

Standardised convolutional filtering for Radiomics

Image Biomarker Standardisation Initiative (IBSI)

Adrien Depeursinge^{1,2}, Vincent Andrearczyk¹, Philip Whybra³,
Joost van Griethuysen^{4,5,6}, Henning Müller^{1,7}, Roger Schaer¹,
Martin Vallières^{8,9,*}, Alex Zwanenburg^{10,11,*}

¹Institute of Information Systems, University of Applied Sciences Western Switzerland (HES-SO),
Switzerland

²Service of Nuclear Medicine and Molecular Imaging, Centre Hospitalier Universitaire Vaudois (CHUV),
Lausanne, Switzerland

³School of Engineering, Cardiff University, Cardiff, United Kingdom

⁴Department of Radiology, the Netherlands Cancer Institute (NKI), Amsterdam, the Netherlands

⁵GROW-School for Oncology and Developmental Biology, Maastricht University Medical Center,
Maastricht, The Netherlands

⁶Department of Radiation Oncology, Dana-Farber Cancer Institute, Brigham and Women's Hospital,
Harvard Medical School, Boston, MA

⁷University of Geneva, Geneva, Switzerland

⁸Department of Computer Science, University of Sherbrooke, Sherbrooke, Québec, Canada

⁹GRIIS, University of Sherbrooke, Sherbrooke, Québec, Canada

¹⁰OncoRay – National Center for Radiation Research in Oncology, Faculty of Medicine and University
Hospital Carl Gustav Carus, Technische Universität Dresden, Helmholtz-Zentrum Dresden - Rossendorf,
Dresden, Germany

¹¹National Center for Tumour Diseases (NCT), Partner Site Dresden, Germany: German Cancer
Research Center (DKFZ), Heidelberg, Germany; Faculty of Medicine, University Hospital Carl Gustav
Carus, Technische Universität Dresden, Dresden, Germany, and; Helmholtz Association /
Helmholtz-Zentrum Dresden - Rossendorf (HZDR), Dresden, Germany

*These authors contributed equally to this work

14th December 2021

The image biomarker standardisation initiative

The Image Biomarker Standardisation Initiative (IBSI) is an independent international collaboration that works towards standardising the extraction of image biomarkers from acquired imaging for the purpose of high-throughput quantitative image analysis (radiomics). Lack of reproducibility and validation of radiomic studies is considered to be a major challenge for the field. Part of this challenge lies in the scantiness of consensus-based guidelines and definitions for the process of translating acquired imaging into high-throughput image biomarkers. The IBSI therefore seeks to provide standardised image biomarker nomenclature and definitions, a standardised general image processing workflow, tools for verifying radiomics software implementations and reporting guidelines for radiomic studies.

Additional information can be found on the official IBSI website^a.

Permanent identifiers

The IBSI uses permanent identifiers for image biomarker definitions and important related concepts such as image processing. These consist of four-character codes and may be used for reference. Please do not use page numbers or section numbers as references, as these are subject to change.

The current version of the reference manual does not contain any permanent identifiers. These will be added in future versions.

Copyright

This work is licensed under the Creative Commons Attribution 4.0 International License (CC-BY).

Copyright information regarding the benchmark data sets may be found on GitHub^b.

Change notes

Changes in version 5:

- Updated list of participants.
- Clarified the intended use of the "mirror" boundary condition (section 2.2).
- The 2D tests in phase 2 are now without interpolation (section 5.2).

Changes in version 4:

^a<https://theibsi.github.io/>, as of June 2020.

^bhttps://github.com/theibsi/data_sets, as of June 2020.

- A list of participants was added.
- Moved embedding of convolutional filtering in radiomics workflow to section 1.1.
- Added several implementation troubleshooting boxes (chapter 4).
- Clarified units of σ and λ parameters of Gabor filters (section 4.5).
- Clarified spatial extent of Gabor filter support (section 4.5).
- Clarified implementation of the *à trous* algorithm (section 4.6.2).
- Extended description of non-separable wavelets (section 4.6.4).
- Added recommendation to not perform image decomposition beyond the first level using separable wavelets (section 4.6.5).
- Extended description of the Riesz transform (section 4.7)
- Removed chapter 5 *Filtering for Radiomics in Practice*.

Participants

The following people contributed to this reference manual and the project as a whole.

v4: Added list of participants.
v5: updated list.

Name	Affiliation
Vincent Andrearczyk	Institute of Information Systems, University of Applied Sciences Western Switzerland (HES-SO), Sierre, Switzerland
Aditya Apte	Department of Medical Physics, Memorial Sloan Kettering Cancer Center, New York (NY), USA
Alexandre Ayotte	Department of Computer Science, Interdisciplinary Research Group in Health Informatics (GRIIS), University of Sherbrooke, Canada
Bhakti Vilas Baheti	Center for Biomedical image Computing and Analytics (CBICA), University of Pennsylvania, Philadelphia (PA), USA
Spyridon Bakas	Center for Biomedical image Computing and Analytics (CBICA), University of Pennsylvania, Philadelphia (PA), USA; Department of Radiology, Perelman School of Medicine, University of Pennsylvania, Philadelphia (PA), USA; Department of Pathology and Laboratory Medicine, Perelman School of Medicine, University of Pennsylvania, Philadelphia (PA), USA
Andrea Bettinelli	Veneto Institute of Oncology - IOV IRCSS, Padova, Italy
Ronald Boellaard	Department of Nuclear Medicine and Molecular Imaging, University of Groningen, University Medical Center Groningen (UMCG), Groningen, The Netherlands; Radiology and Nuclear Medicine, VU University Medical Centre (VUMC), Amsterdam, The Netherlands
Marta Bogowicz	Department of Radiation Oncology, University Hospital Zurich, University of Zurich, Zurich, Switzerland
Luca Boldrini	Fondazione Policlinico Universitario "A. Gemelli" IRCCS, Rome, Italy

Continued on next page

Name	Affiliation
Irène Buvat	Laboratory of Translational Imaging in Oncology, U1288 Inserm, Institut Curie, Université PLS, Orsay, France
Gary J.R. Cook	Cancer Imaging Dept, School of Biomedical Engineering and Imaging Sciences, King's College London, London, United Kingdom
Adrien Depeursinge	Institute of Information Systems, University of Applied Sciences Western Switzerland (HES-SO), Sierre, Switzerland; Department of Nuclear Medicine and Molecular Imaging, Lausanne University Hospital, Lausanne, Switzerland
Florian Dietsche	Department of Radiation Oncology, University Hospital Zurich, University of Zurich, Zurich, Switzerland
Nicola Dinapoli	Fondazione Policlinico Universitario "A. Gemelli" IRCCS, Rome, Italy
Vicky Goh	Cancer Imaging Dept, School of Biomedical Engineering and Imaging Sciences, King's College London, London, United Kingdom
Matthias Guckenberger	Department of Radiation Oncology, University Hospital Zurich, University of Zurich, Zurich, Switzerland
Mathieu Hatt	Laboratory of Medical Information Processing (LaTIM) - team ACTION (image-guided therapeutic action in oncology), INSERM, UMR 1101, IBSAM, UBO, UBL, Brest, France
Aditi Iyer	Department of Medical Physics, Memorial Sloan Kettering Cancer Center, New York (NY), USA
Vincent Jaouen	IMT-Atlantique, Plouzané, France
Jacopo Lenkowicz	Fondazione Policlinico Universitario "A. Gemelli" IRCCS, Rome, Italy
Steffen Löck	OncoRay – National Center for Radiation Research in Oncology, Faculty of Medicine and University Hospital Carl Gustav Carus, Technische Universität Dresden, Helmholtz-Zentrum Dresden - Rossendorf, Dresden, Germany; German Cancer Consortium (DKTK), Partner Site Dresden, and German Cancer Research Center (DKFZ), Heidelberg, Germany; Department of Radiotherapy and Radiation Oncology, Faculty of Medicine and University Hospital Carl Gustav Carus, Technische Universität Dresden, Dresden, Germany
Francesca Marturano	Veneto Institute of Oncology - IOV IRCSS, Padova, Italy
Francesca Marturano	Veneto Institute of Oncology - IOV IRCSS, Padova, Italy
Olivier Morin	Department of Radiation Oncology, University of California, San Francisco (CA), USA
Henning Müller	Institute of Information Systems, University of Applied Sciences Western Switzerland (HES-SO), Sierre, Switzerland; University of Geneva, Geneva, Switzerland
Christophe Nioche	Laboratory of Translational Imaging in Oncology, U1288 Inserm, Institut Curie, Université PLS, Orsay, France
Fanny Orlhac	Laboratory of Translational Imaging in Oncology, U1288 Inserm, Institut Curie, Université PLS, Orsay, France

Continued on next page

Name	Affiliation
Sarthak Pati	Center for Biomedical image Computing and Analytics (CBICA), University of Pennsylvania, Philadelphia (PA), USA; Department of Radiology, Perelman School of Medicine, University of Pennsylvania, Philadelphia (PA), USA
Christopher Rookyard	Cancer Imaging Dept, School of Biomedical Engineering and Imaging Sciences, King's College London, London, United Kingdom
Roger Schaer	Institute of Information Systems, University of Applied Sciences Western Switzerland (HES-SO), Sierre, Switzerland
Andreas Schindele	Universitätsklinikum Augsburg, Augsburg, Germany
Emiliano Spezi	School of Engineering, Cardiff University, Cardiff, United Kingdom; Department of Medical Physics, Velindre Cancer Centre, Cardiff, United Kingdom
Stephanie Tanadini-Lang	Department of Radiation Oncology, University Hospital Zurich, University of Zurich, Zurich, Switzerland
Florent Tixier	Laboratory of Medical Information Processing (LaTIM) - team ACTION (image-guided therapeutic action in oncology), INSERM, UMR 1101, IBSAM, UBO, UBL, Brest, France
Taman Upadhaya	Department of Radiation Oncology, University of California, San Francisco (CA), USA
Vincenzo Valentini	Fondazione Policlinico Universitario "A. Gemelli" IRCCS, Rome, Italy
Martin Vallières	Department of Computer Science, Interdisciplinary Research Group in Health Informatics (GRIIS), University of Sherbrooke, Canada
Joost van Griethuysen	Department of Radiology, the Netherlands Cancer Institute (NKI), Amsterdam, The Netherlands; GROW-School for Oncology and Developmental Biology, Maastricht University Medical Center, Maastricht, The Netherlands; Department of Radiation Oncology, Dana-Farber Cancer Institute, Brigham and Women's Hospital, Harvard Medical School, Boston (MA), USA
Philip Whybra	School of Engineering, Cardiff University, Cardiff, United Kingdom
Alex Zwanenburg	OncoRay – National Center for Radiation Research in Oncology, Faculty of Medicine and University Hospital Carl Gustav Carus, Technische Universität Dresden, Helmholtz-Zentrum Dresden - Rossendorf, Dresden, Germany; National Center for Tumor Diseases (NCT), Partner Site Dresden, Germany; German Cancer Research Center (DKFZ), Heidelberg, Germany; Faculty of Medicine and University Hospital Carl Gustav Carus, Technische Universität Dresden, Dresden, Germany, and; Helmholtz Association / Helmholtz-Zentrum Dresden - Rossendorf (HZDR), Dresden, Germany; German Cancer Consortium (DKTK), Partner Site Dresden, and German Cancer Research Center (DKFZ), Heidelberg, Germany

Continued on next page

Name	Affiliation
------	-------------

Table 1 — Alphabetical list of IBSI2 contributors.

Contents

1	Introduction	1
1.1	Convolutional Filtering within the Radiomics Workflow	2
1.2	Notations and Conventions	2
1.3	Steps in convolutional filtering	4
2	Convolution	6
2.1	Separability of the Convolution	7
2.2	Boundary Conditions	7
3	Filtering Properties	11
3.1	Geometric Invariances for Medical Image Analysis	11
3.2	Directional Sensitivity	12
3.3	Combining Directional Sensitivity and Invariance to Local Rotations	12
3.4	Spectral Coverage	13
4	Common Convolutional Filters in Radiomics	16
4.1	Identification of Common Approaches	16
4.2	Mean Filter	16
4.3	Laplacian of Gaussian	17
4.4	Laws Kernels	18
4.5	Gabor	21
4.6	Wavelets	23
4.7	Riesz	29
4.8	Qualitative Comparison of Linear Filter Properties	33
4.9	Interpolation and convolutional filtering	33
5	Benchmarking	37
5.1	Phase 1: Benchmarking filters using digital phantoms	37
5.2	Phase 2: Benchmarking filters using a lung cancer CT image	42

<i>CONTENTS</i>	vii
6 Conclusion	47
Bibliography	48
A Equivariant right angle rotational representation for separable filters	53
A.1 <i>2D</i> case	53
A.2 <i>3D</i> case	53

Chapter 1

Introduction

Medical imaging is often used to support clinical decision-making, but currently only through visual inspection or simple measures. Additional relevant information concerning *e.g.*, disease phenotypes, may be present in medical images but remains unassessed²⁹. Radiomics characterises regions of interest in medical images using quantitative measures for *e.g.* morphology, image intensities and texture. Such characteristics are called image biomarkers or image features. Since 2012, an exponentially growing number of radiomics-related publications are demonstrating the relevance of artificial intelligence and medical image analysis in the context of various diseases and imaging modalities. However, the standardisation and reproducibility of image biomarkers is a major remaining challenge⁴⁹. In particular, the interoperability of radiomics software is hindered by the lack of consensus concerning the exact calculation of image biomarkers. As a result, considerable variations in biomarker values have been reported, even if they are computed from the same image^{7,24,26}. To improve interoperability, the Image Biomarker Standardisation Initiative (IBSI), an independent international collaboration, proposes (i) reference manuals documenting mathematical definitions of image analysis processes and (ii) reference values for image biomarkers based on phantoms with controlled image input and well-defined configurations of the image processing parameters.

This reference manual draws on and extends the first IBSI reference manual⁵⁰ by defining and standardising image biomarkers based on so-called convolutional filters, *e.g.*, wavelets, Laws kernels and Laplacians of Gaussian (LoG). Here, we specifically address several so-called convolutional filters. Filters such as entropy, median, max and morphological filtering are not covered.

This document is structured as follows. The first section of this chapter describes the place of convolutional filtering in the overall radiomics workflow. The second section of this chapter introduces notations and conventions (Section 1.2). The final section provides a brief overview of the image filtering process for image biomarker extraction (Section 1.3). Chapter 2 details all aspects of convolutional operations, the fundamental process for linear image filtering. Chapter 3 lists important properties of convolutional filters. Most commonly used convolutional filtering methods in radiomics are identified and formally defined in Chapter 4. Chapter 4.8 qualitatively compares these filtering methods in terms of the properties introduced in Chapter 3. Finally, benchmarking of software implementations of the filters defined in Chapter 4 is proposed in Chapter 5, where phantoms are defined and the benchmarking procedure, including response map comparisons, is detailed.

1.1 Convolutional Filtering within the Radiomics Workflow

Convolutional filtering is part of image processing for radiomics. Within the image processing workflow, filtering is conducted after image interpolation, see Figure 1.1. Filtering an image creates a response map that for the filters discussed in this document has the same dimensions as the input image. Radiomic features, such as those detailed in the first IBSI reference manual⁵⁰, may then be computed from the response map as well.

v4: Moved embedding of convolutional filtering in radiomics workflow to section 1.1.

1.2 Notations and Conventions

We follow the notations of Depeursinge *et al.*¹⁴ and Unser⁴¹. The imaginary symbol for complex numbers is denoted by j . A continuous image is modeled as a D -dimensional function of the spatial coordinates $\mathbf{x} = (x_1, \dots, x_D) \in \mathbb{R}^D$, taking values $f(\mathbf{x}) \in \mathbb{R}$. The Fourier transform of an integrable $f(\mathbf{x})$ is noted $\hat{f}(\boldsymbol{\omega}) \in \mathbb{C}$ and is defined as

$$f(\mathbf{x}) \xleftrightarrow{\mathcal{F}} \hat{f}(\boldsymbol{\omega}) = \int_{\mathbf{x} \in \mathbb{R}^D} f(\mathbf{x}) e^{-j\langle \boldsymbol{\omega}, \mathbf{x} \rangle} d\mathbf{x}, \quad (1.1)$$

where $\boldsymbol{\omega} = (\omega_1, \dots, \omega_D) \in \mathbb{R}^D$ is the frequency coordinate vector and $\langle \cdot, \cdot \rangle$ denotes the scalar product.

A discrete image is modeled as a D -dimensional function of the variable $\mathbf{k} = (k_1, \dots, k_D) \in \mathbb{Z}^D$, taking values $f[\mathbf{k}] \in \mathbb{R}$. The discrete Fourier transform^a of a summable $f[\mathbf{k}]$ is noted $\hat{f}[\boldsymbol{\nu}] \in \mathbb{C}$, with $\boldsymbol{\nu} = (\nu_1, \dots, \nu_D) \in \mathbb{Z}^D$ the discrete frequency coordinate vector, and is defined as

$$f[\mathbf{k}] \xleftrightarrow{\mathcal{F}} \hat{f}[\boldsymbol{\nu}] = \sum_{\mathbf{k} \in \mathbb{Z}^D} f[\mathbf{k}] e^{-j\langle \boldsymbol{\nu}, \mathbf{k} \rangle}. \quad (1.2)$$

1.2.1 Image Directions

Two reference frames are important for image filters^b: (i) the patient frame of reference and (ii) the image frame of reference. The patient frame of reference determines how the image itself is oriented with regard to anatomical directions. Filters are computed in the image frame of reference, which does not need to exactly match the patient frame of reference. However, to use filters consistently, all images in a given study should have the same orientation relative to the patient frame of reference. This may require the rotation of image sets (*e.g.* scans) to a common orientation relative to the patient frame of reference, *e.g.* in a study with mixed sagittal and axial images. Besides, if data stored in different image formats are used in the same study, special care needs to be taken to ensure that the image reference frame of all image sets have the same orientation relative to the patient frame of reference (*e.g.* this may require flipping and/or transposing images along different directions). For example, image file types such as Digital Imaging and COmmunications in Medicine (DICOM) and Neuroimaging Informatics Technology Initiative (NIfTI) formats may store an image using a different orientation relative to the patient frame of reference. Overall, to maximize the reproducibility potential of a given study, two important parameters should be reported: (i) the common orientation of the image frame of reference of all image sets relative to the patient frame of reference (*e.g.* the Image Orientation Patient (0020 0037) field for the DICOM format) and (ii) if applicable, the (potentially different) rotation operations applied to each image set separately to ensure a common orientation of the image frame of reference relative

^aIn this discrete setting, the normalised Nyquist frequency is denoted by $\nu_B = \pi$.

^bFor further information, please see https://www.slicer.org/wiki/Coordinate_systems, as of November 2019.

