Keypoint Masking for Analyzing Segmented Medical Image Data

by

Étienne PEPIN

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT OF A MASTER'S DEGREE
WITH THESIS IN AUTOMATED MANUFACTURING ENGINEERING
M.A.Sc.

MONTREAL, 22 DECEMBER 2020

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Matthew Toews, Memorandum supervisor
Department of Automated Production Engineering, École de technologie supérieure

Mrs. Rola Harmouche, co-Supervisor
Simulation and digital health, National Research Council Canada

Mr. Alessandro Lameiras Koerich, President of the board of examiners
Department of Software and Information Technology Engineering, École de technologie supérieure

Mr. Simon Drouin, External examiner
Department of Software and Information Technology Engineering , École de technologie supérieure

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON 8 DECEMBER 2020

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

Utilisation de points-clés 3D SIFT-Rank avec des neuro-images

Étienne PEPIN

RÉSUMÉ

Ce mémoire examine plusieurs utilisations des masques de segmentation en conjonction avec l'analyse par point-clé dans le contexte de l'imagerie médicale. Actuellement, la procédure la plus populaire pour les neuro-images est l'extraction de points-clés sur une image sans tissu provenant de l'extérieur du cerveau. Notre hypothèse initiale est que cette technique génère du bruit près de la bordure du cerveau et nuit aux études cliniques. Nous proposons une nouvelle méthode basée sur l'extraction de points-clés sur l'image naturelle avant le filtrage par masque. Nous avons reproduit une étude récente par Chauvin, Kumar, Wachinger, Vangel, de Guise, Desrosiers, Wells & Toews (2020) sur la classification des relations familiales utilisant la base de données Human Connectome Project et comparé la méthode classique avec notre nouvelle méthode. Celle-ci dépasse par 2% la méthode classique dans les tests de classification familiale. Les résultats ont été publiés dans l'atelier MLCN (Pepin, Carluer, Chauvin, Toews & Harmouche, 2020). Nous avons développé un modèle théorique expliquant et prédisant ces comportements en se basant sur les propriétés de la loi normale multidimensionnelle. Notre méthodologie est générale et nous nous attendons à ce que nos résultats soient généralisables pour d'autres types de données et de systèmes de classifications basés sur la convolution linéaire, les réseaux de neurones convolutifs par exemple.

Mots-clés: skull-strip, extraction de cerveau, points-clés, neuro-image, filtre de Gauss

Keypoint Masking for Analyzing Segmented Medical Image Data

Étienne PEPIN

ABSTRACT

This thesis investigates different methods of using segmentation masks, in the context of keypoint analysis of medical images and specifically the human brain in magnetic resonance images (MRI). Recent studies have used keypoints extracted following skull-stripping, i.e. first removing all non-brain image content. However we hypothesized that skull-stripping prior to convolution filtering (e.g. Gaussian derivative filtering used in 3D SIFT-Rank keypoint extraction) will lead to random boundary effects that will hinder brain analysis. To test this hypothesis, we compare against keypoints extracted from natural images prior to skull-stripping. Our experiment replicates a recent large-scale neuroimage family indexing experiment on data from Human Connectome Project, where classification results improve on average 2% for keypoints extracted from natural data vs. skull-stripped data. We develop a theoretical model explaining and predicting experimental results based on the properties of a n-dimensional normal distribution. Our methodology is general, and we expect our results to generalize to other non-brain data, e.g. natural image regions and other classification systems based on linear convolution, e.g. convolutional neural networks.

Keywords: skull-strip, brain extraction, keypoint, neuroimage, Gaussian filter

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

DC   Dice Coefficient

SIFT   scale-invariant feature transform

DoG   Difference-of-Gaussian

ROI   region of interest

KDE   kernel density estimation

PDF   probability density function

HCP   Human Connectome Project

AUC   area under curve

ROC   receiver operating characteristic

MRI   magnetic resonance imaging

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

I	Image
Ω_I	Sample space of Image I
$\mathbb{N}$	Natural numbers
$\mathbb{Z}$	Integers
$\mathbb{R}$	Reals
$|A|$	Cardinality of set A
$\|\mathbf{p}\|$	Euclidean magnitude of vector p
$G(x:\sigma^2)$	Gaussian filter
σ	Keypoint scale
X	Keypoint's spatial location
Θ	Keypoint's orientation
g	Keypoint's geometry
a	Keypoint's descriptor
F	Image as a set of keypoint
m	Binary mask
$M(F,m)$	Masking operation of Image F by mask m
R_k	Ratio of data encoded in the keypoint originating from the volume of the keypoint

INTRODUCTION

Neuroimages studies often apply brain masking (as seen in figure 0.1 (b)) to remove non-brain content prior to analysis (Smith (2002)). Masking algorithms may introduce random variations and noise into the image, due to the imprecision in identifying the boundary of the brain, and these may adversely affect subsequent image analysis algorithms involving convolution filtering. For example, keypoint-based analysis via the 3D SIFT-Rank algorithm (Toews, Wells III, Collins & Arbel (2010)) where salient keypoints are extracted based on the strong reaction of the Laplacian filter to sharp image borders. Does border noise due to image masking negatively impact keypoint analysis? If so, can masking be used in an alternative fashion to reduce the impact of noise linked to masking?

(a) Natural Image   (b) Masked Image
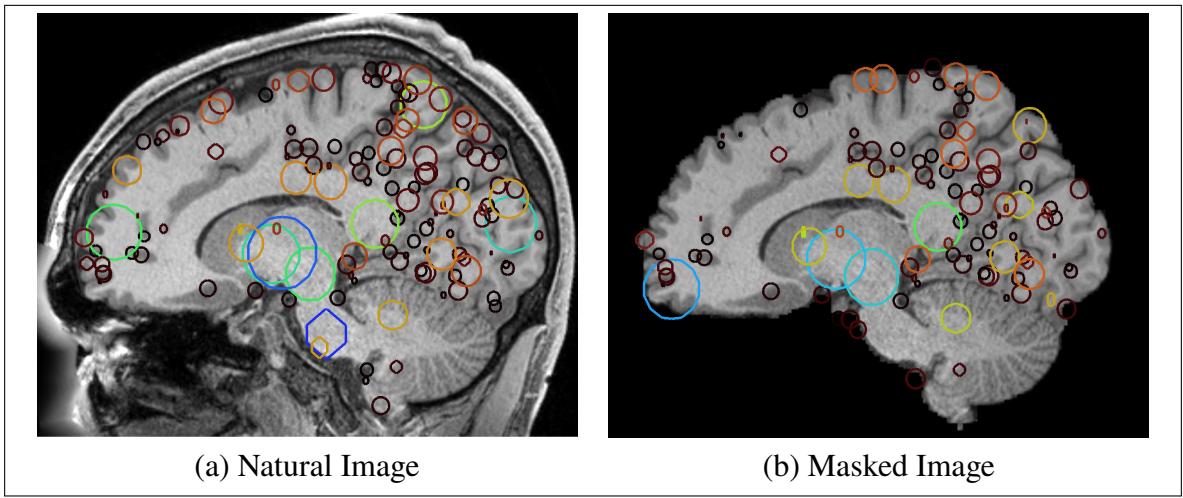
Figure 0.1   Keypoints (circle) extracted in original and skull-stripped images. Note how different they are.

The main contribution of this thesis is: a procedure detailing the best practice for keypoint extraction for the purpose of medcial image analysis. It states that keypoints should always be extracted from natural image data prior to masking. Our novel masking process then allows to remove any keypoint inside a mask containing too much non-mask content according to the project's needs. We also make suggestions for using masked images while reducing noise.

Our masking process removes keypoint closer then a specified distance to the border mask. We tested it in a replication of the HCP family relationship classification experiment by Chauvin *et al.* (2020) and showed improved classification from 86.5% to 88.9% for the full-siblings (FS) class and from 90.9% to 92.6% for the dizygotic twins (DZ) class compared to the original procedure using intensity-masked keypoints.

The primary hypothesis of this thesis is that, because image masking procedures such as skull-stripping may introduce noisy information, region-based masking should be performed *after* feature extraction rather than before, as is the common practice. The novel method we developed to investigate this hypothesis results from developing a theoretical model detailing neighborhood data contribution to keypoint encoded information. Our model quantifies the amount of information from outside the mask encoded in a keypoint as a function of its distance to the mask's border. This enables us to discard uninformative keypoints by removing them depending on their distance to the mask's border. We developed our model by analyzing mathematically how keypoints are extracted. Keypoint data is extracted from the image with Gaussian filters and the data encoded can be analyzed mathematically based on those filters. Knowing the original location of encoded data allows us to estimate the amount of information within the keypoint coming from background regions, based on a model of Gaussian diffusion, and reject keypoints near the mask boundary. For example, keypoints extracted from intensity-masked images situated on the mask's border will attribute around 50% of their content to blank artificial content.

We confirmed our model by comparing a of family members using classification test on keypoints extracted from natural images and skull-stripped images. Many masks using various distances from the original border were tested on both types of keypoints. At no distance from the brain mask's border, we get the result we shared above, around a 2% increase in classification performance. At a distance equal to the diameter of a keypoint, results are the same for natural

and skull-stripped images. Since our model told us that at that distance, 94% of the data originated from inside the brain mask, it makes sense that results are the same for skull-stripped and natural images. At a distance of half the keypoint diameter, results are mostly better for natural images, but less markedly. Interestingly, results were better with skull-stripped images at that distance than when not removing any keypoints.

We performed the above mentioned experiment on images with an isotropic resolution of 0.75mm and 1.25mm. The results we discussed above are for the 0.75mm resolution images. They offer the overall better classification rates. But at a 1.25mm resolution, the advantage of natural images is clearer than at 0.75mm. The uninformative keypoints generated from a skull-stripped image are situated near the border of the brain mask. The better the resolution, the smaller the brain mask's border is compared to the brain volume. This suggests that our procedure is particularly needed when working with brain images at lower resolutions.

We also generate a visualization of keypoints extracted from skull-stripped images matching to keypoints extracted from natural images. The visualization enables us to observe how intensity-masked keypoints are affected near the border of the skull-stripping mask.

In parallel, we analyze how feature scale influences signature matching between twins. We also provide a novel keypoint segmentation algorithm (annex III), based on the Kernel Density Algorithm, which classifies keypoints invidually. This segmentation experiment is a totally different inquiry. We interrupted since it did not seem to produce interesting results.

CHAPTER 1

BACKGROUND AND FUNDAMENTALS

This chapter details most of the technical knowledge necessary to understand our work, including general computer vision knowledge, mathematical notions, specifics on used technologies and pertinent literature review. This thesis makes use of medical imaging and machine learning technologies. We describe this here focusing specifically on filtering techniques, including the Gaussian filter. We present the 3D SIFT-Rank algorithm, explain in detail our initial hypothesis about border interference and introduce the HCP family relationship classification experiment we use to grade our method.

1.1 Medical Imaging

Medical imaging is the representation of the interior of the body. In our case we will analyze neuroimages, which are representations of the brain. Neurostudies can be based on different modalities, depending on the available technology and needs of the study. Magnetic Resonance Imaging (MRI) is the most common modality in brain studies. It uses the magnetization properties of atomic nuclei to produce volumetric images. The 2 most common procedures for MRI scans are named T1 and T2 weighted MRI. The different parameters used result in different intensities in the resulting image for the same tissues. Another modality used for neurostudies is the Computed Tomography (CT). It uses X-rays emitted from many angles to generate a volumetric image, by measuring the amount blocked by body tissues. Since it is based on X-rays, the amount of radiation received by the patient must be controlled, which limits the frequency of exposure. We will work with MRI T1-Weighted images, mainly because of it's prevalence in the field. But much of our conclusions could be applied to different modalities.

1.2 Computer Vision Fundamentals

This section contains fundamental notions of computer vision knowledge, essential to understand some aspects of our research. Information present in this section can be found mostly in Gonzalez & Woods (2006) and is commonly seen in computer vision classes.

1.2.1 Digital Images

Digital images are images represented with picture elements, named pixels in 2 dimensions and voxels in 3 dimensions. Each picture element contains a discrete quantity called intensity. We will represent images in our work with this mathematical model:

$$I(\mathbf{p}) = i \mid \mathbf{p} \in \Omega_I \tag{1.1}$$

$$\mathbb{N}^3 \rightarrow \mathbb{R}^1 \tag{1.2}$$

$$\Omega_I = \{\mathbf{p} = [x, y, z] \mid \mathbf{p} \in \mathbb{N}^3, \ x < A, \ y < B, \ z < C\} \tag{1.3}$$

Where I is an image of size (A, B, C) and Ω_I is the image space.

1.2.1.1 Distance Between Two Points

It is often useful to calculate the distance between 2 points in an image. Many techniques can be used to calculate distances in an image, but we will be using the Euclidean distance to calculate distances between 2 points $p = (p_1, p_2, p_3)$ and $q = (q_1, q_2, q_3)$:

$$d(p, q) = \sqrt{\sum_i (q_i - p_i)^2} \tag{1.4}$$

Sometime a point p may be represented as a vector $\mathbf{p}$, in which case the norm of the vector is the distance from the origin to p

$$\|\mathbf{p}\| = \sqrt{\sum_i p_i^2} = \sqrt{\mathbf{p} \cdot \mathbf{p}} \tag{1.5}$$

Where "$\cdot$" is the dot product. The Euclidean distance between 2 vectors p and q is:

$$\|\mathbf{q} - \mathbf{p}\| = \sqrt{\|\mathbf{q}\|^2 + \|\mathbf{p}\|^2 - 2\mathbf{p} \cdot \mathbf{q}} \tag{1.6}$$

1.2.1.2   Erosion Operation

The erosion is part of morphological operations. It is used on binary images to reduce the size of regions or eliminate very small regions. The erosion of set A by set B is defined by:

$$A \ominus B = \{\mathbf{p} \in \Omega_I | B_z \subseteq A\} \tag{1.7}$$

where B_z is defined by:

$$B_z = \{\mathbf{p} + \mathbf{b} | \mathbf{b} \in B\} \tag{1.8}$$

Where $B \in \mathbb{Z}^3$ and $A \in \Omega_I$. B is called a structuring element. Typically, the structuring element is isotropic and some of the common shapes are a sphere, cube and diamond. The sphere has the advantage of having the same distance in any direction, but since structuring elements are usually small, the sphere shape can be a rough pixelated approximation.

1.2.2   Spatially Local Filtering

The content of most human-interpretable images can be described as a collage of localized regions containing pixels of the same object, scene or texture. A fundamental operation in computer vision is thus to characterize the image content within spatially localized image regions.

Spatially local filtering can be viewed producing a result image $R : \mathbb{R}^3 \to \mathbb{R}$, where the value of $R(x, y, z)$ at each voxel (x, y, z) is a function of the voxel values in the original image I at neighborhood $N_{(x,y,z)} = \{(x_i, y_i, z_i) : \|x_i - x, y_i - y, z_i - z\| \leq Thres\}$, where $Thres$ is a proximity threshold.

$$R(x, y, z) = f(I, N_{(x,y,z)}). \tag{1.9}$$

In the case of a linear filter, e.g. convolution, it modifies each voxel of an Image I with a linear function of neighbouring voxels, involving voxel-wise multiplication and addition.

1.2.2.1   Gaussian Filter

The Gaussian filter is commonly used in computer vision applications to reduce image noise. It can be seen as a local average with more weight given to the closest neighbours. The filter is defined in n-Dimensions as

$$G(\mathbf{x} : \sigma^2) = \frac{1}{(\sqrt{2\pi}\sigma)^n} e^{-\frac{\|\mathbf{x}\|^2}{2\sigma^2}} \tag{1.10}$$

Where $n \in \mathbb{N}$, $\sigma^2 \in \mathbb{R} > 0$ is the variance and $\mathbf{x} \in \Omega_I$.

It is seen in probability as the multivariate Normal distribution. The random variable $\mathbf{X} = [X_1, X_2, ..., X_n]$ is said to follow a multivariate distribution, noted as $\mathbf{X} \sim \mathcal{N}(\mu, \Sigma)$ if $\mathbf{x}$ follows the probability density function (PDF):

$$f(\mathbf{x} : \mu, \Sigma) = \frac{1}{\sqrt{|\Sigma|(2\pi)^k}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right) \tag{1.11}$$

Where $\mu \in \mathbb{R}^n$ is the mean, Σ is a square covariance matrix and $|\Sigma|$ is the determinant of the covariance matrix. The Gaussian filter is an isotropic filter, which reduces the covariance matrix

to a matrix with value σ^2 on the diagonal. It also has an average of $\mu = 0$, which removes the term.

The Gaussian filter is a core component of the 3D SIFT-Rank keypoints, which will be discussed later.

1.2.2.2 Laplacian Filter

A Laplacian filter is a spatial filter of the second order derivative often used to detect or highlight sudden change in intensity. Sharpening filters often make use of a Laplacian filter. The second order derivative is the rate of change of a function. Let's say $f(x, y)$ represents any point $(x, y) \in \mathbb{R}^2$ in an image I. Then

$$\frac{\partial f}{\partial x} = f(x + 1) - f(x) \tag{1.12}$$

represents the change of the image I in the direction x. Since x is a discrete variable, the derivative can be calculated by comparing a pixel to its neighbour's value along the x axis.

$$\frac{\partial^2 f}{\partial x^2} = [f(x + 1) - f(x)] - [f(x) - f(x - 1)] = f(x + 1) + f(x - 1) - 2f(x) \tag{1.13}$$

The second order derivative is calculated by subtracting a pixel's first order derivative to its neighbours along the x axis.

The suggested filter above is anisotropic since it has different properties depending on the direction. Such properties can be desired to detect the rate of change in a particular direction, but most often an isotropic filter is used. The equation below and figure 1.1 are an example of an isotropic Laplacian filter.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y) \quad (1.14)$$

0	1	0
1	-4	1
0	1	0

Figure 1.1   Laplacian Filter

And here is the equation for the 3D filter:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} \quad (1.15)$$

$$= f(x+1, y, z) + f(x-1, y, z) + f(x, y+1, z) + f(x, y-1, z) \quad (1.16)$$

$$+ f(x, y, z+1) + f(x, y, z-1) - 6f(x, y, z) \quad (1.17)$$

1.2.3   Skull-stripping

Skull-stripping, sometimes called brain extraction, is a standard preprocessing step in most image-based neurostudies. For instance the Brain Extraction Tool (BET) (Smith, 2002) has been widely used since 2002. Volumetric images generated for neurostudies are imprecise, they often include the skull as well as other body parts above the shoulders. To strictly limit the data used in those studies to brain tissues, non-brain tissues in the image are masked, a process called skull-stripping. This is an active area of medical imaging research because of its vital role in any image-based neurostudy. Skull-stripping procedure are often compared together with a metric called the Dice Coefficient.

(a) Original Image   (b) Skull-stripped Image
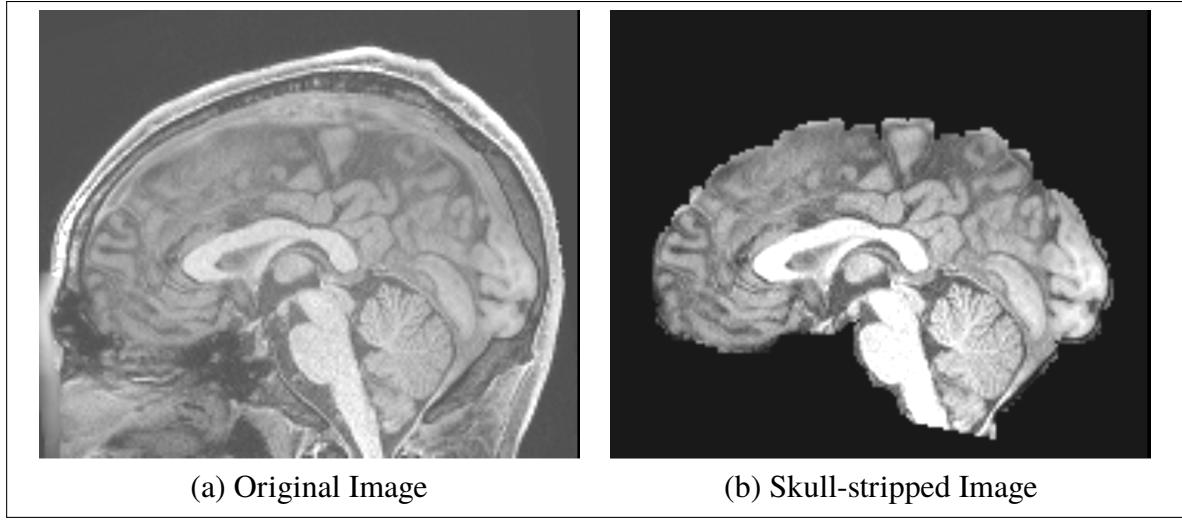
Figure 1.2   Original and skull-stripped images

The Dice Coefficient is one of the most common metric of performance of skull-stripping. It's a measure of how much a set matches another. In our case, how much the computed brain's mask overlaps with the real brain, ranging from 0 to 1 when it matches perfectly.

$$DC = \frac{2|A \cap B|}{|A| + |B|} \tag{1.18}$$

Where $|A|$ is the cardinality (size) of set A and $|A \cap B|$ is the intersection of set A and B. A weakness of the DC is that bigger objects tend to obtain better DC, since the area near the border represents a smaller percentage of the whole organ.

