopsies (adenocarcinoma and squamous cell-carcinoma), taken from patients diagnosed with this type of cancer at different stages. These images were acquired by the tumor immunology group at the Oslo University Hospital in Norway. Half the acquired images were marked using Cytokeratin to label normal and malignant tumor cells, while the other half were marked with CD3 to label T cells.

3.1 Registration principles

Image registration is the mapping of a *source* image so that it is as similar as possible to a *target* image, using the information in the images or external information that can help to find the optimal transformation. It is also possible to register multiple target images using the same target image. Image registration is a subject that has been extensively studied in image processing because these methods can be used in many context such as medical imaging [62], pattern recognition [63], biological imaging [64], shape tracking on videos [65], etc. [15]. These methods are often used to track objects on image sequences, to align multiple images, or to find the correlation between multiple viewpoints of the same scenario.

In general, the main steps of the registration process are as follows.

- ***Feature detection on images:*** characteristic elements of the images are detected so that they can then guide the registration. For this step, there are methods that use Harris salient features [66, 67], scale invariant features [16, 68, 69], and edge detection features (Canny [70], Marr-Hildreth [71]). This step is very important because the quality of the result is strongly dependent on the quality of the features extracted from the images [72]. There are methods that do not detect features present in the images, called iconic registration methods.
- ***Feature mapping:*** features found in the previous step are linked between the source and the target image. These links are created from feature descriptors or with similarity criteria of the features. Methods using spatial relation are usually used for this [72], as well as the consistent feature labeling [72] and the classification of invariant feature descriptors [68, 69]. Furthermore, similarity criteria such as cross-correlation [73, 74] or mutual information [75] can be used for this step.
- ***Transformation model estimation:*** with the paired features a transformation model is constructed to transform the source image and maximize the similarity with the target image. The model created in this step can be described using rigid transformations [76] (scaling, translation, rotation), affine transformations [77] (rigid and shear), or elastic transformations [78] (a different transformation at each point of the space). Another way of understanding these types of transformation is by looking at how they relate to

the transformed image. In rigid and affine transformations the transformation is homogeneously applied to the whole image (global transformation), while elastic transformations have a specific transformation applied at each position in the image (local transformation).

- ***Resampling and transformation of images***: once the transformation model is built, it is applied to the target image to obtain the registered image as result. Different interpolation techniques are used in this step to handle pixel coordinates and intensities with non-integer values resulting from the transformation model [79, 80]. In addition, writing the image to disk is performed during this step. For this, both serial and parallel read/write techniques can be applied to optimize the creation of the resulting image file [19].

In addition, registration methods can be classified into two main types. Some of the references presented in the methods classification below can be repeated here to give an glimpse of the relation between the steps and the types of registration used by existing methods.

- ***Feature methods (geometric registration)***: methods in this category are based on the characteristics identified during the first stage of the registration to calculate the transformation model. In this type of registration, popular methods can be mentioned here such as scale invariant feature transform (SIFT [68], SURF [69]), edge detection registration (Harris-Laplace [67]), or line detection registration (Marr-Hildreth [71], Canny [70]).
- ***Local area-based methods (iconic registration)***: these methods use the pixel neighborhood information (gradient, texture, color similarity, etc.), to correlate the images to be registered. these methods typically use a pyramid approach to analyze images at different resolution levels and take into account the coarse and fine detail to align images. This category includes cross-correlation registration [73], mutual information [75, 81] and B-splines regularization [82].

3.2 Registering multi-stained large images: Big-BUnwarp

Histological images acquired with different instruments (multiple modalities), or using multiple dyes (multi-staining), present an additional challenge to that of conventional image registration. Intensities on source and target images are not linearly correlated, which prevents a reliable extraction of redundant features in the images. For that we propose to go through a pre-processing step which puts the two images in the same space of intensities thus allowing to correctly register the images with multiple dyes or modalities. Given the elastic nature of the tissues present on histological images, existing registration methods for this type of image

make use of elastic deformable transformation models, which make it possible to model local deformations of the tissue [83]. For the method presented in this chapter, the registration is carried out using B-Spline-regularized transformations, they allow to approximate the elastic nature of the tissue while at the same time modeling the transformation necessary to register the images [82]. In addition, this type of model makes large image processing simpler. The steps involved in the registration are presented and explained in more detail in the following sections.

3.2.1 Image pre-processing: Common information extraction

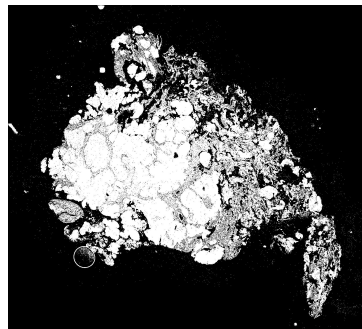
Despite having the same tissue sample on both images to be registered, the intensities are too different to attempt a registration directly on the original images. Indeed, the common information on this type of images is restricted to the delineation of the tissue of the image background. This information can be extracted by segmenting the tissue present in the image.

In the case of this project, we have applied a threshold segmentation of the histogram of the converted gray-scale image, this thresholding is done with a k-means partitioning of the histogram [84] (See Figure 3.3). Because the dyes used to label specific structures on the images are different, each image type uses a different amount of classes to separate the background from the tissue. Several configurations were tested giving the best results as follows. For the image containing the immune cells, its histogram is empirically separated in 3 classes. And for the image containing the cancerous tissue, the histogram is split in 5 classes, also empirically. These classes are fused to define a binary image separating the pixels of the tissue from the pixels representing the background. These parameters were set by testing 8 pairs of corresponding images (8 images marking the immune cells and 8 images marking the cancerous tissue). Other binarization methods such as watersheds and graph-cuts among others have been tested without giving satisfactory results (See Figure 3.2). It is important to note that these parameters do not apply for other type of images (with other stains) as the thresholds are specifically adapted to the intensities of the dyes used on these images. Different thresholds or even different methods might be needed to extract the tissue contour on images using different staining techniques (also called image matting).

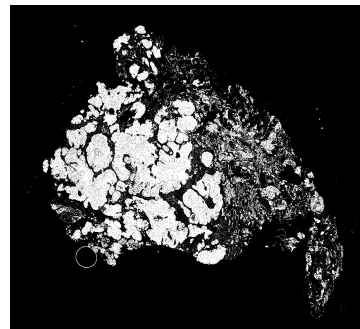
3.2.2 Image registration method: Building the transformation model

In order to create a transformation model it is necessary to specify how it is stored. For this a strong use of B-Spline functions is done, which act as link between the discrete features of digital images and the continuity of intensities present on the pixels of the images [79].

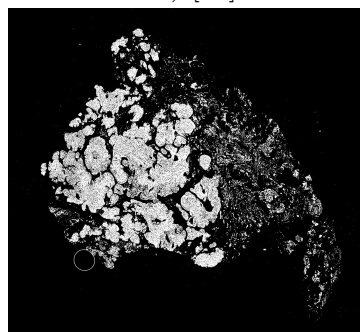
B-Splines can be expressed with the equation $\mathbf{C}(t) = \sum_{i=0}^n \mathbf{P}_i N_{i,p}(t)$, where the value of the curve or function can be found by interpolating the set of points \mathbf{P} and



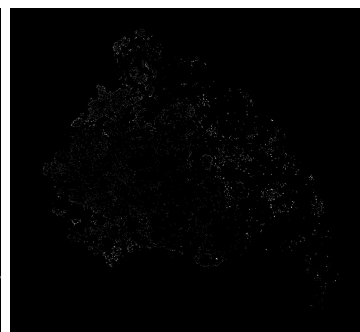
(a) K-Means Thresholding
(highest of 5-classes) [84]



(b) Otsu Thresholding [85]

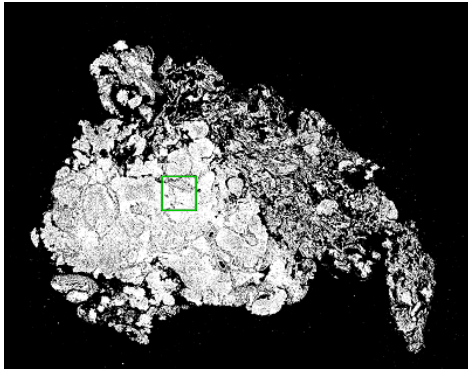


(c) Huang Thresholding [86]

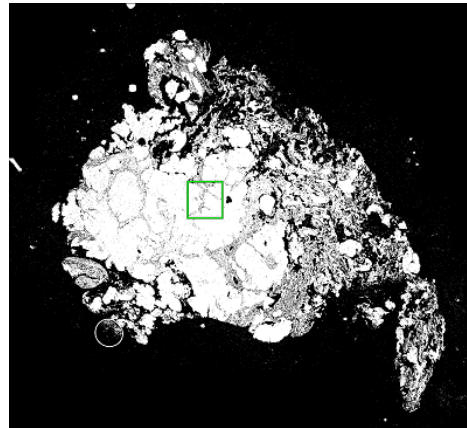


(d) Shanbhag Thresholding [87]

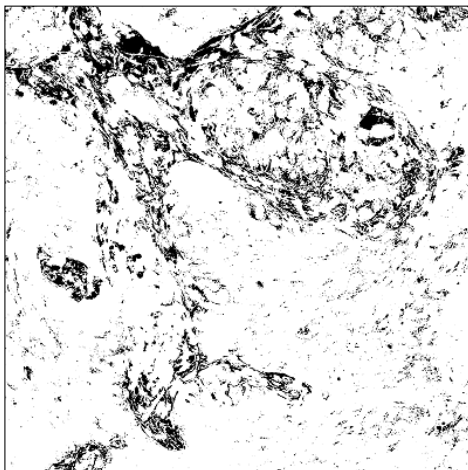
Figure 3.2 – Tissue segmentation methods comparison example. Several methods were tested to extract the tissue out of virtual slide images. However, adapted K-Means segmentation gave the most suitable results from our tests.



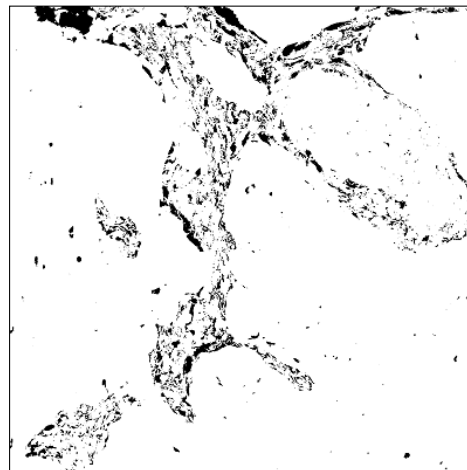
(a) Tissue segmentation on biopsy labeling immune cells



(b) Tissue segmentation on biopsy labeling cancerous tissue



(c) Detail of segmentation of Figure 3.3a



(d) Detail of segmentation of Figure 3.3b

Figure 3.3 – Segmented images by applying K-means thresholding on their gray-level histogram.

a B-spline basis function N_p of degree p . For this project we chose cubic B-Spline functions (of degree 3) because they offer a good trade-off between performance and interpolation precision [88]

With B-splines described as above it is now possible to express a two-dimensional image in terms of a B-Spline function in two dimensions:

$$I(x, y; h) = \sum_{k,l \in \mathbb{Z}^2} c_{k,l} \beta^3\left(\frac{x}{h} - k\right) \beta^3\left(\frac{y}{h} - l\right) \quad (3.1)$$

where $\beta^3(x)$ is the cubic B-Spline function and the coefficients $c_{k,l}$ are the control points at position (k, l) . $h = 2^0, 2^1, 2^2, \dots$ is a parameter that controls the detail of the representation (smaller values give a more detailed representation), which allows for image processing at multiple resolution levels, hence a pyramidal approach.

In addition, to express elastic deformations a transformation model is used, obtained from the linear combination of B-Splines

$$\begin{aligned} \mathbf{g}(\mathbf{x}) &= \mathbf{g}(x, y) \\ &= (g_1(x, y), g_2(x, y)) \\ &= \sum_{k,l \in \mathbb{Z}^2} \begin{pmatrix} c_{1,k,l} \\ c_{2,k,l} \end{pmatrix} \beta^3\left(\frac{x}{s_x} - k\right) \beta^3\left(\frac{y}{s_y} - l\right) \end{aligned} \quad (3.2)$$

where s_x and s_y are scalars controlling the level of detail of the deformation field representation.

As described by Sorzano [89], the transformation model is described as an energy minimization function composed of three terms: A similarity measure E_{img} between the images to register, an optional constraint E_μ of corresponding points on the images, and two regulation terms E_{rot} and E_{div} to restrict undesired deformations. The energy equation using these three terms is as follows:

$$E = w_{img} E_{img} + w_\mu E_\mu + (w_d E_{div} + w_r E_{rot}) \quad (3.3)$$

Here w_x is a weight coefficient for each energy term. The higher the value of w_x is, the more penalized the term x is going to get.

The terms of the energy function are described below:

- E_{img} : The difference between the source image and the target image, described by:

$$E_{img} = \frac{1}{\#\Omega} \sum_{\mathbf{x} \in \Omega} (I_t(\mathbf{x}) - I_s(\mathbf{g}(\mathbf{x})))^2 \quad (3.4)$$

where $\Omega = \{x \in \Omega_t \cap \mathbb{Z}^2 : \mathbf{g}(\mathbf{x}) \in \Omega_s \cap \mathbb{Z}^2\}$ is a mask common to both source and target images, and $\#\Omega$ is the size of this mask in pixels.

- E_μ : The difference between the corresponding points in the source(μ_s) and target(μ_t) images. These point can be found using salient feature extraction

3.2. REGISTERING MULTI-STAINED LARGE IMAGES: BIGBUNWARP

methods such as SIFT or SURF. These points can improve the registration results by giving a hint of the transformation at specific positions of the image. This term is described by:

$$E_\mu = \frac{1}{N} \sum_{n=1}^N \|\mu_t^{(n)} - \mathbf{g}(\mu_s^{(n)})\|^2 \quad (3.5)$$

- **E_{div} and E_{rot} :** The regulation terms based on the second derivative of the transformation model that allows for an estimation of point-wise(E_{div}) and radial(E_{rot}) image tearing at each position on the result image. These terms are described by:

$$\begin{aligned} E_{rough} &= w_d E_{div} + w_r E_{rot} \\ &= w_d \int_{\mathbb{R}^2} \|\nabla \text{div} \mathbf{g}\|^2 dx dy \\ &\quad + w_r \int_{\mathbb{R}^2} \|\nabla \text{rot} \mathbf{g}\|^2 dx dy \end{aligned} \quad (3.6)$$

where $\text{div} \mathbf{g} = \partial_x g_1 + \partial_y g_2$ represents the divergence of the vector field \mathbf{g} , $\text{rot} \mathbf{g} = -\partial_y g_1 + \partial_x g_2$ is the measure of the rotational of \mathbf{g} , and $\nabla f = (\partial_x f, \partial_y f)$ is the gradient of the scalar function f . The divergence of a vector field is linked to the existence of source/sink type of deformations, and the rotational is related to the rotation of the image around a specific point on the image (see Figure 3.4).

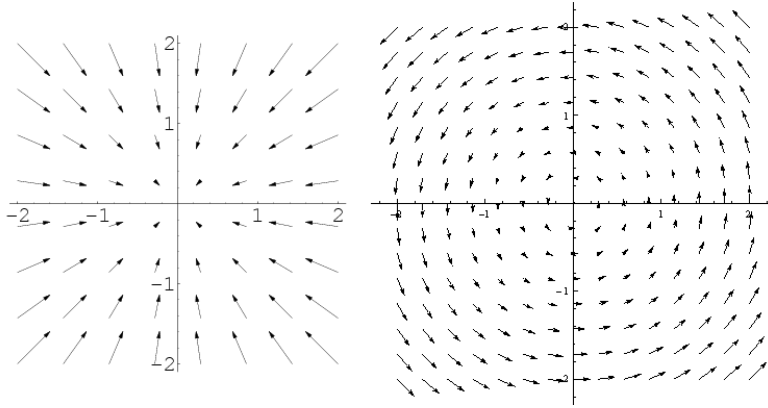


Figure 3.4 – Examples of deformation vector field of divergent (left) and rotational (right) types.

With these terms defined, the energy minimization function can be solved using a non-linear regression optimization algorithm. In this case we chose to use the Levenberg-Marquardt method [90], which behaves as a quasi-Newton gradient descent. The coefficients $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$ of the transformation model are updated by iterating $\mathbf{c}^{(k+1)} = \mathbf{c}^{(k)} + \Delta \mathbf{c}^{(k)}$, where $\Delta \mathbf{c}$ represents the solution of the equation system $\mathbf{H} \Delta \mathbf{c}^{(k)} = \nabla E(\mathbf{c}^{(k)})$. Here, $\nabla E(\mathbf{c}^{(k)})$ is the gradient of the energy with

respect to the deformation evaluated at $\mathbf{c}^{(k)}$, and $\widetilde{\mathbf{H}}$ is a modified version of the hessian \mathbf{H} defined as $[\widetilde{\mathbf{H}}]_{ij} = (1 + \lambda\delta_{ij})[\mathbf{H}]_{ij}$, where $\delta_{ij} = 1 - |\text{sign}(i - j)|$ is the Krönecker function. When $0 < \lambda \leq 1$ is close to 1, the step of the iteration behaves like a gradient descent, limiting the step size. On the contrary, for values of λ close to 0, the step behaves like a Newton method with fast convergence.

When the energy function E converges, the result transformation model $g(x)$ is used to reconstruct the final registered image. Since the registration is done on the gray-level version of the original images, to create the final registered image the transformation model is applied on each of the channels of the original color image. This way the resulting image is on the same color space of the input image. To compute the color intensity values at each pixel of the result image it is necessary to use interpolation on the intensities at positions $(x, y) | x, y \in \mathbb{R}$ of the input image. For this, B-Spline functions are also used to perform interpolations.

One way to show the images before and after the registration is by superposing the segmented images with $\Delta_{s_0t}(\mathbf{x}) = \mathbf{x}_{s_0} - \mathbf{x}_t$ and $\Delta_{s_1t}(\mathbf{x}) = \mathbf{x}_{s_1} - \mathbf{x}_t$ for the comparisons before and after registration, respectively (See Figure 3.5). The whole registration workflow can be summarized on Figure 3.6.

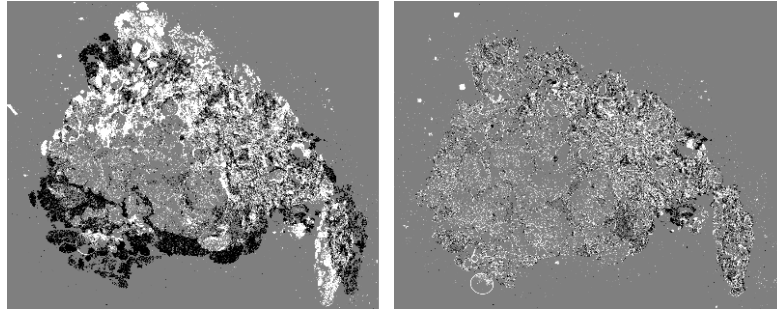


Figure 3.5 – Comparison between segmented images before and after registration. White/black areas represent registration errors, and gray areas represent correctly registered areas. The parameters used for this examples were $w_{img} = 1$, $w_{\mu} = 0$, $w_d = 0.01$, $w_r = 0.01$, which provide precise (taking image intensities into account) and smooth (by relaxing the energy equation with small values for w_d and w_r) results.

3.2.3 Image post-processing

Although the registration process described above is carried as a pyramidal process at different resolution levels, it is necessary that the image be fully loaded in RAM memory to perform the processing. Given that images processed for this project are large, loading entirely a single image becomes difficult; and it is even more difficult to register these large images because the calculation time becomes prohibitive and it is necessary to duplicate the images several times when processing the images. Then, to handle this constraint and taking advantage of the fact that most of the deformations needed to register these images are found at low resolution levels,

3.2. REGISTERING MULTI-STAINED LARGE IMAGES: BIGBUNWARP

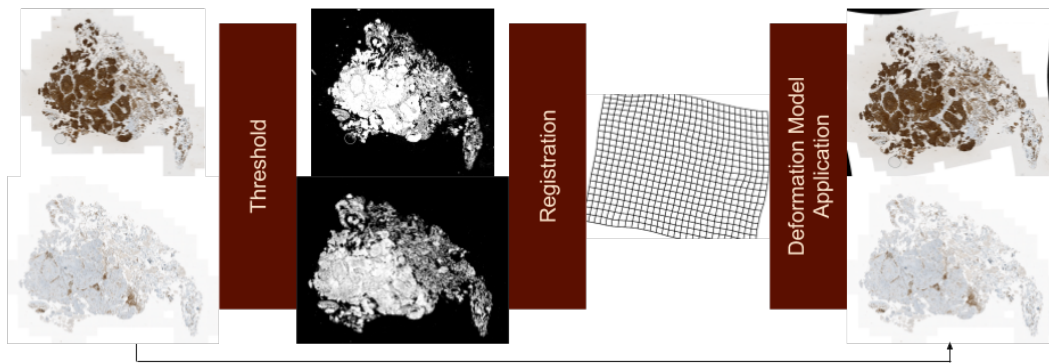


Figure 3.6 – Image registration process summary. Tissue on input images is segmented to identify common information areas and put them on the same intensity space. Next, target image is registered by computing the transformation model on the segmented images. Finally, the model is applied on the input color image to obtain the resulting registered image.

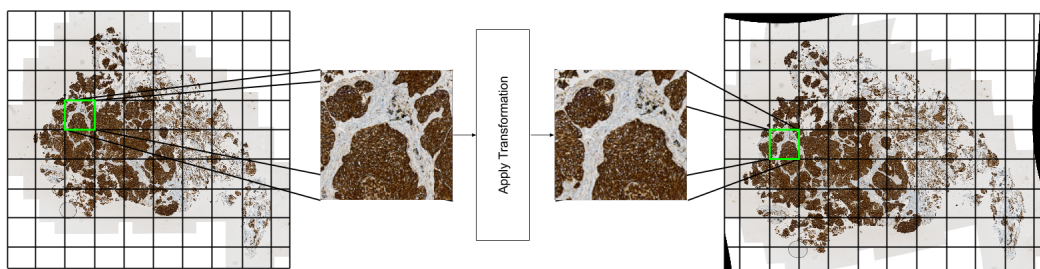


Figure 3.7 – Applying the transformation model to large input images. The transformation is applied by tiles, processing multiple tiles concurrently to take full advantage of the machine CPU cores.

the registration is performed on a reduced version of the original images. This makes it possible to quickly calculate the transformation model without exceeding the memory limits imposed by currently used machines.

To create the resulting image with the original size, the transformation model is then scaled by $\mathbf{G}(\mathbf{X}_s) = k\mathbf{g}(\mathbf{x}_s)$ where k = output image width / low res transformed image width is the scaling constant of the images computed. With \mathbf{G} we build the resulting image by traversing the image by tiles and calculating concurrently a small number of tiles at a time by $\mathbf{G}(\mathbf{X}_s) = \{\mathbf{G}_1(\mathbf{X}_{s1}), \mathbf{G}_2(\mathbf{X}_{s2}), \dots, \mathbf{G}_n(\mathbf{X}_{sn})\}$ where \mathbf{G}_i is the transformation associated with the tile \mathbf{X}_{si} (See Figure 3.7). Creating the result image this way, allows the memory constraints of the machine to be always respected. In addition, since all the processors are used simultaneously, it takes full advantage of the computing power of the machine.

3.3 Method evaluation

Although there exist several registration methods, the lack of a standard evaluation method has made this task a challenge. Some approaches point in the direction of using the pixel intensities to evaluate the registration success [82]. However, this approach does not give an accurate evaluation when source and target images are not exactly the same image, differing just by a geometric transformation, which is the case in multi-stained image registration. Other methods propose the use of corresponding points on source and target images to measure how much the positions of the points differ from one image to the other once they are transformed [81]. The problem with this approach is that, for once, it does not take into account the whole image but just a set of positions in the image, ignoring the rest of the transformed image space. In addition, this type of evaluation requires a qualified person to spot corresponding key points on both source and target images, which can be a complex task when using a large datasets for evaluation. Furthermore, other evaluations can be performed to quantify the performance of the registration such as registration time, registration robustness, etc. [91] which are more like additional benchmarks than real accuracy metrics. In order to get a meaningful comparable evaluation, we based our evaluation on the recently published challenge for automatic non-rigid histological image registration (ANHIR) [92]. In this challenge authors propose an evaluation of a large set of corresponding points on almost 50 histological image sets. Different dyeing techniques are used on each of the cuts of the same tissue that is acquired. These images vary in size from 15000 by 15000 pixels up to 50000 by 50000 pixels. More specifically the stains used in these images are Clara cell 10 protein (Cc10), prosurfactant protein C (proSPC), hematoxylin and eosin (H&E), antigen KI-67 (Ki67), platelet endothelial cell adhesion molecule (PECAM-1, also known as CD31), human epidermal growth factor receptor 2 (c-erbB-2/HER-2-neu), estrogen receptor (ER), progesterone receptor (PR), cytokeratin, podocin (See examples in Figure 3.8).

3.3.1 Error evaluation method

According to the scoring method [91, 92], the relative Target Registration Error (rTRE) metric is used to evaluate the accuracy of registered images. Let $TRE = d_e(x_l^T, x_l^W)$, where x^T and x^W are the landmark coordinates in the target and warped image and d_e is the Euclidean distance between a pair of corresponding landmarks. From this, the relative target registration error is defined as $rTRE = \frac{TRE}{\sqrt{w^2+h^2}}$, where w and h are the image width and height respectively. On each image the $rTRE$ is evaluated for all corresponding points in the images and then the median $rTRE$ value is taken for each image so that slight inaccuracies are not penalized when the rest of landmarks are correctly aligned. From the median $rTRE$ computed on all images, a average registration error can be computed from these values as Average median $rTRE = a_d(m_i(rTRE))$ where a_d is the mean average and m_i is the median value.

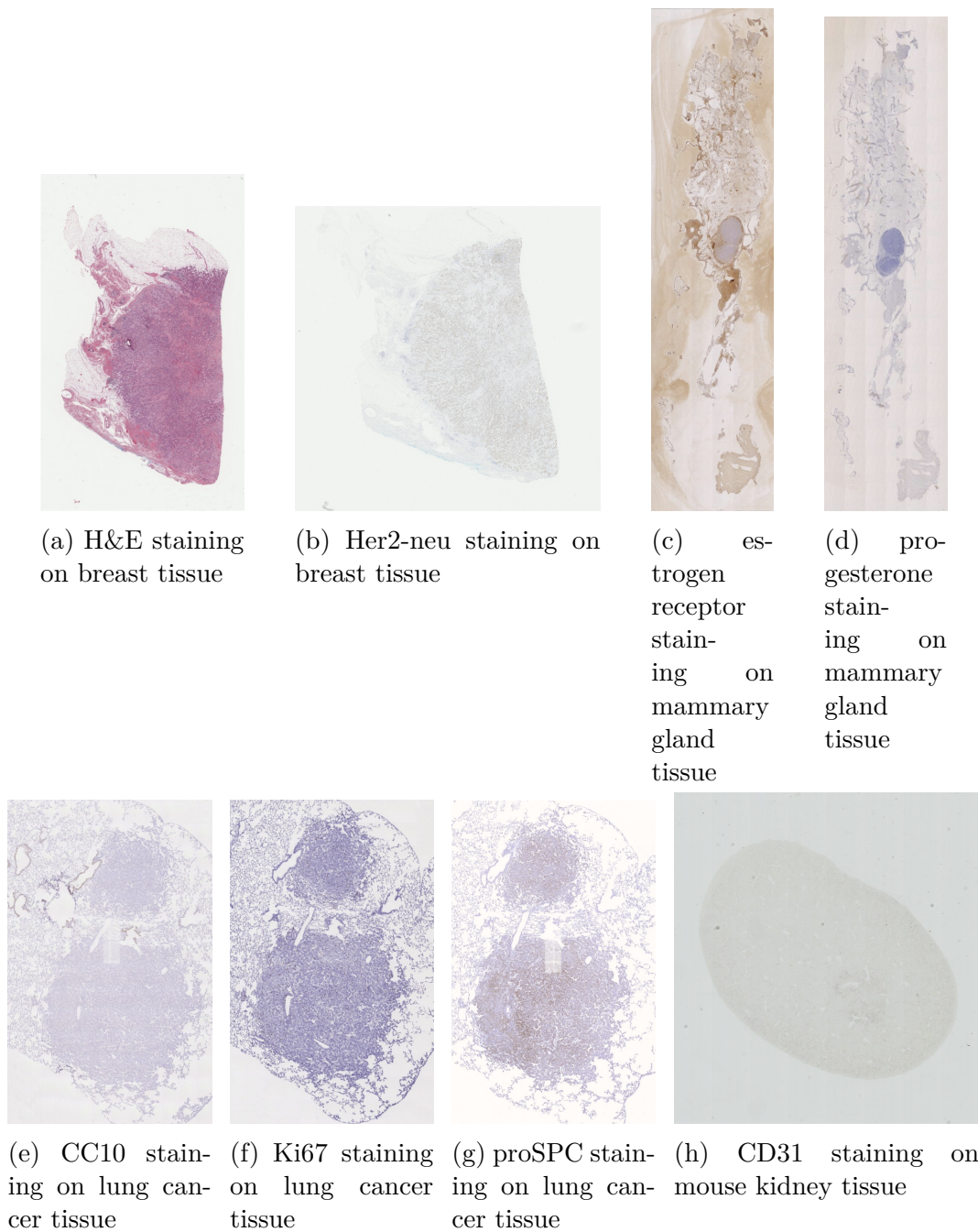


Figure 3.8 – Different staining techniques were used for the ANHIR challenge. Here some examples are presented to show the heterogeneity of these techniques.