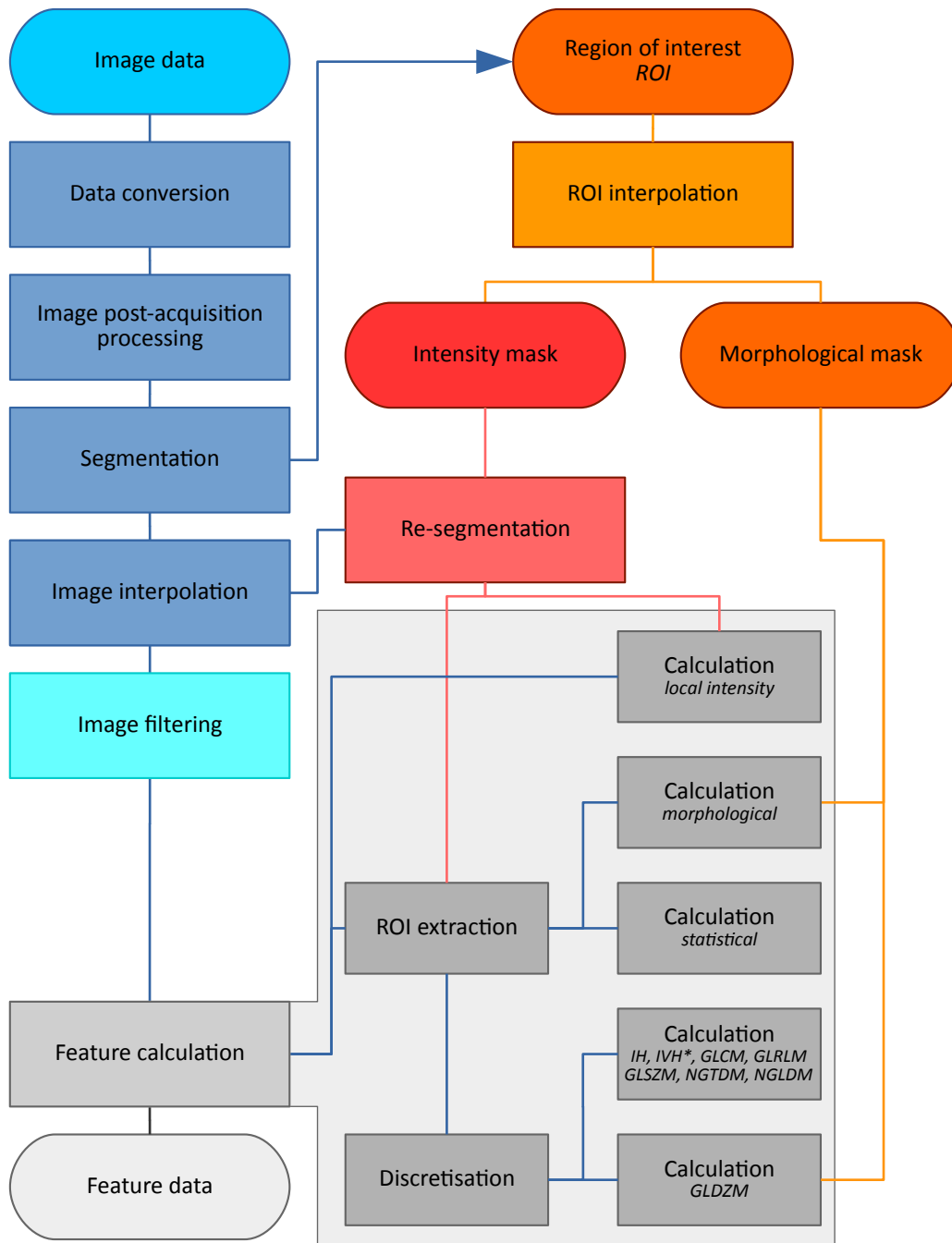


Figure 1.1 — Overall image processing scheme with image filtering (adapted from Zwanenburg *et al.*⁵⁰). After loading a medical image, the image data are optionally converted (*e.g.* SUV normalisation). The image may then undergo further post-processing (*e.g.* noise reduction, bias field correction). Subsequently, a segmentation mask is created or loaded to identify the ROI in the image. The image is subsequently interpolated to ensure that image voxels are isometric, after which image filters may be applied to the image. The ROI mask is interpolated to the same grid as the image, prior to forming intensity and morphological ROI masks. The intensity (but not morphological) mask is optionally re-segmented based on image intensities of the unfiltered image. Subsequently, features are computed from the filtered image and the applicable ROI masks. The intensities in ROI intensity mask may undergo discretisation prior to computing features from *e.g.* texture families. IH: intensity histogram; IVH: intensity-volume histogram; GLCM: grey level cooccurrence matrix; GLRLM: grey level run length matrix; GLSZM: grey level size zone matrix; NGTDM: neighbourhood grey tone difference matrix; NGLDM: Neighbouring grey level dependence matrix; GLDZM: grey level distance zone matrix; *A different discretisation scheme is usually used for computing IVH features.

to the patient frame of reference.

The image frame of reference determines how the stored image grid is oriented and thus where image voxels are located. Within the image reference frame, we define filter directions as follows:

- k_1 (*i.e.* x) goes from left to right with increasing grid index values,
- k_2 (*i.e.* y) goes from top to bottom with increasing grid index values,
- k_3 (*i.e.* z) goes from front to back with increasing grid index values.

Implementation Troubleshooting

- The orientation phantom (section 5.1) can be used to assess whether the orientation used in an implementation conforms to the expected orientation.
- The orientation phantom has a dimension of (32, 48, 64) voxels along k_1 (x), k_2 (y) and k_3 (z) axes, respectively. The pixel intensity increases with the distance from the origin, which has an intensity of 0. The most distal voxel has an intensity of 141.

1.3 Steps in convolutional filtering

This section provides an overview of the filter-based image biomarker extraction process, which is shown in Fig. 1.2. It consists of the following steps:

1.3.1 Image Padding

The input image $f[\mathbf{k}]$ is first padded with an appropriate boundary condition to form an extended image $f_{\text{ext}}[\mathbf{k}]$. This step is further required to calculate the response of a filter close to the boundaries of f and is detailed in Section 2.2.

1.3.2 Convolution

$f_{\text{ext}}[\mathbf{k}]$ is convolved with a filter $g[\mathbf{k}]$, which yields a response map $h[\mathbf{k}]$ with the same dimension as $f[\mathbf{k}]$. This process is described in Section 2 and consists of applying the filter at all possible locations \mathbf{k} in the input image. Local or global rotation invariance operations may be applied during this step to minimise the variation of $h[\mathbf{k}]$ with respect to rotations of the input image, as well as variations in orientations of local image patterns (*e.g.* tumor walls or vessels, see Chapter 3).

1.3.3 Aggregation

$h[\mathbf{k}]$ is not a directly usable quantitative image feature and needs to be aggregated (*i.e.* summarised) over a given Region Of Interest (ROI) M using an aggregation function¹⁴. The latter will transform the response map h into a scalar measurement η , which can be further used in a statistical or machine learning model. h can be seen as a new image derived from f , on which any feature defined in the first IBSI reference manual can be computed⁵⁰. However, the interpretability of the latter would be difficult (unless the filter is simply used to *e.g.* remove high frequency noise), and the large number of possible combinations of filters and features increases the risk of fortuitous correlations between features and clinical endpoints. In practice, the most obvious aggregation

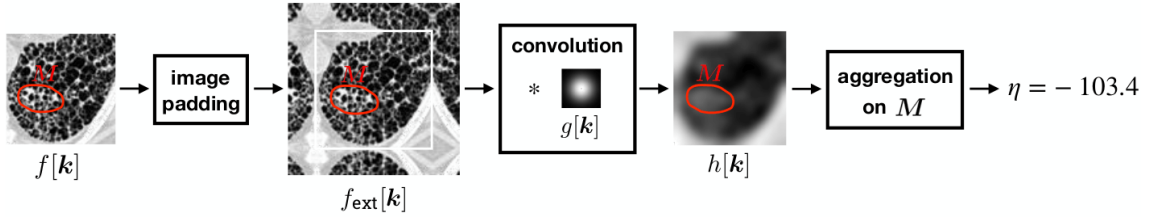


Figure 1.2 — Overview of the filter-based image biomarker extraction process.

functions are the first four statistical moments of the voxel value distribution in M , the first one being the average:

$$\eta = \frac{1}{|M|} \sum_{\mathbf{k} \in M} h[\mathbf{k}], \quad (1.3)$$

where $|M|$ denotes the number of voxels in the ROI. It is worth noting that the influence of the aggregation function is as important as the filter and it should be considered carefully¹². For the particular case of average-based aggregation in Eq. (1.3), one must be careful when using zero-mean filters (*e.g.* LoG, some Laws kernels, Gabor, wavelets, Riesz) leading to $\eta \approx 0$. In this case, taking the energy $h^2[\mathbf{k}]$ or the absolute value $|h[\mathbf{k}]|$ of the response map before the aggregation is recommended.

Chapter 2

Convolution

The D -dimensional continuous convolution of a filter $g(\mathbf{x})$ and an image $f(\mathbf{x})$ yields a response map $h(\mathbf{x})$ with the same dimension N^D as f following

$$h(\mathbf{x}_0) = (g * f)(\mathbf{x}_0) = \int_{\mathbf{x} \in \mathbb{R}^D} g(\mathbf{x}) f(\mathbf{x}_0 - \mathbf{x}) d\mathbf{x}. \quad (2.1)$$

The continuous convolution is only explained here for reference. In practice, images are discretised and the corresponding discrete convolution^a of a $M^D = M \times \dots \times M$ filter $g[\mathbf{k}]$ and a $N^D = N \times \dots \times N$ image $f[\mathbf{k}]$ is

$$h[\mathbf{k}_0] = (g * f)[\mathbf{k}_0] = \sum_{\mathbf{k} \in M^D} g[\mathbf{k}] f[\mathbf{k}_0 - \mathbf{k}]. \quad (2.2)$$

(2.1) and (2.2) can also be computed in the Fourier domain and the convolution becomes in this case just the product

$$(g * f)(\mathbf{x}) \xrightarrow{\mathcal{F}} \hat{g}(\boldsymbol{\omega}) \hat{f}(\boldsymbol{\omega}), \quad (2.3)$$

which remains valid for the discrete case via a Hadamard product between the arrays of the image and the filter.

The latter implementation becomes efficient when the filter size is large as the computational cost of the Fast Fourier Transform (FFT) becomes small when compared to implementing the D interleaved sums in (2.2). As a rule of thumb in $2D^b$, for a $N \times N$ image and a $M \times M$ filter, the computational cost to convolve them in the spatial domain is $N^2 \cdot M^2$. In the Fourier domain, the cost is $N^2 \cdot (1 + 2 \log_2(N^2))$ when both the filter and the image are already defined in the Fourier domain and the filter is interpolated to the dimension N to allow pointwise multiplication. For instance, with a $N^2 = 512 \times 512$ image, FFT filtering becomes more efficient than spatial filtering when the filter is larger than $M^2 = 7 \times 7$.

At a fixed position \mathbf{x}_0 , the value of the convolution response $(g * f)(\mathbf{x}_0)$, *i.e.*, the application of the filter to the image, is the scalar product

$$(g * f)(\mathbf{x}_0) = \langle f(\mathbf{x}_0 - \cdot), g(\cdot) \rangle, \quad (2.4)$$

with $f(\mathbf{x}_0 - \cdot)$ is the function $x \mapsto f(\mathbf{x}_0 - x)$.

^aWe consider here filters and images with square ($D = 2$) or cubic ($D = 3$) dimensions for simplicity.

^bThis estimation only counts the number of operations. In practice, the efficiency will strongly depend on software language (complex data storage and manipulation) and hardware.

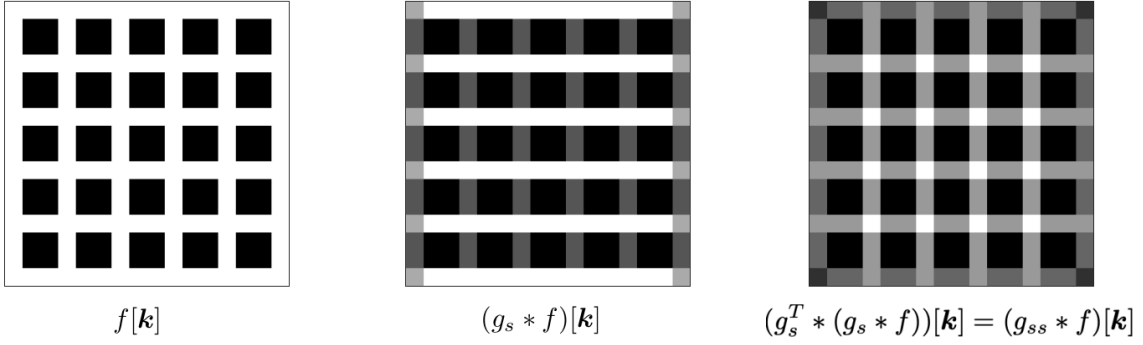


Figure 2.1 — A 16×16 image $f[\mathbf{k}]$ (left) is filtered by g_{ss} defined in Eq. (2.6) using a separable convolution. The intermediate image $(g_s * f)[\mathbf{k}]$ is shown (centre), where the convolution is performed with g_s aligned along the lines (i.e. k_1) of f . After convolving this intermediate image with g_s^T (i.e. aligned along the columns k_2), the result (right) is equivalent to a 2D convolution of f and g_{ss} .

2.1 Separability of the Convolution

Specific filter subtypes are *separable*, which means that their D -dimensional kernel $g_{123\dots D}[\mathbf{k}]$ can be obtained from the outer product of D 1-dimensional kernels $g_i[k_i]$ as

$$g_{123\dots D}[\mathbf{k}] = \prod_{i=1}^D g_i[k_i]. \quad (2.5)$$

An example of a separable 2D unit-norm 3×3 smoother $g_{ss}[\mathbf{k}]$ is

$$g_{ss}[\mathbf{k}] = g_s[k] \otimes g_s[k] = g_s[k] \cdot g_s^T[k] = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \cdot \frac{1}{\sqrt{6}} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \quad (2.6)$$

The steps of a separable convolution process of a 16×16 image $f[\mathbf{k}]$ with $g_{ss}[\mathbf{k}]$ are illustrated in Fig. 2.1. Thanks to the associativity of convolution, separable filters can be implemented efficiently with D successive convolutions with 1D filters when compared to more computationally expensive convolutions with D -dimensional filters. For instance, we have the composition

$$(g_2 * (g_1 * f))[\mathbf{k}] = (g_1 * (g_2 * f))[\mathbf{k}] = (g * f)[\mathbf{k}]. \quad (2.7)$$

However, satisfying Eq. (2.5) strongly constrains the design of the filter. Moreover, because image axes are analysed separately, separable convolutions yield directional image analyses biased along k_1 , k_2 and k_3 with no rotation invariance by default (the importance of the latter is detailed in Section 3.1). The only separable and directionally insensitive kernel is the Gaussian.

As a convention, the filter responses are noted $h_{123}[\mathbf{k}]$ when resulting from a convolution with the three 1D filters as follows: g_1 along k_1 , g_2 along k_2 , and g_3 along k_3 (see section 1.2.1). The corresponding 3D kernel is noted $g_{123}[\mathbf{k}]$.

2.2 Boundary Conditions

Computing the discrete convolution as in Eq. (2.2) when the distance between the center \mathbf{k}_0 of the filter and a boundary of the image is smaller than half of the support M of the filter requires

accessing pixel values outside the support of the image. Therefore, a method must be used to impute pixel values in the vicinity of the image boundaries, *i.e.*, to generate an extended image $f_{\text{ext}}[\mathbf{k}]$ that includes a margin with a width/height/depth of $\lfloor M/2 - 1 \rfloor$. Four common padding methods are described in the following subsections and are compared qualitatively in Fig. 2.2 and 2.3, where a CT image of honeycombing lung parenchyma (Fig. 2.2a, $N = 301$) is smoothed by convolution with an isotropic Gaussian filter (Fig. 2.2b, $M = 70$, $\sigma = 15$).

Because all padding methods make arbitrary assumptions concerning image content beyond the boundaries, a method should be chosen based on the expected image background. However, general advantages and disadvantages exist and are discussed in the next sections.

2.2.1 Constant Value Padding

The simplest method to extend the support of $f[\mathbf{k}]$ is to pad it with a constant value C as

$$\begin{cases} f_{\text{ext}}^{\text{constant}}[\mathbf{k}] = f[\mathbf{k}] & \forall (k_1, \dots, k_D) \in \{0, \dots, N-1\}^D \text{ and,} \\ f_{\text{ext}}^{\text{constant}}[\mathbf{k}] = C & \text{otherwise.} \end{cases}$$

Thus, all pixels outside the original image are assigned the constant value C . This is illustrated in Fig. 2.2c. Though constant value padding is simple to implement, potentially sharp transitions between C and the image value at a boundary may yield inconsistent filter responses. Distortions may appear in the boundary region of the response image as a consequence. Such artefacts can be observed in all boundaries of the filtered image in Fig. 2.2d. With $C = 0$, this method is also called *zero padding*.

2.2.2 Nearest Value Padding

The nearest value padding method consists of repeating the values of the image at the boundary. We have

$$\begin{cases} f_{\text{ext}}^{\text{nearest}}[\mathbf{k}] = f[\mathbf{k}] & \forall (k_1, \dots, k_D) \in \{0, \dots, N-1\}^D \text{ and,} \\ f_{\text{ext}}^{\text{nearest}}[\mathbf{k}] = f[\mathbf{k}_{\text{nearest}}] & \text{otherwise,} \end{cases}$$

where $\mathbf{k}_{\text{nearest}} := \arg \min_{\mathbf{k}_{\text{nearest}}} (||\mathbf{k} - \mathbf{k}_{\text{nearest}}||)$ is the closest neighbour in f . Thus, all pixels outside the original image get the intensity value of the closest pixel in the original image. This is illustrated in Fig. 2.2e. The advantage of this method is that the transitions at the boundary are usually relatively small. However, it introduces a nonexistent pattern in the response image. This method is also called *replicate*.

2.2.3 Periodisation

Another straightforward method is to repeat the image along every dimension, *i.e.*, so to periodise the image content. The extended image is therefore equivalent to the original image modulo its support as

$$f_{\text{ext}}^{\text{periodise}}[\mathbf{k}] = f[k_1 \bmod N, \dots, k_D \bmod N]$$

for indices $k_i = 0, \dots, N-1$. This is illustrated in Fig. 2.3a. The introduced patterns are consistent with the actual image content. However, strong transitions are possible at the boundaries. The subsequent artefacts can be observed in the upper part of the left boundary of the example response image (see Fig. 2.3b). It is worth noting that this periodisation is implicit if the convolution operation is performed in the Fourier domain using Eq. (2.3). This method is also called *wrapping*,

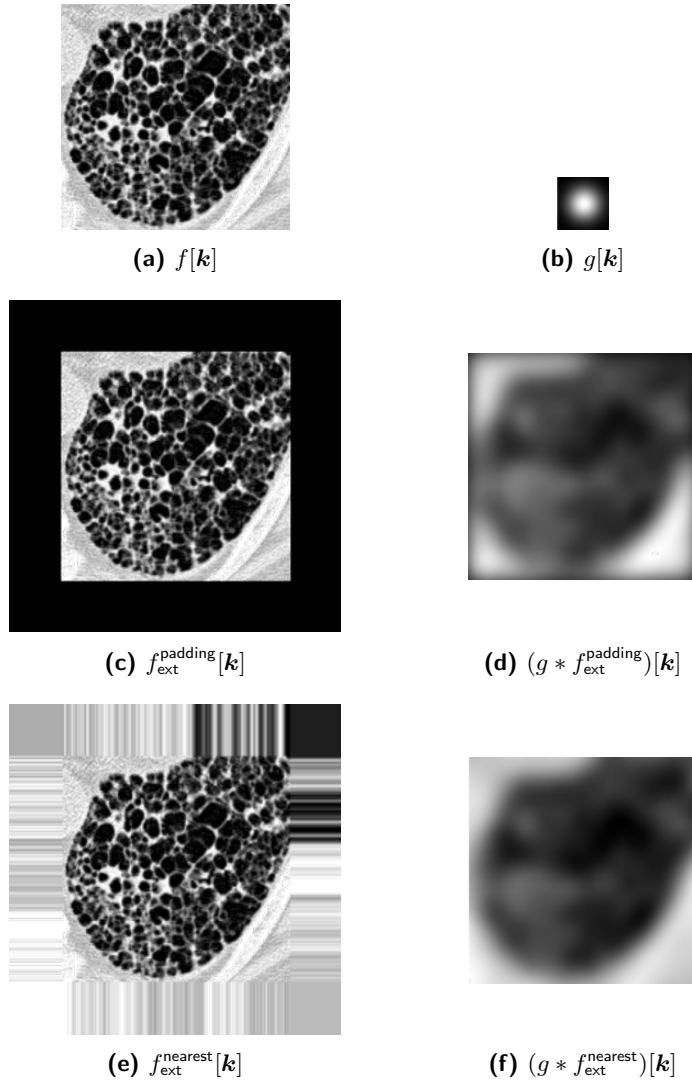


Figure 2.2 — Qualitative comparison of various methods for imputing image values at the boundaries. The image $f[\mathbf{k}]$ is smoothed by convolution with the Gaussian filter $g[\mathbf{k}]$. The response maps using either zero padding or nearest methods are compared.

circular or tiling.