The different skull-stripping techniques available are well documented by Palanisamy & Prasath (2015). They can be categorized into five categories: mathematical morphology-based methods, intensity-based methods, deformable surface-based methods, atlas-based methods and hybrid methods as stated in Palanisamy & Prasath (2015). These different techniques have been compared by Iglesias, Liu, Thompson & Tu (2011) and often obtain a Dice Coefficient above 90%.

Performances of skull-stripping depends on many factors, including the skull-stripping algorithm, the characteristics of the imaging apparatus, the imaging modality and the patient morphology. Many open-source software are available for performing those algorithms. The data we used for our experiments was skull-stripped with FreeSurfer (Fischl, 2012), which obtained an average DC of 93.9% on the OASIS dataset(Iglesias *et al.* (2011)).

1.3   Salient Keypoints

Salient keypoints are representations an image as a list of distinctive local features from that image. Those features typically aim to be invariant to scale and rotation variations. This approach offers many benefits. The features provide robust matching over a variety of affine transformations and resulting image representations are more compact, taking less memory. This enables the use of nearest neighbour strategies for matching over large databases.

One of the most popular salient keypoint algorithm is SIFT (scale-invariant feature transform), which was published by Lowe (2004). At the time, the technique was well positioned for object recognition compared to other available algorithms. Since then, other algorithms have been developed to generate keypoints and deep learning algorithms are often preferred over keypoints matching for object recognition.

We will be using the 3D SIFT-Rank keypoint method (Toews *et al.*, 2010) in our work, which is based on the SIFT algorithm but expanded to work with 3D images. As described in the above mentioned article: "The goal of invariant feature extraction is to identify and characterize informative image patterns in a manner independent of global variations in image geometry and appearance, e.g. due to misalignment or intensity changes". An overview of the algorithm is provided below and shown in figure 1.3.

1.3.1   3D SIFT-Rank: Keypoint Extraction

Let $f_i = \{g_i, a_i\}$ represents a feature of an image represented as keypoints $F = \{f_1, f_2, ..., f_n\}$. $g = \{X, \sigma, \Theta\} \in \mathbb{R}^7$ represents a 3D SIFT-Rank keypoint's geometry and $a \in \mathbb{R}^{64}$ is it's
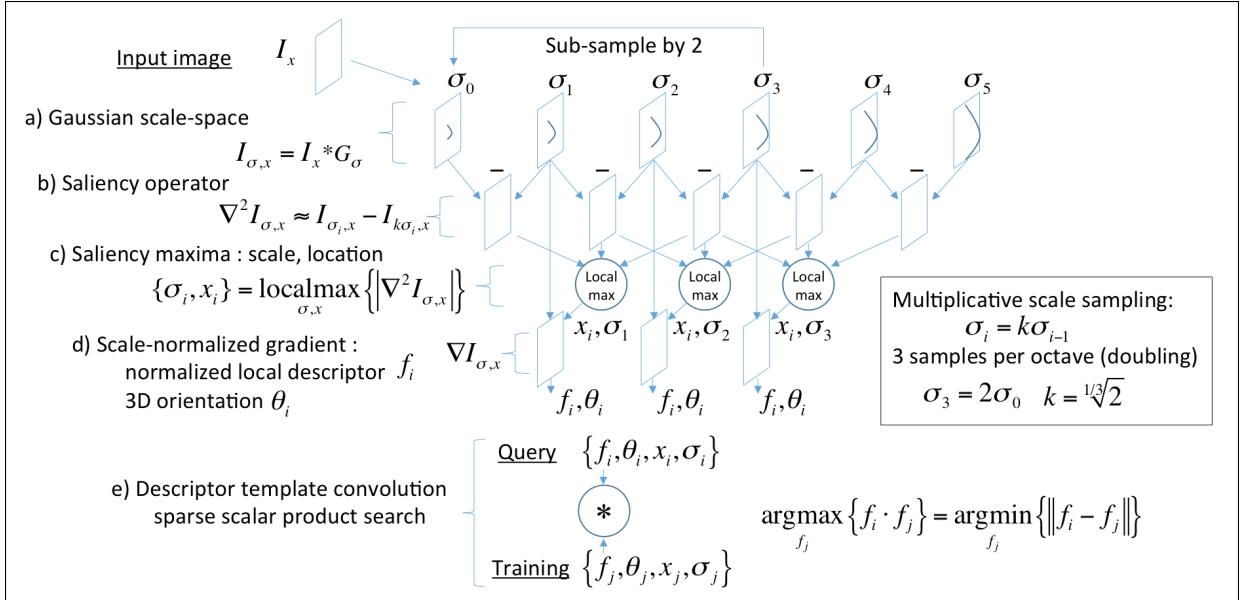
Figure 1.3   3D SIFT-Rank extraction process. Step a) is the sub-sampling of the image using Gaussian filters of different scales. Step b), Saliency operator, computes the Difference of Gaussian (DoG) of images of different scales σ_i. Step c), Saliency maxima, detects local maximum. Step d), Scale-normalized gradient, computes the value of the keypoint.

descriptor encoding local intensities. Let $X = [x, y, z] \mid X \in \Omega_I$ represent the spatial location of the centre of the keypoint, σ the scale of the keypoint and $\Theta = \{\theta_1, \theta_2, \theta_3\}$ the 3 orthonormal angles necessary to specify an orientation in 3D. The first step in extracting keypoints is identifying potential keypoint locations, then assigning orientation and finally encoding intensity. Code is available online [1] and Figure1.3 shows the process.

1.3.1.1   3D SIFT-Rank: Detection of scale-Space Extrema

The first step in keypoint detection is determining the location around which each keypoint descriptor will be measured. The points of interest are located by identifying local maximums across scale and space. A Gaussian scale-space is used to subsample the image at different scales.

$$I(X, \sigma^2) = I(X) * G(\sigma^2) \tag{1.19}$$

$$G(X, \sigma^2) = \frac{1}{(\sqrt{2\pi}\sigma)^3} e^{\frac{-||X||^2}{2\sigma^2}} \tag{1.20}$$

Where $*$ is a convolution operation.

There are 3 samples for every scale doubling, also called an octave. It follows the formula:

$$\sigma_i = 1.6x2^{i/3} \tag{1.21}$$

In this case $\sigma_0 = 1.6$. The choice of having 3 subsamples per octave was made in the original SIFT Lowe (2004) design and performed well against other designs. The Difference-of-Gaussian saliency operator is used to detect stable keypoint locations in scale space.

$$D(X, \sigma_i) = I(X, \sigma_i) - I(X, \sigma_{i+1}) \tag{1.22}$$

Local maxima are identified by comparing a voxel from a DoG image to its 80 $(26 + 2 * 27)$ voxel neighbours across the 2 adjacent (in scale) DoG images (see Figure 1.3).

1.3.1.2 3D SIFT-Rank: Assigning Orientation

The orientation and intensity encoding starts by selecting a cubic patch of side 4σ centred around local maxima. This patch is resized to a cube of $11x11x11$ pixels.

Pixel gradients are calculated through the biggest sphere fitting in this cube. Those gradients are used to populate a gradient histogram, that can be visualized as the surface of the sphere depicted in 1.4 section a). Normalization and smoothing are applied to the histogram to reduce noise. The bin of the histogram with the highest count is considered the primary orientation vector $\hat{\theta}_1$. The secondary orientation is obtained by selecting the highest gradient on the circle
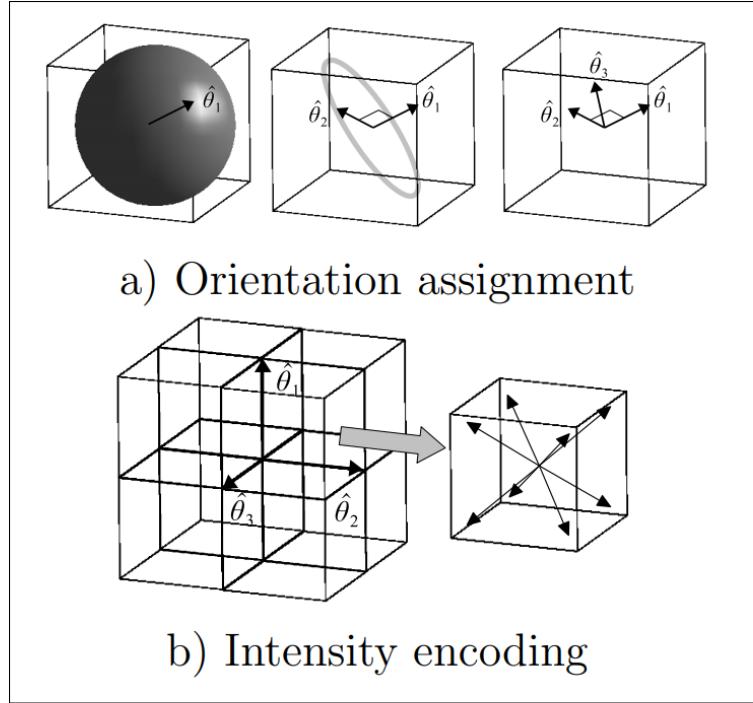
Figure 1.4   3D SIFT-Rank encoding taken from
Toews & Wells III (2013)

orthogonal to the primary orientation. And finally the tertiary orientation is the cross product between the 2 other orientations.

1.3.1.3   3D SIFT-Rank: Intensity Encoding

We reuse the cube mentioned before, as well as the calculated gradients inside. The cube is divided in octants. The gradients of the octants will be incremented for each pixel's gradient in the octant. Each octant contains 8 orientation bins. The process is shown in Figure 1.4 section b). Using intensity gradient instead of intensity provides the advantage of producing registration more resistant to small changes in the image Lowe (2004). Finally, the gradient intensities are ranked from 1 to 64, which provides invariance to monotonic deformations (Toews & Wells III, 2013) and good encoding performance. This is similar to the gradient normalization used by Lowe (2004) in its use against monotonic deformations, such as change in illumination.

1.3.1.4 Masking Keypoints

Masking is used to specify the region of interest (ROI) of an image. We will define the operation mathematically for keypoints using the set operations and Ω_I we defined previously.

Let our mask be a set m:

$$m = \{\mathbf{p} \in \Omega_I \mid \mathbf{p} \in ROI\} \tag{1.23}$$

An example of ROI could be the brain. In which case we would interpret m as the set of all keypoints part of image space of I, such that they are in the brain.

The masking operation is defined with the function $M(F, m)$ for a mask m and an image as a set of keypoints $F = \{f_1, f_2, ... f_n\}$, where keypoint $f_i = \{g_i, a_i\}$ and $g_i = \{X_i, \sigma, \Theta\}$:

$$M(F, m) = \{f_i \in F \mid X_i \in m\} \tag{1.24}$$

1.3.2 3D SIFT-Rank keypoint Uses

In the original article by Toews & Wells III (2013), 3D SIFT-Rank were used for model-to-image alignment, which consists of aligning an image in a particular standard orientation dictated by a model. 3D SIFT-Rank have also been used by Wachinger, Toews, Langs, Wells & Golland (2015) for whole body segmentation. In general, salient keypoints provide a compact signature of an image. That signature can be compared to a large database containing similar signatures via a nearest neighbour search. The same process can be used with a subset of an image, enabling object detection. Nearest neighbour searches, contrary to neural networks, work without training. This is particularly useful when dealing with small number of samples.

1.4 Border Interference of Skull-Stripped Image

This section details our reasoning for believing that border noise is generated when extracting salient keypoints from skull-stripped images. Skull stripped images are sharply divided in 2 sections: brain tissue, with varying greyscale intensities and non-brain tissue, all with an intensity of 0. Skull-stripping algorithms often get over 90% Dice Coefficient, but the border tends to be jagged and varies depending on the skull-stripping algorithm used. We believe those properties interact poorly with the Laplacian Filter used to extract 3D SIFT-Rank keypoints. Laplacian filters react strongly to sudden changes in intensities, which will always be present on a skull-stripped image near the mask's edge. This will cause the algorithm to find keypoints near the edge of the mask solely because the mask ends there, making the keypoint extraction very dependant on the exact mask's borders and the skull-stripping algorithm.

1.5 HCP Family Relationship Classification

A recent study by Chauvin *et al.* (2020) investigated the neuroimage signature obtained with salient keypoints and the signatures of observed semblance between family members' signature. This study used a collection of 4 public databases, including the Human Connectome Project(HCP), containing a total of 8152 images. The Human Connectome Project (Van Essen, Smith, Barch, Behrens, Yacoub, Ugurbil, Consortium et al., 2013) contains a number of related family members appropriately tagged.

The signature of each subject is represented as a set of 3D SIFT-Rank keypoints. The keypoints are extracted for each subject after skull-stripping the images using the FreeSurfer (Fischl, 2012) pipeline. This results in a set of keypoints situated in the skull-stripped brain mask or near the edge of it.

The pairwise relations between all subjects were categorized as one of the 5 following relationships: same subject (SM), monozygotic twins (MZ), dizygotic twins (DZ), full- sibling (FS) or unrelated subjects (UR). A pairwise comparison is made by measuring the Jaccard overlap (eq. 1.25) of each of the $\binom{N}{2} = N(N-1)/2$ image pairs. It is a measure of the proportion of the

keypoint correspondences shared by an image pair:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{1.25}$$

where $|A|$ is the cardinality of the set A, $|A \cap B|$ represents the number of keypoint correspondences between image pair (A, B) and $A \cup B$ is the union between set A and B. The correspondences are not binary, they are a measure of the similarity between a pair of keypoints' descriptors. Each class of relationship between pairs has a distinct Jaccard coefficient distribution that enables us to classify the relationship with a Jaccard coefficient threshold as shown by the figure 1.5 produced in the study.
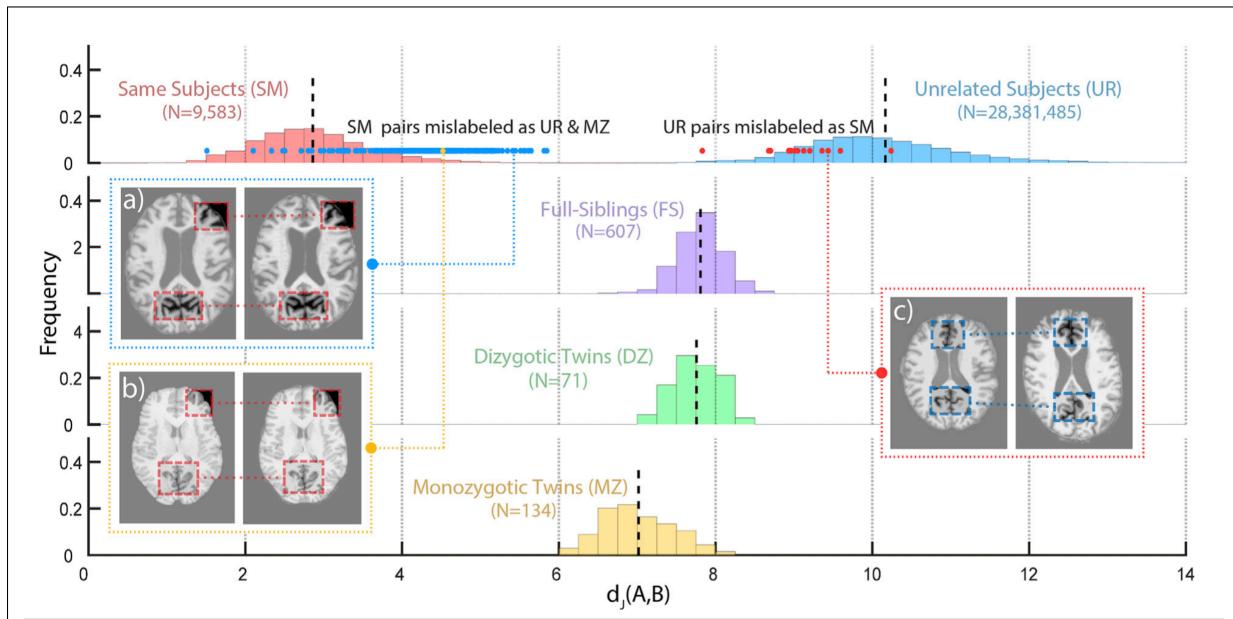
Figure 1.5   Distributions of the pairwise Jaccard distances conditional on relationship labels taken from Chauvin *et al.* (2020).

The SM, MZ, DZ, FS and UR labels all have their distinct distributions (*difference between FS and DZ twins is statistically insignificant) There's no overlap between SM distribution and UR distribution and the study found many labeling inconsistencies in databases using this property. On the other hand, the MZ, DZ and FS distribution have some overlap. In our study we will reproduce the original experiment and we expect less overlap of the MZ, DZ and FS distributions by using more informative keypoints in higher number.

1.5.1 Technology Used

Most of the experiments conducted during this research are coded in Python. Python is an open-source, high-level, interpreted language with a wide array of public libraries available. Here are the libraries that were used in this project:

- Numpy[2]: widely used open-source library providing support for multi-dimensional matrices and functions to operate on these matrices.
- NiBabel[3]: library providing read/write utilities to common medical image formats.
- SciPy ndimage[4]: library compatible with Numpy matrices and providing support for multidimensional image processing.
- Flann[5]: We use it for its randomized KD-tree Muja & Lowe (2014), an approximate nearest neighbor algorithm that perform particulary well with high-dimension data.
- Numba[6]: translates Python functions to optimized machine code at runtime. Particularly useful when needing a function not available in any library that iterates over every voxel of a 3D image. Those functions run slowly on Python since it's an interpreted language. Numba circumvent this by compiling functions at runtime, but, from experience, coding functions compatible with Numba can be hard and should be left as a last resort.
- pandas[7]: Useful library to accelerate the reading and writing of files containing large amount of data.

Some additional notes on the Scipy ndimage library: working with 3D images is less common than working with 2D images, so there tends to be less support for it. The OpenCV library is widely used in computer vision with Python, but has poor support for 3D images. Scipy ndimage is the best library we could find supplying n-dimensional functions to work with images.

---

[2]https://numpy.org/
[3]https://nipy.org/nibabel/
[4]https://docs.scipy.org/doc/scipy/reference/ndimage.html
[5]https://pypi.org/project/flann/
[6]http://numba.pydata.org/
[7]https://pandas.pydata.org/

Some of the code used in our experiments is available online[8]. This repertory contains many useful utility tools to manipulate 3D SIFT-Rank keypoints.

---

[8]https://github.com/pEtienn/3D-SIFT-keypoints-utilities

CHAPTER 2

METHODOLOGY

In this section we compare two methods for representing medical images with keypoints: extracting keypoints on a masked image and filtering keypoints extracted from the natural image. We will start by comparing those 2 methods mathematically, so that some of our conclusion might be valid for any keypoint masking application. Then we will perform experiments on brain keypoints and compare the results with our mathematical conclusions. While this thesis focus on keypoints analysis, results should apply to any system involving linear filtering such as CNNs, when working with masked images.

