

**Diseño y evaluación de técnicas de registro de  
imágenes hiperespectrales aplicadas al diagnóstico  
de cáncer de mama**



**Trabajo Fin de Grado en Ingeniería Electrónica  
Industrial y Automática**

**Escuela de Ingenierías Industriales y Civiles**

**Universidad de Las Palmas de Gran Canaria**

**Autor:** D. Santana Núñez, Javier

**Tutores:** Dr. Marrero Callicó, Gustavo Iván

Dr. Ortega Sarmiento, Samuel

D.<sup>a</sup> Quintana Quintana, Laura

**Fecha:** diciembre de 2022



**Design and evaluation of hyperspectral images  
registration techniques applied to breast cancer  
diagnosis**



**Industrial and Automatic Electronics Engineering  
Degree's Thesis**

**School of Industrial and Civil Engineering**

**University of Las Palmas de Gran Canaria**

**Author:** Mr. Santana Núñez, Javier

**Supervisors:** Dr. Marrero Callicó, Gustavo Iván

Dr. Ortega Sarmiento, Samuel

Mrs. Quintana Quintana, Laura

**Date:** December 2022







# Acknowledgment

---

I would like to thank in the following lines to everyone to have contributed to everyone that have directed me and assisted me to carry out this project. First, to my thesis' supervisors, Gustavo Marrero, Samuel Ortega, and Laura Quintana which have introduced me and guided me in the world of research.

Second, to my friends Adrián Pérez, Javier Hernandez, Hector Blas, and Hector Guerra, and colleagues Javier Medina, Gabriel Naya, and David Cerdán that have been advising me in various fields and have been there whenever I needed them.

Finally, to my family that have been supporting me at my weakest moments, putting me up, and making me feel lucky to have them. For all of this and much more, thanks you all.

# Abstract

---

Each year, thousands of people are affected by breast cancer. One of the most cutting-edge strategies for cancer detection is to analyze biopsy samples using Hyperspectral (HS) Imaging (HSI) techniques. HS images can highlight tumoral cells since every substance has a different spectral signature. Due to the complexity of this cell highlighting, a categorization method, such as Deep Learning (DL) Neural Networks (NNs), is required to carry out the classification. The biggest issue while employing NNs is that the algorithm needs a large dataset of categorization instances to be trained and tested.

The dataset required for these training is a bunch of HS images with cell annotations. Traditionally, a pathologist would manually annotate them on digitized RGB histological slides, but those labels must also be annotated on the HS images. Hence an image registration method to extract the labels and overlay them with the HS image must be established. The main goal of this project is to develop and test different image registration methods to find which one shows the highest performance. To evaluate the performance of these registration methods, a dataset of HS, not annotated RGB and annotated RGB images has been captured.

The first step is the study of the state of the art of every field involved to try to understand the different challenges and solutions adopted by other authors. The results of this research are the following characteristics of the registration algorithms to implement: it should search for affine transformations globally in the plane of the image, automatically, by using intrinsic information of the intrasubject multimodal images and using a linear interpolator. Other attributes that are not defined by the dataset images, and so their influence on the registration performance will be tested, are the intrinsic element to use (either intensity values or feature points), the similarity measure, and optimizer used to perform the registration. A validation of these methods and the performance of each combination of these properties have been tested by implementing them in *Matlab 2022b* and collecting metrics such as Structural Similarity Index Metric (SSIM), Mean Squared Error (MSE), and the time required to perform the registration.

The final dataset obtained consists of three groups (HS images, annotated RGB slides, and RGB slides) of 490 images from 18 different patients, 1470 images in total. The calculation of different registration algorithms for this dataset has shown a variety of results. The best performances were given by the feature-based registration methods, which use detectors with scale and rotation invariance (median of the SSIM is almost 75%) such as Speeded-Up Robust Features (SURF), KAZE or Scale Invariant Feature Transform (SIFT). In contrast, the worst performing methods were the intensity-based registration algorithms (maximum median of the SSIM is 15%). All these tests were repeated ten times to reduce the distortions introduced by the system. Overall,



our analysis of the dataset showed that using area-based registration in this context was not efficient enough, but using certain detectors such as SURF, KAZE, or SIFT provided satisfactory results.

# Resumen

---

Cada año, miles de personas se ven afectadas por el cáncer de mama. Una de las estrategias más innovadoras para la detección del cáncer es el análisis de muestras de biopsias utilizando técnicas de Imagen Hiperespectral (HSI). Las imágenes HS pueden resaltar las células tumorales debido a que cada sustancia tiene una firma espectral diferente. Por la complejidad de estos métodos, se requiere un algoritmo de categorización, como las Redes Neuronales (NN) de Aprendizaje Profundo (DL). El mayor problema al utilizar NN es que el algoritmo necesita un gran conjunto de datos de categorización para ser entrenado y testeado.

El conjunto de datos necesario para estos entrenamientos es un conjunto de imágenes HS con anotaciones a nivel celular. Tradicionalmente, un patólogo las resaltaría manualmente en imágenes RGB digitalizadas, pero esas etiquetas deben encontrarse en las imágenes HS. Por lo tanto, se ha de establecer un método de registro de imágenes para extraer las etiquetas en las imágenes RGB y superponerlas con la imagen HS. El objetivo principal de este proyecto es desarrollar y analizar diferentes métodos de registro de imágenes para encontrar cuál muestra el rendimiento más alto. Para evaluar la productividad de estos métodos de registro, se ha capturado un conjunto de datos de imágenes HS, RGB sin anotaciones y RGB anotadas.

El primer paso es el estudio del estado del arte de cada campo involucrado para tratar de comprender los diferentes desafíos y soluciones adoptadas por otros autores. El resultado de esta investigación es el siguiente conjunto de características que debe cumplir los algoritmos de registro a implementar: ha de buscar transformaciones afines globalmente en el plano de la imagen, de forma automática, utilizando información intrínseca de las imágenes multimodales del intrasujeto y utilizando un interpolador lineal. Otros atributos que no están definidos por las imágenes del conjunto de datos y que, por lo tanto, se probará su influencia en el rendimiento de registro son el elemento intrínseco usado (valores de intensidad o puntos de característica), la medida de similitud y el optimizador utilizado para realizar el registro. Se ha validado estos métodos y el rendimiento de cada combinación de estas propiedades mediante su implementación en *Matlab 2022b* y las siguientes métricas: el Índice de Similitud Estructural (SSIM), el Error Cuadrático Medio (MSE) y el tiempo requerido para realizar el registro.

El conjunto de datos final obtenido consta de tres grupos (imágenes HS, láminas RGB anotadas y láminas RGB) de 490 imágenes de 18 pacientes diferentes, un total de 1470 imágenes. El uso de diferentes algoritmos de registro para este conjunto de datos ha dado como resultado una variedad de resultados. Los mejores fueron proporcionados por los métodos de registro basados en puntos característicos, que utilizan detectores con invariabilidad de escala y rotación (la mediana del SSIM es casi del 75%) como *Speeded-Up Robust Features* (SURF), KAZE o *Scale Invariant*

*Feature Transform (SIFT)*. Por otro lado, los métodos de registro basados en intensidad fueron los que obtuvieron peores resultados (la mediana máxima del SSIM es del 15%). Todas estas pruebas se repitieron diez veces para reducir las distorsiones introducidas por el sistema. En general, nuestro análisis del conjunto de datos mostró que el uso del registro basado en área en este contexto no fue lo suficientemente eficiente, pero el uso de ciertos detectores como SURF, KAZE o SIFT tuvo resultados satisfactorios.

# Contents

---

1	Introduction .....	26
1.1	Context .....	26
1.2	Objectives .....	27
1.3	Document organization.....	27
2	State of the art .....	29
2.1	Breast cancer .....	29
2.1.1	Lump detection.....	32
2.1.2	Location and characterization .....	32
2.1.3	Final diagnosis .....	33
2.2	Pathology .....	33
2.2.1	Molecular pathology .....	34
2.2.2	Cytopathology .....	34
2.2.3	Surgical pathology .....	34
2.3	HSI – Applications .....	37
2.3.1	Imaging science - Basics .....	37
2.3.2	HSI camera configurations.....	39
2.3.3	Mathematical representation of images.....	41
2.4	DL NNs .....	42
2.4.1	Classification of ML algorithm.....	43
2.4.2	DL NNs – Basics operation, categorisation and applications .....	44
2.4.2.1	Definition and interconnectivity of neurons .....	44
2.4.2.2	Layer classification.....	44
2.4.2.2.1	FC layers.....	44
2.4.2.2.2	Convolution and deconvolution layers .....	45
2.4.2.2.3	Pooling layer.....	46
2.4.2.2.4	Recurrent layer .....	46

2.4.2.3	Activation functions .....	46
2.4.3	Types of DL NN algorithms .....	47
2.4.3.1	Artificial NN.....	47
2.4.3.2	Recurrent NN .....	48
2.4.3.3	Convolutional NN .....	48
2.5	Image registration – Methods comparative .....	49
2.5.1	Dimensionality and modality of registration.....	50
2.5.2	Interaction .....	51
2.5.3	Domain of transformation.....	51
2.5.4	Variations and nature of transformation.....	52
2.5.4.1	Rigid transformations .....	53
2.5.4.1.1	Translation.....	53
2.5.4.1.2	Rotation .....	53
2.5.4.1.3	General rigid transformation .....	53
2.5.4.1.4	Reflection .....	54
2.5.4.2	Non-rigid transformation.....	54
2.5.4.2.1	Scaling transformation.....	54
2.5.4.2.2	Similarity transformation.....	54
2.5.4.2.3	Shearing transformation .....	54
2.5.4.2.4	Affine transformation .....	55
2.5.4.2.5	Other transformations.....	55
2.5.5	Interpolation and resampling.....	56
2.5.5.1	Nearest neighbour interpolation function.....	57
2.5.5.2	Linear interpolation function.....	57
2.5.5.3	Cubic B-spline interpolation function .....	57
2.5.5.4	General cubic spline interpolation function.....	58
2.5.5.5	Two-dimensional interpolation functions.....	58
2.5.6	Nature of the registration basis .....	59
2.5.6.1	Non-image based.....	59

2.5.6.2	Extrinsic.....	59
2.5.6.3	Intrinsic.....	59
2.5.6.3.1	Point-based registration.....	60
2.5.6.3.2	Area-based registration.....	61
2.5.7	Similarity measure and optimization procedure.....	62
2.5.7.1	Similarity of features.....	62
2.5.7.2	Similarity of intensities values.....	63
2.5.7.3	Optimization procedure.....	63
2.5.8	Subject and object.....	63
2.6	Summary.....	64
3	Methodology and materials.....	65
3.1	Materials.....	65
3.1.1	Acquisition system.....	65
3.1.2	Software.....	68
3.1.3	Other materials.....	68
3.2	Methodology.....	68
3.2.1	Acquisition framework.....	69
3.2.2	Pre-processing framework.....	71
3.2.3	Registration process and evaluation.....	72
3.2.3.1	Feature Registration.....	73
3.2.3.2	Intensity registration.....	75
3.2.3.3	Validation of the registration methodology.....	77
3.2.3.4	Evaluation of the results.....	78
3.2.3.4.1	Evaluation of the feature registration.....	78
3.2.3.4.2	Evaluation of the intensity registration.....	79
3.2.3.5	Analysis of variance test of the results.....	79
3.2.3.5.1	GenerateBoxChart function.....	79
3.2.3.5.2	PlotNormDistribution function.....	79
3.2.3.5.3	GenerateAnovaPValueTable function.....	80

3.2.4	Annotation extraction and transformation.....	80
3.3	Summary.....	82
4	Results.....	83
4.1	Dataset.....	83
4.2	Method validation.....	84
4.3	Intensity registration.....	85
4.3.1	Metrics results for intensity registration.....	86
4.3.2	Statistical analysis of the intensity registration.....	88
4.4	Feature registration.....	90
4.4.1	Metric results for feature registration.....	91
4.4.2	Statistical analysis of feature registration.....	93
4.5	Analysis of the results.....	95
4.6	Synthetic annotated RGB results.....	96
4.7	Summary.....	97
5	Conclusion.....	98
5.1	Discussion.....	98
5.2	Limitations.....	99
5.3	Future lines.....	100
6	Bibliography.....	102
7	Annexes.....	111
7.1	Dataset example and registration results.....	111
7.2	Developed code.....	117
7.2.1	FeatureRegistration.m.....	117
7.2.2	IntensityRegistration.m.....	119
7.2.3	MethodValidation.m.....	121
7.2.4	Similarity_Metric_Feature_Registration.m.....	122
7.2.5	Similarity_Metric_Intensity_Registration.m.....	123
7.2.6	GenerateBoxChart.m.....	124
7.2.7	PlotNormDistribution.m.....	124

7.2.8	GenerateAnovaPValueTable.m.....	124
7.2.9	VisualizeResultsANOVA.m .....	124
7.2.10	ExtractAnnotations.m.....	125
7.2.11	SyntheticAnnotated.m.....	126



# Table of figures

---

Figure 1 Percent distribution of the 10 leading causes of death, by sex (USA, 2019) [11].	30
Figure 2 Death rate of different cancer types (b) between men and women and (a) only in women (USA,2019).	31
Figure 3 Cancer trends in death rates, 1930-2019, per 100,000, age adjusted to the 2000 USA standard population.	31
Figure 4 Anatomy of the female breast [18].	32
Figure 5 Breast cancer localization systems and their output images: ((a)[20] and (b)[21]) X-ray scan, ((c)[22] and (d)[23]) ultrasound scan, and ((e)[24] and (f)[25]) MRI scan.	33
Figure 6 Cost of each type of pathology in euros.	34
Figure 7 Histopathological workflow [31] [32] [33] [34].	35
Figure 8 Tissue dyed with (a) H&E and ((b) and (c)) various antibodies using IHC [36].	35
Figure 9 Difference between two dye processes of the same slide [36].	36
Figure 10 Ray diagram of image formation [48].	37
Figure 11 Electronic scheme of image generation [50].	38
Figure 12 Bayer Filter Array [52].	38
Figure 13 Schematic of a (a) point, (b) line, and (c) focal plane scanner camera [53].	40
Figure 14 Example of white and black image matrix representation.	41
Figure 15 Example of RGB image matrix representation.	41
Figure 16 Spectrum of single pixels.	41
Figure 17 Spectral signature of playa surface [55].	42
Figure 18 Relationship between AI, ML and DL [59].	43
Figure 19 RL basic diagram.	43
Figure 20 Single neuron of a NN system.	44
Figure 21 Backpropagation network model [67].	45
Figure 22 Example of convolution layer.	45
Figure 23 Visualization of recurrent layer [65].	46
Figure 24 Illustration of the linearity of the scalar product.	46
Figure 25 DL activation functions graphics: (a) Sigmoid, (b) Tanh, (c) Rectified Linear Unit (ReLU), (d) Leaky ReLU, and (e) non-linear cube.	47
Figure 26 CNN schematic for image recognition [77].	48
Figure 27 ReCAPTCHA example (a) for text and (b) for images [78].	49
Figure 28 Summary of the image registration methods classification	50
Figure 29 Visualization example of the registration of a point.	52

Figure 30 Visual example of each transformation: (a) translation, (b) rotation, (c) rigid, (d) point reflection, (e) line reflection, (f) scaling, (g) similarity, (h) shearing, and (i) affine [85][84].	56
Figure 31 Interpolation and resampling diagram [88].	56
Figure 32 Main interpolating functions: (a) nearest neighbour, (b) linear, (c) cubic B-spline, and ((d) and (e)) general cubic spline [88].	57
Figure 33 Interpolation of an image by three methods: (top left) the original image, (top right)nearest neighbour, (bottom left) bilinear, and (bottom right) bicubic [87].	59
Figure 34 Diagram of point-based registration [81].	60
Figure 35 Diagram of an area-based registration [98].	62
Figure 36 Microscopic HS acquisition system: (A) HS camera, (B) RGB camera, (C) XY linear stage, (D) joystick controller, and (E) lamp power supplies [102].	65
Figure 37 Olympus BX53 scheme [103].	66
Figure 38 Diagram of (a) reflected light and (b) transmitted light.	67
Figure 39 HS camera used [108].	67
Figure 40 (a) Motorized base [109] and (b) joystick used in these experiments [110].	68
Figure 41 Workflow of image acquisition	69
Figure 42 Normalized responsivity spectra of the human eye [112]	72
Figure 43 Workflow of (a) feature registration, (b) intensity registration, and (c) final image result.	73
Figure 44 Examples of the dataset captured: (a) not annotated RBGs, (b) annotated RBGs, and (c) synthetic RBGs.	84
Figure 45 Method validation results for area-based registration ((a) SSIM and (c) $MSE^{-1}$ ) and point-based registration ((b) SSIM and (d) $MSE^{-1}$ )	85
Figure 46 Results of intensity registration of the (a) left column and (b) right column of Figure 45	86
Figure 47 Box chart of (a) SSIM, (b) $MSE^{-1}$ , and (c) $Time^{-1}$ results for intensity registration.	88
Figure 48 (left) PDF analysis and (right) multiple comparison of the means of (a) SSIM, (b) $MSE^{-1}$ , and (c) $Time^{-1}$ intensity registration.	89
Figure 49 ANOVA p-value results for the (a) SSIM, (b) $MSE^{-1}$ and (c) $Time^{-1}$ intensity registration.	90
Figure 50 ((a) and (b)) SURF features detection, ((c) and (d)) pairing, and ((e) and (f)) result of the feature registration.	91
Figure 51 Box chart of (a) SSIM, (b) $MSE^{-1}$ , and (c) $Time^{-1}$ results for feature registration.	93
Figure 52 (left) PDF analysis and (right) multiple comparison of the means of (a) SSIM, (b) $MSE^{-1}$ , and (c) $Time^{-1}$ feature registration.	94

Figure 53 ANOVA p-value results for the (a) SSIM, (b) MSE <sup>-1</sup> , and (c) Time <sup>-1</sup> feature registration .....	95
Figure 54 Result of the (a) mask creation, (b) annotation extraction, and (c) synthetic annotated RGB generation.....	97
Figure 55 Methodology to obtain similar RGB annotated slices and HS cubes [33] [34] [3]. .....	100

# List of tables

---

Table 1	Properties of some descriptors and detector algorithm [89][90][91][92][93][93][94][95][96][97] .....	61
Table 2	List of each pair of optimizer/similarity metric and detector used. ....	78
Table 3	SSIM, MSE <sup>-1</sup> and Time <sup>-1</sup> results of intensity registration. ....	87
Table 4	SSIM, MSE <sup>-1</sup> and Time <sup>-1</sup> results of feature registration. ....	92
Table 5	Summary of registration performance. ....	96
Table 6	Examples of the dataset and its registration result .....	116

# Table of code fragments

---

Code Fragment 1 Help FeatureRegistration output.....	73
Code Fragment 2 Variable definition of FeatureRegistration function.....	74
Code Fragment 3 Read and modify the input parameters .....	74
Code Fragment 4 Read and transform to grey scale the images .....	74
Code Fragment 5 Detect the feature points of each image .....	75
Code Fragment 6 Extract and match the descriptors.....	75
Code Fragment 7 Perform the transformation.....	75
Code Fragment 8 Help IntensityRegistration output.....	76
Code Fragment 9 Variable definition of IntensityRegistration function.....	76
Code Fragment 10 Definition of the similarity metric and optimizer .....	76
Code Fragment 11 Perform the iterative search of the transformation .....	77
Code Fragment 12 Validation of the registration methodology .....	77
Code Fragment 13 Evaluation of the performance of the feature registration algorithm .....	78
Code Fragment 14 Evaluation of the performance of the intensity registration algorithm.....	79
Code Fragment 15 Plot box chart of each group.....	79
Code Fragment 16 Plot the PDF of a group.....	80
Code Fragment 17 Plot of the ANOVA tests.....	80
Code Fragment 18 Extraction of the annotations.....	81
Code Fragment 19 Generation of the synthetic RGB with annotations .....	82
Code Fragment 20 FeatureRegistration.m .....	119
Code Fragment 21 IntensityRegistration.m .....	121
Code Fragment 22 MethodValidation.m.....	122
Code Fragment 23 Similarity_Metric_Feature_Registration.m.....	123
Code Fragment 24 Similarity_Metric_Intensity_Registration.m.....	124
Code Fragment 25 GenerateBoxChart.m .....	124
Code Fragment 26 PlotNormDistribution.m.....	124
Code Fragment 27 GenerateAnovaPValueTable.m .....	124
Code Fragment 28 VisualizeResultsANOVA.m.....	125
Code Fragment 29 ExtractAnnotations.m.....	126
Code Fragment 30 SyntheticAnnotated.m .....	127

# Acronyms

Acronym	Meaning
A	Amps
ACS	American Cancer Society
AI	Artificial Intelligence
ANN	Artificial Neural Network
ANOVA	Analysis Of Variance
BRISK	Binary Robust Invariant Scalable Keypoints
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CCD	Charge-Coupled Devices
CDC	Centers of Disease Control and prevention
CMOS	Complementary Metal-Oxide-Semiconductor
CNN	Convolutional Neural Network
DL	Deep Learning
FAST	Features from Accelerated Segment Test
FC Layer	Fully Connected Layer
FNR	False Negative Rate
FOV	Field Of View
FPR	False Positive Rate
GBCA	Gadolinium-Based Contrast Agent
H&E	Hematoxylin and Eosin
HELICoiD	HypErspectraL Imaging Cancer Detection
HExPERIA	HypErsPEctRal Imaging for Artificial intelligence applications
HS	Hyperspectral
HSI	Hyperspectral Imaging
ID	Identification
IDE	Integrated Development Environment
IHC	Immunohistochemistry
IoT	Internet of Things
ITHaCA	<i>IdeNtificación Hiperespectral de tumores CerebrAles</i>
IUMA	<i>Instituto Universitario de Microbiología Aplicada</i>
MHSI	Medical Hyperspectral Imaging
ML	Machine Learning
MRI	Magnetic Resonance Imaging
MSER	Maximally Stable Extremal Regions
NCC	Normalized Cross-Correlation
NCHS	National Center for Health Statistics
NN	Neural Network
ORB	Oriented FAST and Rotated BRIEF
PC	Personal Computer
PDF	Probability Density Function
PLC	Programmable Logic Controller
PNG	Portable Network Graphics
PROFIBUS	Process Field Bus

ReLU	Rectified Linear Unit
RGB	Red-Green-Blue
RL	Reinforced Learning
RNN	Recurrent Neural Network
SAD	Sum of Absolute Differences
SCADA	Supervisory Control and Data Acquisition
SIFT	Scale Invariant Feature Transform
SL	Supervised Learning
SSD	Sum of Squared Differences
SSIM	Structural Similarity Index Metric
SURF	Speeded-Up Robust Features
TALENT	arTificiAl inteLLigence and high-pErformaNce systems applied to e-health and smarT farming
TFG	<i>Trabajo Fin de Grado</i>
UL	Unsupervised Learning
ULPGC	Universidad de Las Palmas de Gran Canaria
USA	United States of America
USD	United States Dollars
V	Volts
W	Wats
WSI	Whole Slide Imaging

# 1 Introduction

---

Breast cancer is a disease that disturbs the lives of nearly twenty thousands of people each year [1]. As an early detection is a crucial step to improve the survival rate, the development of new technologies and techniques of diagnosis have been increasing since the last decades of the previous century. The most modern approach is to use the results of Hyperspectral Imaging (HSI) techniques applied to biopsy samples. As every material interacts differently with light, Hyperspectral (HS) images can be used to highlight tumoral cells. This cell emphasizing is not trivial, so a classification algorithm is needed to perform the autonomous identification. The use of Deep Learning (DL) Neural Networks (NNs) is the most recent and also most standardized way to approach classification problems similar to this one. The main concern while using DP NN is the need of a large dataset of categorization examples to train and test the algorithm. For this particular experiment, the dataset needed is a collection of HS images where the cells are labelled. Classically, this annotation is done by hand by a pathologist on Red-Green-Blue (RGB) images of the stained biopsies. There is no way these same doctors perform this tagging on HS images, due to their lack of experience with HS technology. Therefore, an alternative way to extract the annotations from one image to overlay them on another one must be developed. This process is called image registration. This Bachelor Thesis (in Spanish *Trabajo Fin de Grado* (TFG)) firstly presents an overview of the state of the art of image registration methods. Next, many approaches to tackle this question are presented, developed and tested using a custom dataset. Finally, some final conclusions are offered supported by the experimental and statistical results.

## 1.1 Context

This TFG is built around the TALENT (arTificial inteLLigence and high-pErformaNce systems applied to e-health and smarT farming- PID2020-116417RB-C42) project, financed by the Spanish Ministry of Science and Innovation (I+D+i investigation National Plan) and developed by four Spanish universities: *Universidad Politécnica de Madrid*, *Universidad de Las Palmas de Gran Canaria* (ULPGC), *Universidad de Castilla La Mancha* and *Universidad de Cantabria*. Specifically, this TFG is developed within a TALENT subproject: *HypErsPEctRal Imaging for Artificial intelligence applications* (TALENT-HExPERIA). This subproject is carried out by the Division of Integrated Systems (from Spanish *División de Sistemas Integrados* (DSI)), which is a research group of the Research Institute for Applied Microelectronics (in Spanish *Instituto Universitario de Microelectrónica Aplicada* (IUMA)) [2].



TALENT general objective is to obtain, process and analyse, by using Artificial Intelligence (AI), many digital data sources to improve the early decision-making in real-time applications. At the same time, the IUMA's main goal in relation to this project is to create HS image acquisition systems to generate high-quality datasets of microscopic and macroscopic captures to train, test and validate the performance of different Machine Learning (ML) algorithms and DL NNs. Some of the fields where these systems will be implemented are brain tumour detection [3] and smart agriculture [4].

Finally, it should be noted that TALENT-HEXPERIA leans on two other past projects: HypErspectraL Imaging Cancer Detection (HELICoiD) and HS identification of brain tumour (in Spanish *Identificación Hiperespectral de tumores Cerebrales* (ITHaCA)).

## 1.2 Objectives

The main objective of this TFG is to study and implement different HS image registration techniques to generate a valuable dataset to train and test AI algorithms for the autonomous detection of tumoral cells.

The subgoals of this TFG are the following ones:

- Study the state of the art on the use of HSI in breast cancer histopathology applications.
- Study the state of the art on image registration algorithms.
- Handle the HS image capture instrumentation installed at IUMA's laboratory.
- Capture an HS dataset for further image registration.
- Develop a methodology to compare the different image registration techniques.
- Register the HS images of the database with the different algorithms previously selected.
- Evaluation of the quality of the results obtained from the different algorithms.
- Documentation of the work accomplished.

## 1.3 Document organization

The structure of this document is shown below:

**Chapter 1:** Introduction. In this chapter the context and goals of the TFG and the structure of the document are presented.

**Chapter 2:** State of the art. This part of the document tries to establish the start point of the various fields where this project is involved and to show the importance of this TFG.

**Chapter 3:** Methodology and materials. This section consists of an extensive explanation of the methodology used to achieve the objectives of the project and the materials needed to make it possible.

**Chapter 4:** Results. In this section the outcomes of the methodology described in the previous chapter is shown.

**Chapter 5:** Conclusion. This chapter tries to summarize the conclusions of the TFG, points to future lines of research, and illustrates the limitations found.

**Chapter 6:** Bibliography. Lastly, in this part of the project, the reader can find all the references used in the document.

## 2 State of the art

---

This chapter is an exposition of the state of the art of all the fields and concepts needed to develop and understand the work accomplished in this project. Firstly, it will be shown the urge to develop new and better early cancer diagnosis systems, specifically for breast cancer, to improve the death rate, saving this way thousands of lives each year.

Secondly, a brief explanation of how cancer is diagnosed by histopathology nowadays and the limitations of this methods which reveals the need of non-human dependent approaches. The solution proposed by the project TALENT is the use of HSI techniques to extract more information from the tissues examined.

Consequently, the third part of this chapter will introduce the reader to HSI and its different usages in the food industry [5] [6], agriculture [4] [7], and the medical field [8] [9]. Most of these applications lean on the use of ML algorithms or DL NNs to extract information from these HS images.

As TALENT-HExPERIA approaches the problem this way as well, the fourth piece of this chapter will be focused on describing the different types of NNs and the pros and cons of each one. Some examples of the use of each type will be shown as well.

Spotting the type of NN that has been developed for the TALENT-HExPERIA project and its characteristics will clarify the necessity of the use of registration techniques to obtain a useful dataset. So, concluding this chapter, the last section will show a classification of the different registration methods and their characteristics.

### 2.1 Breast cancer

The *Collins Dictionary of Medicine*© [10] defines tumour as “any mass of cells resulting from abnormal degree of multiplication”. A further classification of these lumps may be done according to the damage they can carry to the patient. Following this categorization, tumours may be benign or malignant. While benign tumours increase locally, are often bounded in capsules, and do not invade other tissues, malignant tumours do infiltrate locally and also into lymphatic vessels and the bloodstream, establishing secondary growths elsewhere in the body (metastases). Malignant tumours are also known as cancer.

Indicators developed by the National Vital Statistics Reports show that more than one in five United States of America (USA) citizens is dying due to cancer nowadays (Figure 1) [11]. In absolute terms, it's estimated that cancer will produce approximately 609.360 death only in the USA this year, about a death per minute [12].

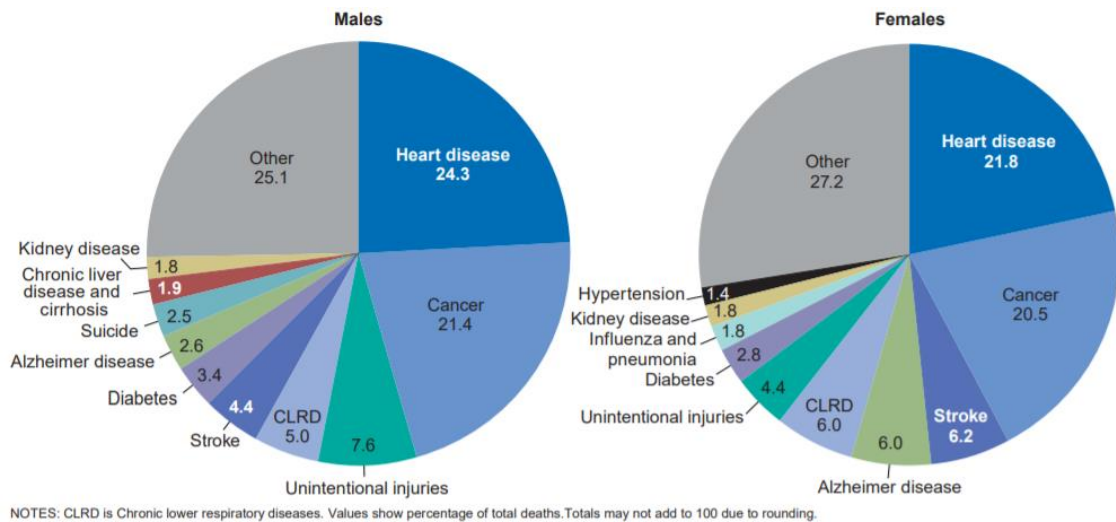


Figure 1 Percent distribution of the 10 leading causes of death, by sex (USA, 2019) [11].

In the breast cancer case, the cells affected are the mammary ones. Distinguishing the three main parts in a breast (lobules, ducts, and connective tissue), the Centers of Disease Control and prevention (CDC) identify the two more common breast cancer types as: invasive ductal carcinoma, which affects firstly to the ducts and then spreads to the closer tissues, metastasizing; and invasive lobular carcinoma, that affects firstly to the lobules and then spreads to the closer tissues, metastasizing [13].

Regardless of the popular knowledge, breast cancer affects both males and females, impacting primarily to women as they have a larger amount of these cells. The American Cancer Society (ACS) estimates that in 2022 about 530 American men will die from breast cancer [14], while 43.250 women will pass away for the same reason [15]. As the ratio between men and women's deaths is 1 to 100, and there is more information available on breast cancer in women, from now on it will be used women's breast cancer statistics, specifying in other case.

Subdividing the cancer sector from Figure 1, we can appreciate that breast cancer is the fourth most lethal type of tumour between males and females, but it is the second one for women, just after lung and bronchus cancer (Figure 2 (a)<sup>1</sup>). Despite these astonishing indicators, the lethality of cancer has been dropping down significantly since 1991, when the peak of cancer death rate was reached (Figure 3). This improvement was caused due in part because of the improving in the early-detection programs and more affordable treatments [12] [16].

<sup>1</sup> Data source: National Center for Health Statistics (NCHS), CDC, 2021 [11].

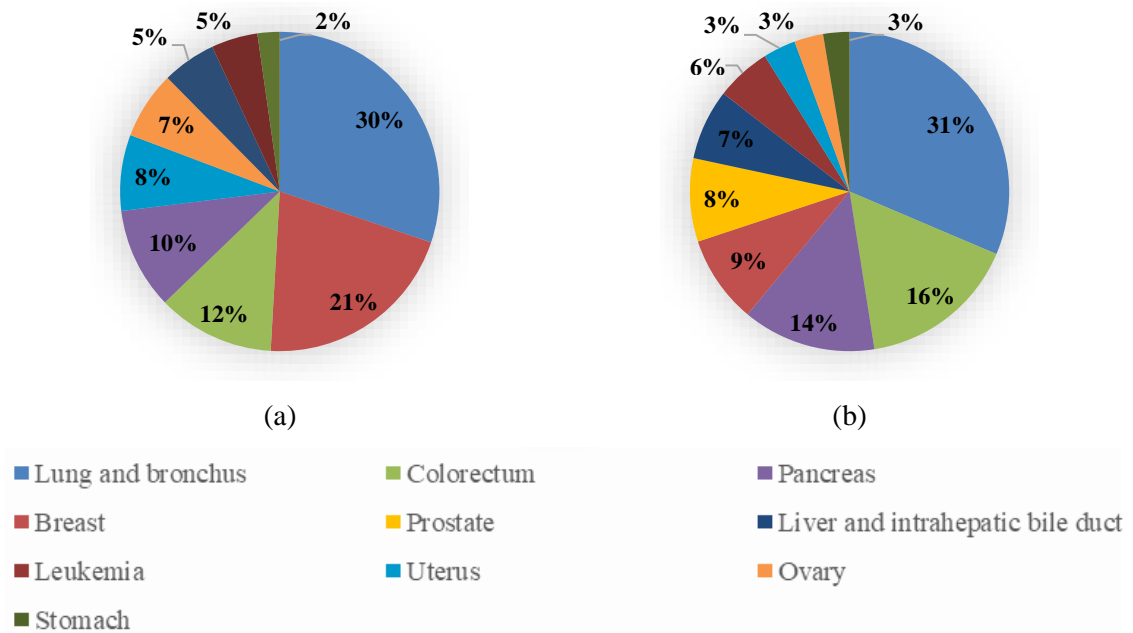


Figure 2 Death rate of different cancer types (b) between men and women and (a) only in women (USA,2019).

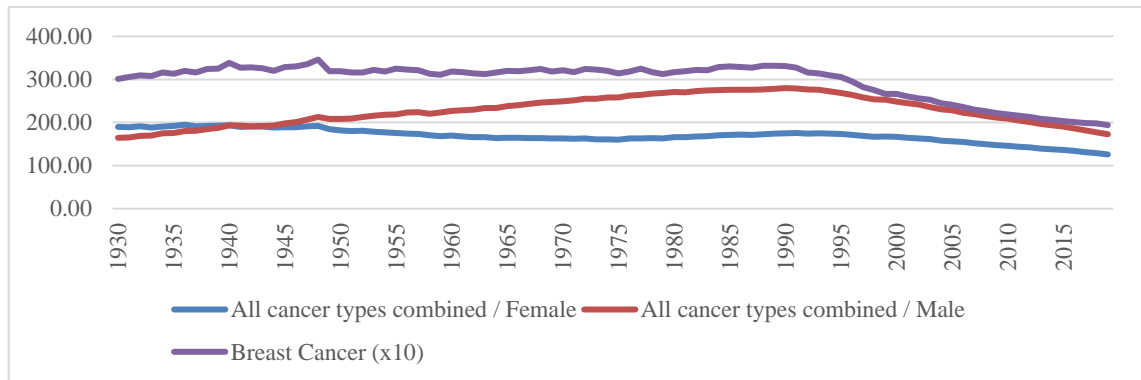


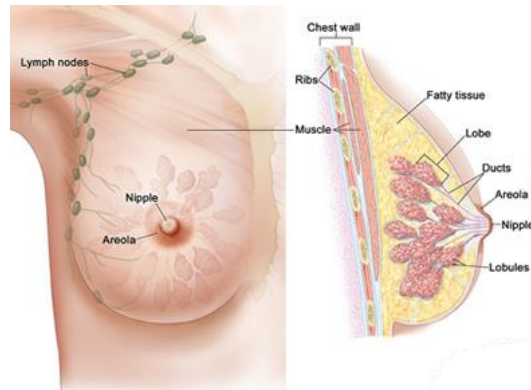
Figure 3 Cancer trends in death rates, 1930-2019, per 100,000, age adjusted to the 2000 USA standard population<sup>2</sup>.