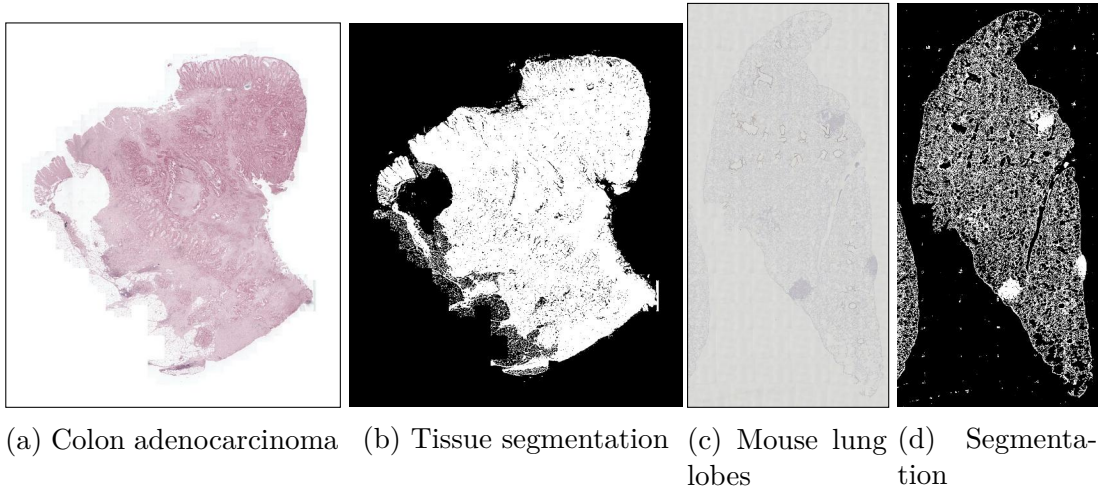


Figure 3.9 – Successful segmentation examples of tissue on virtual slides from ANHIR registration challenge.

3.3.2 Preparing the dataset

For this dataset we set the pre-processing segmentation as a 5 class K-means histogram classification on the image converted gray levels and taking the highest threshold to differentiate tissue from background. This proved to be a good approach in most cases where the background was uniform and the tissue made enough contrast with the background to differentiate both (see Figure 3.9). Some few images had a very complex background that made the segmentation step fail (see Figure 3.10). As for the registration the best parameters were found at $w_{img} = 1$, $w_d = 0.1$ and $w_r = 0.1$, which allowed to register according to the segmentation pixel intensities while keeping the transformation away from strong rotational and divergent deformations. Although the majority of the images got to be correctly registered and provided accurate results (see Figure 3.11), our method failed to perform a correct registration when the tissue had a strong misalignment between source and target images (see Figure 3.12). This had a much greater impact on the score of our evaluation since some of the image sets presented this kind of transformation which our method could not handle correctly.

3.3.3 Scoring our method

Evaluating our solution (see Figure 3.13), we obtained an average median $rTRE = 0.03027$, and an average registration time = 0.90592 minutes which places our method right in front of the original BUnwarpJ method with a score of 0.03409, a comparison that confirms the method precision, since it is based on BUnwarpJ. However, in this challenge BUnwarpJ was placed at the 8th position, mostly because it does not use either mutual information or normalized gradient fields (NGF [93]) as the registration criterion (which were used by winner methods) and fails to match correlated pixel intensity sets. In our case, we improved BUnwarpJ

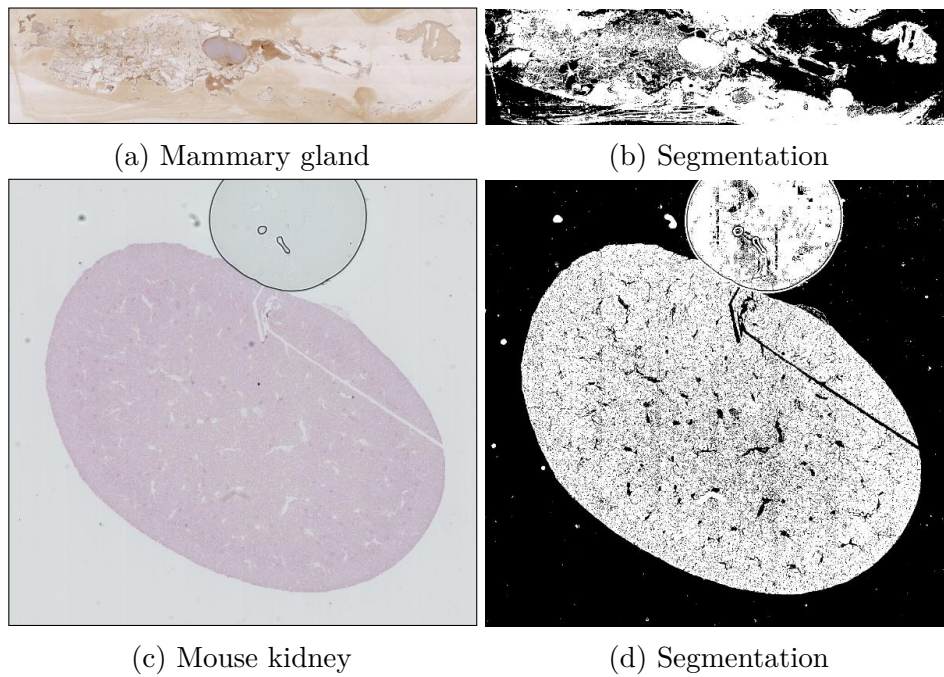


Figure 3.10 – Unsuccessful segmentation of tissue on virtual slides from ANHIR registration challenge.

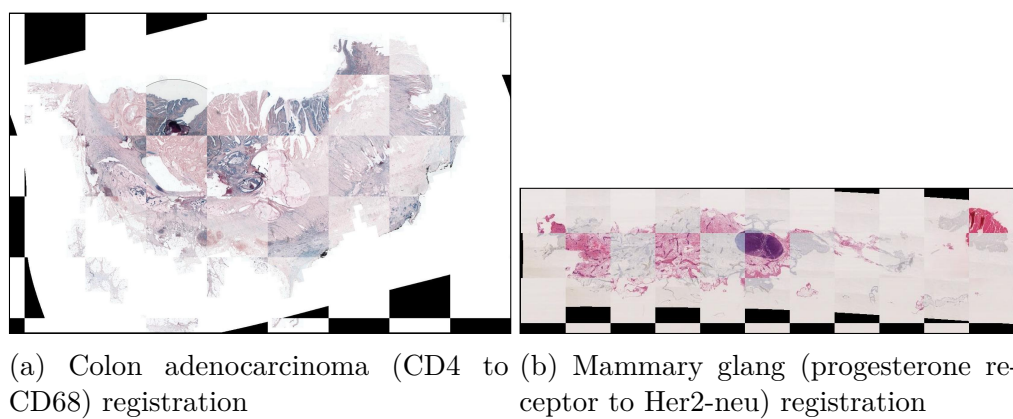


Figure 3.11 – Successful registration examples on virtual slides from ANHIR registration challenge.

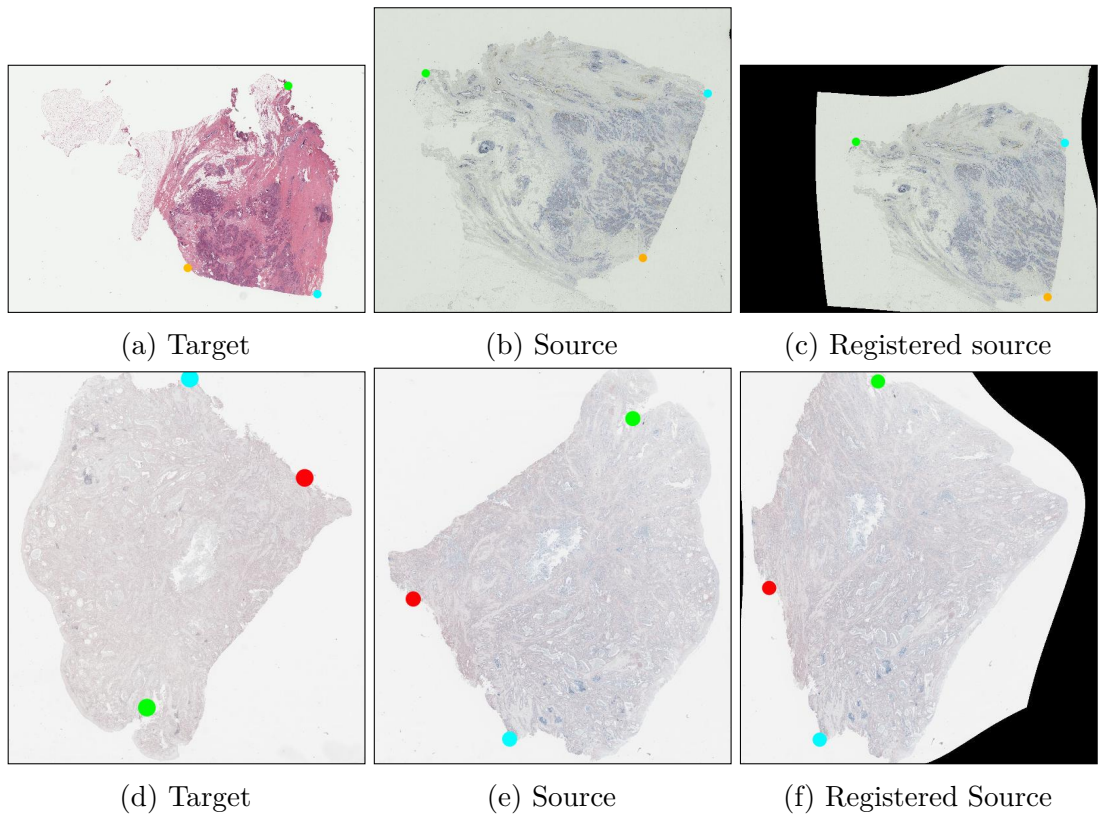


Figure 3.12 – Unsuccessful registration examples on virtual slides from ANHIR registration challenge. (top) Breast tissue (H&E to Her2-neu). (bottom) Gastric adenocarcinoma tissue (CD4 to CD68). Color dots are used to make evident the registration error

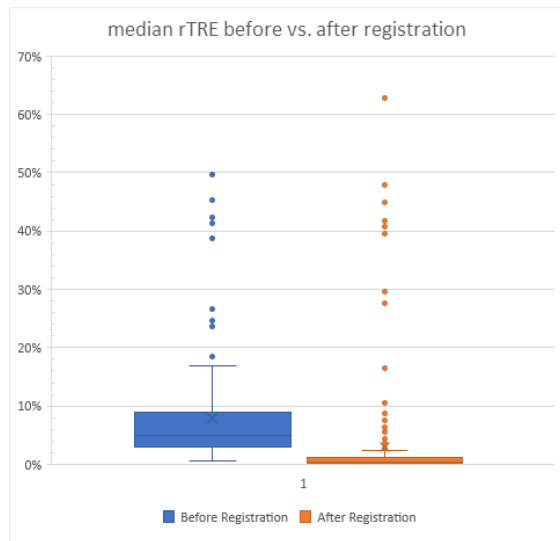


Figure 3.13 – Histogram of the Median-rTRE measured on the training data ($N_{pairs} = 230$) before and after our registration process is applied.

by choosing the data that should be registered through a pre-processing step which simplified the subsequent registration.

In addition to evaluating our solution with large histological images, we also tried our method using different imaging techniques such as Masson’s trichrome and Second Harmonic Generation on kidney tissue with successful subjective results, which shows the extent of application of our solution. (See Figure 3.14)

3.4 Conclusion and perspective

In this chapter we presented a method that allows to register large histological images having multiple staining techniques with competitive results. We also explained the method to process large images in the case of image registration by holding the transformation model when creating the output image by tiles. Taking advantage of the tissue contour we managed to find the appropriate transformation to register pairs of images. And we extended our work to not only multiple staining techniques, but also multiple acquisition methods. This work has been published in the IEEE International Symposium on Biomedical Imaging in 2017 [94]. Although our method is competitive with other modern registration methods, this task is still challenging because of the variability of positioning of the tissue during tissue preparation and image acquisition. This task is even harder when slides are damaged or get deteriorated over time. A possible solution could be the use of mutual information to associate sets of pixel intensities on source and target images. However, this implies a pre-alignment step that, again is difficult due to the variability of the images. Taking a look at the challenge winner method [93], the pre-alignment using normalized gradient fields (NGF) distances looks promising as it takes into account morphological structures rather than their intensities. Which

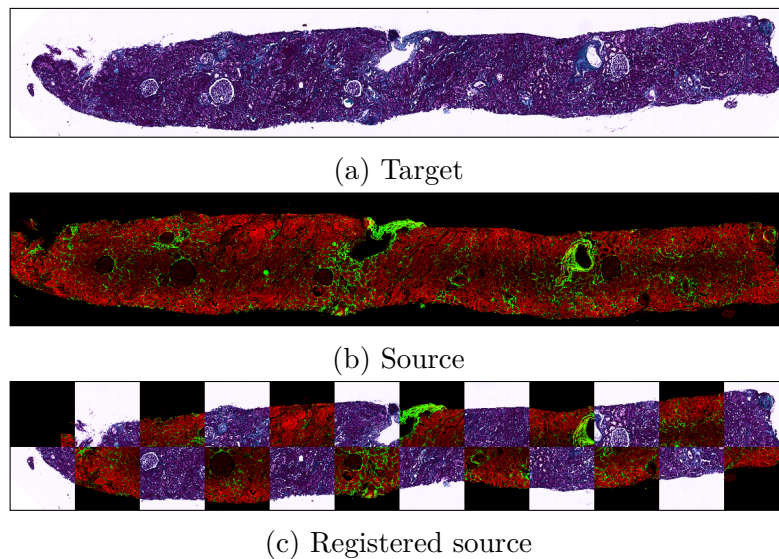


Figure 3.14 – Successful registration example (bottom) on Masson’s trichrome (top) and Second Harmonic Generation (middle) on kidney tissue.

is very well adapted to our problem (needing a pre-alignment anyways). Finally, a further step is the 3-D slide stack generation, which introduces a new challenge since not only slides have to be registered one to one, but also the general shape of structures must be sustained along the Z axis. This means that the registration would need to be constrained to both the shape of structures with respect to adjacent slides, and the whole stack to prevent spherical structures from becoming cylinders (an example of this effect can be seen in [95]).

Chapter 4

Conclusions of Part I

All along this first part we have made strong emphasis on creating an integrated framework that allows for handling large images including but not limited to those acquired for histology. We first started by presenting two technologies that integrated can provide a powerful tool for transferring remote large images, as well as perform analysis locally and store results remotely. I presented the architecture of the solution and the building blocks of Icytomine, alongside the means to perform automated large image analysis. In addition, I presented an image registration solution targeting whole-slide images acquired with slide scanners. In this subject I presented the means to obtain the contours of the tissue to register, as well as the principles to perform the registration and the subsequent transformation, taking special care to respect memory limits when dealing with images of large size.

It is clear at this point that a lot of work was put into making this tools that aimed to improve the productivity of image analysts. However, there is still a lot of space for improvement. First, the registration procedure presented has yet to be integrated into Icytomine to provide a seamless experience for users. Additionally, more extending blocks need to be developed for Icytomine to further improve analysts productivity. For example, tools to identify from remote images (possibly interactively) the target areas to explore or perform analysis. Concerning image registration, a volume reconstruction by stacking registered slides is yet to be developed and further research needs to be done on this field to preserve geometric features of the tissue [96, 95]. Furthermore, newer distance measures (mutual information or normalized gradient fields [93]) could be integrated to our solution to improve the image registration precision.

Now we change our focus to provide tools to improve analysis of annotations. We take advantage of their vectorial nature to develop methods for morphology analysis using only straight lines, which are convenient and efficient for spatial analysis and storage of big data.

Part II

Vectorial shape morphology

Chapter 5

Introduction: Describing shapes as vectorial data

In image analysis one of the fundamental tasks is shape description (morphology), which is one of the steps involved in shape recognition and classification methods. Most applications of morphology are found in GIS solutions (Geographic Information Systems) and OCR (Optical Character Recognition) systems. Some examples for GIS solutions are urban morphology to analyze population concentration based on shape analysis [97], or in [98], where authors assess ecological conditions of european rivers (e.g. amount of gravel and sand bars, free flowing sections), based on the morphology of their boundaries. Examples of OCR solutions can be found in [99] where line reconstruction techniques are applied to detect tables on scanned images of degraded documents.

Morphology has also been successfully used in biology to detect and quantify different structures in cells and tissue. For example, in [100] authors use the medial axis transform of segmented filopodia to quantify the filament characteristics of the cells (quantity, length and direction). In addition, morphology can make part of other analyses such as spatial analysis. For example, in [21] authors use bitmap based morphological operators to perform segmentation of stained cells from nuclei dilations on patches of virtual slides. In spatial analysis it is usual for analysts to use point pattern spatial statistics, in which point clustering is described using functions derived from that of Ripley's K function, proposed in [101]. More recently, Lagache et al. in [102] have developed SODA (Statistical Object Distance Analysis) to perform analysis of aggregated molecules in micro- or nanoscopy images to understand protein organization. However, to this day this principle can only be applied to perform point-to-point statistics, very few studies have been performed to apply shape-to-point or shape-to-shape statistics. Few works have tried to address shape to shape spatial relationship analysis, in [103] spatial relation descriptors are developed and presented using entwining and interlacing histograms.

As seen in the previous part, annotations in large image analysis systems are stored as polygonal structures in order to economize disk space and loading times

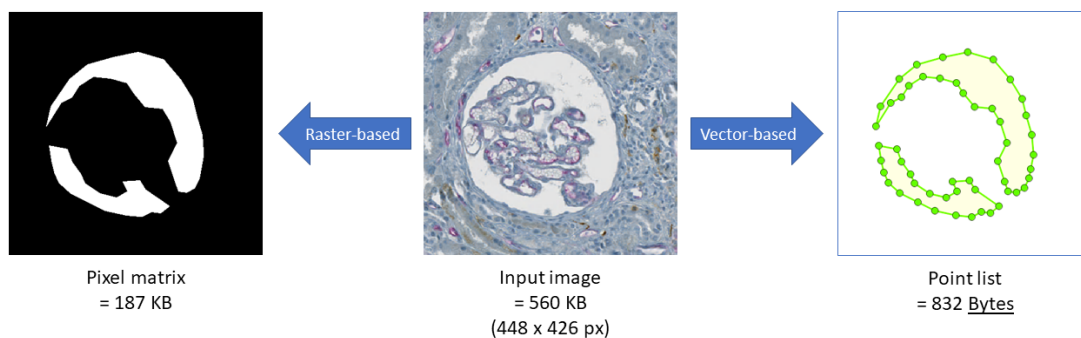


Figure 5.1 – Comparison between raster-based and vector-based annotations. A considerable reduction in file size is achieved when using vectorial annotations.

when storing and manipulating them. In fact, classic approaches for applying morphological operations on shapes is not well adapted for large images (where the size of the image goes beyond 100K by 100K pixels). In this part we focus on presenting methods to perform these operations directly on polygonal data, avoiding any raster-based operation to improve the performance of calculations (See difference between raster-based and vector-based annotations in Figure 5.1). This chapter introduces the basic concepts of morphology and the existing methods related to morphology which are in their great majority raster-based. These concepts allow to understand the rest of this part which is dedicated to vectorial annotations morphology.

One of the most common ways for describing a shape is the *morphological skeleton*. In its most basic interpretation it is the inner-most part of a given shape allowing to understand its geometrical structure. This descriptor has been proved really useful for several applications since its introduction in the 1960s. Among its applications we find shape matching [104, 105], shape decomposition [106, 107, 108, 109, 110, 111], data compression [112, 113], handwriting recognition [114, 115, 116], tracking and analyzing blood vessels [117], pulmonary airways volume estimation [118], among many others.

To understand better what a skeleton is and what is not, let us formally define it.

Definition 5.0.1. Morphological skeleton [119] The reduction of a 2-D geometrical object to a set of 1-D curves, and the reduction of a 3-D object to either a set of 2-D surfaces and 1-D curves or a set of only curves, can be considered a skeleton if it has the following properties:

1. It should have the same topology as the input object, i.e., the same number of components and holes (tunnels in 3-D);
2. It should be thin;
3. It should be centered within the object;

4. It should preserve the geometric features of the object, usually meaning that the skeleton should have components corresponding to the various parts of the object;
5. It should allow complete recovery of the original object.

From this definition, one can also define a closely related concept known as *medial axis*, whose set of points complies with all except the last property (recovery).

A 2-D skeleton consists on a set of points divided in three types:

- *End-points*: With only one neighbor;
- *Normal points*: With two neighbors;
- *Branch-points*: With more than two neighbors.

A 3-D skeleton is more complex as it can contain surfaces, thus points can be classified into many more types. Most existing algorithms store and conceive skeletons as a set of skeletal points in the form of *pixels* (2-D) and *voxels* (3-D).

5.1 Background

Blum first introduced the skeletonization as the medial loci of an object in R^n that forms its skeleton. More precisely, he introduced the medial axis as a grass-fire propagation process (also called wavefront propagation), in which a fire is simultaneously lit on the borders of the shape. This fire then propagates at constant speed at every point of the the fire-front and directing it to the interior of the shape until the whole shape is consumed by the fire. The skeleton is formed at quench points where two non-contiguous fire-fronts collide with each other, which conveniently are the loci of the centers of maximally inscribed balls (MIBs) [120]. Although Blum was the first to describe the morphological skeleton, there are other ways of defining them. For example, Brady and Asada [121] proposed an approach consisting of considering the loci of a cord, each making an equal angle on the tangents at the two points where the cord meets the object boundary. Similarly, Leyton [122] proposed the loci of the centers of the shortest geodesic paths, each connecting two points where a bi-tangent sphere meets the object boundary.

Since its introduction, several authors have proposed different ways to compute the skeletal points of an object: (a) quench points at meeting opposite fire-fronts; (b) the loci of centers of MIBs; and (c) the centers of the enclosed balls touching the object boundary at two (or more) disjoint locations. These three interpretations yield to the same results for objects in R^2 and R^3 (if their boundaries are Jordan curves). However, this is not the case for digital objects. Results usually vary when these three methods are applied on the same digital object. Therefore, classifying these skeletonization algorithms is important to understand their advantages and their purpose. To this end, three categories are established based on their computational strategy to solve the problem.

1. **Geometric approaches:** The object boundary is represented by discrete sets of points in continuous space (either point clouds or polygonal representations). Algorithms in this category often use continuous geometric approaches such as Voronoi diagrams. Voronoi edges are used by these algorithms to locate the symmetry structures or the skeleton of a given object.

Methods widely focus on symmetry and other geometric properties of Blum's medial axis to compute the skeleton of an object. One of the most common approaches in this category is based on the principle of the Voronoi diagram [123, 124, 125, 126, 127]. From a sample of points on the object boundary the Voronoi skeleton is computed by finding the boundaries of the closest areas for each input point. With these intersections only those straight segments that are inside the object are taken into account and are considered as the skeleton of the object.

One of the main challenges for this approach is filtering the large amount of branches that are created on the resulting skeleton. Some approaches try to look at the sensibility of each branch to changes on the shape following the principle that main branches should not be deeply affected by small changes on the object [125].

Schmitt [128] demonstrated that, as the amount of input boundary points augments, the Voronoi skeleton yields to the continuous medial locus, with the exception of the edges generated by neighboring pairs of boundary points.

Voronoi skeletons were also generalized for 3-D polyhedral solids [123, 128, 129, 130, 131]. Amenta *et al.* [123] characterized inner and outer Voronoi balls for a set of boundary sample points to reconstruct an approximation of a polyhedral boundary and to compute its Voronoi skeleton. GPU efficient algorithms were developed by Jalba *et al.* [132] for extracting surface and curve skeletons from large meshes. Bucksch *et al.* [133] presented a graph-based approach to extract the skeletal tree from point clouds using collapsing and merging procedures in octree-graphs.

2. **Curve propagation approaches:** Object boundaries are represented as continuous curves or digital approximations of them. Algorithms are based on the principle of continuous curve evolution of the object boundary, where the symmetry structures or the skeleton are formed at singularity locations, specifically at collision points of evolving curves. In this case, and in contrast with digital approaches, curve evolution is computed using partial differential equations [120]. An example of this is the usage of active contours by Leymarie and Levine [134], detecting skeletal branching at curvature feature singularities. In other method [135], Hamiltonian formulated curve evolutions are presented, computing outward flux of the underlying vector field of the system. Skeletons are located on the flux field singularities. Other methods ([136, 137]) rely on general fields for front propagation to smooth resulting skeletons at the expense of topology correctness as they may have

breaks not corresponding to object breaks. On the same line, force vector fields are used for curve skeletonization in [111] and gradient vector flow is used in [138]

3. **Digital approaches:** Objects are represented by a set of pixels/voxels in a digital space Z^n . Digital morphological operators (erosions and dilations) are used to find singularities on a digital distance transform field to locate skeletal structures. Such algorithms use explicit criteria for topology preservation. Since this is the most popular and straightforward approach since the early 1990s, a lot of literature is available on this topic. For example, fully predicate-kernel-based iterative algorithms are available in [139, 140, 141]. Furthermore, boundary peeling methods with topological and geometrical constraints are presented in [142, 143, 144]. Finally, methods using distance transforms to guide the propagation can be found in [145, 146, 147]
4. **Other approaches:** Although many of the existing methods are covered by these categories, there are some works that do not fit into any of these categories. For example, zoom-invariant cores [148], where medial cores are produced at multiple resolution levels and only the most representative branches are kept. In addition, in [149] pixels grid are represented in hexagonal format for 2-D images and their corresponding skeletonization is presented in [150]. Another example are fuzzy skeletons, which are created from fuzzy images instead of binary images. Pal and Ronsenfeld [151] introduced the basis for this field by using the concept of fuzzy disk and created fuzzy distance maps to detect centers of fuzzy maximal distances in [152].

The method that is going to be presented in the following chapters can be considered as a hybrid approach. It can be a geometric approach, taking into account a point discretization of a continuous input shape and computing the skeleton from these points. But it can also be viewed as a curve propagation approach, because the skeleton is built by performing a front propagation and locating singularities at collisions of the evolving front. The result of this method is what is known as a *straight skeleton*, which is a version of the morphological skeleton made entirely from straight line segments. The definition of this structure and its construction is described in the Chapter 6.

Chapter 6

Introduction to straight skeletons

Although there are several methods that can compute morphological skeletons from shapes using different approaches, most of them are limited or separated from other tools such as the application of morphological operators. In this chapter straight skeletons are presented as an integrated tool to compute morphological skeletons and morphological operations from polygonal data. Some insights about dealing with degenerate simple polygons will be then introduced in Chapter 7. This kind of skeletons will be further used as base for our main contribution, the refined straight skeleton (Chapter 8).

6.1 Notation

This and the following chapters describe several geometrical concepts using the notation described in this section. Please refer to this list for any clarification needed on the notation.

- \mathbb{R}^n : The n -dimensional euclidean space.
- \mathbb{Z}^n : The discrete n -dimensional euclidean space. Mostly used for describing digital images.
- G : A planar straight line graph that accepts straight segments described by a set of points and a set of edges connecting them.
- P : A Polygon composed of an ordered set of vertices joined by straight segments. In contrast with a planar straight line graph, a polygon is a closed set of straight segments. Hence, polygons are generalized by planar straight line graphs.
- $\mathcal{W}_G(t)$: The wavefront propagating from G at instant t .
- $\mathcal{S}(G)$: The straight skeleton of the polygon G .
- $\mathcal{M}(P)$: The motorcycle graph induced by a non-degenerate polygon P .

- $\mathcal{W}_G^*(t)$: The wavefront propagation $\mathcal{W}_G(t)$ extended with $\mathcal{M}(G)$.
- e_i : The i -th edge of a given planar graph.
- v_i : The i -th vertex describing a given planar graph.
- m_i : The i -th motorcycle in $\mathcal{M}(P)$ moving at constant speed and direction, starting at a specific position.
- $f(e)$: The skeleton face adjacent to the wavefront edge e .
- $e(t)$: The wavefront edge at instant t .
- $\overline{e(t)}$: The supporting line of $e(t)$.
- e_i^S : The i -th edge of the straight skeleton S .
- e_i^A : The i -th edge of the straight medial axis A .
- $\delta_B(G)$: The morphological dilation of the graph G using the structuring element B . In the text only circumferences are used as structuring elements.
- $\epsilon_B(G)$: The morphological erosion of the graph G using the structuring element B .
- $\gamma_B(G)$: The morphological opening of the graph G using B as structuring element.
- $\phi_B(G)$: The morphological closing of the graph G using B as structuring element.