2.1 HCP Database

Experiments were conducted using a specific brain database, but we expect results will generalize for other databases as well. The Human Connectome Project (Van Essen *et al.*, 2013) database was chosen so that we could compare our results to the original experiments in Chauvin *et al.* (2020), who also used this database. A subset of 1010 subjects from the HCP Q4 release containing 439 unique families, including some unrelated subjects (see table 2.1). T1-weighted MR images have been acquired between 2012-2015 on a 3T MR scanner, at a 0.7mm isotropic resolution. Through the Freesurfer pipeline, images have been registered to the MNI space, brain masks have been generated, and images have been resampled to a 1.25mm isotropic resolution, as well as corrected for image artefacts such as eddy-currents and head-motion. Keypoints are extracted from individual images. The number of keypoints per method is shown in table 2.2 .

2.2 Intensity masking and keypoint masking

For brain studies, the standard procedure is to remove all non-brain tissue using intensity masking on the image before doing any analysis. In the case of studies based on keypoints, it makes certain any keypoint generated thus is based on brain intensity. Our hypothesis is that keypoints extracted near the border of such image are affected by the sudden change in intensity in the

Table 2.1   HCP demographic
information

Image number	1010
age	29 ± 13
male	468
female	542
Full Siblings (FS)	607
Dizygotic Twins (DZ)	71
Monozygotic Twins (MZ)	134

Table 2.2   Average number of keypoints extracted and
pairwise correspondence counts

methods		# keypoints	corres.
	intensity-masked	1468 ± 189	233.8
0.7 mm	masked	1662 ± 241	264.8
	original	2102 ± 277	335.4
	intensity-masked	180 ± 34	28.9
1.25 mm	masked	253 ± 54	40.8
	original	334 ± 60	53.8

masked image. If so, is there any alternatives for selecting keypoints such that guaranties can be made on the provenance of keypoint information? In our experiments we will compare 2 main methods of using keypoints belonging to a specific class j: intensity masking and keypoints masking. Currently the intensity masking method is the one commonly in use when analyzing neuroimages with keypoints. We introduce the keypoint masking method, which extracts keypoints from the natural image instead of a modified one. Both methods are shown in figure 2.1.

(a) intensity-masked keypoints   (b) Masked Keypoints (proposed method)
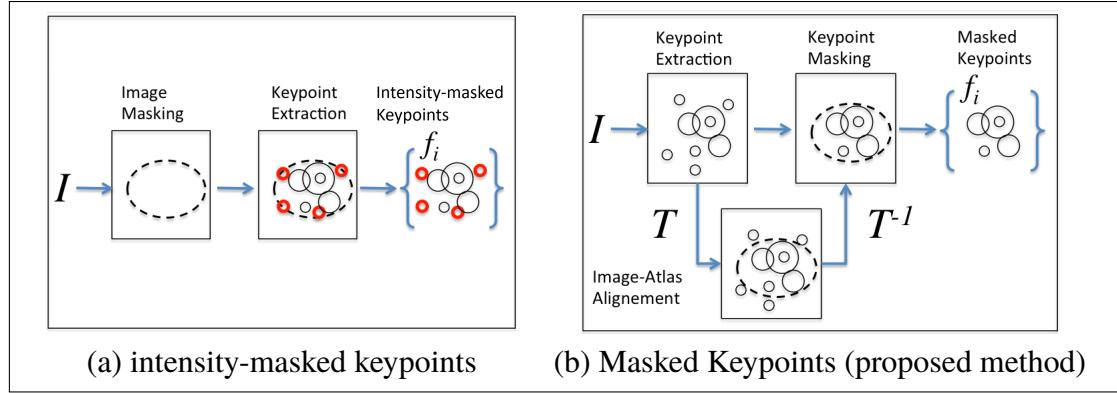
Figure 2.1   Illustrating a) Intensity-masked keypoint analysis, where intensity masking prior to filtering may lead to artifacts. b) Keypoint masking, where keypoints are extracted from natural image data, then filtered according to a mask. Reproduced and adapted from Pepin *et al.* (2020).

Intensity-masked keypoints is the term we use to refer to keypoints extracted from an intensity-masked image. An intensity image (I) masked by a class j mask is defined this way:

$$m_j = \{\mathbf{p} \in \Omega_i \mid \mathbf{p} \in \text{object } j\} \tag{2.1}$$

$$I_j(\mathbf{p}) = \begin{cases} I(\mathbf{p}), & \text{if } \mathbf{p} \in m_j \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

In the case of brain images, a brain mask is used to keep only brain content (i.e skull-stripped), then 3D SIFT-Rank keypoints are extracted from the image. The extraction process is thus applied to an image with an artificial, arbitrary border that can lead to the problems discussed in the section 1.4.

Keypoint masking is the process of removing keypoints from a keypoint image to keep only keypoints of a specific class. For example, keypoints are extracted from a natural brain image resulting in the set of keypoints F. Then, keypoint masking is used to keep only keypoints in image F situated in the brain mask. We define the keypoint masking operation mathematically:

$$F_j = M(F, m_j) \tag{2.3}$$

Standard keypoints (or simply "keypoints") are extracted directly from the natural image.

Some **notations**:

- keypoints: $F = \{f_1, f_2, \ldots, f_n\}$
- intensity-class j-masked keypoints: $F(I_j)$
- masked keypoints: F_j
- intensity-class j-masked class-c-masked keypoints: $F_c(I_j)$

Another approach for keypoint masking is to keep only keypoints whose centre is a certain distance inward away from the class j mask's border. We will use a distance of $d\sigma$ for now, where $d \in \mathbb{R} >= 0$. One way of selecting keypoints this way is using the erosion operation to create smaller masks as a function of σ, the scale of keypoints.

$$m(\sigma) = m_j \ominus e_{d\sigma} \tag{2.4}$$

where $e_{d\sigma}$ is a structuring element with radius $d\sigma$, a sphere for example. The masked keypoints thus becomes:

$$F_m(\sigma_j) = M(F(\sigma_j), m(\sigma_j)) \tag{2.5}$$

where $F(\sigma_j) = \{f_l \in F \mid \sigma_l \approx \sigma_j\}$ The erroded masks and selected keypoints can be seen in figure 2.2.

With this approach, a different mask must be used for each range σ_j, We will develop this strategy in a later section. In general, we will use the notation $F_{j-d\sigma}$ to refer to those keypoints.

Using only the centre position is the fastest approach, but some of the resulting keypoints will contain information from outside the brain mask. Adding a distance to the border could insure that mainly brain information is used, but it can be a lot longer to process and reduces the number of keypoints for each patient.
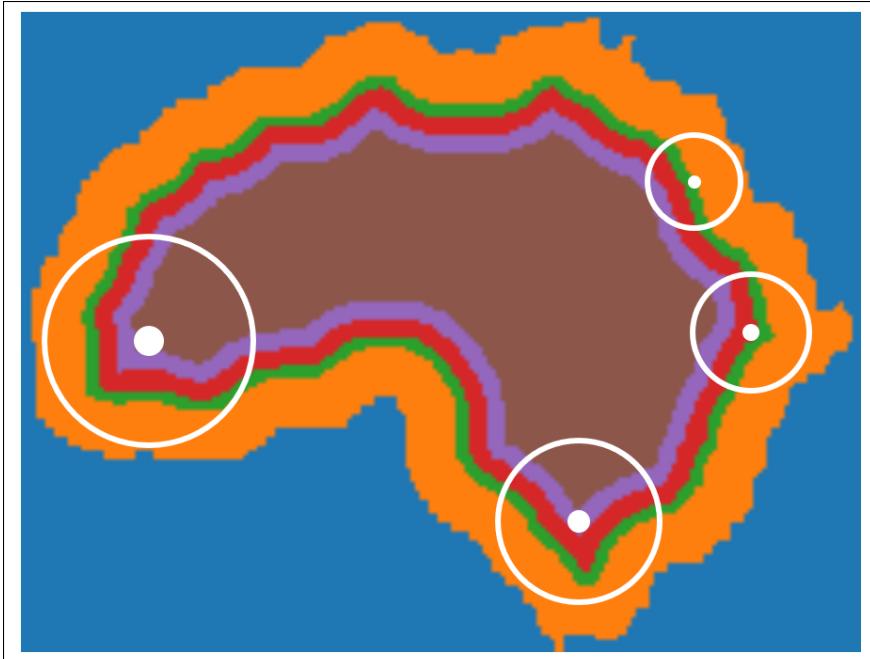
Figure 2.2   Erroded Masking

2.2.1   Data From Outside the Mask in Masked Keypoints

Using intensity-masked keypoints assures us that data from outside the mask is not contained in extracted keypoints. But doing so reduces the number of useful keypoints extracted. Our hypothesis is that intensity-masked keypoints near the border of the mask tend to provide lower quality information. Some of those keypoints even have their centres outside of the mask. We want to develop a scale dependent method for masking based on the proportion of brain content encoded in keypoints. In this section, we analyze the data encoded by keypoints in various situations to quantify the origin of data encoded.

Keypoints are often depicted as spheres in 3D or as circles in 2D. But the data encoded in those keypoints is not restricted to those spheres (or circles). The data encoded is generated from the Difference-of-Gaussian operator, which is calculated with 2 Gaussian filters of variance σ_i^2 and σ_{i+1}^2. Gaussian filters have an infinite range. Hence, data encoded in masked keypoints is attributed in part to non-masked tissues. We will design an experiment to calculate exactly how much data originates from non-masked tissue in different scenarios. **We will use the variable**

R_k **to refer to the ratio of data encoded in the keypoint originating from the volume of the keypoint**. A higher R_k means that the keypoint is more local.

We can get an idea of what to expect by simplifying the problem a bit. Let's assume that the region of the keypoint with the highest R_k is the centre of the keypoint, since it's surrounded by the largest coefficients of the Gaussian kernel. The centre's R_k is equivalent to calculating the probability of a 3-dimensional normal distribution of drawing a point at random in a sphere of radius 2σ. We can calculate it mathematically.

We will use the Gaussian filter equation 2.6 using a radius r instead of a vector $\mathbf{x}$ and in 3 dimensions:

$$G(\mathbf{x} : \sigma^2) = \frac{1}{(\sqrt{2\pi}\sigma)^3} e^{-\frac{r^2}{2\sigma^2}} \tag{2.6}$$

Equation **??** can be thought of as the probability of a point at a particular distance from the center of the sphere of being drawn if you stay on a single line. But the quantity of content is a function of radius and increases proportionally. To account for this we will use equation **??** as the density of a sphere and calculate the mass of that sphere which is equivalent to calculating the probability of a 3-dimensional normal distribution of drawing a point at random in a sphere of a given radius.

For the mass equation we will use:

$$M(R) = \int_0^R density(r) \, dV \tag{2.7}$$

In our case, $V = \frac{4}{3}\pi r^3$ and $dV = 4\pi r^2 dr$. You can picture dV as being the surface area of the sphere multiplied by dr, resulting in a small volume proportional to the varying surface area of a sphere of growing radius. Which results in :

$$M(R) = 4\pi \int_0^R density(r) r^2 dr \tag{2.8}$$

Now we can use this equation to calculate the mass of a sphere with a Gaussian density:

$$M(R) = 4\pi \int_0^R \frac{1}{(\sqrt{2\pi}\sigma)^3} e^{-\frac{r^2}{2\sigma^2}} r^2 dr \tag{2.9}$$

As expected $M(\infty) = 1$, which is the sum of a Gaussian filter of infinite radius.

With this knowledge, we can adapt the normal distribution model to better suit our needs. Let the random variable X have an isotropic 3-dimensional normal distribution if it's cumulative distribution function is:

$$F(R) = P(X < R) = 4\pi \int_0^R \frac{1}{(\sqrt{2\pi}\sigma)^3} e^{-\frac{r^2}{2\sigma^2}} r^2 dr \tag{2.10}$$

We can generalize this for isotropic n-dimensional normal distributions:

$$F(R) = P(X < R) = \int_0^R \frac{1}{(\sqrt{2\pi}\sigma)^n} e^{-\frac{r^2}{2\sigma^2}} dV \tag{2.11}$$

where V is the n-dimensional volume (content) of the hypersphere. In the n-dimensionnal case $P(X < R)$ is the probability that a random sample lies in the hypersphere with radius R. We will use the notation $X \sim \mathcal{IN}_n(\sigma)$ to represent an isotropic multivariate normal distribution where n is the number of dimensions. We calculated the isotropic multivariate normal CDF for $X \sim \mathcal{IN}_n(\sigma)$:

$$P(X_n < r) = \frac{\gamma(\frac{n}{2}, \frac{r^2}{2\sigma^2})}{\Gamma(\frac{n}{2})} \tag{2.12}$$

Where Γ is the Gamma function, a generalization of the factorial function to real numbers. Details for reaching this equation are given in the annex II as well as some interesting properties of multivariate isotropic normal distributions.

We can easily calculate the centre's R_k for any possible size of 3D keypoints with equation 2.10, which can be interpreted as the sum of a Gaussian filter's coefficients as a function of radius. Equation 2.13 is an example for a 3D SIFT-Rank keypoint, i.e. a sphere with a 2σ radius. Let

$X \sim \mathcal{IN}_3(\sigma)$, then:

$$P(X < 2\sigma) = 4\pi \int_0^{2\sigma} \frac{1}{(\sqrt{2\pi}\sigma)^3} e^{-\frac{r^2}{2\sigma^2}} r^2 dr \approx 0.738 \tag{2.13}$$

In other words, if you apply a Gaussian filter with variance σ^2, centered on a sphere with radius 2σ, the sum of all coefficients of the Gaussian kernel in the sphere is 0.738 compared to the total sum of 1 for a Gaussian filter. $P(X < r)|X \sim \mathcal{IN}_n(\sigma)$ for different n and r are shown in table 2.3 and figure 2.3 for various configurations. The original 2D SIFT implementation of Lowe uses a radius of σ and the 3D SIFT-Rank Toews and Wells a radius of 2σ. Other values are given for comparison.

Table 2.3 $P(X < r)$ if
$X \sim \mathcal{IN}_n(\sigma)$

Radius (r)	$P(X^* < r)$		
	n=3	n=2	n=1
σ	0.199	**0.393**	0.682
2σ	**0.738**	0.865	0.954
3σ	0.971	0.989	0.997
∞	1	1	1

*Isotropic multivariate normal distribution where $X \in \mathcal{R}_{\geq 0}$ is the distance from the average

The values in table 2.3 tell us the maximum R_k we can expect for a whole keypoint of various radii. It is interesting to note that R_k is lower for 3D keypoints than 2D keypoints at the same radius. $P(X < x)|X \sim \mathcal{IN}_n(\sigma)$ diminishes when **X** includes more dimensions as it is clearly shown in figure 2.3. As the number of dimension gets higher, the total amount of content (or n-dimensional volume) also increases, but $P(\mathbf{X} < \infty)$ is still 1. The probability of a point of being drawn at a fixed distance from the center of the filter diminishes across dimensions according to $\frac{1}{(\sqrt{2\pi}\sigma)^n}$ (from 2.6). It is also pertinent to note that content is not distributed in the same way for hyperspheres of different dimensions. For example, the average euclidean distance (d_c) from any point to the centre is smaller for a circle than a sphere. As seen from the
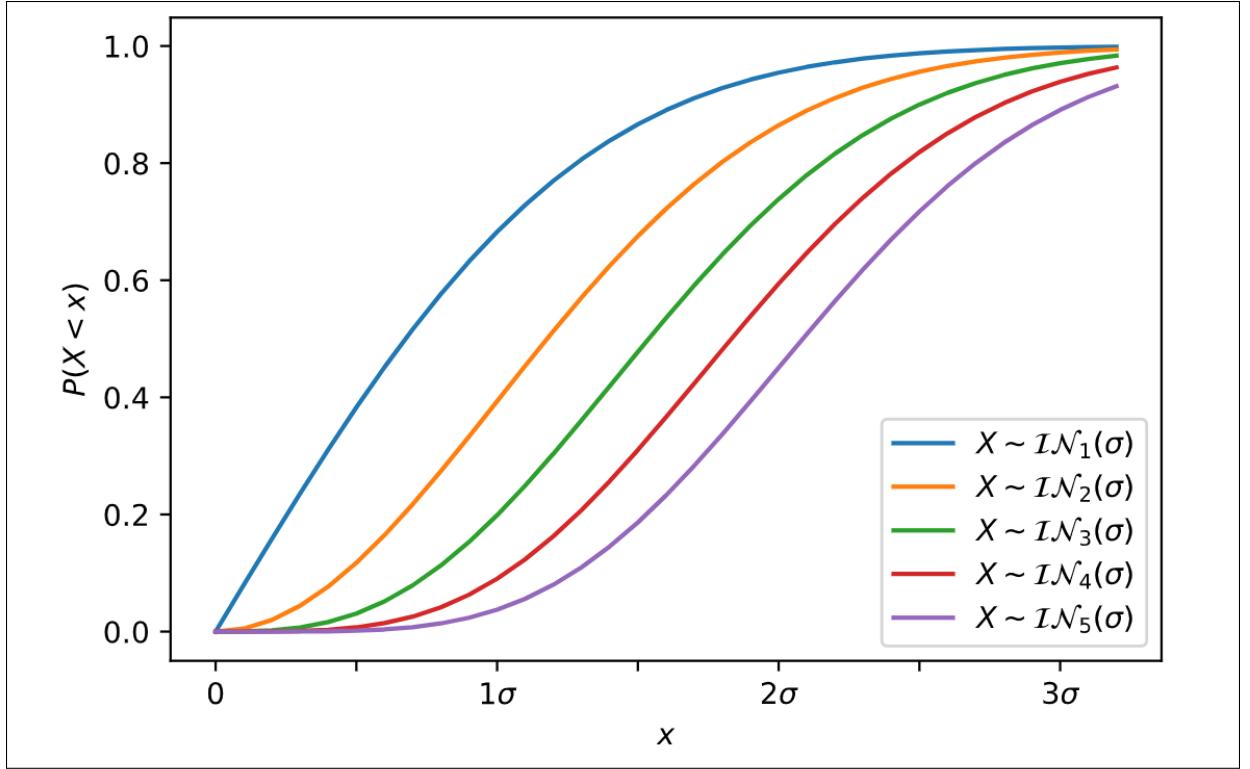
Figure 2.3 $P(X < x)|X \sim \mathcal{IN}_n(\sigma)$

Probability of a value being drawn from an isotropic multivariate Normal distribution of being within a radius x away from the mean.

equations 2.14, which represents the sum of the distance from all points to the center divided by the content (area or volume), i.e. area for a circle and volume for a sphere.

$$\frac{\sum d_c}{content} = \begin{cases} \frac{3R}{4} = \frac{4\pi \int_0^R r*r^2 dr}{\frac{4}{3}\pi*R^3}, & \text{for a sphere} \\ \frac{2R}{3} = \frac{2\pi \int_0^R r*r dr}{\pi*R^2}, & \text{for a circle} \end{cases} \tag{2.14}$$

In short, **a 2D Gaussian filter will encode information originating closer to it's center than a 3D filter**.

We provide a mathematical definition of R_k in the annex 1 for the whole keypoint (equation A I-1), but we will calculate R_k experimentally with an estimation algorithm presented in the next section.

2.2.2 R_k Estimation Algorithm

Our algorithm simulates an image of infinite size, with a sphere of radius 2σ populated by ones, convoluting with a Gaussian kernel with parameter σ and calculating the sum of all voxels in the sphere. We then divide that sum by the volume of the sphere to obtain R_k.

Here we demonstrate mathematically how the aforementioned algorithm results in R_k. Let I_1 be an image of size (c, c, c) filled with 0s except for a small sphere K of radius a filled with 1s, as shown in figure 2.4, where the grey represents 1s. Let I_2 be a copy of I_1 except the 0s and 1s are swapped. Let $G(\sigma) * I_1(x, y, z)$ be the Gaussian convolution of I_1. Let m_s be the sphere mask defined as:

$$m_s = \{[x, y, z] \in \Omega_{I_1} \mid (x - \frac{c}{2})^2 + (y - \frac{c}{2})^2 + (z - \frac{c}{2})^2 \leq a^2\} \tag{2.15}$$

where $a << c$.

$$I_1(\mathbf{p}) = \begin{cases} 1, & \text{if } \mathbf{p} \in m_s \\ 0, & \text{otherwise} \end{cases} \tag{2.16}$$

$$I_2(\mathbf{p}) = \begin{cases} 0, & \text{if } \mathbf{p} \in m_s \\ 1, & \text{otherwise} \end{cases} \tag{2.17}$$

Which results in $\sum M(I_1, m_s) = \sum I_1$ and $\sum M(I_2, m_s) = 0$.