The conventional breast cancer diagnosis nowadays has three stages in common: lump detection, where the patient or a doctor finds a suspicious protuberance; location and characterization, where a series of imaging techniques are used to confirm the bump nature; and final diagnosis, where a pathologist checks parts of the tumoral tissue to make a report [17].

<sup>2</sup> Data source: NCHS, CDC, 2021 [11]. The breast cancer data has been multiplied by a factor of 10 for better representation.

### **2.1.1 Lump detection**

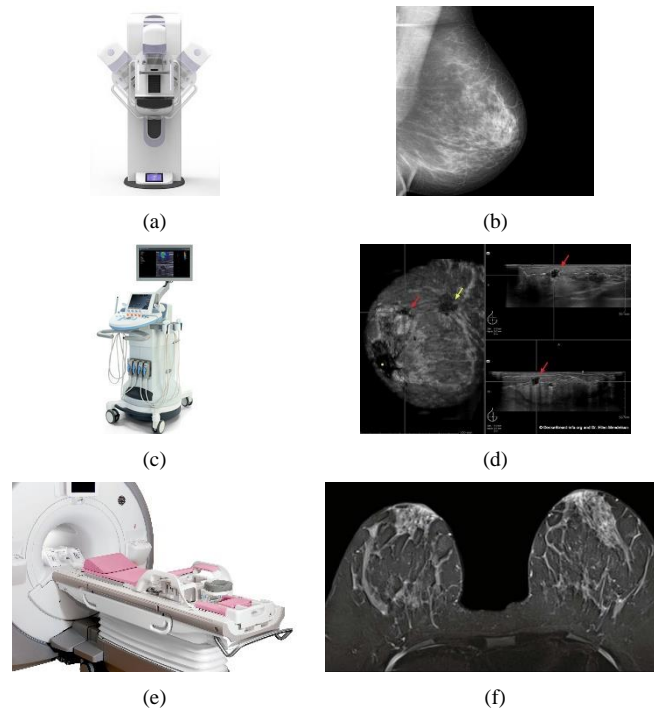
Normally, the first contact between a breast cancer patient and its tumour is a breast examination, where a doctor checks both breast and lymph nodes in the patient's armpits (Figure 4) with their hands feeling for lumps or any other abnormality in the tissue explored. When the doctor suspects there may be a breast tumour, they will ask the patient for some procedures to confirm the existence of a tumour and to locate and characterize it.



*Figure 4 Anatomy of the female breast [18].*

### **2.1.2 Location and characterization**

There are mainly three methods to obtain the location and some characteristics (size, density, composition, etc.) of a lump: mammography, ultrasound, and Magnetic Resonance Imaging (MRI) scan. A mammography is an X-ray scan of the breast that reveals the existence or non-existence of a lump or microcalcifications that can indicate the presence of cancer. This procedure is the more ordinary one due to the low cost and speed of the technique. An ultrasound scan uses sound waves to obtain images of the structures of the body. While mammography is used to know if there is any lump, ultrasound scans are used to know if the lump is solid or a fluid-filled cyst. An MRI scan uses the interaction between magnets and radio waves with Gadolinium-Based Contrast Agents (GBCAs) [19] that are injected into the patient before the procedure to produce high-quality images of the inner tissues of the breast. In Figure 5 each scanner system and their output images are illustrated.



*Figure 5 Breast cancer localization systems and their output images: ((a)[20] and (b)[21]) X-ray scan, ((c)[22] and (d)[23]) ultrasound scan, and ((e)[24] and (f)[25]) MRI scan.*

### **2.1.3 Final diagnosis**

Even with the previous methods, the final breast cancer diagnosis is accomplished by a biopsy where a lump sample is extracted with a surgical needle and an X-ray device to guide it through the breast. These samples are sent to pathological laboratories to analyse them and determine if a lump is malignant and its aggressiveness of it. In the next section of this chapter, it will be discussed how pathologists are able to determine whether a tissue has cancerous cell or not and what are the limitations of these methods.

## **2.2 Pathology**

The Department of Pathology of the McGill University defines pathology as “a branch of medical science that involves the study and diagnosis of disease through the examination of surgically removed organs, tissues (biopsy samples), bodily fluids, and in some cases the whole body (autopsy)” [26]. Consequently, a pathologist is the doctor in charge of examining the biopsy samples from tissues suspicious of having cancer cells. To know the typical pathological cancer diagnosis workflow and its issues, it is important to differentiate between molecular pathology, cytopathology, and surgical pathology.

### 2.2.1 Molecular pathology

Being the newest type of pathology, molecular pathology study and diagnoses diseases by examining molecules from organs, tissues, or fluids. This discipline, backed by recently discovered genetic knowledge, is able to detect and classify mutations in the genetic code with high sensitivity. While molecular pathology could detect the alterations in cancer cells and thus diagnosis cancer, the cost is up to eight thousand United States Dollars (USD) or seventy thousand and three hundred euros, many orders of magnitude more than other techniques [27].

### 2.2.2 Cytopathology

In cytopathology, pathologists diagnose diseases at the cellular level. The extraction of these cells is done by scraping or brushing the tissue surface, by collecting body fluids or by fine-needle aspirations [28]. Although this technique causes less harm to the patient, its cost, about five hundred USD or four hundred and fifty six euros [29], makes the following method the more usual one of the three.

### 2.2.3 Surgical pathology

Surgical pathology is the more common practice in pathology as it only requires examining the tissue sample from biopsies macroscopically with the naked eye or microscopically with a microscope, resulting in cheaper tests, between two and twenty USD or one point eighty three and eighteen point twenty seven euros [30]. When a microscope is needed, as in cancer diagnosis, it is also call histopathology.

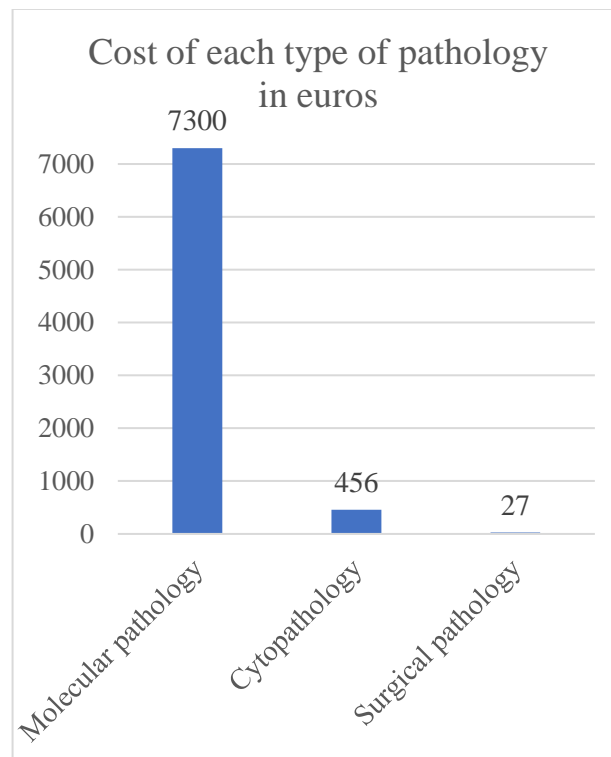


Figure 6 Cost of each type of pathology in euros.



Although the histopathologist only needs a microscope to examine the samples, the biopsy samples need to be pre-processed in order to prevent changes over time and help doctors visualize some types of cells. In Figure 7 there is illustrated a diagram summarizing the usual histopathological workflow.

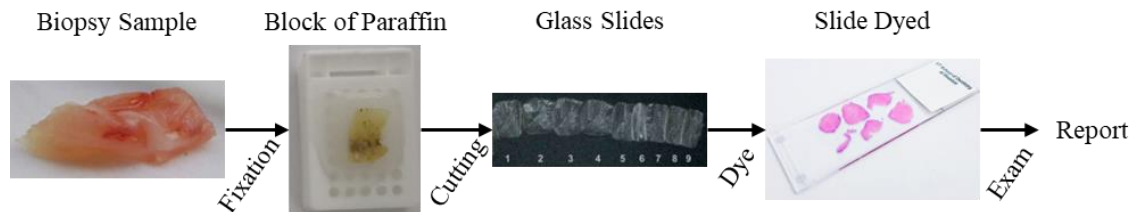


Figure 7 Histopathological workflow [31] [32] [33] [34].

As seen in Figure 7, the first step is the fixation of the biopsy sample using formalin and embedding the result in paraffin. The aim of this process is to avoid temporal changes due to bacterial metabolism or transportation damage [35].

The next phase is to cut the paraffin block in slices with a thickness of three to five micrometres ( $\mu\text{m}$ ) using a microtome. These cuts are then embedded in glass slides [36]. This process has three main objectives: to ease the manipulation of the samples, to allow to examine them under a microscope, as the slides can be easily put under it; and to reduce distortions in the view, due to the lack of curvature in the samples.

The last process before the examination is the dye of the mounted sections to make internal structures visible to histopathologists. First, the sample is made translucent using some clearing substances, such as xylene, and then it is dyed using a selected stain [37]. The usual stain used is Hematoxylin and Eosin (H&E). However, depending on which entities are needed to be highlighted, other stains can be used: antibodies (Immunohistochemistry (IHC)) [36], Van Gieson, Toluidine blue, Alcian blue, etc [37]. In Figure 8 there is an example of three different tissue samples dyed with different stains.

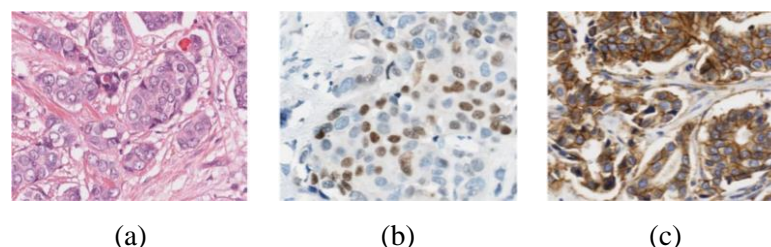
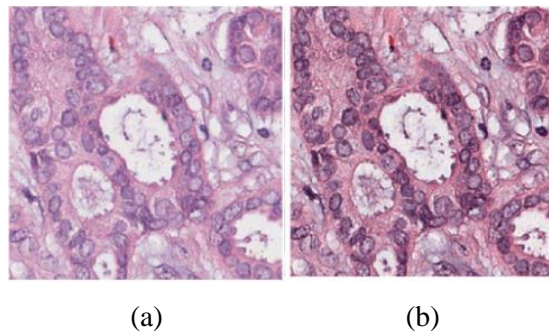


Figure 8 Tissue dyed with (a) H&E and ((b) and (c)) various antibodies using IHC [36].

Once dyed, the slide can be directly examined under the microscope (analogue pathology) or digitalized (digital pathology) using Whole Slide Imaging (WSI) scanners. These WSI scanners generate high-resolution images (half  $\mu\text{m}/\text{pixel}$  and a quarter  $\mu\text{m}/\text{pixel}$ ) with a size of a few gigabytes thanks to its compression [36]. These images can be examined later using specialized software.

When a patient's biopsy sample is examined, the histopathologist generates a report informing the types of cells and diseases found [38]. This information is used by the patient's doctor to decide what treatment the patient should follow, if there were any. The pathologist workflow previously explained has several severe problems in the two final steps. The first one is the absence of a standardized protocol in the dye of the biopsy samples [36] which can lead to extreme changes in the appearance of malicious tumour cells or diseases at the same slide. As shown in Figure 9, the cells on the right have more contrast than those on the left.



*Figure 9 Difference between two dye processes of the same slide [36].*

The second and perhaps more alarming issue with this workflow is the lack of specialization of the pathologists in some of the diseases they are diagnosing, as the diagnosis relies on their criteria exclusively. As the organization *Susan G. Komen for the Cure* published in June 2006 [39], there is not pathologist specialization for breast cancer. The American Society of Clinical Oncology and the College of American Pathologists recommend having a minimum annual caseload of between a hundred and two hundred fifty cases for a pathologist to maintain its expertise [40], which may be challenging to achieve for many small medical centres.

These problems are partially the cause of the high False Positive Rates (FPR) and False Negative Rates (FNR). A study carried out by Andrew A Renshaw and Edwin W Gould [41] found disagreements in ten percent of the cases, but these results variate between studies [42] [43].

Due to these difficulties, and many others, the *Komen* association and other authors suggest asking for a second opinion when a pathology report is received [39] [44]. However, as seen before, the unique method to stain of each histopathologist may mislead a second pathologist, and there is no evidence that the second one will be better than the first one.

It has been shown that relying exclusively on the pathologist's skills is not optimal, so many researchers have developed various tools and protocols to assist them in this process. One of them is the mentioned WSI scanners, but it is not the single improvement given by the imaging field. The introduction of HSI in the Medical field (MHSI) has opened various lines of research that prove its potential to be a helpful instrument in cancer detection [45] [3] [46]. The next section of this chapter will explain in detail the concept of HSI and its applicability in actual real-life challenges.

## 2.3 HSI – Applications

HSI is an analytical technique based on spectroscopy consisting of the assembly of hundreds of images at different wavelengths, composing this way a HS cube and a HS signature for each pixel. A previous explanation of general imaging science is needed to understand in depth what HSI is and why it is useful in cancer detection. The vastness of this discipline forces us to constrain the explanation to the vary basics, but an extensive description can be found in Cambridge's book *Introduction to Color Imaging Science* [47].

### 2.3.1 Imaging science - Basics

Imaging science is the study of the formation, manipulation, display, and evaluation of images [47]. In the subsequent paragraphs it will be enlightened how images are obtained, how they are represented in modern computers and why and how they might be useful for cancer detection.

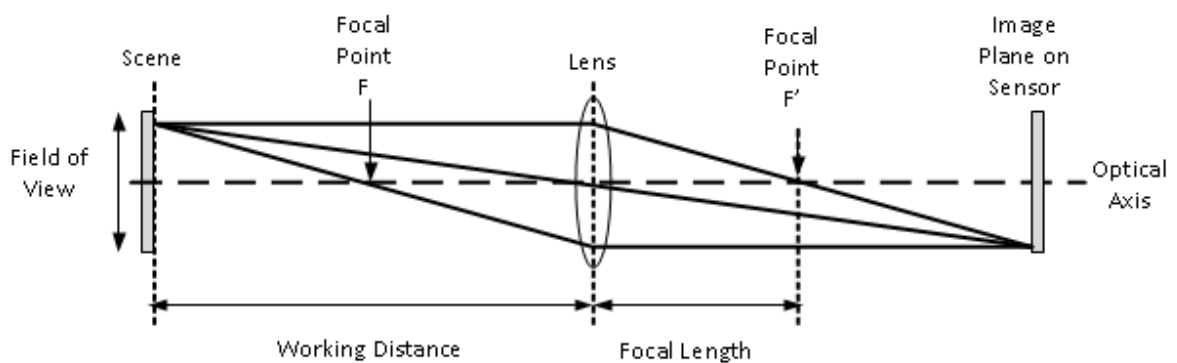


Figure 10 Ray diagram of image formation [48].

An image is a visual representation of an object formed by the collection of all focus points of light rays when it's placed in front of a mirror or a lens [49]. In imaging science, the word image designates a type of image called the real image, which is the one formed due to the convergence of the rays after meeting the lens (Figure 10); in contrast to a virtual image where those rays only appear to converge.

The formation of an image in modern cameras are made by the excitation of photo-sensible sensors. Those sensors are usually Charge-Coupled Devices (CCD) or Complementary Metal-Oxide-Semiconductor (CMOS) light sensors [50]. Regardless of the type of sensor (CCD, CMOS, electron-multiplying charge-coupled device, back-illuminated CMOS, etc [51]) the principle behind all of them is similar: when photons collide in a matrix of these sensors, a voltage difference can be measured, digitalized and reproduced (Figure 11). Each cell of the previously mentioned sensor matrix represents a pixel or point of the final image.

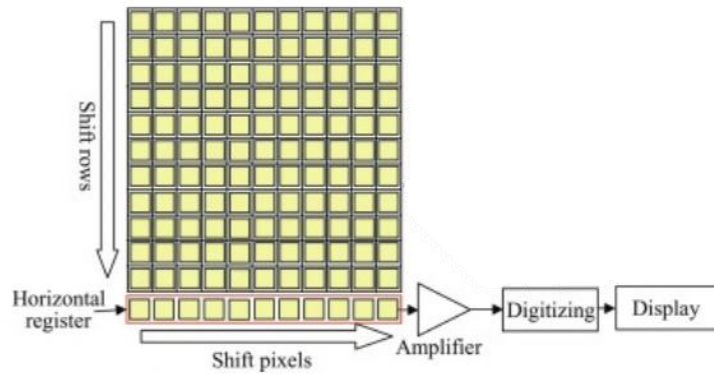


Figure 11 Electronic scheme of image generation [50].

This imaging system has the major flaw of generating black and white images as they activate with the presence of light or not, or grey-scaled images if the sensors are analogic. To obtain a coloured image, a wavelength-specific matrix sensor or a filter layer is needed. The more common approach is the second one, which consists in a film that only allows a specific wavelength to pass through it [52]. For typical coloured images, also called RGB images, a Bayer Filter Array is frequently used (Figure 12).

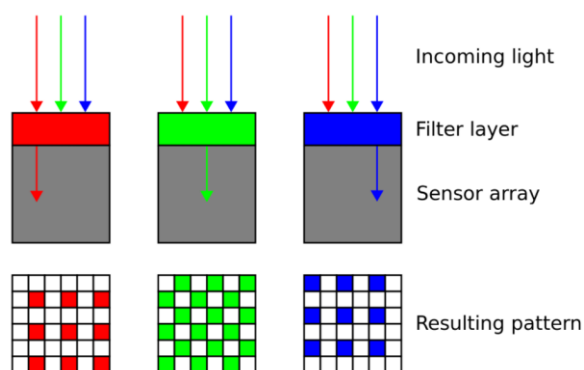


Figure 12 Bayer Filter Array [52].

In the last paragraph, three types of images were identified: black and white, grey-scale, and RGB images. While the difference between black and white and grey-scale images is the continuity of their pixels, the RGB images can be seen as three separate grey-scale images but for three distinct wavelengths. By increasing the number of unique wavelengths or bands, it is possible to obtain special types of images. If less than a hundred bands are captured, the images are called multispectral images, above that threshold they are denominated HS images [53].

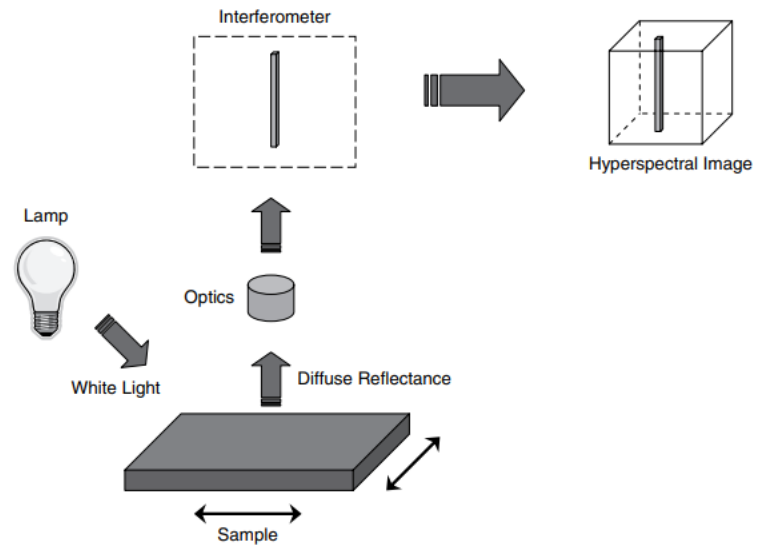
### **2.3.2 HSI camera configurations**

The Bayer Filter Array layer used for RGB images seems inefficient for HS images as it would be needed one sensor for each wavelength and pixel. For a low-definition image, which has thirty pixels per centimetre approximately [54], it would need at least three hundred sensors per centimetre. As it is not suitable, three different camera configurations have been developed to overcome this circumstance: point scan, line scan, and plane scan.

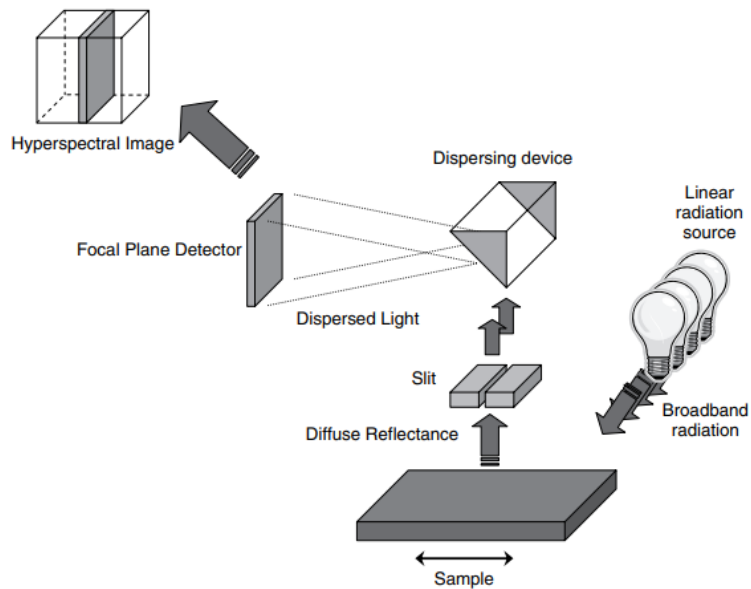
In point scanning imaging, the spectrum (collection of wavelengths) of each pixel is measured at different times. By scanning systematically in the two spatial dimensions, a complete HS image can be acquired meshing every point [53]. In Figure 13 (a), there is a schematic set-up of a point scanning camera where an interferometer is used as the spectrum sensor.

In contrast with the previous camera, in line scanning imaging the line scan or push-broom camera configuration allows to measure the spectrum of several spatial points (a line) from the sample using two-dimensional detectors. By moving the Field Of View (FOV) of the camera respect to the sample, a whole HS image can be obtained [53]. In Figure 13 (b) a schematic of the capture mechanism of a line scanning camera is illustrated.

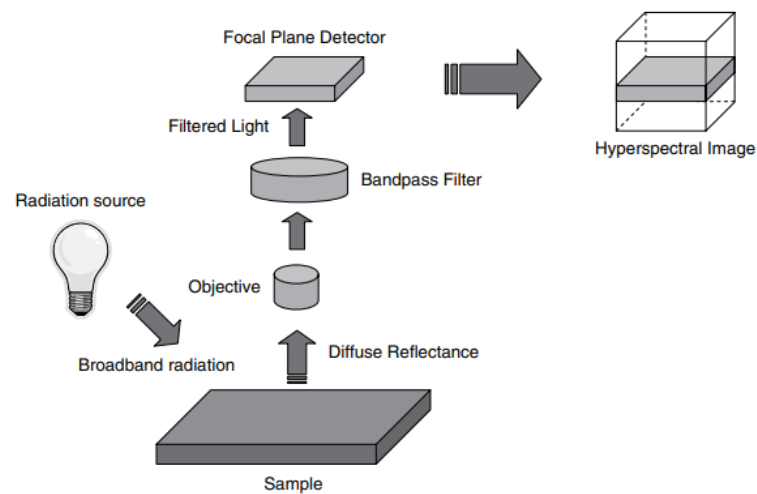
While point and line scanning cameras get all the spectrum information from certain parts of the sample in the same moment, in focal plane scanning imaging the whole sample is measured at the same time but for a specific wavelength, filtering the incoming light. Changing the central wavelength of the filter, the whole spectrum can be sampled [53]. In Figure 13 (c) the schematic of a focal plane scanner camera is exemplified.



(a)



(b)



(c)

Figure 13 Schematic of a (a) point, (b) line, and (c) focal plane scanner camera [53].

### 2.3.3 Mathematical representation of images

As seen in Figure 11 and Figure 12, each photo-sensible sensor in the sensor's matrix provides, once digitalized, a decimal luminance value. So, it seems obvious that the optimal way to store the data is in a mathematical matrix form (Figure 14).

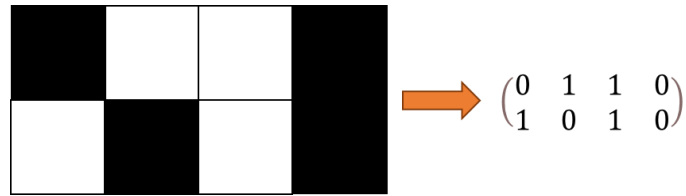


Figure 14 Example of white and black image matrix representation.

For RGB, multispectral, or HS images each wavelength needs to be represented by its own bidimensional matrix, stacking them all together in a third dimension, forming this way a cube. In Figure 15 there is an example of one of these cubes for an RGB image.

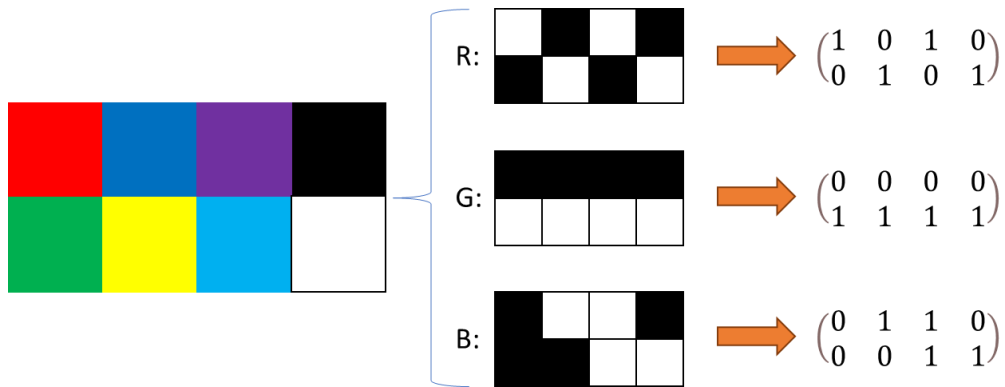


Figure 15 Example of RGB image matrix representation.

In Figure 15 the sub-images extracted from the main image represent the values of each color channel in the entire picture. However, if the information of all the colour channels is needed, they can be obtained as well (Figure 16).

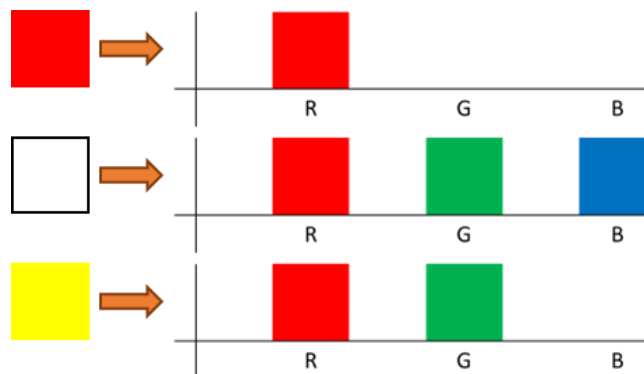


Figure 16 Spectrum of single pixels.

As seen in Figure 16, in RGB images this kind of information is discontinuous as there are only three channels. In HSI, as the number of spectral channels captured is high enough, the spectrum representation becomes continuous, and it is named spectral signature (Figure 17).

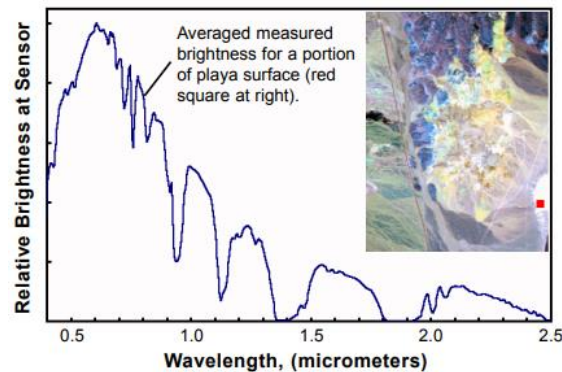


Figure 17 Spectral signature of playa surface [55].

The name spectral signature is due to the fact that every material has its unique light interaction [55]. This property of HS images has been used in multiple fields and industries as the food industry [5] [6], agriculture [4] [7], and medical field [8] [9], among others. The common point of all these HSI applications is that they use spectral signatures to classify the sample (variety and quality of rice [5], normal and cancerous cells [9], etc.). A collection of biomedical related spectral signatures can be found in spectral libraries such as *SpectraST*, *BlibSearch*, *PeptideAtlas*, etc [56].

In most HSI research papers, some sort of ML algorithms or DL NNs are used [57] [58] [3] to perform a classification as it is the most modern way to deal with this kind of problem. In the next section of this chapter, there will be an introduction to the ML field, an exposition of kinds of NNs algorithms and their advantages and disadvantages when dealing with classification problems.

## 2.4 DL NNs

Prior to showing what DL NNs are made of and how they work, first it is crucial to differentiate AI, ML and DL. While AI is the general term for the discipline of developing intelligent machines, ML refers to systems that can learn from their own experiences. If those systems need a large data set to be trained, it would be part of the DL field [59]. Consequently, it may be seen as AI involves ML, and DL is a subgroup of ML systems (Figure 18).



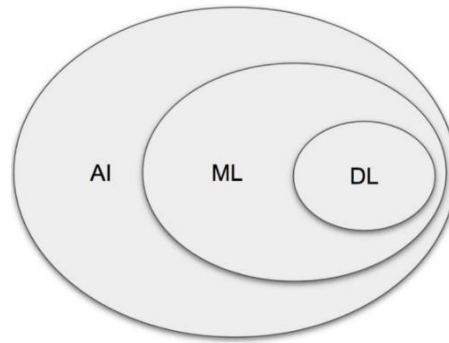


Figure 18 Relationship between AI, ML and DL [59].

#### 2.4.1 Classification of ML algorithm

Another way to classify ML algorithm is by its training method: reinforced, supervised or unsupervised. In Reinforcement Learning (RL) a reward system is used to train an algorithm to maximize a score. In the reinforced training process, there are eight main entities: environment, agent, action, state, reward, policy, value and value function [60]. In Figure 19 there is a diagram of the relationship. While conventionally this kind of algorithm is not used in classification problems [61], using a very well-designed policy can lead to a behaviour similar to a sorter [62] [63].

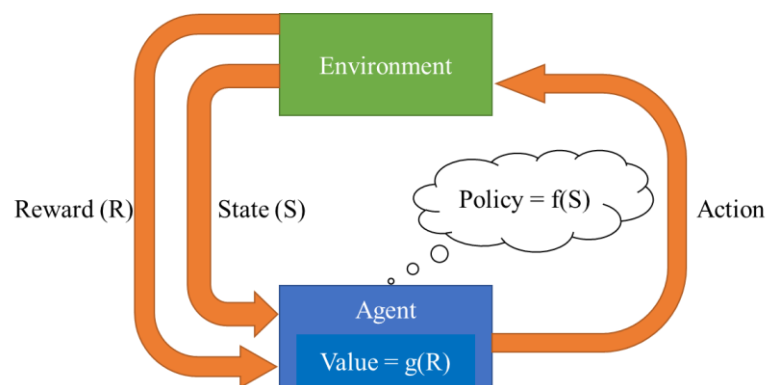


Figure 19 RL basic diagram.

While RL relies on a good environment and a fine-tuned reward definition, in Supervised Learning (SL) and Unsupervised Learning (UL) it is needed a dataset that the algorithms must learn to cluster. The main difference between SL and UL is that a supervised trained algorithm needs a labelled dataset, so the classification is constrained to the tags given; in contrast with a non-supervised trained algorithm whose training data is unlabelled, forcing it to find common, and perhaps hidden, data patterns [64].

Depending on the type of training used in DL, the algorithms can be classified as supervised DL, or unsupervised DL [65]. The following section will explain the basic functioning of NN algorithms, their classification and their uses.

### 2.4.2 DL NNs – Basics operation, categorisation and applications

The NN algorithms are systems that use human-like neurons organized in layers to extract the inner relationships of a dataset [66]. Four basic components constitute a NN: neurons, weights, activation functions, and layers [67].

#### 2.4.2.1 Definition and interconnectivity of neurons

A neuron simply executes the scalar product of an input row vector and a column vector called weight vector (equation (1)). Usually, one of the inputs is a bias to modulate the behaviour of the neuron [67]. Visually, it is represented in Figure 20.

$$Output = X \otimes W \quad \forall X \in M_{1 \times n}(\mathbb{Q}) \wedge \forall W \in M_{n \times 1}(\mathbb{Q}) \quad (1)$$

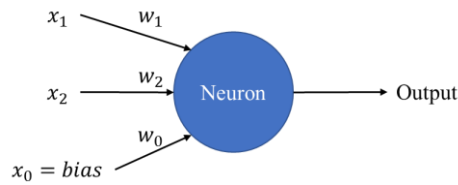


Figure 20 Single neuron of a NN system.

The weights and the bias values are learnable parameters in a NN that are initialized randomly, and, thanks to the supervised training, they are tuned iteratively to obtain the expected output. While biases represent the disparity among actual and expected outputs, weights represent how much effect has an input on the output [68]. These neurons are usually interconnected by a structure called backpropagation network (Figure 21) where a collection of neurons or layer is influenced by and/or influences another layer or the external world [69].

#### 2.4.2.2 Layer classification

Classifying the layers by their interconnection, the five more common layers are: Fully Connected layer (FC layer), convolution layer, deconvolution layer, pooling layer, and recurrent layer [70] [71].

##### 2.4.2.2.1 FC layers

FC layers connect every neuron of one layer to every one of another layer. While they may be computationally expensive, they are usually used in NN algorithms to classify image data [70]. The layers shown in Figure 21 are examples of FC layers.

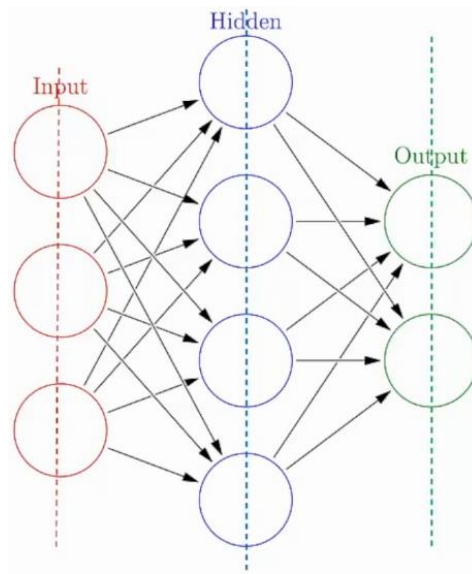


Figure 21 Backpropagation network model [67].

2.4.2.2.2 Convolution and deconvolution layers

Convolution layers perform a convolutional operation between their inputs and a filter, called kernel, which acts as the weights. The dimension of the kernel is usually smaller than the number of neurons in the previous layer or external inputs, so it must be shifted among the possible inputs. As a convolution operation expresses the overlap of two functions [72], this type of layer is used to extract features from images in imaging processing [73]. In Figure 22 an example of a convolution layer is illustrated. A deconvolutional layer makes the inverse above explained process.

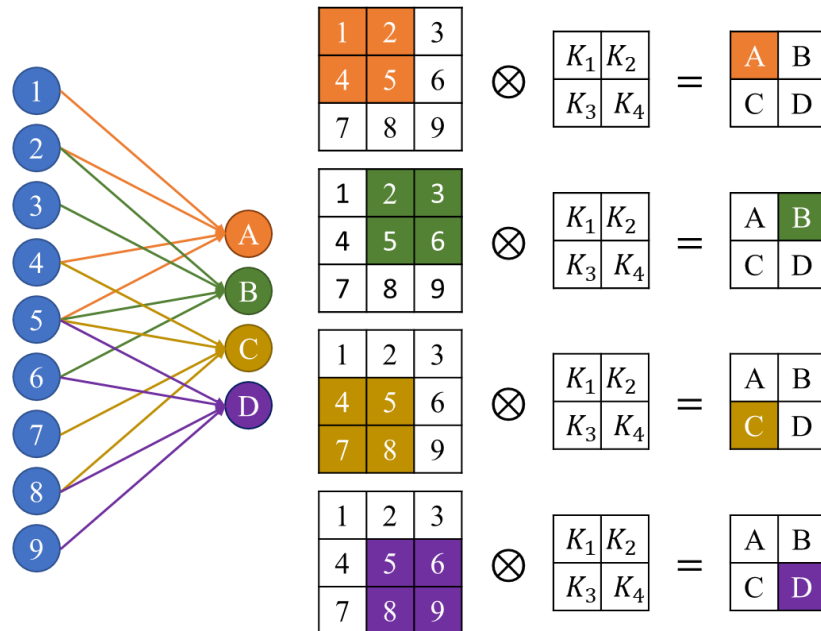


Figure 22 Example of convolution layer.