2.2.4 Mirror

The mirror folding method consists of symmetrising the image at the boundaries. The extended image is $f_{\text{ext}}^{\text{mirror}}[\mathbf{k}] = f[\mathbf{k}']$, with $\mathbf{k}' = (k'_1, \dots, k'_D) \in \mathbb{Z}^D$, where

$$k'_i = \begin{cases} k_i \bmod N & \text{if } \lfloor \frac{k_i}{N} \rfloor \text{ is even,} \\ N - (k_i \bmod N + 1) & \text{otherwise,} \end{cases}$$

for indices $k_i = 0, \dots, N - 1$.

This is illustrated in Fig. 2.3c. The introduced patterns are consistent with the actual image content and the transitions at the boundaries are relatively smooth, minimising convolution artefacts (see Fig. 2.3d). Depending on the software implementation, the extended image may include or exclude the boundary pixels of the original image. This method is also called *symmetric*. For consistency reasons, we assume that boundary pixels are included.

v5: Clarified units of σ and λ parameters.

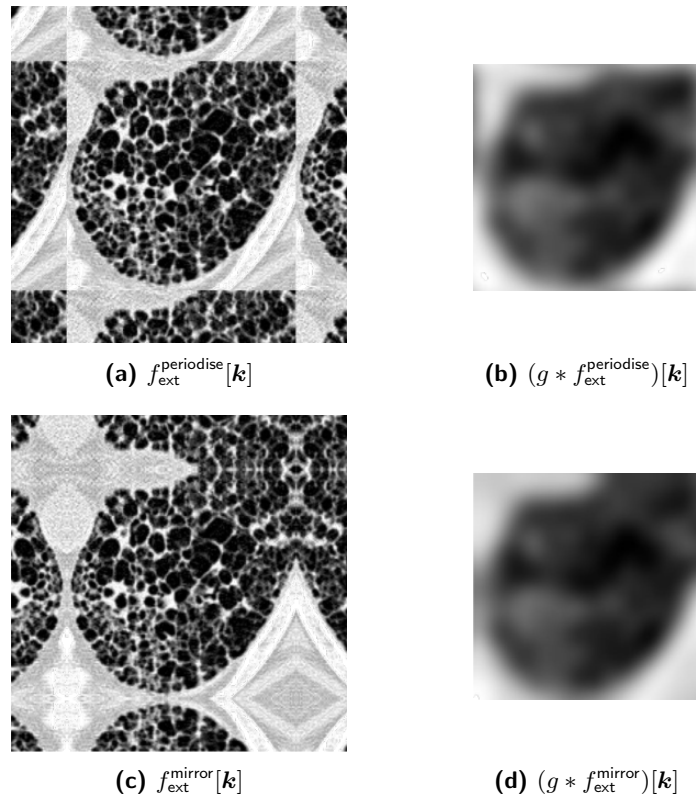


Figure 2.3 — Qualitative comparison of various methods for imputing image values at the boundaries. The image $f[\mathbf{k}]$ is smoothed by convolution with the Gaussian filter $g[\mathbf{k}]$ (see Fig. 2.2). The response maps using either periodisation or mirroring methods are compared.

2.2.5 Considerations for Radiomics

In many radiomics applications, the choice of boundary extension method is generally not important, as long as the ROI is sufficiently far from the image boundary, *i.e.* at least by more than half of the convolution filter width/height/depth. In the uncommon case where the ROI is close to the border, the *mirror* method may generally be recommended because it avoids sharp transitions and artificial patterns.

Chapter 3

Filtering Properties

Due to the characteristics common to medical images, several filter properties are important for image analysis methods to be optimal. In this section, we list these properties and suggest how they can be fulfilled with filtering techniques.

3.1 Geometric Invariances for Medical Image Analysis

Analysis of medical imaging requires invariance or equivariance to various geometric transformations. As in Depeursinge *et al.*¹⁴, we define the invariance of a function $g(\cdot)$ to a transformation \mathcal{T} of the image $f(x)$ as $g(\mathcal{T}\{f\})(x) = g(f)(x)$. In the case of filtering, this means that the response map obtained by the convolution of the image with a filter is unchanged under the effect of \mathcal{T} . The equivariance is defined as $g(\mathcal{T}\{f\})(x) = \mathcal{T}\{g(f)\}(x)$, *i.e.* the response map undergoes the same transformation \mathcal{T} as the image.

1. **Translation:** The response maps of linear filters are equivariant to translation by construction, thanks to the convolution operation. For a translation transformation $\mathcal{T}_{t,\mathbf{x}_0}$ by \mathbf{x}_0 , $g(\mathcal{T}_{t,\mathbf{x}_0}\{f\})(x) = \mathcal{T}_{t,\mathbf{x}_0}\{g(f)\}(x)$, meaning that the filtering process does not differ according to position, *e.g.* if an input $(0, 1, 2, 1, 0)$ leads to response $(0, 0, 1, 0, 0)$, then input $(0, 0, 1, 2, 1)$ should lead to response $(0, 0, 0, 1, 0)$. Equivariance to translations is required in medical imaging since we want to keep track of the positions where the filter responded to create full response maps, where patterns of interest may appear at random positions.
2. **Scaling:** In most cases, scale is a discriminative property in medical images. Thus, filters should not be scale invariant.
3. **Rotation:** Response maps should be equivariant to global rotations and invariant to local rotations as defined in Depeursinge *et al.*¹⁴. We consider a global rotation when applied around the origin $\mathbf{0}$ of the image. In this case, equivariance to global rotations is required for the same reasons as translation equivariance, where the positions where the filter responded should rotate in the same fashion as the image itself. A local rotation around a given position \mathbf{x}_0 requires the following: $\mathcal{T}_{r,\mathbf{x}_0} = \mathcal{T}_{t,\mathbf{x}_0}\mathcal{T}_r\mathcal{T}_{t,-\mathbf{x}_0}$. For $\mathcal{T}_{r,\mathbf{x}_0}$ a transformation that locally rotates patterns, we seek the invariance $g(\mathcal{T}_{r,\mathbf{x}_0}\{f\})(x) = g(f)(x)$. Invariance to local rotations is required to equivalently obtain filter responses to any orientation of a local pattern. For example, patterns such as vessels may have arbitrary orientations and all must be equivalently characterised (see Section 3.3.1. of Depeursinge *et al.*¹⁴). In general, we are not interested in the local orientation of the pattern itself, but rather to its presence only, hence justifying the invariance to local rotations.

3.2 Directional Sensitivity

Most structures of interest in medical images are composed of directional components such as edges and corners. Filters may be sensitive to image directions, but not necessarily. This has a one-to-one relation with the circular/spherical symmetry of the filter¹⁶. We adopt the definition of directionality introduced in Tamura *et al.*⁴⁰. Formally, a non-directionally sensitive filter consists of a circular/spherical averaging, *i.e.* $g[\mathbf{k}]$ only depends on the radius $\|\mathbf{k}\|$. Directional sensitivity is important: such circularly/spherically symmetric filters cannot differentiate between blob and tubular structures (*e.g.* a small nodule and a vessel). An example of a non-directionally sensitive filter is the Laplacian of Gaussian (LoG) defined in Section 4.3 and in particular Eq. (4.2) (see Fig. 4.2). An example of a directionally sensitive filter is a Gabor filter defined in Section 4.5 (see Fig. 4.7).

3.3 Combining Directional Sensitivity and Invariance to Local Rotations

Ideally, filters should combine directional sensitivity with invariance to local rotations. A simple convolutional filter is invariant to local rotations (as defined in Section 3.1) if and only if the filter is directionally-insensitive, *i.e.* circularly symmetric (see Proposition 1 of Andrearczyk *et al.*²). Therefore, slightly more complex and non-linear filtering designs are needed to combine the two properties.

A suitable strategy to achieve invariance to local rotations with a directionally-sensitive filter is to (i) compute a pseudo rotation equivariant representation via a collection of rotated filter responses at uniformly sampled orientations and (ii) voxelwise orientation pooling: taking either the average or the max of the oriented responses, yielding the locally rotation invariant response maps h_{avg} or h_{max} , respectively.

The pseudo equivariant representation (i) is

$$\{f * g[\mathbf{R}\cdot]\}_{\mathbf{R} \in B}, \quad (3.1)$$

where \mathbf{R} is a rotation matrix and B is a uniformly sampled set of rotations (*e.g.* the set of right-angle rotations in $2D$). The process is illustrated in $2D$ in Fig. 3.1. It is worth noting that Eq. (3.1) can be efficiently computed using steerable filters, which obviates the need to reconvolve the image with rotated versions of the filters^{1,46}. In the particular case of separable filters and right angle rotations, the equivariant representation can be efficiently obtained using unidimensional filter flipping and permutation. This is described for the $2D$ and $3D$ cases in Appendix A.

Alternatively, an equivariant representation can be obtained by permuting the image orientation in the same manner, convolving the rotated image with the separable filters, and then rotating the response map back to the original orientation. This method is useful when transforms are performed using a standard software implementation, *e.g.* of discrete wavelet transforms, instead of using filter kernels directly.

As illustrated in Fig. 3.2, both methods are equivalent if and only if the filter kernels have an odd length, or odd dimensions for $2D$ and $3D$ filters. This can be achieved by appending a 0 to extend the kernel prior to filter flipping and permutation. Note that some implementations of convolution might require that a 0 is prepended instead, but we did not encounter this for MATLAB and scipy standard implementations. Filter kernels do not need to be altered when permuting the image orientation instead.

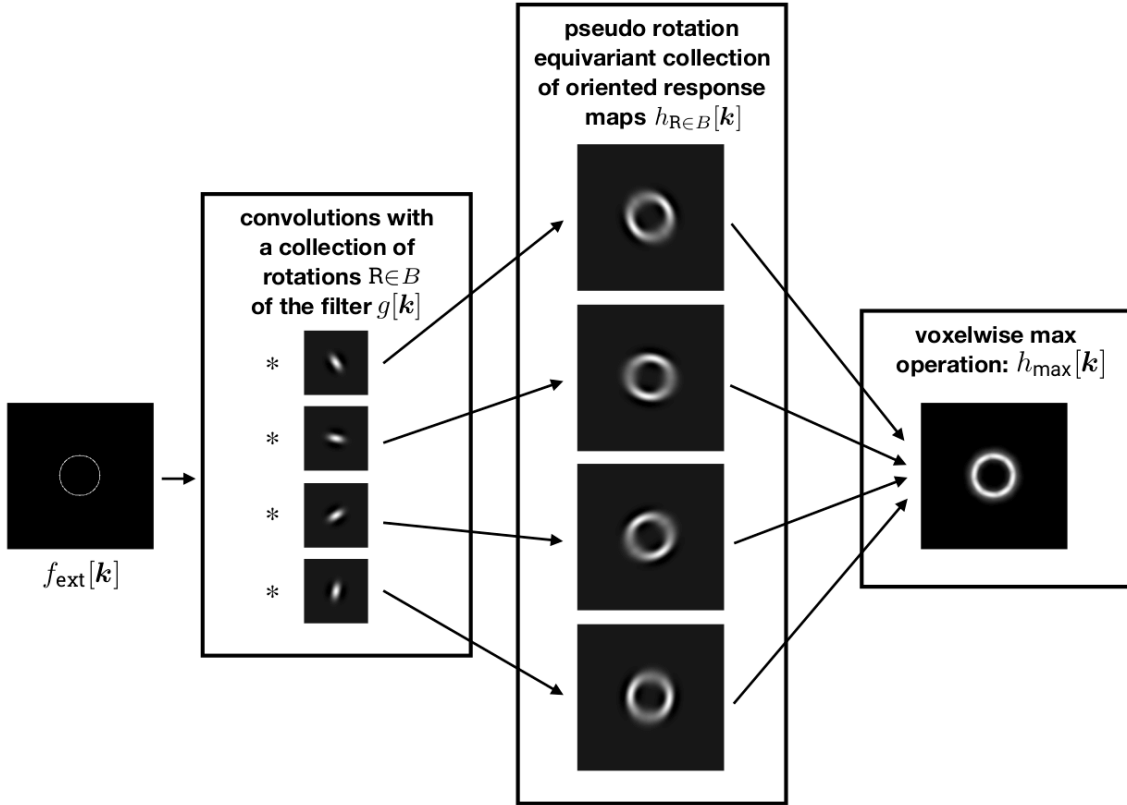


Figure 3.1 — Overview of the process to achieve filtering operations combining directional sensitivity with invariance to rotations.

Voxelwise orientation pooling (ii) can be done using either the average or the max over the elements of the equivariant representation provided by Eq. (3.1). Although commonly used, average orientation pooling strongly deteriorates the directional sensitivity of the filtering operation. For a large set of rotations, averaging is equivalent to filtering with a single circularly symmetric filter. The max response preserves the directional sensitivity of the filters and achieves invariance to local rotations^{1,10}. Taking the max response can be interpreted as “aligning” the filter locally to seek for the best match between the filter and the local image pattern (see Fig. 3.1).

It is worth noting that particular filtering methods allow combining directional sensitivity with invariance to rotations via the calculation of invariants. Notable examples are circular and spherical harmonics^{16,22,28}.

3.4 Spectral Coverage

The type of filters considered here change the frequency contents of an image in the Fourier domain, thereby altering its appearance. Each filter has a frequency profile that can be characterised in the Fourier domain. Filters are usually split into one of four categories based on their frequency profiles:

1. **All-pass filter:** An all-pass filter does not change the frequency content of the image. Such filters are rarely used. Examples are the identity filter, where the response image is identical to the input image, and a translation filter, where the response image is identical to the input, except for a shift.
2. **Low-pass filter:** A low-pass filter attenuates the high-frequency content of the image. Ex-

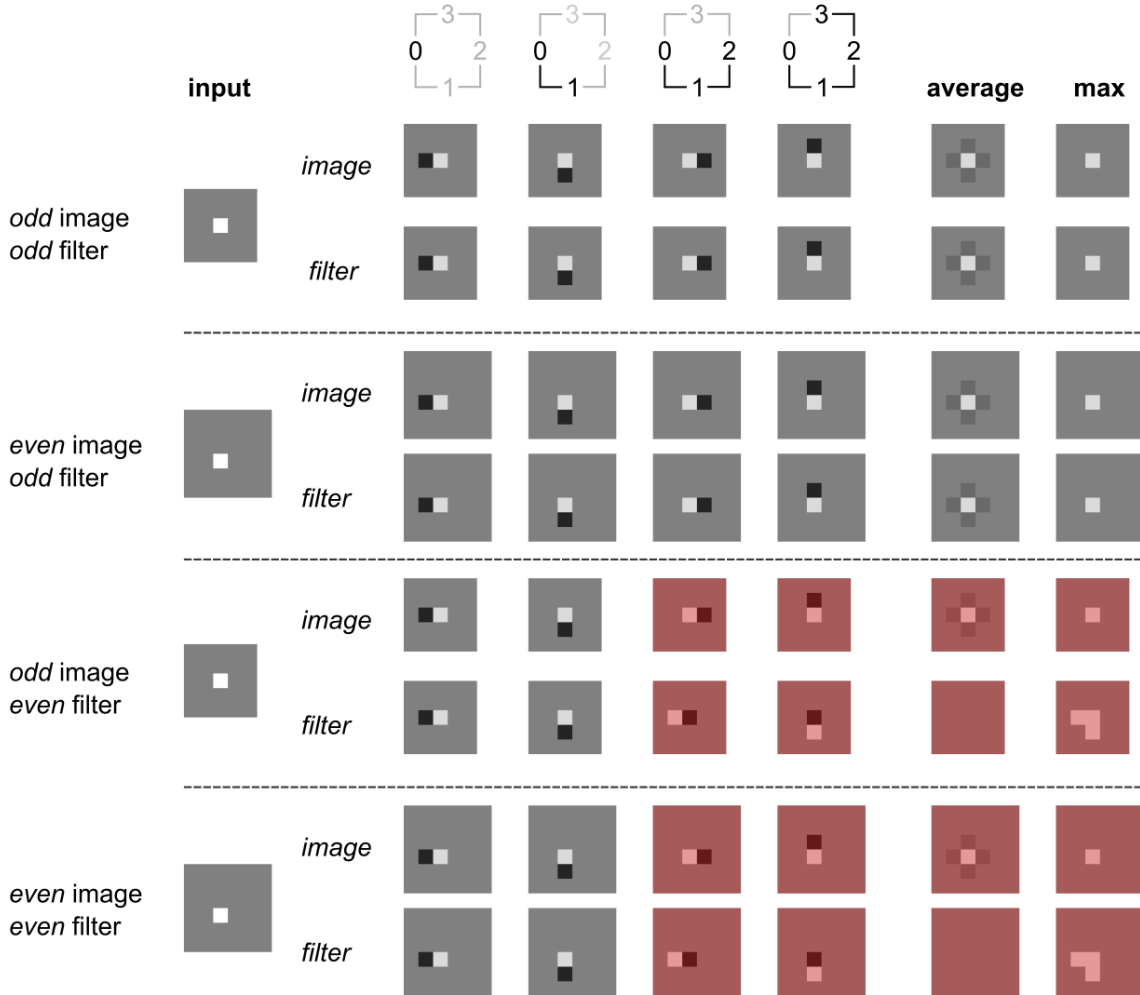


Figure 3.2 — Comparison of methods for achieving rotation equivariance for filtering. Two methods can be used: 1) *image*: rotating the image, convolution with the filter, and then rotating the response map back to the original orientation; 2) *filter*: through filter flipping and permutation. In this example, the image is either 5×5 pixels (*odd image*) or 6×6 pixels (*even image*). A single pixel with intensity 1.0 is located at (3, 3). Both 1D filter kernels in this example are based on the high-pass filter of the Haar wavelet. The *even* kernel is $[-1/\sqrt{2}, 1/\sqrt{2}]$, whereas the *odd* kernel is $[-1/\sqrt{2}, 1/\sqrt{2}, 0]$. The *even* and *odd* kernels are completely equivalent except for their length, and produce the same response maps and pooled images for the *rotation* method. For the *odd* kernel both methods to achieve rotational invariance are equivalent. However, for the *even* kernel several response maps (red overlay) can be seen to differ between *rotation* and *filter* methods. Hence we recommend that, when rotational invariance is desired, any even-sized kernel should be extended by appending a 0 to create an odd-sized kernel for the *filter* method. This behaviour moreover generalises to 2D and 3D filters, and is independent of whether the input image has an even or odd size.

amples are mean and Gaussian filters, which produce a smoothed version of the input image.