Then:

$$g_1 = G(\sigma) * I_1(x, y, z) \tag{2.18}$$

$$g_2 = G(\sigma) * I_2(x, y, z) \tag{2.19}$$

$$I_3 = I_1 + I_2 \tag{2.20}$$

therefore, since a convolution is a linear map that has additivity:

$$g_3 = G(\sigma) * I_3(x, y, z) = g_1 + g_2 \tag{2.21}$$
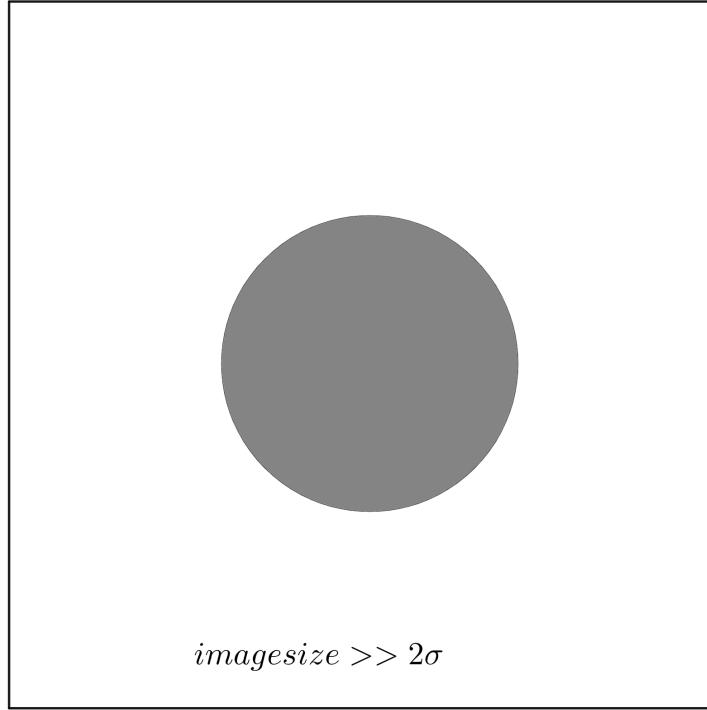
Figure 2.4   Keypoint mask - where grey is 1s and
white is 0s

Note that $\sum g_3 = \sum I_3$ since the Gaussian convolution does not affect the intensity sum in an image. Every voxel in g_1 contains a value that determines the ratio of this voxel's intensity that will contribute to the sum of intensities of the keypoint. We can define the ratio of data encoded in a keypoint from the keypoint volume as:

$$R_k = \frac{\sum M(g_1, m_s)}{\sum M(g_1, m_s) + \sum M(g_2, m_s)} \tag{2.22}$$

Which can be simplified to:

$$R_k = \frac{\sum M(g_1, m_s)}{\sum I_1} \tag{2.23}$$

Since:

$$\sum M(g_3, m_s) = \sum M(g_1, m_s) + \sum M(g_2, m_s) \text{ , because of 2.21} \tag{2.24}$$

and

$$\sum M(g_3, m_s) = \sum I_1 \tag{2.25}$$

since applying a convolution (with symmetric padding) with a Gaussian kernel to an image of infinite size full of 1s does not modify the image. So to calculate the ratio we only need $M(g_1, m_s)$ and I_1.

R_k provides good insight, but a more interesting value is the ratio of data encoded in the keypoint originating from the mask containing the keypoint: R_m. Obviously, the value of R_m depends on the location of the keypoint. When the keypoint's centre is on the mask's border, R_m tends toward 0.5 when the border tends toward a flat plane relatively to the size of the keypoint. Which is often the case when considering neuroimages if the brain is a lot bigger than the keypoint. We hypothesize it is true for most keypoints in a brain mask. Exceptions will occur mainly near area of the brain with low radius, the cerebellum for example. Results may vary for other masks, particularly if the mask is smaller than a brain mask. This hypothesis also depends on the resolution of the image used.

To avoid keypoints with too much non-mask content, we propose to reject keypoints that are too close to the border. To determine what is too close, we calculate R_m using the same procedure as for R_k but using a plane mask m_p instead of a sphere masks, as depicted in figure 2.5. The plane mask represents an object's mask, with the hypothesis that the mask is a lot bigger than the keypoint.

$$R_m = \frac{\sum M(g_1, m_p)}{\sum I_1} \tag{2.26}$$

This gives us the ratio of data encoded in a keypoint originating from a mask. We will calculate R_m for different values of $d\sigma$, simulating a keypoint with its centre $d\sigma$ away from the mask's border. This data will be useful in determining which keypoint we want to use from all the keypoints generated by the masked procedure depending on the percentage of non-mask content. It could also tell us at what point the noise in intensity-masked keypoints starts to be low enough as to be inconsequential.

It is also possible to calculate R_m for real keypoints. You can use the same procedure we just described, except the sphere is a real keypoint with radius 2σ and instead of a plane we use a real brain mask.
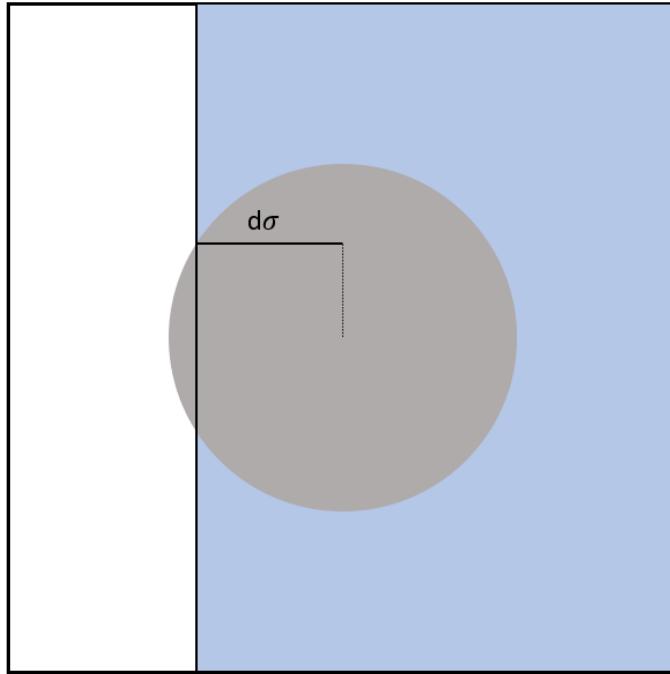
Figure 2.5   Plane mask - Grey is 1s
and everything else is 0s Blue is the plane mask.

2.3   Pairwise Keypoints Registration Between Masked and Skull-Stripped Images

This is the first experiment we did to compare intensity-masked keypoints and masked keypoints on brain images. It was a small-scale experiment designed to decide if it is worth investing more time in larger experiments comparing intensity-masked keypoints and masked ones. In this experiment we use 4 subjects from the HCP datasets at 1.25mm resolution, consisting of 2 pairs of monozygotic twins. The subjects' numbers can be seen in table 2.4. The aim of the experiment is to measure the usefulness of the different keypoint representations in matching twins together. Twins have similar brain patterns, so they tend to have a higher number of similar keypoints than non-twins. This small experiment is to see if intensity-masked keypoints have a lower number of matching keypoints than masked ones, as we expect. We test the following keypoints types:

- original keypoints: Keypoints extracted from the original image

- masked keypoints: Keypoints extracted from the original image and masked to keep only keypoints located in the brain mask
- intensity-masked keypoints: Keypoints extracted from a skull-stripped image.

Matches will be computed between all keypoint representations mentioned above of all 4 subjects. This means a total of (4 patients x 3 representations)^2 computed set of matches and $\binom{12}{2} = 66$ unique pairs.

The main objective of this experiment is to compare masked keypoints to intensity-masked keypoints. We know beforehand that we can expect the most matches with the original keypoints and we expect the masked keypoints to yield more matches then the intensity-masked keypoints.

Table 2.4   Twins data from HCP

pair #	member #	subject #
pair 1	#1	100307
	#2	255639
pair 2	#1	100408
	#2	705341

2.3.1   Matching with FeatMatchMultiple.exe

For each of the subjects, 3 different keypoint representations are extracted: intensity-masked keypoints, masked keypoints and keypoints extracted from the original image. The featMatch-Multiple.exe tool (available online [1]) is used to calculate the number of matches between keypoint representations. The program uses the appearance and location of keypoints from the 2 compared images to calculate the most likely geometrical transformation between the 2 images and returns matching keypoints according to transformation fit and appearance similarity. Here's an example of the command line used :

```
> ./featMatchMultiple.exe ./Original.key ./intensity-masked.key
```

---

[1]http://www.matthewtoews.com/fba/featExtract1.3.zip

One thing to note is that

```
./FeatMatchMultiple.exe File1 File2
```

is not equal to

```
./FeatMatchMultiple.exe File2 File1
```

Both commands achieve slightly different results which makes the matching algorithm noncommutative. A simplified explanation of the noncommutative property is provided by examining a simplified algorithm for matching elements between 2 sets:

Algorithm 2.1 ClosestNeighbours

1 **Input**: Set of vectors A and B

2 **Output**: Matches between A and B

3 Let $A = \{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n\}$ and $B = \{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_m\}$ where $\mathbf{a}_i, \mathbf{b}_j \in \mathbb{R}^k \mid k \in \mathbb{N}$.

4 **for** $a_i \in A$ **do**

5   **for** $b_j \in B$ **do**

6   Calculate $d(i, j) = ||\mathbf{a}_i - \mathbf{b}_j||^2$

7   **end for**

8      $m = argmin_j(d(i, j))$

9      $\mathbf{a}_i$'s closest match in B is $\mathbf{b}_m$

10      $matches = \{matches, \{\mathbf{a}_i, \mathbf{b}_m\}\}$

11 **end for**

This algorithm results in a list of closest neighbour $|A|$ long, whatever the length of B. If you called ClosestNeighbours(B,A) instead you would get a list of match $|B|$ long. Depending on the contents of both lists the result between both calls can be very similar or different. When

dealing with 2 lists of keypoints of a similar size the order of the call tends to matter less. Real algorithms are more complex. They may contain distance thresholds, impose a limit of 1 match per element and many other variations.

2.3.2 Visualization

This experiment aims at visualizing keypoint matches between intensity-masked keypoints and original keypoints. Instead of using different subjects as in the previous experiment, we use a single subject (patient 100206 from HCP) and find matches between the 2 different sets of keypoints. For this experiment, we don't use FeatMatchMultiple.exe. Instead, we use the algorithm detailed in algorithm 2.1 with a slight modification. To be considered a match, the closest neighbour needs to be closer than a hand-picked distance threshold. The threshold is chosen arbitrarily so that the highest possible euclidean distance between 2 matched keypoints is 4 voxels. The equation for the distance used is the Euclidean distance as shown in equation 2.27 where $\mathbf{a}_j$ is the appearance vector of keypoint f_j.

$$d(f_j, f_i) = ||\mathbf{a}_j - \mathbf{a}_i|| \tag{2.27}$$

The first step of the process is creating a list of matching keypoints between masked and intensity-masked keypoints. Matching keypoints will then be shown on a slice of the MRI. It's hard to visualize 3D data as a whole, hence the most representative slice is chosen manually. Keypoints are represented on the 3D model as hollow spheres. They appear on the slice as circles of varying radii depending on the scale of the keypoint and the distance from the centre of the keypoint to the plane of the chosen slice.

2.4 HCP Family Relationship Classification

This is a replication of the study of Chauvin *et al.* (2020) using masked keypoints instead of intensity-masked keypoints. The study is replicated trying to follow as much as possible the original protocol. We use the same tools as originally used for most of the task.

Two sets of data are used (subsection 2.1), the 0.7mm resolution HCP set and the 1.25 mm subsampled one. The 0.7mm resolution datasets was used in the original study and the 1.25 mm version is used to evaluate the impact of resolution on the differences between masked and intensity-masked representations.

The first step is calculating pairwise matching votes, which is a function of feature matches. It is done using the tool made available online[2] by the author of the original study. With matching votes and the features count for each image you can calculate the Jaccard coefficient for each pair of images. We do it using the same MATLAB script originally used. Finally, we calculate the AUCs of the ROC curve by varying the Jaccard threshold to classify pairs.

2.5 Feature Scale Test

In an additional investigation, we test the impact of the scale of keypoints on family indexing. Our hypothesis is that signature matching between family members is dependant on keypoint scale. This experiment is designed to test this hypothesis.

Masked keypoints of the HCP are used at 0.7 mm resolution. Keypoint files for each subject are separated into subsets based on scale. Those subsets are designed to include the most populated scale ranges as well as some combinations of them. The different possible scales can be obtained with the equation 1.21, the scale equation of SIFT keypoints. But in practice, scale values vary slightly from this equation because the scale is calculated using the final size of the patch represented by the keypoint descriptor. This is why we are using scale ranges instead of values.

---

[2]https://github.com/3dsift-rank/3DSIFT-Rank
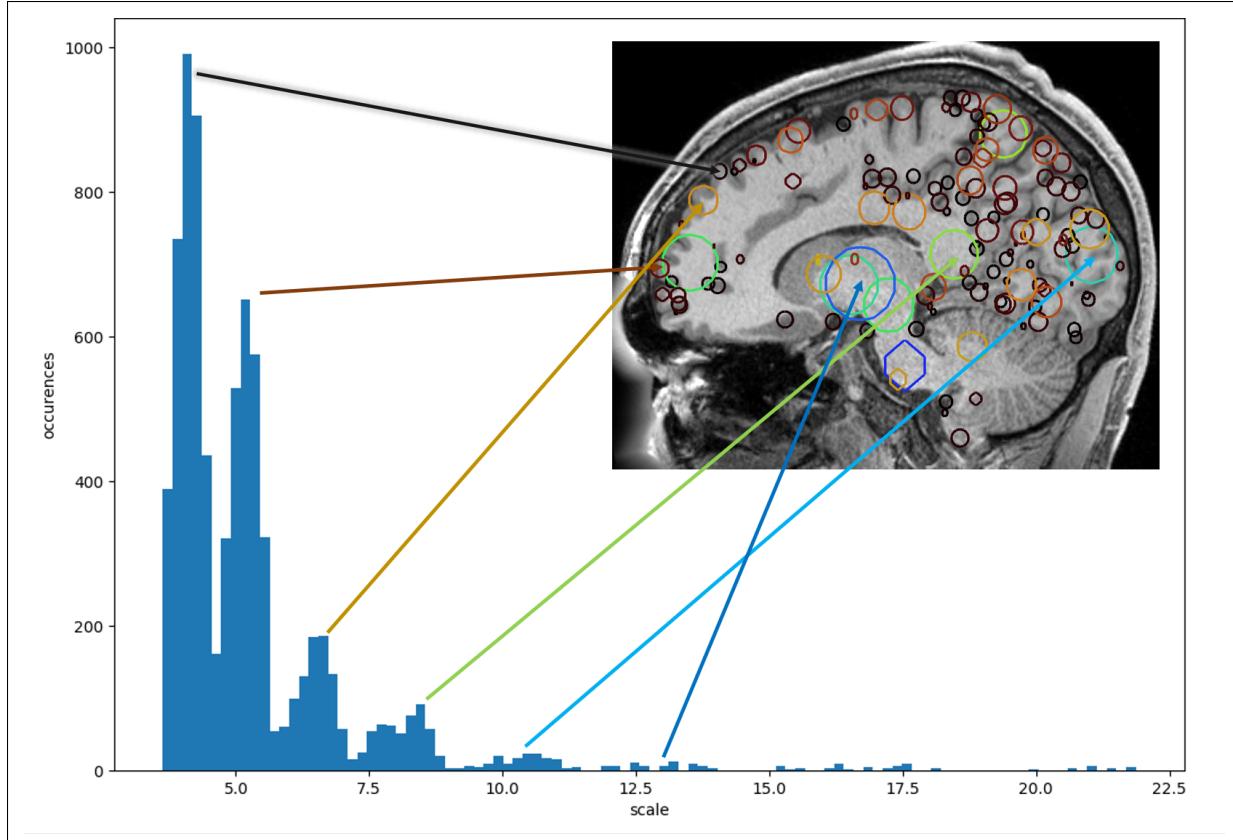
Figure 2.6   Scale histogram
Distribution of keypoint scale in a brain image

The scale cutoff demarcations are obtained by using KDE on the histogram shown in figure 2.6 to obtain a single continuous function and detecting minima. With the data separated by scale ranges, we replicate the family indexing procedure for each scale range and observe if a range performs notably better then the others.

<div align="center">

CHAPTER 3

RESULTS AND INTERPRETATION

</div>

In this chapter we present results from the experiments detailed in the previous chapter. We calculate R_k and R_m with our model, measure pairwise patient keypoint matches between different keypoint types, perform the HCP family relationship classification experiments with our new keypoint representations and provide visualisation of matching keypoints. Each experiment section includes a small discussion of the results, but results from different experiments will be compared in the next chapter.

3.1 Data From Non-Brain Tissue in Masked Keypoints

Table 3.1 details the R_k and R_m values for the most frequent scales, where R_k is the ratio of data encoded in a keypoint originating from the volume of that keypoint and R_m the ratio of data encoded in a keypoint originating from the volume of the mask with a simplified model. The left column's σ lists the Gaussian scale, which is also half the keypoint's radius. R_m has been calculated for 2 different plane distances. The distance σ leads to the plane going through part of the keypoint's sphere and 2σ is tangent to the keypoint's sphere. As a reference, we also calculate R_k for 2D keypoints as per Lowe's implementation, results are shown in Table 3.2. Table 3.1 and 3.2 contain the average values for the experiment using a Gaussian filter with parameter σ_i and σ_{i+1} to account for the use of both by the Difference-of-Gaussian operator.

First off, we can compare R_k values to Table 2.3. As in our hypothesis, the centre's R_k is higher than the keypoint's R_k, using 2σ radius for 3D and σ radius for 2D. We can also observe that R_k values vary slightly around the average while R_m values tend to go down with higher σs.

It is interesting to note that 3D keypoints' design compensates for the lower R_k inherent to 3D compared to 2D by having double the radius. As a result, 3D keypoints achieve a slightly higher R_k than 2D keypoints.

Table 3.1   Empirically calculated R_k and R_m for common
3D keypoint scales

| i | σ_i | R_k | R_m | |
			$distance = \sigma$	$distance = 2\sigma$
4	4.03	0.42	0.82	0.95
5	5.08	0.43	0.81	0.94
6	6.40	0.41	0.80	0.94
7	8.06	0.44	0.80	0.94
8	10.16	0.43	0.79	0.94
9	12.80	0.43	0.77	0.93
average		0.42 ± 0.01	0.80 ± 0.01	0.94 ± 0.00

Table 3.2   Empirically
calculated R_k for common 2D
keypoint scales

i	σ_i	R_k
4	4.03	0.32
5	5.08	0.33
6	6.40	0.30
7	8.06	0.32
8	10.16	0.32
9	12.80	0.30
average		0.32 ± 0.01

3.1.1   Example from a single brain

We measured the real R_m for every keypoint in a single brain (patient 100206 from HCP), using masked and intensity-masked keypoints, relative to the distance to the border. The code to perform those measures is available online [1].