2.4.2.2.3 Pooling layer

To generalize the features extracted by a convolutional layer, a pooling layer is usually applied to the convolutional output [71]. A pooling layer simply filters or operates a pooling operation, such as max pooling or average pooling, to its inputs in order to reduce the dimensions of the data [74].

2.4.2.2.4 Recurrent layer

Finally, the last more common type of layer is the recurrent one. This kind of layer uses its previous outputs as an input for the next iteration. This capability provides de NN a sort of memory of previous states (Figure 23) [70].

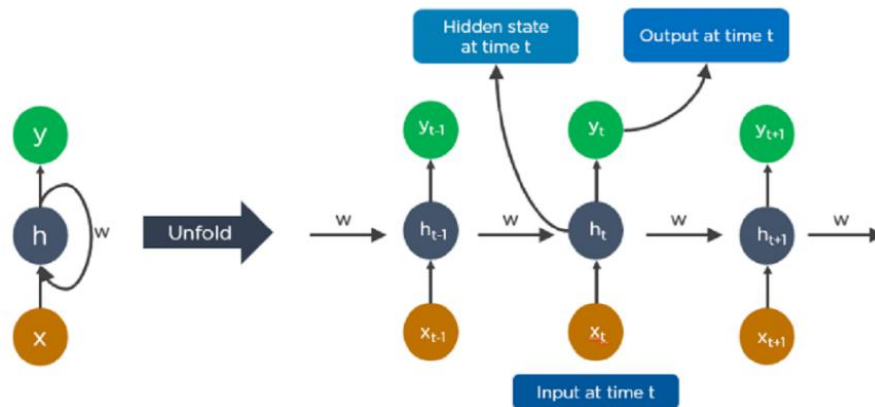


Figure 23 Visualization of recurrent layer [65].

2.4.2.3 Activation functions

The linearity results of the use of equation (1) exclusively perform poorly for complex problems (Figure 24). To improve it, a mathematical non-linear function called activation function (Figure 25) is applied to each output to break the linearity and to mathematical non-linear function. There are some desired features that useful activation functions may have: zero-centred, computationally inexpensive to calculate, and differentiable [75].

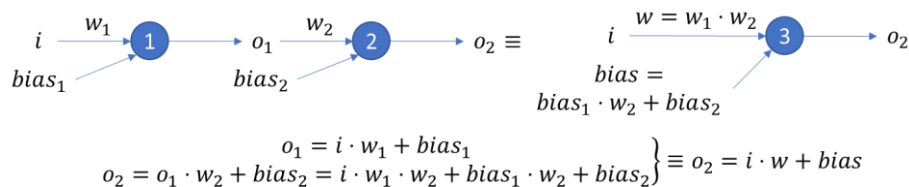


Figure 24 Illustration of the linearity of the scalar product.

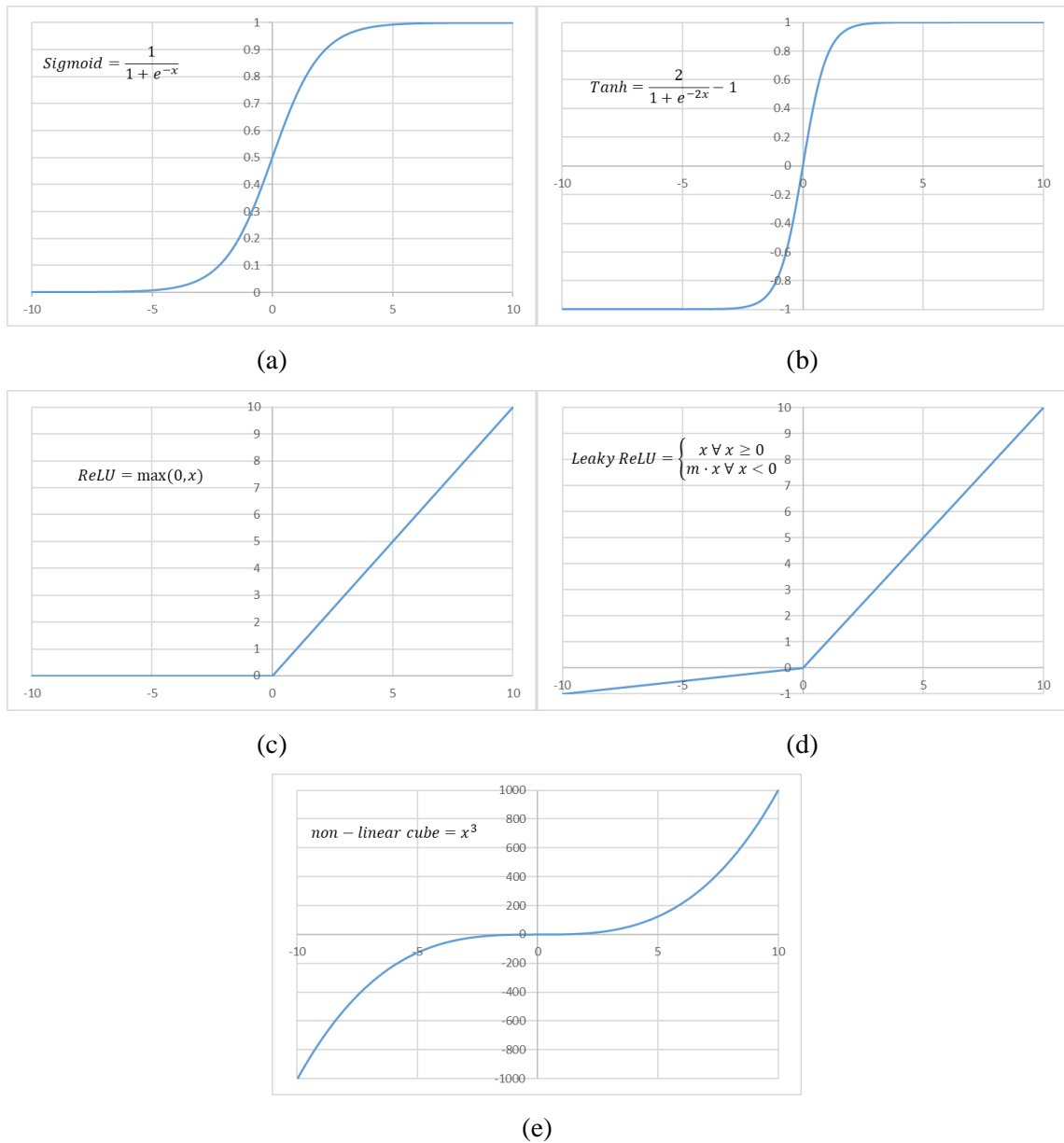


Figure 25 DL activation functions graphics: (a) Sigmoid, (b) Tanh, (c) Rectified Linear Unit (ReLU), (d) Leaky ReLU, and (e) non-linear cube.

### 2.4.3 Types of DL NN algorithms

Once the functioning of DL NN algorithms is known, a classification of the types of NN can be done depending on the types of layers used and their unique characteristics.

#### 2.4.3.1 Artificial NN

Also known as feed-forward NN, Artificial NN (ANN) algorithms are one of the simplest versions of DL NN algorithms as they only process their inputs in the forward direction and they do not even need to have hidden layers [76].

### 2.4.3.2 Recurrent NN

Recurrent NN (RNN) algorithms use recurrent layers to perform processes that need long short-term memory. This type of DL NN is usually used for text-to-speech conversions as previous information of a sentence is needed in order to achieve good intonation. The main issue is the difficulty of the training process [76].

### 2.4.3.3 Convolutional NN

The main characteristic of Convolutional NN (CNN) algorithms is the use of at least one convolution layer. Commonly CNN algorithms are made of modules in series which are built putting together a convolution layer, with a ReLU as activation function, and a pooling layer [65]. These modules are helpful in extracting features of images, that is why this type of NN is used in image recognition problems [76]. Once these modules are used, the feature maps extracted (output of the modules) may be introduced in an FC layer to perform the final classification [65].

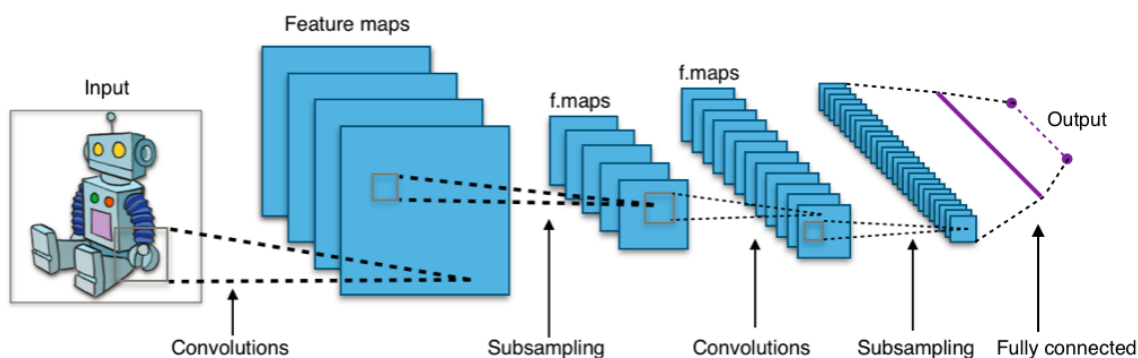


Figure 26 CNN schematic for image recognition [77].

The main concern while making CNN algorithms is the need for a large labelled training dataset, as CNNs are supervised DL algorithms [76]. Alphabet's Google developed a Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) named ReCAPTCHA, where the user must transcript two distorted words, one for control and the other one for test, to prove the user is not a robot (Figure 27) [78] . This way, Google has been able to obtain a labelled dataset of written words and the human ReCAPTCHA answers, used as labels, to train an IA which is used to transcript books stored in Google Books and to provide a transcription service to others, as The New York Times [79].

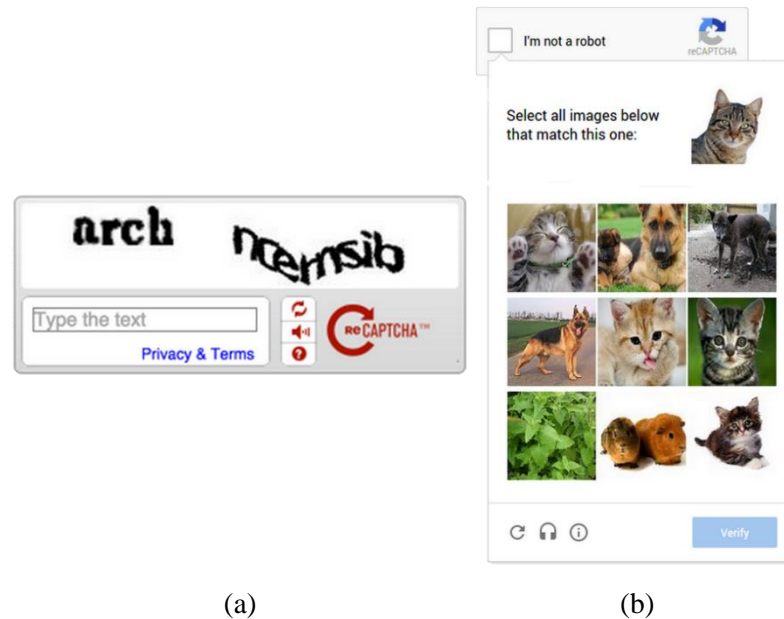


Figure 27 ReCAPTCHA example (a) for text and (b) for images [78].

In subsequent ReCAPTCHA versions, Google introduced image labelling as CAPTCHA where the user must select every picture similar to another or that represents a concept, such as a cat or a pedestrian crossing. There is speculation that these test results have also been used to train algorithms, but the company is not clear about it [80].

At this point, it seems obvious that a reasonable tumour detection approach is the training of CNNs with a dataset of HS images of cancerous and non-tumoral tissue properly labelled. However, as shown in Section 2.2, those labels are in the RGB WSI images, so a way to extract and display them on the HS images must be developed. As the images are not captured by the same sensor and there is no known coordinate system relation between them, this process is not trivial and a transformation to superimpose each pair of images must be calculated. The process of finding that transformation is called image registration. The last section of this chapter shows a general explanation and a comparison of various methods.

## 2.5 Image registration – Methods comparative

Image registration, also known as fusion, matching or warping [81], is defined as the determination of a geometrical transformation that allows to overlay two images of the same object [82]. The vast image registration methods can be classified in many different ways, but a usual approach is the one proposed by J. B. Antoine Maintz and Max A. Viergever in their “*A survey of medical image registration*” paper [83]. In the following section, the Maintz and Viergever classification of medical image registration methods, slightly modified to introduce complementary general concepts by other authors, is shown (Figure 28).

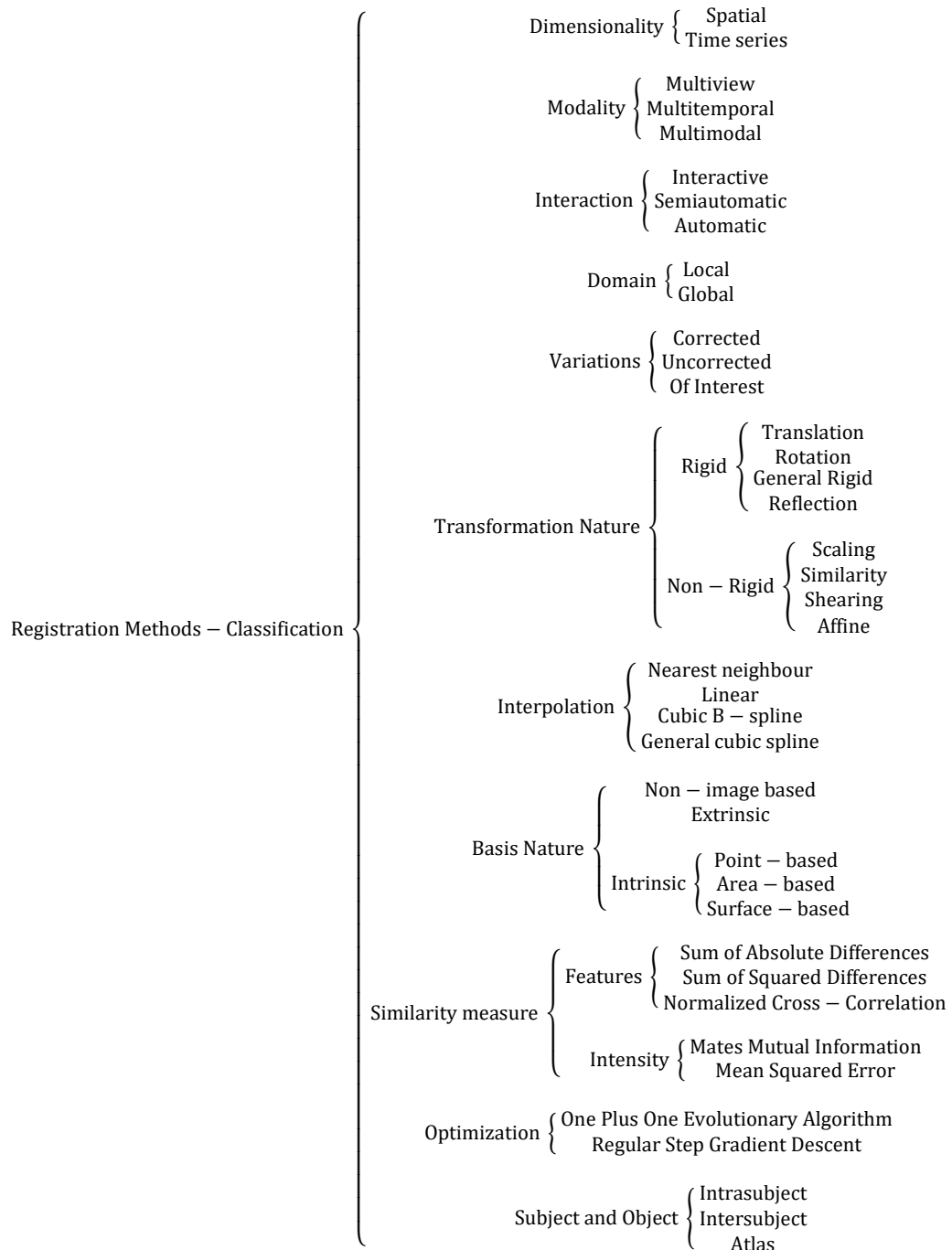


Figure 28 Summary of the image registration methods classification.

### 2.5.1 Dimensionality and modality of registration

The dimensionality of a registration method refers to the number and type of dimensions involved in the registration process [83]:

- **Spatial registration methods:** these methods use only the spatial dimensions. It can be subcategorized into: 2D-2D, 2D-3D and 3D-3D.
- **Time series registration methods:** these methods use the values of the time dimension and of the spatial dimensions to perform the registration. While uncommon, this kind of registration is helpful for monitoring bone growth or lump evolution [83].



On the other hand, modality refers to the relationship between the image acquisition system of each view [83][84]:

- **Multiview or monomodal registration:** the images are acquired by the same sensors but in a different viewpoint or environmental conditions.
- **Multitemporal registration:** the images are acquired by the same sensors in similar conditions but at different moments in time.
- **Multimodal registration:** the images are acquired by diverse sensors. This kind of registration is frequently used to fuse the information obtained by both systems.

### **2.5.2 Interaction**

Interaction refers to the roll that the user plays in the registration. Attending to this characteristic, the registration can be [83]:

- **Interactive:** the user performs the transformation itself, supported by a visual tool for the registration process or a mathematical representation of the images.
- **Semiautomatic:** the user may perform a pre-processing or rectification [82] of the images to reduce the differences between them. For example, if one view is rotated, approximately, ninety degrees; the user may perform that rotation a priori to reduce the computation cost of the registration process. Another action the user may execute is a qualitative evaluation of the results to accept or reject them.
- **Automatic:** the user only provides the images and their metadata to the algorithm, which autonomously performs the image registration. While the image registration should ideally be done automatically, proper rectification may reduce the computational cost, the time per registration and improve the robustness [83].

### **2.5.3 Domain of transformation**

The domain of the transformation performed by the registration method refers to whether all the points of the view, pixels or voxels, are altered equally or not. Therefore, the transformation could be [83]:

- **Local:** the transformation is applied to a subset of points in the view, or different subsets require different transformations. This type of transformation is rarely used due to its possible violation of image continuity.
- **Global:** the transformation is applied to every point of the image.

### 2.5.4 Variations and nature of transformation

Transformation is defined as the mapping of the location of points in a view into another [85]. These transformations are used to reverse the variations source of misregistration or distortions between the images, but some of them should not be corrected as they are useful changes, e.g., the growth of a lump or colour shift of a field. So, prior to showing the nature of the possible transformations, it is necessary to comprehend the types of variations [85]:

- **Corrected distortions:** variations that are easy to model mathematically. These are the variations that must be deleted by the registration methods.
- **Uncorrected distortions:** variations that are difficult to model mathematically. These distortions are usually scene dependent variations such as lighting or shadows.
- **Distortions of interest:** these variations should not be corrected by the registration method used as they represent useful changes in the views.

Mathematically, a registration is defined as the process of finding a transformation  $\mathcal{T}$  such as:

$$u' = \mathcal{T}(u) : u' - v = \min_{\mathcal{T} \in \mathbb{T}} (\mathcal{T}(u) - v) \quad (2)$$

Where  $v$  is a point of the fixed image ( $V$ ) used as a template,  $u$  is the corresponding point of the moving image ( $U$ ) to be registered,  $\mathcal{T}$  is the transformation applied to  $u$  to land on  $v$ ,  $\mathbb{T}$  is the set of all possible transformations  $\mathcal{T}$  and  $u' - v$  is the registration error (Figure 29).

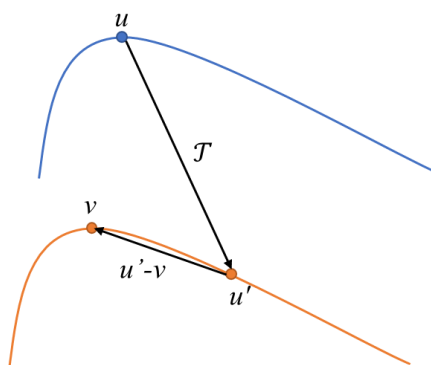


Figure 29 Visualization example of the registration of a point.

Knowing which variations are presented in the images to register, the transformation to apply can be selected. There are primarily two categories: the rigid transformations and the non-rigid transformations

The next sections are a brief mathematical description of the types of transformations in bi-dimensional images. A three-dimensional explanation can be seen in the eighth chapter of “*Handbook of Medical Imaging Volume 2. Medical Image Processing and Analysis*” by Milan Sonka and J. Michael Fitzpatrick [82].

### 2.5.4.1 Rigid transformations

Rigid transformations are geometrical transformations that preserve the distances between points in the image. There are four types of rigid transformations: translation, rotation, general rigid transformation, and reflection.

#### 2.5.4.1.1 Translation

To move an image to another location (Figure 30 (a)), it is necessary to add a motion vector to each point:

$$u' = u + t = \begin{pmatrix} u_x \\ u_y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (3)$$

#### 2.5.4.1.2 Rotation

To rotate an image  $\theta$  radians (Figure 30 (b)), it is needed to multiply each point by a rotation matrix  $R$ :

$$u' = R u = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} \quad (4)$$

It is worth mentioning that the rotational matrix  $R$  has two main properties:

- $R$  is orthogonal, meaning that the matrix product between its transpose, or vice versa, is equal to the identity matrix.
- The determinant of  $R$  is equal to plus or minus one, depending on if the rotation is proper or improper<sup>3</sup>.

$$R^t R = R R^t = I \quad (5)$$

$$\det(R) = \pm 1 \quad (6)$$

#### 2.5.4.1.3 General rigid transformation

A general rigid transformation, sometimes simply called rigid transformation (Figure 30 (c)), is a combination of a translation and a rotation:

$$u' = R u + t = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (7)$$

To simplify the notation, equation (7) can be represented by means of homogeneous coordinates using the augmented matrices of  $u'$ :

$$u' = \begin{pmatrix} u'_x \\ u'_y \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & t_x \\ \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_x \\ u_y \\ 1 \end{pmatrix} \quad (8)$$

---

<sup>3</sup> An improper rotation is defined as the product of a proper rotation and a mirror reflection through a plane normal to the axis of rotation [117].

#### 2.5.4.1.4 Reflection

An image may be reflected over a line that passes through the origin of the coordinate system (Figure 30 (d)), characterized by its slope  $\varphi$ , or over a point  $M$  (Figure 30 (e)):

$$u' = 2M - u = \begin{pmatrix} 2m_x \\ 2m_y \end{pmatrix} - \begin{pmatrix} u_x \\ u_y \end{pmatrix} \quad (9)$$

$$u' = F u = \begin{pmatrix} \cos 2\varphi & \sin 2\varphi \\ \sin 2\varphi & -\cos 2\varphi \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} \quad (10)$$

#### 2.5.4.2 Non-rigid transformation

The non-rigid transformations do not preserve the distances between points. While the first four transformations still preserve the parallelism of straight lines, the more general ones do not.

##### 2.5.4.2.1 Scaling transformation

An image may be scaled (Figure 30 (f)) simply by multiplying a diagonal matrix by each point:

$$u' = S u = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} \quad (11)$$

If  $s_x$  and  $s_y$  are equal, the scaling will be equal in both axes, as shown in Figure 30 (f). On the other hand, if they are not, there will be a change in the aspect ratio of the image. While this transformation does preserve the parallelism of straight lines, the angles between lines are modified.

##### 2.5.4.2.2 Similarity transformation

A similarity transformation (Figure 30 (g)) is a combination of a rigid transformation, a reflection<sup>4</sup>, and a scaling transformation [86]:

$$Sm = \begin{pmatrix} Sm_{11} & Sm_{12} \\ Sm_{21} & Sm_{22} \end{pmatrix} = S F R = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} \cos 2\varphi & \sin 2\varphi \\ \sin 2\varphi & -\cos 2\varphi \end{pmatrix} \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad (12)$$

$$u' = Sm u + t = \begin{pmatrix} Sm_{11} & Sm_{12} \\ Sm_{21} & Sm_{22} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \equiv \begin{pmatrix} u'_x \\ u'_y \\ 1 \end{pmatrix} = \begin{pmatrix} Sm_{11} & Sm_{12} & t_x \\ Sm_{21} & Sm_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_x \\ u_y \\ 1 \end{pmatrix} \quad (13)$$

##### 2.5.4.2.3 Shearing transformation

A shearing transformation (Figure 30 (h)), as the aspect ratio change, allows breaking the consistency of angles of non-parallel straight lines is performed by multiplying a shear matrix to each point:

$$u' = Sh u = \begin{pmatrix} 1 & sh_x \\ sh_y & 1 \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} \quad (14)$$

<sup>4</sup> It is worth noticing that a point reflection is analogous to a translation where  $t$  is equal to  $2M$  and the point to reflect is scaled by  $-1$ . So, for now on, the term reflection is used to name a line reflection.

2.5.4.2.4 Affine transformation

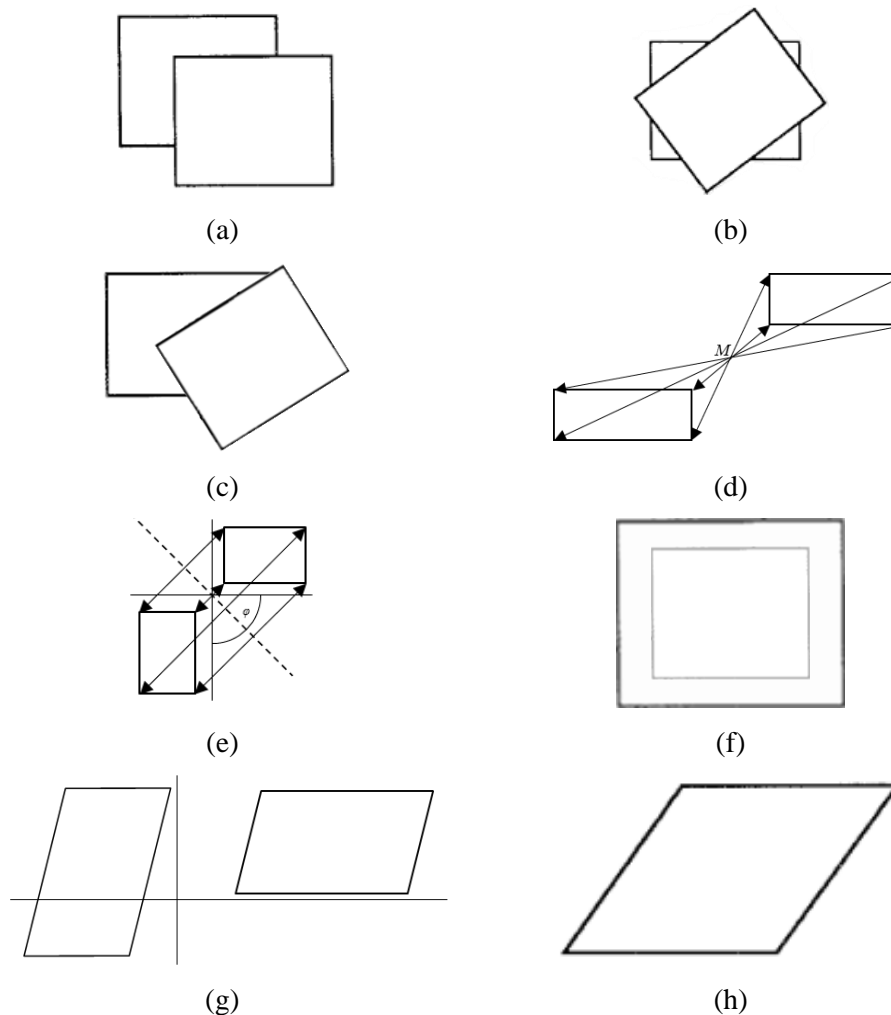
An affine transformation (Figure 30 (i)) is the more general transformation that maintains the parallelism of straight lines. It can be conceptualized as a combination of all the transformations described previously:

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = Sh Sm = \begin{pmatrix} 1 & sh_x \\ sh_y & 1 \end{pmatrix} \begin{pmatrix} Sm_{11} & Sm_{12} \\ Sm_{21} & Sm_{22} \end{pmatrix} \quad (15)$$

$$u' = A u + t = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \equiv \begin{pmatrix} u'_x \\ u'_y \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_x \\ u_y \\ 1 \end{pmatrix} \quad (16)$$

2.5.4.2.5 Other transformations

There are other transformations that performs projections to planes, changes of perspective or curve transformations in the images However, as the images to be registered in the context of this TFG, HS and RGB images of biopsies slides, do not present any of these distortions, they are not going to be discussed. An illustration of this transformations can be seen at the eighth chapter of “*Handbook of Medical Imaging Volume 2. Medical Image Processing and Analysis*” by Milan Sonka and J. Michael Fitzpatrick [82].



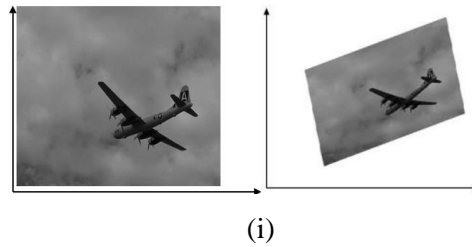


Figure 30 Visual example of each transformation: (a) translation, (b) rotation, (c) rigid, (d) point reflection, (e) line reflection, (f) scaling, (g) similarity, (h) shearing, and (i) affine [85][84].

### 2.5.5 Interpolation and resampling

The transformations previously shown can be realized in a forward or backward manner. The forward method applies the mapping function of every image point to register  $U$ . This approach can generate holes or overlaps of points in the registered image  $U'$ . The backward method on the other hand, applies the inverse of the mapping function to every point of the  $U'$  grid [87].

The main limitation of the backward technique is that when it is applied to a point  $u'$ , the corresponding  $u$  point is no longer part of the original  $U$  grid [81]. Thus, it is necessary to perform an interpolation and a resampling, to transform the discrete image  $U$  into a continuous image, by convoluting the signal with an interpolating function, and converting it back to a discrete image  $V$ , using the new coordinate system [88]. In practice, the interpolation and resampling are often combined to perform the interpolation exclusively at the points to be resampled by centring the interpolating function on each point (Figure 31).

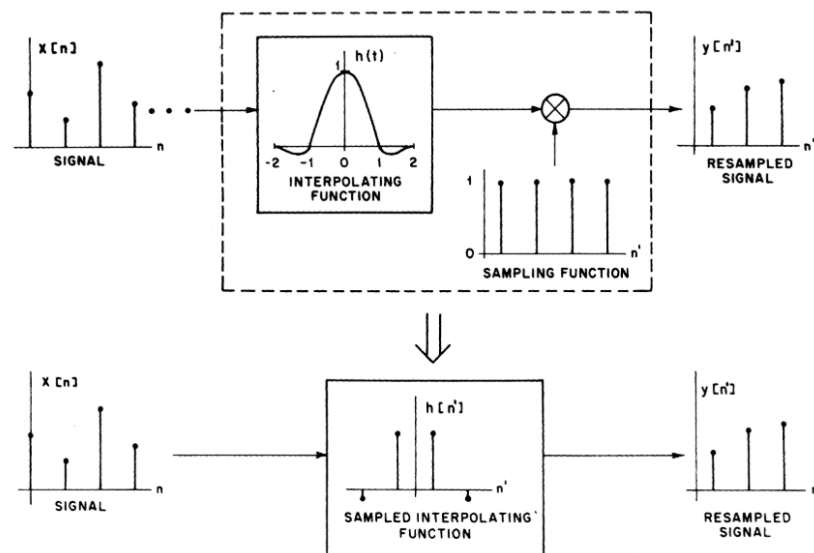


Figure 31 Interpolation and resampling diagram [88].

The interpolating functions are vertically symmetrical, and they are characterized by the piecewise-defined functions that compose them. The following sections will briefly introduce the four main interpolating functions: nearest neighbour, linear, cubic B-spline and general cubic spline [88]. A deeper description of all of them and a comparison between them can be found in J. Anthony Parker, Robert V. Kenyon and Donald E. Troxel “*Comparison of Interpolating Methods for Image Resampling*” paper [88].

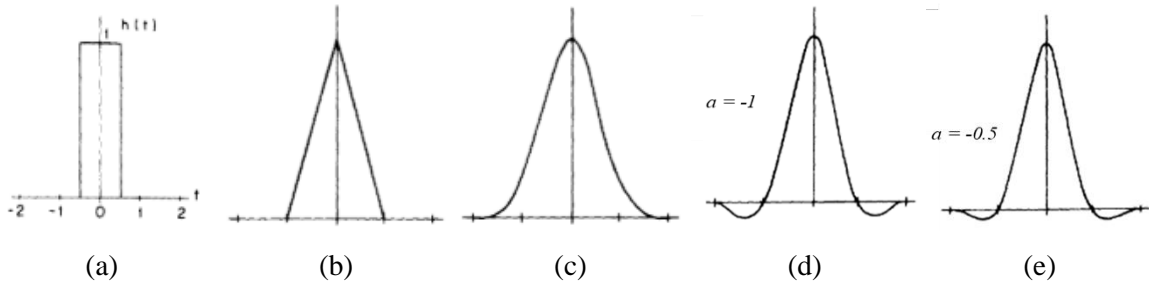


Figure 32 Main interpolating functions: (a) nearest neighbour, (b) linear, (c) cubic B-spline, and ((d) and (e)) general cubic spline [88].

### 2.5.5.1 Nearest neighbour interpolation function

The nearest neighbour algorithm is a rectangle function of unity high and a pixel of length (Figure 32 (a)). As every point is one pixel away from each other, this function replicates the nearest point to the new resample one.

$$f(x) = 1 \quad \forall x \in [-0.5, 0.5] \quad (17)$$

If the resampling frequency is higher than the original one, i.e., the interpolation is done to augment the resolution, the nearest neighbour function will fail as it will simply replicate the pixels. Although, if the frequencies are equal but the points are shifted, this method will perform a perfect linear shift.

### 2.5.5.2 Linear interpolation function

The linear interpolation algorithm is a triangular function of unity high and two pixels of base length (Figure 32 (b)). The use of more than one point of the original image allows to smooth the results and the use of a triangular shape allows the weight or importance of the pixels to be attenuated as they spread from the resample point.

$$f(x) = \begin{cases} -x + 1 & \forall x \in [0, 1] \\ x + 1 & \forall x \in [-1, 0] \end{cases} \quad (18)$$

### 2.5.5.3 Cubic B-spline interpolation function

The cubic B-spline algorithm uses four polynomials of degree three, one per pixel considered in the interpolation process (Figure 32 (c)). As the function is symmetric, defining the positive intervals will be enough to describe the entire spectra, simply replacing  $x$  by  $-x$ .

$$f(x) = \begin{cases} x^3 \frac{1}{2} - x^2 + \frac{4}{6} & \forall x \in [0, 1] \\ x^3 \frac{-1}{6} + x^2 - 2x + \frac{8}{6} & \forall x \in [1, 2] \end{cases} \quad (19)$$

This interpolation function is positive in the whole interval, so it smooths more than is necessary below the cut-off frequency and the gain on the interpolation process is unity.

#### 2.5.5.4 General cubic spline interpolation function

The previous interpolation algorithm is a particular case of a more general function where the coefficients of the independent parameters are not predefined. Every cubic spline function is defined for four pixels, which each one has its own polynomial function, and is symmetrical.

$$f(x) = \begin{cases} a_{30}x^3 + a_{20}x^2 + a_{10}x + a_{00} & \forall x \in [0, 1] \\ a_{31}x^3 + a_{21}x^2 + a_{11}x + a_{01} & \forall x \in [1, 2] \end{cases} \quad (20)$$

While every coefficient may have any value, the function should be differentiable to be useful to interpolate, i.e., it should have some constraints [88]:

$$\begin{aligned} f(0) = 1; f(1) = 0; f(2) = 0; f(1^-) = f(1^+); \\ f'(0) = 0; f'(2) = 0; f'(1^-) = f'(1^+) \end{aligned} \quad (21)$$

Applying these seven constraints to equation (20), which has eight coefficients, produces a high-resolution cubic spline function, which only has one variable and has a unity gain:

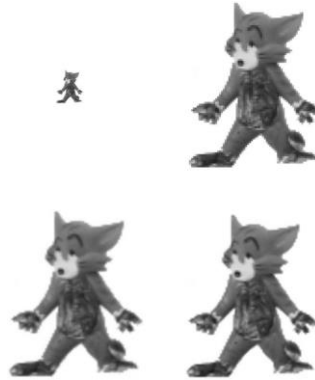
$$f(x) = \begin{cases} (a + 2)x^3 - (a + 3)x^2 + 1 & \forall x \in [0, 1] \\ ax^3 - 5ax^2 + 8ax - 4a & \forall x \in [1, 2] \end{cases} \quad (22)$$

Plotting the function for some values of  $a$  shows that when it is below zero, the interval [0,1] is positive and the [1,2] is negative (Figure 32 (d) and (e)), providing this way a better performance at high frequencies.