6.2 From continuous to discrete skeletons

One of the main issues when using curve propagation and digital approaches for computing the morphological skeletons is the representation of the propagating front. The classic way of representing them is to use a bit-mask that annotates pixels (positions) that have been already visited during the propagation process. This way of representing propagation fronts becomes inefficient as images grow in size. To address this issue Aichholzer *et al.* [153] first presented a different type of skeleton created from shapes represented as polygons. He called this structure the straight skeleton, a skeleton that is made exclusively out of straight segments. Using polygons as the representation of propagation fronts allows to keep a summarized information of the propagation (using control points of the front and their propagation directions) instead of holding a large matrix at every step of the propagation which can be limiting in terms of memory and performance. The formal definition of a straight skeleton is as follows.

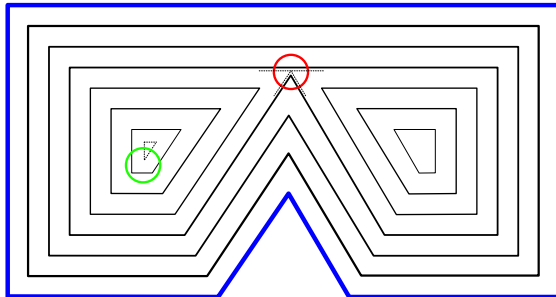


Figure 6.1 – Split and edge events on a straight-line front propagation starting from a simple polygon.

Definition 6.2.1. Straight skeleton A straight skeleton is defined by a front propagation process. Consider a simple polygon P . Every edge of P sends out a parallel wavefront edge that moves with unit speed to the interior of P . The wavefront edges of two adjacent polygons edges e_1, e_2 are joined by a wavefront vertex that moves along the angular bisector of e_1 and e_2 . We interpret the wavefront at any time t as a 2-regular graph that is denoted by $\mathcal{W}_P(t)$. During the propagation, topological changes occur in the wavefront, which are classified into two types of events (See example in Figure 6.1):

- An edge event occurs when a wavefront edge shrinks to length zero and vanishes.
- A split event occurs when a reflex wavefront vertex meets a wavefront edge and causes a split of the wavefront into parts. A wavefront vertex is called reflex if the angle of the incident wavefront edges on the propagation side is larger than π .

If multiple split events occur at the same time and location (i.e. when multiple reflex wavefront vertices meet) then we call this a multi-split event. Multi-split events play a prominent role in the theory of straight skeletons.

The straight skeleton $\mathcal{S}(P)$ of P is defined as the set of loci that are traced out by the wavefront vertices.

Aichholzer generalized this definition to a planar straight line graph G . First, the wavefront propagation is extended to the entire plane as every edge e of G sends out a wavefront copy on both sides of the edge. Second, at terminal vertices v of G the wavefront forms a rectangular cap since an additional wavefront edge is sent out perpendicular to the single incident edge of v . Third, if multiple edges of G meet in a vertex v then only the wavefront edges of neighboring edges of G in the cyclic incidence order of v are joined by a wavefront vertex. The straight skeleton $\mathcal{S}(G)$ is again defined as the set of loci traced out by wavefront vertices of $\mathcal{W}_G(t)$. When $\mathcal{S}(G)$ is laid over G , \mathbb{R}^2 is divided into polygonal faces. Each face $f(e)$ is swept out by a single wavefront edge e . During the propagation of wavefront $e(t)$ is considered the set of straight-line segments covered by the wavefront edge e at

time t . Furthermore, the supporting line of $e(t)$ is denoted as $\overline{e(t)}$. Each edge of $\mathcal{S}(G)$ called arc lies on the boundary of two faces, say $f(e)$ and $f(e')$, and therefore also on the bisector of supporting lines $\overline{e(t)}$ and $\overline{e'(t)}$. Thus, the straight skeleton $\mathcal{S}(G)$ is made entirely by straight line segments of linear size with respect to the amount of vertices in G as it comprises $2n - t + 2$ vertices, where t denotes the number terminal vertices in G .

6.3 Existing methods for building straight skeletons

As an initial "brute-force" approach one could take the chronological order of all edge and split events. While edge events are rather simple to handle (all edge events are stored in a priority queue with collapse time as priority criteria), split events require significantly more effort. In order to find the next split event one would have to test every reflex wavefront vertex against every wavefront edge for a potential hit. This would require $O(n^3 \log n)$ time in total, with n the number of vertices representing the initial polygons.

Aichholzer et al. [153] proposed a method to handle all split events in $O(n^2 \log n)$ on simple polygons. Since then straight skeleton time complexities started using r to express the amount of reflex vertices on the input polygon. Thus Aichholzer's method would be expressed now as $O(nr \log n)$, with r the number of reflex vertices in the initial polygon.

Aichholzer and Aurenhammer [154] presented a method based on kinetic triangulations. The method presented a simulation of the propagation of the initial edges by computing a triangulation of edges that have not yet collapsed at a given time t . Then, during the propagation, there could be *flip events* that would require re-triangulating part of the graph in order to keep an optimal structure. Unfortunately just handling these flip events could take up to $O(n^2 r)$ and for the complete skeleton calculation it would take $O(n^2 r \log n)$ in time.

Eppstein And Erickson [155] were the first algorithm proposing a sub-quadratic solution, computing the skeleton using a hierarchical-style algorithm on closest pairs of points in the wavefront, achieving a complexity in time of $O(n^{8/11+\epsilon} r^{9/11+\epsilon})$. Unfortunately the algorithm turns out to be too complex to be appropriately implemented.

Cheng and Vigneron [156] presented a method improving in computation time and accepting holes on input polygons. They based their algorithm on the usage of on randomized motorcycle graphs, achieving a performance in time of $O(n \log^2 n + r \sqrt{r} \log r)$. A complication of this method is the computation of motorcycle graphs using $1/\sqrt{r}$ -cuttings which made it so hard to implement that there is no known implementation of their algorithm.

Felkel and Obdržálek [157] proposed an simple algorithm to compute straight skeletons that would run in $O(nr + n \log n)$ time. However, their algorithm turned out to be flawed. Cacciola [158] then published an algorithm for the CGAL library

6.3. EXISTING METHODS FOR BUILDING STRAIGHT SKELETONS

based on Felkel's algorithm modified in order to run correctly. Unfortunately, no details of the algorithm implementation have been published other than its worst-case runtime complexity which is $O(n^2 \log n)$.

Huber et al. [159] presented a method to compute the straight skeleton of a simple polygon based on its motorcycle graph. Their method has a complexity of $O(n \log n)$ in time and $O(n)$ in space. We base our work on the same principles as Huber's method, approximating the continuous skeleton by refining the motorcycle graph on reflex vertices. The next chapter presents Huber's method as an introduction for the refinement proposed in Chapter 7.

Chapter 7

Straight skeletons based on motorcycle graphs

As Huber and Held [159] explained, using polygons triangulations could seem a good idea as the sub-divisions of the polygon become convex and thus their skeleton can be computed fast. However the computing flip events (when a vertex of the wavefront changes of triangle during the propagation) takes $O(n^3 \log n)$ on its worst-case, making it a disadvantage to use triangulations. They also noted that even if there are no flip events the worst-case scenario would take $O(n \log n)$.

Nonetheless, there is an interest of finding optimal convex sub-divisions of input polygons to compute their skeleton as the subsequent calculation of the straight skeleton becomes faster. More precisely, dividing input polygons from their reflex vertices is ideal because all split events can only happen on reflex vertices when propagating the front (see Figure 7.1).

In order to compute these subdivisions Huber and Held used the principle of a motorcycle graph introduced by Eppstein and Erickson [155]. The main advantage of using this approach is the time it takes to be computed. This type of graph can be obtained theoretically in time $O(r^2 \log n)$, but in practice most data-sets are computed on average in $O(r \log n)$, given that the vertices of the polygon are uniformly distributed. This type of graph and its calculation is introduced in the following section.

7.1 Generalization of motorcycle graphs

Definition 7.1.1. Motorcycle graph A motorcycle moves at a constant speed along a straight line, starting at a specific position. Given n motorcycles m_1, \dots, m_n , each starting at position pos_i and moving with constant velocity vel_i , with $1 \leq i \leq n$. Each motorcycle moves leaving a trace of its displacement until either of the following things happen, in which case it is called “crashed”:

- It crashes with another motorcycle at the same time and position.
- It crashes with the trace left by another motorcycle.

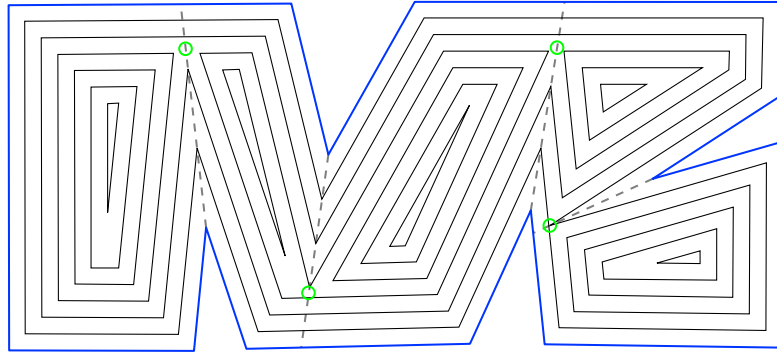


Figure 7.1 – When the wavefront propagates, split events only happen on the trajectory of reflex vertices. These trajectories can be computed on a motorcycle graph

A motorcycle is called “escaped” if it never stops. The motorcycle graph $\mathcal{M}(m_1, \dots, m_n)$ is defined as the set of traces left by motorcycles m_1, \dots, m_n after infinite time. From this, the track of m_i is an infinite ray defined as $pos_i + t * vel_i$, for any time $t \geq 0$.

Cheng and Vigneron [156] found an interesting geometric relationship between non-degenerate polygons and their motorcycle graphs, which actually is the key for building straight skeletons from non-degenerate polygons. But first let us introduce the motorcycle graph induced by a non-degenerate simple polygon.

Definition 7.1.2. Motorcycle graph induced by a non-degenerate simple polygon

Consider the motorcycle graph resulting from launching a motorcycle from every reflex vertex v of a non-degenerate simple polygon P with a velocity equal to the wavefront vertex originated from v . In addition, let’s consider the edges of P as solid walls which motorcycles can crash against. This motorcycle graph denoted by $\mathcal{M}(P)$, obtained after processing all crash events is defined as the motorcycle graph induced by the polygon P .

From this definition it becomes possible to see that for any non-degenerate simple polygon P , the arcs of $\mathcal{S}(P)$ that are incident to a reflex vertex of P are covered by $\mathcal{M}(P)$ (a visual example is available in Figure 7.2a).

Although this definition looks good for the construction of straight skeletons, it is not suitable when dealing with degenerate simple polygons or planar straight line graphs where edges can have particular configurations such as terminal edges (see Figure 7.2b), thus introducing a problem when computing $\mathcal{S}(P)$ from $\mathcal{M}(P)$ because $\mathcal{M}(P)$ does not cover the initial reflex vertices of $\mathcal{S}(P)$.

To solve this issue, Huber et al. [159] proposed to add two reflex vertices when this scenario occurs. In Chapter 8 we propose a different approach to obtain refined straight skeletons that are closer to the continuous morphological skeleton of a shape.

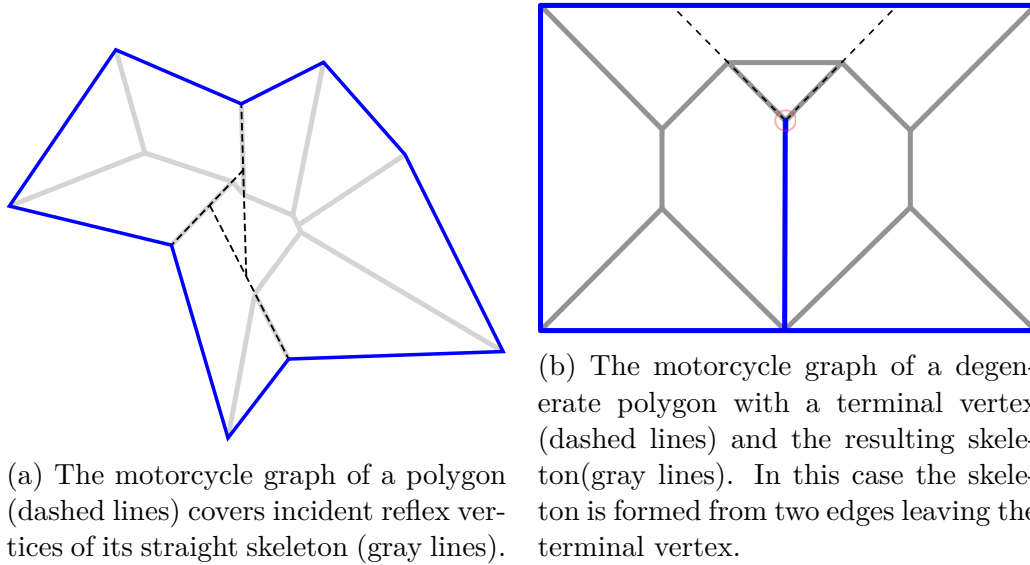


Figure 7.2 – Examples of motorcycle graph on simple non-convex polygons. It is possible to obtain a motorcycle graph from degenerate simple polygons, where two or more edges intersect on the same line they are defined (7.2b).

Another consideration to be handled when creating motorcycle graphs as means for the construction of straight skeletons are simultaneous motorcycle crashes. In these cases, and with the goal of creating convex sub-regions on the polygon, new motorcycles are launched. For this purpose we state the case of a motorcycle m with a movement defined by wavefront edges e and e' , such that the position of m at time t is the intersection $e(t) \cap e'(t)$. When m hits one or multiple motorcycles at the same time t' and position p' we consider the local disk D around p' . D is tessellated into slices by motorcycle traces established up to the simulation time t' . If one of the slices forms a reflex angle then new motorcycles are launched in the following way.

Let's denote m_1, \dots, m_k the motorcycles that crash at p' such that their traces appear counter-clockwise around p' , and traces of m_1 and m_k form a reflex angle on disk D . We distinguish two cases:

- The left defining edge of m_1 and the right defining edge of m_k span a convex angle. In this case a motorcycle m is launched following the bisector angle of the left defining edge of m_1 and the right defining edge of m_k (see Figure 7.3a).
- The left defining edge e_{1l} of m_1 and the right defining e_{kr} edge of m_k span a reflex angle. Then a new motorcycle is launched in the direction of the bisector using e_{1l} and e_{kr} as defining edges (see Figure 7.3b).

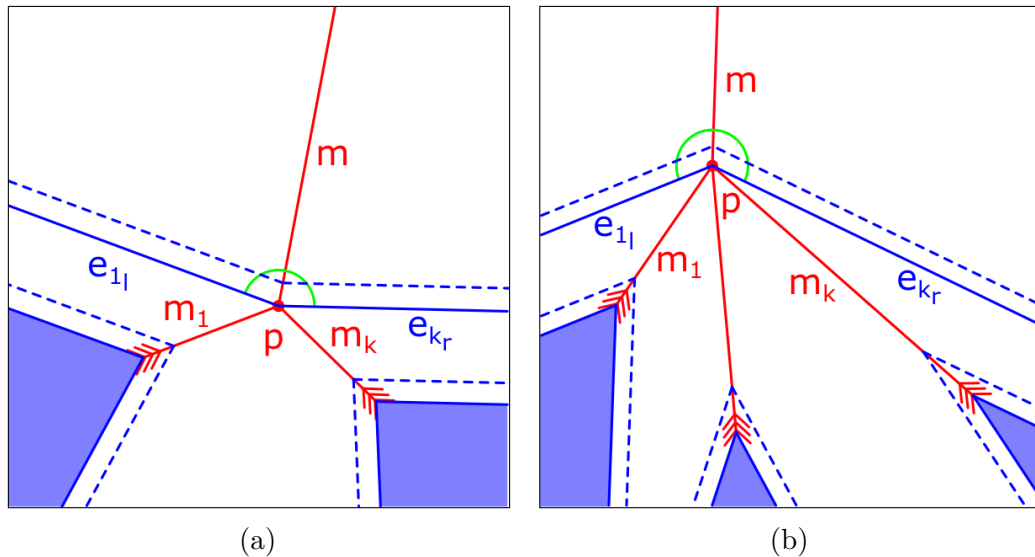


Figure 7.3 – Simultaneous motorcycle crash handling. (a) Convex motorcycle collisions create a single motorcycle on the bisector angle of the motorcycle defining edges at the collision position. (b) Reflex motorcycle collisions can create multiple motorcycles depending on the defining wavefront edges at the collision position and the maximum amount of motorcycles launched at a reflex vertex.

7.2 Wavefront propagation using motorcycle graphs

Huber and Held demonstrated that it was possible to perform flip-event-free Steiner triangulations when using motorcycle traces as edges on the triangulation [159] in order to obtain a straight skeleton. In fact, they showed that no triangulation was necessary to create the straight skeleton, only the motorcycle graph is needed. To do this they introduced the concept of an extended wavefront graph \mathcal{W}_G^* . This wavefront provides the straight skeleton as a result of simulating its propagation.

Definition 7.2.1. Extended wavefront The extended wavefront $\mathcal{W}_G^*(t)$ is defined by the overlay of $\mathcal{W}_G(t)$ and $\mathcal{M}(G) \cap \bigcup_{t' \geq t} \mathcal{W}_G(t')$.

This means that only the part of $\mathcal{M}(G)$ that has not been already swept by $\mathcal{W}_G(t)$ is overlaid on top of $\mathcal{W}_G(t)$.

Since the $\mathcal{W}_G(t)$ is motivated by Steiner triangulations, vertices of the extended wavefront that are not present in $\mathcal{W}_G(t)$ are referred to as *Steiner vertices* as they are additional points on the wavefront graph from the motorcycle graph collisions. More specifically, Steiner vertices forming part of the intersection of $\mathcal{W}_G(t)$ and $\mathcal{M}(G)$ are called *moving Steiner vertices*, those vertices that have not been reached by $\mathcal{W}_G(t)$ are called *resting Steiner vertices*. In addition, a resting Steiner vertex that corresponds to a simultaneous crash of multiple motorcycles in $\mathcal{M}(G)$ is called a *multi Steiner vertex* (See Figure 7.4a).

7.2. WAVEFRONT PROPAGATION USING MOTORCYCLE GRAPHS

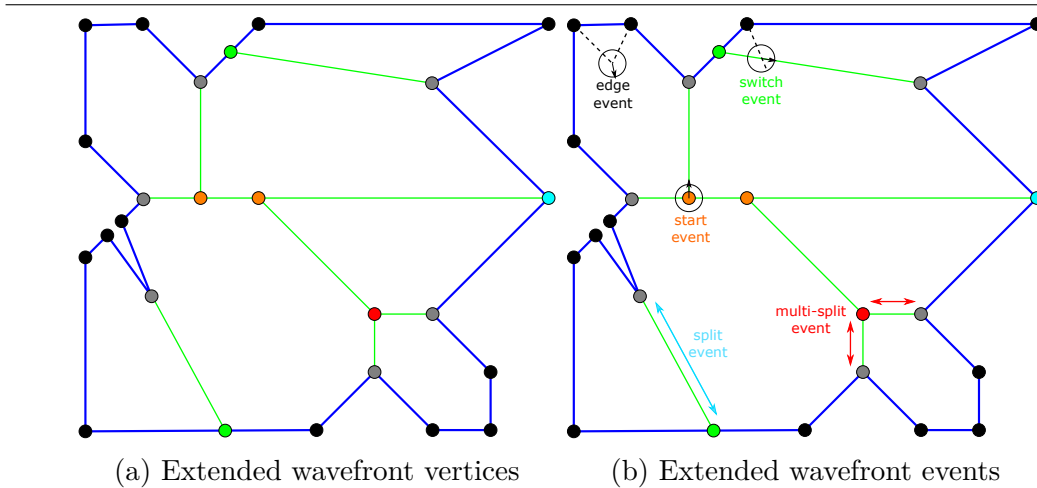


Figure 7.4 – Representation of the extended wavefront $W_G^*(t)$ of the polygon in blue. (a) presents the different types of vertices. *Convex* vertices are represented in black, *reflex* vertices in gray, *moving Steiner* vertices in green, *resting Steiner* vertices in orange, *multi-Steiner* vertices in red, and *multi-convex* vertices in cyan. In (b) topological events on the wavefront are depicted. In black *edge* events, in cyan *split* events, in red *multi-split* events, in orange *start* events, and in green *switch* events.

The useful property presented by this extended wavefront is that during the propagation of $W_G^*(t)$ only adjacent vertices can meet. In fact, this is the central part of this algorithm to compute straight skeletons as it ensures its efficiency.

Aichholzer et al stated that the main problem efficient-wise of their algorithm was finding the next split event in the wavefront. Using the extended wavefront, a split event is simply indicated by the collapse of an edge of $W_G^*(t)$ when a reflex wavefront vertex meets a moving Steiner vertex. Thus, this avoid the costly search for the next split event.

The extended wavefront is used in the following section by inspecting edges collapse points on the propagation of $W_G^*(t)$, which is used in turn to build the straight skeleton of G .

7.3 Straight skeletons from wavefront propagations

The algorithm to build the straight skeleton from a polygon is as follows.

```

Result:  $\mathcal{S}(G)$ 
 $\mathcal{W}_G^* \leftarrow initializeExtendedWavefront(G, \mathcal{M}_G);$ 
 $\mathcal{S}_G \leftarrow initializeStraightSkeleton(\mathcal{W}_G^*);$ 
Fill event priority queue  $Q$  with initial events;
while  $Q$  is not empty do
     $e \leftarrow Q.poll();$ 
    if  $isEventStillValid(e)$  then
         $newPossibleEvents \leftarrow processEvent(e);$ 
        // This will update  $\mathcal{W}_G^*$  and  $\mathcal{S}(G)$ 
         $Q.addAll(newPossibleEvents);$ 
    else
         $continue;$ 
    end
end

```

Algorithm 1: Event handling on the wavefront propagation when building the straight skeleton $\mathcal{S}(G)$.

Following the algorithm, initial events are first added to the queue from every edge on $\mathcal{W}_G^*(0)$ with the collapse time as priority, if it is finite and positive. Then, events are handled by applying the topological changes described below to $\mathcal{W}_G^*(t)$ and adding to the queue new future events that might happen after modifying \mathcal{W}_G^* . Events are handled only if the event edge is still present on the graph at the time the event is supposed to happen. Otherwise the event is considered stale and consequently ignored.

7.3.1 Topological events on the front propagation

The topological changes can be classified as follows and are shown in Figure 7.4b:

- **Edge event:** Two convex vertices u and v meet each other. In this case a skeleton vertex is added at meeting position and it is connected by an edge to skeleton vertices of u and v . Then, u and v are merged to create a new convex vertex w . A special case has to be accounted for when the angle of the incident edges of w span a null angle and both incident edges have the same end vertex w' (i.e. 2 fronts collapsed). In this case, a skeleton edge is added to connect skeleton vertices of w and w' and the wavefront edge linking w and w' is then removed.
- **Split event:** A reflex vertex u meets a moving Steiner vertex v and they are moving towards each other.

For this event a skeleton vertex is added at meeting point and an edge is added to connect the skeleton vertex of u .

Then, the left side of the wavefront edge $e = (u, v)$ is considered. If this side collapsed, a skeleton edge is added from the created skeleton vertex and the skeleton vertex of the collapsing edge opposite vertex. Otherwise a new wavefront vertex is added connecting edges incident to u and v and left of e . The same procedure is applied for the right side of e .

- **Multi-split event:** Multiple reflex vertices u_1, \dots, u_k arrive simultaneously at a multi-Steiner vertex u . Although this event is conceptually similar to the single-split event, a multi-split event handles multiple splits around the same vertex.

First, a skeleton vertex is added at u and it is connected to the skeleton vertices associated to front vertices u_1, \dots, u_k .

Second, taking u_1, \dots, u_k organized clockwise around u , consider all consecutive pairs $(u_i, u_{1+i \bmod k})$ with $1 \leq i \leq k$. With e_i denoting the edge (u, u_i) and e_{i+1} denoting the edge $(u, u_{1+i \bmod k})$, the wavefront is patched at each sector (e_i, e_{i+1}) as follows.

A new vertex v is created patching the counter clockwise edge e_l of e_i at u_i and the clockwise edge e_r of e_{i+1} at $u_{1+i \bmod k}$ together. Additional care should have to be taken when an additional edge e incident to u between e_i and e_{i+1} . If e is exactly on the trajectory of v , then e becomes incident to v . For the rest of cases, e splits e_l , respectively e_r by an additional moving Steiner vertex, depending on whether e lies left or right to the trajectory of v .

- **Switch event:** A convex vertex u meets a reflex or moving Steiner vertex v . Here the convex vertex u is changing of convex face to make part of a neighboring one by jumping over v . For this event, if v is a reflex vertex it becomes moving Steiner and the corresponding skeleton edges are added. Otherwise, if v is a moving Steiner vertex, no skeleton edges are added.
- **Start event:** A reflex or moving Steiner vertex u meets a resting Steiner vertex v . In this case v starts to make part of the moving front. For this v becomes moving Steiner and one of the incident edges of u (not (u, v)) is split by v .
- **Multi-start event:** A moving Steiner vertex u meets a multi-Steiner vertex v . In this case the multi-Steiner vertex becomes part of the moving front. For this, the event is treated similarly to a multi-split event but with $k = 1$ and u_1 is not a reflex vertex (it is u). In addition, no skeleton vertex or edge is added for u . Actually, this event is handled simultaneously with the multi-split event.
- **Other possible events:** When two moving Steiner vertices meet they are just removed with their corresponding edges. Other event should not occur given the convexity of the faces of the extended wavefront $\mathcal{W}_G^*(t)$

7.4 Performance analysis

In order to obtain a full view of the complexity of the straight skeleton method proposed by Huber and Held it is necessary to analyze the complexities of creating the motorcycle graph and the wavefront propagation.

The motorcycle graph presented before in this chapter only launches motorcycles on reflex vertices on the input polygon, so the complexity of motorcycle graph is expressed in terms of r as the amount of reflex vertices in the polygon P . Huber and Held claim that on average the runtime of the motorcycle graph calculation is $\Omega(r \log r)$ in average, and this complexity is kept for most "real-world" examples [160]. It is also worth noting that for convex polygons the motorcycle graph calculation takes $O(1)$ since no reflex vertices are present on the graph.

For the straight skeleton we start by creating the extended wavefront $\mathcal{W}_P^*(t)$. Creating this structure can take as much as $O(n \log n)$ since we need to add both the input polygon and the motorcycles which can be at most $O(n \log n)$.

Since the number of edge events is $O(n)$ and the number of start and split events is $O(r) \subseteq O(n)$, and that for each event the events changed in the priority queue Q is in $O(1)$, the time to process all edges, start and split events take $O(n \log n)$. Additionally, each multi-split event changes $O(i)$ entries in Q , where i is the degree of the multi-Steiner vertex involved in the event, making the processing of these events take $O(n \log n)$ in time. For the case of switch events, they are bound by $O(nr)$ as a convex vertex may meet the same moving Steiner or reflex vertex at most once.

Now this means that the actual worst case scenario would take $O(nr \log rn)$: $r \ll n \implies O(nr \log n) + \epsilon$ in time; however, this case may rarely arrive. In fact, only $O(n)$ of the $O(nr)$ switch cases can actually happen because every time a switch event happens an edge is removed, reducing possible future switch events. With this clarification on the complexity, it becomes clear that the complexity of computing the straight skeleton (including the time of computing the motorcycle graph) is generally $O(n \log n)$.

In addition to the algorithm complexity presented above, Table 7.1 also presents a comparison between some classic bitmap skeletonization methods and our implementation of straight skeletons. In this table it is evident the gain in computation time of straight skeletons with respect to classic bitmap methods.

7.5. STRAIGHT SKELETON PRECISION LIMITS

Image size W x H (px)	Points in polygon	Time Vincent (msecs)	Time Guo (msecs)	Time Zhang (msecs)	Time Straight skeletons (msecs)
250 x 194	52	13	137.239	110.254	10.90
300 x 233	57	17	145.683	143.966	11.43
350 x 272	67	33	155.221	145.637	12.30
400 x 311	74	42	172.495	158.056	17.17
450 x 350	83	63	187.624	172.375	15.57
500 x 389	87	89	202.811	182.217	19.45
550 x 428	95	121	224.427	187.787	18.33
600 x 467	98	143	245.708	190.950	21.64
650 x 506	101	183	274.336	199.867	22.63
700 x 545	108	297	299.136	228.684	23.07
750 x 584	118	463	321.907	248.115	24.40
800 x 623	120	646	386.240	289.133	22.05
850 x 662	127	756	423.903	352.170	24.20
900 x 701	169	971	457.716	361.060	38.89
1000 x 779	166	1452	664.056	431.305	38.17
1500 x 1169	169	5394	1625.040	1102.850	34.33
2000 x 1559	172	10407	3528.130	2361.700	39.84
2500 x 1949	174	17504	6714.950	4563.720	39.52
3000 x 2339	181	28505	11456.600	7525.110	44.73

Table 7.1 – Skeletonization methods computation time comparison. This comparison was made using an binary image at multiple resolution levels, extracting a polygon from the contour of the shape at each resolution level. Tested methods were Vincent [161], Guo [162], Zhang [163], and our implementation of straight skeletons.