Figure 3.1 and 3.2 shows R_m vs. distance factor ($distance\ to\ border/\sigma$). In the previous experiment we used values of 1 and 2 for the distance factor d. At first glance we can see that both distributions follow a similar pattern. For both distribution R_m is markedly higher when d is greater than 1. In the case of intensity-masked keypoints there are some keypoints with a

---

[1]https://github.com/pEtienn/3D-SIFT-keypoints-utilities
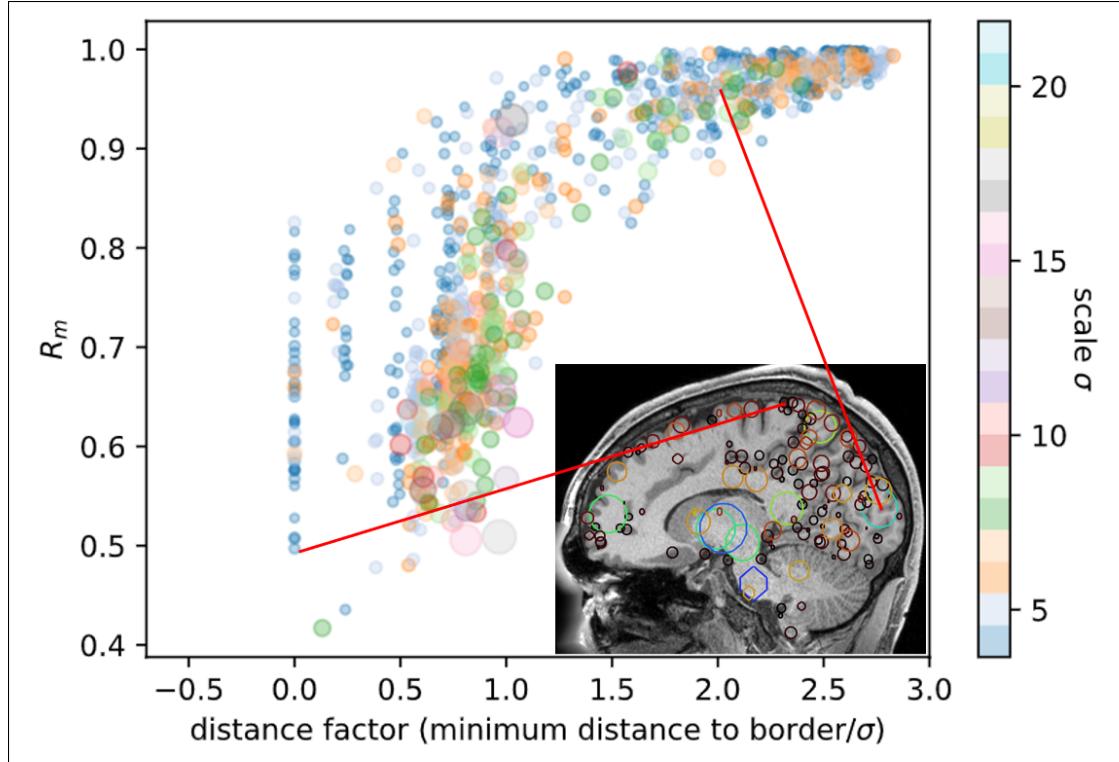
Figure 3.1 **R$_m$ vs. distance factor(d) for masked keypoints**. Contains only keypoints close to the brain mask border (to keep the focus on the border region). The color and size of points represent σ. The distance to the border is not calculated with sub-voxel precision, this results in some rounding artifacts for low distance values.

negative d. The few of those with a high R_m tend to be situated in concave regions. Statistics are presented in table 3.3 for an accurate comparison with the R_m model results presented in the previous section.

Minimum R_m values are similar to the ones calculated in the previous section (0.8 and 0.94), except for min $R_m \mid d >= 1$. The previous R_m calculation method was based on the hypothesis that a plane was a good representation of the brain border compared to a keypoint. In most case it is. But for masked keypoints with $d < 1$ there are a few outliers. They tend to be located near convex sections of the brain with a small curvature. If the plane model was accurate $|F| \mid R_m > 0.94$ would be equal to $|F| \mid d > 2$, but it is not the case. $|F| \mid R_m > 0.94$ is higher by an average of 87.5 keypoints for intensity-masked and masked keypoints.
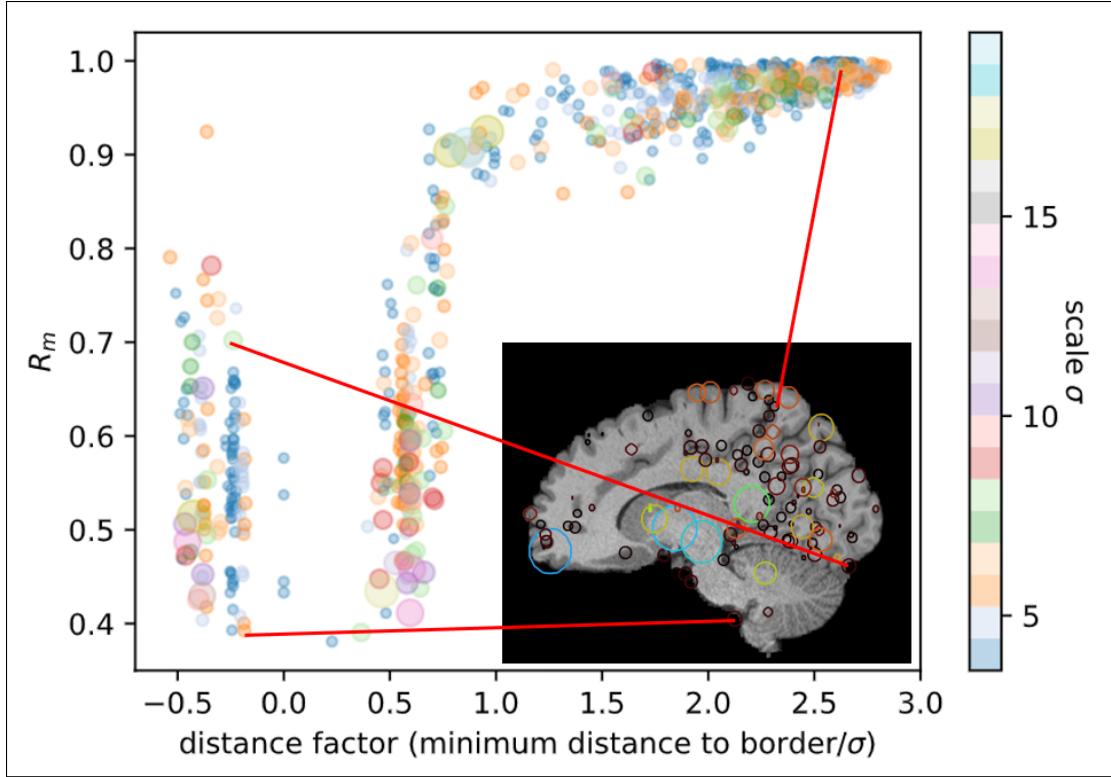
Figure 3.2 **R_m vs. distance factor(d) for intensity-masked keypoints**

Table 3.3 R_m characteristics for single patient

statistics	masked	intensity-masked
min R_m | $d >= 1$	0.567	0.812
min R_m | $d >= 2$	0.907	0.907
$|F|$ | $d > 0$	2214	1679
$|F|$ | $d > 1$	*1601*	*1496*
$|F|$ | $d > 2$	**1327**	**1324**
$|F|$ | $R_m > 0.80$	*1662*	*1530*
$|F|$ | $R_m > 0.94$	**1415**	**1411**
$|F|$	2214	1819

min R_m: minimum R_m value among R_m's off all keypoints $f \in F$.

$|F| \mid R_m > a$: number of keypoints in image F with R_m greater than a.

This experiment enabled us to verify how accurate our model is. While some differences exists and some keypoints will have a different R_m than the one predicted by the model, the model is accurate in general. We believe calculating R_m for any keypoint as a function of it's distance to

the brain border is the most efficient method for filtering keypoints according to R_m in the case of brain studies. Calculating R_m for each keypoint is time consuming, taking us a total of 4.5 hours for the 2214 masked keypoints. The technique could still be useful in cases other than brain study, with organs or objects of irregular shape where the keypoints radius is closer to the radius of the object's surface.

3.2   Pairwise Keypoints Registration Between Masked and Intensity-Masked Images

This section investigates the pairwise keypoint registration between different keypoint representations of 2 sets of twins detailed in section 2.3. A pairwise matching is performed for all of the 12 representations (4 patients x 3 type of keypoint representations), where a robust voting-style algorithm(Toews & Wells III, 2013) identifies keypoint-to-keypoint correspondences between each image pair that are inliers of a global similarity transform. Values in Figure 3.3 represent the number of matches found. Table 3.4 contains averages of number of matches for subsets that are particularly interesting.

For Figure 3.3, row keypoint representations are the first argument in the "featMatchMultiple.exe" command and columns are the second argument. The table is symmetric regarding compared representations. On each side of the top-left to bottom-right diagonal, the same pairs are computed, but with the parameters' order reversed in the command. This yields slightly different results.

Table 3.4   Matching between twins synthesis

Keypoint type 1	↔	Keypoint type 2	avg. number of matches	avg ratio of matches
masked	↔	masked	150.75	0.16
intensity-masked	↔	intensity-masked	78.75	0.26

Table 3.4 contains the most important data from figure 3.3. The left side contains matched keypoint representations and the right contains the average number of matches and the average match ratio ($\frac{number\_of\_matches}{max\_number\_of\_matches}$) for matched representations of the twins. With 150.75 average matches there is about twice the number of matches for masked keypoints vs. intensity-

twins		pair 1: twin 1			pair 2: twin 1			pair 1: twin 2			pair 2: twin 2		
	keys	ori	m	i	ori	m	i	ori	m	i	ori	m	i
p1:t1	original (ori)	803	767	152	73	73	36	148	142	52	93	91	40
	masked (m)	766	766	152	73	73	36	134	**137**	52	91	90	40
	intensity-masked (i)	149	149	233	26	26	38	51	52	**72**	37	37	45
p2:t1	original (ori)	73	73	37	1115	1094	297	52	52	28	159	158	70
	masked (m)	73	73	37	1094	1094	297	52	52	28	157	**156**	70
	intensity-masked (i)	39	39	52	291	291	403	33	33	48	62	62	**89**
p1:t2	original (ori)	148	140	62	64	64	35	1008	967	182	74	73	41
	masked (m)	142	**140**	61	63	63	35	966	966	181	73	73	40
	intensity-masked (i)	47	48	**69**	22	22	39	153	154	250	34	34	47
p2:t2	original (ori)	88	88	50	172	171	78	76	77	52	1166	1142	291
	masked (m)	87	87	50	171	**170**	78	73	74	52	1142	1142	291
	intensity-masked (i)	34	34	49	59	59	**85**	39	37	56	272	272	394

Figure 3.3   Keypoint matching between twins - Number of matches
The number of keypoints per image are in yellow and matches between different twins of pair 1 and 2 are in green and blue. Values used for table 3.4 are bold and underlined.

masked keypoints which can lead to better matching. But, the ratio is lower for matching between masked images. This experiment shows that using masked keypoints results in more matches found and that using masked keypoints might yield better results in certain applications.

3.3   HCP Family Relationship Classification

We perform experiments to measure the pairwise similarity between 1010 subjects from the HCP dataset using the Jaccard overlap score introduced by Chauvin *et al.* (2020), following the same method. In the original article, intensity-masked keypoints at 0.7mm resolution were used. We compare the following keypoint representations: original, masked, and intensity-masked at both 0.7mm and 1.25mm resolutions. The figure 3.4 shows ROC curves for the masked and intensity-masked representations at 0.7mm. Though the area under the curve (AUC) is similarly very high for Monozygotic (MZ) cases using masked and intensity-masked points, a higher AUC is observed in the case of Full-Siblings (FS) and Dizygotic (DZ) using the masked points when compared to the intensity-masked points. Unlike previous work, the proposed masked method also leads to statistically significant differences between DZ and FS brain similarity at a 1.25mm

resolution. This may be because we have never been able to observe cortical morphology in this amount of detail, due to skull-stripping noise.

Relationship, Method	AUC
MZ masked	0.999
MZ intensity-masked	0.999
DZ masked	**0.926**
DZ intensity-masked	0.909
FS masked	**0.889**
FS intensity-masked	0.865
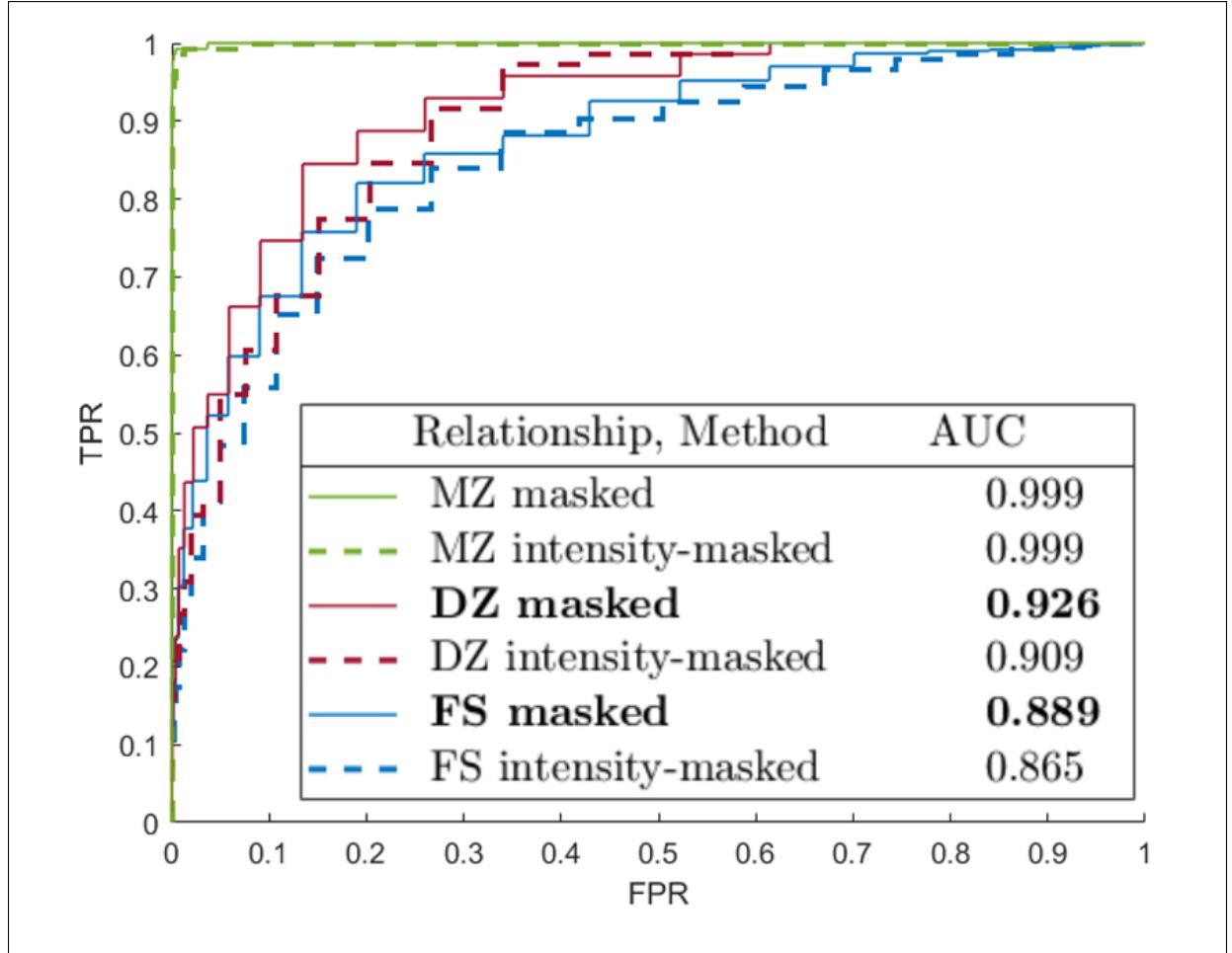
Figure 3.4   ROC curves

Table 3.5 compares relationship classifications using original, masked, and intensity-masked keypoints at different resolutions. Using masked keypoints results in higher AUC than intensity-masked keypoints for any relationship at both resolutions. The increase in AUC is amplified at a lower resolution, because a higher fraction of the intensity-masked keypoints are affected by the brain mask.

In this experiment, masked representations always perform as well or better then intensity-masked representations. We think it is due to a combination of a higher number of keypoints and

Table 3.5   AUC values for different keypoint representations and resolutions. Masked keypoints are a subset of original keypoints, and thus have lower AUC.

	keypoints	FS	DZ	MZ	# keypoints
	intensity-masked	0.865	0.909	**0.999**	1468 ± 189
0.7 mm	masked (ours)	**0.889**	**0.926**	**0.999**	1662 ± 241
	original	0.931	0.970	**0.999**	2102 ± 277
	intensity-masked	0.824	0.851	0.991	180 ± 34
1.25 mm	masked (ours)	**0.858**	**0.905**	**0.998**	253 ± 54
	original	0.889	0.950	**0.998**	334 ± 60

that none of those keypoints suffer from border noise. We also saw that original keypoints perform better than masked ones, implying that non-brain tissues are useful in predicting family relationship. But in case of studies interested solely by the brain, those tissues should not be included. But, as we saw before, non-brain tissue information is partly included in the masked keypoints we used. In the case of some keypoints, R_m goes as low as 50%. So the usage of masked keypoints is not without compromise.

3.3.1   Ignoring Keypoints Close to the Mask's Border: Eroded Brain Mask

In this section we repeat the family relationship classification experiment incorporating our results about data originating from non-brain tissue. Our experiment calculating R_m has shown that if we do not consider any keypoint with a centre at a distance of σ from the mask's border, the minimum R_m is around 0.8. This goes up to 0.94 when the distance is 2σ. In the case of intensity-masked keypoints, encoding data from outside the brain leads to noisy features that, we believe, are highly correlated to the skull-stripping process. We think it might affect poorly the results of many experiments. In the case of masked keypoints, considering data from outside the brain leads to non-brain information being encoded in the keypoint, which is not always a negative. For example, we demonstrated better results with the whole head than with only the brain during the family relationship classification experiment. But, considering non-brain tissue can be a problem for studies that want to focus solely on the brain.

To avoid those problems, we propose rejecting features too close to the brain mask's border using a brain mask eroded by a spherical element of radius $d\sigma$, named $m_{d\sigma}$. Where d is called the **distance factor**. Our resulting keypoint image, named $F_{b-d\sigma}$, contains only keypoints with centres $d\sigma$ or more away from the mask's border. The model can be seen in figure 3.5.

| (a) Erroded brain masks | (b) Erroded masks model |
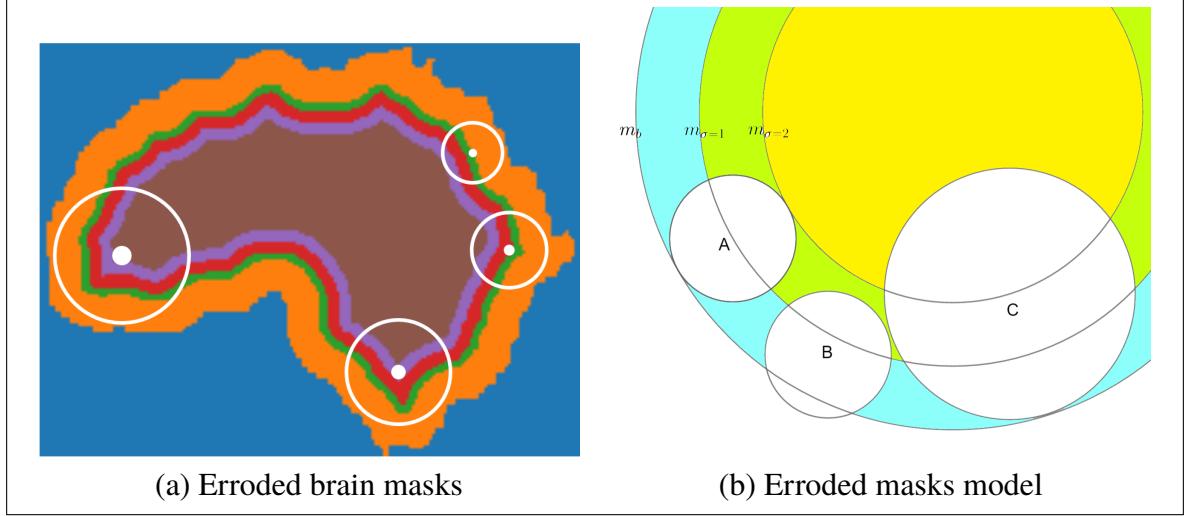
Figure 3.5 **Eroded masks** - Left image is an example (modified for visibility) of erroded masks with distance factor of 2 applied on a brain and keypoints filtered by the different masks. Right image represents the general model. m_b (blue) represents the brain mask and $m_{\sigma=1}$(green) represents a brain mask eroded 2σ inward. Circles A, B, and C represent keypoints with radii 2,2,4 of a scale σ equal to 1,1 and 2. Keypoints A and C are on the border of mask $m_{\sigma=1}$ and $m_{\sigma=2}$. Keypoint B is outside of mask $m_{\sigma=1}$ and is excluded from $F_{b-2\sigma}$, contrary to A and C.