#### 2.5.5.5 Two-dimensional interpolation functions

All the above shown algorithms are applied to one-dimensional signals. The general approach to interpolate two-dimensional signals, as images, is to define an interpolation function as the product of two one-dimension functions [88]. In Figure 33, an image interpolation by three different methods is shown.





*Figure 33 Interpolation of an image by three methods: (top left) the original image, (top right) nearest neighbour, (bottom left) bilinear, and (bottom right) bicubic [87].*

### **2.5.6 Nature of the registration basis**

The nature of the registration basis can be defined as the feature space used to obtain the final result. Following J. B. Antoine Maintz and Max A. Viergever [83], there are mainly three types of features to be searched: non-image based, extrinsic and intrinsic.

#### **2.5.6.1 Non-image based**

A non-image-based registration technique relies on the sensor's metadata, such as time stamps or calibration of the different coordinate systems. These methods are rarely used as they need that the images are taken in the same physical location and that the object stays motionless between both acquisitions as prerequisites [83].

#### **2.5.6.2 Extrinsic**

The extrinsic registration methods use well-visible and accurately detectable artificial artefacts in the image FOV. Thanks to these objects, the registration process is comparatively easy, fast, and automatable. The main disadvantage of these techniques is anticipation required, since it is needed to know which entity is useful or if any is needed at all [83].

#### **2.5.6.3 Intrinsic**

The intrinsic registration processes rely exclusively on the image pixels values or a subset of them. There are mainly three types of intrinsic registration methods: point-based, area-based and surface-based<sup>5</sup> [82].

---

<sup>5</sup> As surface-based registration methods are used for three-dimensional images, they will not be discussed in the present document, but it is explained in detail at the eighth chapter of "*Handbook of Medical Imaging Volume 2. Medical Image Processing and Analysis*" by Milan Sonka and J. Michael Fitzpatrick [82].

### 2.5.6.3.1 Point-based registration

Point-based or feature-based registration techniques use pairs of correspondent points in each image to obtain the necessary transformations to perform the registration. Those points may be called key points [89], primitive points [82], fiducial points [82], or feature points [87].

The pair of key points may be directly specified by the user, but it can be automatically performed by feature detectors, that search for fiducial points on each image, and a descriptor algorithm, to represent the characteristics of every point detected [89]. The use of a detector and a descriptor algorithm is called segmentation (Figure 34) [81].

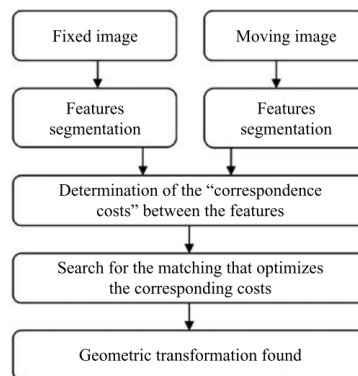


Figure 34 Diagram of point-based registration [81].

The feature points detected, and their descriptors should be invariant to the distortions presented between the images. Some of these algorithms are Harris Corner Detector, Features from Accelerated Segment Test (FAST), Binary Robust Invariant Scalable Keypoints (BRISK), Oriented FAST and Rotated BRIEF (ORB), KAZE, Scale Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Maximally Stable Extremal Regions (MSER) and Minimum Eigenvalue. Table 1 shows whether a detector has an associated descriptor and indicates if they are not altered by distortions produced by rotations, scale, and illumination changes.

	Detect	Feature	Descript	Scale invariant		Rotation invariant		Illumination invariant	
				Detect	Descript	Detect	Descript	Detect	Descript
				Harris	✓	Corner	✗	✗	—
FAST	✓	Corner	✗	✗	—	✗	—	✓	—
BRISK	✓	Corner	✓	✓	✓	✗	✓	✓	✓
ORB	✓	Corner	✓	✗	✗	✓	✓	✓	✓
KAZE	✓	Corner	✓	✓	✓	✓	✓	✓	✓
SIFT	✓	Corner	✓	✓	✓	✓	✓	✓	✓
SURF	✓	Blob	✓	✓	✓	✓	✓	✓	✓
MSER	✓	Blob	✗	✓	—	✓	—	✗	—
MinEigen	✓	Corner	✗	✗	—	✓	—	✓	—

\*Detect: Detector; Descript: Descriptor

Table 1 Properties of some descriptors and detector algorithm [89][90][91][92][93][93][94][95][96][97].

### 2.5.6.3.2 Area-based registration

Area-based or intensity-based methods use the value of each pixel or a subset of them to perform the registration. While this technique has interesting characteristics such as the lack of user interaction and, as a consequence, the simplicity of its automatization [82]. It is heavily environment dependent, as it relies exclusively on the light intensity capture.

An intensity-based registration is usually an iterative process where on each step a transformation is performed, and a similarity measure is computed. This procedure keeps optimizing until the similarity measure is minimized, if it evaluates differences, or maximized, if it rates the equivalence of the images (Figure 35).

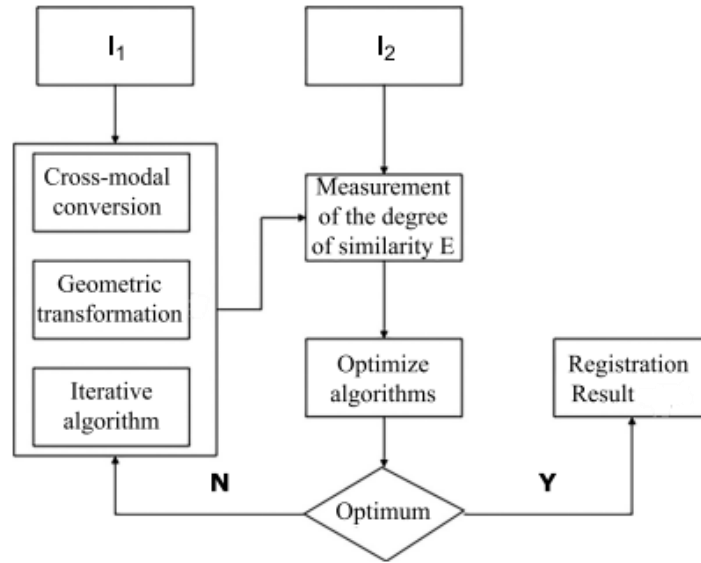


Figure 35 Diagram of an area-based registration [98].

### 2.5.7 Similarity measure and optimization procedure

A similarity measure is an instrument used to obtain a quantitative value of the disparity or parity between two images. The optimization is the process of modifying the registration transformation in order to minimize/maximize the similarity measure [85].

As seen in Section 2.5.6, if the nature of the registration is the coordinate system of the sensors (non-imaged based) or is based on artificial artefacts (extrinsic), there is no need to use any similarity measure since the transformation needed is known or easily computed. In contrast, feature-based or intensity-based algorithms need it to match the key points or to compare the overlap of the images.

#### 2.5.7.1 Similarity of features

For point-based registration, the target of the similarity function is to determine the “distance” between pairs of descriptors (mathematically represented as  $f_1$  and  $f_2$  vectors): if the distance of a descriptor of an image to one of other is less than a threshold, they are matched [99]. This matching process may be unique, i.e., a key point is exclusively matched to another or not. There are three basic possible distance functions:

- Sum of Absolute Differences (SAD):

$$SAD(f_1, f_2) = \sum |f_{1_i} - f_{2_i}| \quad (23)$$

- Sum of Squared Differences (SSD):

$$SSD(f_1, f_2) = \sum (f_{1_i} - f_{2_i})^2 \quad (24)$$

- Normalized Cross-Correlation (NCC):

$$NCC(f_1, f_2) = \frac{\sum f_{1i} \cdot f_{2i}}{\sqrt{\sum f_{1i}^2 \cdot \sum f_{2i}^2}} \quad (25)$$

#### 2.5.7.2 Similarity of intensities values

For intensity-based registration, the similarity function is performed on the whole image or on a piece of it. There are two main similarity functions for intensity values that uses the intensity values  $x$  and  $y$  of the images  $X$  and  $Y$ , respectively; or its individual ( $p_X(x)$  or  $p_Y(y)$ ) or joint ( $p_{X,Y}(x,y)$ ) probability, defined as the frequency of a pixel value or pair of pixel values divided by the sum of the total number of pixels in each the image:

- Mates Mutual Information (MMI):

$$MI(X, Y) = \sum_{x \in X} \sum_{y \in Y} p_{X,Y}(x, y) \log \frac{p_{X,Y}(x, y)}{p_X(x)p_Y(y)} \quad (26)$$

- Mean Squared Error (MSE):

$$MSE(X, Y) = \frac{1}{N} \sum_{x \in X, y \in Y} (x - y)^2 \quad (27)$$

#### 2.5.7.3 Optimization procedure

While the optimization procedure for feature-based registration is embedded in the matching process, as a match is obtained when the distance is minimized; for intensity-based registration there are two optimizers commonly used:

- **One Plus One Evolutionary Algorithm** (1+1 EA): this procedure consists of randomly selecting a transformation as a parent and evaluating the similarity metric to close transformations, selecting those with higher similarity score as new parents and iterating [100].
- **Regular Step Gradient Descent** (RSGD): this method uses the gradient of the similarity measure to find a minimum/maximum point, i.e., the optimal transformation [101].

### 2.5.8 Subject and object

Subject and object are terms related exclusively to medical image registration. Subject refers to the origin of the images, while object describes the body part represented by the images. Although the list of possible registration objects is vast, there are only three possible subjects [82]:

- **Intrasubject**: both images to be registered come from the same patient.
- **Intersubject**: the images come from two different patients.
- **Atlas**: one image comes from a patient, and the other is a combination of images from a database of the same object blended together.

## 2.6 Summary

This second chapter has guided the reader through the state of the art of every one of the fields this TFG is involved in, showing the necessity of fast and robust early breast cancer detection programs and techniques, the issues the pathologists and patients confront while making a biopsy diagnosis, how the use of HSI procedures may alleviate those complications, aided by CNN to perform classification of the images. While this approach seems to be promising, it brings its own difficulties as the generation of a dataset of labelled HS images. As the solution proposed in this TFG is the use of image registration techniques to transform the annotations from the reference system (digitized histological slides) to the HS images, an extensive classification and description of these methods, attending to their characteristics, has been shown.

## 3 Methodology and materials

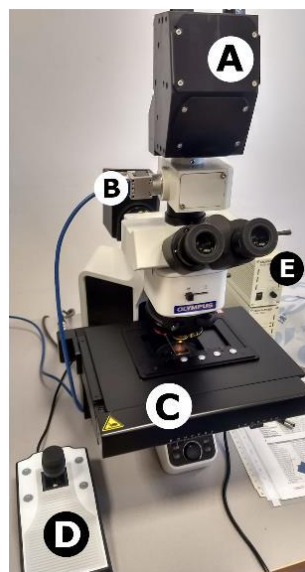
---

This third chapter describes the systems and materials needed to obtain the images to perform the registration and the methodology followed to carry it out. Firstly, an overview of the optical-mechanical subsystems, HS capture system and RGB camera is presented. Next, it is displayed the software necessary to take the images to be registered and the Integrated Development Environment (IDE) used to elaborate the registration algorithms. Then, the rest of the IUMA's laboratory materials used are presented, as they are crucial in the development of this project. Once the necessary materials have been explained, the methodology followed to obtain the HS and RGB images and the one used to perform and analyse the registration are explained.

### 3.1 Materials

#### 3.1.1 Acquisition system

The HS acquisition system can be subdivided into two subsystems: the optical part and the mechanical one. The acquisition system basis is a bright field microscope with two main modifications: the addition of an RGB and an HS cameras (optical subsystem) and a motorized base controlled by a joystick (mechanical subsystem) (Figure 36). The microscope is an *Olympus BX53* [102]. It is worth pointing to four of its components as they have been modified and/or crucial in the capture of the HS images: the revolving nosepiece, the halogen lamp housing, the light path selector and the slide holder (Figure 37 7, 1, 4 and 5 respectively).



*Figure 36 Microscopic HS acquisition system: (A) HS camera, (B) RGB camera, (C) XY linear stage, (D) joystick controller, and (E) lamp power supplies [102].*

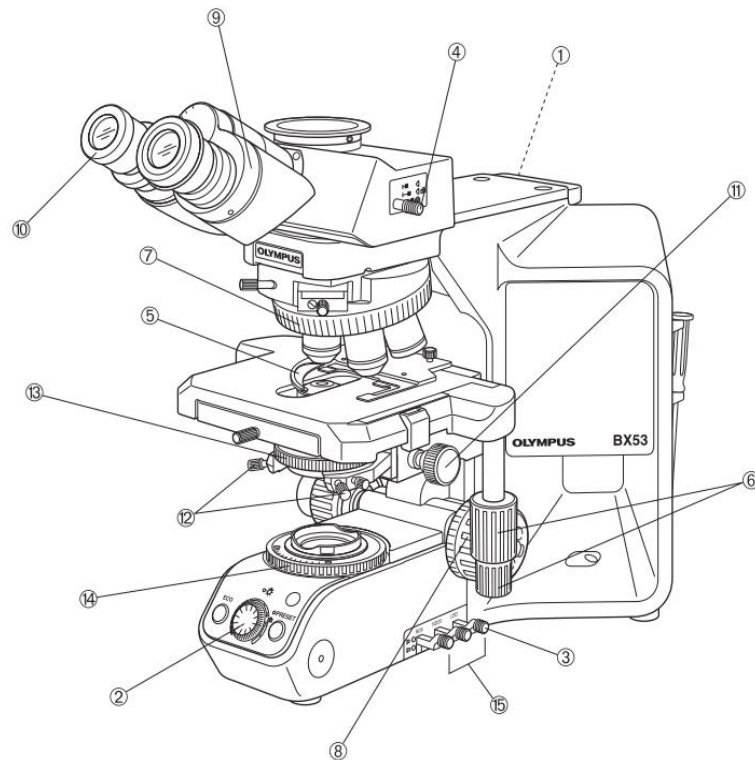
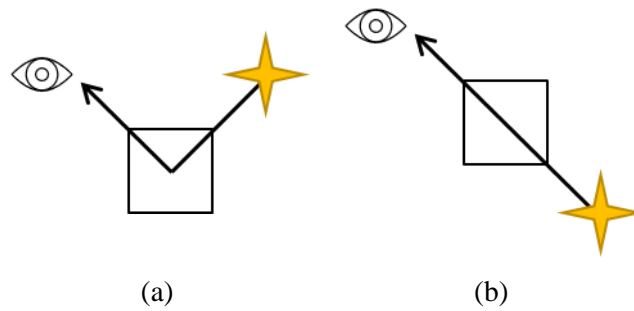


Figure 37 Olympus BX53 scheme [103].

The revolving nosepiece has four different magnification lenses: 5 $\times$ , 10 $\times$ , 20 $\times$  and 50 $\times$ . The magnification and FOV used to capture the HS and the RGB images should be similar to reduce the search space in the registration process. Two pictures of the same object with different zoom factors are equivalent to a scale distortion. A constriction of the search space will improve the computational cost and the registration results.

The halogen lamp used in this kind of microscope is a *Philips 7724* [104], which must be driven by a power supply of twelve Volts (V) and a hundred Wats (W). The power supply selected is the *Olympus TH4-200* [105], whose output is 0.9 V to 12.6 V of direct current (DC) and 8.4 amps (A), i.e., a maximum of a 105.84 W. The microscope supports the use of two lamps, one to capture the reflected light from opaque samples, shining from above (Figure 38 (a)); and another to visualize the transmitted light from translucent samples, shining from below (Figure 38 (b)).





*Figure 38 Diagram of (a) reflected light and (b) transmitted light<sup>6</sup>.*

The light reflected or transmitted from the sample may be directly seen by the microscope eyepieces (Figure 37 10) or to the push-broom HS camera Hyperspec® VNIR A-Series from HeadWall Photonics [106]. This camera is made up by an imaging spectrometer coupled to the Adimec-1000m, a CCD sensor [107]. The RGB camera will only be used to see which FOV the microscope lenses are capturing.



*Figure 39 HS camera used [108].*

As described in Section 2.3.2, to obtain a HS cube from a push-broom camera, the sample must be moved. To achieve this motion, the slide holder has been replaced with a mechanical subsystem structured by a motorized base, *Scanning Stage SCAN 130 × 85 for Upright Microscopes*, and a joystick controller, *3-Axis Joystick*, which allows to move in the three space dimensions at different speeds.

---

<sup>6</sup> The square represents the glass slide, the star symbolizes the source of light and the eye is the camera sensor.

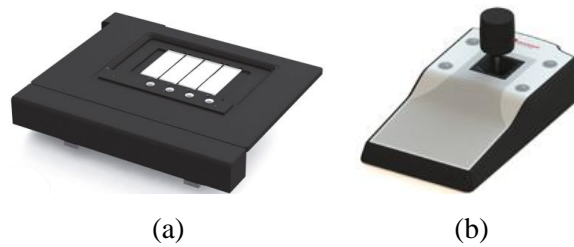


Figure 40 (a) Motorized base [109] and (b) joystick used in these experiments [110].

### 3.1.2 Software

There are six main programs used in the capture, manipulation and registration of the dataset:

- *SwitchBoard*: this software allows to control the mechanical slide holder remotely.
- *Pylon Viewer*: this program makes possible the capture and real-time visualization of the RGB camera.
- *vnirCameraController\_v2*: this IUMA's software was developed to control the motorized base and to capture the push-broom HS images, obtaining an HS cube.
- *HyperCube*: this program is used to observe the HS cube capture previously.
- *Pannoramic Viewer*: this software is used to visualize and obtain RGB images from the digitized slides.
- *Matlab 2022b*: this IDE is used to develop each piece of code used to pre-process the images, to perform the image registration in the dataset, and to evaluate the results..

### 3.1.3 Other materials

Additional materials employed in this TFG are:

- **Personal Computers (PC) and storage units**: both to capture the dataset and to develop the code needed for the registration, a couple of PCs with *Windows 10*, from *Microsoft*; 8 GB of Random Access Memory (RAM), an *Intel Core i5* Central Processing Unit (CPU) and an external storage unit *WD Elements* of a capacity of 1 TB.
- **Glass slides and WSI RGB images**: the biopsy samples used to generate HS images and its corresponding WSI scanners RGB images were given by the Hospital Virgen de la Cinta in Tortosa, Catalonia, Spain. These RGB images have been carefully labelled at cellular level by their pathologists.

## 3.2 Methodology

This section will show the methodology followed in the acquisition, pre-processing and registration of the dataset, allowing other colleges to replicate the results shown in Chapter 4.

### 3.2.1 Acquisition framework

The first step of this process is the selection of the glass slide to be captured. Reading the identification (ID) tag, the corresponding WSI RGB image can be visualized using the *Pannoramic Viewer* software. Then, the slide is mounted in the microscope and the lamp for translucent samples is powered (Figure 41).

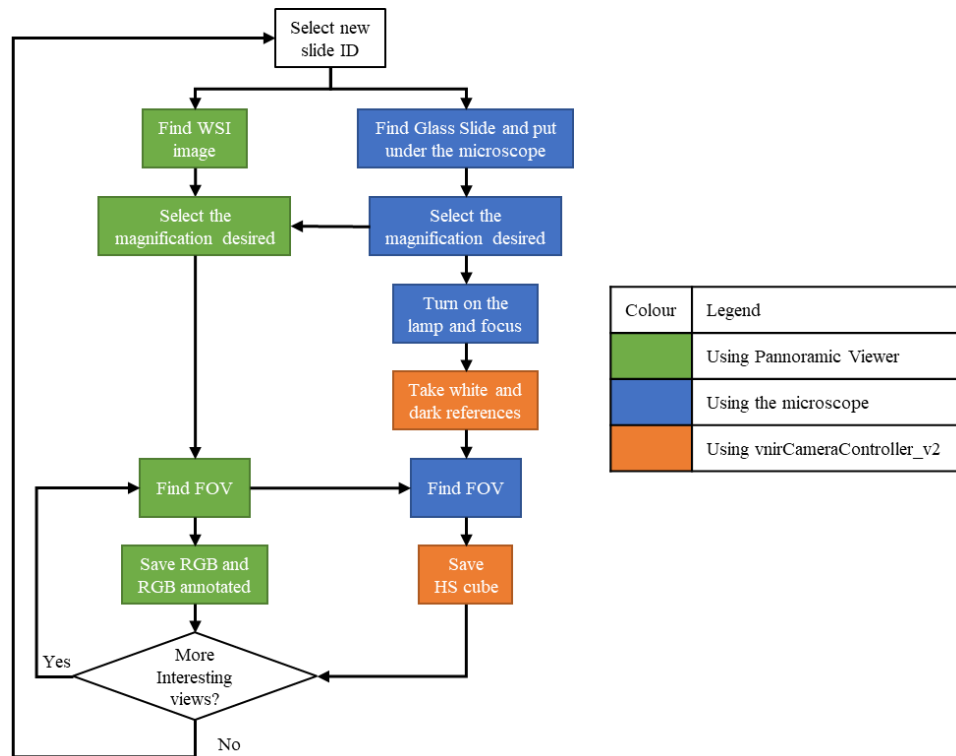


Figure 41 Workflow of image acquisition.

The next step is to bring the sample in focus using the chosen magnification, in this case 20×. This is done by moving the sample with the joystick, pushing it forward/backward and left/right to move in the X and Y axes and rotating the joystick wheel to lift up and down in the Z axis. The *Pylon Viewer* software is used to visualize the input light of the RGB camera and so checking if the samples are in focus (Figure 42 (left)).

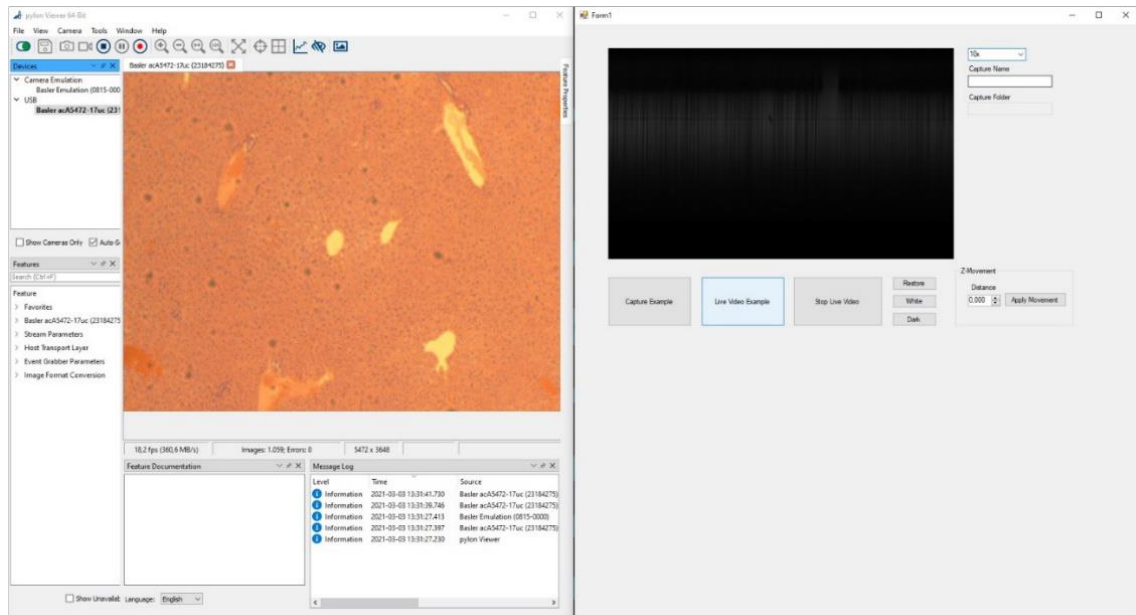


Figure 42 (left) Pylon Viewer and (right) vnrCameraController\_v2 user interfaces.

Once the sample is focused, a white and a dark reference must be captured to normalize the light range that may vary between captures due to changes in the environment light. The white reference needs the camera to be pointing to a part of the slide with no tissue, while for the dark reference, it can be taken simply by turning off the lamp power supply. The references are obtained using *vnrCameraController\_v2* (Figure 42 (right)). To enable the control of the slide holder by the PC, the software *SwitchBoard* must be initiated.

At this point, the RGB and HS images can be taken, using *Pannoramic Viewer* and *vnrCameraController\_v2*, respectively. As explained in Section 2.5.2, it is desirable that the images to be registered are as similar as possible, reducing the search space, i.e., using a similar FOV. Therefore, the RGB image will be taken by taking the same portion of the sample that the microscope is pointing and using the same magnification.

To capture an image with the *Pannoramic Viewer* software (Figure 43), the export button, symbolized by a camera, must be pressed. In the pop-up window, the “Show scale bar” must be unselected [111]. As the registration transformation will be obtained using a non-annotated RGB image, but will be applied to the labels, both images must be taken at the same time by selecting/unselecting the “Show annotations”. The extension used to save the images is Portable Network Graphics (PNG).

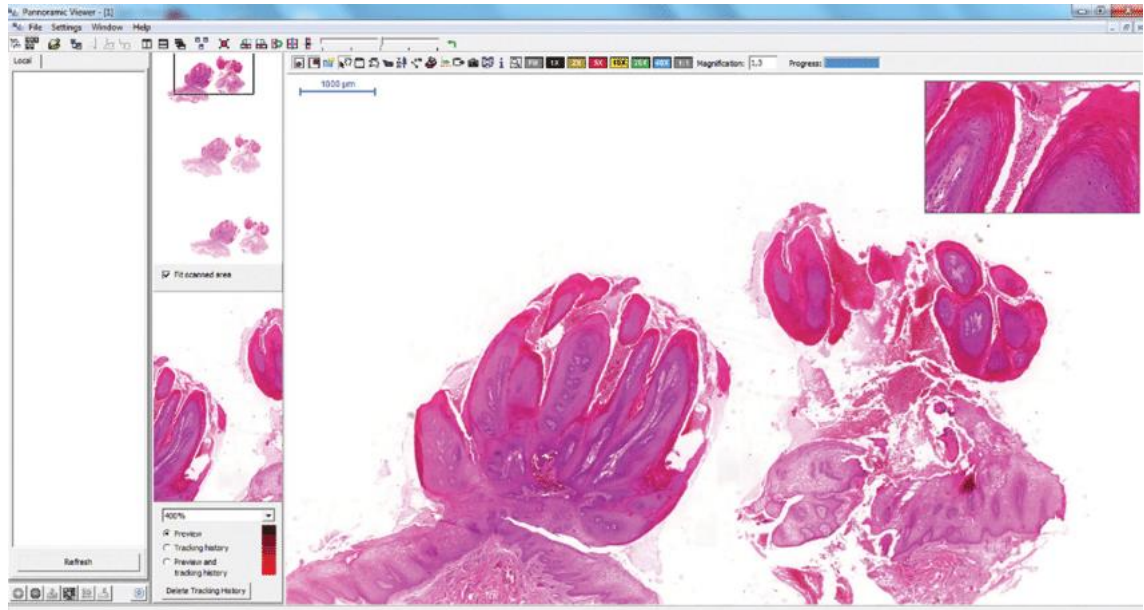


Figure 43 Panoramic Viewer user interface [112].

On the other hand, to capture the HS cubes, the magnification used must be selected and then press the “Capture Example” button. While capturing the HS cubes, it is important to keep the microscope as still as possible to avoid vibrations that will disturb the result. When obtaining the images, it is crucial to use a systematic naming method to easily cluster each FOV of each image. With this purpose, the images obtained were named following this criterion: {Sample ID}\_{Magnification used}\_C{Index of the point of view} {If annotated: “\_annotated”}.

### 3.2.2 Pre-processing framework

The pre-processing of the HS cubes consists of correcting the distortions produced by the changes in the ambient light or environmental conditions. In order to reduce these alterations, the HS cubes are normalized using the mean of the white and dark references as extreme values (equation (28)). The next step is the creation of a synthetic RGB image from the HS cube to perform the registration to it. This image is obtained by simulating the sensitivity of the human eye to white light (Figure 44). This is done to work with a visible version of the HS cube and to use similar bands of the images, reducing the computational costs with no reduction of the structural information. Another important pre-processing step is the correction of reflections between the images to be registered, as the code developed cannot solve such distortions.

$$Normalized\ Pixel = \frac{Pixel - \overline{Dark\ Reference}}{\overline{White\ Reference} - \overline{Dark\ Reference}} \quad (28)$$

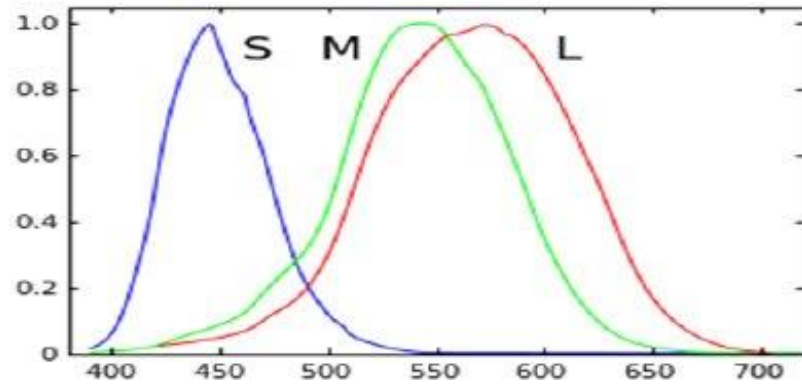


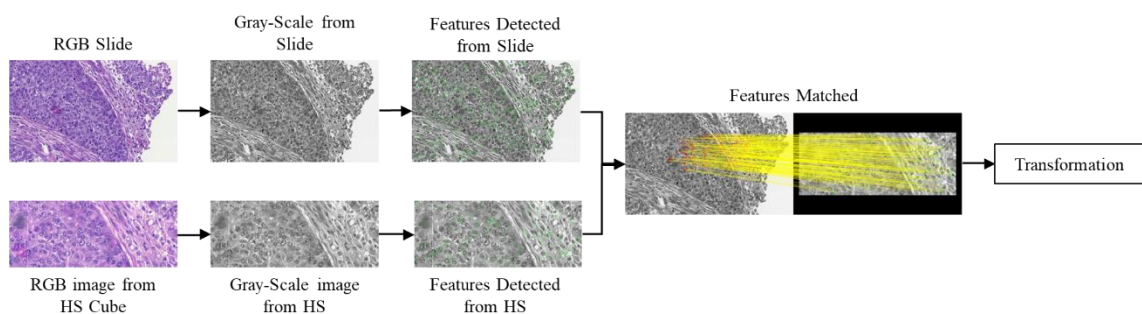
Figure 44 Normalized responsivity spectra of the human eye [113].

### 3.2.3 Registration process and evaluation

The code necessary to perform the registration of the dataset and the evaluation of the different methods was performed on the IDE *Matlab 2022a* using the *Computer Vision* and *Statistics and Machine Learning* toolboxes. The IDE and toolboxes selection was done considering different medical image registration research papers [114] [115] [116]. To illustrate the process of registration and evaluation of the dataset, code fragments developed exclusively for this project will be shown and explained. Many safety and visualization functions will be skipped, the complete code is shown in Section 7.2 Developed code .

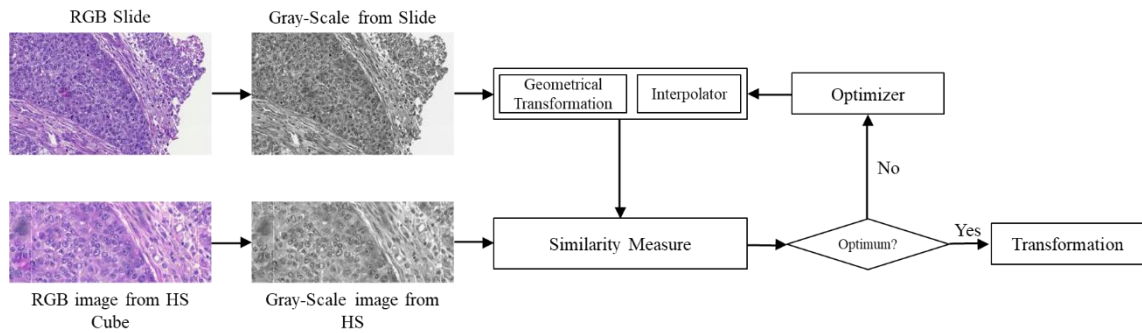
Following the classification shown in Section 2.5, the algorithms implemented are spatial (2D-2D), multimodal and automatic. The variations corrected are global and affine, the interpolator used is linear, and the subject is intrasubject of breast tissue (object). The two intrinsic registration methods have been applied, feature registration and intensity registration, following the workflow from Figure 45, using different similarity measures and optimizers.

The registration process has been developed using the synthetic RGB image as a template, so, to obtain the registered image, the computed transformation must be applied to the RGB slide. In the following subsections, there is shown every function and piece of code developed to perform the registration and to evaluate the results.

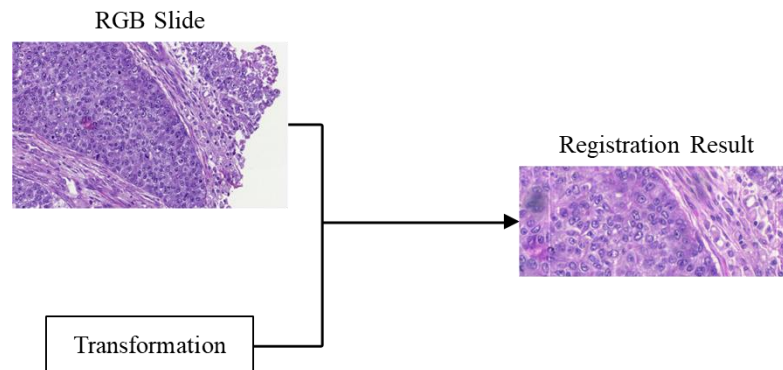


(a)





(b)



(c)

Figure 45 Workflow of (a) feature registration, (b) intensity registration, and (c) final image result.

### 3.2.3.1 Feature Registration

It has been created a function called *FeatureRegistration* which uses as input parameters the two images and other optional values such as *detector*, *tfType*, *showFeatures*, *showMatches* or *showRegister*. Typing the command “*help FeatureRegistration*” will show a brief explanation of the function and its parameters (Code Fragment 1).

```
[tform,Ir] = FeatureRegistration (moved,fixed,varargin)
This function registers the moved image to the fixed template and returns the
transformation (tform) and the moving image registered (Ir).
Optional parameters: detector, tfType, showFeatures, showMatches, showRegister
detector (default = "SURF"): string that defines the detector used to detect features.
showFeatures (default = true): logical to show the MSER features extrated of each image.
showMatches (default = true): logical to show the matched points.
tfType (default = "affine"): string that defines the type of transformation.
showRegister (default = true): logical to show the register result.
```

Code Fragment 1 Help *FeatureRegistration* output.

The first step of this function is to define the variables that may not be given by the user and to assign their default values (Code Fragment 2). Once the default values have been assigned, the code checks the optional parameters to modify its values if needed or to show some warnings in case of misassignment (Code Fragment 3).

```
detector = "SURF";  
tfType = "affine";
```

Code Fragment 2 Variable definition of FeatureRegistration function.

```
switch varargin{i}7  
    case "detector"  
        if ischar(varargin{i+1}) || isstring(varargin{i+1})  
            detector = string(varargin{i+1});  
        else  
            warning("The value of the parameter " + varargin{i} + " is not valid. It  
will be ignored.");  
        end  
    case "tfType"  
        [...]8  
    otherwise  
        warning("The parameter " + varargin{i} + " is not valid. It will be ignored.");  
end
```

Code Fragment 3 Read and modify the input parameters.

The next step is to modify the images to a form where the registration methods from the *Computer Vision* toolbox, can be used to perform the registration, i.e., a grey scale image, so they will be read and grayscaled or simply grayscaled<sup>9</sup>, depending on if a path to the image or the image itself is passed (Code Fragment 4)

```
switch class(moved)  
    case 'string'  
        moved = rgb2gray(imread(moved));  
    case 'uint8'  
        if ndims(moved)==3  
            moved = rgb2gray(moved);  
        end  
    otherwise  
        error("The moving image does not have a useful format.")  
end  
  
switch class(fixed)  
    [...]  
end
```

Code Fragment 4 Read and transform to grey scale the images.

When the images are ready, the features from the selected detector can be found (Code Fragment 5). The detectors supported by the *Computer Vision* toolbox are the ones listed in Table 1 (*MSER*, *MinEigen*, *Harris*, *FAST*, *ORB*, *SURF*, *KAZE*, *BRISK* and *SIFT*) and the functions that detect each feature follow this structure: detect{detector}Features.

<sup>7</sup> The `varargin` is an array where each cell stores an optional input.

<sup>8</sup> The “[...]” means that part of the code have been hidden due to its repetition or to simplify the comprehension of the code.