3. **High-pass filter:** A high-pass filter attenuates the low-frequency content of the image. High-pass filters may be used to sharpen an image, at the cost of amplifying noise.
4. **Band-pass filter:** A band-pass filter attenuates both high- and low-frequency content in an image in specific ranges. Band-pass filters can capture specific spectral signatures of patterns at relevant scales that are related to the problem at hand (*e.g.*, fibrosis, necrosis).

Examples of low-, high- and band-pass filtered images are shown in Fig. 4.11, respectively h_L^3 , h_H^1 and h_H^2 .

Chapter 4

Common Convolutional Filters in Radiomics

This section details the definitions of common convolutional filters used for radiomics. Guidance is also provided on how to set their parameters and to obtain appropriate features via matching aggregation functions.

4.1 Identification of Common Approaches

As a starting point, we listed the filtering techniques used in common radiomics software libraries. The considered libraries are TexRAD^a, Definiens⁴, PyRadiomics⁴⁴, CGITA²³, IBEX⁴⁸, CERR³, MAZDA³⁹, QIFE²¹, LIFEx³⁴, and QuantImage¹⁷.

4.2 Mean Filter

One of the simplest existing kernels is the mean filter that computes the average intensity over its $M \times \dots \times M$ spatial support^b as

$$g[\mathbf{k}] = \begin{cases} \frac{1}{M^D} & \text{if } k_1, \dots, k_D \in \left[-\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor\right], \\ 0 & \text{otherwise.} \end{cases} \quad (4.1)$$

Implementation Troubleshooting

- Ensure that filter support M is defined in voxel units.
- Check that the padding method is correctly applied (particularly mirror). An example of an outlier submission due to padding is shown in Fig. 4.1.

^a<http://texrad.com>, as of September 2019.

^bWe restrict the definition to odd values of M .

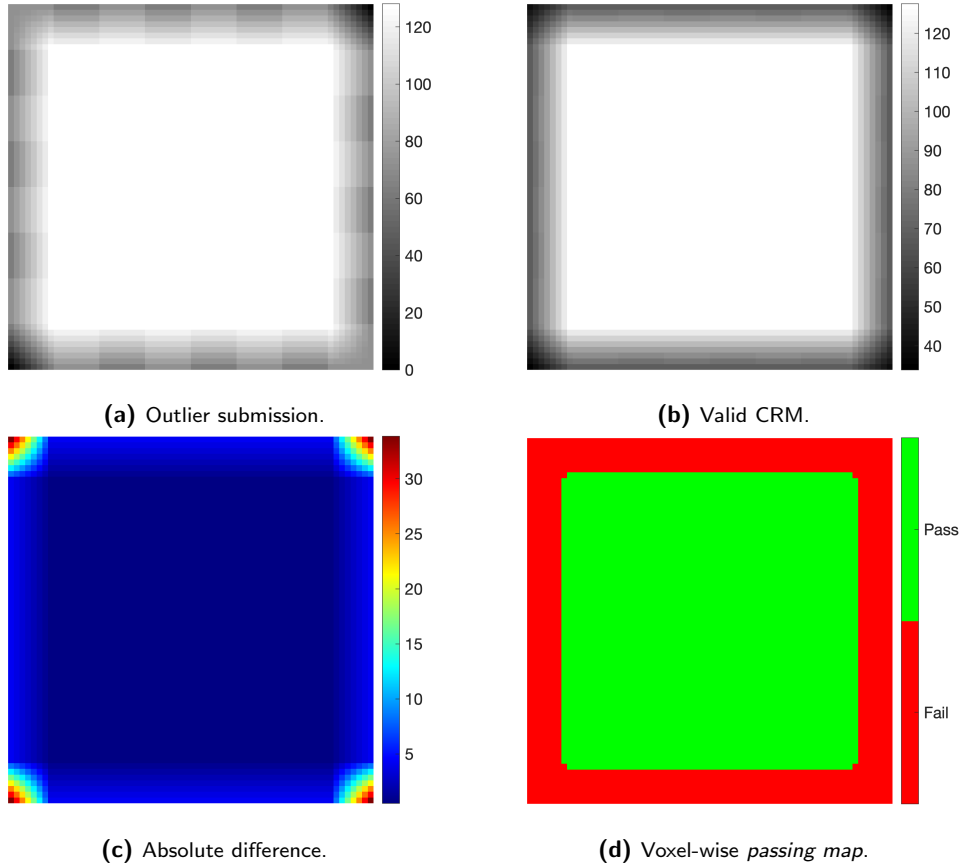


Figure 4.1 — Example of outlier discrepancy for mean filter test 1.a.1 (Table 5.1) caused by padding differences. The central 2D slice of the 3D volumes are visualised for: **(a)** an outlier submission, **(b)** the valid *consensus response map* (CRM) found, **(c)** the absolute difference between the outlier and consensus, **(d)** and voxel-wise passing map of this difference.

4.3 Laplacian of Gaussian

The Laplacian of Gaussian (LoG) is a band-pass and circularly/spherically symmetric convolutional operator. It is therefore invariant to local rotations but also directionally insensitive. Its profile $g[\mathbf{k}]$ only depends on the 1D radius $\|\mathbf{k}\|$ and corresponds to a radial second-order derivative of a D -dimensional Gaussian filter as

$$g_{\sigma}[\mathbf{k}] = -\frac{1}{\sigma^2} \left(\frac{1}{\sqrt{2\pi}\sigma} \right)^D \left(D - \frac{\|\mathbf{k}\|^2}{\sigma^2} \right) e^{-\frac{\|\mathbf{k}\|^2}{2\sigma^2}}, \quad (4.2)$$

where the standard deviation of the Gaussian σ controls the scale of the operator. Note that σ is implied to be in voxel units, i.e. $\sigma = \sigma^*/s$, with s the voxel spacing.

LoG filtering cannot be implemented with separable convolution and requires a full D -dimensional convolution. However, it can be approximated using a Difference of two Gaussians (DoG) when the ratio between their respective standard deviations σ_1 and σ_2 is $\sigma_1 = \frac{\sigma_2}{\sqrt{2}}$. Because Gaussian kernels are separable, DoG filtering can also be efficiently implemented using separable convolutions. With an adequate sequence of σ_i , a collection of LoGs can cover the entire image spectrum. In this case, they form wavelets and are often called Mexican hat, Ricker, or Marr wavelet.

The spatial support of the LoG is $(-\infty, \infty)$ (i.e. not compact). Because this would require a filter with infinite spatial support, the LoG filter is cropped in practice, usually based on the σ



Figure 4.2 — Examples of 2D LoG filters with $\sigma = 16$, $M = 129$ (left) and $\sigma = 8$, $M = 65$ (right).

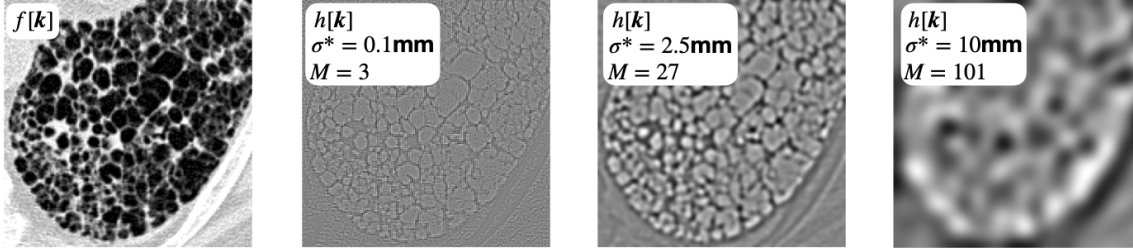


Figure 4.3 — Examples of image filtering with a LoG filter (the pixel spacing is 0.8mm, mirror boundary conditions used for the convolution) at various scales. The small scale ($\sigma^* = 0.1\text{mm}$) highlights tiny collagen fibers, whereas the larger scale ($\sigma^* = 10\text{mm}$) highlights larger image blobs or clusters present in the diseased lung tissue (honeycombing).

parameter. The 1D filter size is then

$$M = 1 + 2[d\sigma + 0.5],$$

with d the truncation parameter, σ in voxel units (e.g., $\sigma = 2.5$ voxels when it parameterised as $\sigma^* = 5$ mm with voxel spacing of 2 mm). As a consequence, the size of a (1D) LoG filter is at least $M = 1$ and will have an odd, integer value. $d = 4$ is commonly used for truncation.

The profile of a 2D LoG is illustrated in Fig. 4.2. The LoG filter can be used to enhance image blobs and ridges at a specific scale, controlled by σ^* . This is illustrated in Fig. 4.3.

Implementation Troubleshooting

- The scale parameter σ^* for the filter tests (Table 5.1) is defined in physical distance. Voxel dimensions for the digital phantoms are stored in millimeters in NifTI or DICOM headers. Many standard implementations assume that σ is defined in voxel units, so a conversion between physical and voxel units is required (e.g. *fspecial3* in Matlab).
- To test implementation of Eq. 4.2, check that the LoG kernel sums to approximately 0.

4.4 Laws Kernels

Laws kernels are a collection of five types of small 1D filters $g[k]$ ³⁰. They are combined using outer products to obtain 2D and 3D filters.

The first type is a low-pass kernel called *Level* for grey level averaging, which is available in two scales with a spatial support of 3 or 5 pixels:

$$g_{L3}[k] = \frac{1}{\sqrt{6}} \cdot [1, 2, 1], \quad g_{L5}[k] = \frac{1}{\sqrt{70}} \cdot [1, 4, 6, 4, 1].$$

The next kernels are all zero mean, which makes them insensitive to the average grey level. They

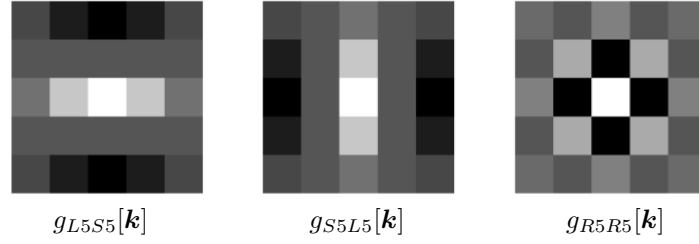


Figure 4.4 — Example of 2D Laws kernels.

will solely focus on spatial transitions between the values (*i.e.* texture). The four types of transitions covered are image *Edges*:

$$g_{E3}[k] = \frac{1}{\sqrt{2}} \cdot [-1, 0, 1], \quad g_{E5}[k] = \frac{1}{\sqrt{10}} \cdot [-1, -2, 0, 2, 1],$$

Spots and circular blobs:

$$g_{S3}[k] = \frac{1}{\sqrt{6}} \cdot [-1, 2, -1], \quad g_{S5}[k] = \frac{1}{\sqrt{6}} \cdot [-1, 0, 2, 0, -1],$$

Wave or undulation:

$$g_{W5}[k] = \frac{1}{\sqrt{10}} \cdot [-1, 2, 0, -2, 1],$$

and *Ripples*:

$$g_{R5}[k] = \frac{1}{\sqrt{70}} \cdot [1, -4, 6, -4, 1].$$

Laws 2D and 3D kernels are separable by design, and response maps are created by 1D kernel convolution along each image direction. Such response maps are referred to by their 1-D kernel names. For instance, a 2D response map called h_{L5S5} is obtained after first convolving the image with the g_{L5} kernel along the image lines (*i.e.* k_1) and a subsequent convolution of the image with the g_{S5} kernel along the image columns (*i.e.* k_2). Examples of 2D Laws kernels are shown in Fig. 4.4. Note that the above definitions for Laws kernels were normalised, and in this sense deviate from those originally defined by Laws himself³⁰.

4.4.1 Laws Texture Energy Images

Laws used his kernels to generate texture energy images³⁰. This is done in two steps. In the first step, a response map h is generated by convolving the image with a Laws kernel along each image direction, as described above. Then, a smoothed image is computed where the absolute intensities^c of voxels in h within Chebyshev distance δ of a centre voxel are summed to create an energy image h_{energy} :

$$h_{\text{energy}}[\mathbf{k}] = \frac{1}{W} \sum_{k_{0,1}=-\delta}^{\delta} \cdots \sum_{k_{0,d}=-\delta}^{\delta} |h[\mathbf{k} + \mathbf{k}_0]|,$$

where $\mathbf{k}_0 = (k_{0,1}, \dots, k_{0,d})$ and $W = (2\delta + 1)^N$ the number of voxels in the N -dimensional neighbourhood. In practice, h_{energy} can be computed using kernel convolutions by convolving $|h|$ with a $2\delta + 1$ element long 1D kernel with constant values $1/(2\delta + 1)$ along each of the image directions, *i.e.* a mean filter as described in Section 4.2.

Note that the definition given above deviates from the one given by Laws³⁰ by introducing the normalisation factor W . Laws moreover suggested using a sliding window of 15×15 pixels, which

^cIt is worth noting that whereas "energy" often involves the computation of squared quantities, the absolute value was proposed by Laws. The goal is to regroup negative and positive filter responses.

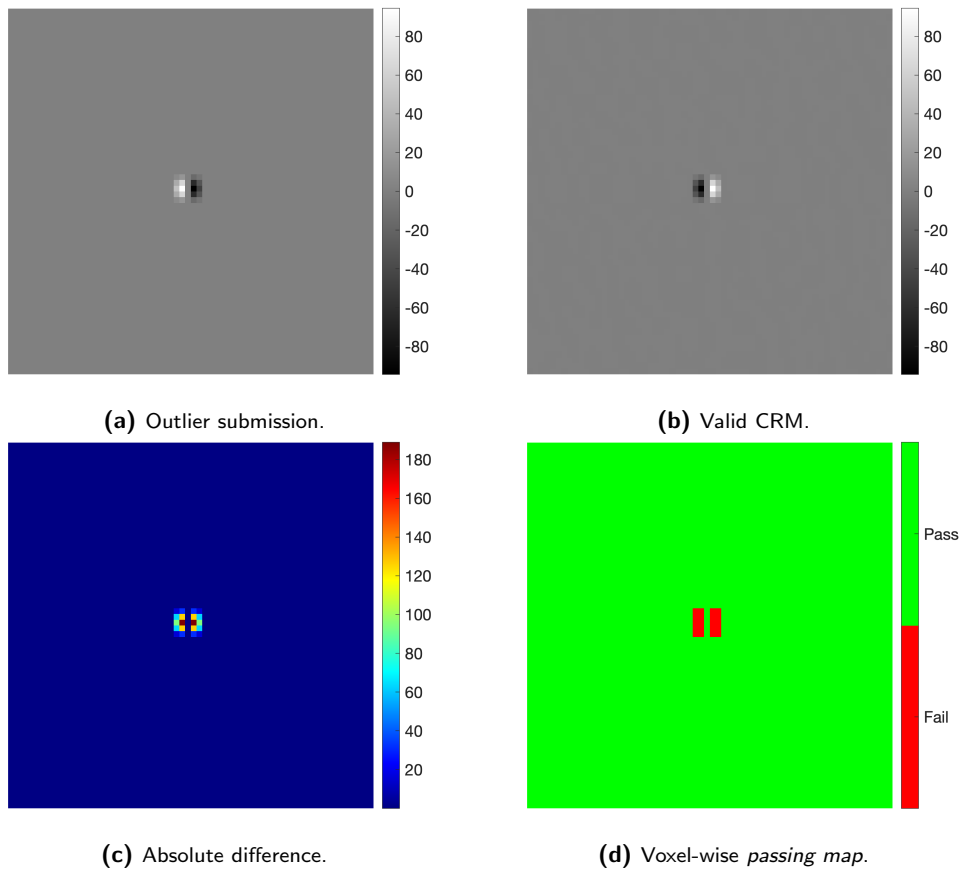


Figure 4.5 — Example of outlier discrepancy for Laws filter test 3.a.1. (Table 5.1) The central 2D slice of the 3D volumes are visualised for: **(a)** an outlier submission, **(b)** the valid *consensus response map* (CRM) found, **(c)** the absolute difference between the outlier and consensus, **(d)** and voxel-wise passing map of this difference. The two response maps **(a)** and **(b)** are identical apart from orientation.

corresponds to $\delta = 7$. We recommend that δ is chosen within the context of a given application.

To summarize, Laws filtering requires the following sequence of operations: (i) pad the input image, (ii) filter with a given Laws kernel, (iii) pad the response map and (iv) compute the energy image via the mean filter. We suggest using the same padding (see Section 2.2.1) used to compute the initial response of the Laws kernel to compute the energy image. Laws filtering is not rotation-invariant. However, since the kernels are separable, rotational invariance can be efficiently approximated using permutations and unidimensional filter flipping, followed by orientation pooling with *e.g.* the max (see Section 3.3 and Appendix A). In this case, we recommend computing the texture energy image after orientation pooling.

An example of image filtering with the g_{L5S5} kernel is shown in Fig. 4.6.

Implementation Troubleshooting

- Filter orientation and convolution direction is a key cause of discrepancy, as some Laws kernels are not symmetric. An example of an outlier response map for filter test 3.a.1 (Table 5.1) that appears to have a different orientation is shown in Fig. 4.5. Check that filter kernels are applied in the direction consistent with Section 1.2.1. Use the orientation phantom to check if your software orients the image as expected.
- Only one Energy Image should be calculated. This is performed after orientation pooling and on absolute intensity values.

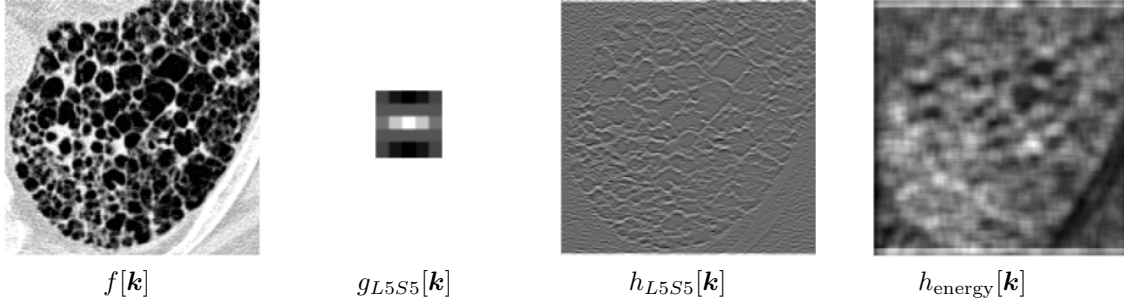


Figure 4.6 — Example of image filtering with the g_{L5S5} kernel. The zero padding boundary condition was used to calculate both h_{L5S5} and h_{energy} .

4.5 Gabor

Gabor filter banks allow for extracting multi-directional and multi-scale texture information via a systematic parcellation of the Fourier domain with elliptic Gaussian windows⁶ (see Fig. 3.2 of Depaepe *et al.*¹³). In the spatial domain, $2D$ Gabor kernels are complex Gaussian-windowed oscillatory functions defined as

$$g_{\sigma,\lambda,\gamma,\theta}[\mathbf{k}] = e^{-\frac{\tilde{k}_1^2 + \gamma^2 \tilde{k}_2^2}{2\sigma^2} + j \frac{2\pi \tilde{k}_1}{\lambda}}, \quad (4.3)$$

where σ controls the scale of the filter (standard deviation of the Gaussian envelope) and λ is the wavelength (*i.e.* inverse of the frequency $F = \frac{1}{\lambda}$ of the oscillations). Both σ and λ are implied to be in voxel units, *i.e.* $\sigma = \sigma^*/s$ and $\lambda = \lambda^*/s$, with s the voxel spacing, and σ^* and λ^* defined in mm. γ is the spatial aspect ratio (*i.e.* the ellipticity of the support of the filter as $\gamma = a_1/a_2$), $(\tilde{k}_1, \tilde{k}_2) = R_\theta \mathbf{k}$ defines the radial and orthoradial elliptic Gaussian axes at the orientation θ via the $2D$ rotation matrix $R_\theta = \begin{pmatrix} \cos \theta & \sin \theta \\ \sin \theta & -\cos \theta \end{pmatrix}$. As a convention, we define θ to turn clockwise in the axial plane (k_1, k_2) . The considered coordinate system follows the conventions introduced in Section 1.2.1. It is depicted in Fig. 4.7 along with an example of a $2D$ Gabor filter seen in the spatial domain.

v4: Clarified units of σ and λ parameters.