Let m_b be the brain mask and $e_s(d\sigma)$ be a spherical element with radius $d\sigma$. $F_{b-d\sigma}$ can then be calculated with the following model:

$$e_s(d,\sigma) = \{[x, y, z] \in \mathbb{Z} \mid x^2 + y^2 + z^2 \leq (d\sigma)^2\} \tag{3.1}$$

$$m(d,\sigma) = m_b \ominus e_s(d,\sigma) \tag{3.2}$$

$$F(\sigma) = \{f \in F \mid f = \{.., \sigma, ...\}\} \tag{3.3}$$

$$F_{b-d\sigma} = \{f_\sigma \in M\big(F(\sigma), m(d,\sigma)\big) \; \forall \sigma \} \tag{3.4}$$

Brain masks eroded $d\sigma$ inward are tested on intensity-masked and masked keypoints at 0.7 and 1.25mm resolution. Table 3.6 contains the result of the experiment, where the distance is the minimum distance of any keypoint's centre to the brain mask's border. Figure 3.6 contains the data in the colum "AUC change" of the table 3.6 but in graph form. The implementation differs a bit from the equation above. As we detail in figure 2.6, there are a multitude of possible scale values for a keypoint. In our implementation, we create an erroded mask $m(d,\sigma)$ for values of $\sigma = 1.6x2^{\frac{i}{3}} \mid i = \{4, 5, ..., 8\}$. These are the most populated keypoint scales. For each keypoint, we use the mask with the closest scale value to the keypoint's σ. Some of the keypoints with bigger scales don't have a mask with a close-scale value. This is a compromise we chose to save computing time, since there are few keypoints with high scale values and performing erosion with large elements is time consuming. Some results of the previous family relationship classification experiment are shown in the table for reading ease. Keypoints extracted from skull-stripped images from the previous experiment don't have an established minimum distance from the mask (referred by none), sometimes they are even outside the mask. Masked keypoints from the previous experiment have a minimum distance of 0.

Table 3.6   AUC values with eroded brain mask - Border intensity-masked keypoints provide no increase in AUC.

keypoints		distance	FS	DZ	MZ	AUC change	# keypoints
		none	0.865	0.909	0.999	-	1468±189
	intensity-masked	σ	0.866	0.908	0.999	**0%**	1248±157
0.75 mm		2σ	0.861	0.898	0.999	**-0.85%**	1168±150
		0	0.889	0.926	0.999	-	1662 ± 241
	masked	σ	0.877	0.905	0.999	**-1.81%**	1897±187
		2σ	0.863	0.901	0.999	**-1.01%**	1186±152
		none	0.824	0.851	0.991	-	180±34
	intensity-masked	σ	0.819	0.867	0.989	**0.66%**	138±22
1.25 mm		2σ	0.819	0.860	0.989	**-0.42%**	121±19
		0	0.858	0.905	0.998	-	253±54
	masked	σ	0.849	0.898	0.997	**-0.91%**	229±50
		2σ	0.817	0.851	0.989	**-4.52%**	121±19

"AUC change" represents the AUC difference between the row above and the current row. It is an average of the FS and DZ AUC difference value. We decided to ignore MZ data since it's "capped" most of the time.
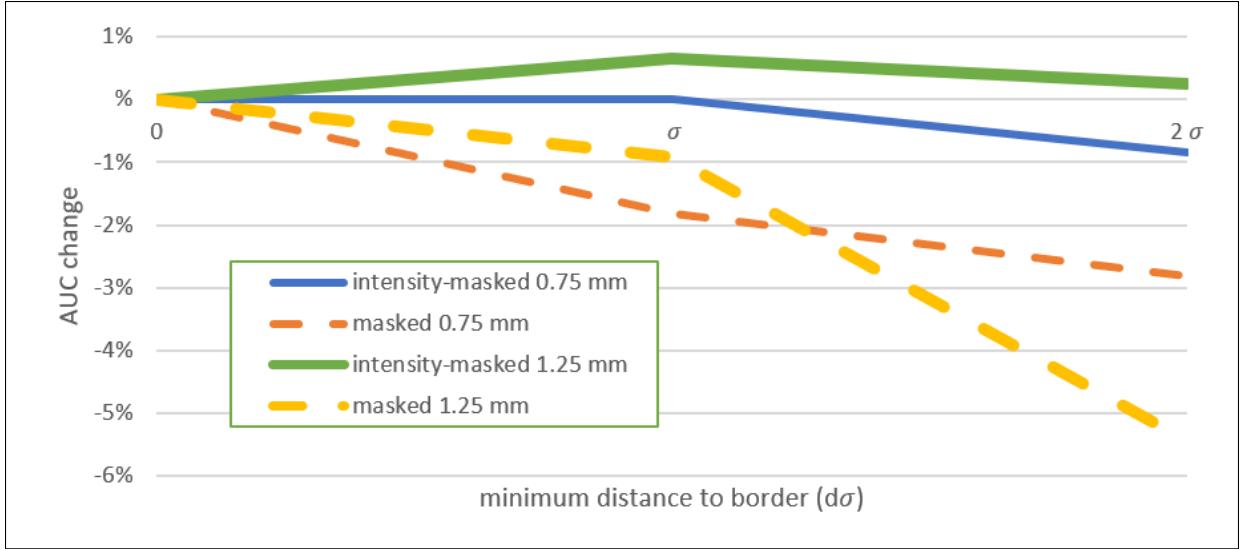
Figure 3.6   Total AUC change in percentage for FS and DZ average

We see in figure 3.6 that results for intensity-masked representations remain stable or even improve when using eroded masks at a distance of σ. At a 1.25 mm resolution, the AUC is still better when using $F_{b-2\sigma}$. This implies that **keypoints with centers near the border of intensity-masked images are not useful in the family classification task and can sometimes be detrimental**. This is not the case for masked keypoints, where using eroded masks results in a loss of classification performance. Using eroded brain masks reduces the number of keypoints available. In the case of intensity-masked images, the removed keypoints are noisy and don't contribute to classification, contrary to masked keypoints.

At a distance of 2σ, the results are very similar for masked and intensity-masked keypoints, including the number of keypoints. This confirms our previous experiment calculating a minimum R_m of 0.94 for keypoints at 2σ from the border. Using a brain mask eroded of 2σ on intensity-masked and masked images results in 2 sets of very similar keypoints. So similar in fact, that both representations offer the same performance in family relationship classification.

We mentioned that intensity-masked keypoints near the border are not useful for classification. At 0.75 mm resolution, they don't affect the classification rate while at 1.25 mm they reduce it by 0.66% (see section 3.4.1) The resolution is an important factor, another one is the classification

task we perform and how we match keypoints together. The feature matching algorithm used in family classification is designed to react strongly to close feature matches and ignore distant matches. So, if noisy features are random, it should not impact negatively the family classification. While noisy features don't seem to affect negatively this experiment (except for one case), we can't be sure it is a generalized behaviour and noisy features might be detrimental to some experiments.

3.4 Visualization

This section shows the results of the visualization experiment. The aim is to provide a visual explanation of the relation between intensity-masked keypoints and masked keypoints.

Figure 3.7 shows a saggittal brain MRI slice with original and intensity-masked keypoints. Matches and non-matches between the keypoints are represented by different coloured circles. Most matches between the two sets of keypoints are further inwards from the mask edge, and most unmatched keypoints are closer to the edge of the brain.
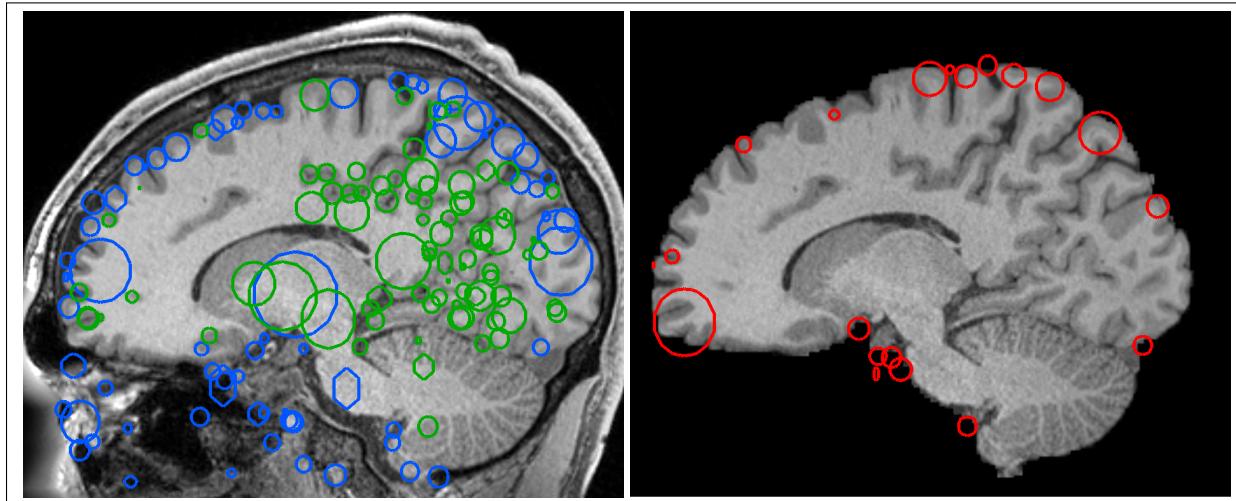

Figure 3.7   Visualizing keypoints (circles) in original (left) and skull-stripped (right) images. Keypoints present in both images are shown as (green, left), unique to original image as (blue, left) and unique to skull-stripped image (red, right). Keypoint masking generally identifies additional keypoints located primarily on the cortex in regions affected by boundary artifacts.

Across all images at 0.7mm resolution, 85% of the intensity-masked keypoints matched with the original keypoints of the same image, while at a 1.25 mm resolution, 75% of the intensity-masked keypoints matched with the original keypoints. This was measured across the subset of 1010 images we used from the HCP dataset.

3.4.1 Modelling Non-Matching intensity-masked keypoints as a Sphere Surface

At first glance the percentage of matches scales with the volume of the brain mask. If we model the brain as a sphere and assume the number of intensity-masked keypoints that did not match are relative to the surface of the sphere, we can use this model to project the number of intensity-masked keypoints not matching at different resolutions. Using this model, we will predict that 86% of the intensity-masked keypoints will match at 0.7mm resolution using the 1.25mm data. This is a simple model representing our intuition that the border interference effect on classifiers is dependent on the size of the mask.

We show that the number of intensity-masked features not matching original features fits our hypothesized sphere based model. In our experiment 85% of skull-stripped features matched with original features at 0.7mm, and 75% matched at 1.25mm. We now try to predict the difference when going from 1.25mm to 0.7mm resolution, the voxel size ratio is 1.78. Equation set 3.5 represents the area(A) and volume(V) of a sphere as a function of radius(r).

$$
\begin{aligned}
A(r) &= 4\pi r^2 \\
V(r) &= \frac{4}{3}\pi r^3
\end{aligned}
\tag{3.5}
$$

Equation set 3.6 represent our model to predict the number of features in 0.7mm resolution from 1.25mm resolution images.

$$r_{0.7mm} = 1.78r_{1.25mm}$$

$$A_{0.7} = 3.17A_{1.25} \tag{3.6}$$

$$V_{0.7} = 5.64V_{1.25}$$

From the experiment in the previous section, if we take a brain at 1.25mm resolution containing X skull-stripped features, $0.25X$ skull-stripped features won't match on average. We can consider those as the area around the sphere in our model. With our model we know that the area of $A_{0.7} = 3.17A_{1.25}$. Since we know that $A_{1.25} = 0.25$, we can deduce X: $3.17A_{1.25} = 3.17x0.25X = 0.7925X$. We also know that $V_{0.7} = 5.64X$. And since $\frac{0.7925}{5.64} = 0.14$, our model predicts that 14% of skull-stripped features won't match at 0.7mm resolution, compared to the real value of 15%. The model seems to fit well, but with only 2 data points, we can't be sure of how well it would perform in other situations. Other factors to consider would be the dataset used as well as the skull-stripping algorithm, since intensity-masked keypoints around the edge of the brain mask are likely to be strongly influenced by the skull-stripping algorithm. In conclusion, better resolutions lessen intensity-masked problems, since it's correlated to the area divided by the volume of a sphere, which gets lower with a bigger radius.

3.5   Feature Scale Test

The objective of this experiment is to test the following hypothesis: "keypoints' family relations information is scale dependent". To test this, we create 5 separate scale groups as seen in table 3.7 and measure AUC using images as feature containing only features in those scale ranges.

Note that the different scale groups yield different numbers of keypoints. The number of keypoints is inversely proportional to the scale value.

We can see in table 3.7 that the AUC doesn't seem to vary depending on the scale value. The scale range 4.72-5.96 yields the best AUC, but not by a big margin. It's also interesting to

note that even if the number of keypoints is lower at higher scales, the AUC is not affected proportional to the number of keypoints. Two combined scale groups are also tested and generate higher AUC then smaller groups, but obtain similar results between them.

Table 3.7   AUC by scale groups

| scale groups | AUC | | | \|group\| | scale mode | theoretical scale values | |
	FS	DZ	MZ			$1.6 * 2^{i/3}$	i
0-4.72	0.76	0.79	0.97	845	4.1	4	4
4.72-5.96	0.79	0.82	0.99	478	5.2	5.1	5
5.96-7.4	0.75	0.82	0.97	183	6.5	6.4	6
7.4-9	0.74	0.78	0.95	90	8.4	8.1	7
9-∞	0.77	0.82	0.95	66			
0-5.96	0.84	0.88	0.99	1323			
7.4-∞	0.85	0.89	0.99	156			
All	0.889	0.926	0.999	1662			

We conclude that the scale of the keypoint has no impact on family classification. Family brain resemblances are not more prominent at a particular scale. We also note that the number of keypoints between scale ranges does not affect the AUC. This could be due to a higher volume covered by keypoints in scale groups with higher scale average.

CHAPTER 4

DISCUSSION

4.1 Comparing Intensity-Masked to Masked Representation

The previous sections detailed many experiments comparing intensity-masked and masked keypoint representations. With those results, we can compare both techniques in order to better choose which one to use.

With our models, we have seen that intensity-masked keypoints near the border of the mask encode artificial data. We have been able to calculate what proportion of the keypoint information is composed of such artificial data theoretically and verify it on an actual brain. The theoretical method assumes the mask is a lot bigger than the keypoint radius and this hypothesis should hold for a variety of organs, as it did on the brain tested. The same is true for masked keypoints encoding data from outside the mask. In both those case, knowing the ratio of data encoded in the keypoint originating from the mask enables researcher to choose which keypoint is adequate for their study.

We also compared intensity-masked and masked keypoints in the specific context of a brain study. As mentioned before, those studies typically incorporate the skull-stripping as part of the image pre-processing. This ensures that no non-brain tissue is part of the study. Using masked keypoints results in the presence of some non-brain tissue data. But with the results obtained in section 3.1, the designer of a pre-processing procedure can choose himself what minimum R_m is suitable. For example, rejecting any keypoint 2σ or less away from the brain mask's border would insure a minimum R_m of around 0.94. At a distance of σ, it goes to 0.77. This is a compromise between the resulting number of keypoint and the minimum R_m. In experiments where using non-brain data is not critical we demonstrated notably better results using masked keypoints than intensity-masked keypoints, particularly at a lower resolution.

Intensity-masked keypoints don't contain any non-brain tissue data. But there are some drawbacks. Intensity-masked keypoints near the border are strongly influenced by the skull-stripping process and tend to be noisy. We demonstrated with the eroded mask experiment that those keypoints don't provide much useful information in the context of family classification and can lead to worse classifications in some cases. There's also less intensity-masked keypoints than masked ones, particularly at a resolution of 1.25mm. Some of those disadvantages can be worked around. Most of the border keypoints can be avoided by rejecting any keypoints 1σ or less away from the brain's border, using eroded masks. The smaller number of keypoints can't be avoided, but using a better image resolution will lessen the problem.

CONCLUSION AND RECOMMENDATIONS

5.1 Recommended Keypoint Representation for Brain Studies

The objective of this thesis was to determine the best way to use keypoints with brain images. **We always recommend using masked keypoints**. It is fast to compute and always offer better performance than intensity-masked keypoints. If avoiding non-mask data is important, we suggest using an eroded mask on original keypoints. The minimum ratio of non-mask data encoded in a keypoint can be consulted in figure 3.1. If necessary, those ratios can be computed for every keypoint in a real organ with equation 2.23. Code is also available on our repository to perform this task [1] .

When using intensity-masked representations, we suggest using an eroded mask with a distance of σ. This should avoid most of the noisy keypoints while not lowering performance.

Those recommendations are particularly effective with images at low resolutions, 1.25mm in our tests. At better resolutions, the difference between masked and intensity-masked representations becomes less noticeable.

In our comparison we used the family relationship classification experiment with brain images as a performance metric. Different experiments might perform differently with each representation, but this does not affect our analysis of R_k ($\frac{\text{encoded data from keypoint volume}}{\text{all encoded data in keypoint}}$) and of the theoretical R_m. These ratios can be good tools to judge how useful a keypoint is in a new situation, even if it's not extracted in a brain.

---

[1]https://github.com/pEtienn/3D-SIFT-keypoints-utilities

5.2 Future Works

We investigated artifacts in keypoint as a result of intensity masking. Keypoint technology is not widely used compared to CNNs. It would be interesting to examine the impact of intensity masking on CNNs instead, conclusions could potentially be far reaching. Although, finding a solution equivalent to masked keypoint seems harder for CNNs.

The mathematical properties of the multivariate normal distribution we demonstrated in appendix II could contribute to a better understanding of the scale-space. While the properties shown were mostly already known mathematically, sometimes in a different form, some of them are not common knowledge in the computer vision field. The CDF gives us a better idea of what information a point contains after Gaussian filtering. The variance of the distance to the center of the filter gives us a good tool to compare filter's variance at different dimensions. The univariate equation of the normal distribution gives a simpler tool to represent a system that can get complicated in n-dimensions. We made use of the CDF in this thesis and we believe that other properties we found could find some uses as well.

In closing, we think our work will provide a simple procedure for better using keypoints in medical image studies. We also provided a mathematical model to contribute to a better understanding of salient features and the data they encode.

R_K EXPERIMENT

1. Calculating R_K

We provide a mathematical definition of R_k for the whole keypoint (A I-1). Let B_k be the sum of all Gaussian filter with their centre in the keypoint sphere S_k, then:

$$B_K = 4\pi \int_0^{2\sigma} 4\pi \int_0^{\infty} G(r_g : \sigma) \, r_g^2 \, dr_g \, r_k^2 \, dr_k = \frac{4}{3}\pi(2\sigma)^3 \qquad \text{(A I-1)}$$

$$G(r_g : \sigma) = \frac{1}{(\sqrt{2\pi}\sigma)^3} e^{-\frac{r_g^2}{2\sigma^2}} \qquad \text{(A I-2)}$$


(a) variables   (b) intersection
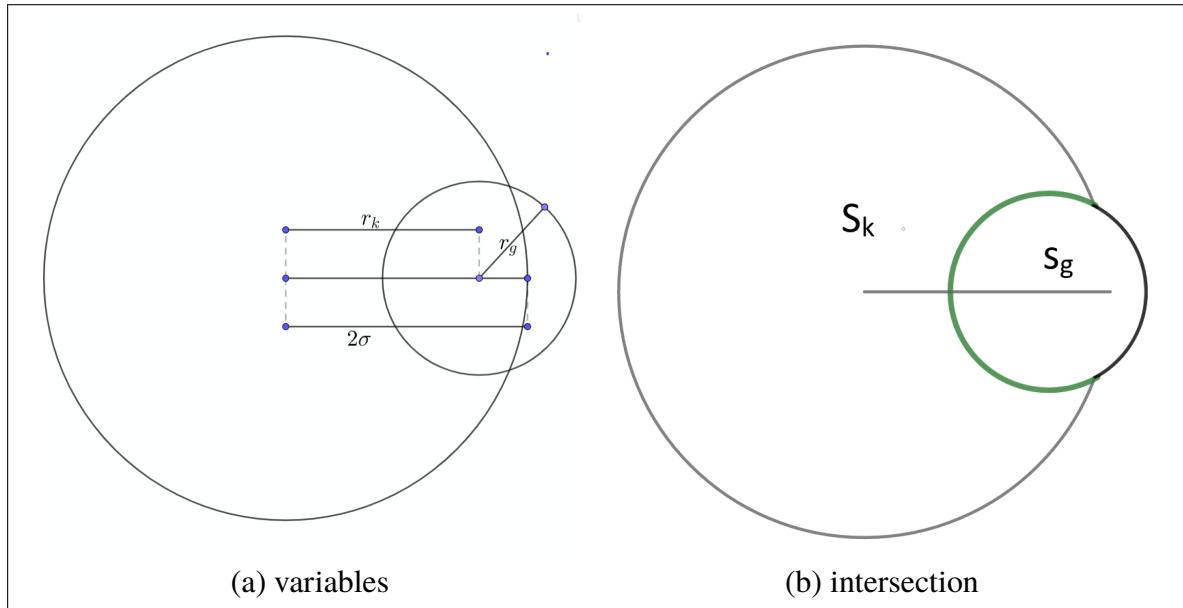
Figure-A I-1   Illustrating equation A I-1

where $G(r_G : \sigma)$ is the Gaussian 3D kernel as a function of distance and the keypoint has a radius of 2σ. Note that B_k is equal to the volume of the sphere, since Gaussian filters have a sum of 1 over infinity.