<sup>9</sup> The grey scalation of an image is done by forming a weighted sum of the R, G, and B components. The weights are selected following the BT.601 standard [118]:  $0.2989 * R + 0.5870 * G + 0.1140 * B$ .



```

switch detector
  case "MSER"
    reg1=detectMSERFeatures(moved);
    reg2=detectMSERFeatures(fixed);
  case "MinEigen" [...]
  case "Harris" [...]
  case "FAST" [...]
  case "ORB" [...]
  case "SURF" [...]
  case "KAZE" [...]
  case "BRISK" [...]
  case "SIFT" [...]
end

```

Code Fragment 5 Detect the feature points of each image.

Once the features are detected, the next step is the extraction of the descriptors and location and orientation of each key point with the method *extractFeatures*. Then, the points are matched using the *matchFeatures* method, which applies the SSD similarity metric to obtain the distance between descriptors, optimizing it to get the indexes of the feature points paired. As not every point will be matched, those paired can be separated from the rest (Code Fragment 6).

```

[f1,val1]=extractFeatures(moved,reg1);
[f2,val2]=extractFeatures(fixed,reg2);

indexPairs = matchFeatures(f1,f2);

match1=val1(indexPairs(:,1),:);
match2=val2(indexPairs(:,2),:);

```

Code Fragment 6 Extract and match the descriptors.

At this point, the transformation can be estimated by giving the function *estimateGeometricTransform2D* the matched points, and the search space, i.e., the type of transformation desired. To visualize the result, the *moved* image can be warped using the transformation obtained, taking as reference the coordinate system of the fixed image (Code Fragment 7)

```

tform = estimateGeometricTransform2D(match1,match2,tfType);
Ir=imwarp(moved,tform,'OutputView',imref2d(size(fixed)));

```

Code Fragment 7 Perform the transformation.

### 3.2.3.2 Intensity registration

It has been developed a function called *IntensityRegistration* which uses as input parameters the two images, as a relative path to the files, RGB matrices or grey scale matrices, and other optional parameters such as *modality* or *tfType*. Using the command “*help IntensityRegistration*” will show a brief explanation of the function and its parameters (Code Fragment 8).

```
[tform, Ir] = IntensityRegistration(moved, fixed, varargin)
This function registers the moved image to the fixed template and returns
the transformation (tform) and the moving image registered (Ir).
Optional parameters: modality, tfType
optimizer (default = "1+1 EA"): string that defines the optimizer to use on
the iterative process.
metric (default = "MMI"): string that defines the metric to use on the
iterative process.
tfType (default = "affine"): string that defines the type of transformation.
```

*Code Fragment 8 Help IntensityRegistration output.*

As with the previous function, the first step is the definition and default values assignment of the optional parameters (Code Fragment 9). The next steps are similar to the feature registration case: check the optional parameters and read and transform to grey scale the images (Code Fragment 3 and Code Fragment 4). Once the parameters are established and the images are grayscaled, the definition of the similarity metric and the optimizer algorithm used in the iterative registration process can be performed. While other authors tune some parameters of the optimizer and the similarity metric [114] [115] [116], that would be an interactive or semiautomatic registration. As the objective of this TFG is to obtain an automatic algorithm, the default configuration will be used (Code Fragment 10). As the optimizer and the similarity metric have been defined, the iterative process of finding the transformation may be started by calling the *imregtform* function (Code Fragment 11).

```
tfType = "affine";
metric = "MMI";
optimizer = "1+1 EA";
```

*Code Fragment 9 Variable definition of IntensityRegistration function.*

```
switch optimizer
    case "1+1 EA"
        optimizer=registration.optimizer.OnePlusOneEvolutionary;
    case "RSGD"
        optimizer=registration.optimizer.RegularStepGradientDescent;
    otherwise
        warning("The optimizer selected is not supported. The default value will be
used.")
        optimizer=registration.optimizer.OnePlusOneEvolutionary;
end

switch metric
    case "MMI"
        metric=registration.metric.MattesMutualInformation;
    case "MSE"
        metric=registration.metric.MeanSquares;
    otherwise
        warning("The optimizer selected is not supported. The default value will be
used.")
        metric=registration.metric.MattesMutualInformation;
end
```

*Code Fragment 10 Definition of the similarity metric and optimizer.*

```
tform=imregtform(moved, fixed, tfType, optimizer, metric, "InitialTransformation", tformEstimate);  
Ir=imwarp(moved, tform, 'OutputView', imref2d(size(fixed)));
```

Code Fragment 11 Perform the iterative search of the transformation.

### 3.2.3.3 Validation of the registration methodology

As every method is implemented by built-in *Matlab* functions, it seems logical to claim that they work for common RGB images. However, the context of this TFG needs the validation of these methods using our specific dataset. With this purpose, a known transformation will be applied to some images, and then the resulting image will be registered with the original one using each detector or pair of optimizer/similarity metric listed in Table 2. By measuring the difference between the original image and the registered one, it is possible to determine whether or not the implementation is effective. This validation process has been done to ten random images of the dataset following the next code (Code Fragment 12).

```
phi=[-10 -10];  
scale=[0.59 0.59];  
xtranslation=[200 200];  
ytranslation=[50 50];  
  
knownTform =  
randomAffine2d("Rotation", phi, "Scale", scale, "XTranslation", xtranslation, "YTranslation", ytranslation);  
  
invKnownTform=invert(knownTform);  
[...]  
I=imread(strcat(folder, imageList(randi(length(imageList),1))));  
Ir=imwarp(I, knownTform);  
  
[unknownTform{j,i}, newI] =  
FeatureRegistration(Ir, I, "showFeatures", false, "showMatches", false, "showRegister", false, "tfType", "similarity", "detector", methods{i}); %Feature Registration  
  
[...]  
  
[unknownTform{j,i}, newI] =  
IntensityRegistration(Ir, I, "optimizer", methods{i}{1}, "metric", methods{i}{2}, "showRegister", false, "tfType", "similarity"); %Intensity Registration  
end  
  
[...]  
mseVs(j,i)=immse(newI, rgb2gray(I));  
ssimVs(j,i)=ssim(newI, rgb2gray(I));
```

Code Fragment 12 Validation of the registration methodology.

Intensity-based Optimizers / Similarity Metrics	1+1 EA / MMI
	1+1 EA / MSE
	RSGD / MMI
	RSGD / MSE
Feature-based Detectors	MSER
	MinEigen
	Harris
	FAST
	ORB
	SURF
	KAZE
	BRISK
	SIFT

Table 2 List of each pair of optimizer/similarity metric and detector used.

#### 3.2.3.4 Evaluation of the results

To evaluate the performance of each registration method, three metrics are calculated: Structural Similarity Index Metric (SSIM) and MSE between the fixed and registered images, as well as the execution time required for the image registration. To try to reduce the distortion introduced by the system or the bias caused by random initialization, every image will be registered ten times, and the metrics are calculated as the mean of all the repetitions.

##### 3.2.3.4.1 Evaluation of the feature registration

To evaluate the performance of each detector used for the feature registration, the algorithm described in Section 3.2.3.1 will be applied to the entire dataset. The time values collected represent the ones needed to detect the features of the two images to be registered. To obtain these metrics, four methods are used: *tic* and *toc*, to start and stop a timer; *ssim*, to calculate the SSIM of two images; and *immse*, to obtain the MSE of two images.

```
tic;
regs{1}={detectMSERFeatures(moved),detectMSERFeatures(fixed)};
time(i,1)=toc;
tic;
regs{2}={detectMinEigenFeatures(moved),detectMinEigenFeatures(fixed)};
time(i,2)=toc;
[...]
ssimVals(i,j)=ssim(Ir,fixed);
mseVals(i,j)=immse(Ir,fixed);10
```

Code Fragment 13 Evaluation of the performance of the feature registration algorithm.

<sup>10</sup> The *i* variable refers to the *i*<sup>th</sup> image of the dataset and the *j* variable refers to the *j*<sup>th</sup> feature detector evaluated.

#### 3.2.3.4.2 Evaluation of the intensity registration

As with the feature registration, the intensity registration will be evaluated using the same metrics and methods. In this case, the time evaluated is the one elapsed between the beginning of the iterative registration process and the transformation of the *moved* image.

```
tic;
tformEstimate = imregcorr(moved, fixed, "similarity");
tform=imregtform(moved, fixed, tfType, optimizer, metric, "InitialTransformation", tformEstimate);
Ir=imwarp(moved, tform, 'OutputView', imref2d(size(fixed)));
time(i, j, u)=toc;
ssimVals(i, j, u)=ssim(Ir, fixed);
mseVals(i, j, u)=immse(Ir, fixed);11
```

Code Fragment 14 Evaluation of the performance of the intensity registration algorithm.

#### 3.2.3.5 Analysis of variance test of the results

To prove that the results are statistically truthful, an Analysis Of Variance (ANOVA) test, which studies the significant difference between means of different groups, can be performed using a significance level of 0.05. To visualize the difference between the different groups, four different charts will be used: a box chart, a plot of the Probability Density Function (PDF) of each group, a graph where the means and confidence level of each group are shown, and a heatmap of the p-values from the ANOVA tests performed to each pair of methods. For each chart, a different function has been developed.

##### 3.2.3.5.1 GenerateBoxChart function

The *GenerateBoxChart* function takes as inputs an array (*groups*) with the names of each group and a matrix (*data*), where each column stores the data of each group. To generate the box chart, an iterative process is performed, where each column of *data* is plot with its corresponding label.

```
boxchart(categorical(k), data(:, groups==string(i)))
```

Code Fragment 15 Plot box chart of each group.

##### 3.2.3.5.2 PlotNormDistribution function

The *PlotNormDistribution* function uses as inputs a matrix (*data*) where the information is stored and an array (*groups*), similar to the previous function, and also requires as inputs the x and y limits of the chart. To generate the PDF of each group, the mean and standard deviation is extracted using the method *fitdist* and then the PDF is calculated for each desired x value using the method *normpdf*.

---

<sup>11</sup> The i variable refers to the i<sup>th</sup> image of the dataset, the j and the u variable refers to the j<sup>th</sup> optimizer and u<sup>th</sup> similarity metric evaluated.

```
norm=fitdist(data(:,i), "Normal");  
y_norm = normpdf(xlimits,norm.mu,norm.sigma);  
plot(xlimits,y_norm)
```

Code Fragment 16 Plot the PDF of a group.

### 3.2.3.5.3 GenerateAnovaPValueTable function

Finally, the *GenerateAnovaPValueTable* function inputs are similar to the previous ones a matrix (*data*) and an array (*groups*), where the groups and their names are stored respectively. In this case, two figures will be generated, both showing the results of the ANOVA tests, qualitatively and quantitatively. The *anova1* method extracts some statistics from the groups that the *multcompare* function uses to perform the ANOVA tests. The p-values are stored at *comparison* and then processed to generate the heatmap.

```
[~,~,stats] = anova1(groups,{}, "off");  
[comparison,means,H,~] = multcompare(stats, "Display", "on");  
[...]  
h = heatmap(pTable, "GridVisible", "off", "MissingDataColor", "w");
```

Code Fragment 17 Plot of the ANOVA tests.

### 3.2.4 Annotation extraction and transformation

Once an image registration has experimentally shown the higher performance, it can be used to obtain the transformation needed to register the cell annotations to the synthetic RGB. However, a mask for the labels needs to be extracted. To perform this process, the function *ExtractAnnotations* has been developed. It uses as inputs the annotated RGB (*I*) and a collection of flags that indicates which annotations must be extracted or a set of RGB values of the labels. The method then search for those RGB values on the image and copy their location and value on a matrix (*tags*) (Code Fragment 18).

```
function [tags,mask]=ExtractAnnotations(I,varargin)
for i=varargin
    switch i{1}
        case "All"
            colours=[255 0 0; 255 242 0; 0 255 0; 0 174 239; 255 128 0; 0 0 255];
        case "Red" [...]
        case "Yellow" [...]
        case "Green" [...]
        case "Ligth Blue" [...]
        case "Orange" [...]
        case "Blue" [...]
        otherwise
            if ndims(i{i})==3 && max(i{i}(:))<=255 && min(i{i})>=0
                colours(end+1,:)=i{i}; [...]
            end
        end
    end
end

for i=1:size(colours,1)
    colourMask=(I(:, :,1)==colours(i,1) & I(:, :,2)==colours(i,2) &
I(:, :,3)==colours(i,3)); % Search for the pixel with same RGB values

    [rows, columns] = find(colourMask); % Extract their location
    [...]

    for j=1:length(rows)
        tags(rows(j),columns(j),:)=colours(i,:); % Copy the RGB values of each pixel
    end
end
end
```

Code Fragment 18 Extraction of the annotations.

As the annotations are separated from the slide, they can be registered and overlaid over the synthetic RGB. With that purpose Code Fragment 19 has been made. This script scans each annotated RGB to extract all the annotations. Then register the synthetic RGB with the not annotated RGB and applies that transformation to the labels. Finally, the registered annotations and the synthetic RGB are overlaid.

```
for i=imageUrlList
    [tags,mask]=ExtractAnnotations(strcat(annotatedDir,i{1})(1:end-4),
    "_annotated.png"), "All");
    Ihs=imread(strcat(hsDir,i{1}));

    [tform,Ir]=FeatureRegistration(strcat(slidesDir,i{1}),Ihs,"detector",detector,"showFeatures",false,"showMatches",false,"showRegister",false);
    % or
    [tform,Ir] =
    IntensityRegistration(Ir,I,"optimizer",optimizer,"metric",metric,"showRegister",false);

    tagsR=imwarp(tags,tform,"OutputView",imref2d(size(Ihs))); % Register annotations
    maskR=imwarp(mask,tform,"OutputView",imref2d(size(Ihs))); % Register mask of
    annotations

    [rows,columns]=find(maskR==0); % Where there is no annotations
    Ihs_annotated=zeros(size(Ihs),"uint8");

    for j=1:length(rows)
        Ihs_annotated(rows(j),columns(j),:)=Ihs(rows(j),columns(j),:); % Copy the
        synthetic RGB
    end

    Ihs_annotated=Ihs_annotated+tagsR; % Where there is annotations, copy them
end
```

Code Fragment 19 Generation of the synthetic RGB with annotations.

### 3.3 Summary

This chapter has indicated every material used in capturing, pre-processing, and registering the dataset. Afterwards, the methodology applied to obtain them is shown, from the manipulation of the microscope to obtain the HS cubes, to the code developed to perform the image registration and the analysis of its outcomes.



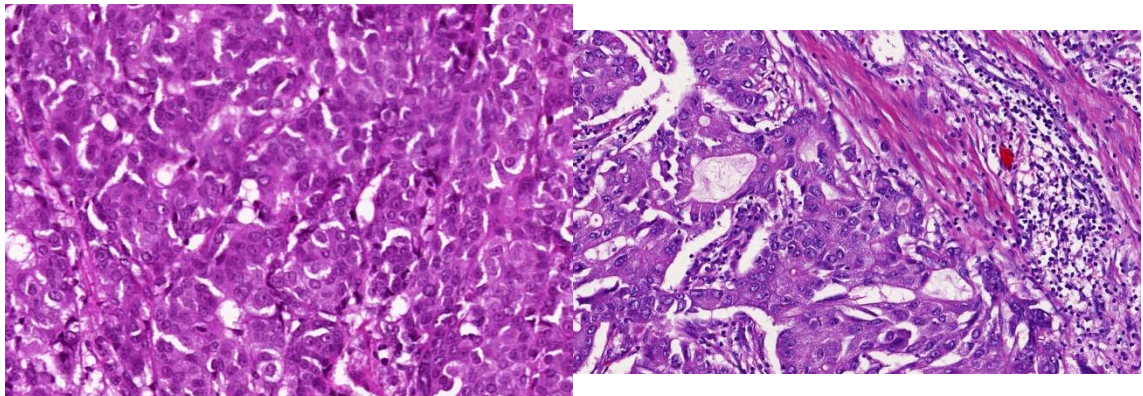
## 4 Results

---

In this chapter, the results obtained, following the described methodology and using the materials illustrated at the previous chapter, will be shown. Firstly, a short description of the dataset and a couple of examples of the RGB and HS images captured in this TFG will be shown. Then, examples of the outputs from the registration functions presented will be displayed. Finally, an analysis of the ANOVA test results is performed and analysed, resulting in a rank of the best registration techniques.

### 4.1 Dataset

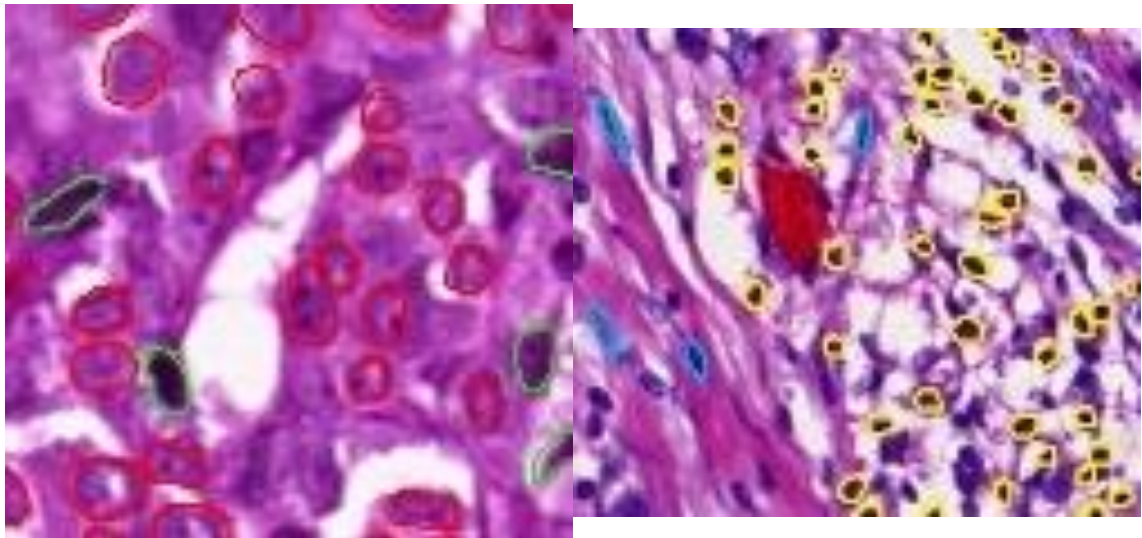
The resulting dataset, captured from 18 different biopsy samples<sup>12</sup>, consists of three groups, non-annotated RGBs (Figure 46 (a)), annotated RGBs (Figure 46 (b)), and synthetic RGBs from the HS cubes (Figure 46 (c)). Each group is composed by 490 images, 1470 in total. Those groups were kept isolated in different folders to identify and manipulate them easily.



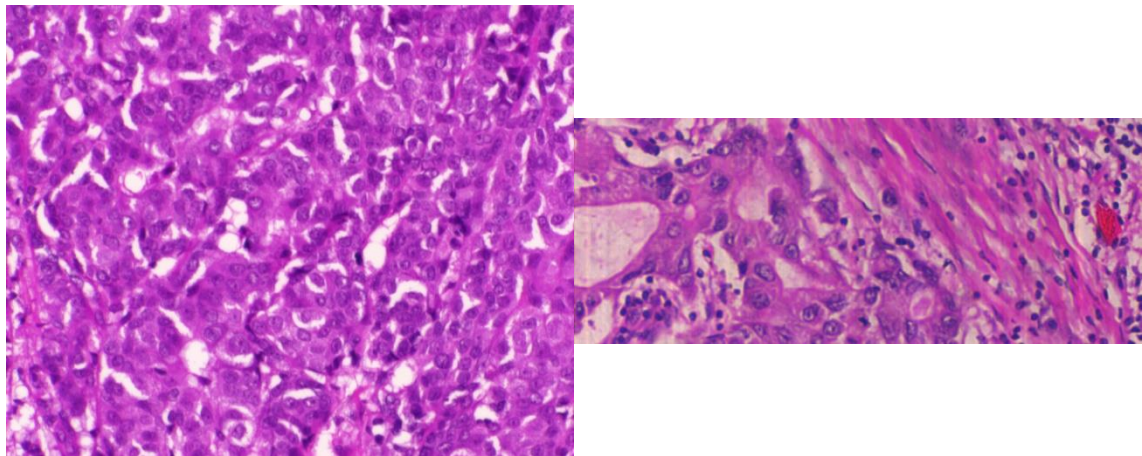
(a)

---

<sup>12</sup> In Section 7.1 an example of images captured from each biopsy sample and the result of its registration is illustrated.



(b)



(c)

Figure 46 Examples of the dataset captured: (a) not annotated RBGs, (b) annotated RBGs<sup>13</sup>, and (c) synthetic RBGs.

## 4.2 Method validation

The validation of the functions developed to perform the intensity-based and feature-based registration shows the results illustrated at Figure 47. While at the top row, (a) and (b), two box charts can be seen comparing the SSIM values of each method; at the second row, (c) and (d), a similar comparison is shown using the MSE<sup>-1</sup> results.

---

<sup>13</sup> A zoom has been done to these examples to get an easier illustration of the annotations presented in the images.

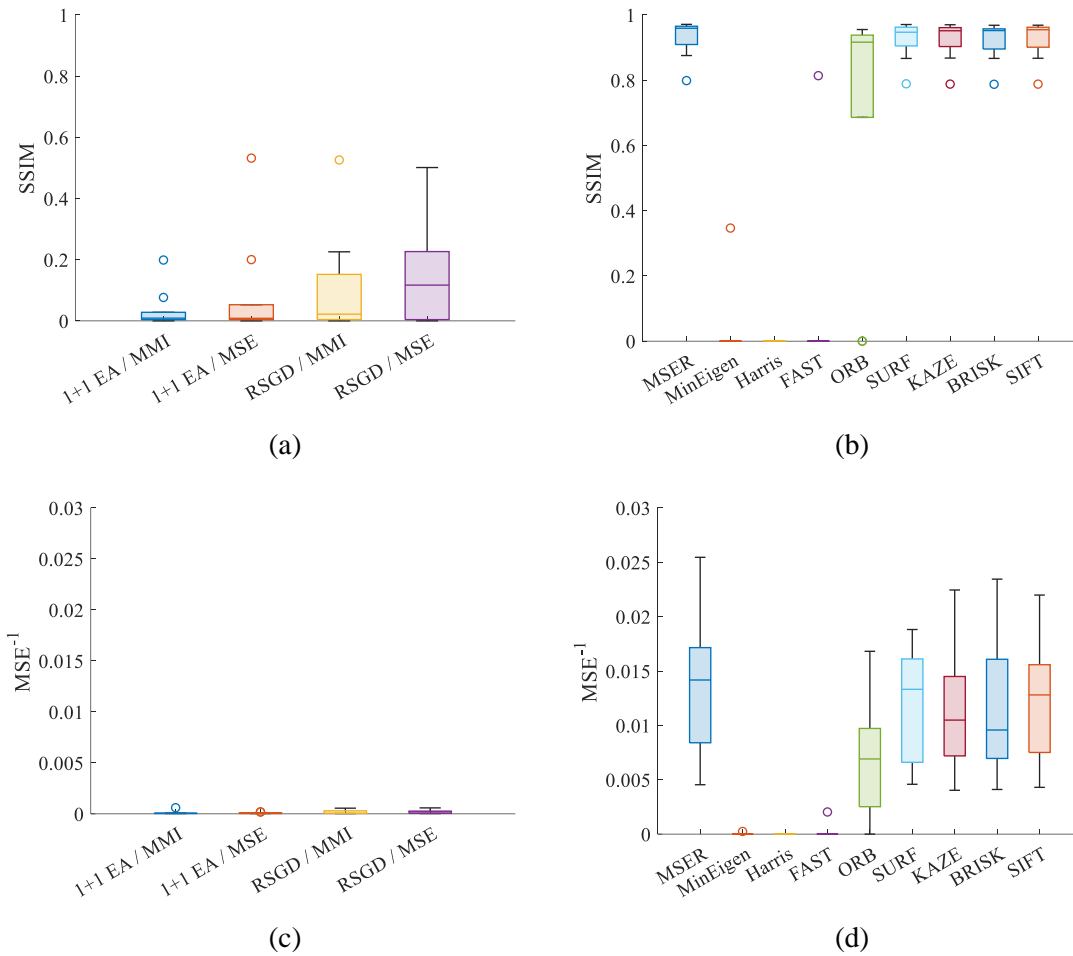


Figure 47 Method validation results for area-based registration ((a) SSIM and (c)  $MSE^{-1}$ ) and point-based registration ((b) SSIM and (d)  $MSE^{-1}$ ).

### 4.3 Intensity registration

The use of intensity values to perform the registration of the Figure 46 (a) and (c) led to results similar to the ones illustrated in Figure 48, where magenta and green regions show where the intensities are different. Performing this registration method to every pair of images of the dataset and following the evaluation shown at the previous chapter, Table 3 data can be extracted<sup>14</sup>.

<sup>14</sup> The MSE and time data will be displayed by its inverse to make every metric directly proportional to its performance, i.e., the higher the value of a metric, the better its execution.

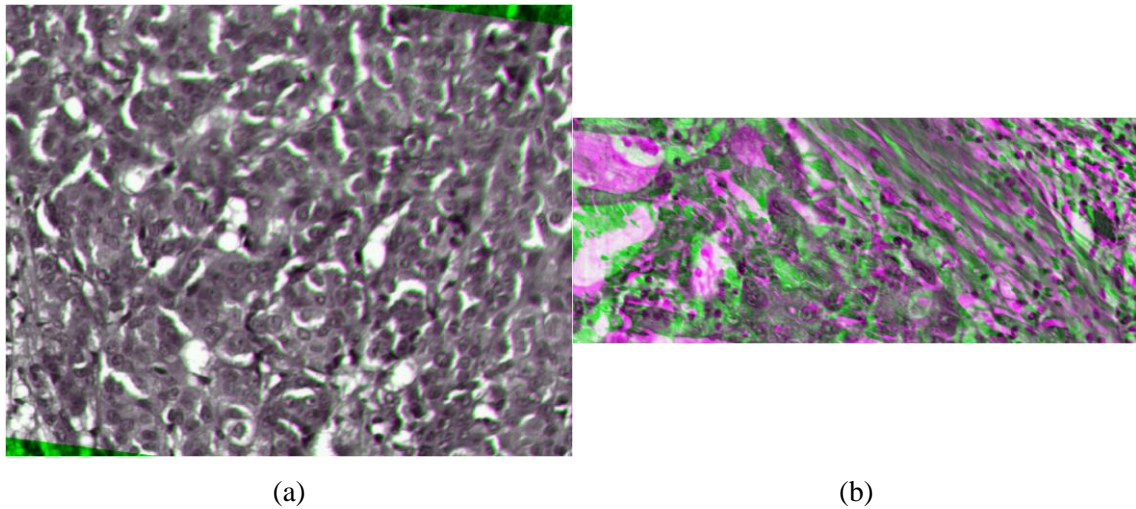


Figure 48 Results of intensity registration of the (a) left column and (b) right column of Figure 47.

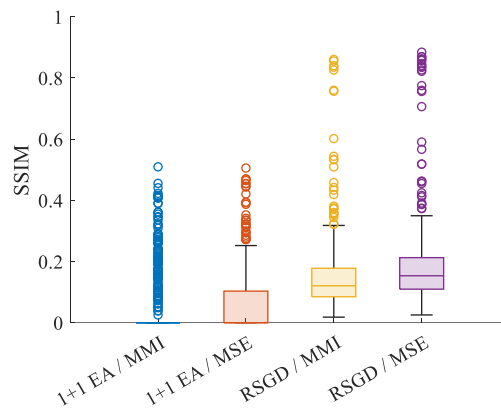
### 4.3.1 Metrics results for intensity registration

Using the SSIM,  $MSE^{-1}$ , and  $time^{-1}$  metrics to evaluate each pair optimizer / similarity metric performance, the following table and figure can be obtained. In Table 3 the maximum, mean, minimum, and quartiles of the evaluation metrics collected for each pair of images is shown, highlighting the higher values of each one. Next to the table, Figure 49 illustrates a box chart per metric comparing each couple of optimizers and similarity metrics.

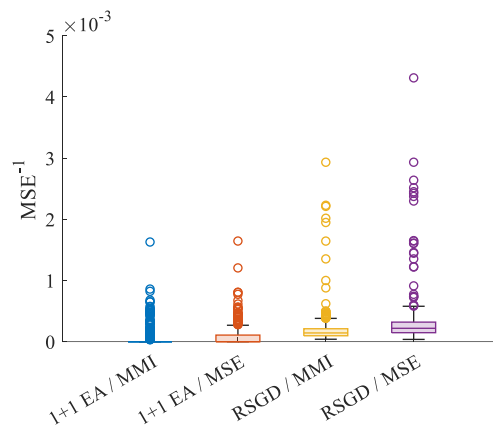
		SSIM			
Optimizer / Similarity Metric		1+1 EA / MMI	1+1 EA / MSE	RSGD / MMI	RSGD / MSE
Max		0.5095	0.5055	0.8604	<b>0.8842</b>
Mean		0.0443	0.0613	0.1518	<b>0.1897</b>
Min		0	0	0.0182	<b>0.0253</b>
Q3		0	0.1033	0.178	<b>0.2126</b>
Median		0	0	0.1206	<b>0.1534</b>
Q1		0	0	0.0847	<b>0.1096</b>
		$MSE^{-1} (x10^{-3})$			
Max		1.6	1.6	2.9	<b>4.3</b>
Mean		0.05	0.0757	0.1976	<b>0.3018</b>
Min		0	0	<b>0.04024</b>	0.03735
Q3		0	0.1079	0.2115	<b>0.3198</b>
Median		0	0	0.1429	<b>0.2188</b>
Q1		0	0	0.0969	<b>0.147</b>

	Time <sup>-1</sup> (seconds <sup>-1</sup> )			
Max	<b>0.5493</b>	0.4993	0.1191	0.3224
Mean	0.0473	0.1182	0.0568	<b>0.1592</b>
Min	0	0	0.0255	<b>0.0781</b>
Q3	0	<b>0.2327</b>	0.0675	0.1727
Median	0	0	0.0552	<b>0.1669</b>
Q1	0	0	0.0514	<b>0.1616</b>

Table 3 SSIM, MSE<sup>-1</sup> and Time<sup>-1</sup> results of intensity registration.

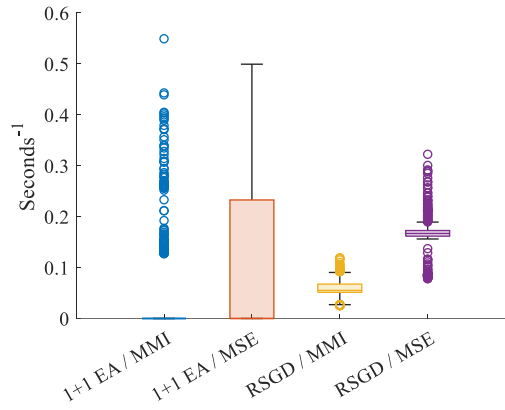


(a)



(b)



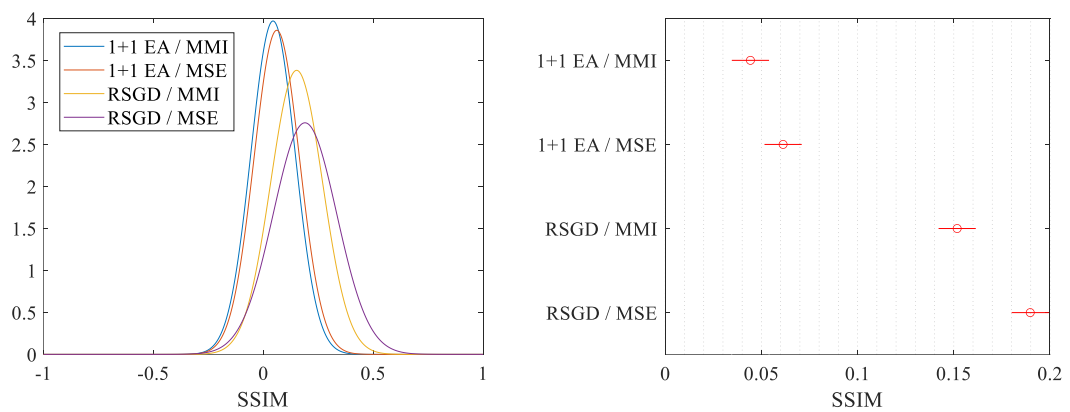


(c)

Figure 49 Box chart of (a) SSIM, (b)  $MSE^{-1}$ , and (c)  $Time^{-1}$  results for intensity registration.

### 4.3.2 Statistical analysis of the intensity registration

The next tests will indicate if any of the optimizer/similarity metric groupings have the same mean and therefore perform equivalently. To identify if this is the case of any of the methods, in Figure 50 is displayed the PDFs of each population (left column) and a multiple comparison (right column), where the circle is the mean, and the line represents the confidence interval of a 95%. So, if any pair of populations share the mean, they would look similar on the first graph, and their confidence intervals will overlap vertically. Quantitatively, this is illustrated by the p-values of the ANOVA tests shown in the heatmaps of Figure 51, where a light blue represents low p-values, i.e., the groups do not perform equally; and green to brown represents higher values, i.e., there is higher probability that the populations have similar means.



(a)

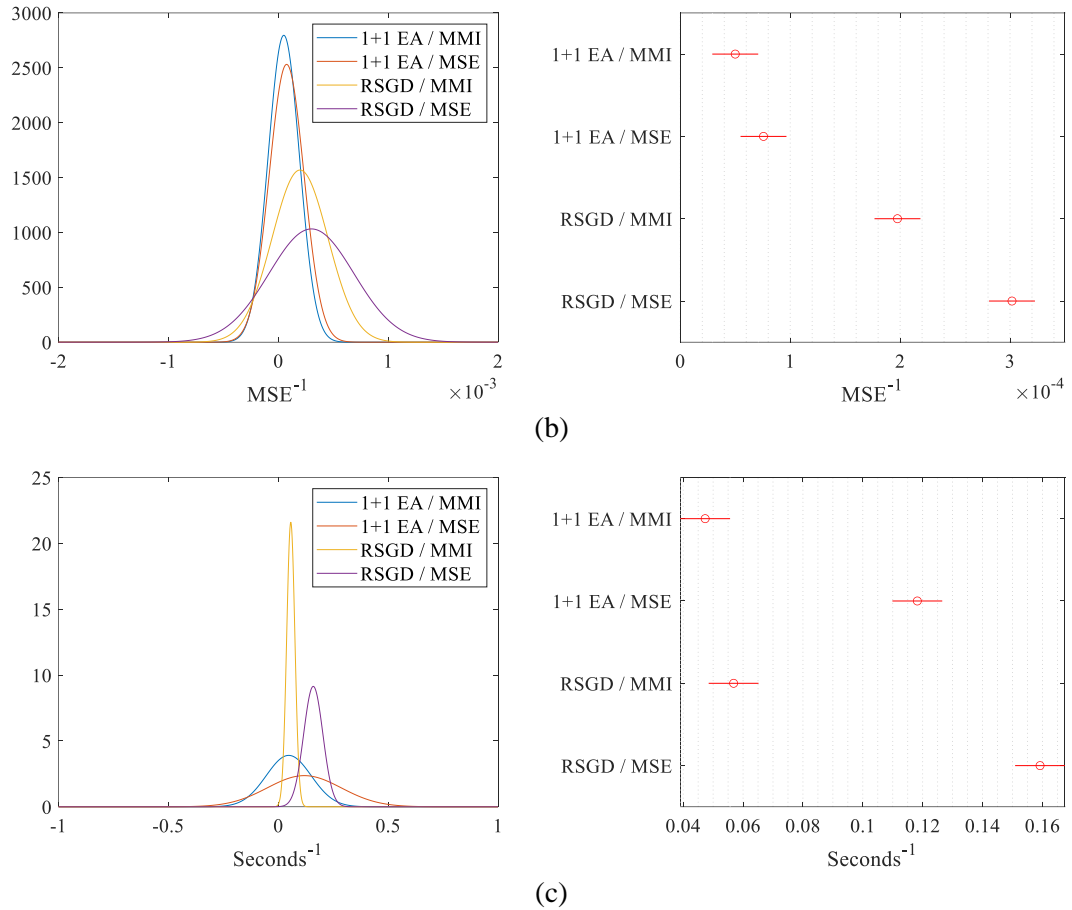
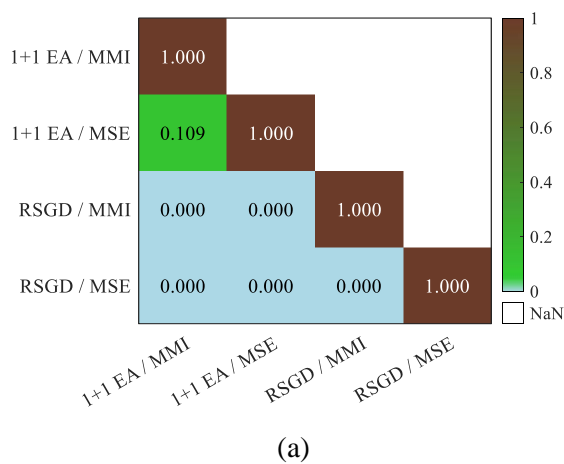


Figure 50 (left) PDF analysis and (right) multiple comparison of the means of (a) SSIM, (b)  $MSE^{-1}$ , and (c)  $Time^{-1}$  intensity registration.