7.5 Straight skeleton precision limits

Even though straight skeleton of a set of straight segments can keep the morphological data, they have one particular disadvantage, when reflex angles on vertices come close to 2π , the wavefront propagation speed becomes noticeably faster than the rest of the front. This means that the values of distance from the input shape stored on the skeleton vertices are not really centered with respect to the shape. An example of this can be seen on Figure 8.1. Keeping this in mind and the fact that a reconstruction of a shape from this version of straight skeletons would not allow for a proper reconstruction of the shape, the next section presents an alternative that allows straight skeletons to refine the propagating front to approximate an accurate distance to the shape at every step of the propagation.

Chapter 8

Refined straight skeletons

Although straight skeletons can provide interesting results concerning the morphology of shapes as well as respecting the homotopy with the original shape, when applied to noisy polygons the underlying skeletons become considerably less precise when compared to other methods for skeletonization. More important, in many cases, noisy polygons are the output of the segmentation of tissues on biological images (See Figure 8.1c). This effect happens due to the discretization performed on reflex angles as explained in the following sections. An opportunity arises here to improve the skeleton precision without compromising too much computational time. This chapter presents our method based the straight skeleton proposed by Huber and Held and refined by augmenting the amount of reflex vertices on the wavefront when the propagation is performed, allowing for better precision of the wavefront and a more accurate approximation of the center line of a shape [164]. Additionally, a runtime analysis and an evaluation is shown. Finally, we provide some conclusions, as well as some possible future developments on this subject.

8.1 Refining motorcycle graphs

The amount of motorcycles at reflex vertices is vital to establish the precision of the resulting straight skeleton. As shown in Figure 8.2, the more motorcycles are launched at a terminal vertex, the closer the straight line wavefront will be to the continuous wavefront in orange. In the figure, green vertices (5 motorcycles) evolve the wavefront slower than cyan vertices (2 motorcycles), hence decreasing the error on the final skeleton due to possible early front collisions.

In order to define the amount of motorcycles launched at reflex angles of a polygon, including the cases where angle spanned by the clockwise and counter-clockwise edges is 2π , we propose to establish an adaptive motorcycle definition given a maximum number of motorcycles in the terminal edge case $maxMcs$. According to the spanned angle of incident edges θ_v , the amount of launched motorcycles $numMcs$ is adjusted by $numMcs = \theta_v / maxMcs$. This allows to estimate the maximum error on the straight line wavefront (see Table 8.1) which is computed by Equation 8.1.

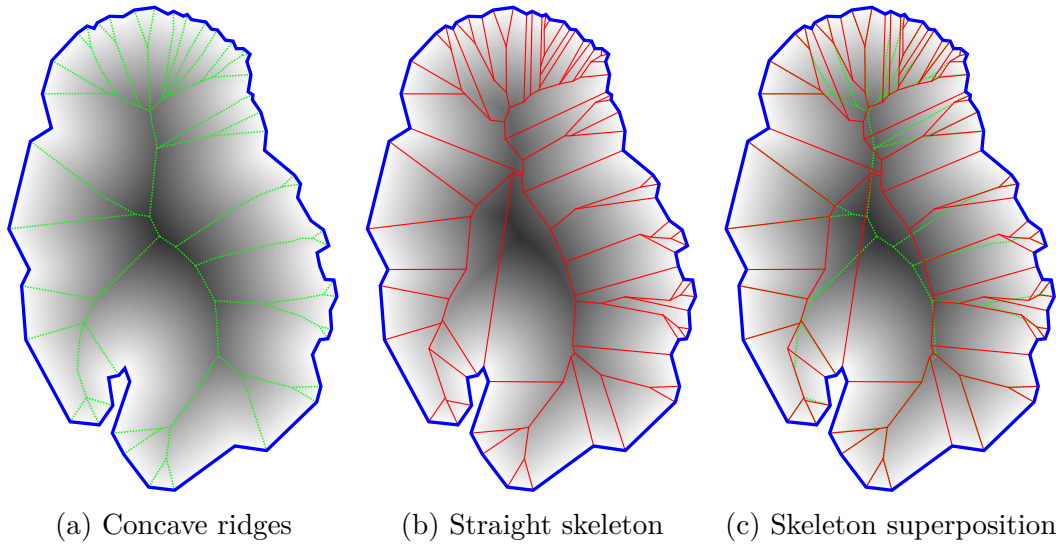


Figure 8.1 – Straight skeleton (in red) on a noisy polygon. In these cases the straight skeleton does not reflect the center line of the input polygon (in dotted green).

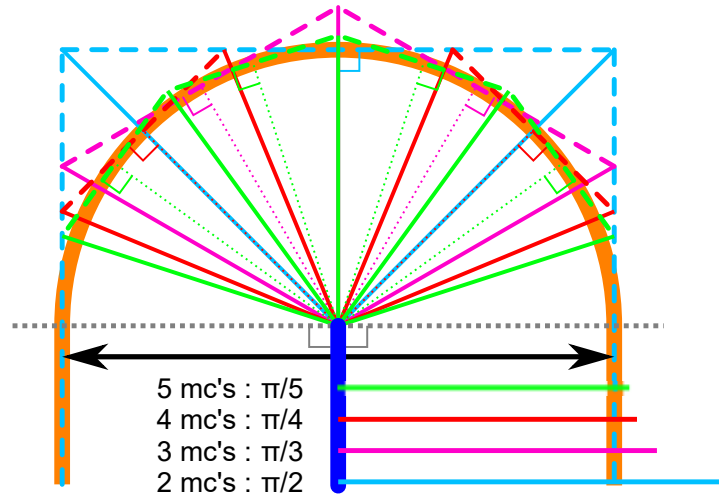


Figure 8.2 – Approximating a straight line wavefront to a continuous one. The amount of motorcycles launched from a reflex vertex has a huge impact on the precision of the wavefront propagation. The lines with the numbers show the length of the rays at unit time of the propagation from the reflex vertex. The more motorcycles launched at reflex vertices, the closer to the continuous wavefront (in orange).

8.1. REFINING MOTORCYCLE GRAPHS

Num mc's	Num divisions	Inter-mc angle	Mc speed	Speed error
2	3	$\pi/2$	1.414	0.414
3	4	$\pi/3$	1.154	0.154
4	5	$\pi/4$	1.082	0.082
5	6	$\pi/5$	1.051	0.051
6	7	$\pi/6$	1.035	0.035
7	8	$\pi/7$	1.025	0.025
8	9	$\pi/8$	1.019	0.019
9	10	$\pi/9$	1.015	0.015
10	11	$\pi/10$	1.012	0.012
11	12	$\pi/11$	1.010	0.010
12	13	$\pi/12$	1.008	0.008
13	14	$\pi/13$	1.007	0.007
14	15	$\pi/14$	1.006	0.006
15	16	$\pi/15$	1.005	0.005

Table 8.1 – Error of the continuous wavefront approximation by straight lines at a terminal vertex. As more motorcycles are launched from the terminal vertex, the approximation error converges to zero.

$$speedError = \sec\left(\frac{\pi}{2 * numMcs}\right) - 1 \quad (8.1)$$

Let's take for example the terminal edge on Figure 7.2b and having at most $maxMcs = 2$ motorcycles on terminal edges. Since the edge is terminal the angle spanned by the inwards normals of the incident edges is $\pi > \pi/maxMcs$, hence two motorcycles are launched from the terminal vertex in the polygon. Now let's take Figure 7.2a at any of the reflex vertices with the same $maxMcs$. Since the angle spanned by the inwards normals of the incident edges is less than $\pi/maxMcs$, there is only one motorcycle launched from the vertex.

Now, in order to determine the direction of each of the vertices launched at a given vertex two parameters are needed. First, the amount of motorcycles that can be launched at the vertex $numMcs$. And second, the angle θ_v spanned by the normals of the clockwise and counter-clockwise edges. Algorithm 2 explains the procedure. As reference for the reader, in Figure 8.2 directing lines are presented as dotted lines and discrete wavefront lines as dashed lines; Thin lines represent launched motorcycles.

Another example of motorcycle creation at reflex vertices is available on Figure 8.3.


```

input :  $v, numMcs, \theta_v, \theta_{cw}$ 
output: Directions for motorcycles at vertex  $v$ 
 $\theta_d \leftarrow \theta_v / numMcs$ ;
for  $i \leftarrow 0$  to  $numMcs + 1$  do
     $directingLines[i] \leftarrow$ 
    Segment( $vertex = v, angle = \theta_{cw} + (i * \theta_d), length = 1$ );
     $unitWavefrontLines[i] \leftarrow$ 
    Line( $position = directingLines[i], direction =$ 
     $directingLines[i].cw().direction()$ );
end
for  $i \leftarrow 0$  to  $numMcs$  do
     $mcVelocity[i] \leftarrow$  LineIntersection( $unitWavefrontLines[i],$ 
     $unitWavefrontLines[i + 1]$ );
end
return  $mcVelocity$ ;
    
```

Algorithm 2: Finding motorcycle directions at a reflex vertex.

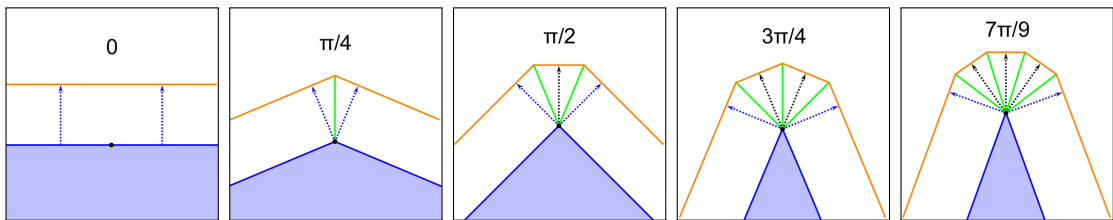


Figure 8.3 – Motorcycle creation on reflex vertices at different angles with a maximum amount of motorcycles set to 4. Motorcycles are represented in green, the wavefront at unit time in orange, defining normals as dotted blue arrows and middle normals as dotted black arrows. Notice that no motorcycle is launched when the angle of the defining edges is π .

8.2 Wavefront propagation using refined motorcycle graphs

In a similar manner that original straight skeletons are obtained, refined straight skeletons are computed by a front propagation process where an extended wavefront is created using the input shape vertices and edges, as well as using the motorcycle graph applying the approach presented in the previous section. The rest of this section presents the steps to compute the wavefront and its propagation that are particularly different from the original method proposed by Huber and Held.

8.2.1 Extending the wavefront with the refined motorcycle graph

In order to take into account the result of the refined motorcycle graph when propagating the wavefront it is necessary to include additional vertices where multiple motorcycles are launched from the same position taking careful attention when connecting them in order to keep the topology coherent when vertices start moving on the front. To do this they are sorted by the angle of the motorcycle trajectory.

8.3 Implementation considerations

The implementation for this method and the underlying performance is highly dependent on the amount of initial reflex wavefront vertices launched at every reflex vertex of the input shape. A comparative table of computation time versus number of vertices on reflex vertices is presented as an indicative of the impact on performance on Table 8.2. The values for this comparison were acquired by computing the refined straight skeleton on a noisy polygon with 320 vertices. A graphic of the comparison is available in Figure 8.4. It is important to remember that the amount of initial wavefront vertices is adjusted not only to the maximum amount set for the skeleton, but also the angle at each reflex vertex. This allows for an optimal performance over detail ratio. This also means that a polygon with a high count of acute reflex angles will take more time when computing its straight skeleton compared with a smoother version of it, because more vertices will be generated when creating the extended wavefront.

8.3.1 Runtime analysis

To define the algorithm computation complexity of refined straight skeletons we base our analysis on the classic straight skeletons. We start by analyzing the motorcycle graph complexity, launching $O(kr)$ motorcycles in the graph where k is the maximal amount of vertices launched at a single reflex vertex and r the

Num. reflex vertices	1	2	3	4	5	6	7	8	9	10	Average (msecs)
2	29	24	25	25	24	31	15	32	31	32	26.8
3	30	31	34	20	31	31	32	33	33	38	31.3
4	47	56	40	39	40	40	47	47	40	47	44.3
5	54	52	50	47	74	55	49	45	51	49	52.6
6	63	56	72	57	65	58	67	62	57	59	61.6
7	63	71	62	74	82	57	70	65	62	63	66.9
8	83	78	83	83	79	79	74	89	77	82	80.7
9	87	90	89	102	91	105	96	95	92	93	94.0
10	109	104	109	131	149	111	112	112	109	108	115.4
11	135	123	129	154	120	125	123	121	117	117	126.4
12	141	132	130	132	126	125	139	138	134	161	135.8
13	176	175	162	170	182	140	144	143	155	143	159.0
14	153	163	169	169	241	169	169	153	162	163	171.1
15	172	187	172	191	191	204	184	171	187	179	183.8
16	202	204	203	269	188	187	203	203	187	198	204.4
17	235	219	219	203	235	218	221	210	213	312	228.5
18	229	233	225	238	237	250	235	235	234	234	235.0
19	253	251	266	359	268	266	250	250	265	250	267.8
20	295	274	283	297	284	328	282	281	297	312	293.3
21	397	324	335	359	325	372	338	320	330	345	344.5
22	328	337	328	334	328	332	362	338	343	329	335.9
23	377	359	375	359	376	387	375	368	426	375	377.7
24	409	374	375	401	412	423	391	395	391	399	397.0
25	408	419	408	407	409	416	437	409	391	417	412.1

Table 8.2 – Comparative performance over maximum number of initial wavefront vertices (skeleton detail) on reflex input vertices.

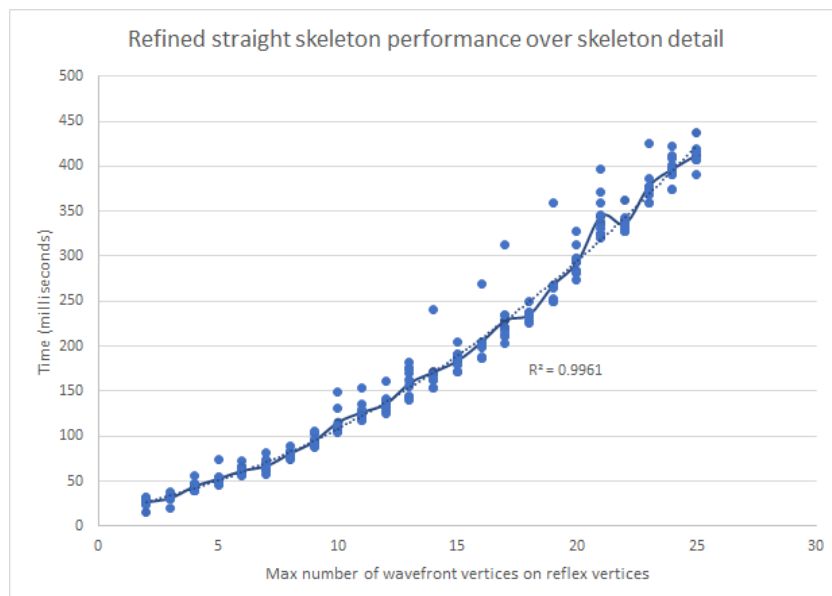


Figure 8.4 – Comparison of refined skeleton detail against computation time.

amount of reflex vertices on the input polygon, the complexity of computing the motorcycle graph is $\Omega(kr \log kr)$ taking the same principle as Huber and Held [160]. Of course, on convex polygons this complexity comes to $O(1)$ because there are no reflex vertices. Next, the wavefront extension takes $O(kn \log kn)$ for n vertices on the input polygon, since the the maximal amount of reflex vertices becomes important when it takes large values. This means that all edge, start and split events take the same time ($O(kn \log kn)$). And for the case of switch events the complexity becomes $O(nkr)$ as convex vertices only meet the same moving Steiner vertex at most once. This leaves us with a worst case scenario of $O(nkr \log nkr)$: $r \ll n \implies O(nkr \log nk)$, which is logical since now the complexity is highly related to the maximal amount of vertices launched from reflex vertices. Despite the worst case scenario including r in its definition, in most general cases the complexity yields to $O(nk \log nk)$ because the amount of reflex vertices remains low with respect to n (See Figure 8.5).

8.3.2 Evaluation

Since there is no ground truth to compare the skeletons produced by the straight skeleton method, it is not possible to evaluate the precision of the shape of the skeleton. However, it is possible to evaluate how the straight skeleton performs in comparison to a continuous version of the Voronoï diagram on segments and arcs [165]. A method proposed by Held et al. which computes curved skeletons. Their implementation computes curved lines that define a precise continuous skeleton, with a computational complexity of $O(n^2)$ in time with respect to the input number of vertices. Our implementation produces skeletons that should approximate this curved skeleton using just straight segments. To evaluate our method,

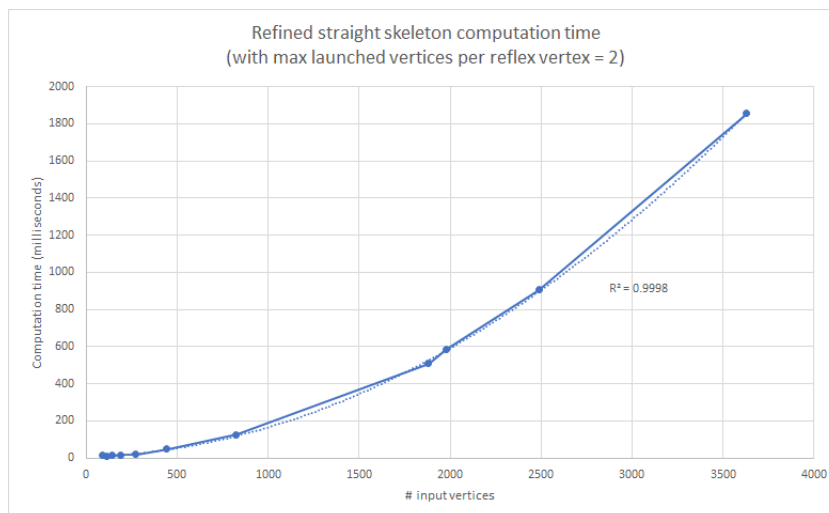


Figure 8.5 – Comparison of refined skeleton number of input vertices against computation time.

we computed the continuous Voronoï diagram of a set of shapes [166] and we then compute a distance map (of size 1300 by 1300 pixels) of the skeleton edges and compute the distance of the edges from our solution to the continuous skeleton using the distance map. The evaluation results are presented in Table 8.3 showing a fast improvement on the precision of straight skeletons within the first added vertices. However, the more vertices are added to reflex angles, the lower is the gain achieved in approximating continuous skeleton.

8.3.3 Conclusions

Having implemented a refined version of straight skeletons allows us to achieve an accurate and efficient discrete approximation of the continuous skeleton of shapes. Using straight skeletons we achieve independence of scale, rotation, translation and symmetry since the method works from the control points of the polygonal input shape. In the following chapters we will concentrate on the operations possible with refined straight skeleton and the opportunities of application on large planar images. As perspective, with the same principle for computing the wavefront propagation on reflex angles of shapes, it could be possible to extend it to single points. In this case a circle would be approximated by a set of straight segments (at least 4 to form a square or more for better approximation) and could provide further shape analysis opportunities, because shape and point skeletons would be processed at once and would provide a complete perspective of the spatial organization. We continue now with the creation of morphological operations and work with them to provide means for shape smoothing and point grouping by creating rings from shape isocontours.

Max Vertices on Reflex Vertices	Mean Distance (pixels)	Max Distance (pixels)	Distance Standard Deviation
1	0.66285	26.0	1.959
2	0.30619	7.0	0.669
3	0.26288	5.7	0.544
4	0.22347	3.0	0.430
5	0.21773	2.0	0.421
6	0.21475	2.0	0.417
7	0.20158	1.4	0.401
8	0.20326	1.0	0.402
9	0.19975	1.0	0.400
10	0.19784	1.0	0.398
11	0.19804	1.0	0.399
12	0.20361	1.0	0.403
20	0.19561	1.0	0.397

Table 8.3 – Evaluating the refined straight skeleton at different values of maximum reflex vertices. This evaluation was performed creating a distance map of the edges in the Voronoï diagram of the shapes on images of size 1294 by 1269 pixels. As visible, from 8 vertices, the maximum distance is fixed to 1 because the resolution of the distance map does not allow to capture more precision. However, the mean distance and standard deviation keep improving.

Chapter 9

Morphological operations on straight skeletons

A popular approach to study the set of elements composing a shape is the application of morphological operators on the shape. These operators are normally used for shape smoothing, structure simplification, among other applications [167]. In this chapter we present the application of morphological operators using out solution for straight skeletons as the base for these operations [168]. The base notion when working with mathematical morphology is presented as follows.

Definition 9.0.1. Morphological operation Let E be a set in \mathbb{R}^d with $d > 0$ usually equal to 2 or 3. In addition, let B be a subset of E called structuring element. If x is an element of E then we denote B_x the set B translated by x .

$$B_x = \{b + x | b \in B\}$$

The structuring element can be translated to any point of E and act as a local probe. If the test on B_x is positive, then x is considered part of the result set $f_B(E)$.

Following this definition, sections below describe the different morphological operators and the underlying application with straight skeletons.

9.1 Shape dilation

Let's first introduce the dilation operator:

Definition 9.1.1. Dilation[H] Let X be a subset of E . A dilation on X with the structuring element B is defined as a Minkowski sum:

$$\delta_B(X) = X \oplus B = \{x + b | b \in B, x \in X\} = \bigcup_{x \in X} B_x$$

A dilation is non-invertible in general. The operation that imitates, at some extent, the inverse of a dilation is the erosion, explained in Section 9.2.

Now, in order to apply the same principle to straight skeletons it is easy to think about using the wavefront that is used to create the straight skeleton. For this, let's set the structuring element B as a circle with radius r . Then a dilation

on a given input shape P is defined as the wavefront $\mathcal{W}_P(r)$ with the wavefront moving outwards w.r.t. P (see Figure 9.1).

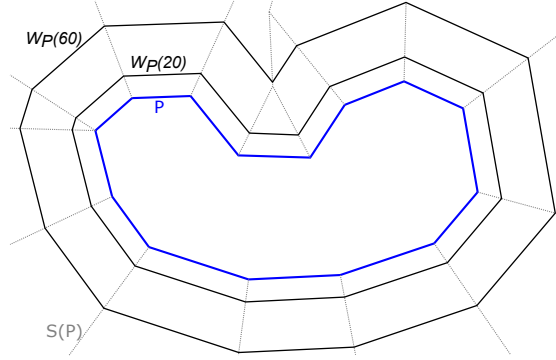


Figure 9.1 – Shape Dilation using straight skeleton. Input polygon in blue, refined straight skeleton in dotted gray with maximal reflex vertices set to 3, dilations at $t = 20$ and $t = 60$ in black (t being the radius of the structuring element).

The algorithm to compute the dilation of a polygonal shape given its straight skeleton is described as follows.

```

input :  $\mathcal{S}_{out}(P), r$ 
output: Dilation  $\delta_P(r)$ 
 $edgesToVisit \leftarrow \mathcal{S}_{out}(P).edges;$ 
while ! $edgesToVisit.isEmpty()$  do
     $e \leftarrow edgesToVisit.next();$ 
     $edgesToVisit.remove(e);$ 
    if  $\mathcal{S}_{out}(P).vertexTime(e.start) \leq r$  AND
     $r < \mathcal{S}_{out}(P).vertexTime(e.end)$  then
         $previousVertex = \delta_P(r).addVertex(e.getPositionAtTime(r));$ 
         $currentEdge = e;$ 
        flag = true while flag do
             $currentEdge = getLNext();$ 
             $edgesToVisit.remove(currentEdge);$ 
            if  $currentEdge == e$  then
                 $flag = false;$ 
            else
                 $currentVertex =$ 
                 $\delta_P(r).addVertex(currentEdge.getPositionAtTime(r));$ 
                 $\delta_P(r).addEdge(previousEdge, currentEdge);$ 
                 $previousVertex = currentVertex;$ 
            end
        end
    end
end
end
return  $\delta_P(r);$ 

```

Algorithm 3: Dilation calculation from a straight skeleton.

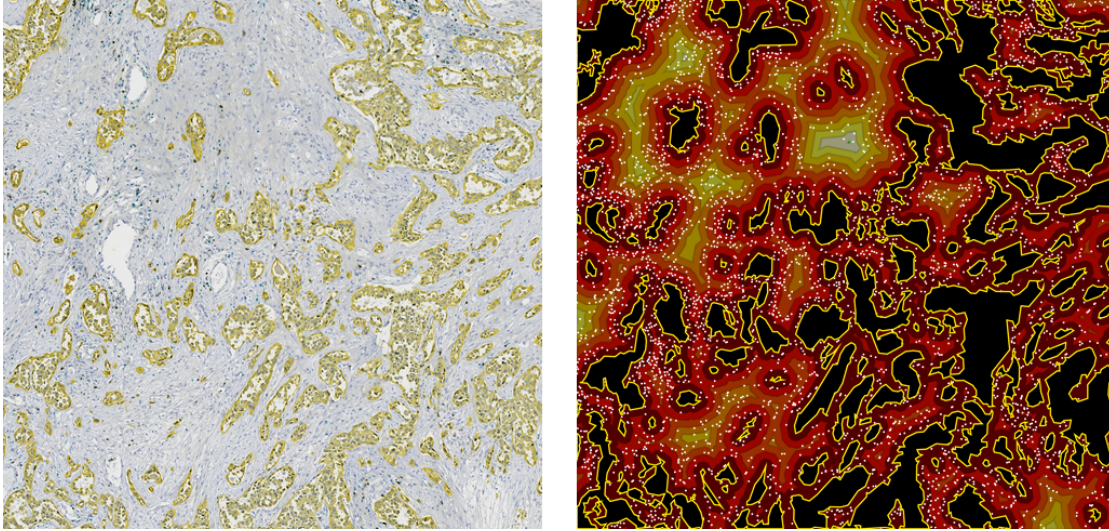


Figure 9.2 – Example application of dilations on histology. In this image of small-cell lung cancer, cancerous tissue stained in yellow is segmented (yellow lines on the right) and then dilated to identify distance intervals from initial shapes. Detected cell (as white dots on the right) can be assigned to a specific distance interval and allow for identifying cell clusters in function of the distance to the cancerous tissue.

An application using dilations in histology is the density analysis at different distances from segmented tissue. On Figure 9.2 segmentations of lung cancer are dilated at intervals of constant length. The different cells around this tissue are detected and can be assigned to a specific distance interval. From this a spatial analysis can be performed to compute cell clustering at certain distances from the cancerous tissue and possibly detect behavioral patterns on the cells.

9.2 Shape erosion

Similar to a dilation, erosions follow the same principle of offset calculation using the straight skeleton. The erosion operator is defined as follows:

Definition 9.2.1. Erosion[H] Let X be a subset of E . An erosion on X with the structuring element B is defined as a Minkowski sum (actually a difference):

$$\epsilon_B(X) = X \ominus B = \{x | B_x \subset X\}$$

Similar to dilations, erosions are non-invertible in general.

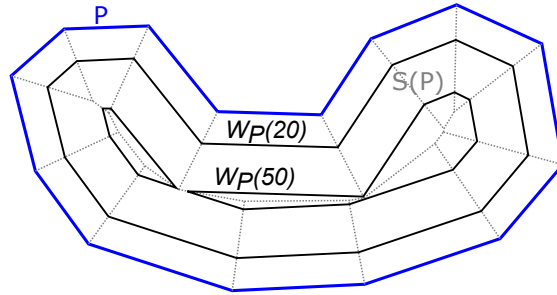


Figure 9.3 – Shape erosion using straight skeletons. Input polygon in blue, refined straight skeleton in dotted gray with maximal reflex vertices set to 3, erosions at $t = 20$ and $t = 50$ in black (t being the radius of the structuring element).

The algorithm to compute the erosion of a polygonal shape given its straight skeleton is described as follows.

```

input :  $\mathcal{S}_{in}(P), r$ 
output: Erosion  $\epsilon_P(r)$ 
edgesToVisit  $\leftarrow \mathcal{S}_{in}(P).edges$ ;
while !edgesToVisit.isEmpty() do
     $e \leftarrow edgesToVisit.next()$ ;
    edgesToVisit.remove(e);
    if  $\mathcal{S}_{in}(P).vertexTime(e.start) \leq r$  AND  $r < \mathcal{S}_{in}(P).vertexTime(e.end)$ 
    then
        previousVertex =  $\epsilon_P(r).addVertex(e.getPositionAtTime(r))$ ;
        currentEdge = e;
        flag = true while flag do
            currentEdge = getLNext();
            edgesToVisit.remove(currentEdge);
            if currentEdge == e then
                | flag = false;
            else
                currentVertex =
                 $\epsilon_P(r).addVertex(currentEdge.getPositionAtTime(r))$ ;
                 $\epsilon_P(r).addEdge(previousEdge, currentEdge)$ ;
                previousVertex = currentVertex;
            end
        end
    end
end
end
return  $\epsilon_P(r)$ ;

```

Algorithm 4: Erosion calculation from a straight skeleton.