To obtain the proportion of this sum generated from data inside the keypoint, we calculate the intersection between the Gaussian sphere shell and the keypoint sphere: $s_g \cap S_k$. It is drawn in green in figure I-1 (b), where S_k is the keypoint sphere and s_g is a Gaussian sphere shell that is a function of r_g and r_k written as $s_g(r_g, r_k)$. $\frac{|S_k \cap S_g(r_g, r_k)|}{|S_g(r_g, r_k)|}$ gives us the ratio of mass inside the keypoint sphere contributing to a point r_k away from the keypoint sphere's centre and with a Gaussian weight of $G(r_g : \sigma)$.

$$Data \in K = 4\pi \int_0^{2\sigma} 4\pi \int_0^{\infty} \frac{|K \cap s_g(r_g, r_k)|}{|s_g(r_g, r_k)|} * G(r_g : \sigma) \, r_g^2 \, dr_g \, r_k^2 \, dr_k \qquad \text{(A I-3)}$$

$$R_k = \frac{Data \in K}{B_K} \qquad \text{(A I-4)}$$

The equation of the intersection between 2 spheres is known[1], but is complex enough as to make the computing of the integral impractical. Some simplifications can be made since our case is narrower then the general case.

---

[1]https://mathworld.wolfram.com/Sphere-SphereIntersection.html

APPENDIX II

ISOTROPIC MULTIVARIATE NORMAL PROPERTIES

We analyse isotropic multivariate normal distribution by transforming it to a univariate function, the only variable being the distance to center of the distribution. We base our demonstration on an hypersphere with a isotropic normal density. We will calculate it's CDF, PDF and moments with this model.

1.   Cumulative density function

We will start with an isotropic distribution and then generalise to any mutltivariate normal distribution. The multivariate normal density function is

$$f(\mathbf{x} : \mu, \Sigma) = \frac{1}{\sqrt{|\Sigma|(2\pi)^n}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right) \qquad \text{(A II-1)}$$

Where $\mu \in \mathbb{R}^n$ is the mean, Σ is a square covariance matrix, n is the number of dimensions and $|\Sigma|$ is the determinant of the covariance matrix. To get an isotropic normal density, we reduce the covariance matrix to a diagonal matrix with value σ^2 on the diagonal. This leads to

$$\mathbf{x}^T \Sigma^{-1} \mathbf{x} = \frac{||\mathbf{x}||^2}{\sigma^2} \qquad \text{(A II-2)}$$

and

$$|\Sigma| = (\sigma^2)^n. \qquad \text{(A II-3)}$$

Using an average of $\mu = 0$ removes the term and does not restrict us since we want to calculate our c.d.f. from the center of the hypersphere. We then obtain:

$$f(\mathbf{x} : \mu, \sigma) = \frac{1}{(\sqrt{2\pi}\sigma)^n} e^{-||x||^2/2\sigma^2} \qquad \text{(A II-4)}$$

In equation A II-4, $\mathbf{x}$ is a vector, but since we only use it's norm we can use the norm as a parameter of the function without losing generality. This gives us an univariate function instead of a multivariate one.

$$f(r : \mu, \sigma) = \frac{1}{(\sqrt{2\pi}\sigma)^n} e^{-r^2/2\sigma^2} \tag{A II-5}$$

Where $||\mathbf{x}||^2 = r^2$.

Let X_n be a random variable following the isotropic normal density function from eq.A II-5. $P(X_n < r)$ is the probability that a sample drawn from X_n lies in the hypersphere with radius r. To calculate this we will construct an hypersphere with a gaussian density and calculate it's mass. For n-dimensional normal distributions, many points share the same probability, contrary to the 1-dimensional normal distribution. We use the hypersphere to count those points.

The mass of an hypersphere can be calculated with this formula:

$$M(r) = \int_0^r density(r) \, dV \tag{A II-6}$$

For the hypersphere density we will use eq.A II-5. Which results in the CDF:

$$F(r) = P(X_n < r) = \int_0^r \frac{1}{(\sqrt{2\pi}\sigma)^n} e^{-\frac{x^2}{2\sigma^2}} \, dV \tag{A II-7}$$

where V is the n-dimensional volume (content) of the hypersphere. Let S be the hyper-surface area of an hypersphere of unit radius. From the literature[1]:

$$S = \frac{2\pi^{n/2}}{\Gamma(\frac{n}{2})} \tag{A II-8}$$

and

$$V = \int_0^r S_n x^{n-1} dx = \frac{S_n r^n}{n}. \tag{A II-9}$$

---

[1] https://mathworld.wolfram.com/Hypersphere.html

Which we use to calculate dV:

$$V = \frac{2\pi^{n/2}r^n}{n\Gamma(\frac{n}{2})}$$

$$dV = \frac{V}{dr} = \frac{2\pi^{n/2}r^{n-1}}{\Gamma(\frac{n}{2})}$$

(A II-10)

We can now combine equation A II-7 and A II-10 to calculate the CDF:

$$P(X_n < r) = \int_0^r \frac{2\pi^{n/2}}{\Gamma(\frac{n}{2})} \frac{1}{(\sqrt{2\pi}\sigma)^n} e^{-x^2/2\sigma^2} x^{n-1} \; dx$$

(A II-11)

We move the terms we can out of the integral.

$$= \frac{2}{2^{n/2}\Gamma(\frac{n}{2})\sigma^n} \int_0^r e^{-x^2/2\sigma^2} x^{n-1} \; dx$$

(A II-12)

We will modify the terms of the integral in order to obtain the lower gamma function γ starting with a variable substitution:

$$\frac{x^2}{2\sigma^2} = y \rightarrow \begin{cases} \frac{x}{\sigma^2}dx = dy \\ x = \sqrt{2y\sigma^2} \end{cases}$$

(A II-13)

The upper limit of the integral goes from r to $\frac{r^2}{2\sigma^2}$.

$$= \frac{2\sigma^2}{2^{n/2}\Gamma(\frac{n}{2})\sigma^n} \int_0^r e^{-x^2/2\sigma^2} x^{n-2} \frac{x}{\sigma^2} \; dx = \frac{2\sigma^2}{2^{n/2}\Gamma(\frac{n}{2})\sigma^n} \int_0^{\frac{r^2}{2\sigma^2}} e^{-y}(\sqrt{y2\sigma^2})^{n-2} \; dy$$

(A II-14)

$$= \frac{2^{n/2}\sigma^n}{2^{n/2}\Gamma(\frac{n}{2})\sigma^n} \int_0^{\frac{r^2}{2\sigma^2}} e^{-y} y^{n/2-1} \; dy$$

(A II-15)

By definition of the γ function we obtain:

$$= \frac{1}{\Gamma(\frac{n}{2})} \, \gamma(\frac{n}{2}, \frac{r^2}{2\sigma^2}) \tag{A II-16}$$

$$P(X_n < r) = \frac{\gamma(\frac{n}{2}, \frac{r^2}{2\sigma^2})}{\Gamma(\frac{n}{2})} \tag{A II-17}$$

You can change variable r to $x\sigma$, where x is the mahalanobis distance:

$$P(X_n < x) = \frac{\gamma(\frac{n}{2}, \frac{x^2}{2})}{\Gamma(n/2)} \tag{A II-18}$$

Equation A II-18 is the CDF of the Chi-square distribution in a modified form. The Chi-square is the distribution of the sum of n squared standard normal distribution. X_n is similar. It is the radius of a normally distributed hypersphere. We can write it this way:

$$X_n = \sqrt{\sum_{i=1}^{n} X_i^2} \tag{A II-19}$$

, where X_i is the distance along each axis. By squaring it, we get the same distribution as a Chi-square with the same CDF:

$$P(X < x) = f(x) \mid X \geq 0 \tag{A II-20}$$

$$P(X^2 < x) = f(\sqrt{x}) \tag{A II-21}$$

$$P(X_n^2 < x) = \frac{\gamma(\frac{n}{2}, \frac{x}{2})}{\Gamma(n/2)} \tag{A II-22}$$

We can think of the Chi-square CDF as the probability that a randomly chosen point in a normaly distributed hypersphere is $\sqrt{x}$ or less away from the center.

2. Probability density function and moments

We will now look at other interesting properties of isotropic normal distributions. First, we can obtain the probability density function from equation A II-11 by deriving it.

$$f(x) = \frac{2\pi^{n/2}}{\Gamma(\frac{n}{2})} \frac{1}{(\sqrt{2\pi}\sigma)^n} e^{-x^2/2\sigma^2} x^{n-1} \tag{A II-23}$$

$$f(y\sigma) = \frac{2}{\Gamma(\frac{n}{2})2^{n/2}\sigma} e^{-y^2/2} y^{n-1} \tag{A II-24}$$

With the PDF we can calculate moments i of a random variable X_n following the isotropic mutltivariate normal distribution.

$$E[X_n^i] = \int_0^\infty x^i \frac{2}{2^{n/2}\Gamma(\frac{n}{2})\sigma^n} e^{-x^2/2\sigma^2} x^{n-1} \, dx \tag{A II-25}$$

$$= \frac{2}{2^{n/2}\Gamma(\frac{n}{2})\sigma^n} \int_0^\infty e^{-x^2/2\sigma^2} x^{n+i-1} \, dx \tag{A II-26}$$

We will modify the terms of the integral to obtain the Gamma function Γ starting with a variable substitution:

$$\frac{x^2}{2\sigma^2} = y \rightarrow \begin{cases} \frac{x}{\sigma^2} dx = dy \\ x = \sqrt{2y\sigma^2} \end{cases} \tag{A II-27}$$

$$= \frac{2}{2^{n/2}\Gamma(\frac{n}{2})\sigma^n} \int_0^\infty e^{-x^2/2\sigma^2} x^{n+i-2} \sigma^2 \frac{x}{\sigma^2} \, dx = \frac{2}{2^{n/2}\Gamma(\frac{n}{2})\sigma^n} \int_0^\infty e^{-y} (\sqrt{y2\sigma^2})^{n+i-2} \sigma^2 dy \tag{A II-28}$$

$$= \frac{2^{(n+i)/2}\sigma^{n+i}}{2^{n/2}\Gamma(\frac{n}{2})\sigma^n} \int_0^\infty e^{-y} y^{(n+i)/2-1} \, dy \tag{A II-29}$$

By definition of the Γ function we obtain:

$$E[X_n^i] = 2^{i/2} \sigma^i \frac{\Gamma(\frac{n+i}{2})}{\Gamma(\frac{n}{2})} \tag{A II-30}$$

Below are values of the Γ function related to our case:

$$\Gamma(\frac{1}{2}) = \sqrt{\pi} \tag{A II-31}$$

$$\Gamma(n + \frac{1}{2}) = \frac{1 \cdot 3 \cdot 5 \cdots (2n-1)}{2^n} \sqrt{\pi} \tag{A II-32}$$

The first 2 moments are:

$$E[X_n^1] = \sqrt{2} \sigma \frac{\Gamma(\frac{n}{2} + \frac{1}{2})}{\Gamma(\frac{n}{2})} \tag{A II-33}$$

$$E[X_n^2] = n\sigma^2 \tag{A II-34}$$

We can see that the variance for X_1 is σ^2, as is well known. This is not the case for distributions of higher dimensions. This is because we are looking at the variance of the distance to the center, while the parameter σ^2 of the normal multivariate isotropic distribution represents the variance along any axis.

in 2D for example, the variance on the x axis is σ^2 and so is the variance on the y axis. But the variance of the distance to the center of the distribution is $2\sigma^2$. In the case of an isotropic distribution the variance of the distance to the center seems more appropriate than the variance along an axis as a metric to analyse dispersion. It also enables comparison between distribution of different dimensions.

Table II-1 contains some values for $E[X_n^1]$:

Table-A II-1   Expected
values for different
dimensions

n	$E[X_n^1]$
1	0.7979σ
2	1.2533σ
3	1.5957σ

APPENDIX III

SKULL-STRIPPING AS KEYPOINTS

This section contains our work on skull-stripping brain image as keypoints. It was performed before the work presented in this thesis and was put on hold for lack of promising results. The technique is based on the KDE, which is presented below.

1. Kernel Density Estimation

The KDE is a method to estimate the probability density function of a random variable. We will use it in our segmentation algorithm. In figure III-1 you can see a visualization of the KDE for a 1D random variable x.

The KDE is similar to the histogram of a sample but generates a continuous function instead of a discrete one. To estimate the PDF at any point it uses near samples and gives a higher weight to closer ones. The kernel is used to specify the weight given to each sample. Different kernels can be used, for instance figure III-1 shows a Gaussian kernel. Other kernels are commonly used: Epanechnikov, triangular, uniform and others. Here is the equation defining the KDE:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} K_h(x - x_i) \qquad \text{(A III-1)}$$

where $\hat{f}$ is an estimator of the true PDF, n is the number of samples and K_h is a kernel function with a bandwith parameter, which is σ in the case of a Gaussian kernel. The KDE will be used later on to estimate the class of a keypoint.

2. Method

This section details the design of a keypoint segmentation algorithm to separate brain keypoints from non-brain keypoints. This method aims to be used with MRIs taken in less than ideal conditions. It uses salient keypoints to mitigate scale and rotation and noise variances. The
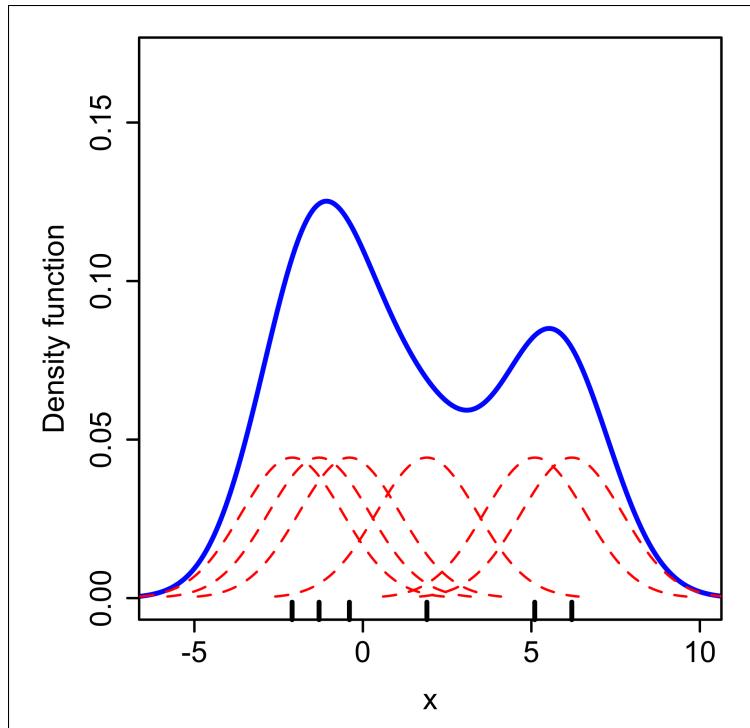
Figure-A III-1   KDE visualization - Gaussian Kernel
(Drleft, 2010)

The blue line represents the estimated PDF, the red lines
represent the kernels (Gaussian) generated by the short
vertical lines representing samples of the random variable

whole process pipeline uses only SIFT 3D keypoints, which could facilitate the sharing of data

between institutions for large-scale studies.

2.1 Database: OASIS-1

The OASIS-1 (Marcus, Wang, Parker, Csernansky, Morris & Buckner, 2007) dataset was used for segmentation experiments. It includes 416 subjects aged 18 to 96. T1-weighted MRI scans are acquired in a single session at a 1mm isotropic resolution. 100 subjects have been clinically diagnosed with Alzheimer's disease. All subjects are right-handed and the dataset contains men and women.

Table-A III-1 OASIS-1
demographic information

Image number	416
age	52 ± 25
male	160
female	256

FreeSurfer, a popular program performing skull-stripping, sometimes has trouble with images straying too far from the most common brain patterns. In Chauvin *et al.* (2020), 616 out of 8152 failed the pre-processing pipeline with FreeSurfer. Our skull-stripping algorithm could be useful in those cases.

The algorithm starts by classifying every keypoint f from an image $F = \{f_1, f_2, ..., f_n\}$ using KDE. It compares the keypoint f to a training set containing keypoints labelled as either brain or not brain with no patient tag, essentially 2 sets of features: brain and not brain. After this step, we perform spatial regularization to help classify keypoints with no close matches.

2.2 Training Set Generation

Keypoints extracted from the OASIS dataset are used in this experiment. The data is filtered with computer-generated brain masks and divided in 2 groups: brain and not brain. An example can be see in figure b for a single patient.

|	
(a) brain keypoints	(b) non-brain keypoints
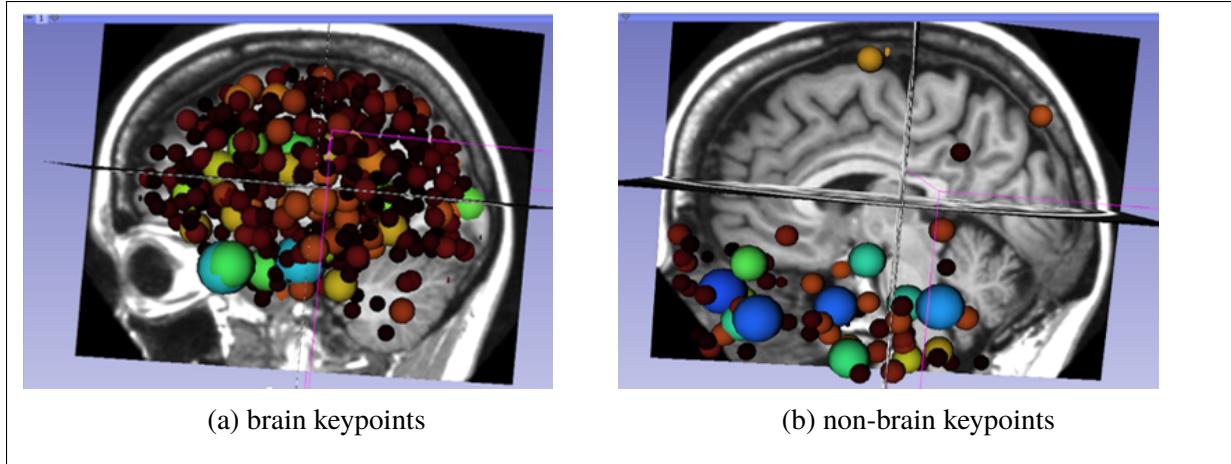
Figure-A III-2   Training set example from one patient. Note how the keypoints are distributed: most of the non-brain keypoints are situated below the brain near the eyes and jaw, with only some keypoints on the skull.

2.2.1   Kernel Density Estimation

Kernel Density Estimation is often used with salient keypoints. The appearance of 3D SIFT-Rank keypoints exists on a high-dimensional space, which makes it hard to know the true density. KDE is a useful tool to approximate a probability distribution in this context. Often the algorithm is used to predict a class for the whole image. In this article, we calculate a class probability for each keypoint separately by considering the similarity of appearance between each keypoint of the test image and all keypoints in our training set.