The Figure 50 shows there is no overlap between the confidence levels of any group except for the first and third one. Therefore, there are strong evidence to affirm that the means of each pair are statistically different. This is illustrated quantitatively by the heatmaps of Figure 51.



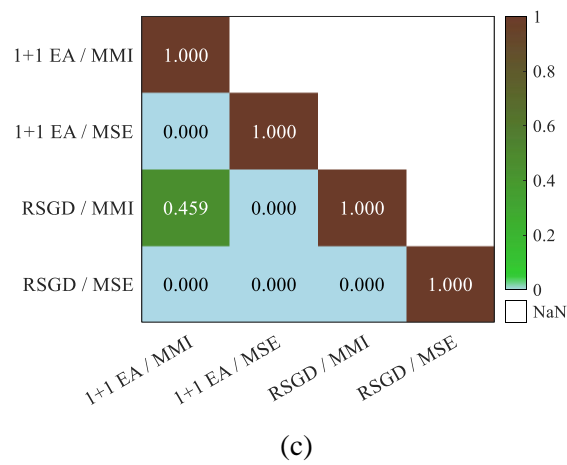
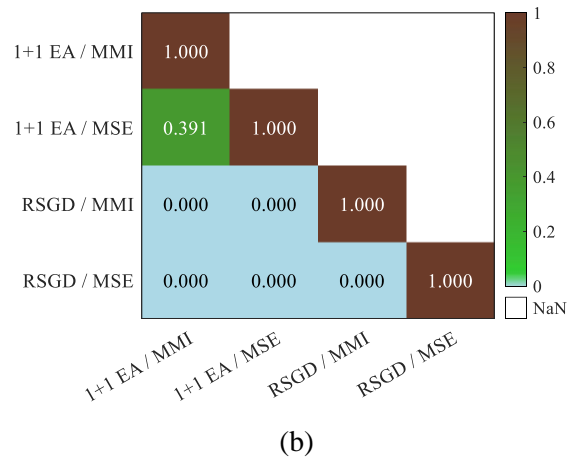
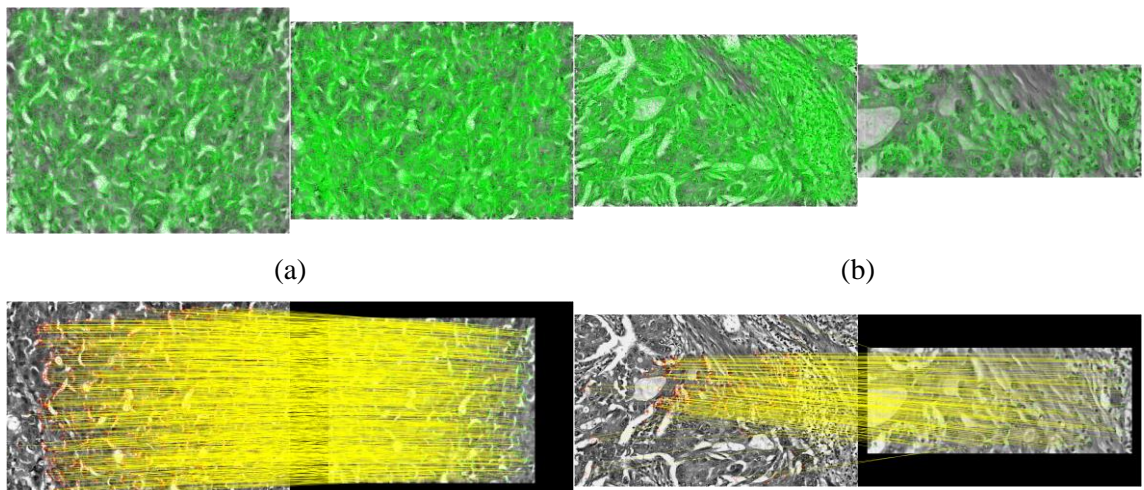


Figure 51 ANOVA  $p$ -value results for the (a) SSIM, (b)  $MSE^{-1}$  and (c)  $Time^{-1}$  intensity registration.

#### 4.4 Feature registration

Similar to the opening of the previous section, in Figure 52 a registration of the Figure 46 images using SURF features is illustrated. The first row, (a) and (b), displays the features detected on each image. The second one, (c) and (d), indicate the pairs of feature points matched. Finally, the third line, (e) and (f), presents the results of the registrations.





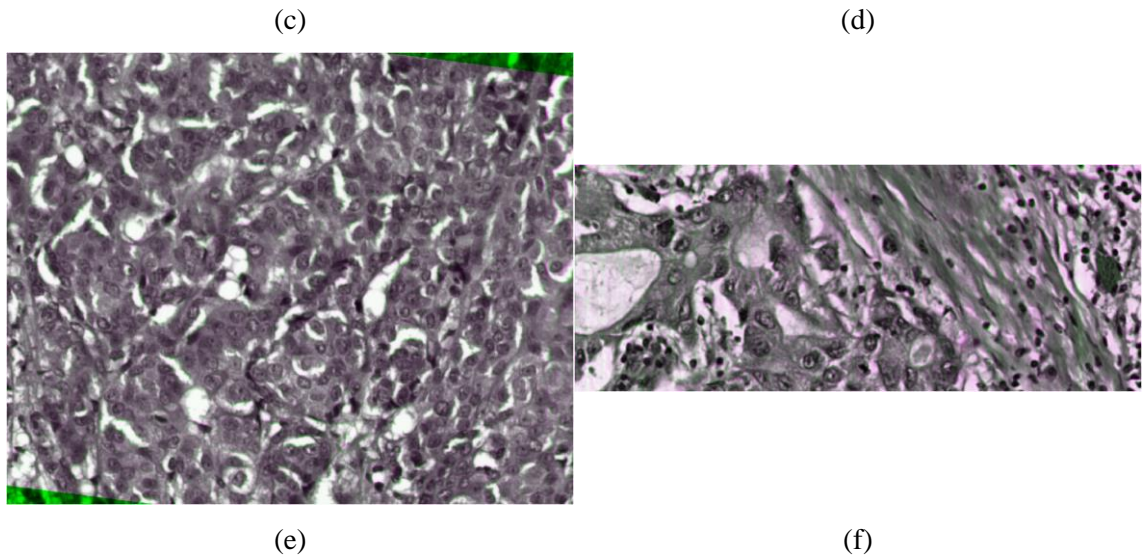


Figure 52 ((a) and (b)) SURF features detection, ((c) and (d)) pairing, and ((e) and (f)) result of the feature registration.

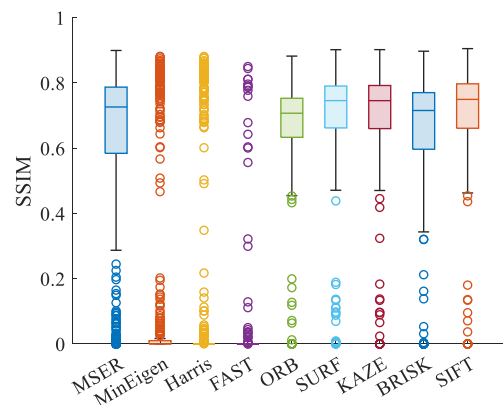
Performing this registration method with the detectors specified to every pair of images of the dataset, Table 4 data can be extracted.

#### 4.4.1 Metric results for feature registration

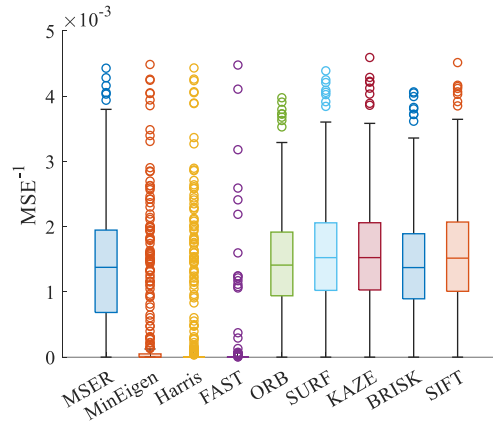
Similar to the analysis done for intensity registration, using the SSIM,  $MSE^{-1}$ , and  $time^{-1}$  metrics to evaluate each detector performance, Table 4 and Figure 53 are obtained. The table shows the maximum, mean, minimum and quartiles of the data collected, highlighting the higher values. The figure illustrates the box charts that contrast the performance of each detector.

SSIM									
Detector	MSER	MinEigen	Harris	FAST	ORB	SURF	KAZE	BRISK	SIFT
Max	0.8996	0.8815	0.8813	0.8503	0.8823	0.9019	0.9021	0.8973	<b>0.9052</b>
Mean	0.6032	0.1361	0.1285	0.025	0.6631	0.6971	<b>0.6993</b>	0.6422	0.6944
Min	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
Q3	0.7872	0.01	0	0	0.753	0.7905	0.7922	0.7703	<b>0.7974</b>
Q2/Median	0.7263	0	0	0	0.7073	0.7458	0.7457	0.7155	<b>0.7496</b>
Q1	0.5843	0	0	0	0.6334	<b>0.6621</b>	0.6601	0.5968	0.6511
MSE <sup>-1</sup> (x10 <sup>-3</sup> )									
Max	0.0044	0.0045	0.0044	0.0045	0.004	0.0044	<b>0.0046</b>	0.0041	0.0045
Mean	0.0014	0.0003	0.0003	0.0001	0.0014	<b>0.0016</b>	<b>0.0016</b>	0.0014	<b>0.0016</b>
Min	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
Q3	0.0019	0.0001	0	0	0.0019	<b>0.0021</b>	<b>0.0021</b>	0.0019	<b>0.0021</b>
Q2/Median	0.0014	0	0	0	0.0014	<b>0.0015</b>	<b>0.0015</b>	0.0014	<b>0.0015</b>
Q1	0.0007	0	0	0	0.0009	<b>0.001</b>	<b>0.001</b>	0.0009	<b>0.001</b>
Time <sup>-1</sup> (seconds <sup>-1</sup> )									
Max	5.0994	14.47	15.126	<b>524.95</b>	65.674	34.553	3.5567	4.8629	18.034
Mean	1.6177	10.698	11.345	<b>326.18</b>	22.991	23.305	2.1591	4.0001	13.009
Min	0.5724	5.8231	6.3984	<b>185.42</b>	13.233	13.086	0.8981	2.8171	7.5856
Q3	1.9101	11.699	12.348	<b>357.08</b>	25.242	25.799	2.398	4.2722	14.223
Q2/Median	1.4778	11.521	12.197	<b>324.37</b>	21.674	23.959	2.2477	4.0486	13.905
Q1	1.1291	11.081	11.785	<b>289.28</b>	19.209	21.931	2.0433	3.7179	13.04

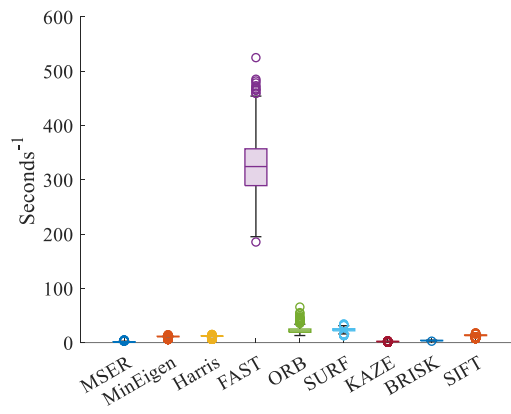
Table 4 SSIM, MSE<sup>-1</sup> and Time<sup>-1</sup> results of feature registration.



(a)



(b)

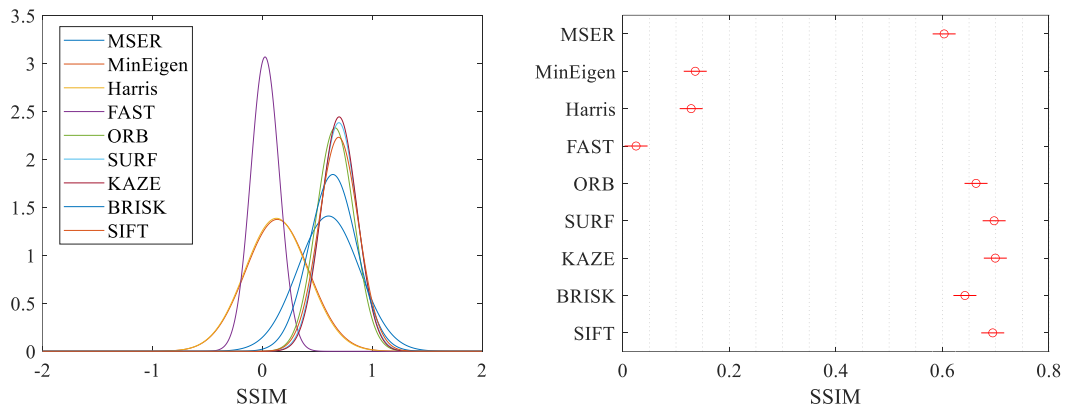


(c)

Figure 53 Box chart of (a) SSIM, (b)  $MSE^{-1}$ , and (c)  $Time^{-1}$  results for feature registration.

#### 4.4.2 Statistical analysis of feature registration

Homologous to Section 4.3.2, the following tests indicate which detectors have the same mean and therefore perform equivalently. The results are also represented by the PDFs of each population (left column of Figure 54) and by a multiple comparison (right column of Figure 54). Finally, the heatmaps of each ANOVA test is illustrated in Figure 55.



(a)

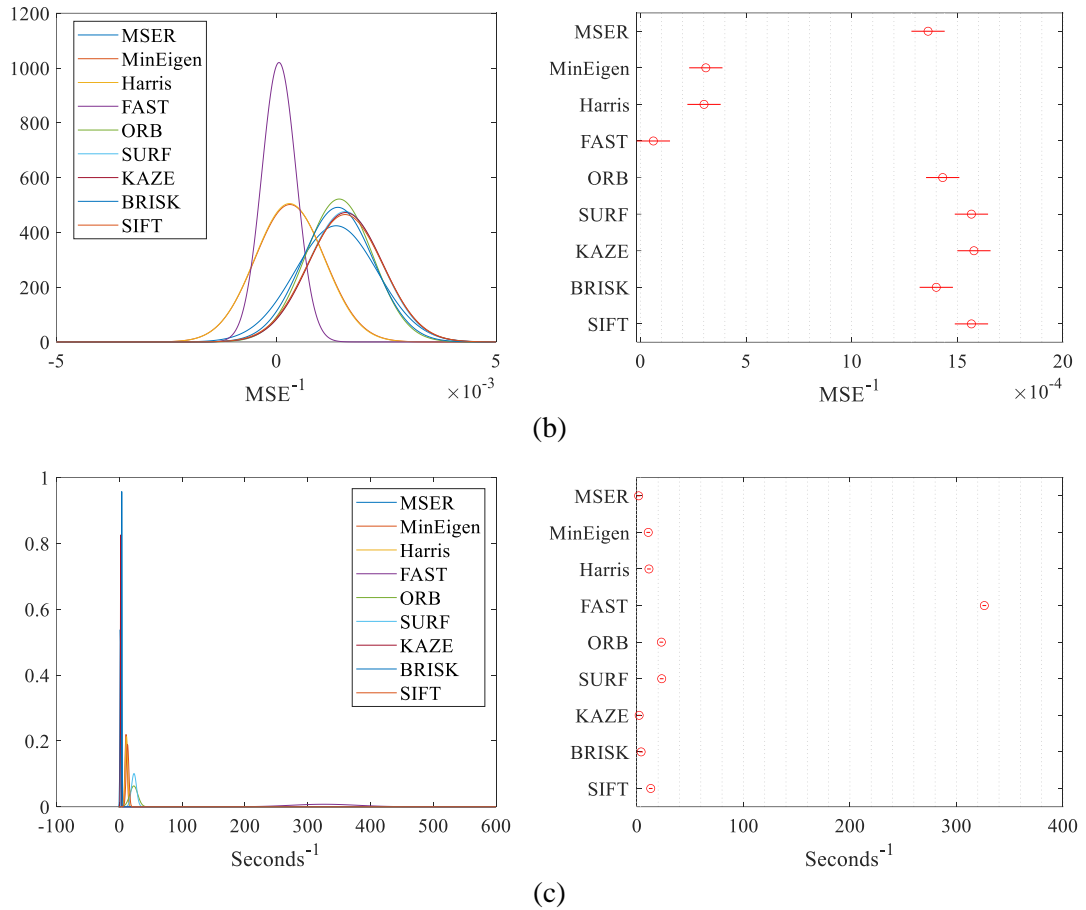
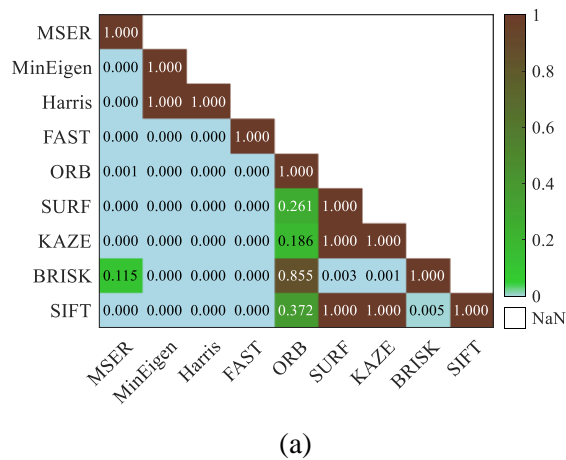
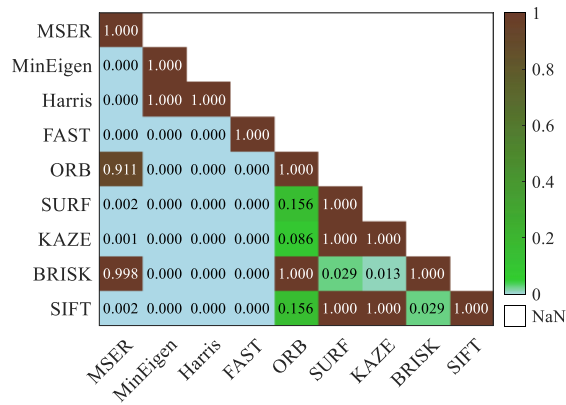


Figure 54 (left) PDF analysis and (right) multiple comparison of the means of (a) SSIM, (b)  $MSE^{-1}$ , and (c)  $Time^{-1}$  feature registration.

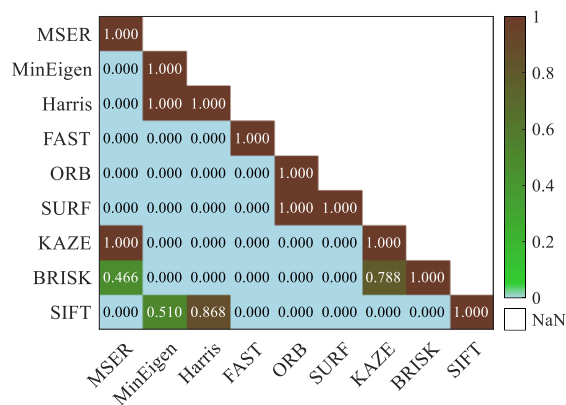
The results from Figure 54 shows that there are some pairs of detectors that performs equally, as the population of the metrics looks similar. Therefore, there are strong evidence to affirm that some pairs of detectors have same means while others do not. In Figure 55 it is shown a quantitative comparison of each pair of detectors.



(a)



(b)



(c)

Figure 55 ANOVA p-value results for the (a) SSIM, (b)  $MSE^{-1}$ , and (c)  $Time^{-1}$  feature registration.

## 4.5 Analysis of the results

As a form of synopsis of the results shown in Sections 4.3 and 4.4, Table 5 shows which registration method performs better on each metric. For the intensity-based registration methods there is a clear superior pair of optimizer/similarity metric, the RSGD/MSE one; in contrast with the feature-based one, where many detectors show good results.

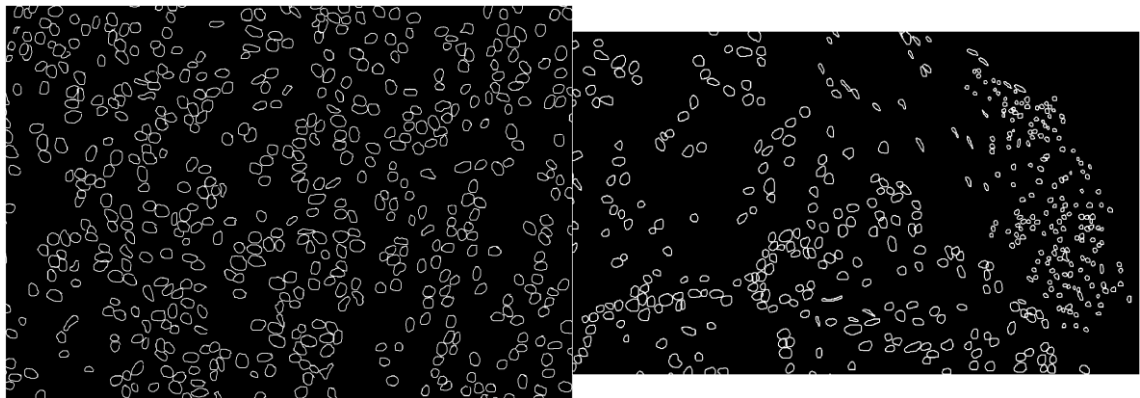
Metric	Intensity			Feature		
	SSIM	MSE <sup>-1</sup>	Time <sup>-1</sup>	SSIM	MSE <sup>-1</sup>	Time <sup>-1</sup>
Max	RSGD /	RSGD /	1+1 EA /	SIFT	KAZE	FAST
	MSE	MSE	MMI			
Mean	RSGD /	RSGD /	RSGD / MSE	KAZE	SURF / KAZE /	FAST
	MSE	MSE			SIFT	
Min	RSGD /	RSGD /	RSGD / MSE	=	=	FAST
	MSE	MSE				
Q3	RSGD /	RSGD /	1+1 EA /	SIFT	SURF / KAZE /	FAST
	MSE	MMI	MSE		SIFT	
Q2 / Median	RSGD /	RSGD /	RSGD / MSE	SIFT	SURF / KAZE /	FAST
	MSE	MSE			SIFT	
Q1	RSGD /	RSGD /	RSGD / MSE	SURF	SURF / KAZE /	FAST
	MSE	MSE			SIFT	

\* “=”: every detector performed equally.

Table 5 Summary of registration performance.

## 4.6 Synthetic annotated RGB results

Finally, in Figure 56 it is shown the results for the annotation extraction and synthetic annotated RGB generation scripts using as inputs the FOVs of Figure 46. The top row shows the mask of the annotations, the second one, the labels coloured extracted, and at the bottom row, the synthetic RGB with the cell annotations.



(a)

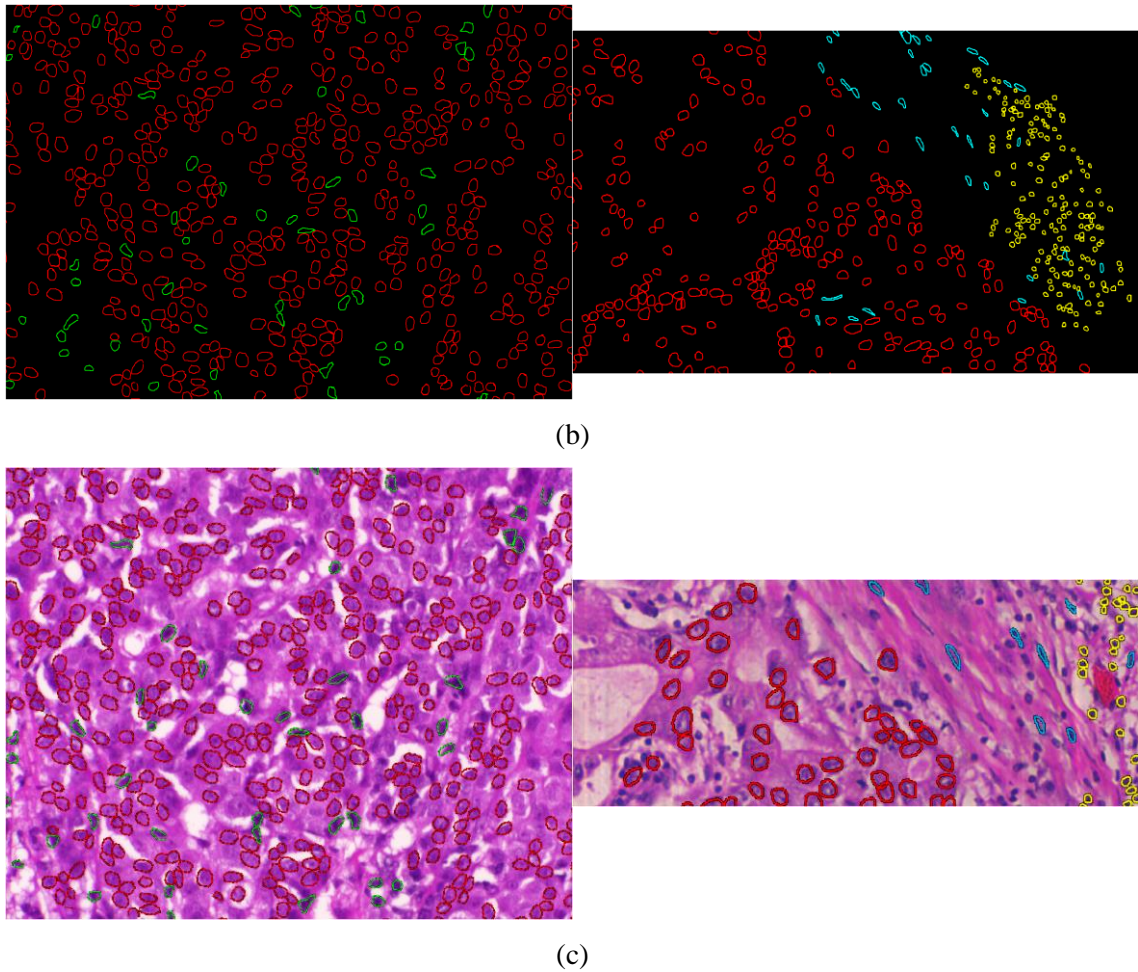


Figure 56 Result of the (a) mask creation, (b) annotation extraction, and (c) synthetic annotated RGB generation.

## 4.7 Summary

This chapter showed the experimental results obtained using the methodology and materials shown in the previous chapter. The final captured dataset is constituted by 1470 different images. The SSIM, MSE, execution time metrics, and its statistical analysis shown that, while the best intensity-based registration algorithm is the one that uses the RSGD optimizer and the MSE similarity metric, the feature-based registration has many detectors that gives similar results, such as SURF, SIFT or KAZE.

## 5 Conclusion

---

This last chapter will present the conclusions of the TFG, discussing about the results previously presented, exposing the limitations found during the process of capturing and registering the dataset, and proposing future lines that may improve the effectiveness and efficiency of the whole process.

### 5.1 Discussion

The validation of the registration algorithms developed reveals, thanks to Figure 47, the weaknesses of the use of the default parameters for intensity registration and the limitations of the detectors with no scale and rotation invariance (Table 1), as expected. Just by looking at the box charts of the SSIM values (first row of Figure 47), it stays clear that the area-based registration method used is far from optimal, while the uses of point-based registration perform almost ideally in most of the cases.

The ANOVA results for area-based registration shows there are two optimizer/similarity metric pairs statistically similar: the 1+1 EA/MMI and 1+1 EA/MSE. So, from Table 3, where the maximum, mean, minimum, and quartiles of each metric and pair are shown, we can determine that the highest value of every parameter studied is obtained using the RSGD/MSE couple. These results indicate that the RSGD/MSE pair is the most effective and efficient in time, and it is equally robust as the RSGD/MMI pair, as both have minimum non-null metrics.

The p-value of a statistical test reflects the probability that the null hypothesis is true, i.e., the probability that the means of two groups will have a significant difference. While typically with a p-value lower than 0.05, the null hypothesis is rejected, that does not mean that the alternative hypothesis is accepted and vice versa. Application of that in the case of feature-based registration, the ANOVA tests (Figure 54 (a) and (b)) show that, based on their performance, the detectors may be categorized into four groups with p-values higher than 0.95 between them:

- **Class A:** this set is formed by KAZE, SURF, and SIFT detectors.
- **Class B:** this cluster is formed by ORB, BRISK, and MSER detectors.
- **Class C:** this group is formed by MinEigen and Harris detectors.
- **Class D:** this one is formed by FAST detector exclusively.

Applying these categories to Table 4, the class A has the best performance, as it scores higher or equal at almost every metric. Looking at the minimum values, no detector of any class has a non-null value. This implies that every detector has failed at least one time while trying to perform the registration.



As every detector from class A has statistically equal effectiveness, to determine which one is the best, the time values must be taken into consideration. In this case, the fastest detector in general is FAST, but for the class A is SURF.

The disparity between Table 3 and Table 4 illustrates the poor performance of the intensity-based registration in contrast to the feature-based one, as expected from the method validation. While the highest SSIM median of the first one barely exceeds 15%, the one of the second one surpasses the 75%; 60 percentage points of difference approximately. Looking at the minimum values from the same tables, there are only two methods that have not failed when trying to perform the registration, the RSGD/MMI and the RSGD/MSE. In conclusion, while using the SURF detector is more effective, and the use of the RSGD/MSE pair is more robust.

After analysing the results of each method, the registration methodology proposed is the use of feature-based registration with the SURF detector and, in case the SSIM value is under a threshold, the use of the SIFT detector as an emergency backup. This way, the effectivity of the class A detectors is obtained while upgrading the robustness of the system, with the drawback of losing some efficiency in case of using both detectors.

## 5.2 Limitations

While applying the methodology shown in the third chapter, three main limitations were found. Firstly, the PC and its peripheral devices, i.e., keyboard and mouse, were placed on the same table where the microscope is. So, when it is capturing the HS cubes, as there should not be any vibration on the base of the microscope, any of those devices can be used, limiting this way the parallelization of the process.

The next issue was matching the FOV of the microscope and the one at the *Pannoramic Viewer* due to the vast displacement of the image with small changes performed by the motorized slide holder, consequence of the magnification selected, and to the similarity of the surrounding tissues. Also, the naming process of every capture is a tedious task that may be improved.

Lastly, the concerns while performing the registration are the general lack of robustness of the different methods and the amount of time it had required to execute all the registrations, as, for a single image and a single method, it may take up to forty seconds per registration (Table 3).

### 5.3 Future lines

To reduce or eliminate some of the previous limitations exposed there are a couple of future lines that, once implemented, would improve the workflow and the results of the dataset capture and registration. The first one is the development of an auto-focus system, as this is one of the most critical points of the image acquisition process.

Once this issue is solved, it would be ideal to develop a homologous of the WSI scanners for HS images, this way, instead of having many images for a biopsy sample, there will only be one, reducing the number of registrations to perform. Once the registration process is complete, the HS WSI image can be cropped using the magnification desired to increase the volume of the dataset with no redundancy of the information.

When a WSI scanner for HS images can be used, artifacts to the glass slides can be introduced to use an extrinsic registration method instead of an intrinsic registration method, e.g., if a one-millimetre square is printed at the slides, the registration process could consist of identifying the four corners of the square on each image and trying to match them.

Also, as mentioned in Section 2.2, there is no standardized protocol in the dye of the biopsy samples and that might lead to differences among HS cubes. So, to reduce the uncertainty of the results, a similar methodology to generate the dataset like the presented by this project can be implemented using not stained slides (Figure 57).

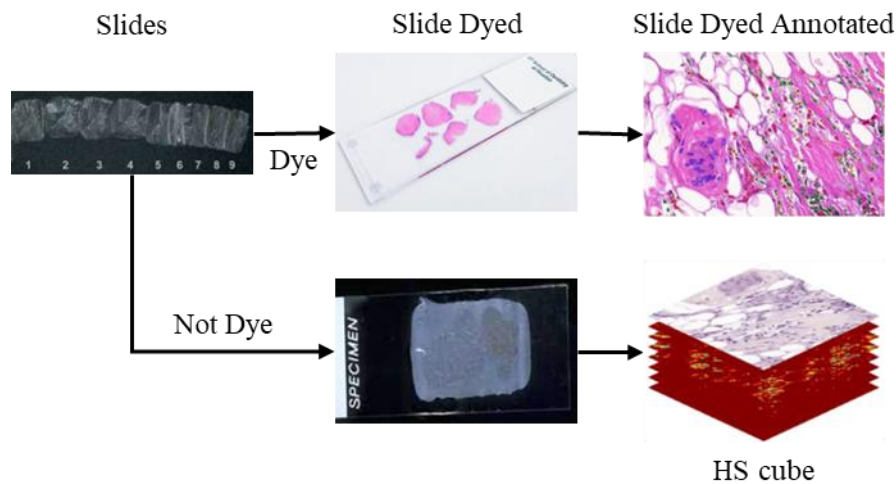


Figure 57 Methodology to obtain similar RGB annotated slices and HS cubes [33] [34] [3].

If the previous upgrades are implemented, the process can be completely automated, as there is no need for any more human interaction. It may be automated using the same microscope and a robotic arm to mount and dismount the glass slides or by using some kind of conveyor belt that move the sample underneath the HS camera. This process might be supervised by a Supervisory Control and Data Acquisition (SCADA), controlled by a Programmable Logic Controller (PLC) and communicated by industrial communication networks such as Modbus or Process Field Bus (PROFIBUS). As this methodology might be implemented compactly, many different hospitals may have the capability to implement it, making faster breast cancer diagnosis, reducing the amount of work for histopathologist and improving the survival rate of the patients.

## 6 Bibliography

---

- [1] “Cancer Today.” [https://gco.iarc.fr/today/online-analysis-table?v=2020&mode=population&mode\\_population=continents&population=900&populations=900&key=asr&sex=0&cancer=39&type=0&statistic=5&prevalence=0&population\\_group=0&ages\\_group%5B%5D=0&ages\\_group%5B%5D=17&group\\_cancer=1&include\\_nmsc=0&include\\_nmsc\\_other=1](https://gco.iarc.fr/today/online-analysis-table?v=2020&mode=population&mode_population=continents&population=900&populations=900&key=asr&sex=0&cancer=39&type=0&statistic=5&prevalence=0&population_group=0&ages_group%5B%5D=0&ages_group%5B%5D=17&group_cancer=1&include_nmsc=0&include_nmsc_other=1) (accessed Oct. 31, 2022).
- [2] “DSI – Instituto Universitario de Microelectrónica Aplicada.” <https://www.iuma.ulpgc.es/div-dsi/> (accessed Jan. 23, 2022).
- [3] S. Ortega *et al.*, “Hyperspectral imaging and deep learning for the detection of breast cancer cells in digitized histological images,” <https://doi.org/10.1117/12.2548609>, vol. 11320, pp. 206–214, Mar. 2020, doi: 10.1117/12.2548609.
- [4] L. M. Dale *et al.*, “Hyperspectral Imaging Applications in Agriculture and Agro-Food Product Quality and Safety Control: A Review,” <http://dx.doi.org/10.1080/05704928.2012.705800>, vol. 48, no. 2, pp. 142–159, Mar. 2013, doi: 10.1080/05704928.2012.705800.
- [5] L. Wang, D. Liu, H. Pu, D. W. Sun, W. Gao, and Z. Xiong, “Use of Hyperspectral Imaging to Discriminate the Variety and Quality of Rice,” *Food Anal. Methods*, vol. 8, no. 2, pp. 515–523, Feb. 2015, doi: 10.1007/S12161-014-9916-5/FIGURES/5.
- [6] D. Lorente, N. Aleixos, J. Gómez-Sanchis, S. Cubero, O. L. García-Navarrete, and J. Blasco, “Recent Advances and Applications of Hyperspectral Imaging for Fruit and Vegetable Quality Assessment,” *Food Bioprocess Technol.*, vol. 5, no. 4, pp. 1121–1142, May 2012, doi: 10.1007/S11947-011-0725-1/TABLES/3.
- [7] S. Mahesh, D. S. Jayas, J. Paliwal, and N. D. G. White, “Hyperspectral imaging to classify and monitor quality of agricultural materials,” *J. Stored Prod. Res.*, vol. 61, pp. 17–26, Mar. 2015, doi: 10.1016/J.JSPR.2015.01.006.
- [8] M. A. Calin, S. V. Parasca, D. Savastru, and D. Manea, “Hyperspectral Imaging in the Medical Field: Present and Future,” <http://dx.doi.org/10.1080/05704928.2013.838678>, vol. 49, no. 6, pp. 435–447, Aug. 2013, doi: 10.1080/05704928.2013.838678.
- [9] A. M. Siddiqi *et al.*, “Use of hyperspectral imaging to distinguish normal, precancerous, and cancerous cells,” *Cancer Cytopathol.*, vol. 114, no. 1, pp. 13–21, Feb. 2008, doi: 10.1002/CNCR.23286.
- [10] R. M. Youngson, *Collins dictionary of Medicine Fourth Edition*. 2005.
- [11] M. Heron, “National Vital Statistics Reports Volume 70, Number 9 July 26, 2021 Deaths: Leading Causes for 2019,” 2021, Accessed: Jan. 24, 2022. [Online]. Available: <https://www.cdc.gov/nchs/products/index.htm>.