In practice, the Gabor function is used as a filter kernel and the spatial frequency bandwidth of the filter needs to be defined. Petkov *et al.*³⁵ established the relation between the half-response spatial frequency bandwidth F_b (in units of octaves) and the ratio σ/λ as

$$F_b = \log_2 \left(\frac{\frac{\sigma}{\lambda} \pi + \sqrt{\frac{\ln 2}{2}}}{\frac{\sigma}{\lambda} \pi - \sqrt{\frac{\ln 2}{2}}} \right), \text{ and inversely } \frac{\sigma}{\lambda} = \frac{1}{\pi} \sqrt{\frac{\ln 2}{2}} \cdot \frac{2^{F_b} + 1}{2^{F_b} - 1}. \quad (4.4)$$

Gabor filters can then be constructed at multiple scales and orientations to explore the spectrum of patterns in an image. Bianconi *et al.*⁶ proposed to extract response maps at multiple orientations $\{\theta_1, \dots, \theta_P\}$ and frequencies $\{F_1, \dots, F_Q\}$ along with γ are defined to cover all directions and scales up to the maximum frequency $F_Q = \frac{1}{\lambda_Q}$. It is worth noting that when using a suitable sequence of scales, Gabor filters can also satisfy the wavelet admissibility condition (*i.e.* referred to as ‘‘Gabor wavelets’’) and can, in this particular case, fully cover the Fourier domain. Finally, since $g_{\sigma,\lambda,\gamma,\theta}[\mathbf{k}]$ is complex, the modulus of the associated response map can be used before aggregation and feature calculation as $|h[\mathbf{k}]| = |(g_{\sigma,\lambda,\gamma,\theta} * f)[\mathbf{k}]|$. An example of filtering with Gabor is depicted in Fig. 4.8.

As with Laplacian-of-Gaussian filters the spatial support of Gabor filters is $(-\infty, \infty)$ (*i.e.* not compact). Because this would require a filter with infinite spatial support, the Gabor filter is

v4: Clarified spatial extent of Gabor filter support.

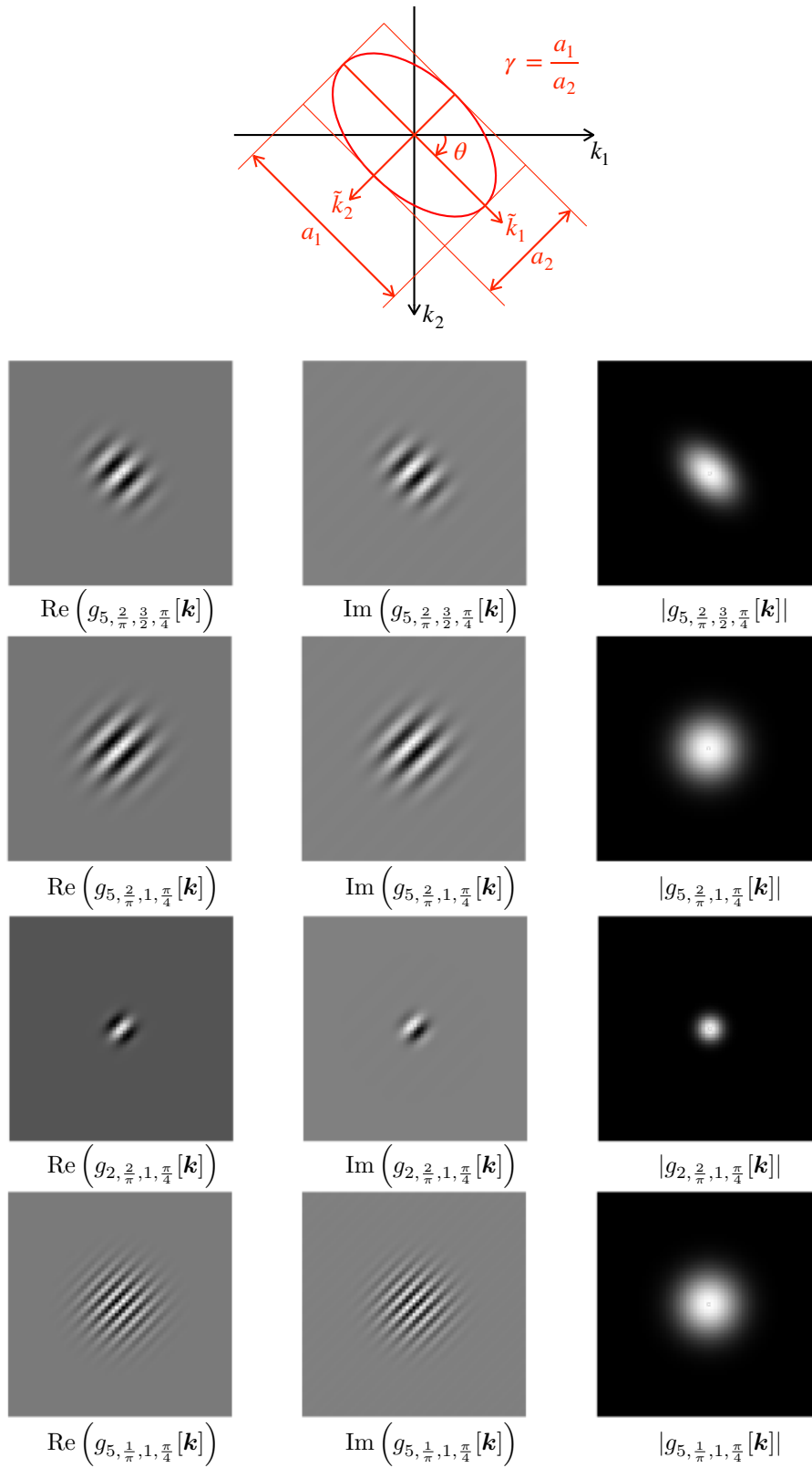


Figure 4.7 — Coordinate system and examples of 2D Gabor filters in the spatial domain computed on 65×65 grids with a pixel spacing of 0.8mm.

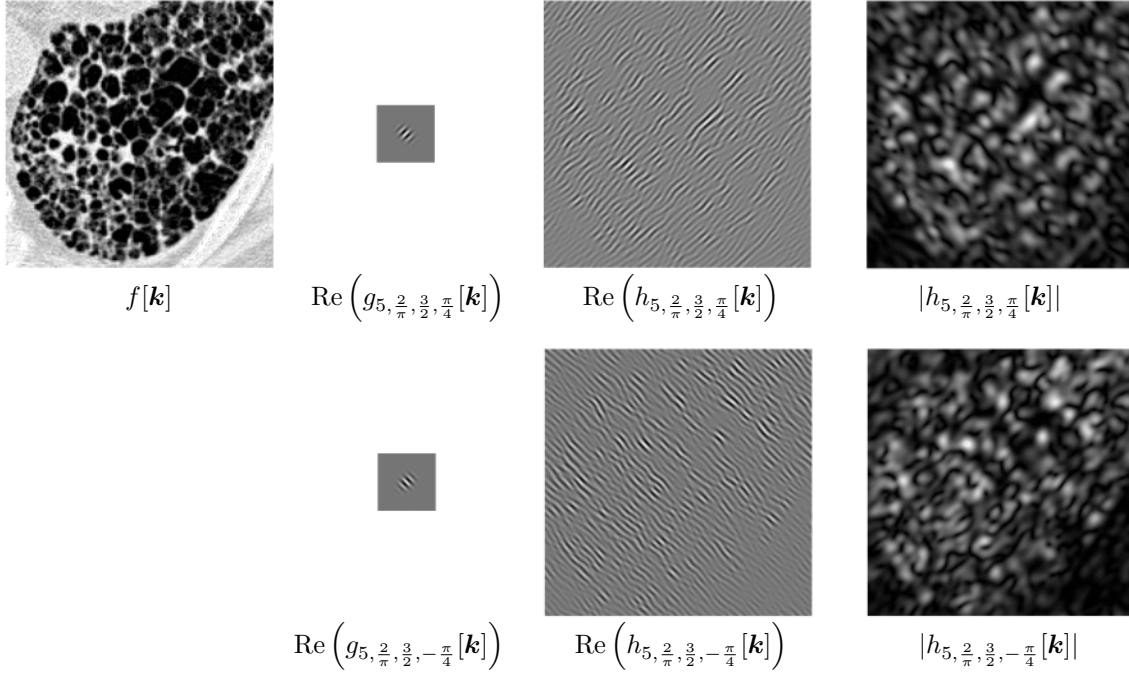


Figure 4.8 — Example of image filtering with Gabor filters (the pixel spacing is 0.8mm, mirror boundary conditions used for the convolution). Collagen junctions oriented at $\theta = -\frac{\pi}{4}$ (top) and $\theta = \frac{\pi}{4}$ (bottom) are highlighted.

cropped in practice, based on the σ and γ parameters. The 1D filter size is then:

$$M = \begin{cases} 1 + 2[d\sigma + 0.5] & \text{if } \gamma \leq 1, \\ 1 + 2[d\gamma\sigma + 0.5] & \text{if } \gamma > 1. \end{cases} \quad (4.5)$$

with d the truncation parameter, and σ and γ in voxel units. Note that it is possible to more closely truncate the spatial support by accounting for the orientation of the filter, at the cost of increased complexity.

Gabor filters are not rotation-invariant and are therefore best suited for applications where the absolute feature orientation is meaningful. Rotation equivariance and invariance can be approximated by combining the response maps of several elements of the Gabor filterbank (see Section 3.3).

The 3D extension of Gabor filters can be found in Qian *et al.*³⁸. However, no recommendations are provided in the 3D case to sample scales and orientations for further feature extraction. One simple option to achieve (*i.e.* approximate) 3D image analysis is to extract 2D Gabor features as recommended above in the three orthogonal planes of the image frame of reference, followed by an averaging of the response maps over the three planes.

4.6 Wavelets

Wavelets form a large category of filtering methods based on a collection of high-pass and low-pass filters that are designed to cover the entire image spectrum³² (see Section 3.4). Pairwise combinations of one high- and one low-pass filter result usually in a sequence of band-pass response maps (*i.e.* wavelet coefficients) with a factor of 2 between their scales and one remaining low-pass response map. Two properties must be considered when implementing wavelet transforms, namely *decimation* and *separability*. Decimation relates to the downsampling operation of the response maps and is compared in Sections 4.6.1 and 4.6.2. Separable and non-separable wavelets concern

the separability of high- and low-pass filters and are detailed in Sections 4.6.3 and 4.6.4.

4.6.1 Decimated Transform

The decimated transform is not redundant and allows coding images with a minimal number of coefficients. However, the response maps containing the coefficients are iteratively decimated, which means that their size decreases throughout the levels of the decomposition, leading to a lack of translation invariance.

For instance, with separable wavelets (see Section 4.6.3), the image $f[\mathbf{k}]$ is first convolved with a high-pass filter g_H and a matching low-pass filter g_L along each image direction. In $2D$, this yields four response maps: h_{LL} , h_{LH} , h_{HL} and h_{HH} . All response maps are then downsampled by a factor of two in all directions to become h_{LL}^1 , h_{LH}^1 , h_{HL}^1 and h_{HH}^1 . This concludes the first iteration of the discrete wavelet transform. For the next iterations, the low-pass coefficients h_{LL}^j are subsequently convolved with g_H and g_L along each image direction and downsampled. This means that the h_{LL}^j response map of each decomposition level is used as input image for the next level $j + 1$. It is worth noting that h_{LL}^j is traditionally discarded when the wavelet transform is used for image compression and reconstruction. Because the response maps h_{LL}^j are downsampled j times, this has the same effect as dilating (*i.e.* upsampling) the filters by 2^j in Eq. (4.6). The downsampling of the response maps yields the wavelet coefficients of iteration $j + 1$. The response maps resulting from the decimated separable wavelet decomposition is illustrated in Fig. 4.9 for the $2D$ case. Although illustrated in the context of separable wavelets, decimated transforms can also be used with non-separable wavelets.

It is worth noting that the convolution is, in this case, modified because the shifts are restricted to the resolution of the j times downsampled response maps¹¹. In addition, this must be taken into account when aggregating the response maps (see Section 1.3), where the ROI mask \mathbf{M} must also be downsampled to match the dimensions of the corresponding response maps $h^j[\mathbf{k}]$.

4.6.2 Undecimated Transform

The undecimated transform, also called *stationary* transform, yields a translation-invariant decomposition by obviating the downsampling steps required by the decimated transform. Although the transform becomes redundant (*i.e.* it yields more coefficients than strictly required for a perfect reconstruction), it is better suited to our case where we are not interested in image coding but rather image analysis.

The separable undecimated transform involves upsampling of the wavelet filters g_H^j and g_L^j to achieve the multiscale decomposition via the matched-size filters in Eq. (4.6). While others exist, a common upsampling approach is to use the *à trous* algorithm²⁰. The *à trous* algorithm involves inserting zeros, or holes, into the filter kernel. For example, the high-pass kernel of the Haar wavelet is $[-1/\sqrt{2}, 1/\sqrt{2}]$. The first level decomposition of this high-pass kernel is $[-1/\sqrt{2}, 0, 1/\sqrt{2}]$, or alternatively $[-1/\sqrt{2}, 0, 1/\sqrt{2}, 0]$. The second level decomposition of the same kernel is formed by inserting zeros between the values of the first level kernel, forming $[-1/\sqrt{2}, 0, 0, 0, 1/\sqrt{2}]$, or alternatively $[-1/\sqrt{2}, 0, 0, 0, 1/\sqrt{2}, 0, 0, 0]$. Both alternatives are valid, but result in different response maps. Popular standard implementations in `MATLAB` and `pywavelets` use the second alternative. We therefore recommend using the second alternative for reproducibility.

v4: Clarified implementation of the *à trous* algorithm.

The response maps $h^j[\mathbf{k}]$ produced through undecimated transform have the same dimensionality as the input image f and are simply obtained via the convolution of $f[\mathbf{k}]$ with the filters g_H^j and g_L^j along each image direction. In $2D$, this yields four response maps for every iteration: h_{LL}^j , h_{LH}^j , h_{HL}^j and h_{HH}^j .

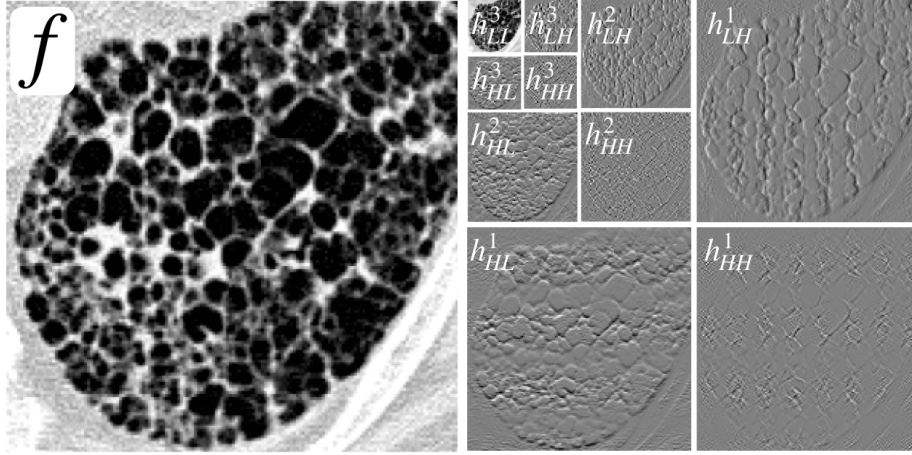


Figure 4.9 — Response maps from the 2D decimated separable wavelet transform of the input image f . The first three iterations are shown when using the Haar wavelet.

The response maps resulting from the undecimated separable wavelet decomposition is illustrated in Fig. 4.10 for the 2D case. Again, although illustrated here for separable wavelets, non-decimated transforms can also be used with non-separable wavelets.

4.6.3 Separable Wavelets

The discrete separable wavelet transform yields a collection of 1D wavelet kernels obtained from J dilations of one unique mother wavelet function, which is a high-pass filter $g_H[k]$ ¹¹. The remaining low frequencies are covered by a low-pass filter g_L called scaling function. When considering dyadic dilations, the scale of the kernels g_H and g_L are indexed by j as

$$g^j[k] = 2^{j/2}g[2^j k]. \quad (4.6)$$

The distinctive property of the separable wavelet transform is to use the separable convolution. The latter is computationally efficient but for radiomics has two distinct disadvantages: it is not rotationally invariant, and only strictly separable wavelets can be used (see Section 2.1).

Directionality Considerations

In 3D, the convolution of the low-pass and high-pass filters in the three directions of space yields eight different wavelet response maps: h_{LLL} , h_{LLH} , h_{LHL} , h_{LHH} , h_{HLL} , h_{HLH} , h_{HHL} and h_{HHH} . Within the image frame of reference^d, these three directions are k_1 , k_2 and k_3 . For example, let us consider the h_{LHL} response map: a low-pass filter is applied in the k_1 direction, a high-pass filter is applied in the k_2 direction and a low-pass filter is applied in the k_3 direction (see Section 2.1).

Haar

The simplest separable wavelet is the Haar wavelet. The Haar wavelet and scaling function form the simplest admissible function pair and are defined as³¹

$$g_{H,\text{Haar}} = \left[\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right], \quad g_{L,\text{Haar}} = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right].$$

^dThe image frame of reference of every image set (e.g. a scan) of a study should have the same common orientation relative to the patient reference frame, see Section 1.2.1.

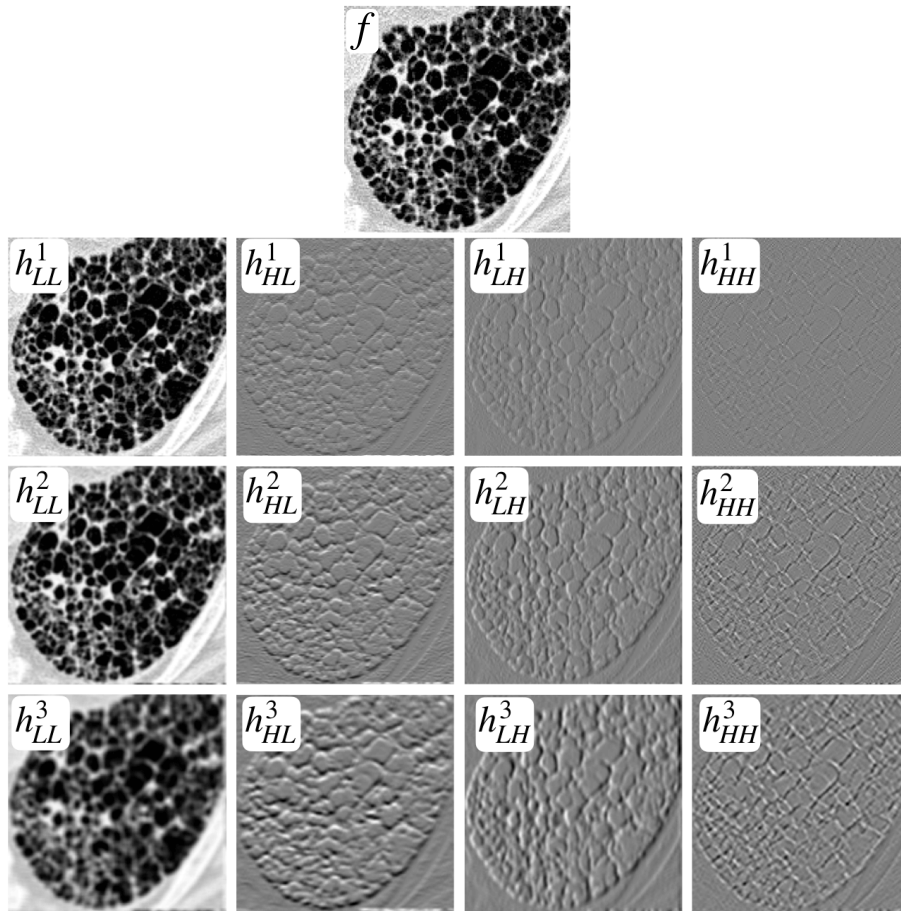


Figure 4.10 — Response maps from the 2D undecimated separable wavelet transform of the input image f . The first three iterations are shown when using the Haar wavelet.