An example of application of erosions can be seen on Figure 9.4, where the tissue contour is eroded in order to identify the damaged area of the tissue due to bad sample aging.

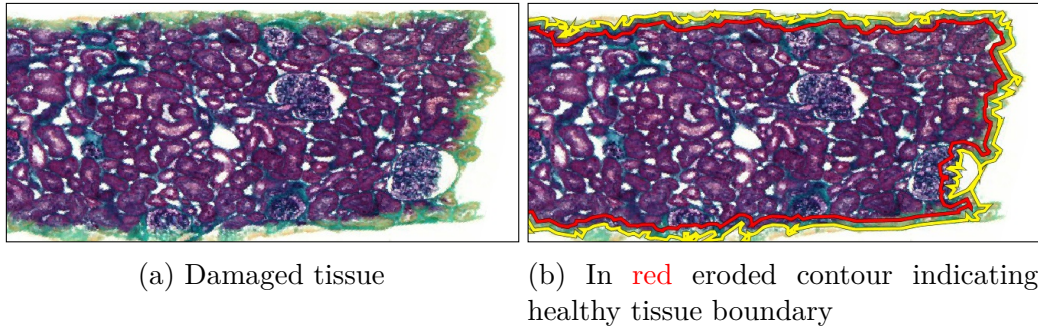


Figure 9.4 – Using straight skeleton morphological operations it is possible to define regions of a kidney tissue that should be ignored on further processing as they have been affected by tissue aging.

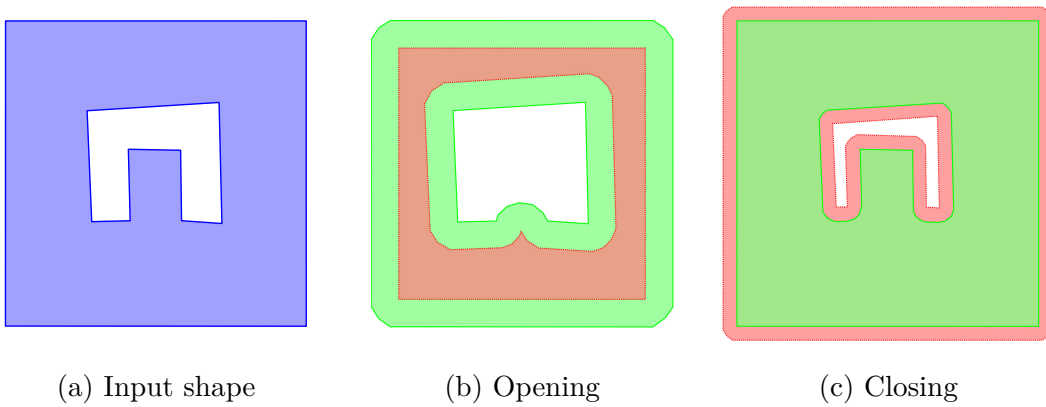


Figure 9.5 – Example of shape openings and closings. These operations require two morphological steps (in red and green). The result (in green) is a smoothed version of the input shape.

9.3 Shape openings and closings

Following the same principle of dilations and erosions it is possible to perform openings and closings on the input shape by reusing the dilation and erosion operations described below. An example of these operations can be seen in Figure 9.5. In addition, smoothing can be applied to shapes to remove acute angles on the input shape (see Figure 9.6).

9.3.1 Shape opening

An opening is the result of applying a dilation of an erosion of the input shape.

$$\gamma_B(X) = \delta_B(\epsilon_B(X))$$

This means that the input polygon is first eroded by B . Then, this result is dilated by B to obtain the opening result. The effect on the resulting shape is that small

outward variations (reflex input vertices) on the shape boundary disappear or are smoothed. For an example see Figure 9.5b.

9.3.2 Shape closing

A closing is the opposite operation of an opening. It is the result of applying an erosion of a dilation of the input shape.

$$\phi_B(X) = \epsilon_B(\delta_B(X))$$

In this case the input polygon is first dilated by B, and followed by a dilation by B on this result to obtain the closing result. The effect on the resulting shape is that small inward variations (convex input vertices) on the shape boundary disappear or are smoothed. For an example see Figure 9.5c.

9.3.3 Shape smoothing

With the morphological operators presented up to this point it is possible to perform smoothing of polygons to remove sharp edges on the contours. To perform this a procedure similar to a closing or opening is performed in the following ways.

$$Smoothing1(X) = \epsilon_{B''}(\delta_{B'}(\epsilon_{B'}(X)))$$

$$Smoothing2(X) = \delta_{B'}(\epsilon_{B'}(\epsilon_{B''}(\delta_{B''}(X))))$$

These two operations can seem pointless at first sight as B'' and B' are both used to perform both one dilation and one erosion. However, this operation allows to remove sharp interior reflex angles (with B'') and sharp exterior reflex angles (with B'). The structuring elements B' and B'' can be adjusted to the detail level needed on the resulting shape where B' handles internal sharp angles and B'' takes care of external ones. An example of this procedure can be seen in Figure 9.6. It is important to notice that, although *Smoothing1* and *Smoothing2* have both two erosions and two dilations, results on these operations differ because the smoothing of internal and external details is done in a different order.

9.4 Straight Medial Axis

One of the main issues with straight skeletons is the amount of branches that they hold. This obfuscates the general structure of the shape that can be obtained when some of the most locally descriptive branches are removed. When these branches are removed the resulting graph is generally known as the medial axis.

As mentioned in Definition 5.0.1, a medial axis is almost the same as a morphological skeleton except that the original shape might not be completely recovered from the medial axis. The reasoning for this property is that medial axes aim to describe the general features of the target shape. An example of application of this structures can be seen in Figure 9.7 where the anomalies on the thickness of

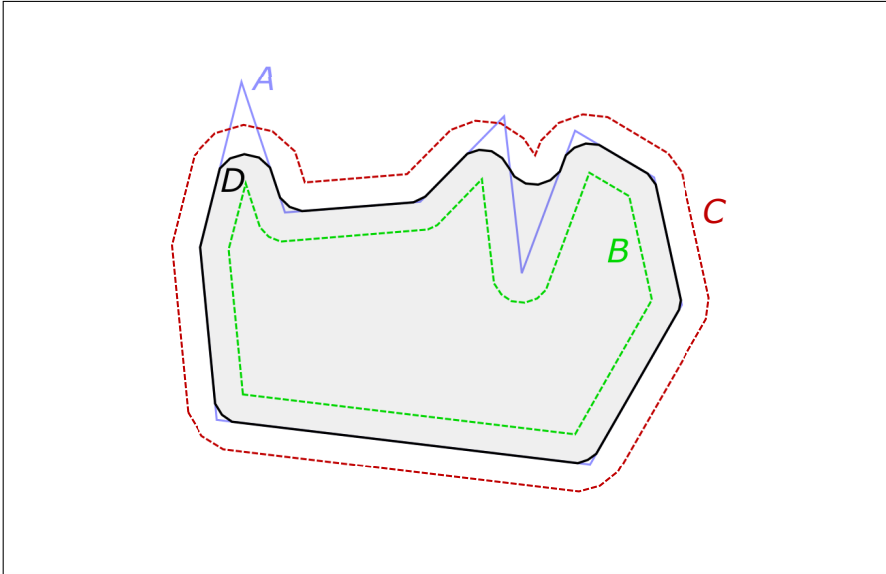


Figure 9.6 – Example of polygon smoothing by applying a sequence of erosion, double dilation, and erosion on the initial shape. This allows to remove internal and external sharp angles on the input shape. In this example the input shape (**A**) is affected with an erosion (**B**) followed by a double dilation (**C**) and a final erosion to obtain the smoothed shape (**D**).

a uterus is estimated through the calculation of the mean distance to the uterus contour of the medial axis as well as its variance. We can assess the tissue by comparing the variance of these distances and compare between wild-type with knock-out samples.

Several studies have been used to prune skeletons. For any significance criterion used, it should establish a strict relation between skeleton branches and the relevance of the shape parts that they represent. In the case of two dimensions, some criteria involving propagation velocity, maximal thickness radius function, axis arc length and the boundary/axis length ratio can be found in [169]. Other pruning methods use contour partitioning via discrete curve evolution [170] and with bending potential ratio [171], where the decision on whether pruning a branch is based on the context of the contour segment corresponding to that branch. A more recent pruning criterion takes a group-wise medial axis transform that yields a fuzzy significance measure for each branch [172]. In [173] the author recurred to branch level features that are evaluated to select the best describing branches of the skeleton. This last method resulted an interesting method for us and easily applicable to straight skeletons. In this section we describe the principles for extracting the medial axis of a vectorial shape from its refined straight skeleton based on the features of the branches at different depth levels.

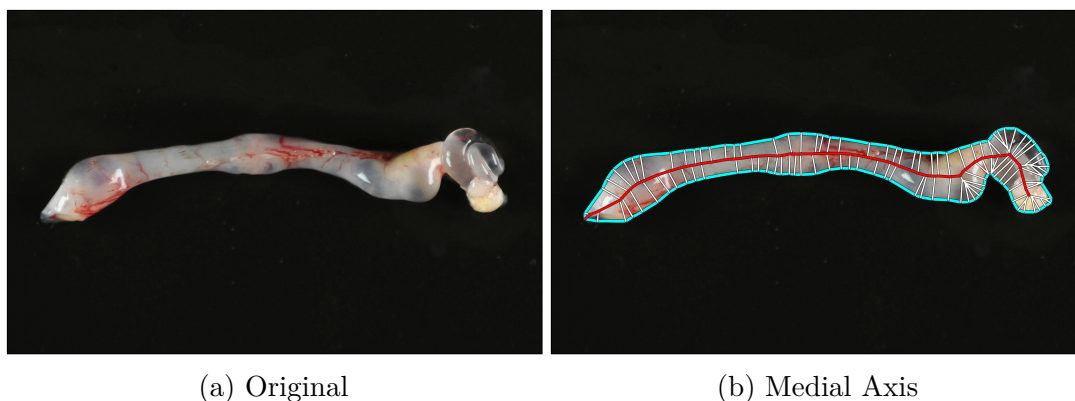


Figure 9.7 – Medial axis applied on the contour of a uterus to assess its behaviour with respect to the thickness of the sample. Mean and variance of the thickness can be used to characterize samples as wild-type or knock-out.

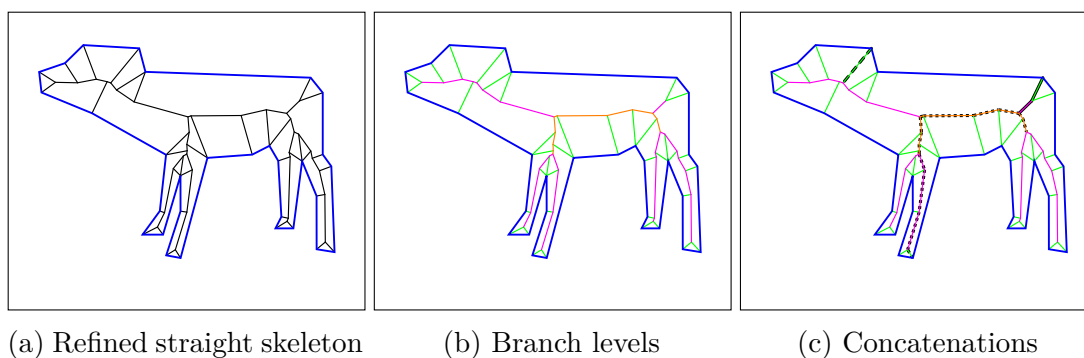


Figure 9.8 – Example of medial axis computation. (9.8a) Refined straight skeleton with maximum reflex vertices set to 3. (9.8b) Levels of the skeleton, level 1 in green, level 2 in magenta and level 3 in orange. (9.8c) Three possible concatenations on the skeleton.

9.4.1 Detecting branch levels

In order to prune a straight skeleton from the branch levels of the skeleton it is necessary first to describe the way to compute the level of each branch segment. The method described here takes a level as the minimal amount of pruning steps needed in order to have a branch as a terminal branch. Let's take the Figure 9.8 as an example. Branch segments in green are branches that are already terminal and are labeled as being level 1 branches. Next, if those green branches are removed, branches that become terminal (in magenta) are labeled as level 2 branches. This process continues until all the skeleton edges have been labeled with a specific level (see Figure 9.8b).

Once the labels are assigned it is possible to define branch concatenations up to a given level. A concatenation C_h is a set of connected edges that start at a leaf node h of the skeleton and follow a path that is always increasing in level. The concatenation continues until no more increasing paths are available or until

there is more than one increasing path, this level is known as the maximum level n that a concatenation can achieve. As an example, in Figure 9.8c there are three possible concatenations, an example for each of the three possible levels. As the figure shows, the level 3 concatenation (dotted line) stops when it cannot find an edge of higher level. In the case of the level 2 (full line) and 1 (dashed line) concatenations, they stop when they find more than one neighboring edge of higher level. In addition to full concatenations, sub-concatenations up to specific levels can be considered (i.e. The sub-concatenation C_h^i starting at leaf h and going up to level i , $1 \leq i \leq n$).

The interest of using concatenations is that they can provide incremental information about each branch of the skeleton going up to the center of the shape, thus becoming a useful tool to decide when to cut the skeleton.

9.4.2 Defining branch features

Once all skeleton edges have been associated with a specific level and with concatenations defined, it is possible to use concatenations to take measurements and decide where to prune the skeleton. First of all, a vertex on the skeleton is relevant if it describes a feature of the shape (the circle it covers) more than any other vertex in the skeleton (and any other point in the shape). Second, a vertex is not relevant if it is too close to another vertex with a similar covering radius. Third and last, the larger a concatenation gets, the more relevant it becomes because it holds features of the shape at more resolution levels.

From this rationale three criteria have been used to define the relevance of a skeleton branch:

- δ_h^i : The difference of the radii of the covering circles of vertices at the leaf node i and the other end of C_h^i .
- $\lambda_h^i = \delta_h^i - d(p, q)$: The distance covered by the circle at end point vertex p that is not covered by the circle of the vertex q on the other end of C_h^i . Here, $d(p, q)$ is the euclidean distance between p and q .
- L_h^i : The length of the sub-concatenation C_h^i .

9.4.3 Pruning strategy

Since the medial axis describes the central axis of concave features on the shape, the first step is to filter the branches that are concave enough. In the case of straight skeletons this can be achieved by visiting the vertices of the initial polygon and filtering out branches that start from non-concave vertices ($\angle(v_i) > \theta_1 | 0 < \theta_1 < \pi$). This allows to obtain meaningful medial axes without losing the quality of the skeleton (and the reconstruction). Any branch with its leaf vertex having an angle greater than θ_1 this is considered to be removed.

From the criteria presented in the last subsection, these values need a threshold to identify a branch that is candidate to be pruned. Based on the same principle

than [173] and adapting it to our purposes the following operation identifies a pruning candidate branch if its evaluation is true:

$$\delta_h^i < 0 \text{ AND } \lambda_h^i < \theta_2 \text{ AND } L_h^i < \theta_3$$

The values for θ -values must be set according to the needs for the resulting medial axis. For this work we found as best values the following.

- θ_1 : As explained before, this criterion simplifies the skeleton to prune before its gets actually pruned. The best results were found by setting this parameter to $2\pi/3$.
- θ_2 : Since this threshold is closely related to the size of the input shape, we took as threshold a percentage of the length $side_{min}$ of the shortest side of the bounding box covering the input shape. We found the best results using $\theta_2 = 5\% \text{ } side_{min}$.
- θ_3 : As this criterion is not directly related to the size of the shape but to the length of the branches at different levels of depth we use the mean length of concatenations at a given level i for all concatenations of the skeleton attaining that level. Based on our experiments we found that $\theta_3 = 3 * \bar{L}_i$, where \bar{L}_i is the mean length of all sub-concatenations C_h^i at level i .

Now, in order to find the pruning level for all concatenations, each one is evaluated from its deepest sub-concatenation and descending in level until finding a positive candidate. When a sub-concatenation meets the criteria described above, all edges associated to that sub-concatenation become pruning candidates. The rest of possible sub-concatenations are not considered once the first positive candidate is found as they are less general than the first positive candidate.

Once the pruning candidates are identified, they are removed taking care of respecting the original skeleton connectivity. This is done because some candidate edges belonging to one concatenation can be non-candidate edges of other concatenations (An example is available in Figure 9.9).

It is worth noting that, since a medial axis is based on the information coming from concavities in the input shape, the skeleton has to be pre-pruned, removing all reflex vertices and their incident edges. In addition, a maximum angle is defined on the edges incident to a vertex of the shape, to make sure that only relevant concavities are taken into account for the medial axis. In our experiments we have set this angle to $7\pi/9$, terminal edges coming from vertices with more than this angle are removed before the actual pruning takes place. An analysis of the three parameters of the medial axis can be seen on Section 10.3 where different values are assigned to these parameters to explore the impact they have on the resulting medial axis and the reconstruction of the initial shape obtained from it.

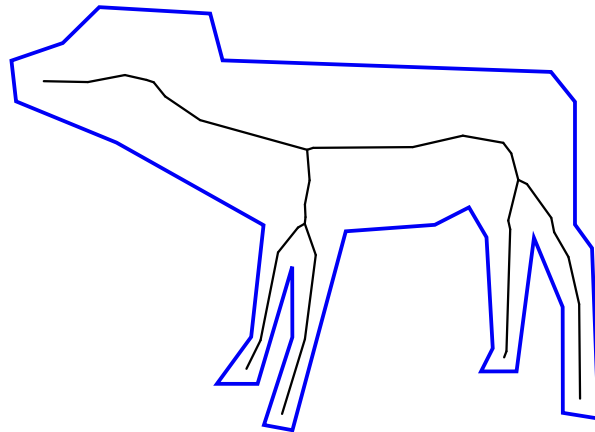


Figure 9.9 – Example of resulting medial axis on shape taken from [174]

9.5 Evaluation

Since the medial axis is meant to summarize the skeleton of a given shape, we decided to evaluate the medial axis comparing the reconstruction obtained from it against the original shape. In order to do this it is necessary to have a reconstruction of the shape starting from the medial axis. For this, we describe in the next chapter the reconstruction method we used. Following this, we provide the medial axis evaluation, as well as some understanding of the parameters for calculating the medial axis.

Chapter 10

Shape reconstruction from Straight Medial Axes

Evaluating skeletons can be a challenging task due to the lack of a ground truth skeleton to compare existing methods with. However, there are several ways to assess the robustness of a skeletonization method [175]. More precisely, skeletons can be evaluated on their structural properties which are homotopy [176], geometry preservation [177, 178, 179, 176] and skeleton robustness to distortions like noise [180, 181, 178] or rotation [181].

In the case of single and refined straight skeletons all the input information (all polygon points) is taken into account to build the skeleton. This guarantees the homotopy of the skeleton to be respected as single shape skeletons remain connected and also every hole is taken into account for the skeleton construction. Taking all the vertices of the input shape also allows for full shape reconstruction from the skeleton, even the small concavities have a branch to describe them. It is precisely this criterion (reconstruction robustness) the one we use in this chapter to evaluate the robustness of the straight medial axis and to analyze its parameters. The reconstruction of the shape from the medial axis is compared to the original shape to obtain an error rate relative to the shape dimension.

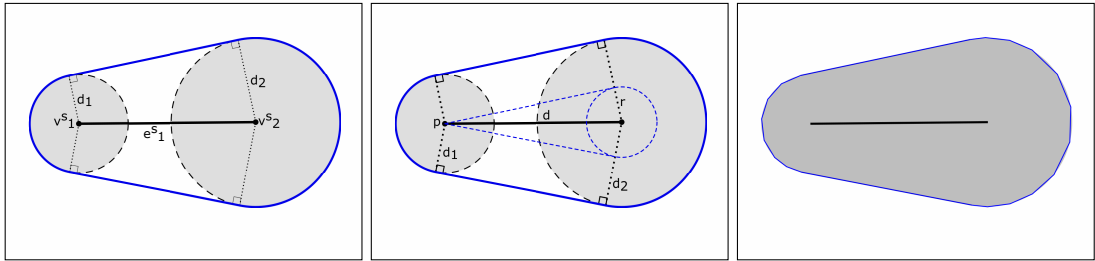
10.1 Reconstructing a shape from a medial axis

In order to reconstruct a shape out of a medial axis two things are needed:

- The medial axis $S(P)$: This structure holds the distribution of the object on the space.
- The minimum distance d_i^S from each point v_i^S in the medial axis to the shape boundary: This holds the information of the thickness of each branch on the medial axis allowing to obtain the desired reconstruction.

Most skeletonization and medial axis methods provide both needed items to reconstruct the original shape. From them, a circle with a radius equal to the

CHAPTER 10. SHAPE RECONSTRUCTION FROM STRAIGHT MEDIAL AXES



(a) Medial axis and theoretical reconstruction (b) Circle-to-circle tangents calculation (c) Polygonal shape reconstruction

Figure 10.1 – Single edge medial axis shape reconstruction. A single edge medial axis (in black) can be reconstructed by the union of boundary circles (in gray) and their common outer tangents (in blue).

described distance can be drawn at each point of the skeleton, finally obtaining the reconstruction from the skeleton as the union of all the drawn circles. However, in practice this method can be inefficient as neighbor points in the skeletons tend to overlap a lot their corresponding circles, which becomes a lot of wasted effort by superposing the majority of created circles. However this is different when using straight skeletons as it is possible to use circle tangents to reconstruct the original shape.

As straight skeletons are made out of straight line segments and a distance to the object boundary is given at each end of these segments, one can rebuild the original shape by making the union of the reconstructed sub-segments of the shape (See Figure 10.1). This is achieved by computing the line tangents to both circles of the ends of each medial axis segment.

Given a medial axis segment e^A and its end points v_1^A and v_2^A , each at a distance d_1 and d_2 to the shape boundary (respectively). It is possible to create a reconstruction of the shape by a polygonal reconstruction of the circles c_1 and c_2 with radius d_1 and d_2 , centered at v_1^A and v_2^A , respectively (dashed black lines in Figure 10.1b). Alongside the reconstructed circles (which are approximated by a fixed amount of points), their common outer tangents can be used to join the circles [182]. This is done by computing the tangents of a point p and a circle of radius $r = \max(d_1, d_2) - \min(d_1, d_2)$ (dashed blue lines in Figure 10.1b), and then translating them in perpendicular direction to the tangents by a distance of $\min(d_1, d_2)$. From this, one can obtain the original shape approximation (blue line in Figure 10.1c).

This same principle of reconstruction can be used for medial axes with multiple edges. In this scenario, all circle tangents are computed first and then they are joined by partial circles joining neighboring tangents (see Figure 10.2).

The reconstruction obtained by the method just described opens the possibility of measuring the error of the reconstruction. This is achieved by using the Hausdorff distance (Equation 10.1) between the input shape and the reconstruction from its medial axis [183], computing the distances between control points of

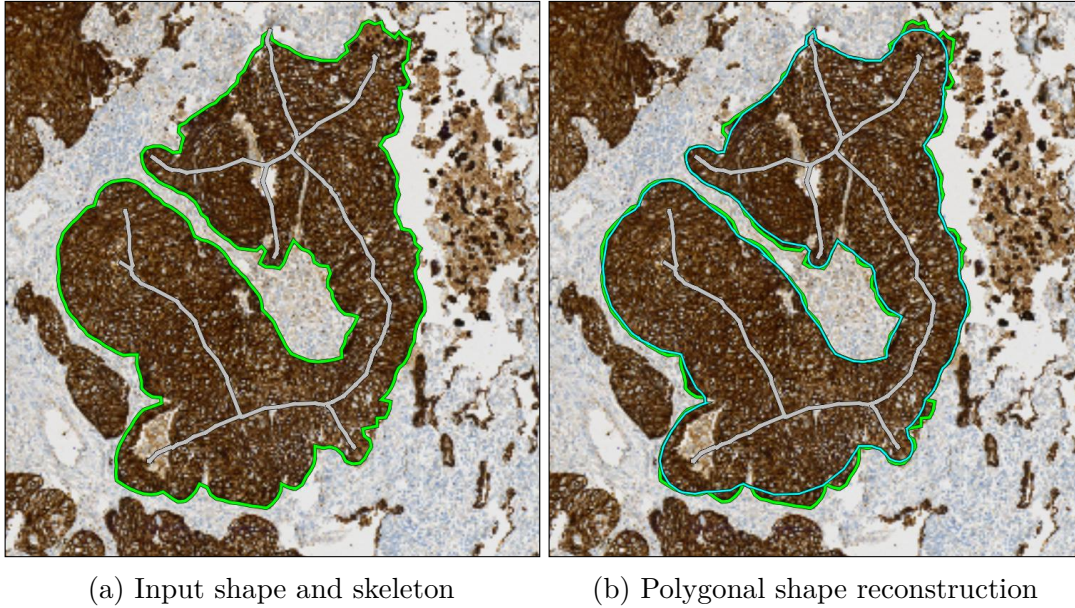


Figure 10.2 – Multi-edge medial axis shape reconstruction. Multiple medial axis edges (gray lines) can be used to approximate the original input shape (green lines) by concatenating circle-to-circle tangents and segments approximating circles (cyan lines).

both the input shape and its medial axis reconstruction.

$$d_H(X, Y) = \max \left(\sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y) \right) \quad (10.1)$$

In this case, the distance is defined by the largest minimal distance from the original shape edges to its reconstruction, and the largest minimal distance from the reconstruction edges to the original shape. The largest distance from these two is the actual Hausdorff distance. This measure is relevant for this particular case in the sense that it is measuring how close is the reconstruction to the original shape.

10.2 Straight medial axis evaluation by reconstruction

To test our medial axis method we used the database used in [184] used to exemplify symmetry-based indexing, containing 90 different shapes as binary images. These shapes were then vectorized using the method described in [174] where curvature is used to detect dominant points on the border of input shapes. To compute the medial axis of these shapes we used our medial straight skeleton with the following parameters:

- θ_1 (terminal vertex angle) = $2\pi/3$

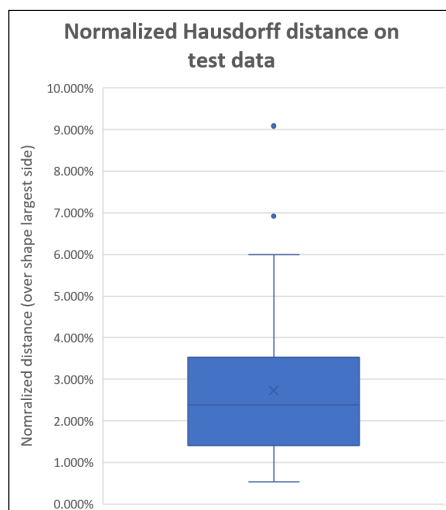


Figure 10.3 – Normalized Hausdorff distance of the medial axis reconstruction against the original shape. The value is normalized against the largest side of the bounding box of the shape.

- θ_2 (difference between circles on neighboring node)= 5
- θ_3 (minimum concatenation length at each level relative to the mean length at each level)= $3\bar{L}_i$

In addition, in order to compare distances from polygons of different sizes, the distances are normalized by the maximum value between the width and height of the polygon bounding box, this brings distances to the same scale so that they can be compared against each other. These are expressed as percentages with respect to the maximum value presented above.

From these configurations the following evaluation was obtained. The mean distance is 2.723% (using percentages due to normalization), with a standard deviation of 1.528%. A maximum distance of 9.084% and a minimum distance of 0.53% (See Figure 10.3).

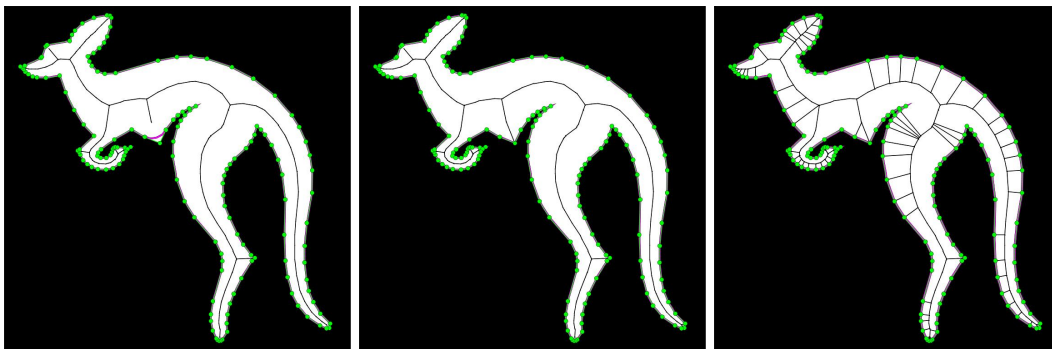
10.3 Understanding parameters of the straight medial axis

Another analysis that can be performed on the straight medial axis is the impact of each of the pruning parameters by comparing the reconstruction of the resulting medial axis. For this an example shape (see Figure 10.4) will be used to find its medial axis, varying the parameters to present the consequences of these changes to the reconstructed shape. In the following figures **green** will be used to describe the input shape, **black** lines will represent the medial axis, and **magenta** will describe the shape reconstruction from the medial axis.