- [12] R. L. Siegel, K. D. Miller, H. E. Fuchs, and A. Jemal, "Cancer statistics, 2022," *CA. Cancer J. Clin.*, vol. 72, no. 1, pp. 7–33, Jan. 2022, doi: 10.3322/CAAC.21708.
- [13] "What Is Breast Cancer? | CDC." [https://www.cdc.gov/cancer/breast/basic\\_info/what-is-breast-cancer.htm](https://www.cdc.gov/cancer/breast/basic_info/what-is-breast-cancer.htm) (accessed Jan. 27, 2022).
- [14] "Key Statistics for Breast Cancer in Men." <https://www.cancer.org/cancer/breast-cancer-in-men/about/key-statistics.html> (accessed Jan. 27, 2022).
- [15] The American Cancer Society, "Key Statistics for Breast Cancer," [Online]. Available: <https://www.cancer.org/cancer/breast-cancer/about/how-common-is-breast-cancer.html>.
- [16] A. M. Ilbawi and B. O. Anderson, "Cancer in global health: How do prevention and early detection strategies relate?," *Sci. Transl. Med.*, vol. 7, no. 278, Mar. 2015, doi: 10.1126/SCITRANSLMED.3008853.
- [17] "Breast cancer - Diagnosis and treatment - Mayo Clinic." <https://www.mayoclinic.org/diseases-conditions/breast-cancer/diagnosis-treatment/drc-20352475> (accessed Jan. 28, 2022).
- [18] "Lymph Nodes and Breast Cancer | Virginia Oncology Associates." <https://www.virginiacancer.com/breast-cancer/the-role-of-lymph-nodes-in-breast-cancer/> (accessed Oct. 31, 2022).
- [19] M. S. Davenport, "Choosing the safest gadoliniumbased contrast medium for MR imaging: Not so simple after all," *Radiology*, vol. 286, no. 2, pp. 483–485, Feb. 2018, doi: 10.1148/RADIOL.2017172224.
- [20] "Hospital Breast Scanner Digital Mammography X-ray Machine." <https://www.medical-equipmentmsl.com/sale-16216376-hospital-breast-scanner-digital-mammography-x-ray-machine.html> (accessed Oct. 31, 2022).
- [21] "New Breast Cancer Screening Imaging Method Finds Tumors in Seconds." <https://breastcancer-news.com/2018/07/02/new-breast-cancer-screening-imaging-approach-finds-tumors-in-seconds/> (accessed Oct. 31, 2022).
- [22] "Breast ultrasound imaging system - All medical device manufacturers." <https://www.medicalexpo.com/medical-manufacturer/breast-ultrasound-imaging-system-18181.html> (accessed Oct. 31, 2022).
- [23] "Breast Ultrasound (Sonogram) | DenseBreast-info, Inc." <https://densebreast-info.org/screening-technologies/breast-ultrasound/> (accessed Jan. 29, 2022).
- [24] "MRI, Near Infrared Spectral Tomography Increases Specificity in Breast Cancer Imaging | Imaging Technology News." <https://www.itnonline.com/content/mri-near-infrared-spectral-tomography-increases-specificity-breast-cancer-imaging> (accessed Oct. 31, 2022).
- [25] "Breast MRI." <https://www.siemens-healthineers.com/ec/magnetic-resonance-imaging/clinical-specialities/breast-mri> (accessed Oct. 31, 2022).
- [26] "What is Pathology? | DEPARTMENT OF PATHOLOGY - McGill University."

- <https://www.mcgill.ca/pathology/about/definition> (accessed Jan. 30, 2022).
- [27] “How Much Does an Molecular Pathology Level 5 Cost Near Me? - MDsave.” <https://www.mdsave.com/procedures/molecular-pathology-level-5/d584fccc> (accessed Mar. 05, 2022).
- [28] “Cytology | Johns Hopkins Medicine.” <https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/cytology> (accessed Mar. 05, 2022).
- [29] P. de Jager, E. Singh, B. Kistnasamy, and M. Y. Bertram, “Cost and cost-effectiveness of conventional and liquid-based cytology in South Africa: A laboratory service provider perspective,” *S. Afr. J. Obstet. Gynaecol.*, vol. 19, no. 2, pp. 44–48, 2013, doi: 10.7196/SAJOG.619.
- [30] “Histology Price List – The Department of Molecular & Comparative Pathobiology.” <http://mcp.bs.jhmi.edu/histology-price-list/> (accessed Mar. 06, 2022).
- [31] “An Intro to Specimen Preparation for Histopathology.” <https://www.leicabiosystems.com/knowledge-pathway/an-introduction-to-specimen-preparation/> (accessed Nov. 05, 2022).
- [32] A. I. Knyazkova, A. V. Borisov, L. V. Spirina, and Y. V. Kistenev, “Paraffin-Embedded Prostate Cancer Tissue Grading Using Terahertz Spectroscopy and Machine Learning,” *J. Infrared, Millimeter, Terahertz Waves*, vol. 41, no. 9, pp. 1089–1104, Sep. 2020, doi: 10.1007/S10762-020-00673-7/FIGURES/2.
- [33] “TISSUE PROCESSING.”
- [34] “Histotechnology: Helping save lives, one slide at a time - News and Media - About - UTHealth School of Dentistry.” <https://dentistry.uth.edu/about/news-media/story.htm?id=41ee048e-a7d7-48f7-a1df-ba1e1f833e25> (accessed Nov. 05, 2022).
- [35] N. Jain, “Essentials Before Sending Biopsy Specimens: A Surgeon’s Perspective and Pathologists Concern,” doi: 10.1007/s12663-011-0234-9.
- [36] M. Veta, J. P. W. Pluim, P. J. Van Diest, and M. A. Viergever, “Breast cancer histopathology image analysis: A review,” *IEEE Trans. Biomed. Eng.*, vol. 61, no. 5, pp. 1400–1411, 2014, doi: 10.1109/TBME.2014.2303852.
- [37] “Histology: Stains and section interpretation | Kenhub.” <https://www.kenhub.com/en/library/anatomy/interpretation-of-histologic-sections-stains-used-in-histology> (accessed Mar. 06, 2022).
- [38] “After a Biopsy: Making the Diagnosis | Cancer.Net.” <https://www.cancer.net/navigating-cancer-care/diagnosing-cancer/reports-and-results/after-biopsy-making-diagnosis> (accessed Jan. 30, 2022).
- [39] C. Perkins *et al.*, “Why current breast pathology practices must be evaluated. A Susan G. Komen for the Cure white paper: June 2006,” *Breast J.*, vol. 13, no. 5, pp. 443–447, Sep. 2007, doi: 10.1111/J.1524-4741.2007.00463.X.

- [40] A. C. Wolff *et al.*, “American Society of Clinical Oncology/College of American Pathologists guideline recommendations for human epidermal growth factor receptor 2 testing in breast cancer,” *J. Clin. Oncol.*, vol. 25, no. 1, pp. 118–145, Jan. 2007, doi: 10.1200/JCO.2006.09.2775.
- [41] A. A. Renshaw and E. W. Gould, “Reducing false-negative and false-positive diagnoses in anatomic pathology consultation material,” *Arch. Pathol. Lab. Med.*, vol. 137, no. 12, pp. 1770–1773, Dec. 2013, doi: 10.5858/ARPA.2013-0012-OA.
- [42] N. Segnan *et al.*, “Estimate of false-positive breast cancer diagnoses from accuracy studies: a systematic review,” *J. Clin. Pathol.*, vol. 70, no. 4, pp. 282–294, Apr. 2017, doi: 10.1136/jclinpath-2016-204184.
- [43] M. Gursoy, G. Sezgin, E. M. Horoz, B. Dirim Mete, and N. Erdogan, “Histopathological and Tumor Characteristics Associated with False Negative Axillary Ultrasonography Results in Breast Cancer,” *Med. Ultrason.*, vol. 21, no. 3, p. 232, Aug. 2019, doi: 10.11152/mu-1875.
- [44] “Why you should consider a second opinion from a pathologist.” <https://www.kevinmd.com/blog/2013/08/opinion-pathologist.html> (accessed Jan. 30, 2022).
- [45] H. Akbari *et al.*, “Hyperspectral imaging and quantitative analysis for prostate cancer detection,” <https://doi.org/10.1117/1.JBO.17.7.076005>, vol. 17, no. 7, p. 076005, Jul. 2012, doi: 10.1117/1.JBO.17.7.076005.
- [46] H. Akbari, K. Uto, Y. Kosugi, K. Kojima, and N. Tanaka, “Cancer detection using infrared hyperspectral imaging,” *Cancer Sci.*, vol. 102, no. 4, pp. 852–857, Apr. 2011, doi: 10.1111/J.1349-7006.2011.01849.X.
- [47] L. Hsien-Che, *Introduction to Color Imaging Science*. 2005.
- [48] “Seleccionar una lente para su cámara | Teledyne FLIR.” <https://www.flir.es/support-center/iis/machine-vision/application-note/selecting-a-lens-for-your-camera/> (accessed Mar. 09, 2022).
- [49] “Difference Between Real Image and Virtual Image.” <https://circuitglobe.com/difference-between-real-image-and-virtual-image.html> (accessed Mar. 09, 2022).
- [50] “Charge-Coupled Device - an overview | ScienceDirect Topics.” <https://www.sciencedirect.com/topics/engineering/charge-coupled-device/pdf> (accessed Mar. 10, 2022).
- [51] “Types Of Camera Sensor.” <https://www.photometrics.com/learn/camera-basics/types-of-camera-sensor> (accessed Mar. 10, 2022).
- [52] “The Smartphone Camera Sensor: A Simple Introduction | Smartphone Domain.” <https://smartphonedomain.com/smartphone-camera-sensor-introduction/> (accessed Mar. 11, 2022).
- [53] H. F. Grahn and P. Geladi, *Techniques and Applications of Hyperspectral Image Analysis*

*Techniques and Applications of Hyperspectral Image Analysis Edited. 2007.*

- [54] “What is hi-res? | The TechSmith Blog.” <https://www.techsmith.com/blog/what-is-hi-res/> (accessed Mar. 11, 2022).
- [55] R. Smith, “Tutorial: Introduction to Hyperspectral Imaging,” Accessed: Mar. 08, 2022. [Online]. Available: <http://www.microimages.com>.
- [56] “Spectrum Library Central at PeptideAtlas.” <http://www.peptideatlas.org/speclib/> (accessed Mar. 13, 2022).
- [57] B. Jansen-winkel *et al.*, “Feedforward Artificial Neural Network-Based Colorectal Cancer Detection Using Hyperspectral Imaging: A Step towards Automatic Optical Biopsy,” *Cancers 2021, Vol. 13, Page 967*, vol. 13, no. 5, p. 967, Feb. 2021, doi: 10.3390/CANCERS13050967.
- [58] A. Bykov *et al.*, “Hyperspectral imaging of human skin aided by artificial neural networks,” *Biomed. Opt. Express, Vol. 10, Issue 7, pp. 3545-3559*, vol. 10, no. 7, pp. 3545–3559, Jul. 2019, doi: 10.1364/BOE.10.003545.
- [59] “What’s the difference between artificial intelligence, machine learning, and natural language processing? | Sonix.” <https://sonix.ai/articles/difference-between-artificial-intelligence-machine-learning-and-natural-language-processing> (accessed Mar. 15, 2022).
- [60] “Reinforcement Learning: What is, Algorithms, Types & Examples.” <https://www.guru99.com/reinforcement-learning-tutorial.html#what-is-reinforcement-learning> (accessed Mar. 16, 2022).
- [61] “classification - Using Reinforcement Learning for Classification Problems - Stack Overflow.” <https://stackoverflow.com/questions/44594007/using-reinforcement-learning-for-classification-problems> (accessed Nov. 23, 2022).
- [62] M. A. Wiering, H. Van Hasselt, A.-D. Pietersma, L. Schomaker, and H. V. H. Nl, “Reinforcement Learning Algorithms for solving Classification Problems.”
- [63] M. G. Lagoudakis and R. Parr, “Reinforcement Learning as Classification: Leveraging Modern Classifiers.”
- [64] “What is Supervised Learning? | IBM.” <https://www.ibm.com/cloud/learn/supervised-learning> (accessed Mar. 17, 2022).
- [65] “Supervised Deep Learning Algorithms: Types and Applications.” <https://www.analyticsvidhya.com/blog/2021/05/introduction-to-supervised-deep-learning-algorithms/> (accessed Mar. 15, 2022).
- [66] “Neural Network Definition.” <https://www.investopedia.com/terms/n/neuralnetwork.asp> (accessed Mar. 17, 2022).
- [67] “Neural Network | Introduction to Neural Network | Neural Network for DL.” <https://www.analyticsvidhya.com/blog/2021/03/basics-of-neural-network/> (accessed Mar. 15, 2022).



- [68] “Weight (Artificial Neural Network) Definition | DeepAI.” <https://deepai.org/machine-learning-glossary-and-terms/weight-artificial-neural-network> (accessed Mar. 15, 2022).
- [69] “Basic Concepts for Neural Networks.” <https://www.cheshireeng.com/Neuralyst/nmbg.htm> (accessed Mar. 20, 2022).
- [70] “Four Common Types of Neural Network Layers | by Martin Isaksson | Towards Data Science.” <https://towardsdatascience.com/four-common-types-of-neural-network-layers-c0d3bb2a966c> (accessed Mar. 15, 2022).
- [71] “A Gentle Introduction to Pooling Layers for Convolutional Neural Networks.” <https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/> (accessed Mar. 20, 2022).
- [72] “Convolution -- from Wolfram MathWorld.” <https://mathworld.wolfram.com/Convolution.html> (accessed Mar. 20, 2022).
- [73] “Convolutional Layer - an overview | ScienceDirect Topics.” <https://www.sciencedirect.com/topics/engineering/convolutional-layer> (accessed Mar. 20, 2022).
- [74] “What is Pooling in a Convolutional Neural Network (CNN): Pooling Layers Explained - Programmathically.” <https://programmathically.com/what-is-pooling-in-a-convolutional-neural-network-cnn-pooling-layers-explained/> (accessed Sep. 12, 2022).
- [75] “Everything you need to know about ‘Activation Functions’ in Deep learning models | by Vandit Jain | Towards Data Science.” <https://towardsdatascience.com/everything-you-need-to-know-about-activation-functions-in-deep-learning-models-84ba9f82c253> (accessed Mar. 18, 2022).
- [76] “Difference between ANN, CNN and RNN - GeeksforGeeks.” <https://www.geeksforgeeks.org/difference-between-ann-cnn-and-rnn/> (accessed Mar. 22, 2022).
- [77] “CNN For Image Classification | Image Classification Using CNN.” <https://www.analyticsvidhya.com/blog/2021/01/image-classification-using-convolutional-neural-networks-a-step-by-step-guide/> (accessed Mar. 17, 2022).
- [78] “Google Online Security Blog: Are you a robot? Introducing ‘No CAPTCHA reCAPTCHA.’” <https://security.googleblog.com/2014/12/are-you-robot-introducing-no-captcha.html> (accessed Sep. 14, 2022).
- [79] “Captchas Have Us Deciphering Old Text Through Woozy Web Clues - The New York Times.” <https://www.nytimes.com/2011/03/29/science/29recaptcha.html> (accessed Sep. 14, 2022).
- [80] “Are You Unwittingly Helping to Train Google’s AI Models? | by Rugare Maruzani | Towards Data Science.” <https://towardsdatascience.com/are-you-unwittingly-helping-to-train-googles-ai-models-f318dea53aee> (accessed Sep. 14, 2022).

- [81] F. P. M. Oliveira and J. M. R. S. Tavares, “Medical image registration: a review,” <https://doi.org/10.1080/10255842.2012.670855>, vol. 17, no. 2, pp. 73–93, 2013, doi: 10.1080/10255842.2012.670855.
- [82] M. Sonka and J. M. Fitzpatrick, *Handbook of Medical Imaging Vol.2*, vol. 3, no. April. 2015.
- [83] J. B. A. Maintz and M. A. Viergever, “A survey of medical image registration,” *Med. Image Anal.*, vol. 2, no. 1, pp. 1–36, 1998, doi: 10.1016/S1361-8415(01)80026-8.
- [84] S. Nag, “Image Registration Techniques: A Survey,” Nov. 2017, doi: 10.17605/OSF.IO/RV65C.
- [85] L. G. Brown, “A survey of image registration techniques,” *ACM Comput. Surv.*, vol. 24, no. 4, pp. 325–376, Dec. 1992, doi: 10.1145/146370.146374.
- [86] “Similarity Transformations | Rotation, Reflection, & Translation (Video).” <https://tutors.com/math-tutors/geometry-help/similarity-transformations> (accessed Oct. 13, 2022).
- [87] B. Zitová and J. Flusser, “Image registration methods: A survey,” *Image Vis. Comput.*, vol. 21, no. 11, pp. 977–1000, Oct. 2003, doi: 10.1016/S0262-8856(03)00137-9.
- [88] J. A. Parker, R. V. Kenyon, and D. E. Troxel, “Comparison of Interpolating Methods for Image Resampling,” *IEEE Trans. Med. Imaging*, vol. 2, no. 1, pp. 31–39, 1983, doi: 10.1109/TMI.1983.4307610.
- [89] C. Liu, J. Xu, and F. Wang, “A Review of Keypoints’ Detection and Feature Description in Image Registration,” *Sci. Program.*, vol. 2021, 2021, doi: 10.1155/2021/8509164.
- [90] “Introduction to Harris Corner Detector | by Deepanshu Tyagi | Data Breach | Medium.” <https://medium.com/data-breach/introduction-to-harris-corner-detector-32a88850b3f6> (accessed Oct. 17, 2022).
- [91] “Introduction to FAST (Features from Accelerated Segment Test) | by Deepanshu Tyagi | Data Breach | Medium.” <https://medium.com/data-breach/introduction-to-fast-features-from-accelerated-segment-test-4ed33dde6d65> (accessed Oct. 17, 2022).
- [92] “Introduction to ORB (Oriented FAST and Rotated BRIEF) | by Deepanshu Tyagi | Data Breach | Medium.” <https://medium.com/data-breach/introduction-to-orb-oriented-fast-and-rotated-brief-4220e8ec40cf> (accessed Oct. 17, 2022).
- [93] “Introduction to SIFT (Scale-Invariant Feature Transform) — OpenCV-Python Tutorials 1 documentation.” [https://opencv24-python-tutorials.readthedocs.io/en/stable/py\\_tutorials/py\\_feature2d/py\\_sift\\_intro/py\\_sift\\_intro.html#](https://opencv24-python-tutorials.readthedocs.io/en/stable/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html#) (accessed Oct. 17, 2022).
- [94] “Introduction to SURF (Speeded-Up Robust Features) | by Deepanshu Tyagi | Data Breach | Medium.” <https://medium.com/data-breach/introduction-to-surf-speeded-up-robust-features-c7396d6e7c4e> (accessed Oct. 17, 2022).

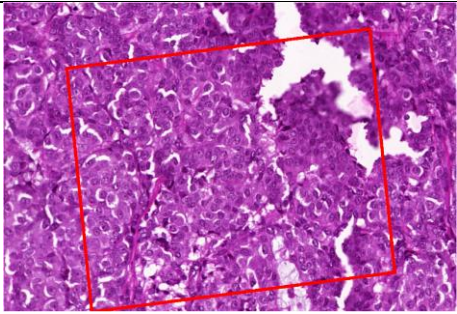
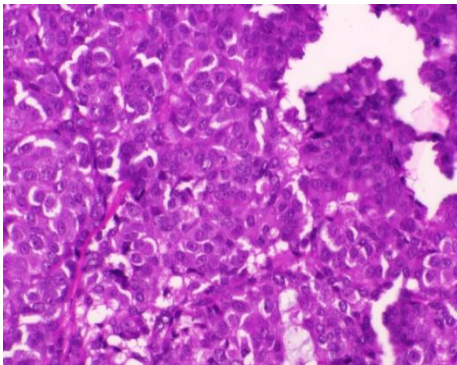
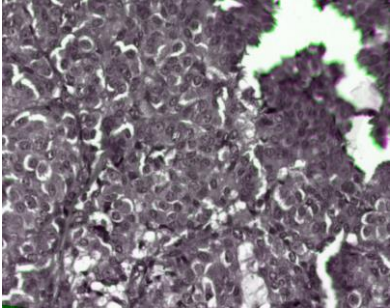
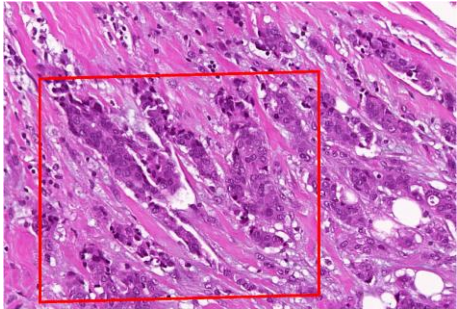
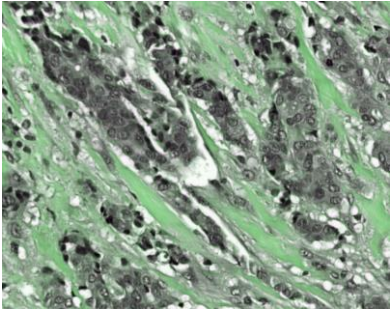
- [95] “Region detectors.”
- [96] K. Y. Kok and P. Rajendran, “A Descriptor-Based Advanced Feature Detector for Improved Visual Tracking,” *Symmetry* 2021, Vol. 13, Page 1337, vol. 13, no. 8, p. 1337, Jul. 2021, doi: 10.3390/SYM13081337.
- [97] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK: Binary Robust invariant scalable keypoints,” *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2548–2555, 2011, doi: 10.1109/ICCV.2011.6126542.
- [98] G. Song, J. Han, Y. Zhao, Z. Wang, and H. Du, “Medical Imaging Reviews BENTHAM SCIENCE,” doi: 10.2174/1573405612666160920123955.
- [99] M. Brown, “Suppose you want to create a panorama.”
- [100] S. Droste, T. Jansen, and I. Wegener, “On the analysis of the (1 + 1) evolutionary algorithm,” *Theor. Comput. Sci.*, vol. 276, no. 1–2, pp. 51–81, Apr. 2002, doi: 10.1016/S0304-3975(01)00182-7.
- [101] “Complete Step-by-Step Gradient Descent Algorithm from Scratch | by Albers Uzila | Towards Data Science.” <https://towardsdatascience.com/complete-step-by-step-gradient-descent-algorithm-from-scratch-acba013e8420> (accessed Oct. 17, 2022).
- [102] C. Rodríguez Rodríguez, “Control software for the acquisition of microscopic hyperspectral images,” ULPGC.
- [103] “INSTRUCTIONS BX53 SYSTEM MICROSCOPE.”
- [104] “Halógenas aplicación industrial Datos del producto Información general,” 2022, Accessed: Nov. 12, 2022. [Online]. Available: [www.lighting.philips.com](http://www.lighting.philips.com).
- [105] OLYMPUS, “TH4 HALOGEN LAMP POWER SUPPLY UNIT.”
- [106] “Micro Hyperspec Imaging Sensors ® FEATURES,” 2022, Accessed: Nov. 17, 2022. [Online]. Available: [www.headwall.be](http://www.headwall.be).
- [107] S. Ortega *et al.*, “Hyperspectral Imaging for the Detection of Glioblastoma Tumor Cells in H&E Slides Using Convolutional Neural Networks,” *Sensors* 2020, Vol. 20, Page 1911, vol. 20, no. 7, p. 1911, Mar. 2020, doi: 10.3390/S20071911.
- [108] “HYPER SPECTRAL BIO-MEDICAL IMAGING: UNFILTERED FLUOROPHORE IMAGING,” 2018.
- [109] “SCAN Series | Motorised Microscope Stage - Analytical Technologies Singapore.” <https://analyticaltechnologies.com.sg/products/accessories-supporting/xyz-microscope-stage/scan-series/> (accessed Nov. 17, 2022).
- [110] “Joysticks - Märzhäuser Sensotech.” <https://www.marzhauser-st.com/en/products/operating-devices/joysticks.html> (accessed Nov. 17, 2022).
- [111] “Pannoramic Viewer 1.15.3 User’s Guide Rev. 2 Pannoramic Viewer 1.15.3 User’s Guide,” 2013.
- [112] C. I. R. Fernandes *et al.*, “Dental Students’ Perceptions and Performance in Use of

- Conventional and Virtual Microscopy in Oral Pathology,” *J. Dent. Educ.*, vol. 82, no. 8, pp. 883–890, Aug. 2018, doi: 10.21815/JDE.018.084.
- [113] H. Y. Kuo *et al.*, “Habituation of steady-state visual evoked potentials in response to high-frequency polychromatic foveal visual stimulation,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 803–806, 2013, doi: 10.1109/EMBC.2013.6609622.
- [114] D. Muthukumar and M. Sivakumar, “Medical Image Registration: A Matlab Based Approach,” *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.* © 2017 IJSRCSEIT, vol. 1, no. 2, pp. 2456–3307, 2017, Accessed: Dec. 01, 2022. [Online]. Available: [www.ijsrcseit.com](http://www.ijsrcseit.com).
- [115] T. Duy Linh and H. Quang Linh, “MEDICAL IMAGE REGISTRATION IN MATLAB.”
- [116] E. Irmak, E. Erçelebi, and A. Hanifi Ertaş, “Brain tumor detection using monomodal intensity based medical image registration and MATLAB,” doi: 10.3906/elk-1403-75.
- [117] “Physics 116A Three-Dimensional Proper and Improper Rotation Matrices 1. Proper and improper rotation matrices,” 2011.
- [118] Itu-r, “Recommendation ITU-R BT.601-7 Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios BT Series Broadcasting service (television),” 2017, Accessed: Dec. 01, 2022. [Online]. Available: <http://www.itu.int/ITU-R/go/patents/en>.

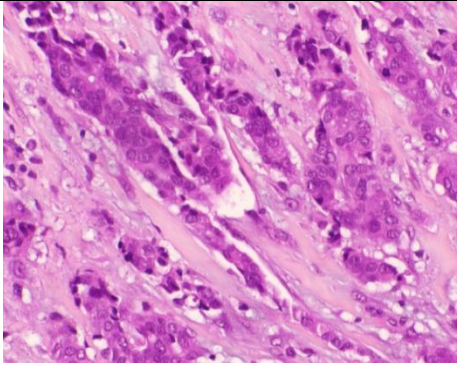
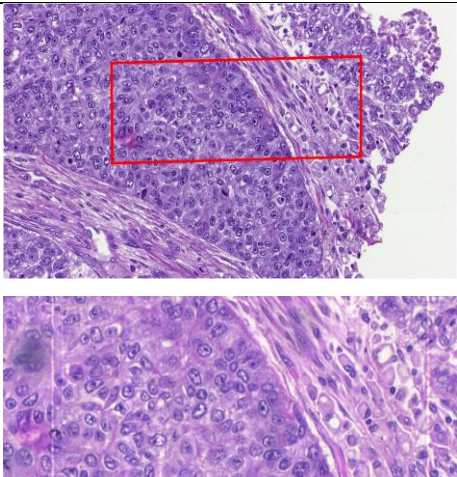
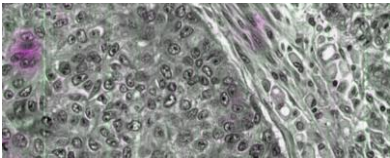
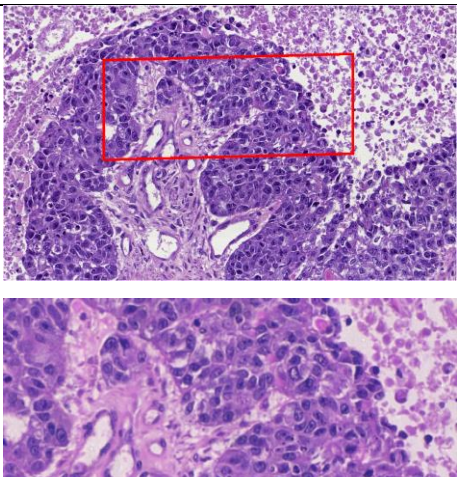
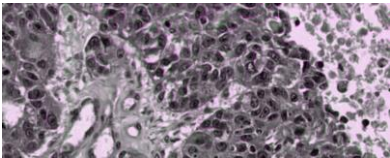
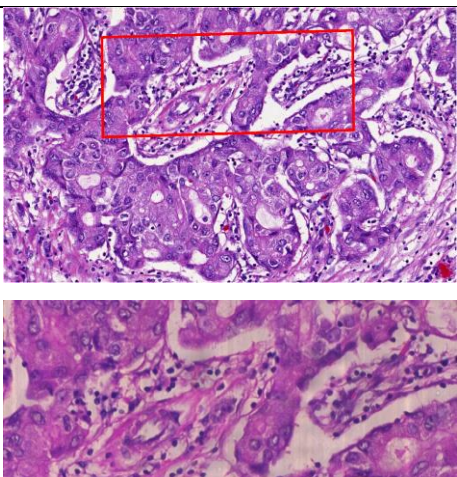
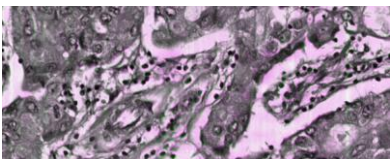
## 7 Annexes

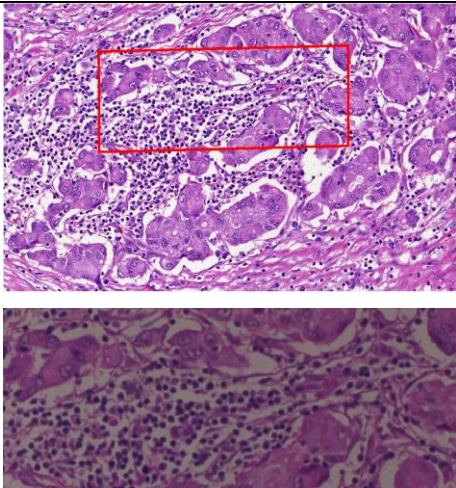
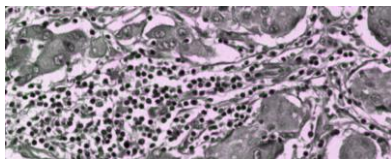
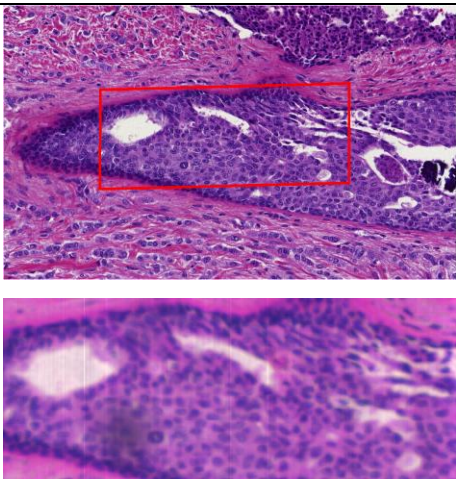
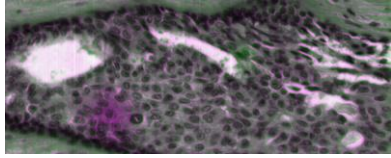
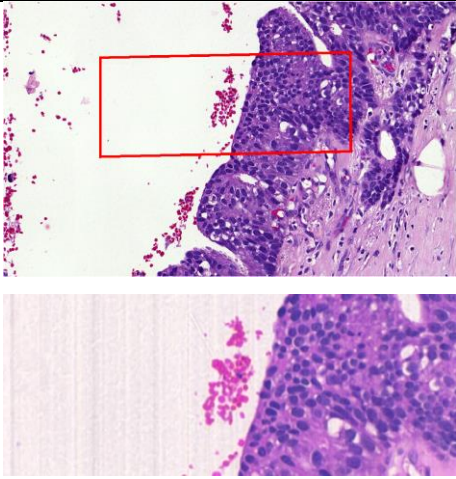
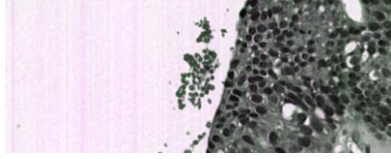
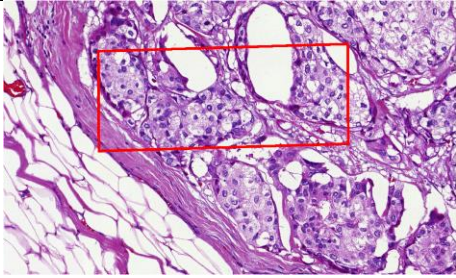
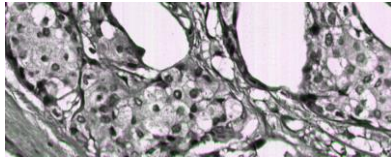
### 7.1 Dataset example and registration results

In Table 6 an example of a not annotated RGB, annotated RGB, and synthetic RGB images from the dataset of each patient is illustrated, with the result from the registration using the SURF detector applied to the grayscale images. The result is illustrated by overlaying the synthetic RGB on top of the RGB slide registered, where magenta and green regions show where the intensities are different. A perfect registration should be a grayscale image.