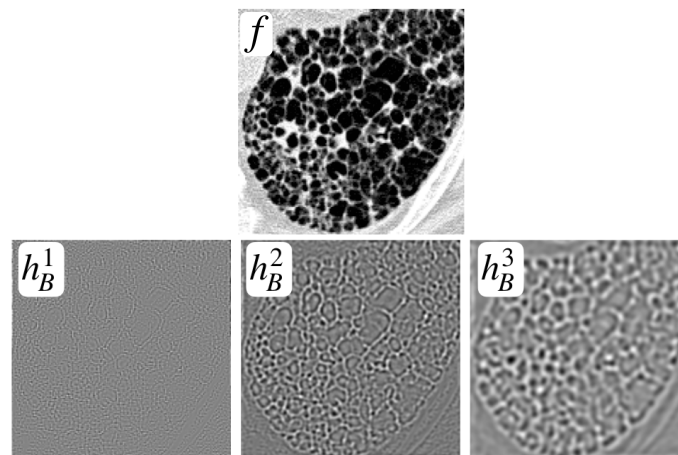


Figure 4.11 — Response maps from the 2D undecimated non-separable wavelet transform of the input image f . The first three iterations are shown when using the Simoncelli wavelet.

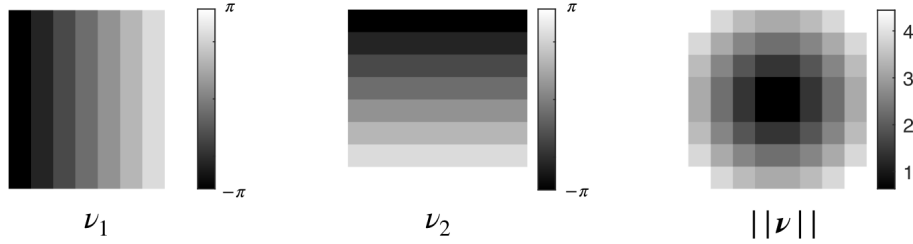


Figure 4.12 — Coordinate grids in voxel dimensions for for $N = 8$.

Daubechies

Daubechies wavelets and scaling functions are characterised by their number of vanishing moments p , *i.e.* their ability to approximate polynomials of order up to $p-1$ ¹¹. For $p = 1$ Daubechies kernels are equivalent to Haar. Kernel values of wavelet and scaling functions for $p > 1$ can be obtained from the PyWavelets website^e.

Other

There exists a large number of other separable wavelet/scaling function pairs (*e.g.* Meyer, Coiflets), each targeting different objectives in terms of signal analysis. The kernel values of most common wavelets can be obtained from the Wavelet Browser of PyWavelets^f.

4.6.4 Non-separable Wavelets

As motivated in Section 3.1, filters (and more generally texture operators) should be invariant or equivariant to local rotations in most cases. However, with the exception of the Gaussian filter, all separable filters (including separable wavelets) are not invariant/equivariant to rotations. Therefore, it is interesting to consider non-separable wavelets to achieve isotropic image analysis. The starting point for the definition of non-separable wavelets is a single (unidimensional) radial profile in the Fourier domain as $\hat{g}[\boldsymbol{\nu}]$, where \hat{g} is a function of $\|\boldsymbol{\nu}\|$ defining the radial coordinate⁴³. As a consequence, these functions are circularly symmetric and can be further combined with directional analysis such as the Riesz transform (Section 4.7).

v4: Extended description of non-separable wavelets.

Non-separable wavelets are easily implemented directly in the Fourier domain using their unidimensional radial profile. The support of the filter in the Fourier domain is the same as the considered $N_1 \times \dots \times N_D$ image f to allow for efficiently computing the convolution as a simple Hadamard product via Eq. (2.3). Along dimension i , each coordinate ν_i is contained in $[-\frac{N_i}{2}, \frac{N_i}{2}]$ which is centered around the null frequency $\boldsymbol{\nu} = \mathbf{0}$. The maximum coordinate value $\frac{N_i}{2}$ will be assigned to the value $\nu_B = \pi$ radians/sample, which is called the normalized Nyquist frequency. The unidimensional radial coordinate is then obtained as $\|\boldsymbol{\nu}\| = \sqrt{\nu_1^2 + \nu_2^2 + \dots + \nu_D^2}$. It is worth noting that the value of $\|\boldsymbol{\nu}\|$ will exceed ν_B in the corners of the image. These “corner” values should be ignored as the corresponding frequencies can only be measured along diagonal directions, yielding anisotropic image analysis. They are ignored by construction for all proposed radial wavelets, *e.g.* (4.7) and (4.8).

In order to implement non-separable wavelets in practice, we recommend defining a coordinate grid in the Fourier domain, where each dimension ν_i is sampled in the interval $[-\pi, \pi]$ with a step of $\frac{2\pi}{N_i}$. Coordinate grids in voxel dimensions are illustrated in Fig. 4.12 for $N = 8$.

Whereas wavelet functions are usually defined as a cascade of high- and low-pass filters, here we focus on the band-pass filters corresponding to the consecutive subbands (*i.e.* response maps)

^e<http://wavelets.pybytes.com/family/db/>, as of July 2018.

^f<http://wavelets.pybytes.com>, as of July 2020.

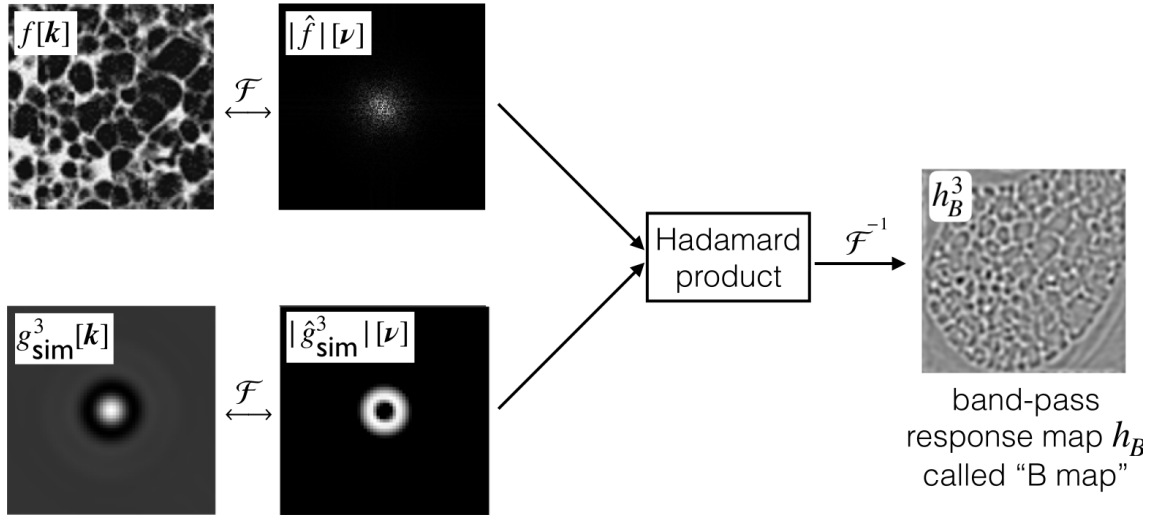


Figure 4.13 — Convolution in the Fourier domain using a Hadamard product. Complex modulus of the Fourier images are shown only for display purposes: the Hadamard product must be done with complex values.

that would result from combined high- and low-pass filtering (see introductory paragraph of Section 4.6). In other words, these response maps can be directly obtained using a corresponding band-pass filter $h_B[\mathbf{k}]$, without the need of re-convolving with previous iterations of the wavelet transform. The resulting response maps are referred to as “B maps”.

The simplest non-separable and circularly symmetric wavelet is the Shannon wavelet⁴³ characterised by the following mother function

$$\hat{g}_{\text{sha}}[\boldsymbol{\nu}] = \begin{cases} 1 & \text{if } \frac{\nu_B}{2} < \|\boldsymbol{\nu}\| \leq \nu_B, \\ 0 & \text{otherwise.} \end{cases} \quad (4.7)$$

Eq. (4.7) corresponds to the sinc wavelet in the spatial domain, which has the disadvantage of having large spatial supports. Therefore, an interesting alternative is the smoother and more compactly supported (in space) Simoncelli wavelet³⁶. Its band-pass function is defined as

$$\hat{g}_{\text{sim}}[\boldsymbol{\nu}] = \begin{cases} \cos\left(\frac{\pi}{2} \log_2\left(\frac{2\|\boldsymbol{\nu}\|}{\nu_B}\right)\right) & \text{if } \frac{\nu_B}{4} \leq \|\boldsymbol{\nu}\| \leq \nu_B, \\ 0 & \text{otherwise.} \end{cases} \quad (4.8)$$

An easy way to implement several consecutive iterations of such a non-separable wavelet transform is to start from the initial definition of the band-pass filter (the one with highest frequency band, *e.g.* Eq. (4.8)), and constructing the filter with half ν_B at every consecutive step. Concretely, the bandpass filter at a given scale level j is obtained by replacing ν_B by $\frac{\nu_B}{2^j}$ in (4.7) or (4.8). The response map can then be computed by taking the Hadamard product of the bandpass filter and the image in Fourier domain, and then performing an inverse Fourier transform. This process is illustrated in Fig. 4.13.

The response maps resulting from the undecimated non-separable wavelet decomposition is illustrated in Fig. 4.11 for the 2D case. An example of a Simoncelli wavelet is shown in Fig. 4.14. Other alternatives can be found in Table 1 of Unser *et al.*⁴³.

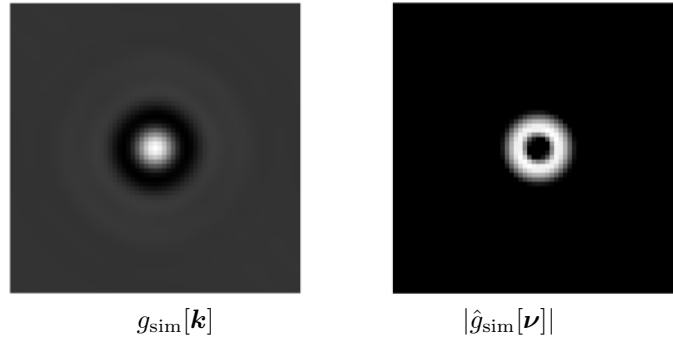


Figure 4.14 — Example of a 2D band-pass Simoncelli wavelet g_{sim} after three decomposition levels, computed on a 66×66 grid. The wavelet is shown in the spatial (left) and in the Fourier (right, complex modulus shown) domains.

4.6.5 Wavelets: Considerations for Radiomics

To summarise, undecimated non-separable wavelets have the advantage of yielding isotropic (*i.e.* rotation invariant/equivariant, see Section 3.3) and translation equivariant image analysis. Moreover, they yield only one response per decomposition level (*i.e.* one per scale), which significantly reduces the number of radiomics features when compared to their separable counterpart. Separable wavelets were mostly designed for image coding, which has very different design constraints. Such wavelets yield a large collection of response maps that are biased towards image axes and lack rotational invariance. While it is possible to make them approximately rotation invariant using orientation pooling over equivariant right angle representations as suggested in Section 3.3 and Appendix A, we do not recommend to use this orientation pooling procedure for decomposition levels larger than 1. Applying orientation pooling after every convolution operation of a multi-level separable wavelet decomposition would result in an overly complicated algorithm, yielding only approximate rotational invariance where the granularity of the approximation depends on the number of group elements (*e.g.* right angles only or using more angular samples) in the used equivariant representation. Therefore, we recommend not to use separable wavelet for rotation invariant image analysis, or to limit the wavelet decomposition to only one level. Isotropic non-separable wavelets such as the Simoncelli wavelet allow achieving multi-level, truly isotropic and rotation invariant image analysis by design. When the directionality of image patterns of interest (*e.g.*, tumour margin, vessels) is expected to be important, aligned directional wavelet filters such as Riesz (Section 4.7) that allow combining directional sensitivity with local invariance to rotations can be considered (see Section 3.3).

v4: Added recommendation to not perform decomposition using separable wavelets beyond the first level.

4.7 Riesz

Whereas non-separable wavelets like Simoncelli constitute an excellent first solution for using wavelets in radiomics studies, they are also circularly symmetric and therefore cannot characterise directional patterns. To address directionality of patterns, these non-separable circularly symmetric wavelets can be advantageously combined with directional analysis methods such as directional derivatives³⁷ or spherical harmonics.

v4: Extended description of the Riesz transform.

An elegant approach to obtain features measuring directional transitions between pixel values is to use image derivatives. Besides being simple to compute, these have the advantage of being interpretable (at least for the first- and second-order), which makes them attractive for understanding the meaning of the texture measures in a particular medical or biological application context. For instance, the first-order derivative is called the gradient and informs about the slope

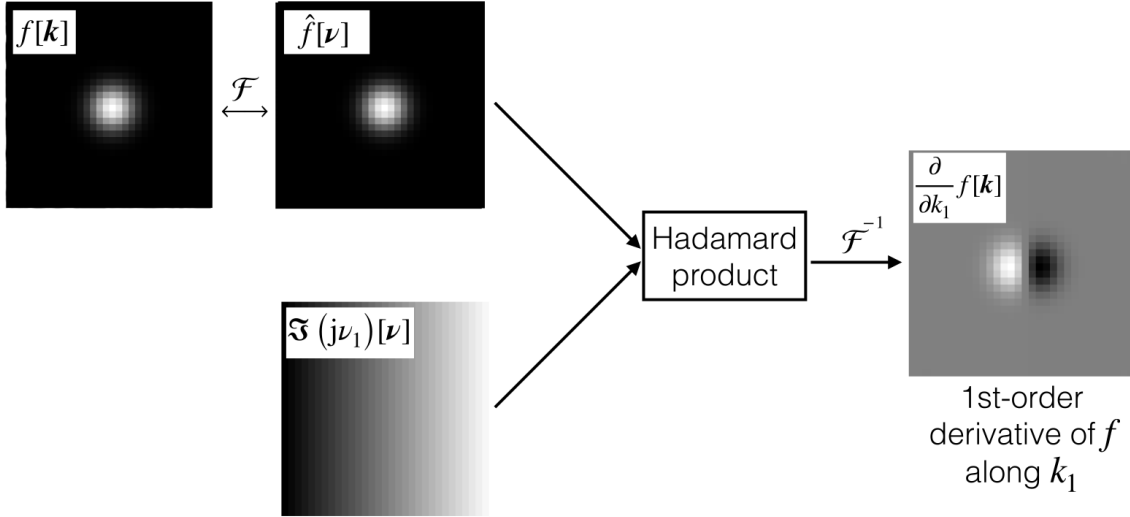


Figure 4.15 — Deriving an image in the Fourier domain (illustrated with a Gaussian function). The notation $\Im(\cdot)$ denotes the imaginary part.

of the transitions in the image, *e.g.* sharp versus smooth transitions. Second-order derivatives are called Hessian and quantify the curvature of the transitions. It is worth noting that the spatial scale of these transitions will be controlled by a preliminary isotropic filtering step based on *e.g.* a Gaussian smoother (low-pass) or a Simoncelli wavelet (band-pass).

An interesting option to compute the derivatives is to do so in the Fourier domain¹³. This also provides the opportunity to easily compute higher-order image derivatives of order l as

$$\frac{\partial^l}{\partial k_i^l} f[\mathbf{k}] \xleftrightarrow{\mathcal{F}} (j\nu_i)^l \hat{f}[\boldsymbol{\nu}], \quad (4.9)$$

where $1 \leq i \leq D$. It can be noticed that differentiating an image along the direction k_i only requires multiplying its Fourier transform by $j\nu_i$. Computing l^{th} -order derivatives has an intuitive interpretation (*e.g.*, gradient for $l = 1$, curvature for $l = 2$). Let us illustrate this by differentiating a simple 2D Gaussian function of dimension $N = 32$ (see Fig. 4.15). A Gaussian function $f[\mathbf{k}]$ is first transformed in the Fourier domain resulting in $\hat{f}[\boldsymbol{\nu}]$. It is worth noting that the Fourier transform of a Gaussian is real-valued and also a Gaussian. In parallel, the derivative kernel $j\nu_1$ can be simply defined by multiplying the Fourier coordinate grid ν_1 (see Fig. 4.12) by the imaginary unit j . When using the Hadamard product in Fourier and then back to the space domain (*i.e.* convolution operation as depicted in Fig. 4.13), the result corresponds to the derivative of f along k_1 , $\frac{\partial}{\partial k_1} f[\mathbf{k}]$.

Unfortunately, a pure image derivative filter as computed in Eq. (4.9) is high-pass (because multiplied by ν_i) and accentuates high frequencies along k_i . Therefore, it is desirable to implement image derivatives as all-pass filters instead of high-pass, by simply normalising the derivative kernel $j\nu_i$ by the coordinate norm $\boldsymbol{\nu}$. This is exactly what the first-order Riesz transform yields $\mathcal{R}\{f\}[\mathbf{k}]$ as⁴²

$$\mathcal{R}\{f\}[\mathbf{k}] = \begin{pmatrix} \mathcal{R}_1\{f\}[\mathbf{k}] \\ \vdots \\ \mathcal{R}_D\{f\}[\mathbf{k}] \end{pmatrix} \xleftrightarrow{\mathcal{F}} -j \frac{\boldsymbol{\nu}}{\|\boldsymbol{\nu}\|} \hat{f}[\boldsymbol{\nu}]. \quad (4.10)$$

It can be noticed that dividing the Fourier representation by the norm of $\boldsymbol{\nu}$ transforms Eq. (4.9) in D all-pass operators \mathcal{R}_i . Eq. (4.10) can be used to compute first-order directional derivatives (*i.e.*

gradient-like for characterising image edges). However, higher-order derivatives can be relevant for radiomics studies (*e.g.* second-order or Hessian-like for characterising image ridges like tumour margin or vessels).

A qualitative comparison between classical image derivatives as defined in (4.9) and Riesz-based image derivatives (4.10) is proposed in Fig. 4.16 for both a Gaussian image (f_1) and a white noise image (f_2).