Figure 10.4 – Example shape used to explore and understand medial axis parameters.

- **Changing θ_1 :** This parameter is associated to the pre-pruning step, where vertices that are not *highly* convex are removed. On Figure 10.5 this parameter is set to π , $2\pi/3$ and $\pi/2$ from left to right to show that higher values will include vertices on flat angles in the medial axis. Decreasing this value will only keep those input vertices which are actually located on pronounced concavities of the shape. Furthermore, it is important to notice that although the reconstruction is better when values of θ_1 approach π , it also causes to have a more complex skeleton. In contrast, having values approaching 0 gives a simpler skeleton, but will have an impact on concavities of the input shape when the reconstruction is performed.



(a) $\theta_1 = \pi/2$

(b) $\theta_1 = 2\pi/3$

(c) $\theta_1 = \pi$

Figure 10.5 – Varying θ_1 parameter on the medial axis

- **Changing θ_2 :** This parameter is in charge of keeping meaningful details of the straight skeleton in the medial axis and excluding edges that do not

provide much detail information. This is done comparing the length of the concatenation at branch concatenation vertices and the minimum distance to the original shape at that position. If the difference between these two values is greater than a percentage (the parameter value) of the shape minimum side length, then the concatenation is kept as it is considered a detail worth keeping in the axis. In Figure 10.6 this value is changed to show that low values will allow to keep more details of the shape at the cost of having a more complex medial axis. On the contrary, a simpler medial axis is achieved for high values but impacting the reconstruction of the input shape.

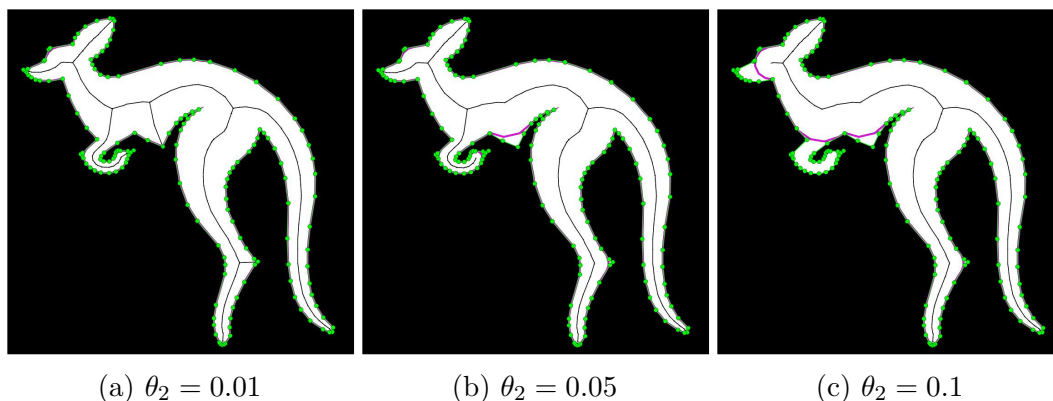


Figure 10.6 – Varying θ_2 parameter on the medial axis

- **Changing θ_3 :** This parameter deals with the length of concatenation at each level of the medial axis. Here, a low value will filter out skeleton edges that are very small compared to the mean length at each level of the concatenations. When setting a high for this parameter, larger edges will also be left out of the medial axis. And if the parameter is set to really high level, it will prune all edges except the last one left (See Figure 10.7).

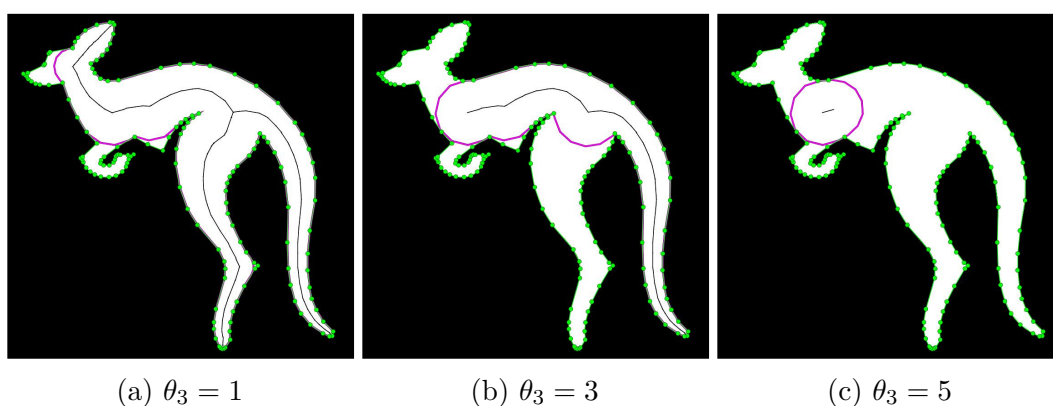


Figure 10.7 – Varying θ_3 parameter on the medial axis

As a perspective of this description, one could imagine an algorithm that ad-

10.3. UNDERSTANDING PARAMETERS OF THE STRAIGHT MEDIAL AXIS

justs $\theta_1, \theta_2, \theta_3$ to search for an "optimal" medial axis using the reconstruction error as the minimized value.

Chapter 11

Exploration on weighted straight skeletons

During the straight skeleton research we became aware of some possible new features on the mathematical morphology methods using straight skeletons. Straight skeletons are an interesting tool here because they only depend on the amount of control points on the input shape, so even objects in large images can be efficiently computed to obtain their skeletons. With this in mind, we thought about an irregular front propagation when constructing the straight skeleton. More precisely, build the skeleton from the propagation of edges that associate each a different weight, causing edges to propagate at different speeds and obtain a *weighted straight skeleton*. For example, if we are interested on a positive horizontal propagation (0°), then an edge propagating at an angle of 45° would move faster than an edge propagating at 90° , and it will be slower than an edge propagating at 10° . We explore in this chapter weighted straight skeletons and present a method that takes the classic straight skeleton and adapts it to work with weights. We show the considerations when constructing the solution as well as the challenges that still exist in the current state of development.

11.1 Introduction and State of the art

Weighted straight skeletons differ from unweighted straight skeletons only in the speed $\sigma(e) \in \mathbb{R} \setminus \{0\}$ at which each edge e moves on the wavefront. In this case σ denotes the weight function and $\sigma(e)$ the weight of e . Edges move in such way that $e(t) = \bar{e} + t \cdot \sigma(e) \cdot n(e)$, where \bar{e} is the supporting line of e , and $n(e)$ is the inwards unit normal vector of e . In the case that $\sigma(e) < 0$ the edge moves outwards with speed $|\sigma(e)|$. In this case, split events happen when the wavefront becomes non-simple because a wavefront vertex v meets a part of the wavefront. With this change, vertex v no longer needs to be reflex. Additionally, split events do not necessarily split the wavefront into multiple parts. And to make the problem even more complex, different possibilities may exist to keep the wavefront as a planar collection of simple polygons between events. One of the main issues with

Polygon with holes	$\sigma = 1$	$\sigma > 0$	σ arbitrary
$\mathcal{S}(P)$ is connected	Y	Y	N
$\mathcal{S}(P)$ has no crossings	Y	Y	N
Edge face $f(e)$ is monotone w.r.t. e	Y	N	N
Edge boundary $f(e)$ is a simple polygon	Y	N	N
Roof model $\mathcal{T}(P, \sigma)$ is z-monotone	Y	Y	N
$\mathcal{S}(P)$ has $n + v - 1 + h$ arcs	Y	Y	N
$\mathcal{S}(P)$ is a tree	N	N	N

Table 11.1 – Comparison of properties in weighted and unweighted straight skeleton.

weighted straight skeletons arrive with negative weights, which can cause wavefront components to be unbounded on their propagation and making impossible to create an appropriate skeleton.

To understand better how weights affect straight skeletons Biedl et al. [185] provides a table (Table 11.1) with a comparison of the properties of unweighted and weighted straight skeletons.

Research on weighted straight skeletons started in 1998 when Eppstein and Erickson introduced the term [155]. They proposed a method for computing non-weighted straight skeletons in time $O(n^{8/5+\epsilon})$ that *could* be extended to compute weighted skeletons without major modifications to the code. This subject remained mostly unknown until 2009 when Kelly published a new algorithm to compute the weighted straight skeleton accepting non-zero weights in time $O(n^2 \log n)$, for polygons with n vertices [186]. Biedl et al. [185] presented on their research work the challenges of adding weights on edges when computing weighted straight skeletons. More specifically, they presented the ambiguities that arrive when propagating front edges collapse with each other. On a publication later the same year [187], the same authors presented an algorithm to compute straight skeletons with weights in time $O(n \log n)$, where they take some assumptions when handling edge collapses by analyzing superior and inferior coverings of the propagation front separately and then fusing the results afterwards. Finally, in a recent publication, Eder [188] proposed an algorithm to compute weighted straight skeletons in time $O(n^2)$.

11.2 Ambiguities and implementation considerations

When creating a weighted straight skeleton, front propagation edges move at different speeds, which makes a great deal of difference with respect to non-weighted straight skeletons because wave front vertices don't follow the same path. In fact, there are some considerations that have to be taken into account to obtain a valid straight skeleton [185].

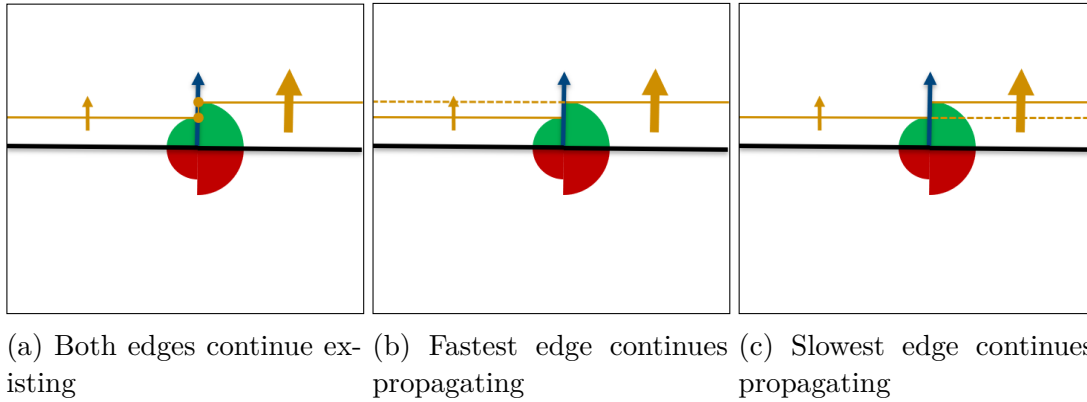


Figure 11.1 – When an edge collapses and its incident edges are colinear, an ambiguity arrives if the weights of the incident edges are not the same. In these cases a decision must be taken, and each possibility has its own compromises.

First, given a vertex v with two incident edges e_1 and e_2 that are supposedly not colinear. Forming an angle α_1 with counter-clockwise edge e_1 and α_2 with clockwise edge e_2 . The vertex speed $\sigma(v)$ of v is defined as

$$\|\sigma(v)\| = \begin{cases} \frac{|\sigma(e_1)|}{\sin \alpha_1} & \text{for } \alpha_1 < \pi \\ \frac{|\sigma(e_2)|}{\sin \alpha_2} & \text{for } \alpha_2 < \pi \end{cases} \quad (11.1)$$

Here, it is impossible to *determine the speed of σ_v when $|\alpha_i| = \pi/2$* because incident edges become colinear, and if edges e_1 and e_2 have different weights it is not possible to define the new vertex speed. In this cases there are three possible outcomes:

- *Two vertices are created each with a different speed.* When an edge collapses and incident edges are colinear, both edges continue moving at their initial speed generating a discontinuity on the wavefront. Supposing that we remove this discontinuity with an edge joining both vertices, this will cause a degeneracy of the wavefront. Hence, taking this option is not recommended (See Figure 11.1a).
- *Terminate the propagation of the edge with the minimum speed.* In this case angle α_1 is set to zero if $\sigma(e_1) < \sigma(e_2)$, or else α_1 is set to zero. This will cause the edge with minimum weight to collapse instantly and only one joint edge will continue the propagation with the maximum weight (See Figure 11.1b).
- *Terminate the propagation of the edge with the maximum speed.* This case is the opposite to the previous one, the angle of the vertex is set to zero on the side of the edge with maximum weight, which will then instantly collapse and a single joint edge will continue the propagation at minimal speed (See Figure 11.1c).

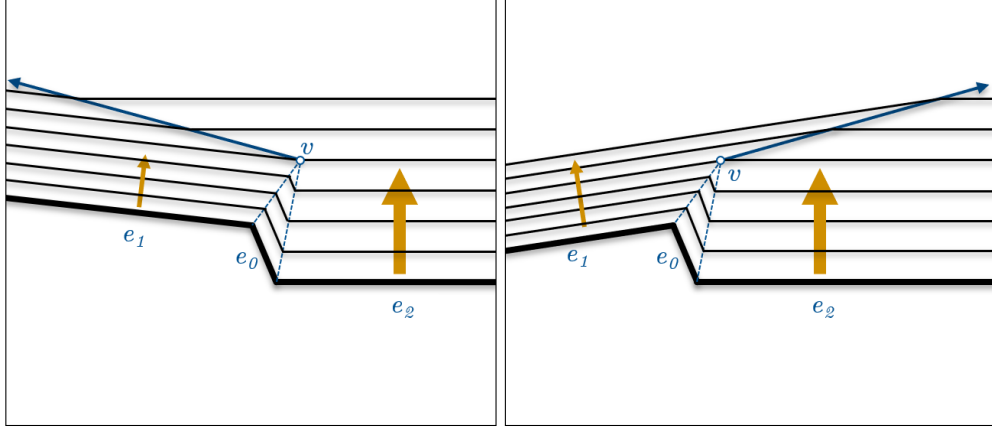


Figure 11.2 – Ambiguity when defining vertex direction from colinear incident edges differently weighted.

Second, given the following equality

$$\frac{\sigma(e_1)}{\sin \alpha_1} = \frac{\sigma(e_2)}{\sin \alpha_2} \quad (11.2)$$

and denoting $\gamma = \alpha_1 + \alpha_2$ the angle spanned from e_1 to e_2 at v we can reformulate the previous equation as

$$\frac{\sin \alpha_1}{\sigma(e_1)} = \frac{\sin \gamma - \alpha_1}{\sigma(e_2)} = \frac{\sin \gamma \cos \alpha_1 - \cos \gamma \sin \alpha_1}{\sigma(e_2)} \quad (11.3)$$

inferring then

$$\cot \alpha_1 = \frac{\cos \gamma + \frac{\sigma(e_2)}{\sigma(e_1)}}{\sin \gamma} \quad (11.4)$$

With the previous equation it is possible to see that the angle α_1 is indeterminate in the case that $\sin \gamma = 0$. And to make things worse, analyzing the limit of α_1 as γ approaches monotonically to π we obtain different results

$$\lim_{\gamma \nearrow \pi} \alpha_1 = 0, \lim_{\gamma \searrow \pi} \alpha_1 = \pi \quad (11.5)$$

Here the *direction of propagation* of the vertex v becomes uncertain because it can take two values. This ambiguity actually comes to the same solution as the previous one, one of the weights of the incident edges has to end up taking the lead on the propagation and thus defining the vertex direction (see Figure 11.2).

And last but not least, when *multiple split* events happen simultaneously at the same position in the plane there exist an ambiguity when reconnecting edges that continue the propagation avoiding self-crossings on the wavefront. Let's take the case where e_1, \dots, e_{2k} are the edges that meet simultaneously at the same point

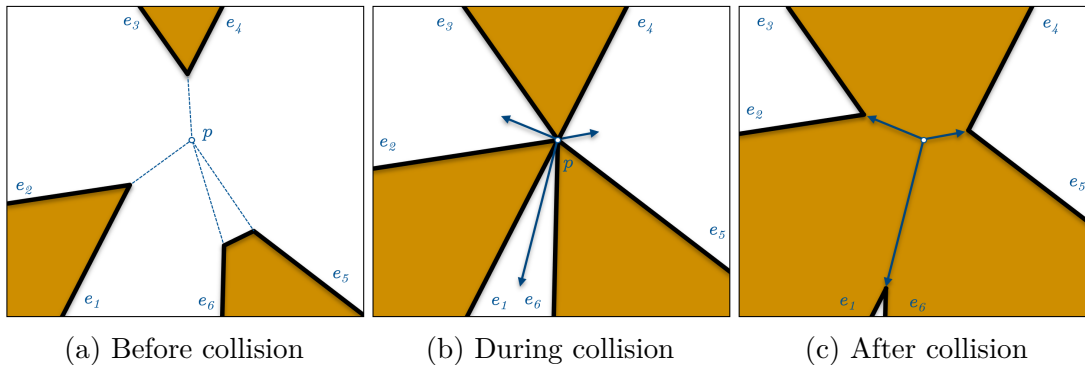


Figure 11.3 – Before, during and after a multiple split event on the wavefront. During a multiple split event edges can collapse into the event position which leaves the challenge of deciding the correct edge pairing that avoids crossing on the propagating front. In this example an edge edge collapses at point p and the edge pairs $(e_1, e_2), (e_3, e_4), (e_5, e_6)$ are re-paired to $(e_2, e_3), (e_4, e_5), (e_6, e_1)$, maintaining the front crossing-free when the propagation continues.

p . Here we note that e_i and e_j are paired edges that should continue on the wavefront after handling possible collapsed edges. It is necessary that this pairing is done correctly to keep the wavefront crossing-free. To do this and to avoid the ambiguity of determining the edge pairings, all concerned clockwise edge pairs $(e_1, e_2), (e_3, e_4), \dots, (e_{2k-1}, e_{2k})$ are rearranged as $(e_2, e_3), (e_4, e_5), \dots, (e_{2k}, e_1)$. This way we make the propagation of the wavefront keeps a correct topology. To understand this change we can simply interpret the operation as a switch of association on the colliding edges (See Figure 11.3).

11.3 Current development status

Considering the previous ambiguities, and to avoid the propagation front to fold onto itself due to the use of negative weights, our implementation only allows positive weights on edges and at the time of writing the current document the implementation only takes convex polygons (See example in Figure 11.4). However it is expected to be able to take weak simple polygons, possibly with holes in it.

11.4 Application opportunities on histological images

When performing segmentation on histological images it happens often that resulting components are incomplete or not connected because of how the tissue is manipulated and its particular state at fixation. For example, when trying to perform a segmentation of the glomerulus on Figure 11.5a the result is two connected component that actually belong to the same glomerulus(Figure 11.5b). In this

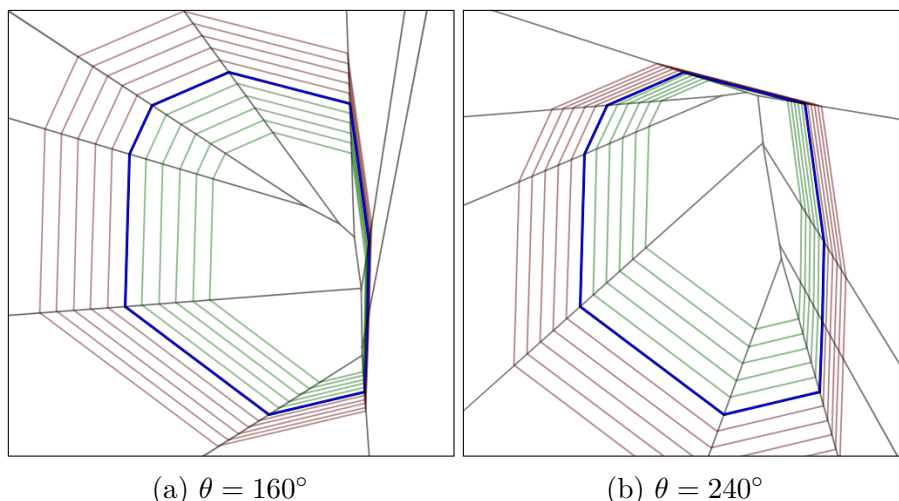


Figure 11.4 – Example of a weighted skeletons computed with our solution, which sets edge weights according to their orientation. Both straight skeleton in gray and offsets (in green for erosions and red for dilations) are presented for the same shape (in blue) oriented to two different angles.

case one could perform a dilation on both of these objects until finding that these two objects overlap (Figure 11.5c). However this could end up with an erroneous overlapping with other objects. Instead, the dilation could be performed orienting the edges to move faster on a specific direction, hence using the weighted straight skeletons to dilate the initial segmentation (Figure 11.5d). Taking the convex hull of the union of the initial objects that overlapped will give us the segmentation that is expected from the glomerulus (Figure 11.5e).

11.5 Conclusion

In this chapter a method for calculating weighted straight skeletons has been discussed, along with some existing ambiguities on their definition, and including some application options for histology images. Although the method is still in development, the opportunities for improving and applying this methods are not limited to biology. As seen in previous chapters, straight skeletons have different applications in multiple domains like spatial analysis, movement planning for mechanical tools, etc [189]. An interesting possible development could be the extension to three-dimensional weighted skeletons where weights would be applied on faces instead of edges, and skeletons would be composed of not only straight segments, but constrained planes as well.

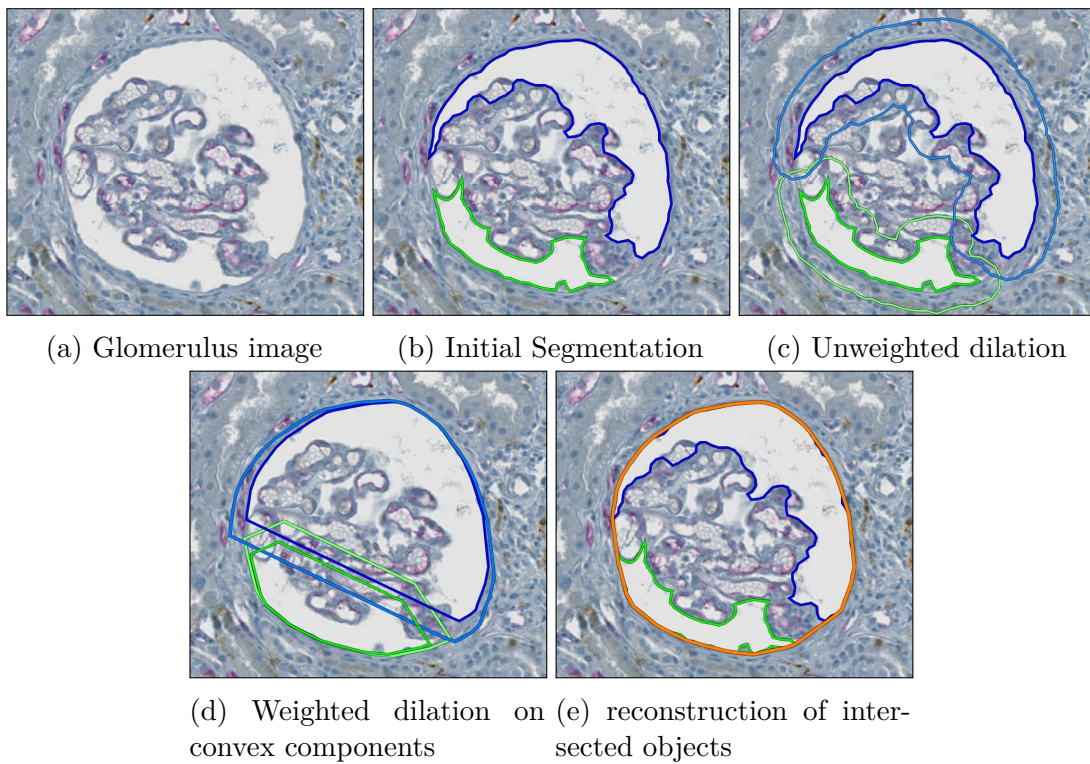


Figure 11.5 – Applying weighted dilations to match directed complementing objects.

Chapter 12

Conclusions on vectorial shape morphology

In this second part, theory and application of straight skeletons has been presented, providing spatial information about the structures found on virtual slides. Straight skeleton is an interesting and efficient tool to use on vectorial data with multiple purposes for shape and spatial analysis. By refining the classical straight skeleton to achieve precision, and approximating the continuous Voronoi diagram of segments by using just straight segments, refined straight skeletons provide an ideal trade-off between performance and precision. In addition, multiple morphological operations have been implemented to give context information on shapes by computing their offsets. Additionally, using refined straight skeletons can complement very well discrete curve detection methods such as [174], providing not only vector-based contours out of images, but also a tool set to analyze them with high precision. All these features allow to not only have contextual data that can help analyzing structures on histological tissue, but also can be the input for other methods such as adapted active contours based on morphological operations [190]. These methods can be re-purposed to provide further information about shapes in a plane. In the context of immunopathology, multiple tools have been presented to improve quantification of biological structures by refining their segmentation. Additionally, using the presented morphology methods it will be possible to apply spatial analysis on fully segmented structures (shapes) rather than just their mass center (points), altogether without worrying about the size of acquired images. From here, new tools can be developed to identify behavioral patterns on immune cells around cancerous tissue and better orient prognoses, not only to avoid hazardous cell behaviours, but also to favor those that improve patient's life. As perspective, more efforts need to be put into developing solutions for 3-D volume skeletons [191]. Weighted straight skeletons can also be further improved, by applying the sample principle of reflex vertex refining applied in refined straight skeletons, achieving a better approximation of a continuous wave-front propagation. As a final thought, generation of 3-D volume weighted straight skeletons could provide an even more complete tool in the context of geometrical

CHAPTER 12. CONCLUSIONS ON VECTORIAL SHAPE
MORPHOLOGY

morphology.

Conclusions

In this thesis several solutions are presented to improve histology image analyst productivity by easing the interaction between large image storage solutions and image analysis software; providing registration methods for these images; and proposing multiple morphology analysis tools optimized for handling vectorial annotations and to provide useful shape descriptors. This work has been entirely done aiming to provide a good basis tool-set to take existing digital pathology and converge into a richer discipline, namely computational pathology, where reproducible analysis is encouraged for the benefit of the research community. These tools have been tested in real scenarios and have been evaluated to demonstrate their performance with histology images. Of course, a lot of possible improvements and new developments are still possible. For Icytomine, the creation of helper plugins to simplify analysis by detecting target areas on images is possible. It is also possible to extend Icytomine to enable interaction with other large image storage solutions. Concerning large image registration, the method presented can be improved in order to use more competitive distance measure techniques better adapted to histological images. With the late virtual reality advances, it is not surprising to think on virtual 3-D slide stack visualization and analysis tools being developed in the coming years, integrating registration techniques to create virtual slide stacks aware of the general tissue shape, not only between neighboring slides. Furthermore, vectorial morphology can be improved by further optimizing internal operations and performing them in parallel on fast GPUs. In addition, further spatial analysis can be performed on vectorized shapes to help the procedure of point clustering detection around specific shapes by using morphological operations to identify distances and areas at given distances from shape boundaries. Further research must be performed to achieve straight skeletons on 3-D volumes given that, as of this day only voxel skeletonization has been achieved. Weighted straight skeletons also must be further studied to solve currently existing ambiguities and provide faster and more accurate solutions.