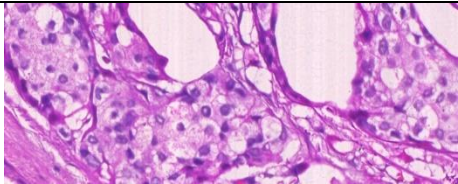
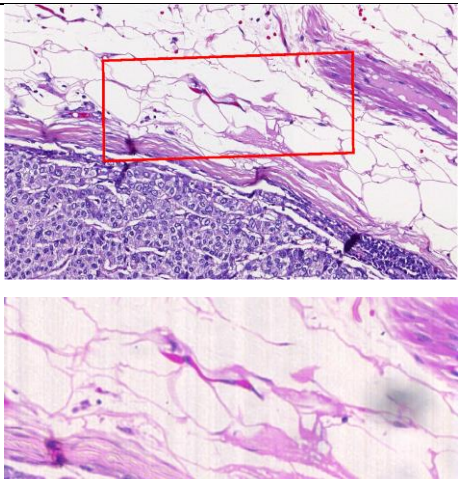
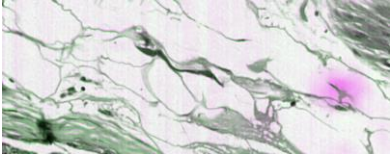
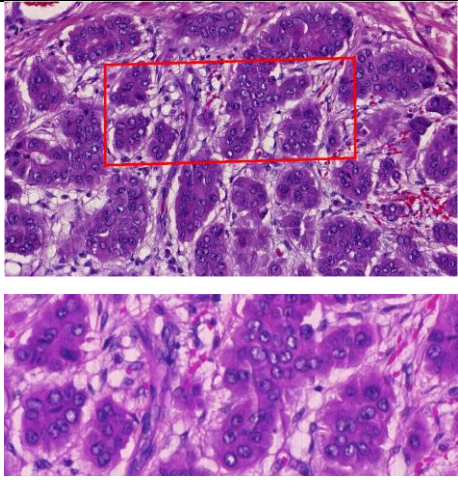
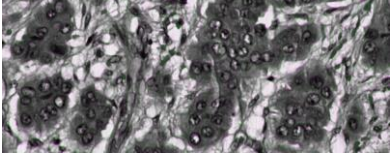
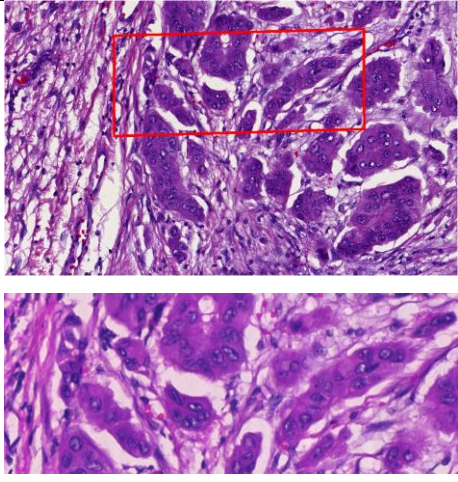
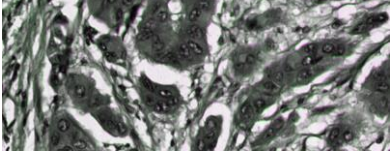
	Images to be registered	Registration result
Patient 1	 	
Patient 2		



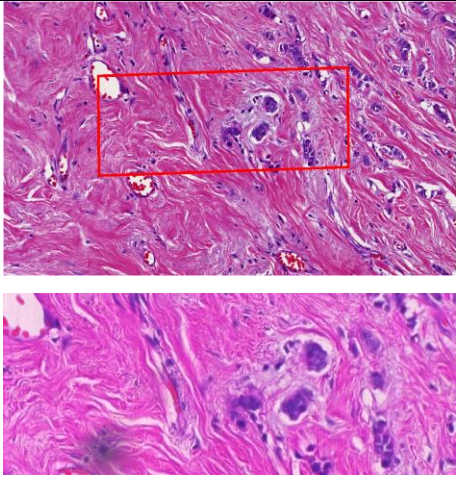
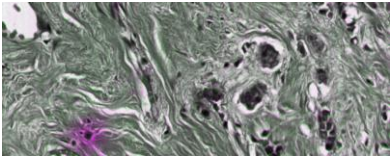
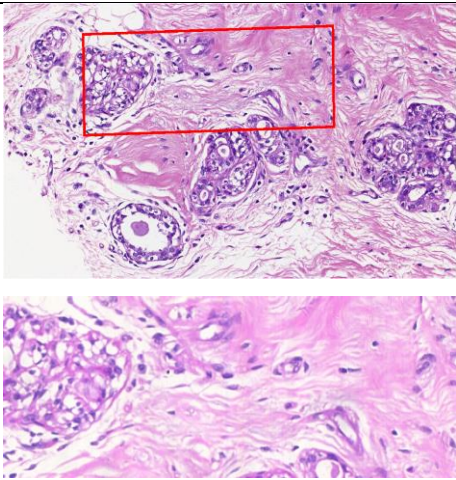
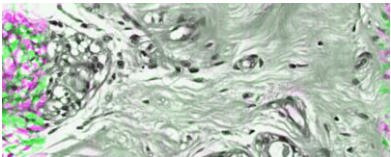
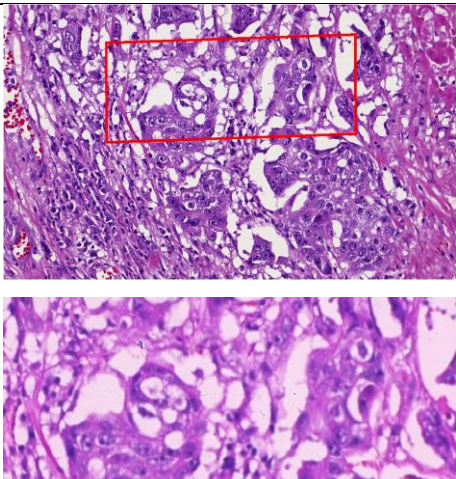
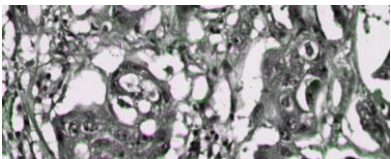
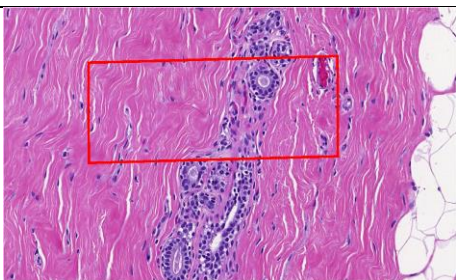
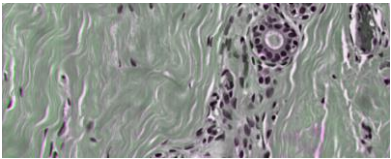
		
Patient 3		
Patient 4		
Patient 5		

<p>Patient 6</p>		
<p>Patient 7</p>		
<p>Patient 8</p>		
<p>Patient 9</p>		



		
<p>Patient 10</p>		
<p>Patient 11</p>		
<p>Patient 12</p>		



Patient 13		
Patient 14		
Patient 15		
Patient 16		

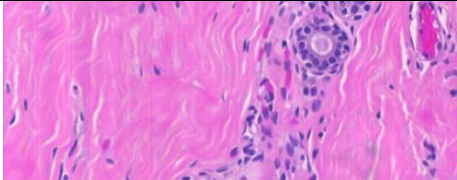
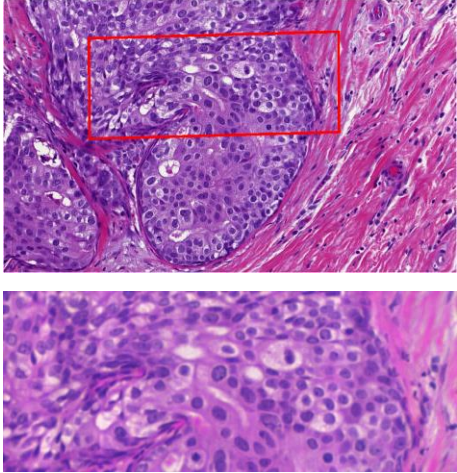
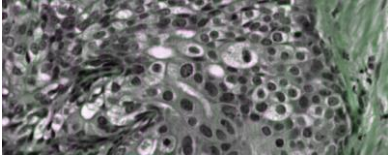
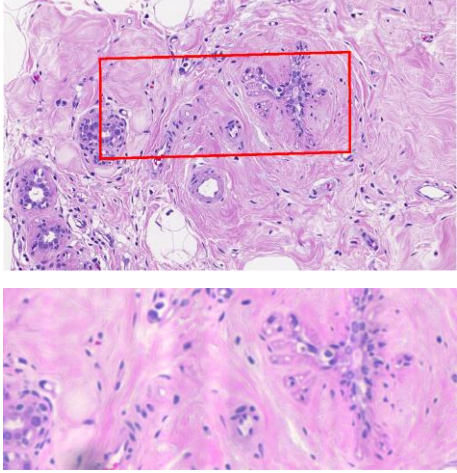
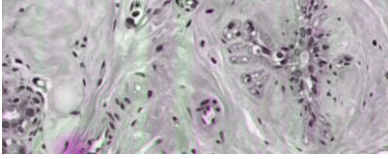
		
Patient 17		
Patient 18		

Table 6 Examples of the dataset and its registration result.

## 7.2 Developed code

This section shows every piece of code developed in *Matlab 2022a* following the same order as shown in chapter 3, Section 3.2.3 Registration process and evaluation. Firstly, the functions developed in the context of this project to perform the point-based and area-based registration will be shown. Secondly, the three programs that implement the method validation of the previous functions and obtain the metrics (SSIM, MSE and execution time) required to perform the evaluation of each registration method. Finally, the functions developed to perform the ANOVA tests and to represent the results are displayed.

### 7.2.1 FeatureRegistration.m

```
function [tform,Ir]=FeatureRegistration(moved,fixed,varargin)
%[tform,Ir] = FeatureRegistration(moved,fixed,varargin)
%
%This function registers the moved image to the fixed template and returns
%the transformation (tform) and the moving image registered (Ir).
%
%Optional parameters: detector, tfType, showFeatures, showMatches, showRegister
%
%detector (default = "SURF"): string that defines the detector used to detect features.
%
%showFeatures (default = true): logical to show the MSER features extracted of each image.
%
%showMatches (default = true): logical to show the matched points.
%
%tfType (default = "affine"): string that defines the type of transformation.
%
%showRegister (default = true): logical to show the register result.

%Set-up of the default parameters
detector = "SURF";
showFeatures = true;
showMatches = true;
tfType = "affine";
showRegister = true;
detectors={'MSER','MinEigen','Harris','FAST','ORB','SURF','KAZE','BRISK','SIFT'};
tfTypes={'rigid','similarity','affine','projective'};

%Modify parameters to the specified ones
try
    if ~isempty(varargin)
        try
            for i=1:2:size(varargin,2)
                switch varargin{i}
                    case "detector"
                        if ischar(varargin{i+1}) || isstring(varargin{i+1})
                            detector = string(varargin{i+1});
                        else
                            warning("The value of the parameter " + varargin{i} + " is not valid. It will be
ignored.");
                        end
                    case "showFeatures"
                        if islogical(varargin{i+1})
                            showFeatures = varargin{i+1};
                        else
                            warning("The value of the parameter " + varargin{i} + " is not valid. It will be
ignored.");
                        end
                    case "showMatches"
                        if islogical(varargin{i+1})
                            showMatches = varargin{i+1};
                        else
                            warning("The value of the parameter " + varargin{i} + " is not valid. It will be
ignored.");
                        end
                    case "tfType"
                        if ischar(varargin{i+1}) || isstring(varargin{i+1})
                            tfType = string(varargin{i+1});
                        else
                            warning("The value of the parameter " + varargin{i} + " is not valid. It will be
ignored.");
                        end
                    case "showRegister"

```

```

        if islogical(varargin{i+1})
            showRegister = varargin{i+1};
        else
            warning("The value of the parameter " + varargin{i} + " is not valid. It will be
ignored.");
        end
    otherwise
        warning("The parameter " + varargin{i} + " is not valid. It will be ignored.");
    end
end
catch
    error("Check if every parameter has a value assigned.")
end
end

%Transform the images from a file path/rgb image to a gray scale image
switch class(moved)
    case 'string'
        moved = rgb2gray(imread(moved));
    case 'uint8'
        if ndims(moved)==3
            moved = rgb2gray(moved);
        end
    otherwise
        error("The moving image does not have a useful format.")
end

switch class(fixed)
    case 'string'
        fixed = rgb2gray(imread(fixed));
    case 'uint8'
        if ndims(fixed)==3
            fixed = rgb2gray(fixed);
        end
    otherwise
        error("The fixed image does not have a useful format.")
end

%Detect de features using the detector specified
if isempty(find(detectors==detector, 1))
    error("The detector selected is not supported");
end

switch detector
    case "MSER"
        reg1=detectMSERFeatures(moved);
        reg2=detectMSERFeatures(fixed);
    case "MinEigen"
        reg1=detectMinEigenFeatures(moved);
        reg2=detectMinEigenFeatures(fixed);
    case "Harris"
        reg1=detectHarrisFeatures(moved);
        reg2=detectHarrisFeatures(fixed);
    case "FAST"
        reg1=detectFASTFeatures(moved);
        reg2=detectFASTFeatures(fixed);
    case "ORB"
        reg1=detectORBFeatures(moved);
        reg2=detectORBFeatures(fixed);
    case "SURF"
        reg1=detectSURFFeatures(moved);
        reg2=detectSURFFeatures(fixed);
    case "KAZE"
        reg1=detectKAZEFeatures(moved);
        reg2=detectKAZEFeatures(fixed);
    case "BRISK"
        reg1=detectBRISKFeatures(moved);
        reg2=detectBRISKFeatures(fixed);
    case "SIFT"
        reg1=detectSIFTFeatures(moved);
        reg2=detectSIFTFeatures(fixed);
    otherwise
        warning("The detector selected ( " + detector + ") to detect features is not supported. Default
detector (SURF) will be used.")
        reg1=detectSURFFeatures(moved);
        reg2=detectSURFFeatures(fixed);
end

%Show the features detected
if showFeatures
    figure; imshow(moved); hold on; plot(reg1); title(detector + " features")
    figure; imshow(fixed); hold on; plot(reg2); title(detector + " features")
end

%Extract the features descriptors and their properties

```



```
[f1, val1]=extractFeatures(moved,reg1);
[f2, val2]=extractFeatures(fixed,reg2);

%Match the features from each image
indexPairs = matchFeatures(f1,f2);

%Pair the matched points
match1=val1(indexPairs(:,1),:);
match2=val2(indexPairs(:,2),:);

%Show the matches
if showMatches; figure; showMatchedFeatures(moved,fixed,match1,match2,"montage"); title(detector + "
matches"); end

%Check if the specified transformation is valid
if isempty(find(tfTypes==tfType, 1))
    error("The tfType selected is not supported");
end

%Calculate the transformation matrix
tform = estimateGeometricTransform2D(match1,match2,tfType);

%Apply the transformation using the coordinated system of the fixed image
Ir=imwarp(moved,tform,'OutputView',imref2d(size(fixed)));

%Show the registration
if showRegister; figure; imshowpair(fixed,Ir,"checkerboard"); title(detector + " registration");end

catch e
    error(e.message)
end
```

Code Fragment 20 FeatureRegistration.m.

## 7.2.2 IntensityRegistration.m

```
function [tform,Ir]=IntensityRegistration(moved,fixed,varargin)
%[tform,Ir] = IntensityRegistration(moved,fixed,varargin)
%
%This function registers the moved image to the fixed template and returns
%the transformation (tform) and the moving image registered (Ir).
%
%Optional parameters: modality, tfType, showRegister
%
%optimizer (default = "1+1 EA"): string that defines the optimizer to use on
%the iterative process.
%
%metric (default = "MMI"): string that defines the metric to use on the
%iterative process.
%
%tfType (default = "affine"): string that defines the type of transformation.
%
%showRegister (default = true): logical to show the register result.

tfType = "affine";
%modality = "multimodal";
metric = "MMI";
optimizer = "1+1 EA";
showRegister=true;
try
    if ~isempty(varargin)
        try
            for i=1:2:size(varargin,2)
                switch varargin{i}
                    case "tfType"
                        if ischar(varargin{i+1}) || isstring(varargin{i+1})
                            tfType = varargin{i+1};
                        else
                            warning("The value of the parameter " + varargin{i} + " is not valid. It will be
ignored.");
                        end
                    case "metric"
                        if ischar(varargin{i+1}) || isstring(varargin{i+1})
                            metric = varargin{i+1};
                        else
                            warning("The value of the parameter " + varargin{i} + " is not valid. It will be
ignored.");
                        end
                    case "optimizer"
                        if ischar(varargin{i+1}) || isstring(varargin{i+1})
                            optimizer = varargin{i+1};
                        else
                            warning("The value of the parameter " + varargin{i} + " is not valid. It will be
ignored.");
                        end
                end
            end
        end
    end
end
```

```

        case "showRegister"
            if islogical(varargin{i+1})
                showRegister = varargin{i+1};
            else
                warning("The value of the parameter " + varargin{i} + " is not valid. It will be
ignored.");
            end
        otherwise
            warning("The parameter " + varargin{i} + " is not valid. It will be ignored.");
        end
    end
catch
    error("Check if every parameter has a value assigned.")
end
end

switch class(moved)
case 'string'
    try
        moved = rgb2gray(imread(moved));
    catch e
        error(e.message)
    end
case 'uint8'
    if ndims(moved)==3
        try
            moved = rgb2gray(moved);
        catch e
            error(e.message)
        end
    end
otherwise
    error("The moving image does not have a useful format.")
end

switch class(fixed)
case 'string'
    try
        fixed = rgb2gray(imread(fixed));
    catch e
        error(e.message)
    end
case 'uint8'
    if ndims(fixed)==3
        try
            fixed = rgb2gray(fixed);
        catch e
            error(e.message)
        end
    end
otherwise
    error("The fixed image does not have a useful format.")
end

switch optimizer
case "1+1 EA"
    optimizer=registration.optimizer.OnePlusOneEvolutionary;
case "RSGD"
    optimizer=registration.optimizer.RegularStepGradientDescent;
otherwise
    warning("The optimizer selected is not supported. The default value will be used.")
    optimizer=registration.optimizer.OnePlusOneEvolutionary;
end

switch metric
case "MMI"
    metric=registration.metric.MattesMutualInformation;
case "MSE"
    metric=registration.metric.MeanSquares;
otherwise
    warning("The optimizer selected is not supported. The default value will be used.")
    metric=registration.metric.MattesMutualInformation;
end

tformEstimate = imregcorr(moved,fixed,"similarity");
tform = imregtform(moved,fixed,tftype,optimizer,metric,"InitialTransformation",tformEstimate);

Ir=imwarp(moved,tform,'OutputView',imref2d(size(fixed)));

if showRegister
    figure; imshowpair(fixed,Ir,"montage"); title("Registration");
end
catch e
    error(e.message)
end

```

end

Code Fragment 21 IntensityRegistration.m.

### 7.2.3 MethodValidation.m

```

clear all
clc

rng('default');
seed=rng; %Save seed for reproducibility

hsFolderDir = "TortosaCompleted\00_PushbroomRGB\";
slideFolderDir = "TortosaCompleted\00_Slides\";
imageList={dir(hsFolderDir).name};
imageList=imageList(3:end);

phi=[-10 -10]; %Aproximate rotation (in grad) in the dataset
scale=[0.59 0.59]; %Aproximate of the scale difference in the dataset
xtranslation=[200 200];
ytranslation=[50 50];

knownTform = randomAffine2d("Rotation",phi,"Scale",scale,"XTranslation",xtranslation,"YTranslation",ytranslation);

invKnownTform=invert(knownTform);
spectedXScale=norm(invKnownTform.T(1:2,1));
spectedYScale=norm(invKnownTform.T(1:2,2));
spectedRot=acos(invKnownTform.T(1,1)/spectedXScale);
spectedXtranslation=invKnownTform.T(3,1);
spectedYtranslation=invKnownTform.T(3,2);

methods={'MSER','MinEigen','Harris','FAST','ORB','SURF','KAZE','BRISK','SIFT',{'1+1 EA','MMI'},{'RSGD','MMI'},{'1+1 EA','MSE'},{'RSGD','MSE'}};
unknownTform={};
error=[];
numImg=10;
flag=true;

for j=1:numImg
    disp(j+"/"+numImg);

    folder=randi(2,1); %Select random folder
    if folder==1; folder=hsFolderDir;
    else folder=slideFolderDir;
    end

    I=imread(strcat(folder,imageList(randi(length(imageList),1)))); %Read image

    Ir=imwarp(I,knownTform); %Apply known transformation

    for i=1:length(methods)
        disp("i = "+i+"/"+length(methods));
        try
            if isa(methods{i},"char")
                [unknownTform{j,i},newI] =
                FeatureRegistration(Ir,I,"showFeatures",false,"showMatches",false,"showRegister",false,"tfType","similarity","detector",methods{i}); %Feature Registration
            else
                [unknownTform{j,i},newI] =
                IntensityRegistration(Ir,I,"optimizer",methods{i}{1},"metric",methods{i}{2},"showRegister",false,"tfType","similarity"); %Intensity Registration
            end
        catch e %If something wrong:
            warning(e);
            mseVs(j,i)=inf;
            ssimVs(j,i)=0;
            unknownTform{j,i}=affine2d();
            flag=false;
        end

        error(j,i) = immse(invKnownTform.T,double(unknownTform{j,i}.T));

        newXScale=norm(unknownTform{j,i}.T(1:2,1));
        newYScale=norm(unknownTform{j,i}.T(1:2,2));
        newRot=acos(unknownTform{j,i}.T(1,1)/newXScale);
        newXtranslation=unknownTform{j,i}.T(3,1);
        newYtranslation=unknownTform{j,i}.T(3,2);

        xScaleError(j,i)=abs(spectedXScale-newXScale);
        yScaleError(j,i)=abs(spectedYScale-newYScale);
        rotError(j,i)=abs(spectedRot-newRot);
        xTranslationError(j,i)=abs(spectedXtranslation-newXtranslation);
        yTranslationError(j,i)=abs(spectedYtranslation-newYtranslation);

    end
end
if flag

```

```

        mseVs(j,i)=immse(newI,rgb2gray(I));
        ssimVs(j,i)=ssim(newI,rgb2gray(I));
    end
    flag=true;
end
end

results =
struct("xScaleError",xScaleError,"yScaleError",yScaleError,"rotError",rotError,"xTranslationError",xTranslationError,"yTranslationError",yTranslationError,"MSE",mseVs,"SSIM",ssimVs);

save("MethodValidation","error","invKnownTform","unknownTform","results");

```

Code Fragment 22 MethodValidation.m.

## 7.2.4 Similarity\_Metric\_Feature\_Registration.m

```

clear all

rng('default');
seed=rng; %Save seed for reproducibility

hsFolderDir = "00_PushbroomRGB\";
slideFolderDir = "00_Slides\";

imageList={dir(hsFolderDir).name};
imageList=imageList(3:end);

for i=1:length(imageList)
    fixed=rgb2gray(imread(strcat(hsFolderDir,imageList(i))));
    moved=rgb2gray(imread(strcat(slideFolderDir,imageList(i))));
    disp(i+"/"+length(imageList));

    invTm=zeros(1,9);
    ssimV=zeros(1,9);
    invMseV=zeros(1,9);
    numIt=10;
    for k=1:numIt
        %regs={};
        tic;
        regs{1}={detectMSERFeatures(moved),detectMSERFeatures(fixed)};
        invTm(1)=invTm(1)+(toc^-1);%time(i,1)=toc;
        tic;
        regs{2}={detectMinEigenFeatures(moved),detectMinEigenFeatures(fixed)};
        invTm(2)=invTm(2)+(toc^-1);%time(i,2)=toc;
        tic;
        regs{3}={detectHarrisFeatures(moved),detectHarrisFeatures(fixed)};
        invTm(3)=invTm(3)+(toc^-1);%time(i,3)=toc;
        tic;
        regs{4}={detectFASTFeatures(moved),detectFASTFeatures(fixed)};
        invTm(4)=invTm(4)+(toc^-1);%time(i,4)=toc;
        tic;
        regs{5}={detectORBFeatures(moved),detectORBFeatures(fixed)};
        invTm(5)=invTm(5)+(toc^-1);%time(i,5)=toc;
        tic;
        regs{6}={detectSURFFeatures(moved),detectSURFFeatures(fixed)};
        invTm(6)=invTm(6)+(toc^-1);%time(i,6)=toc;
        tic;
        regs{7}={detectKAZEFeatures(moved),detectKAZEFeatures(fixed)};
        invTm(7)=invTm(7)+(toc^-1);%time(i,7)=toc;
        tic;
        regs{8}={detectBRISKFeatures(moved),detectBRISKFeatures(fixed)};
        invTm(8)=invTm(8)+(toc^-1);%time(i,8)=toc;
        tic;
        regs{9}={detectSIFTFeatures(moved),detectSIFTFeatures(fixed)};
        invTm(9)=invTm(9)+(toc^-1);%time(i,9)=toc;

        for j=1:length(regs)
            [f,val]=extractFeatures(moved,regs{j}{1});
            f_val{j}={f,val};
            [f,val]=extractFeatures(fixed,regs{j}{2});
            f_val{j}={f_val{j},{f,val}};

            indexPairs{j}=matchFeatures(f_val{j}{1}{1},f_val{j}{2}{1});
            matchs{j}={f_val{j}{1}{2}(indexPairs{j}(:,1),:),f_val{j}{2}{2}(indexPairs{j}(:,2),:)};

            [tform,~,status]=estimateGeometricTransform2D(matchs{j}{1},matchs{j}{2},"affine");

            if status ~= 0
                warning("Status = "+status);
                continue;
            end

            try
                Ir=imwarp(moved,tform,'OutputView',imref2d(size(fixed)));
            end
        end
    end
end

```



```

        catch
            continue;
        end

        invMseV(j)=invMseV(j)+(immse(Ir,fixed))^-1;
        ssimV(j)=ssimV(j)+ssim(Ir,fixed);

    end
end
ssimVals(i,:)=ssimV./numIt;
mseVals(i,:)=(invMseV./numIt).^-1;
time(i,:)=(invTm./numIt).^-1;
end

zeroValues=mseVals==0;
mseVals(zeroValues)=inf;

save("Feature_metric_results_X10",'ssimVals','mseVals','time','seed')

```

Code Fragment 23 Similarity\_Metric\_Feature\_Registration.m.

## 7.2.5 Similarity\_Metric\_Intensity\_Registration.m

```

clear all

rng('default');
seed=rng; %Save seed for reproducibility

hsFolderDir = "00_PushbroomRGB\";
slideFolderDir = "00_Slides\";

imageList={dir(hsFolderDir).name};
imageList=imageList(3:end);

numIt=10;

for i=1:length(imageList)
    fixed=rgb2gray(imread(strcat(hsFolderDir,imageList(i))));
    moved=rgb2gray(imread(strcat(slideFolderDir,imageList(i))));
    disp(i+"/"+length(imageList));

    for j=1:2
        optimizer{1}=registration.optimizer.OnePlusOneEvolutionary;
        optimizer{2}=registration.optimizer.RegularStepGradientDescent;

        metric{1}=registration.metric.MattesMutualInformation;
        metric{2}=registration.metric.MeanSquares;

        for u=1:2
            if j==1; aux=u; else; aux=j+u; end
            invTm=0;
            ssimV=0;
            invMseV=0;
            for k=1:numIt
                try
                    tic;
                    tform =
                    imregtform(moved,fixed,"affine",optimizer{j},metric{u});%,"InitialTransformation",tformEstimate);

                    if lastwarn == ""
                        Ir=imwarp(moved,tform,'OutputView',imref2d(size(fixed)));
                    else
                        lastwarn('');
                        continue;
                    end
                    invTm=invTm+(toc^-1);
                    invMseV=invMseV+(immse(Ir,fixed))^-1;
                    ssimV=ssimV+ssim(Ir,fixed);

                    catch e
                        warning(e.message);
                        continue;
                    end
                end
                time(i,aux)=(invTm/numIt)^-1;
                ssimVals(i,aux)=ssimV/numIt;
                mseVals(i,aux)=(invMseV/numIt)^-1;
            end
        end
    end

end

save("Intensity_metric_results_GaussianFilter",'time','ssimVals','mseVals','seed')

```

Code Fragment 24 Similarity\_Metric\_Intensity\_Registration.m.

## 7.2.6 GenerateBoxChart.m

```
function GenerateBoxChart(groups,data,varargin)
% groups is a list of each type of data
% data is a matrix which each column is represented by the same box

figure
for i=groups
    k={};
    for j=1:size(data,1)
        k(end+1)=i;
    end
    H = boxchart(categorical(k),data(:,groups==string(i)));
    hold on
end

title(varargin{1},'FontName','Times New Roman')
H.Parent.FontName='Times New Roman';
```

Code Fragment 25 GenerateBoxChart.m.

## 7.2.7 PlotNormDistribution.m

```
function PlotNormDistribution(data,groups,xlimits,ymax,varargin)
% data is a matrix where each column is a group
% labels is an array with the names of each group

figure
for i=1:size(data,2)
    norm=fitdist(data(:,i),'Normal');
    y_norm = normpdf(xlimits,norm.mu,norm.sigma);
    plot(xlimits,y_norm)
    hold on
end
legend(groups,FontName='Times New Roman')
title(varargin{2},'FontName','Times New Roman')

end
```

Code Fragment 26 PlotNormDistribution.m.

## 7.2.8 GenerateAnovaPValueTable.m

```
function [pTable,h]=GenerateAnovaPValueTable(data,groups,varargin)
figure
[~,~,stats] = anova1(data,{'','off'});
[comparison,means,H,~] = multcompare(stats,'Display','on');
H.Children.Title.String=varargin{1};
H.Children.YTickLabel=flipplr(groups);
H.Children.XMinorGrid = 'on';
H.Children.XLabel.String="";
for i = 1:(length(H.Children.Children)-1)
    H.Children.Children(i).Color=[1 0 0];
end
H.Children.Children(end).Color=[1 1 1];
H.Children.FontName='Times New Roman';

pTable=eye(size(data,2));
for i = comparison
    pTable(i(2),i(1))=i(end);
end

isupper = logical(triu(ones(size(pTable)),1));
pTable(isupper) = NaN;

figure
h = heatmap(pTable,'GridVisible','off','MissingDataColor','w');
h.XDisplayLabels = groups;
h.YDisplayLabels = groups;
h.CellLabelFormat = '%.3f';
h.Title=varargin{2};
cMap = interp1([0;0.05;1],[0.6784 0.8471 0.9020; 0.1961 0.8039 0.1961; 0.4235 0.2314 0.1647],linspace(0,1,256));
h.Colormap=cMap;
h.FontName='Times New Roman';
end
```

Code Fragment 27 GenerateAnovaPValueTable.m.

## 7.2.9 VisualizeResultsANOVA.m

```
load('ResultsX10\Intensity_metric_results_X10.mat') % Local path to the intensity results
```

```

GenerateBoxChart({'1+1 EA / MMI','1+1 EA / MSE', 'RSGD / MMI', 'RSGD / MSE'},ssimVals,"SSIM Intensity Registration")
GenerateBoxChart({'1+1 EA / MMI','1+1 EA / MSE', 'RSGD / MMI', 'RSGD / MSE'},1./mseVals,"MSE^-1 Intensity Registration")
GenerateBoxChart({'1+1 EA / MMI','1+1 EA / MSE', 'RSGD / MMI', 'RSGD / MSE'},1./time,"Time^-1 Intensity Registration")
hold on
ax = gca;
ax.YAxis.Label.String = 'Seconds^-1';

GenerateAnovaPValueTable(ssimVals,{'1+1 EA / MMI','1+1 EA / MSE', 'RSGD / MMI', 'RSGD / MSE'},"Multicomparison of SSIM Intensity Registration Means","ANOVA P-Value SSIM Intensity Registration");
GenerateAnovaPValueTable(1./mseVals,{'1+1 EA / MMI','1+1 EA / MSE', 'RSGD / MMI', 'RSGD / MSE'},"Multicomparison of MSE^-1 Intensity Registration Means","ANOVA P-Value MSE^-1 Intensity Registration");
GenerateAnovaPValueTable(1./time,{'1+1 EA / MMI','1+1 EA / MSE', 'RSGD / MMI', 'RSGD / MSE'},"Multicomparison of Time^-1 Intensity Registration Means","ANOVA P-Value Time^-1 Intensity Registration");

PlotNormDistribution(ssimVals,{'1+1 EA / MMI','1+1 EA / MSE', 'RSGD / MMI', 'RSGD / MSE'},linspace(-1, 1, 1000),4.5,"2 by 2 SSIM Intensity Normal Distribution Comparison","SSIM Intensity Normal Distribution Comparison");
PlotNormDistribution(1./mseVals,{'1+1 EA / MMI','1+1 EA / MSE', 'RSGD / MMI', 'RSGD / MSE'},linspace(-0.002, 0.002, 1000),3000,"2 by 2 MSE^-1 Intensity Normal Distribution Comparison","MSE^-1 Intensity Normal Distribution Comparison");
PlotNormDistribution(1./time,{'1+1 EA / MMI','1+1 EA / MSE', 'RSGD / MMI', 'RSGD / MSE'},linspace(-1, 1, 1000),25,"2 by 2 Time^-1 Intensity Normal Distribution Comparison","Time^-1 Intensity Normal Distribution Comparison");

load('ResultsX10\Feature_metric_results_X10.mat') % Local path to the feature results

GenerateBoxChart({'MSER','MinEigen','Harris','FAST','ORB','SURF','KAZE','BRISK','SIFT'},ssimVals,"SSIM Feature Registration")
GenerateBoxChart({'MSER','MinEigen','Harris','FAST','ORB','SURF','KAZE','BRISK','SIFT'},1./mseVals,"MSE^-1 Feature Registration")
GenerateBoxChart({'MSER','MinEigen','Harris','FAST','ORB','SURF','KAZE','BRISK','SIFT'},1./time,"Time^-1 Feature Registration")
hold on
ax = gca;
ax.YAxis.Label.String = 'Seconds^-1';

GenerateAnovaPValueTable(ssimVals,{'MSER','MinEigen','Harris','FAST','ORB','SURF','KAZE','BRISK','SIFT'},"Multicomparison of SSIM Feature Registration Means","Anova P-Value SSIM Feature Registration");
GenerateAnovaPValueTable(1./mseVals,{'MSER','MinEigen','Harris','FAST','ORB','SURF','KAZE','BRISK','SIFT'},"Multicomparison of MSE^-1 Feature Registration Means","Anova P-Value MSE^-1 Feature Registration");
GenerateAnovaPValueTable(1./time,{'MSER','MinEigen','Harris','FAST','ORB','SURF','KAZE','BRISK','SIFT'},"Multicomparison of Time^-1 Feature Registration Means","Anova P-Value Time^-1 Feature Registration");

PlotNormDistribution(ssimVals,{'MSER','MinEigen','Harris','FAST','ORB','SURF','KAZE','BRISK','SIFT'},linspace(-2, 2, 1000),3,"2 by 2 SSIM Feature Normal Distribution Comparison","SSIM Feature Normal Distribution Comparison");
PlotNormDistribution(1./mseVals,{'MSER','MinEigen','Harris','FAST','ORB','SURF','KAZE','BRISK','SIFT'},linspace(-0.005, 0.005, 1000),2000,"2 by 2 MSE^-1 Feature Normal Distribution Comparison","MSE^-1 Feature Normal Distribution Comparison");
PlotNormDistribution(1./time,{'MSER','MinEigen','Harris','FAST','ORB','SURF','KAZE','BRISK','SIFT'},linspace(-1, 600, 1000),25,"2 by 2 Time^-1 Feature Normal Distribution Comparison","Time^-1 Feature Normal Distribution Comparison");

```

Code Fragment 28 VisualizeResultsANOVA.m.

### 7.2.10 ExtractAnnotations.m

```

function [tags,mask]=ExtractAnnotations(I,varargin)

colours=[];

switch class(I)
    case 'string'
        I = imread(I);
    case 'uint8'
        if ndims(moved)~=3
            error("Not RGB images are not supported.")
        end
    otherwise
        error("The moving image does not have a useful format.")
end

tags=zeros(size(I));
mask=zeros(size(I,1,2));

if isempty(varargin)
    error("Annotation colours must be given")
else
    for i=varargin
        switch i{1}
            case "All"
                colours=[255 0 0; 255 242 0; 0 255 0; 0 174 239; 255 128 0; 0 0 255];

```

```

        case "Red"
            colours(end+1,:)= [255 0 0];
        case "Yellow"
            colours(end+1,:)= [255 242 0];
        case "Green"
            colours(end+1,:)= [0 255 0];
        case "Ligth Blue"
            colours(end+1,:)= [0 174 239];
        case "Orange"
            colours(end+1,:)= [255 128 0];
        case "Blue"
            colours(end+1,:)= [0 0 255];
        otherwise
            if ndims(i{i})==3 && max(i{i}(:))<=255 && min(i{i})>=0
                colours(end+1,:)=i{i};
            else
                warning("The value "+i{1}+" will be ignored");
            end
        end
    end
end

for i=1:size(colours,1)

    colourMask=(I(:,:,1)==colours(i,1) & I(:,:,2)==colours(i,2) & I(:,:,3)==colours(i,3)); % Search for the
    pixel with same RGB values
    [rows, columns] = find(colourMask); % Extract their location

    mask=mask+colourMask(:, :,1);

    for j=1:length(rows)
        tags(rows(j),columns(j),:)=colours(i,:); % Copy the RGB values of each pixel
    end
end
end

```

Code Fragment 29 ExtractAnnotations.m.

### 7.2.11 SyntheticAnnotated.m

```

clear all
clc

hsDir = "TortosaCompleted\00_PushbroomRGB\";
annotatedDir = "TortosaCompleted\00_AnnotatedSlides\";
slidesDir = "TortosaCompleted\00_Slides\";

imageList={dir(hsDir).name};
imageList=imageList(3:end);
threshold = 0.7;
ERROR=0;
detector="SURF";
backUpDetector="BRISK";

for i=imageList
    flag=strfind(i,'c01');
    if isempty(flag{1}); continue; end
    [tags,mask]=ExtractAnnotations(strcat(annotatedDir,i{1}(1:end-4),'_annotated.png'),'All');
    Ihs=imread(strcat(hsDir,i{1}));

    [tform,Ir]=FeatureRegistration(strcat(slidesDir,i{1}),Ihs,"detector",detector,"showFeatures",false,"showMatches",false,"showRegister",false);

    if ssim(rgb2gray(Ihs),Ir)<threshold
        warning("The " + detector + " detector have not performed as spected (SSIM = " + ssim(rgb2gray(Ihs),Ir) +
        ")...
            +". The " + backUpDetector + " detector will be used.")
    [tform_backup,Ir_backup]=FeatureRegistration(strcat(slidesDir,i{1}),Ihs,"detector",backUpDetector,"showFeatures",false,"showMatches",false,"showRegister",false);

        if ssim(rgb2gray(Ihs),Ir_backup)<threshold
            ERROR=ERROR+1;
            warning("The " + backUpDetector + " detector have not performed as spected (SSIM = " +
            ssim(rgb2gray(Ihs),Ir) + "). The methodology have not work! #error = " + ERROR)

            if ssim(rgb2gray(Ihs),Ir_backup) < ssim(rgb2gray(Ihs),Ir)
                tform=tform_backup;
            end
        end
    end

    tagsR=imwarp(tags,tform,"OutputView",imref2d(size(Ihs)));
    tagsR=uint8(tagsR);
    maskR=imwarp(mask,tform,"OutputView",imref2d(size(Ihs)));

```

```
[rows,columns]=find(maskR==0);
Ihs_annotated=zeros(size(Ihs),"uint8");

for j=1:length(rows)
    Ihs_annotated(rows(j),columns(j),:)=Ihs(rows(j),columns(j),:);
end

Ihs_annotated=Ihs_annotated+tagsR;
imshow(Ihs_annotated)
end
```

*Code Fragment 30 SyntheticAnnotated.m.*