For a fixed (maximal) order L , the collection of higher-order all-pass image derivatives are defined in the Fourier domain as

$$\widehat{\mathcal{R}}^{\mathbf{l}}\{\hat{f}\}[\boldsymbol{\nu}] = (-j)^L \sqrt{\frac{L!}{l_1! \cdots l_D!}} \frac{\nu_1^{l_1} \cdots \nu_D^{l_D}}{(\nu_1^2 + \cdots + \nu_D^2)^{L/2}} \hat{f}[\boldsymbol{\nu}], \quad (4.11)$$

which yields a total of $\binom{L+D-1}{D-1} = \frac{(L+D-1)!}{L!(D-1)!}$ all-pass filters for all combinations of the elements l_i of the vector \mathbf{l} as $|\mathbf{l}| = l_1 + \cdots + l_D = L$. The collection of Riesz operators $\mathcal{R}^{(l_1, l_2, \dots, l_D)}$ of order L is denoted by \mathcal{R}^L . For instance, in $3D$ and with $L = 2$, the element $\widehat{\mathcal{R}}^{(0,2,0)}\{f\}[\boldsymbol{\nu}]$ corresponds qualitatively to a second-order derivative of f along the direction k_2 and we have

$$\widehat{\mathcal{R}}^{(0,2,0)}\{f\}[\boldsymbol{\nu}] = \frac{-\nu_2^2}{\nu_1^2 + \nu_2^2 + \nu_3^2} \hat{f}[\boldsymbol{\nu}].$$

A set of band-pass, multi-scale and multi-orientation filters $\mathbf{g}_{\sigma, \mathbf{l}}[\mathbf{k}]$ can be obtained by simply applying the Riesz transform to circularly symmetric non-separable wavelets *e.g.* Eq. (4.8) or multi-scale filters *e.g.* the LoG filter g_σ Eq. (4.2), as

$$\mathbf{g}_{\sigma, \mathbf{l}}[\mathbf{k}] = \mathcal{R}^{\mathbf{l}}\{g_\sigma\}[\mathbf{k}]. \quad (4.12)$$

An example of a $2D$ Riesz filterbank for $N = 2$ and combined with a Simoncelli wavelet is shown in Fig 4.17. Eq. (4.12) yields a collection of filters measuring L -th order (scaled) derivatives. However, these derivatives are computed along image axes k_1, \dots, k_D , which entails similar challenges as separable wavelets: (i) analyzing images along their axes does not have a particular signification for the problem at hand (*e.g.* characterising local tissue structures within a tumour) and (ii) they are not rotation invariant/equivariant.

4.7.1 Aligning Riesz kernels

To address (i) and (ii), one can locally align all Riesz filters using a given alignment criterion to combine directional sensitivity with local invariance to rotations, as motivated in Section 3.3 and Fig. 3.1. Whereas the max orientation pooling operation can be seen as an alignment criterion, the latter only focuses on the maximum response of one single filter. Since Riesz filter banks include several filters with distinct profiles (see *e.g.* Fig. 4.17 with order $L = 2$), other alignment criteria based on meaningful quantities can be used. In particular, two suitable alignment criteria are those maximizing the gradients or the Hessian¹⁸, which can be efficiently computed directly from the Riesz coefficients using the structure (or Hessian) tensor⁸. The formulae for computing the regularised structure tensor from Riesz coefficients can be found in Section IV-B of Chenouard *et al.*⁸. Regularizing the structure tensor is important to determine the scale of the structures from which the orientation is important at a given position \mathbf{k}_0 . This must be optimized for the problem at hand via the variance σ_{tensor} of a Gaussian window $g_{\sigma_{\text{tensor}}}[\mathbf{k}]$ noted as $\nu(\mathbf{x})$ in Section IV-B of⁸. Computing these alignment maps allows determining how to further orient all elements of the Riesz filter bank to achieve locally rotation invariant image analysis. More specifically, the collection of eigenvectors (sorted by corresponding eigenvalues) of the structure tensor at the position \mathbf{k}_0 yields

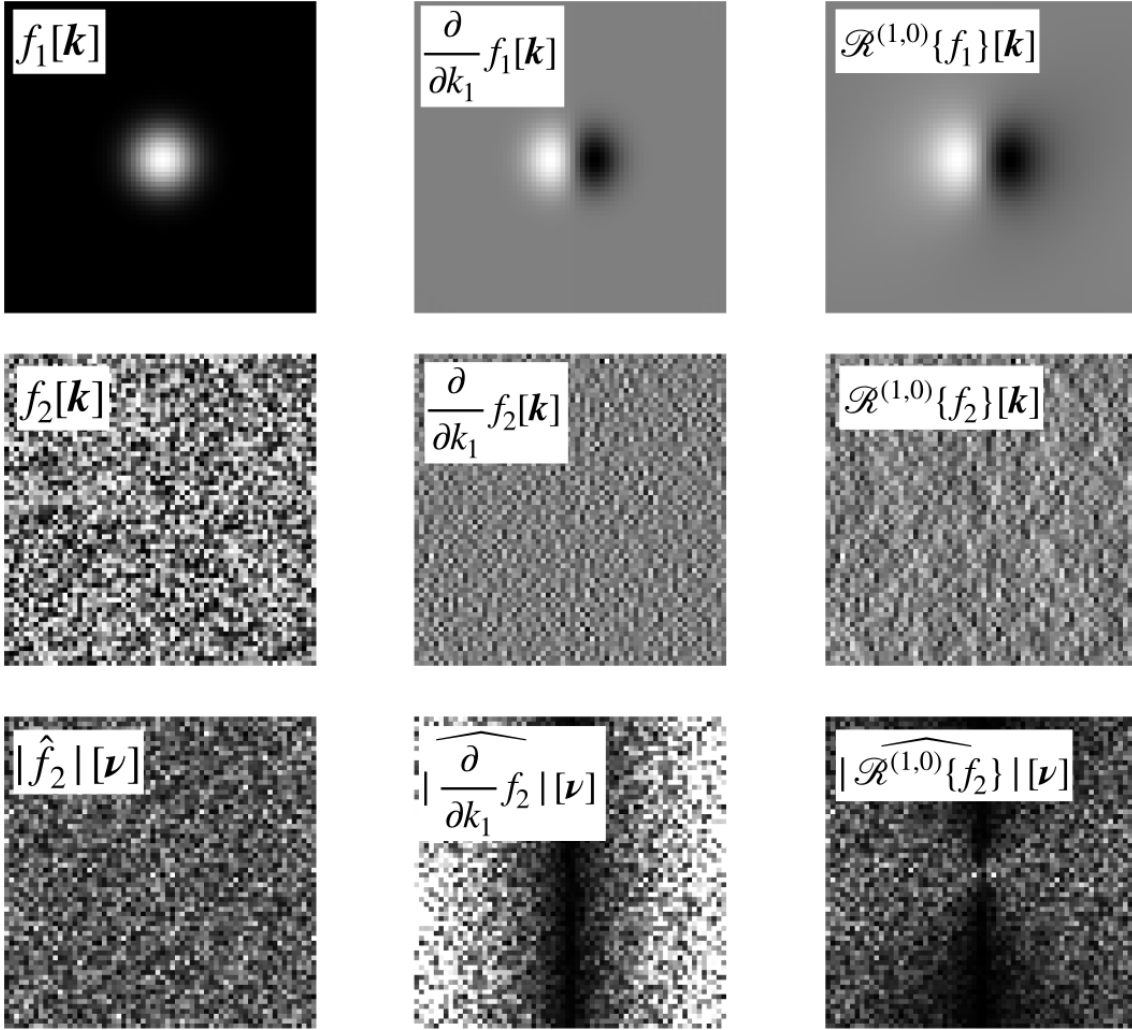


Figure 4.16 — Qualitative comparison between classical image derivatives (high-pass) as defined in (4.9) and Riesz-based image derivatives (all-pass) as defined in (4.10).

First row: Gaussian image f_1 and its derivatives in the spatial domain. The classical derivative $\frac{\partial}{\partial k_1} f_1[\mathbf{k}]$ and the Riesz transform $\mathcal{R}^{(1,0)}\{f_1\}[\mathbf{k}]$ look very similar because the Gaussian image is essentially composed of low frequencies.

Second row: White noise image f_2 and its derivatives in the spatial domain. The classical derivative $\frac{\partial}{\partial k_1} f_2[\mathbf{k}]$ contains mostly high frequencies, whereas the Riesz derivative $\mathcal{R}^{(1,0)}\{f_2\}[\mathbf{k}]$ retains all frequencies.

Third row: White noise image \hat{f}_2 and its derivatives in the Fourier domain (modulus are shown). The difference between the two derivatives approaches is striking, where the boost of high frequencies along ν_1 for the classical derivative approach can be clearly seen in $|\widehat{\frac{\partial}{\partial k_1} f_2}|[\boldsymbol{\nu}]$. The vertical and centered black line results from the fact that first order derivatives are zero mean (*i.e.* they contain no null frequencies).

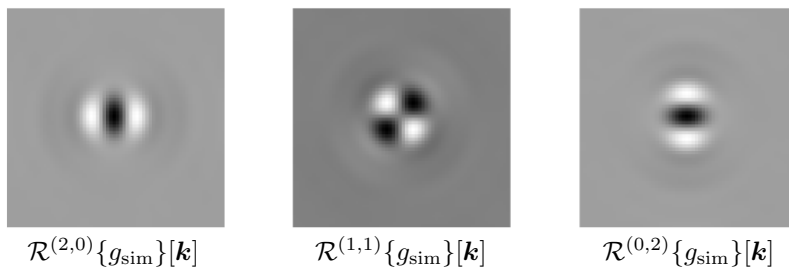


Figure 4.17 — A 2D Riesz filterbank for $N = 2$ and combined with a Simoncelli wavelet.

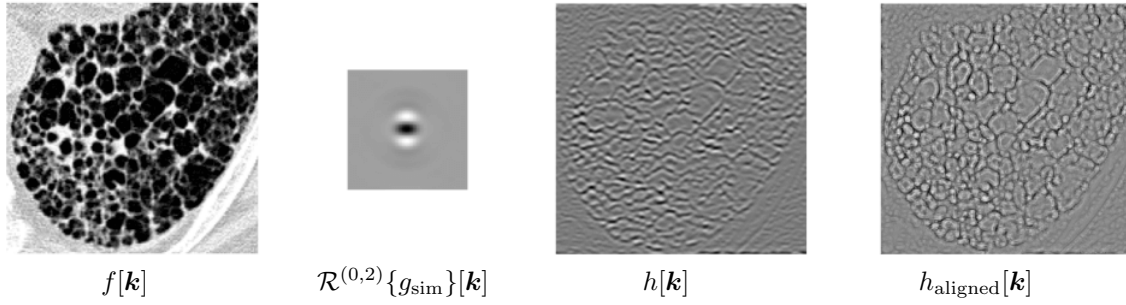


Figure 4.18 — Example of image filtering in $2D$ with the Riesz transform. $h_{\text{aligned}}[\mathbf{k}]$ was obtained by aligning the responses of $\mathcal{R}^{(0,2)}\{g_{\text{sim}}\}[\mathbf{k}]$ based on the structure tensor and allows highlighting of collagen fibers at any orientation.

a rotation matrix that can be used to align the Riesz filters. Moreover, thanks to the steerability property of the Riesz transform⁴², no additional convolution operations are required to compute the response of rotated filters. A steerable filter²⁵ is a filter that can be obtained as a linear combination of steerable kernels, and all rotations of the former can also be obtained via another linear combination of the filter responses parameterised by the desired rotation (*e.g.* one rotation angle in $2D$), which obviates the need to re-convolve the oriented filter with the input image. The formulae and directions for computing the steered Riesz coefficients are detailed in Section III-E of Chenouard *et al.*⁸. To summarise, aligned Riesz filters allow directional and rotation-invariant image analysis that can characterise interpretable transitions in medical images. Rotationally-invariant steerable representations can also be learned to be specific to the problem at hand^{1,15}, also with CNNs^{1,2,5,45–47}.

Implementations of the Riesz transform are available such as in^{27(g), 8(h)} and its adaptation for texture feature extraction, including filter alignment, in¹⁸⁽ⁱ⁾. An example of image filtering in $2D$ with the Riesz kernel $\mathcal{R}^{(0,2)}\{g_{\text{sim}}\}[\mathbf{k}]$ is shown in Fig. 4.18.

4.8 Qualitative Comparison of Linear Filter Properties

Based on the quantitative comparison criteria introduced in Section 3, it appears that not all approaches can fulfill all properties that are desirable for local medical image analysis. One typically observes a trade off between implementation simplicity and efficiency versus filter specificity and invariance/equivariance to geometric transforms¹⁶. Table 4.1 provides a qualitative comparison of the filtering methods detailed in Section 4.

While approximated rotation invariance can be artificially added to all methods by computing the average of directional response maps, this often annihilates directional sensitivity (see Section 3.3). Other designs allow combining rotation invariance with directional sensitivity (see Section 3.3).

4.9 Interpolation and convolutional filtering

When applied to an image, displacements of one voxel may not correspond to a displacement of the same physical distance depending on the direction. Physical distance is defined by voxel spacing which, in $3D$ imaging, commonly differs between directions and images. This case is referred to

^g<https://pypi.org/project/itk-isotropicwavelets/>, as of July 2019.

^h<http://bigwww.epfl.ch/demo/steerable-wavelets-3d/>, as of July 2019.

ⁱ<http://publications.hevs.ch/index.php/publications/show/2035/>, as of July 2019.

Table 4.1 — Qualitative comparison of common linear filtering approaches. “yes/no” indicates filters that can be but are not necessarily designed as wavelets.

	directional sensitivity	local rotation invariance	separability of the convolution	wavelet
LoG	no	yes	no (yes with DoG)	yes/no
Laws kernels	yes	no	yes	no
Gabor	yes	no	no	yes/no
separable wavelets	yes	no	yes	yes
isotropic non-separable wavelets	no	yes	no	yes
aligned wavelets (<i>e.g.</i> Riesz)	yes	yes	no	yes

as “anisotropic” image resolution. For example, in-plane voxel spacings in computed tomography (CT) images are usually smaller than the slice thickness. Also, voxel spacing can differ between acquisition protocols. Thus, simply applying filters to all images generates response maps that cannot be directly compared, because the frequency response of filters is different. To avoid this problem, we recommend performing filtering after resampling the image to uniform voxel spacing using interpolation (see Fig. 1.1).

However, performing image interpolation prior to filtering yields its own issues (see Fig. 4.19). Interpolation itself, by definition, alters the frequency content of the image, which may be particularly visible in response maps created by high-pass filters (see Fig. 4.20). As can be observed from the figure, trilinear interpolation acts as a low-pass filter when upsampling, whereas the tricubic spline method keeps preserves more high-frequency content. Ideal interpolation methods for upsampling do not exist as it is impossible to completely infer missing data. While we expect that superresolution based on deep learning could be used to improve interpolation results in the future¹⁹, in the meantime features derived from response maps created by high-pass filters may lack robustness.

Alternatively, when an analytical expression of the filter is available, the latter is defined in the continuous domain, for example for a Gaussian filter. This provides the opportunity to (re)compute filter values for anisotropic voxel grids as well as inter-image/patient differences in image resolution and obviates interpolation. However, for clarity, we will not consider this approach in the benchmarking process (Section 5).

For most applications, boundary conditions are not critical as ROIs do not tend to be located at an image boundary. Thus, for many medical images, the nearest value boundary conditions may be used. If the ROI is close to a boundary, mirror boundary conditions may be preferable (see Section 2.2).

In general, image features, or a subset of features, may be calculated from response maps in the same way as for the original image. Image filtering also affects the discretisation method that can be used prior to computing, for example texture features. Intensities in most response maps no longer have a direct physical meaning. Hence, *fixed bin size* methods can no longer be used for response maps. *Fixed bin number* or similar discretisation methods should be used instead⁵⁰.

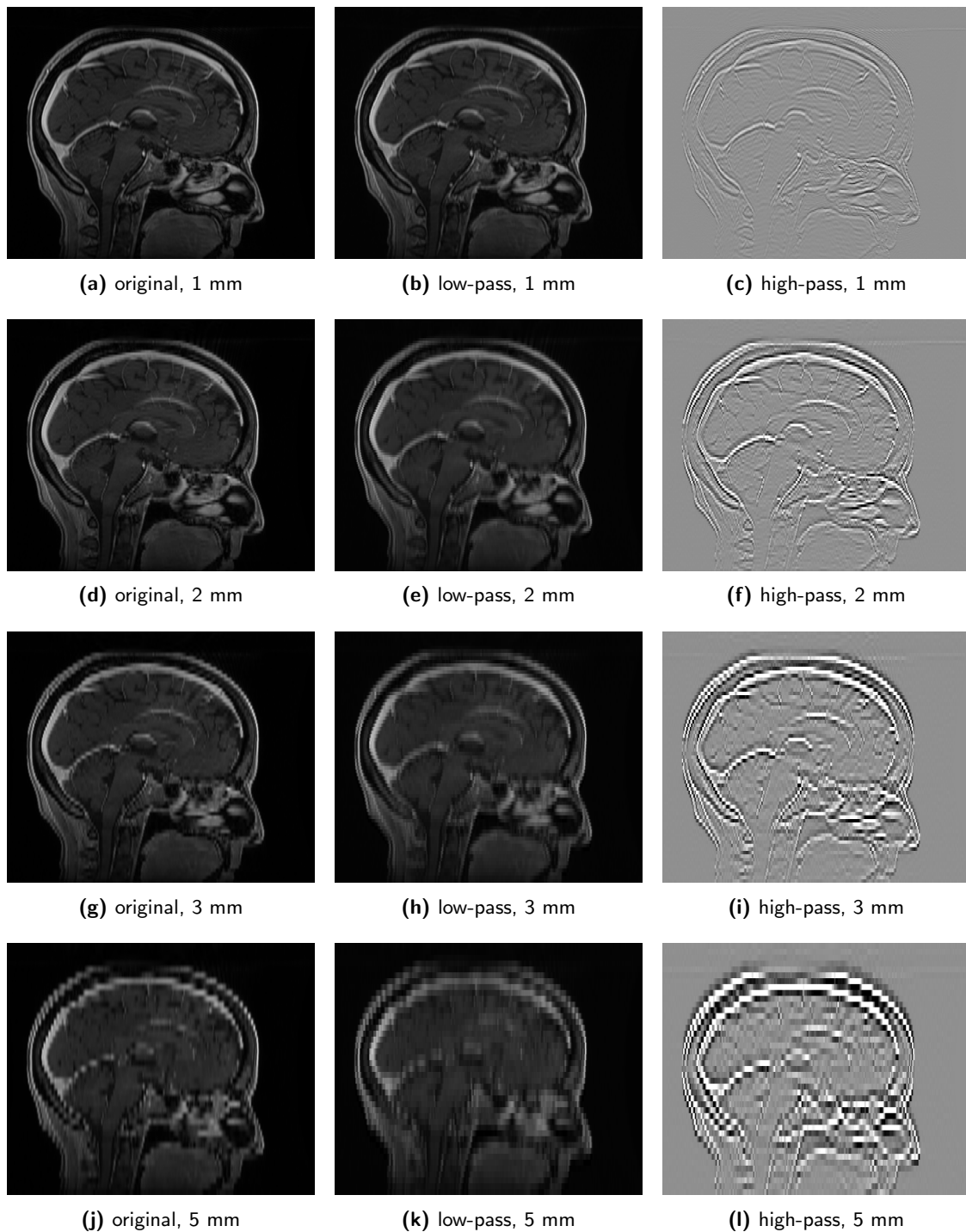


Figure 4.19 — Comparison of the low- and high-pass filters (LLL and HHH, respectively) of the Daubechies 2 wavelet for different slice spacing, without interpolation prior to filtering. All original images, all low-pass and all high-pass response maps share the same intensity scale. Notice that the high-pass filter response visibly depends on the slice spacing. This effect is also present in the low-pass image, but less so. The T1-weighted spoiled gradient echo image dataset was generated by the National Cancer Institute Clinical Proteomic Tumor Analysis Consortium (CPTAC)^{9,33}.