For now the position information of keypoints is ignored to simplify the method. By ignoring the spatial information, we avoid errors linked with registration. Typically keypoint matches are validated by comparing the descriptor distance between the closest neighbour and the second closest (image-to-image match) as described in Lowe (2004). In our case, bad matches get assigned a very low probability due their high Euclidean distance to the tested keypoint.

The first step of our brain extraction technique is to use KDE to give a probability for each keypoint on whether it belongs to the brain or non-brain. The objective is to estimate $p(c_k|f_i)$ where c_k is the class (brain or non-brain) and f_i is a keypoint from test images $F = \{f_1, f_2, ...f_n\}$

using our training set, which is a collection of keypoints $K = \{f_1, f_2, ..., f_l\}$ from m training images.

Using Bayes formula:

$$p(c_k|f_i) = \frac{p(f_i|c_k)p(c_k)}{p(f_i)} \tag{A III-2}$$

$$p(f_i|c_k) = \sum_j p(f_i f_j|c_k) \propto \sum_j p(a_i a_j|c_k) \tag{A III-3}$$

$$p(a_i a_j|c_k) = \frac{p(a_i a_j c_k)}{p(c_k)} = \frac{p(a_i|a_j c_k)p(a_j c_k)}{p(c_k)} \tag{A III-4}$$

$$p(c_k|f_i) = \sum_j \frac{p(a_i|a_j c_k)p(a_j c_k)}{p(c_k)} * \frac{p(c_k)}{p(f_i)} \tag{A III-5}$$

$$p(c_k|f_i) \propto \sum_j p(a_i|a_j c_k) \tag{A III-6}$$

where $p(a_i|a_j c_k) \propto e^{\frac{-(a_j - a_i)^2}{2\sigma^2}} |\sigma = d_i + 1$. Where d_i is the distance of the closest neighbour to f_i + 1.

We speed up the process by only using m keypoints from K. Since there's a maximum of one good match f_i per image from the training set, there's no more than m informative matches to consider. Some of those m aren't be good matches, but since the distance is high, the resulting impact on the probability is low. We use the KD tree algorithm (Muja & Lowe, 2014) to find the m closest matches.

The algorithm goes as follow:

Algorithm-A III-1 Kernel Density Estimation

---

1   **Input**: Training set, keypoints to classify F

2   **Output**: class probability of each keypoint c_k

3   Initialise 2 k-d tree T_k, each containing all keypoints from the training set of their respective class

4   **for** $c_k \in C \mid C = \{brain, notbrain\}$ **do**

5   **for** $f_i \in F$ **do**

6   Find the m closest neighbours in T_k

7   Use equation A III-6 to calculate $p(c_k|f_i)$ with $j = m$

8          $P(j, k) = p(c_k|f_i)$

9   **end for**

10   **end for**

11   To obtain the most probable class:

12   **for** $f_i \in F$ **do**

13       $l = argmax_k(P(j, k))$ The most probable class is c_l

14   **end for**

The figure III-3 is a visualization of the application of the KDE on the same brain that was presented in figure III-2. We can see that some of the skull keypoints are correctly labelled, but also that some brain keypoints have been incorrectly predicted, even if some of them are near the centre of the brain.

(a) predicted brain keypoints   (b) predicted non-brain keypoints
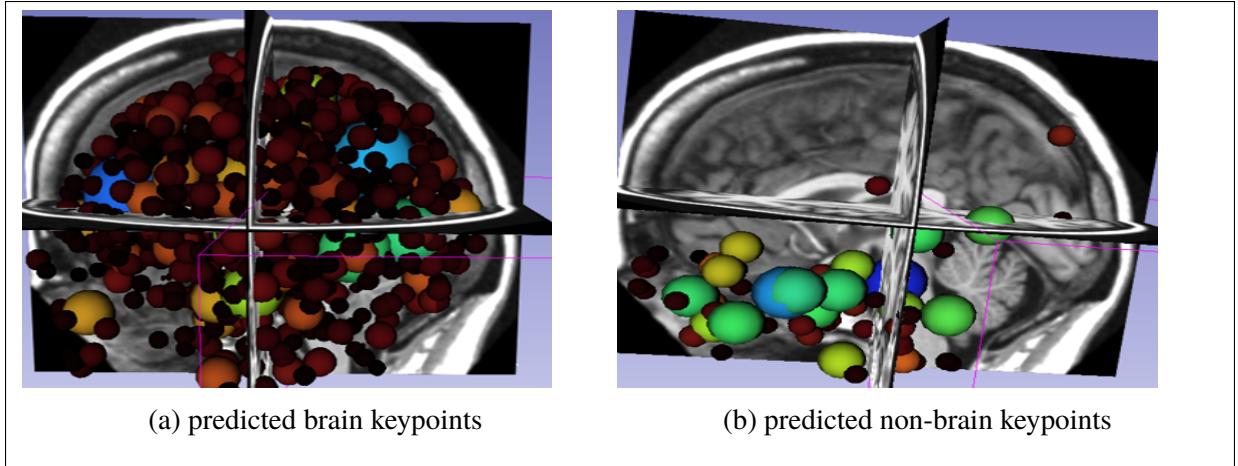
Figure-A III-3   Result of KDE for a single brain

2.2.2   Spatial Regularization of Segmentation Labels

This step aims at correcting some of the problems caused by the previous method. The general idea is that most keypoints were correctly predicted, and we can use those predictions to correct the wrong ones. Previous predictions will affect their neighbour based on the probability

estimated earlier following a Gaussian curve. The space regularization algorithm is presented in detail in III-2.

Algorithm-A III-2 Space regularization

---

1 **Input**: c_k and X for each keypoint

2 **Output**: probability map $\mathcal{M}$

3 Initialize a probability Map $\mathcal{M}$

4 $\mathcal{M}(c_k, \mathbf{p}) = 0 \ \forall \ \mathbf{p} \in \Omega_I$ and $c_k \in C$

5 $\mathcal{M} : \mathbb{N}^4 \rightarrow \mathbb{R}^1$

6 **for** $c_k \in C$ **do**

7   **for** $f_i \in F$ **do**

8   Create a cubic matrix E:

9 $E(\mathbf{p}) = i \mid \mathbf{p} \in \Omega_E, i \in \mathbb{R}$, where

        $\Omega_E = \{\mathbf{p} = [x, y, z] \in \mathbb{Z}^3 \mid -\sigma_i \leq x, y, z \leq \sigma_i\}$, where σ_i is the scale of f_i

10 $E(\mathbf{p}) = \begin{cases} P(j, k), & \text{If } \mathbf{p} = [0, 0, 0] \\ 0, & \text{otherwise} \end{cases}$

11 $G_E = G(\sigma_i) * E(x, y, z)$

12 $\mathcal{M}(c_k) = \mathcal{M}(c_k) + G_{padded}$

13 $G_{padded}(\mathbf{q}) = \begin{cases} G_E(\mathbf{p}), & \forall \ \mathbf{q} = \mathbf{p} + X_i \mid \mathbf{p} \in \Omega_E \text{ and } \mathbf{p} \in \Omega_I \\ 0, & \text{otherwise} \end{cases}$

14   **end for**

15 **end for**

16 To obtain the most probable class:

17 **for** $f_i \in F$ **do**

18 $l = argmax_k(M(c_k, X_i))$ c_l is the most probable class of f_i

19 **end for**

(a) predicted brain keypoints   (b) predicted non-brain keypoints
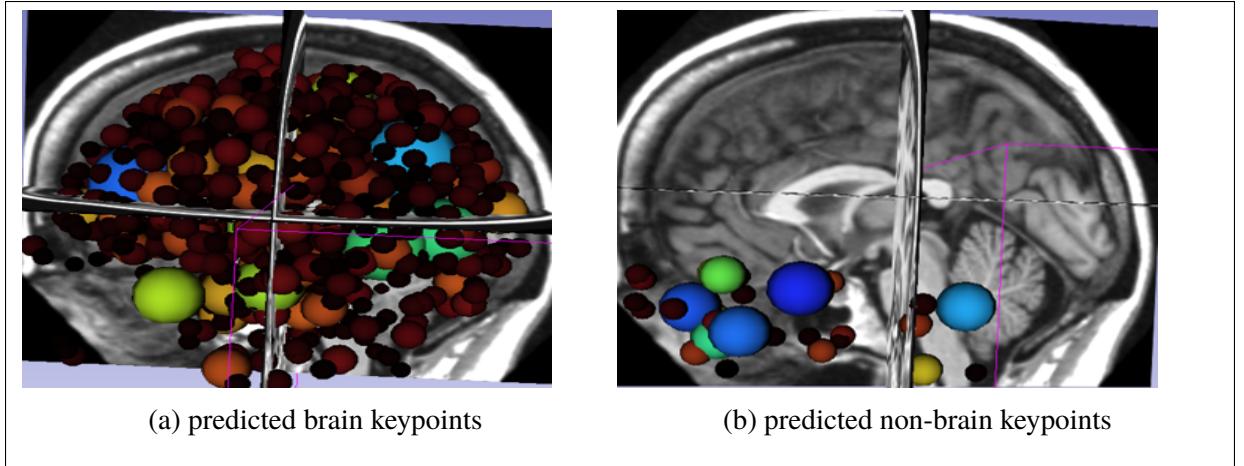
Figure-A III-4   Result of space normalization

The figure III-4 is a good example of the problems and advantages of this algorithm. The keypoints that were wrongfully classified as non-brain while in the middle of the brain are now classified as brain. Since all their neighbours were predicted as brain, the space regularization algorithm tends to classify them as brain. This also leads to some misclassifying. The keypoints on the skull that were previously predicted as non-brain are now predicted as brain. Since the keypoints on the skull are very sparse, we expect the space regularization technique to misclassify non-brain keypoints on the skull most of the time. But, in our dataset, most non-brain keypoints are situated below the brain, very few are on the skull.

One possible avenue of space normalization is the creation of brain masks. During the space normalization process, we create a probability map for each class. This could lead to the creation of a rough brain mask. The skull keypoints help to frame the brain mask, but are too sparse to frame it perfectly. Some other technique is needed to get a complete mask. We could potentially use the brain mask from the training set, but this would require to perform registration between the test images and training images.

3. Results

We test our segmentation algorithm on our training set by using the K-fold method with $k = 4$. This means our training set has 312 patients in it during our tests. We measure the DC for both brain and non-brain tissues. Results are presented in table III-2. The algorithm is tested first using only KDE, then using KDE and spatial regularization. Dice coefficient is calculated for brain tissue and non-brain tissue.

Table-A III-2   Dice similarity coefficient results on the test set

	brain DC	non-brain DC
KDE	0.955	0.594
KDE and regularization	0.960	0.615

Those results are comparable to existing techniques, but some key differences exist. First off, comparing the results of this technique with existing literature is ambiguous since we compare keypoint DC and voxel DC. They have a great degree of similarity, but should not be compared as the same metric. In this comparison article by Iglesias *et al.* (2011) of brain extraction techniques, the most popular techniques were tested on the OASIS dataset. The highest DC was obtained with a combination of FreeSurfer and GCUT-40 getting a DC of 94.1%. Another point to mention is that the technique is very dependent on the training set. Our tests were conducted on OASIS images using a OASIS training set. Worse results are expected when classifying images from a dataset not contained in the training set. This can be due to different scanning equipment, imaging protocol, patient demography, etc.

The technique is also quite fast at around 5 seconds per patient, compared to a 1-minute time for most techniques (Iglesias *et al.*, 2011).

4. Use of KDE Segmentation

The KDE segmentation proved to perform well in certain restricted conditions. It delivers fast and accurate brain keypoint segmentation, based on a simple probability model. It can skull-strip

brains as keypoints which, to our knowledge, is the first technique to achieve this. But it has some restrictions. KDE segmentation works only with keypoints, does not generate a volumetric mask and offers limited performance when classifying images from a dataset not present in the training set.

Future works on KDE segmentation should include testing segmentation on rotated images. It is a task in which this algorithm could shine compared to others. It could also be interesting to quantify the performance of the algorithm on images from datasets not included in the training set. We noted lower performance when testing it briefly, but did not investigate it in detail. Lastly, we think the space regularization can be improved. Right now, the space regularization performs really well in the centre of the brain, but fails near its edge.

Our segmentation algorithm, while having many drawbacks, could be useful in certain niche applications, particularly the sharing of brains between institutions as sets of keypoints.

BIBLIOGRAPHY

Azad, R., Fayjie, A. R., Kauffman, C., Ayed, I. B., Pedersoli, M. & Dolz, J. (2020). On the Texture Bias for Few-Shot CNN Segmentation.

Björkman, M., Bergström, N. & Kragic, D. (2014). Detecting, segmenting and tracking unknown objects using multi-label MRF inference. *Computer Vision and Image Understanding*, 118, 111–127.

Chauvin, L., Kumar, K., Wachinger, C., Vangel, M., de Guise, J., Desrosiers, C., Wells, W. & Toews, M. (2020). Neuroimage signature from salient keypoints is highly specific to individuals and shared by close relatives. *NeuroImage*, 204, 116208. doi: https://doi.org/10.1016/j.neuroimage.2019.116208.

Doshi, J., Erus, G., Ou, Y., Gaonkar, B. & Davatzikos, C. (2013). Multi-atlas skull-stripping. *Academic radiology*, 20(12), 1566–1576. doi: 10.1016/j.acra.2013.09.010.

Drleft. (2010). Comparison of 1D histogram and KDE.png. Consulted https://commons.wikimedia.org/wiki/File:Comparison_of_1D_histogram_and_KDE.png.

Elam, J. S. & Van Essen, D. (2013). Human Connectome Project. In Jaeger, D. & Jung, R. (Eds.), *Encyclopedia of Computational Neuroscience* (pp. 1–4). New York, NY: Springer New York. doi: 10.1007/978-1-4614-7320-6_592-1.

Eskildsen, S. F., Coupé, P., Fonov, V., Manjón, J. V., Leung, K. K., Guizard, N., Wassef, S. N., Østergaard, L. R. & Collins, D. L. (2012). BEaST: Brain extraction based on nonlocal segmentation technique. *NeuroImage*, 59(3), 2362 - 2373. doi: https://doi.org/10.1016/j.neuroimage.2011.09.012.

Fischl, B. (2012). FreeSurfer. *Neuroimage*, 62(2), 774–781.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A. & Brendel, W. (2019). ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. Consulted https://openreview.net/forum?id=Bygh9j09KX.

Gonzalez, R. C. & Woods, R. E. (2006). *Digital Image Processing (3rd Edition)*. USA: Prentice-Hall, Inc.

Iglesias, J. E., Liu, C.-Y., Thompson, P. M. & Tu, Z. (2011). Robust Brain Extraction Across Datasets and Comparison With Publicly Available Methods. 30, 1617-1634.

Kleesiek, J., Urban, G., Hubert, A., Schwarz, D., Maier-Hein, K., Bendszus, M. & Biller, A. (2016). Deep MRI brain extraction: A 3D convolutional neural network for skull stripping. *NeuroImage*, 129, 460 – 469. doi: https://doi.org/10.1016/j.neuroimage.2016.01.024.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F., Burges, C. J. C., Bottou, L. & Weinberger, K. Q. (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 1097–1105). Curran Associates, Inc. Consulted http://papers.nips.cc/paper/ 4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

Laguna, A. B., Riba, E., Ponsa, D. & Mikolajczyk, K. (2019). Key. Net: Keypoint Detection by Handcrafted and Learned CNN Filters. *arXiv preprint arXiv:1904.00889*.

LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436–444.

Lenc, K. & Vedaldi, A. (2018). Large scale evaluation of local image feature detectors on homography datasets. *arXiv preprint arXiv:1807.07939*.

Lindeberg, T. (1994). Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2), 225–270.

Lindholm, S. & Kronander, J. (2011). Accounting for uncertainty in medical data: A cuda implementation of normalized convolution. *Proceedings of SIGRAD 2011. Evaluations of Graphics and Visualization—Efficiency; Usefulness; Accessibility; Usability; November 17-18; 2011; KTH; Stockholm; Sweden*, (065), 35–42.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60(2), 91–110. doi: 10.1023/B:VISI.0000029664.99615.94.

Marcus, D. S., Wang, T. H., Parker, J., Csernansky, J. G., Morris, J. C. & Buckner, R. L. (2007). Open Access Series of Imaging Studies (OASIS): Cross-sectional MRI Data in Young, Middle Aged, Nondemented, and Demented Older Adults. *Journal of Cognitive Neuroscience*, 19(9), 1498-1507. doi: 10.1162/jocn.2007.19.9.1498.

Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R. et al. (2014). The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE transactions on medical imaging*, 34(10), 1993–2024.

Mitchell, D. P. & Netravali, A. N. (1988). Reconstruction Filters in Computer-Graphics. *SIGGRAPH Comput. Graph.*, 22(4), 221–228. doi: 10.1145/378456.378514.

Muja, M. & Lowe, D. G. (2014). Scalable nearest neighbor algorithms for high dimensional data. *IEEE transactions on pattern analysis and machine intelligence*, 36(11), 2227–2240.

Ono, Y., Trulls, E., Fua, P. & Yi, K. M. (2018). LF-Net: learning local features from images. *Advances in Neural Information Processing Systems*, pp. 6234–6244.

Palanisamy, K. & Prasath, S. (2015). Methods on Skull Stripping of MRI Head Scan Images—a Review. *Journal of Digital Imaging*, 29. doi: 10.1007/s10278-015-9847-8.

Pepin, E., Carluer, J.-B., Chauvin, L., Toews, M. & Harmouche, R. (2020). Large-scale Unbiased Neuroimage Indexing via 3D GPU-SIFT Filtering and Keypoint Masking. *MACHINE LEARNING IN CLINICAL NEUROIMAGING, 2020*.

Ritter, S., Barrett, D. G. T., Santoro, A. & Botvinick, M. M. (2017, 06–11 Aug). Cognitive Psychology for Deep Neural Networks: A Shape Bias Case Study. *Proceedings of the 34th International Conference on Machine Learning*, 70(Proceedings of Machine Learning Research), 2940–2949. Consulted http://proceedings.mlr.press/v70/ritter17a.html.

Sinha, S. N., Frahm, J.-M., Pollefeys, M. & Genc, Y. (2006). GPU-based video feature tracking and matching. *EDGE, workshop on edge computing using new commodity architectures*, 278, 4321.

Smith, S. M. (2002). Fast robust automated brain extraction. *Human brain mapping*, 17(3), 143–155.

Ségonne, F., Dale, A. M., Busa, E., Glessner, M., Salat, D., Hahn, H. K. & Fischl, B. (2004). A hybrid approach to the skull stripping problem in MRI. *NeuroImage*, 22(3), 1060 – 1075. doi: https://doi.org/10.1016/j.neuroimage.2004.03.032.

Toews, M. W. (2007 (accessed June 22, 2020)). Standard deviation diagram. Consulted https://en.wikipedia.org/wiki/Normal_distribution#/media/File:Standard_deviation_diagram.svg.

Toews, M. & Wells III, W. M. (2013). Efficient and Robust Model-to-image Alignment using 3D Scale-invariant Features. *Med Image Anal*, 17(3), 271-82. doi: 10.1016/j.media.2012.11.002.

Toews, M., Wells III, W., Collins, D. L. & Arbel, T. (2010). Feature-based morphometry: Discovering group-related anatomical patterns. *NeuroImage*, 49(3), 2318–2327.

Turin, G. (1960). An introduction to matched filters. *IRE transactions on Information theory*, 6(3), 311–329.

Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E., Yacoub, E., Ugurbil, K., Consortium, W.-M. H. et al. (2013). The WU-Minn human connectome project: an overview. *Neuroimage*, 80, 62–79.

Wachinger, C., Toews, M., Langs, G., Wells, W. & Golland, P. (2015). Keypoint Transfer for Fast Whole-Body Segmentation. *IEEE Transactions on Medical Imaging*, 39(2), 273-282. doi: 10.1109/TMI.2018.2851194.

Wachinger, C., Reuter, M. & Klein, T. (2018). DeepNAT: Deep convolutional neural network for segmenting neuroanatomy. *NeuroImage*, 170, 434–445.

Yi, K. M., Trulls, E., Lepetit, V. & Fua, P. (2016). Lift: Learned invariant feature transform. *European Conference on Computer Vision*, pp. 467–483.