Bibliography

- [1] M. T. McCann, John A. Ozolek, C. A. Castro, B. Parvin, and J. Kovacevic. “Automated Histology Analysis: Opportunities for signal processing”. In: *IEEE Signal Processing Magazine* 32.1 (Jan. 2015), pp. 78–87. DOI: [10.1109/msp.2014.2346443](https://doi.org/10.1109/msp.2014.2346443).
- [2] St. Michael’s Hospital. *Histology Core*. 2018. URL: [http : / / stmichaelshospitalresearch . ca / staff - services / research - facilities/facilities/histology-core/](http://stmichaelshospitalresearch.ca/staff-services/research-facilities/facilities/histology-core/) (visited on 08/20/2019).
- [3] M.N. Gurcan, L.E. Boucheron, A. Can, A. Madabhushi, N.M. Rajpoot, and B. Yener. “Histopathological Image Analysis: A Review”. In: *IEEE Reviews in Biomedical Engineering* 2 (2009), pp. 147–171. DOI: [10.1109/rbme.2009.2034865](https://doi.org/10.1109/rbme.2009.2034865).
- [4] T. J. Fuchs and J. M. Buhmann. “Computational pathology: Challenges and promises for tissue analysis”. In: *Computerized Medical Imaging and Graphics* 35.7-8 (Oct. 2011), pp. 515–530. DOI: [10.1016/j.compmedimag.2011.02.006](https://doi.org/10.1016/j.compmedimag.2011.02.006).
- [5] M. D. Wang, S. Kothari, and J. H. Phan. “Eliminating tissue-fold artifacts in histopathological whole-slide images for improved image-based prediction of cancer grade”. In: *Journal of Pathology Informatics* 4.1 (2013), p. 22. DOI: [10.4103/2153-3539.117448](https://doi.org/10.4103/2153-3539.117448).
- [6] 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: *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE, Mar. 2011. DOI: [10.1109/isbi.2011.5872492](https://doi.org/10.1109/isbi.2011.5872492).
- [7] 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”. In: *IEEE Transactions on Medical Imaging* 32.4 (Apr. 2013), pp. 670–682. DOI: [10.1109/tmi.2012.2231420](https://doi.org/10.1109/tmi.2012.2231420).
- [8] U.-D. Braumann, J.-P. Kuska, J. Einkenkel, L.-C. Horn, M. Loffler, and M. Hockel. “Three-dimensional reconstruction and quantification of cervical carcinoma invasion fronts from histological serial sections”. In: *IEEE*

- Transactions on Medical Imaging* 24.10 (Oct. 2005), pp. 1286–1307. DOI: [10.1109/tmi.2005.855437](https://doi.org/10.1109/tmi.2005.855437).
- [9] P.J. Tadrous. “Digital stain separation for histological images”. In: *Journal of Microscopy* 240.2 (May 2010), pp. 164–172. DOI: [10.1111/j.1365-2818.2010.03390.x](https://doi.org/10.1111/j.1365-2818.2010.03390.x).
- [10] A. C. Ruifrok and D. A. Johnston. “Quantification of histochemical staining by color deconvolution”. In: *Anal. Quant. Cytol. Histol.* 23.4 (Aug. 2001), pp. 291–299.
- [11] 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: *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE, June 2009. DOI: [10.1109/isbi.2009.5193250](https://doi.org/10.1109/isbi.2009.5193250).
- [12] M. Gavrilovic, J. C. Azar, J. Lindblad, C. Wählby, E. Bengtsson, C. Busch, and I. Carlbom. “Blind Color Decomposition of Histological Images”. In: *IEEE Transactions on Medical Imaging* 32 (2013), pp. 983–994.
- [13] A. Vahadane, T. Peng, A. Sethi, S. Albarqouni, L. Wang, M. Baust, K. Steiger, A. M. Schlitter, I. Esposito, and N. Navab. “Structure-Preserving Color Normalization and Sparse Stain Separation for Histological Images”. In: *IEEE Transactions on Medical Imaging* 35.8 (Aug. 2016), pp. 1962–1971. DOI: [10.1109/tmi.2016.2529665](https://doi.org/10.1109/tmi.2016.2529665).
- [14] 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”. In: *Computerized Medical Imaging and Graphics* 35.7-8 (Oct. 2011), pp. 557–567. DOI: [10.1016/j.compmedimag.2011.01.010](https://doi.org/10.1016/j.compmedimag.2011.01.010).
- [15] B. Zitová and J. Flusser. “Image registration methods: a survey”. In: *Image and Vision Computing* 21.11 (Oct. 2003), pp. 977–1000. DOI: [10.1016/S0262-8856\(03\)00137-9](https://doi.org/10.1016/S0262-8856(03)00137-9).
- [16] M. I. Patel, V. K. Thakar, and S. K. Shah. “Image Registration of Satellite Images with Varying Illumination Level Using HOG Descriptor Based SURF”. In: *Procedia Computer Science* 93 (2016), pp. 382–388. DOI: [10.1016/j.procs.2016.07.224](https://doi.org/10.1016/j.procs.2016.07.224).
- [17] P. A. Bautista and Y. Yagi. “Improving the visualization and detection of tissue folds in whole slide images through color enhancement”. In: *Journal of Pathology Informatics* 1.1 (2010), p. 25. DOI: [10.4103/2153-3539.73320](https://doi.org/10.4103/2153-3539.73320).
- [18] M. Satyanarayanan, A. Goode, B. Gilbert, J. Harkes, and D. Jukic. “OpenSlide: A vendor-neutral software foundation for digital pathology”. In: *Journal of Pathology Informatics* 4.1 (2013), p. 27. DOI: [10.4103/2153-3539.119005](https://doi.org/10.4103/2153-3539.119005).

-
- [19] M. Linkert, C. T. Rueden, C. Allan, J.-M. Burel, W. Moore, A. Patterson, B. Loranger, J. Moore, C. Neves, D. MacDonald, A. Tarkowska, C. Sticco, E. Hill, M. Rossner, K. W. Eliceiri, and J. R. Swedlow. “Metadata matters: access to image data in the real world”. In: *The Journal of Cell Biology* 189.5 (May 2010), pp. 777–782. DOI: [10.1083/jcb.201004104](https://doi.org/10.1083/jcb.201004104).
- [20] R. Marée, B. Stévens, L. Rollus, N. Rocks, X. Moles 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”. In: *Diagnostic Pathology* 8.S1 (Sept. 2013). DOI: [10.1186/1746-1596-8-s1-s26](https://doi.org/10.1186/1746-1596-8-s1-s26).
- [21] C.-H. Huang, A. Veillard, L. Roux, N. Loménié, and D. Racoceanu. “Time-efficient sparse analysis of histopathological whole slide images”. In: *Computerized Medical Imaging and Graphics* 35.7-8 (Oct. 2011), pp. 579–591. DOI: [10.1016/j.compmedimag.2010.11.009](https://doi.org/10.1016/j.compmedimag.2010.11.009).
- [22] 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”. In: *Computerized Medical Imaging and Graphics* 35.7-8 (Oct. 2011), pp. 603–615. DOI: [10.1016/j.compmedimag.2011.02.005](https://doi.org/10.1016/j.compmedimag.2011.02.005).
- [23] 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”. In: *Medical Image Analysis* 14.4 (Aug. 2010), pp. 617–629. DOI: [10.1016/j.media.2010.04.007](https://doi.org/10.1016/j.media.2010.04.007).
- [24] F. de Chaumont, S. Dallongeville, N. Chenouard, N. Hervé, S. Pop, T. Provoost, V. Meas-Yedid, P. Pankajakshan, T. Lecomte, Y. Le Montagner, T. Lagache, A. Dufour, and J.-C. Olivo-Marin. “Icy: an open bioimage informatics platform for extended reproducible research”. In: *Nature Methods* 9.7 (June 2012), pp. 690–696. DOI: [10.1038/nmeth.2075](https://doi.org/10.1038/nmeth.2075).
- [25] 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”. In: *Pattern Recognition* 42.6 (June 2009), pp. 1080–1092. DOI: [10.1016/j.patcog.2008.10.035](https://doi.org/10.1016/j.patcog.2008.10.035).
- [26] H. Fatakdwala, J. Xu, A. Basavanahally, 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”. In: *IEEE Transactions on Biomedical Engineering* 57.7 (July 2010), pp. 1676–1689. DOI: [10.1109/tbme.2010.2041232](https://doi.org/10.1109/tbme.2010.2041232).

- [27] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam. “Improved Automatic Detection and Segmentation of Cell Nuclei in Histopathology Images”. In: *IEEE Transactions on Biomedical Engineering* 57.4 (Apr. 2010), pp. 841–852. DOI: [10.1109/tbme.2009.2035102](https://doi.org/10.1109/tbme.2009.2035102).
- [28] J. W. Han, T. P. Breckon, D. A. Randell, and G. Landini. “The application of support vector machine classification to detect cell nuclei for automated microscopy”. In: *Machine Vision and Applications* 23.1 (June 2010), pp. 15–24. DOI: [10.1007/s00138-010-0275-y](https://doi.org/10.1007/s00138-010-0275-y).
- [29] L. Roux, D. Racoceanu, N. Loménie, M. Kulikova, H. Irshad, J. Klossa, F. Capron, C. Genestie, G. Naour, and M. N. Gurcan. “Mitosis detection in breast cancer histological images An ICPR 2012 contest”. In: *Journal of Pathology Informatics* 4.1 (2013), p. 8. DOI: [10.4103/2153-3539.112693](https://doi.org/10.4103/2153-3539.112693).
- [30] G. Jiménez and D. Racoceanu. “Deep Learning for Semantic Segmentation vs. Classification in Computational Pathology: Application to Mitosis Analysis in Breast Cancer Grading”. In: *Frontiers in Bioengineering and Biotechnology* 7 (June 2019). DOI: [10.3389/fbioe.2019.00145](https://doi.org/10.3389/fbioe.2019.00145).
- [31] E. Ozdemir and C. Gunduz-Demir. “A Hybrid Classification Model for Digital Pathology Using Structural and Statistical Pattern Recognition”. In: *IEEE Transactions on Medical Imaging* 32.2 (Feb. 2013), pp. 474–483. DOI: [10.1109/tmi.2012.2230186](https://doi.org/10.1109/tmi.2012.2230186).
- [32] K. Belkacem-Boussaid, S. Samsi, G. Lozanski, and M.N. Gurcan. “Automatic detection of follicular regions in H&E images using iterative shape index”. In: *Computerized Medical Imaging and Graphics* 35.7-8 (Oct. 2011), pp. 592–602. DOI: [10.1016/j.compmedimag.2011.03.001](https://doi.org/10.1016/j.compmedimag.2011.03.001).
- [33] C. Chen, J. A. Ozolek, W. Wang, and G. K. Rohde. “A General System for Automatic Biomedical Image Segmentation Using Intensity Neighborhoods”. In: *International Journal of Biomedical Imaging* 2011 (2011), pp. 1–12. DOI: [10.1155/2011/606857](https://doi.org/10.1155/2011/606857).
- [34] M. T. McCann, D. G. Mixon, M. C. Fickus, C. A. Castro, J. A. Ozolek, and J. Kovacevic. “Images as Occlusions of Textures: A Framework for Segmentation”. In: *IEEE Transactions on Image Processing* 23.5 (May 2014), pp. 2033–2046. DOI: [10.1109/tip.2014.2307475](https://doi.org/10.1109/tip.2014.2307475).
- [35] 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”. In: *Pattern Recognition* 42.6 (June 2009), pp. 1104–1112. DOI: [10.1016/j.patcog.2008.07.007](https://doi.org/10.1016/j.patcog.2008.07.007).
- [36] A. C. Simsek, A. B. Tosun, C. Aykanat, C. Sokmensuer, and C. Gunduz-Demir. “Multilevel Segmentation of Histopathological Images Using Cooccurrence of Tissue Objects”. In: *IEEE Transactions on Biomedical Engineering* 59.6 (June 2012), pp. 1681–1690. DOI: [10.1109/tbme.2012.2191784](https://doi.org/10.1109/tbme.2012.2191784).

-
- [37] D. Acuna, H. Ling, A. Kar, and S. Fidler. “Efficient Interactive Annotation of Segmentation Datasets with Polygon-RNN++”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 2018. DOI: [10.1109/cvpr.2018.00096](https://doi.org/10.1109/cvpr.2018.00096).
- [38] Y.-H. Chang, G. Thibault, V. Azimi, B. Johnson, D. Jorgens, J. Link, A. Margolin, and J. W. Gray. “Quantitative analysis of histological tissue image based on cytological profiles and spatial statistics”. In: *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, Aug. 2016. DOI: [10.1109/embc.2016.7590914](https://doi.org/10.1109/embc.2016.7590914).
- [39] S. Kothari, J. H. Phan, A. N. Young, and M. D. Wang. “Histological image classification using biologically interpretable shape-based features”. In: *BMC Medical Imaging* 13.1 (Mar. 2013). DOI: [10.1186/1471-2342-13-9](https://doi.org/10.1186/1471-2342-13-9).
- [40] Y. Zhang, B. Zhang, F. Coenen, and W. Lu. “Breast cancer diagnosis from biopsy images with highly reliable random subspace classifier ensembles”. In: *Machine Vision and Applications* 24.7 (Oct. 2012), pp. 1405–1420. DOI: [10.1007/s00138-012-0459-8](https://doi.org/10.1007/s00138-012-0459-8).
- [41] S. Petushi, F. U. Garcia, M. M. Haber, C. Katsinis, and A. Tozeren. “Large-scale computations on histology images reveal grade-differentiating parameters for breast cancer”. In: *BMC Medical Imaging* 6.1 (Apr. 2006). DOI: [10.1186/1471-2342-6-14](https://doi.org/10.1186/1471-2342-6-14).
- [42] M. D. DiFranco, G. O’Hurley, E. W. Kay, R. W. G. Watson, and P. Cunningham. “Ensemble based system for whole-slide prostate cancer probability mapping using color texture features”. In: *Computerized Medical Imaging and Graphics* 35.7-8 (Oct. 2011), pp. 629–645. DOI: [10.1016/j.compmedimag.2010.12.005](https://doi.org/10.1016/j.compmedimag.2010.12.005).
- [43] K. Roy, D. Banik, D. Bhattacharjee, and M. Nasipuri. “Patch-based system for Classification of Breast Histology images using deep learning”. In: *Computerized Medical Imaging and Graphics* 71 (Jan. 2019), pp. 90–103. DOI: [10.1016/j.compmedimag.2018.11.003](https://doi.org/10.1016/j.compmedimag.2018.11.003).
- [44] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri. “NIH Image to ImageJ: 25 years of image analysis”. In: *Nature Methods* 9.7 (June 2012), pp. 671–675. DOI: [10.1038/nmeth.2089](https://doi.org/10.1038/nmeth.2089).
- [45] C. Daniel, F. Macary, M. García Rojo, J. Klossa, A. Laurinavičius, B. A Beckwith, and V. Della Mea. “Recent advances in standards for collaborative Digital Anatomic Pathology”. In: *Diagnostic Pathology* 6.S1 (Mar. 2011). DOI: [10.1186/1746-1596-6-s1-s17](https://doi.org/10.1186/1746-1596-6-s1-s17).
- [46] V. Della Mea. “25 years of telepathology research: a bibliometric analysis”. In: *Diagnostic Pathology* 6.S1 (Mar. 2011). DOI: [10.1186/1746-1596-6-s1-s26](https://doi.org/10.1186/1746-1596-6-s1-s26).

- [47] M. M. Triola and W. J. Holloway. “Enhanced virtual microscopy for collaborative education”. In: *BMC Medical Education* 11.1 (Jan. 2011). DOI: [10.1186/1472-6920-11-4](https://doi.org/10.1186/1472-6920-11-4).
- [48] W.-K. Jeong, J. Schneider, S. G. Turney, B. E. Faulkner-Jones, D. Meyer, R. Westermann, R. C. Reid, J. Lichtman, and H. Pfister. “Interactive Histology of Large-Scale Biomedical Image Stacks”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (Nov. 2010), pp. 1386–1395. DOI: [10.1109/tvcg.2010.168](https://doi.org/10.1109/tvcg.2010.168).
- [49] R. Maree, S. Dallongeville, J.-C. Olivo-Marin, and V. Meas-Yedid. “An approach for detection of glomeruli in multisite digital pathology”. In: *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*. IEEE, Apr. 2016. DOI: [10.1109/isbi.2016.7493442](https://doi.org/10.1109/isbi.2016.7493442).
- [50] D. F. Gonzalez Obando, D. Mandache, J.-C. Olivo-Marin, and V. Meas-Yedid. “Icytomine: A User-Friendly Tool for Integrating Workflows on Whole Slide Images”. In: *Digital Pathology*. Springer International Publishing, 2019, pp. 181–189. DOI: [10.1007/978-3-030-23937-4_21](https://doi.org/10.1007/978-3-030-23937-4_21).
- [51] M. Gadermayr, A.-K. Dombrowski, B. M. Klinkhammer, P. Boor, and D. Merhof. “CNN Cascades for Segmenting Whole Slide Images of the Kidney”. In: *CoRR* abs/1708.00251 (2017). arXiv: [1708.00251](https://arxiv.org/abs/1708.00251).
- [52] M. Gadermayr, D. Eschweiler, A. Jeevanesan, B. M. Klinkhammer, P. Boor, and D. Merhof. “Segmenting renal whole slide images virtually without training data”. In: *Computers in Biology and Medicine* 90 (2017), pp. 88–97. ISSN: 18790534. DOI: [10.1016/j.combiomed.2017.09.014](https://doi.org/10.1016/j.combiomed.2017.09.014).
- [53] J. N. Marsh, M. K. Matlock, S. Kudose, T.-C. Liu, T. S. Stappenbeck, J. P. Gaut, and S. Joshua Swamidass. “Deep Learning Global Glomerulosclerosis in Transplant Kidney Frozen Sections”. In: *IEEE Transactions on Medical Imaging* 37.12 (Dec. 2018), pp. 2718–2728. DOI: [10.1109/tmi.2018.2851150](https://doi.org/10.1109/tmi.2018.2851150).
- [54] A. W. Fitzgibbon, M. Pilu, and R. B. Fisher. “Direct least squares fitting of ellipses”. In: *Proceedings of the 13th International Conference on Pattern Recognition, 1996. ICPR 1996*. IEEE, 1996. DOI: [10.1109/icpr.1996.546029](https://doi.org/10.1109/icpr.1996.546029).
- [55] P. F. Felzenszwalb and D. P. Huttenlocher. “Efficient Graph-Based Image Segmentation”. In: *International Journal of Computer Vision* 59.2 (Sept. 2004), pp. 167–181. DOI: [10.1023/b:visi.0000022288.19776.77](https://doi.org/10.1023/b:visi.0000022288.19776.77).
- [56] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. “Rethinking the Inception Architecture for Computer Vision”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016. DOI: [10.1109/cvpr.2016.308](https://doi.org/10.1109/cvpr.2016.308).

-
- [57] K. He, X. Zhang, S. Ren, and J. Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016. DOI: [10.1109/cvpr.2016.90](https://doi.org/10.1109/cvpr.2016.90).
- [58] K. Simonyan and A. Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. arXiv: [1409.1556](https://arxiv.org/abs/1409.1556).
- [59] K. He, G. Gkioxari, P. Dollar, and R. Girshick. “Mask R-CNN”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2017. DOI: [10.1109/iccv.2017.322](https://doi.org/10.1109/iccv.2017.322).
- [60] W. Abdulla. *Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow*. 2017. URL: https://github.com/matterport/Mask_RCNN (visited on 08/22/2019).
- [61] T. Tsujikawa, S. Kumar, R. N. Borkar, V. Azimi, G. Thibault, Y. H. Chang, A. Balter, R. Kawashima, G. Choe, D. Sauer, E. El Rassi, D. R. Clayburgh, M. F. Kulesz-Martin, E. R. Lutz, L. Zheng, E. M. Jaffee, P. Leyshock, A. A. Margolin, M. Mori, J. W. Gray, P. W. Flint, and L. M. Coussens. “Quantitative Multiplex Immunohistochemistry Reveals Myeloid-Inflamed Tumor-Immune Complexity Associated with Poor Prognosis”. In: *Cell Reports* 19.1 (Apr. 2017), pp. 203–217. DOI: [10.1016/j.celrep.2017.03.037](https://doi.org/10.1016/j.celrep.2017.03.037).
- [62] A. W. Toga and P. M. Thompson. “The role of image registration in brain mapping”. In: *Image and Vision Computing* 19.1-2 (Jan. 2001), pp. 3–24. DOI: [10.1016/S0262-8856\(00\)00055-X](https://doi.org/10.1016/S0262-8856(00)00055-X).
- [63] Z. Tu, S. Zheng, and A. Yuille. “Shape matching and registration by data-driven EM”. In: *Computer Vision and Image Understanding* 109.3 (Mar. 2008), pp. 290–304. DOI: [10.1016/j.cviu.2007.04.004](https://doi.org/10.1016/j.cviu.2007.04.004).
- [64] C.-W. Wang, S.-M. Ka, and A. Chen. “Robust image registration of biological microscopic images”. In: *Scientific Reports* 4.1 (Aug. 2014). DOI: [10.1038/srep06050](https://doi.org/10.1038/srep06050).
- [65] L. Rai, J. P. Helferty, and W. E. Higgins. “Combined video tracking and image-video registration for continuous bronchoscopic guidance”. In: *International Journal of Computer Assisted Radiology and Surgery* 3.3-4 (June 2008), pp. 315–329. DOI: [10.1007/s11548-008-0241-6](https://doi.org/10.1007/s11548-008-0241-6).
- [66] N. R. Pal and S. K. Pal. “A review on image segmentation techniques”. In: *Pattern Recognition* 26.9 (Sept. 1993), pp. 1277–1294. DOI: [10.1016/0031-3203\(93\)90135-j](https://doi.org/10.1016/0031-3203(93)90135-j).
- [67] L. G. Brown. “A survey of image registration techniques”. In: *ACM Computing Surveys* 24.4 (Dec. 1992), pp. 325–376. DOI: [10.1145/146370.146374](https://doi.org/10.1145/146370.146374).
- [68] D. G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pp. 91–110. DOI: [10.1023/b:visi.0000029664.99615.94](https://doi.org/10.1023/b:visi.0000029664.99615.94).

- [69] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. “Speeded-Up Robust Features (SURF)”. In: *Computer Vision and Image Understanding* 110.3 (June 2008), pp. 346–359. DOI: [10.1016/j.cviu.2007.09.014](https://doi.org/10.1016/j.cviu.2007.09.014).
- [70] J. Canny. “A Computational Approach to Edge Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (Nov. 1986), pp. 679–698. DOI: [10.1109/tpami.1986.4767851](https://doi.org/10.1109/tpami.1986.4767851).
- [71] R. Maini and Dr. H. Aggarwal. “Study and Comparison of Various Image Edge Detection Techniques”. In: *International Journal of Image Processing (IJIP)* 3.1 (2010).
- [72] K. Sharma and A. Goyal. “Classification based survey of image registration methods”. In: *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*. IEEE, July 2013. DOI: [10.1109/icccnt.2013.6726741](https://doi.org/10.1109/icccnt.2013.6726741).
- [73] J. P. Lewis. *Fast Normalized Cross-Correlation*. Tech. rep. Available at <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.6062> (Visited on 22/08/2019). Industrial Light & Magic, 1996.
- [74] R. N. Bracewell and R. Bracewell. *The Fourier Transform and Its Applications*. Electrical engineering series. McGraw Hill, 2000. ISBN: 9780073039381.
- [75] P. Viola and W. M. Wells III. “Alignment by Maximization of Mutual Information”. In: 24.2 (1997), pp. 137–154. DOI: [10.1023/a:1007958904918](https://doi.org/10.1023/a:1007958904918).
- [76] A. I. Ramírez Galarza and J. Seade. In: Birkhäuser Basel, pp. 1–73. DOI: [10.1007/978-3-7643-7518-8_1](https://doi.org/10.1007/978-3-7643-7518-8_1).
- [77] A. I. Ramírez Galarza and J. Seade. In: Birkhäuser Basel, pp. 75–89. DOI: [10.1007/978-3-7643-7518-8_2](https://doi.org/10.1007/978-3-7643-7518-8_2).
- [78] W. R. Crum, T. Hartkens, and D. L. G. Hill. “Non-rigid image registration: theory and practice”. In: *The British Journal of Radiology* 77.suppl_2 (Dec. 2004), S140–S153. DOI: [10.1259/bjr/25329214](https://doi.org/10.1259/bjr/25329214).
- [79] M. Unser, A. Aldroubi, and M. Eden. “B-spline signal processing. I. Theory”. In: *IEEE Transactions on Signal Processing* 41.2 (1993), pp. 821–833. DOI: [10.1109/78.193220](https://doi.org/10.1109/78.193220).
- [80] M. Unser, A. Aldroubi, and M. Eden. “B-spline signal processing. II. Efficiency design and applications”. In: *IEEE Transactions on Signal Processing* 41.2 (1993), pp. 834–848. DOI: [10.1109/78.193221](https://doi.org/10.1109/78.193221).
- [81] C.-W. Wang and H.-C. Chen. “Improved image alignment method in application to X-ray images and biological images”. In: *Bioinformatics* 29.15 (May 2013), pp. 1879–1887. DOI: [10.1093/bioinformatics/btt309](https://doi.org/10.1093/bioinformatics/btt309).

- [82] I. Arganda-Carreras, C. O. S. Sorzano, R. Marabini, J. M. Carazo, C. Ortiz de Solorzano, and J. Kybic. “Consistent and Elastic Registration of Histological Sections Using Vector-Spline Regularization”. In: *Computer Vision Approaches to Medical Image Analysis*. Springer Berlin Heidelberg, 2006, pp. 85–95. DOI: [10.1007/11889762_8](https://doi.org/10.1007/11889762_8).
- [83] R. Szeliski and J. Coughlan. In: *International Journal of Computer Vision* 22.3 (1997), pp. 199–218. DOI: [10.1023/a:1007996332012](https://doi.org/10.1023/a:1007996332012).
- [84] A. Dufour, V. Meas-Yedid, A. Grassart, and J.-C. Olivo-Marin. “Automated quantification of cell endocytosis using active contours and wavelets”. In: *Proceedings of the 19th International Conference on Pattern Recognition, 2008. ICPR 2008*. IEEE, Dec. 2008. DOI: [10.1109/icpr.2008.4761748](https://doi.org/10.1109/icpr.2008.4761748).
- [85] N. Otsu. “A Threshold Selection Method from Gray-Level Histograms”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (Jan. 1979), pp. 62–66. DOI: [10.1109/tsmc.1979.4310076](https://doi.org/10.1109/tsmc.1979.4310076).
- [86] L.-K. Huang and M.-J. J. Wang. “Image thresholding by minimizing the measures of fuzziness”. In: *Pattern Recognition* 28.1 (Jan. 1995), pp. 41–51. DOI: [10.1016/0031-3203\(94\)e0043-k](https://doi.org/10.1016/0031-3203(94)e0043-k).
- [87] A. G. Shanbhag. “Utilization of Information Measure as a Means of Image Thresholding”. In: *CVGIP: Graphical Models and Image Processing* 56.5 (Sept. 1994), pp. 414–419. DOI: [10.1006/cgip.1994.1037](https://doi.org/10.1006/cgip.1994.1037).
- [88] E. W. Weisstein. *B-Spline*. From *MathWorld - A Wolfram Web Resource*. 2016. URL: <http://mathworld.wolfram.com/B-Spline.html> (visited on 07/11/2017).
- [89] C. O. S. Sorzano, P. Thevenaz, and M. Unser. “Elastic Registration of Biological Images Using Vector-Spline Regularization”. In: *IEEE Transactions on Biomedical Engineering* 52.4 (Apr. 2005), pp. 652–663. DOI: [10.1109/tbme.2005.844030](https://doi.org/10.1109/tbme.2005.844030).
- [90] P. Thevenaz, U. E. Ruttimann, and M. Unser. “A pyramid approach to subpixel registration based on intensity”. In: *IEEE Transactions on Image Processing* 7.1 (1998), pp. 27–41. DOI: [10.1109/83.650848](https://doi.org/10.1109/83.650848).
- [91] J. Borovec, A. Munoz-Barrutia, and J. Kybic. “Benchmarking of Image Registration Methods for Differently Stained Histological Slides”. In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, Oct. 2018. DOI: [10.1109/icip.2018.8451040](https://doi.org/10.1109/icip.2018.8451040).
- [92] J. Borovec, J. Kybic, and A. Muñoz-Barrutia. “Automatic Non-rigid Histological Image Registration challenge”. en. In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)* (2019). DOI: [10.13140/rg.2.2.12974.77126/2](https://doi.org/10.13140/rg.2.2.12974.77126/2).
- [93] J. Lotz, N. Weiss, and S. Heldmann. *Robust, fast and accurate: a 3-step method for automatic histological image registration*. 2019. arXiv: [1903.12063](https://arxiv.org/abs/1903.12063).

- [94] D. F. Gonzalez Obando, A. Frafjord, I. Øynebråten, A. Corthay, J.-C. Olivo-Marin, and V. Meas-Yedid. “Multi-staining registration of large histology images”. In: *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*. IEEE, Apr. 2017. DOI: [10.1109/isbi.2017.7950534](https://doi.org/10.1109/isbi.2017.7950534).
- [95] B. J. Rossetti, F. Wang, P. Zhang, G. T., D. J. Brat, and J. Kong. “Dynamic registration for gigapixel serial whole slide images”. In: *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*. IEEE, Apr. 2017. DOI: [10.1109/isbi.2017.7950552](https://doi.org/10.1109/isbi.2017.7950552).
- [96] Y. Song, D. Treanor, A. J. Bulpitt, and D. R. Magee. “3D reconstruction of multiple stained histology images”. In: *Journal of Pathology Informatics* 4.2 (2013), p. 7. DOI: [10.4103/2153-3539.109864](https://doi.org/10.4103/2153-3539.109864).
- [97] N. M. Noor, M. M. Nor, A. Abdullah, and R. K. Zahari. “Geospatial technology approaches in urban morphology for resilient urban governance”. In: *Planning Malaysia Journal* 12.3 (July 2014). DOI: [10.21837/pmjournal.v12.i3.128](https://doi.org/10.21837/pmjournal.v12.i3.128).
- [98] T. Bechter, K. Baumann, S. Birk, F. Bolik, W. Graf, and F. Pletterbauer. “LaRiMo - A simple and efficient GIS-based approach for large-scale morphological assessment of large European rivers”. In: *Science of The Total Environment* 628-629 (July 2018), pp. 1191–1199. DOI: [10.1016/j.scitotenv.2018.02.084](https://doi.org/10.1016/j.scitotenv.2018.02.084).
- [99] H. Alhéritière, W. Amaïeur, F. Cloppet, C. Kurtz, J.-M. Ogier, and N. Vincent. “Straight Line Reconstruction for Fully Materialized Table Extraction in Degraded Document Images”. In: *Discrete Geometry for Computer Imagery - 21st IAPR International Conference, DGCI 2019, Marne-la-Vallée, France, March 26-28, 2019, Proceedings*. 2019, pp. 317–329. DOI: [10.1007/978-3-030-14085-4_25](https://doi.org/10.1007/978-3-030-14085-4_25).
- [100] D. Tsygankov, C. G. Bilancia, E. A. Vitriol, K. M. Hahn, M. Peifer, and T. C. Elston. “CellGeo: A computational platform for the analysis of shape changes in cells with complex geometries”. In: *The Journal of Cell Biology* 204.3 (Feb. 2014), pp. 443–460. DOI: [10.1083/jcb.201306067](https://doi.org/10.1083/jcb.201306067).
- [101] B. D. Ripley. “The second-order analysis of stationary point processes”. In: *Journal of Applied Probability* 13.2 (1976), pp. 255–266. DOI: [10.2307/3212829](https://doi.org/10.2307/3212829).
- [102] T. Lagache, A. Grassart, S. Dallongeville, O. Faklaris, N. Sauvonnnet, A. Dufour, L. Danglot, and J.-C. Olivo-Marin. “Mapping molecular assemblies with fluorescence microscopy and object-based spatial statistics”. In: *Nature Communications* 9.1 (Feb. 2018). DOI: [10.1038/s41467-018-03053-x](https://doi.org/10.1038/s41467-018-03053-x).
- [103] M. Clément. “Modélisation et apprentissage de relations spatiales pour la reconnaissance et l’interprétation d’images”. Université Sorbonne Paris Cité. PhD thesis. 2017.

-
- [104] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. “Skeleton based shape matching and retrieval”. In: *2003 Shape Modeling International*. IEEE Comput. Soc. DOI: [10.1109/smi.2003.1199609](https://doi.org/10.1109/smi.2003.1199609).
- [105] A. Brennecke and T. Isenberg. “3D Shape Matching Using Skeleton Graphs”. In: *Simulation und Visualisierung 2004 (SimVis 2004) 4-5 März 2004, Magdeburg*. 2004, pp. 299–310.
- [106] P.-Y. Chiang and C.-C. Jay Kuo. “Voxel-based shape decomposition for feature-preserving 3D thumbnail creation”. In: *Journal of Visual Communication and Image Representation* 23.1 (Jan. 2012), pp. 1–11. DOI: [10.1016/j.jvcir.2011.07.008](https://doi.org/10.1016/j.jvcir.2011.07.008).
- [107] G. Malandain, G. Bertrand, and N. Ayache. “Topological segmentation of discrete surfaces”. In: *International Journal of Computer Vision* 10.2 (Apr. 1993), pp. 183–197. DOI: [10.1007/bf01420736](https://doi.org/10.1007/bf01420736).
- [108] D. Reniers and A. Telea. “Skeleton-based Hierarchical Shape Segmentation”. In: *IEEE International Conference on Shape Modeling and Applications 2007 (SMI '07)*. IEEE, June 2007. DOI: [10.1109/smi.2007.33](https://doi.org/10.1109/smi.2007.33).
- [109] P. K. Saha and B. B. Chaudhuri. “3D Digital Topology under Binary Transformation with Applications”. In: *Computer Vision and Image Understanding* 63.3 (May 1996), pp. 418–429. DOI: [10.1006/cviu.1996.0032](https://doi.org/10.1006/cviu.1996.0032).
- [110] L. Serino, G. Sanniti di Baja, and C. Arcelli. “Using the Skeleton for 3D Object Decomposition”. In: *Image Analysis*. Springer Berlin Heidelberg, 2011, pp. 447–456. DOI: [10.1007/978-3-642-21227-7_42](https://doi.org/10.1007/978-3-642-21227-7_42).
- [111] N. D. Cornea, D. Silver, X. Yuan, and R. Balasubramanian. “Computing hierarchical curve-skeletons of 3D objects”. In: *The Visual Computer* 21.11 (Sept. 2005), pp. 945–955. DOI: [10.1007/s00371-005-0308-0](https://doi.org/10.1007/s00371-005-0308-0).
- [112] L. Huang and A. Bijaoui. “Astronomical image data compression by morphological skeleton transformation”. In: *Experimental Astronomy* 1.5 (1990), pp. 311–327. DOI: [10.1007/bf00454328](https://doi.org/10.1007/bf00454328).
- [113] J.-M. Lien, G. Kurillo, and R. Bajcsy. “Skeleton-Based Data Compression for Multi-camera Tele-Immersion System”. In: *Advances in Visual Computing. ISVC 2007*. Springer Berlin Heidelberg, pp. 714–723. DOI: [10.1007/978-3-540-76858-6_69](https://doi.org/10.1007/978-3-540-76858-6_69).
- [114] G. Boccignone, A. Chianese, L. P. Cordella, and A. Marcelli. “Recovering dynamic information from static handwriting”. In: *Pattern Recognition* 26.3 (Mar. 1993), pp. 409–418. DOI: [10.1016/0031-3203\(93\)90168-v](https://doi.org/10.1016/0031-3203(93)90168-v).
- [115] H. Bunke, M. Roth, and E. G. Schukat-Talamazzini. “Off-line cursive handwriting recognition using hidden markov models”. In: *Pattern Recognition* 28.9 (Sept. 1995), pp. 1399–1413. DOI: [10.1016/0031-3203\(95\)00013-p](https://doi.org/10.1016/0031-3203(95)00013-p).

- [116] V. Pervouchine and G. Leedham. “Extraction and analysis of forensic document examiner features used for writer identification”. In: *Pattern Recognition* 40.3 (Mar. 2007), pp. 1004–1013. DOI: [10.1016/j.patcog.2006.08.008](https://doi.org/10.1016/j.patcog.2006.08.008).
- [117] Y. Fridman, S. Pizer, S. Aylward, and E. Bullitt. “Extracting branching tubular object geometry via cores”. In: *Medical Image Analysis* 8.3 (Sept. 2004), pp. 169–176. DOI: [10.1016/j.media.2004.06.017](https://doi.org/10.1016/j.media.2004.06.017).
- [118] D. A. Gómez Betancur, A. Fabijańska, L. Flórez-Valencia, A. Morales Pinzón, E. E. Dávila Serrano, J.-C. Richard, M. Orkisz, and M. Hernández Hoyos. “Airway Segmentation, Skeletonization, and Tree Matching to Improve Registration of 3D CT Images with Large Opacities in the Lungs”. In: *Computer Vision and Graphics*. Springer International Publishing, 2016, pp. 395–407. DOI: [10.1007/978-3-319-46418-3_35](https://doi.org/10.1007/978-3-319-46418-3_35).
- [119] P. K. Saha, G. Borgefors, and G. Sanniti di Baja. “Skeletonization and its applications – a review”. In: *Skeletonization*. Elsevier, 2017, pp. 3–42. DOI: [10.1016/b978-0-08-101291-8.00002-x](https://doi.org/10.1016/b978-0-08-101291-8.00002-x).
- [120] K. Siddiqi and S. M. Pizer, eds. *Medial Representations*. Springer Netherlands, 2008. DOI: [10.1007/978-1-4020-8658-8](https://doi.org/10.1007/978-1-4020-8658-8).
- [121] M. Brady and H. Asada. “Smoothed Local Symmetries and Their Implementation”. In: *The International Journal of Robotics Research* 3.3 (Sept. 1984), pp. 36–61. DOI: [10.1177/027836498400300302](https://doi.org/10.1177/027836498400300302).
- [122] M. Leyton. “Symmetry-curvature duality”. In: *Computer Vision, Graphics, and Image Processing* 38.3 (June 1987), pp. 327–341. DOI: [10.1016/0734-189x\(87\)90117-4](https://doi.org/10.1016/0734-189x(87)90117-4).
- [123] N. Amenta, S. Choi, and R. Krishna Kolluri. “The power crust, unions of balls, and the medial axis transform”. In: *Computational Geometry* 19.2-3 (July 2001), pp. 127–153. DOI: [10.1016/s0925-7721\(01\)00017-7](https://doi.org/10.1016/s0925-7721(01)00017-7).
- [124] J. W. Brandt and V. R. Algazi. “Continuous skeleton computation by Voronoi diagram”. In: *CVGIP: Image Understanding* 55.3 (May 1992), pp. 329–338. DOI: [10.1016/1049-9660\(92\)90030-7](https://doi.org/10.1016/1049-9660(92)90030-7).
- [125] R. Ogniewicz and M. Ilg. “Voronoi skeletons: theory and applications”. In: *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Comput. Soc. Press. DOI: [10.1109/cvpr.1992.223226](https://doi.org/10.1109/cvpr.1992.223226).
- [126] R. L. Ogniewicz and O. Kübler. “Hierarchic Voronoi skeletons”. In: *Pattern Recognition* 28.3 (Mar. 1995), pp. 343–359. DOI: [10.1016/0031-3203\(94\)00105-u](https://doi.org/10.1016/0031-3203(94)00105-u).

-
- [127] M. Naf, O. Kubler, R. Kikinis, M. E. Shenton, and G. Szekely. “Characterization and recognition of 3D organ shape in medical image analysis using skeletonization”. In: *Proceedings of the Workshop on Mathematical Methods in Biomedical Image Analysis*. IEEE, 1996. DOI: [10.1109/mmmbia.1996.534066](https://doi.org/10.1109/mmmbia.1996.534066).
- [128] M. Schmitt. “Some examples of algorithms analysis in computational geometry by means of mathematical morphological techniques”. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1989, pp. 225–246. DOI: [10.1007/3-540-51683-2_33](https://doi.org/10.1007/3-540-51683-2_33).
- [129] D. Attali and A. Montanvert. “Computing and Simplifying 2D and 3D Continuous Skeletons”. In: *Computer Vision and Image Understanding* 67.3 (Sept. 1997), pp. 261–273. DOI: [10.1006/cviu.1997.0536](https://doi.org/10.1006/cviu.1997.0536).
- [130] T. K. Dey and W. Zhao. “Approximating the Medial Axis from the Voronoi Diagram with a Convergence Guarantee”. In: *Algorithmica* 38.1 (Oct. 2003), pp. 179–200. DOI: [10.1007/s00453-003-1049-y](https://doi.org/10.1007/s00453-003-1049-y).
- [131] E. C. Sherbrooke, N. M. Patrikalakis, and E. Brisson. “An algorithm for the medial axis transform of 3D polyhedral solids”. In: *IEEE Transactions on Visualization and Computer Graphics* 2.1 (Mar. 1996), pp. 44–61. DOI: [10.1109/2945.489386](https://doi.org/10.1109/2945.489386).
- [132] A. C. Jalba, J. Kustra, and A. C. Telea. “Surface and Curve Skeletonization of Large 3D Models on the GPU”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.6 (June 2013), pp. 1495–1508. DOI: [10.1109/tpami.2012.212](https://doi.org/10.1109/tpami.2012.212).
- [133] A. Bucksch and R. Lindenbergh. “CAMPINO — A skeletonization method for point cloud processing”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 63.1 (Jan. 2008), pp. 115–127. DOI: [10.1016/j.isprsjprs.2007.10.004](https://doi.org/10.1016/j.isprsjprs.2007.10.004).
- [134] F. Leymarie and M. D. Levine. “Simulating the grassfire transform using an active contour model”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.1 (1992), pp. 56–75. DOI: [10.1109/34.107013](https://doi.org/10.1109/34.107013).
- [135] K. Siddiqi, S. Bouix, A. Tannenbaum, and S.W. Zucker. “The Hamilton-Jacobi skeleton”. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. IEEE, 1999. DOI: [10.1109/iccv.1999.790307](https://doi.org/10.1109/iccv.1999.790307).
- [136] C. Aslan, A. Erdem, E. Erdem, and S. Tari. “Disconnected Skeleton: Shape at Its Absolute Scale”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.12 (Dec. 2008), pp. 2188–2203. DOI: [10.1109/tpami.2007.70842](https://doi.org/10.1109/tpami.2007.70842).
- [137] Z.S.G. Tari, J. Shah, and H. Pien. “Extraction of Shape Skeletons from Grayscale Images”. In: *Computer Vision and Image Understanding* 66.2 (May 1997), pp. 133–146. DOI: [10.1006/cviu.1997.0612](https://doi.org/10.1006/cviu.1997.0612).

- [138] M.S. Hassouna and A.A. Farag. “Variational Curve Skeletons Using Gradient Vector Flow”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.12 (Dec. 2009), pp. 2257–2274. DOI: [10.1109/tpami.2008.271](https://doi.org/10.1109/tpami.2008.271).
- [139] T.C. Lee, R.L. Kashyap, and C.N. Chu. “Building Skeleton Models via 3-D Medial Surface Axis Thinning Algorithms”. In: *CVGIP: Graphical Models and Image Processing* 56.6 (Nov. 1994), pp. 462–478. DOI: [10.1006/cgip.1994.1042](https://doi.org/10.1006/cgip.1994.1042).
- [140] C. Lohou and G. Bertrand. “A New 3D 12-Subiteration Thinning Algorithm Based on P-Simple Points”. In: *Electronic Notes in Theoretical Computer Science* 46 (Aug. 2001), pp. 33–52. DOI: [10.1016/s1571-0661\(04\)80977-0](https://doi.org/10.1016/s1571-0661(04)80977-0).
- [141] C. Lohou and G. Bertrand. “A 3D 6-subiteration curve thinning algorithm based on P-simple points”. In: *Discrete Applied Mathematics* 151.1-3 (Oct. 2005), pp. 198–228. DOI: [10.1016/j.dam.2005.02.030](https://doi.org/10.1016/j.dam.2005.02.030).
- [142] P.K. Saha, B.B. Chaudhuri, and D. Dutta Majumder. “A new shape preserving parallel thinning algorithm for 3D digital images”. In: *Pattern Recognition* 30.12 (Dec. 1997), pp. 1939–1955. DOI: [10.1016/s0031-3203\(97\)00016-2](https://doi.org/10.1016/s0031-3203(97)00016-2).
- [143] M.P. Deseilligny, G. Stamon, and C.Y. Suen. “Veinerization: a new shape description for flexible skeletonization”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.5 (May 1998), pp. 505–521. DOI: [10.1109/34.682180](https://doi.org/10.1109/34.682180).
- [144] Y.F. Tsao and K.S. Fu. “A parallel thinning algorithm for 3-D pictures”. In: *Computer Graphics and Image Processing* 17.4 (Dec. 1981), pp. 315–331. DOI: [10.1016/0146-664x\(81\)90011-3](https://doi.org/10.1016/0146-664x(81)90011-3).
- [145] C. Arcelli and G. Sanniti Di Baja. “A Width-Independent Fast Thinning Algorithm”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-7.4 (July 1985), pp. 463–474. DOI: [10.1109/tpami.1985.4767685](https://doi.org/10.1109/tpami.1985.4767685).
- [146] G. Borgefors, I. Nyström, and G. Sanniti Di Baja. “Computing skeletons in three dimensions”. In: *Pattern Recognition* 32.7 (July 1999), pp. 1225–1236. DOI: [10.1016/s0031-3203\(98\)00082-x](https://doi.org/10.1016/s0031-3203(98)00082-x).
- [147] C. Pudney. “Distance-Ordered Homotopic Thinning: A Skeletonization Algorithm for 3D Digital Images”. In: *Computer Vision and Image Understanding* 72.3 (Dec. 1998), pp. 404–413. DOI: [10.1006/cviu.1998.0680](https://doi.org/10.1006/cviu.1998.0680).
- [148] D. S. Fritsch, S. M. Pizer, B. S. Morse, D. H. Eberly, and A. Liu. “The multiscale medial axis and its applications in image registration”. In: *Pattern Recognition Letters* 15.5 (May 1994), pp. 445–452. DOI: [10.1016/0167-8655\(94\)90135-x](https://doi.org/10.1016/0167-8655(94)90135-x).

-
- [149] G. Borgefors and G. Sanniti di Baja. “Skeletonizing the distance transform on the hexagonal grid”. In: *Proceedings of the 9th International Conference on Pattern Recognition, 1988. ICPR 1988*. IEEE, 1988. DOI: [10.1109/icpr.1988.28278](https://doi.org/10.1109/icpr.1988.28278).
- [150] R. Strand. “Surface skeletons in grids with non-cubic voxels”. In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004*. IEEE, 2004. DOI: [10.1109/icpr.2004.1334195](https://doi.org/10.1109/icpr.2004.1334195).
- [151] S. K. Pal and A. Rosenfeld. “A fuzzy medial axis transformation based on fuzzy disks”. In: *Pattern Recognition Letters* 12.10 (Oct. 1991), pp. 585–590. DOI: [10.1016/0167-8655\(91\)90011-a](https://doi.org/10.1016/0167-8655(91)90011-a).
- [152] P. K. Saha, F. W. Wehrli, and B. R. Gomberg. “Fuzzy Distance Transform: Theory, Algorithms, and Applications”. In: *Computer Vision and Image Understanding* 86.3 (June 2002), pp. 171–190. DOI: [10.1006/cviu.2002.0974](https://doi.org/10.1006/cviu.2002.0974).
- [153] O. Aichholzer and F. Aurenhammer. “Straight skeletons for general polygonal figures in the plane”. In: *Computing and Combinatorics*. Ed. by J.-Y. Cai and C. K. Wong. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 117–126. ISBN: 978-3-540-68461-9.
- [154] O. Aichholzer and F. Aurenhammer. “Straight skeletons for general polygonal figures in the plane”. In: *Voronoi’s Impact on Modern Sciences II*. Ed. by A. M. Samoilenko. Vol. 21. Kiev, Ukraine: Proc. Institute of Mathematics of the National Academy of Sciences of Ukraine, 1998, pp. 7–21.
- [155] D. Eppstein and J. Erickson. “Raising Roofs, Crashing Cycles, and Playing Pool: Applications of a Data Structure for Finding Pairwise Interactions”. In: *Proceedings of the Fourteenth Annual Symposium on Computational Geometry*. SCG ’98. Minneapolis, Minnesota, USA: ACM, 1998, pp. 58–67. ISBN: 0-89791-973-4. DOI: [10.1145/276884.276891](https://doi.org/10.1145/276884.276891).
- [156] S.-W. Cheng and A. Vigneron. “Motorcycle Graphs and Straight Skeletons”. In: *Algorithmica* 47.2 (Jan. 2007), pp. 159–182. DOI: [10.1007/s00453-006-1229-7](https://doi.org/10.1007/s00453-006-1229-7).
- [157] P. Felkel and S. Obdržálek. “Improvement of Oliva’s Algorithm for Surface Reconstruction from Contours”. In: *In Proc. 15th Spring Conf. Comp. Graphics*. 1999, pp. 254–263.
- [158] F. Cacciola. *A CGAL implementation of the Straight Skeleton of a Simple 2D Polygon with Holes*. Tech. rep. SciSoft, 2004.
- [159] S. Huber and M. Held. “A Fast Straight-Skeleton Algorithm Based on Generalized Motorcycle Graphs”. In: *International Journal of Computational Geometry & Applications* 22.05 (Oct. 2012), pp. 471–498. DOI: [10.1142/s0218195912500124](https://doi.org/10.1142/s0218195912500124).

- [160] S. Huber and M. Held. “Motorcycle Graphs: Stochastic Properties Motivate an Efficient Yet Simple Implementation”. In: *J. Exp. Algorithmics* 16 (Aug. 2011), 1.3:1.1–1.3:1.17. ISSN: 1084-6654. DOI: [10.1145/1963190.2019578](https://doi.org/10.1145/1963190.2019578).
- [161] L. M. Vincent. “Efficient computation of various types of skeletons”. In: *Medical Imaging V: Image Processing*. Ed. by M. H. Loew. SPIE, June 1991, pp. 1445–1445. DOI: [10.1117/12.45227](https://doi.org/10.1117/12.45227).
- [162] Z. Guo and R. W. Hall. “Parallel thinning with two-subiteration algorithms”. In: *Communications of the ACM* 32.3 (Mar. 1989), pp. 359–373. DOI: [10.1145/62065.62074](https://doi.org/10.1145/62065.62074).
- [163] T. Y. Zhang and C. Y. Suen. “A fast parallel algorithm for thinning digital patterns”. In: *Communications of the ACM* 27.3 (Mar. 1984), pp. 236–239. DOI: [10.1145/357994.358023](https://doi.org/10.1145/357994.358023).
- [164] D. F. Gonzalez Obando, J.-C. Olivo-Marin, and V. Meas-Yedid. “Morphological operations on polygons using straight skeletons for digital pathology”. In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE, Apr. 2018. DOI: [10.1109/isbi.2018.8363826](https://doi.org/10.1109/isbi.2018.8363826).
- [165] M. Held and S. Huber. “Topology-oriented incremental computation of Voronoi diagrams of circular arcs and straight-line segments”. In: *Computer-Aided Design* 41.5 (May 2009), pp. 327–338. DOI: [10.1016/j.cad.2008.08.004](https://doi.org/10.1016/j.cad.2008.08.004).
- [166] M. Held. “VRONI: An engineering approach to the reliable and efficient computation of Voronoi diagrams of points and line segments”. In: *Computational Geometry* 18.2 (Mar. 2001), pp. 95–123. DOI: [10.1016/s0925-7721\(01\)00003-7](https://doi.org/10.1016/s0925-7721(01)00003-7).
- [167] S. Nok Chiu, D. Stoyan, W. S. Kendall, and J. Mecke, eds. *Stochastic Geometry and its Applications*. Wiley, Aug. 2013. DOI: [10.1002/9781118658222](https://doi.org/10.1002/9781118658222).
- [168] D. F. González Obando, J.-C. Olivo-Marin, L. Wendling, and V. Meas-Yedid. “Vector-Based Morphological Operations on Polygons Using Straight Skeletons for Digital Pathology”. In: *Discrete Geometry for Computer Imagery*. Springer International Publishing, 2019, pp. 249–261. DOI: [10.1007/978-3-030-14085-4_20](https://doi.org/10.1007/978-3-030-14085-4_20).
- [169] D. Shaked and A. M. Bruckstein. “Pruning Medial Axes”. In: *Computer Vision and Image Understanding* 69.2 (Feb. 1998), pp. 156–169. DOI: [10.1006/cviu.1997.0598](https://doi.org/10.1006/cviu.1997.0598).
- [170] X. Bai, L. Latecki, and W.-Y. Liu. “Skeleton Pruning by Contour Partitioning with Discrete Curve Evolution”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.3 (Mar. 2007), pp. 449–462. DOI: [10.1109/tpami.2007.59](https://doi.org/10.1109/tpami.2007.59).
- [171] W. Shen, X. Bai, R. Hu, H. Wang, and L. J. Latecki. “Skeleton growing and pruning with bending potential ratio”. In: *Pattern Recognition* 44.2 (Feb. 2011), pp. 196–209. DOI: [10.1016/j.patcog.2010.08.021](https://doi.org/10.1016/j.patcog.2010.08.021).

-
- [172] A. D. Ward and G. Hamarneh. “The Groupwise Medial Axis Transform for Fuzzy Skeletonization and Pruning”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.6 (June 2010), pp. 1084–1096. DOI: [10.1109/tpami.2009.81](https://doi.org/10.1109/tpami.2009.81).
- [173] L. Serino and G. Sanniti di Baja. “A new strategy for skeleton pruning”. In: *Pattern Recognition Letters* 76 (June 2016), pp. 41–48. DOI: [10.1016/j.patrec.2015.04.003](https://doi.org/10.1016/j.patrec.2015.04.003).
- [174] H. Nasser, P. Ngo, and I. Debled-Rennesson. “Dominant point detection based on discrete curve structure and applications”. In: *Journal of Computer and System Sciences* 95 (Aug. 2018), pp. 177–192. DOI: [10.1016/j.jcss.2017.07.007](https://doi.org/10.1016/j.jcss.2017.07.007).
- [175] R. Youssef, A. Ricordeau, S. Sevestre-Ghalila, and A. Benazza-Benyahya. “Evaluation Protocol of Skeletonization Applied to Grayscale Curvilinear Structures”. In: *2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, Nov. 2015. DOI: [10.1109/dicta.2015.7371256](https://doi.org/10.1109/dicta.2015.7371256).
- [176] A. M. AL-Shatnawi, B. M. AlFawwaz, K. Omar, and A. M. Zeki. “Skeleton extraction: Comparison of five methods on the Arabic IFN/ENIT database”. In: *2014 6th International Conference on Computer Science and Information Technology (CSIT)*. IEEE, Mar. 2014. DOI: [10.1109/csit.2014.6805978](https://doi.org/10.1109/csit.2014.6805978).
- [177] R. M. Haralick. “Performance characterization in image analysis: thinning, a case in point”. In: *Pattern Recognition Letters* 13.1 (Jan. 1992), pp. 5–12. DOI: [10.1016/0167-8655\(92\)90108-c](https://doi.org/10.1016/0167-8655(92)90108-c).
- [178] B. K. Jang and R. T. Chin. “One-pass parallel thinning: analysis, properties, and quantitative evaluation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.11 (1992), pp. 1129–1140. DOI: [10.1109/34.166630](https://doi.org/10.1109/34.166630).
- [179] M. Y. Jaisimha, R. M. Haralick, and D. Dori. “A methodology for the characterization of the performance of thinning algorithms”. In: *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR 93)*. IEEE Comput. Soc. Press. DOI: [10.1109/icdar.1993.395731](https://doi.org/10.1109/icdar.1993.395731).
- [180] T. Grigorishin, G. Abdel-Hamid, and Y.-H. Yang. “Skeletonisation: An electrostatic field-based approach”. In: *Pattern Analysis and Applications* 1.3 (Sept. 1998), pp. 163–177. DOI: [10.1007/bf01259366](https://doi.org/10.1007/bf01259366).
- [181] J. Chaussard, M. Couprie, and H. Talbot. “Robust skeletonization using the discrete λ -medial axis”. In: *Pattern Recognition Letters* 32.9 (July 2011), pp. 1384–1394. DOI: [10.1016/j.patrec.2010.09.002](https://doi.org/10.1016/j.patrec.2010.09.002).

- [182] E. W. Weisstein. *Circle-circle intersection*. From *MathWorld—A Wolfram Web Resource*. 2019. URL: <http://mathworld.wolfram.com/Circle-CircleIntersection.html> (visited on 06/24/2019).
- [183] G. Rote. “Computing the minimum Hausdorff distance between two point sets on a line under translation”. In: *Information Processing Letters* 38.3 (May 1991), pp. 123–127. DOI: [10.1016/0020-0190\(91\)90233-8](https://doi.org/10.1016/0020-0190(91)90233-8).
- [184] D. Sharvit, J. Chan, H. Tek, and B. B. Kimia. “Symmetry-Based Indexing of Image Databases”. In: *Journal of Visual Communication and Image Representation* 9.4 (Dec. 1998), pp. 366–380. DOI: [10.1006/jvci.1998.0396](https://doi.org/10.1006/jvci.1998.0396).
- [185] T. Biedl, M. Held, S. Huber, D. Kaaser, and P. Palfrader. “Weighted straight skeletons in the plane”. In: *Computational Geometry* 48.2 (Feb. 2015), pp. 120–133. DOI: [10.1016/j.comgeo.2014.08.006](https://doi.org/10.1016/j.comgeo.2014.08.006).
- [186] T. Kelly. “Unwritten procedural modeling with the straight skeleton”. PhD thesis. University of Glasgow, May 2013.
- [187] T. Biedl, M. Held, S. Huber, D. Kaaser, and P. Palfrader. “A simple algorithm for computing positively weighted straight skeletons of monotone polygons”. In: *Information Processing Letters* 115.2 (Feb. 2015), pp. 243–247. DOI: [10.1016/j.ip1.2014.09.021](https://doi.org/10.1016/j.ip1.2014.09.021).
- [188] G. Eder and M. Held. “Computing positively weighted straight skeletons of simple polygons based on a bisector arrangement”. In: *Information Processing Letters* 132 (Apr. 2018), pp. 28–32. DOI: [10.1016/j.ip1.2017.12.001](https://doi.org/10.1016/j.ip1.2017.12.001).
- [189] M. Held. “Chapter 1 Introduction”. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1991, pp. 17–36. DOI: [10.1007/3-540-54103-9_12](https://doi.org/10.1007/3-540-54103-9_12).
- [190] M. Fouladivanda, K. Kazemi, and M. S. Helfroush. “Adaptive morphology active contour for image segmentation”. In: *2016 24th Iranian Conference on Electrical Engineering (ICEE)*. IEEE, May 2016. DOI: [10.1109/iraniancee.2016.7585660](https://doi.org/10.1109/iraniancee.2016.7585660).
- [191] G. Barequet and A. Vaxman. “Straight skeletons of three-dimensional polyhedra”. In: *Proceedings of the 25th annual symposium on Computational geometry - SCG 2009*. ACM Press, 2009. DOI: [10.1145/1542362.1542384](https://doi.org/10.1145/1542362.1542384).