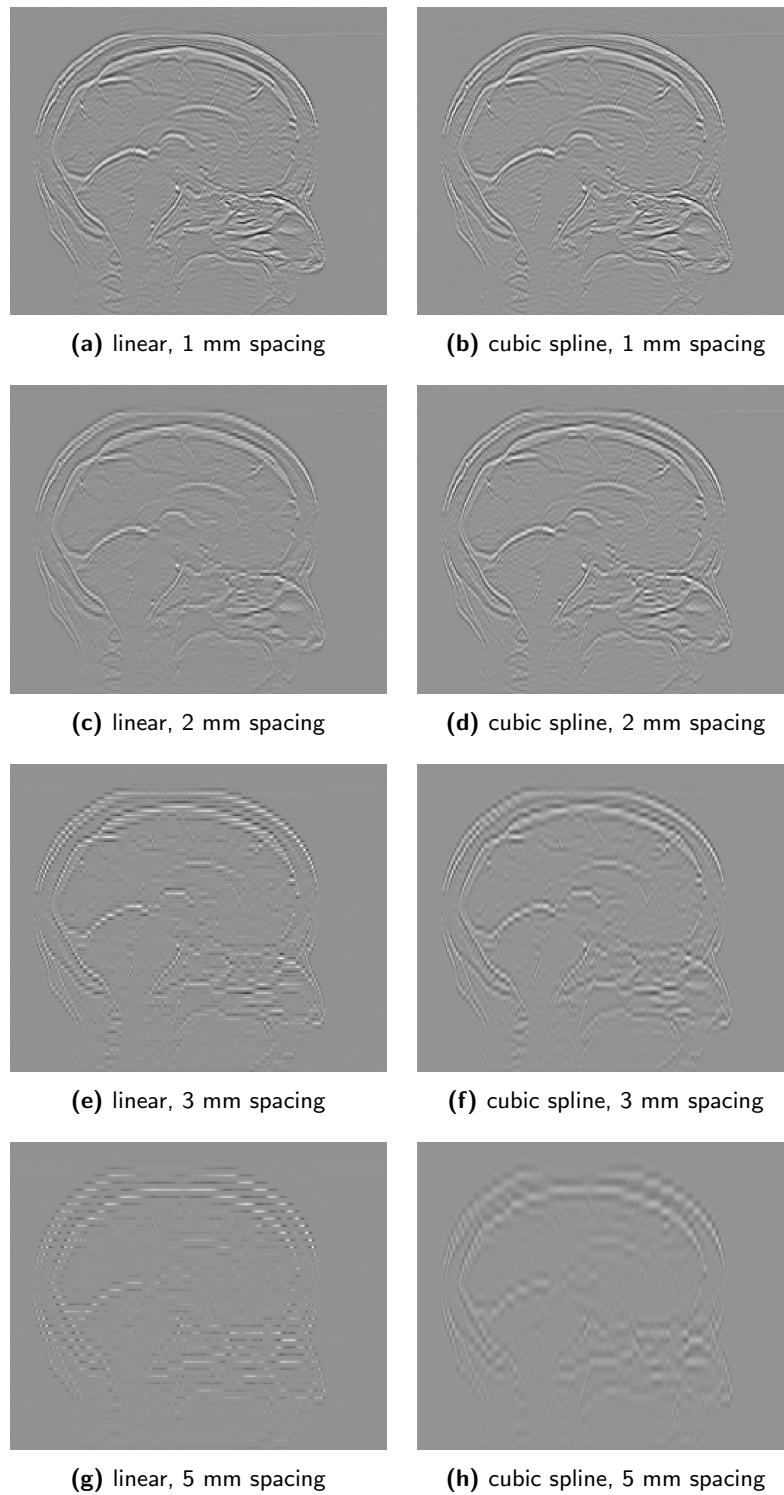


Figure 4.20 — Comparison of response maps of the Daubechies 2 wavelet high-pass filter (HHH) for images with a different slice spacing that are resampled to 1 mm isotropic voxels using trilinear or tricubic spline interpolation prior to filtering. Images with the same interpolation method have the same intensity scale. Unlike in Figure 4.19, image intensities remain similar, though the intensity range degrades with greater slice spacing. The response maps of 3 mm and 5 mm images obtained after trilinear interpolation show clear intermittent gaps between locations of the original slice. Within these slices, linear interpolation acts as a low-pass filter that suppresses high-frequency content. The cubic spline interpolation does not suffer from this problem. The T1-weighted spoiled gradient echo image dataset ($0.5 \times 0.5 \times 1.0$ mm) was generated by the National Cancer Institute Clinical Proteomic Tumor Analysis Consortium (CPTAC)^{9,33}.

Chapter 5

Benchmarking

This section details the two benchmarking phases proposed to assess differences between software implementations, to define standard filter implementations, and to achieve consensus on reference values for filter-based radiomics features. It includes the description of digital phantom images for controlled filter input and the description of a CT image obtained in a patient with lung cancer. The methodology used to find a consensus on standard filter implementations is subsequently detailed in Section 5.1. This includes parameter configurations for image filters, how response maps are compared, as well as the submission procedure for response maps for comparison between software implementations. The methodology used to find reference values for filter-based radiomics features is afterwards described in Section 5.2.

5.1 Phase 1: Benchmarking filters using digital phantoms

The IBSI developed several 3D phantoms to test image filter implementations. All phantoms (but the orientation phantom) have the same dimension and consist of $64 \times 64 \times 64$ isotropic voxels with 2.0 by 2.0 by 2.0 mm spacing. 8-bit voxel intensities in all phantoms fall in the range $[0, 255]$. The phantoms are stored in the NIfTI format. The phantoms are shown in Fig. 5.1. They include:

- **Empty phantom** (`empty.nii.gz`): all voxels have 0 intensity. Intended to investigate the convolution process.
- **Impulse response phantom** (`impulse_response.nii.gz`): all but one voxel have intensity 0. The single remaining centre voxel has an intensity of 255. This allows visualisation of the filter $g[\mathbf{k}]$ itself.
- **Checkerboard phantom** (`checkerboard.nii.gz`): alternates between cubic regions with intensity 0 and with intensity 255.
- **Noise phantom** (`noise.nii.gz`): contains Gaussian noise with mean intensity of 127 and a standard deviation of 48. As such it has no inherent structure.
- **Sphere phantom** (`sphere.nii.gz`): consists of four concentric spherical hulls with intensity 255 that are centred on the phantom centre. Thus, the phantom lacks directionality.
- **Pattern #1 phantom** (`pattern_1.nii.gz`): this is the first of three phantoms that involve directionality. Three perpendicular lines (intensity 255) intersect at the centre of the phantom. Along with Pattern #2 and Pattern #3 phantoms, it is intended to investigate filtering methods able to characterise the local organisation of image directions¹⁷.

- **Pattern #2 phantom** (`pattern_2.nii.gz`): this is the second of the directional phantoms. The phantom contains three parallel lines (intensity 255).
- **Pattern #3 phantom** (`pattern_3.nii.gz`): this the last of directional phantoms. The phantom contains three lines (intensity 255), of which two are parallel.
- **Orientation phantom** (`orientation.nii.gz`): Not all filters are rotationally invariant, and therefore the direction along which filters are applied affects the response map. Using the orientation phantom you can check whether the orientation of the image coordinate system in your software matches the orientation expected by the IBSI. The orientation phantom has a dimension of (32, 48, 64) voxels along k_1 (x), k_2 (y) and k_3 (z) axes, respectively. The pixel intensity increases with the distance from the origin, which has an intensity of 0. The most distal voxel has an intensity of 141.

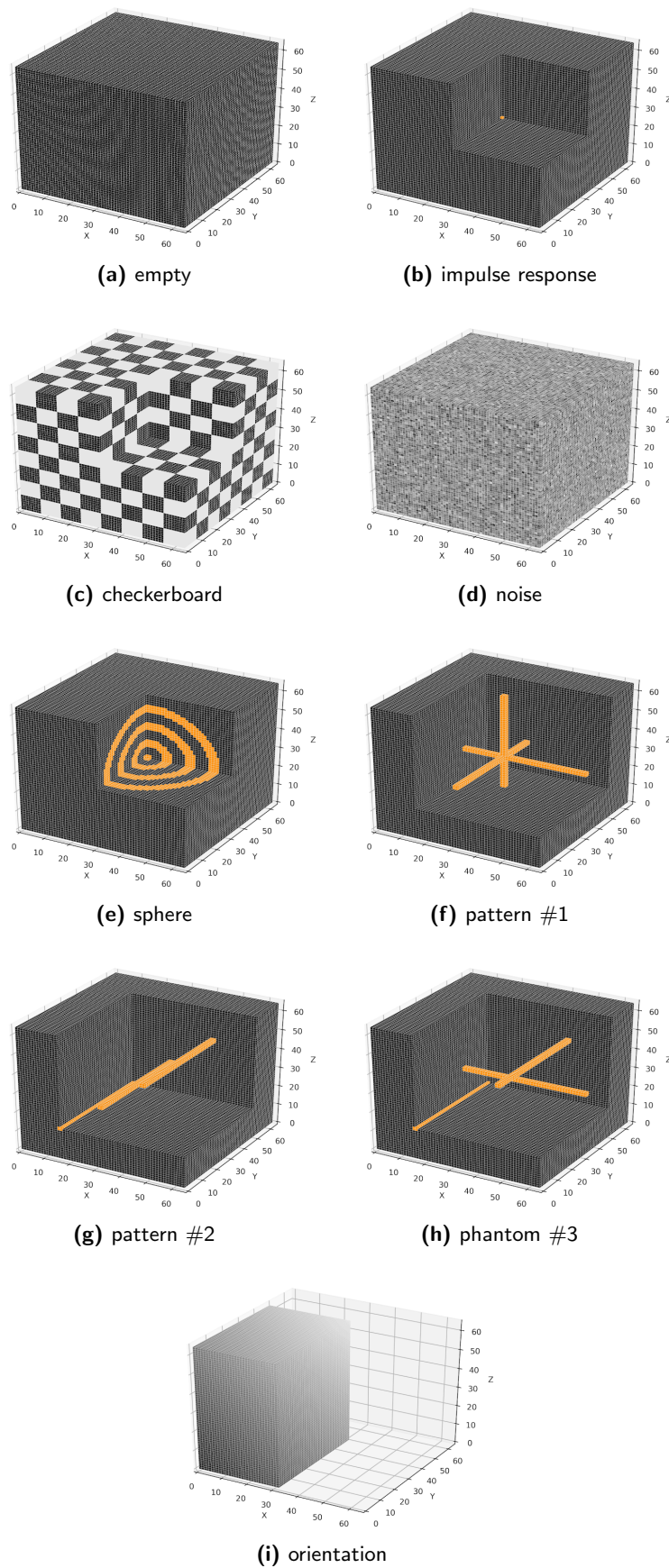


Figure 5.1 — Synthetic phantoms for verifying compliance with reference filter implementation.

The aim of benchmarking the filters using the phantom images is to arrive at a standard implementation. Therefore, filters are applied to the phantoms to create a response maps $h[\mathbf{k}]$, which have the same dimensions as the phantoms ($64 \times 64 \times 64$ voxels). Instead of calculating one or more features from each response map (see Section 1.3), the entire response map is to be submitted online to ease debugging and find differences in software implementations. Therefore it is important to note the following:

- The phantom data need to be converted from an integer data type to at least 32 bit floating point precision, prior to filtering.
- Filters need to be applied using the settings provided in Table 5.1. Some settings refer to real-world spacing (in millimetres), instead of voxel spacing.
- The response map needs to be exported in a (compressed) NIfTI format, with at least 32 bit floating point precision. Please make sure that the response map has the correct dimensions.
- The response map should be uploaded to <https://radiomics.hevs.ch/ibsi>. This requires you to log in using a Github account.

The voxel-wise distance between response maps from the different teams will then be compared to identify a consensus response map. In order to visualise and quantify the discrepancy between response maps $h[\mathbf{k}]$, a variational approach based on Principal Component Analysis (PCA) in the $\mathbb{R}^{64 \times 64 \times 64}$ space of voxel space is used. Each considered response map is an observation in this space. In addition, a consensus is defined as the average cluster (centroid in $\mathbb{R}^{64 \times 64 \times 64}$) of all submissions. A visualisation of the first two PCA components allows assessing the distribution and dispersion of the submissions and the consensus. In addition, a boxplot is proposed to visualise the distribution of the distances to the consensus in $\mathbb{R}^{64 \times 64 \times 64}$ and to suggest outliers. An example of comparison of 120 simulated response maps is shown in Fig. 5.2.

ID	Filter	Phantom	Filter parameters
1.a.1	mean	checkerboard	3D filter, support $M = 15$, zero padding
1.a.2			3D filter, support $M = 15$, nearest value padding
1.a.3			3D filter, support $M = 15$, periodic padding
1.a.4			3D filter, support $M = 15$, mirror padding
1.b.1		impulse	2D filter, support $M = 15$, zero padding
2.a	LoG	impulse	<ul style="list-style-type: none"> • zero padding • 3D filter, scale $\sigma^* = 3.0$ mm, filter size cutoff $4\sigma^*$
2.b		checkerboard	<ul style="list-style-type: none"> • mirror padding • 3D filter, scale $\sigma^* = 5.0$ mm, filter size cutoff $4\sigma^*$
2.c		checkerboard	<ul style="list-style-type: none"> • mirror padding • 2D filter, scale $\sigma^* = 5.0$ mm, filter size cutoff $4\sigma^*$
3.a.1	Laws	impulse	<ul style="list-style-type: none"> • zero padding • 3D filter, E5L5S5 response map
3.a.2			<ul style="list-style-type: none"> • zero padding • 3D filter, E5L5S5 response map • 3D rotation invariance, max pooling
3.a.3			<ul style="list-style-type: none"> • zero padding • 3D filter, E5L5S5 response map • 3D rotation invariance, max pooling • energy map, distance $\delta = 7$ voxels

Continued on next page

ID	Filter	Phantom	Filter parameters
3.b.1		checkerboard	<ul style="list-style-type: none"> • mirror padding • 3D filter, E3W5R5 response map
3.b.2			<ul style="list-style-type: none"> • mirror padding • 3D filter, E3W5R5 response map • 3D rotation invariance, max pooling
3.b.3			<ul style="list-style-type: none"> • mirror padding • 3D filter, E3W5R5 response map • 3D rotation invariance, max pooling • energy map, distance $\delta = 7$ voxels
3.c.1		checkerboard	<ul style="list-style-type: none"> • mirror padding • 2D filter, L5S5 response map
3.c.2			<ul style="list-style-type: none"> • mirror padding • 2D filter, L5S5 response map • 2D rotation invariance, max pooling
3.c.3			<ul style="list-style-type: none"> • mirror padding • 2D filter, L5S5 response map • 2D rotation invariance, max pooling • energy map, distance $\delta = 7$ voxels
4.a.1	Gabor	impulse	<ul style="list-style-type: none"> • zero padding • 2D modulus response map • $\sigma^* = 10.0$ mm, $\lambda^* = 4$ mm, $\gamma = 1/2$ • in-plane orientation $\theta = \pi/3$
4.a.2			<ul style="list-style-type: none"> • zero padding • 2D modulus response map • $\sigma^* = 10.0$ mm, $\lambda^* = 4$ mm, $\gamma = 1/2$ • 2D rotation invariance, $\Delta\theta = \pi/4$, average pooling • average 2D responses over orthogonal planes
4.b.1		sphere	<ul style="list-style-type: none"> • mirror padding • 2D modulus response map • $\sigma^* = 20.0$ mm, $\lambda^* = 8$ mm, $\gamma = 5/2$ • in-plane orientation $\theta = 5\pi/4$
4.b.2			<ul style="list-style-type: none"> • mirror padding • 2D modulus response map • $\sigma^* = 20.0$ mm, $\lambda^* = 8$ mm, $\gamma = 5/2$ • 2D rotation invariance, $\Delta\theta = \pi/8$, average pooling • average 2D responses over orthogonal planes
5.a.1	Daubechies 2	impulse	<ul style="list-style-type: none"> • zero padding • 3D filter, undecimated LHL map – 1st level
5.a.2			<ul style="list-style-type: none"> • zero padding • 3D filter, undecimated LHL map – 1st level • 3D rotation invariance, average pooling
6.a.1	Coifflet 1	sphere	<ul style="list-style-type: none"> • periodic padding • 3D filter, undecimated HHL map – 1st level
6.a.2			<ul style="list-style-type: none"> • periodic padding • 3D filter, undecimated HHL map – 1st level • 3D rotation invariance, average pooling
7.a.1	Haar	checkerboard	<ul style="list-style-type: none"> • mirror padding • 3D filter, undecimated LLL map – 2nd level • 3D rotation invariance, average pooling

Continued on next page

ID	Filter	Phantom	Filter parameters
7.a.2			<ul style="list-style-type: none"> • mirror padding • 3D filter, undecimated HHH map – 2nd level • 3D rotation invariance, average pooling
8.a.1	Simoncelli	checkerboard	<ul style="list-style-type: none"> • 3D filter, B map – 1st level
8.a.2			<ul style="list-style-type: none"> • 3D filter, B map – 2nd level
8.a.3			<ul style="list-style-type: none"> • 3D filter, B map – 3rd level
9.a	Riesz-transformed LoG	impulse	<ul style="list-style-type: none"> • zero padding • 3D filter, scale $\sigma = 3.0$ mm, filter size cutoff 4σ • $\mathbf{l} = (1, 0, 0)$
9.b.1		sphere	<ul style="list-style-type: none"> • zero padding • 3D filter, scale $\sigma = 3.0$ mm, filter size cutoff 4σ • $\mathbf{l} = (0, 2, 0)$
9.b.2			<ul style="list-style-type: none"> • zero padding • scale $\sigma = 3.0$ mm, filter size cutoff 4σ • 3D filter, $\mathbf{l} = (0, 2, 0)$ • aligned by structure tensor, $\sigma_{\text{tensor}} = 1\text{mm}$
10.a	Riesz-transformed Simoncelli	impulse	<ul style="list-style-type: none"> • zero padding • 3D filter, B map – 1st level • $\mathbf{l} = (1, 0, 0)$
10.b.1		pattern 1	<ul style="list-style-type: none"> • nearest value padding • 3D filter, B map – 1st level • $\mathbf{l} = (0, 2, 0)$
10.b.2			<ul style="list-style-type: none"> • nearest value padding • 3D filter, B map – 1st level • $\mathbf{l} = (0, 2, 0)$ • aligned by structure tensor, $\sigma_{\text{tensor}} = 1\text{mm}$

Table 5.1 — Filters, parameters and phantoms for comparing and standardising filter implementations. Note that 2D and 3D rotation invariance for Laws and undecimated wavelet filters (Daubechies 2, Coifflet 1, Haar) is estimated using equivariant right angle rotational representation for separable filters (Appendix A). Pooling refers to pooling over response maps obtained under different orientations. 2D filters are applied in the axial plane by default, except for Gabor filters. For Gabor filters, 2D rotation invariance is estimated by rotating the the filter kernel over multiple orientations in the image plane.

5.2 Phase 2: Benchmarking filters using a lung cancer CT image

CT images from four patients with non-small-cell lung carcinoma were made available to serve as radiomics phantoms (DOI:10.17195/candat.2016.08.1). The IBSI previously used the image for the first patient (PAT1) to establish reference values for image features⁵⁰. This image is used here again.

The lung cancer CT image is stored as a stack of slices in DICOM format. The image slices can be identified by the DCM_IMG prefix. Additionally, the RT structure set (starting DCM_RS) contains a delineation of the Gross Tumour Volume (GTV) which is used to create a segmentation mask for the ROI. The image and the mask have been converted to the NIfTI format. Note that some import algorithms alter the floating point bit depth of the image and mask. If this occurs, the

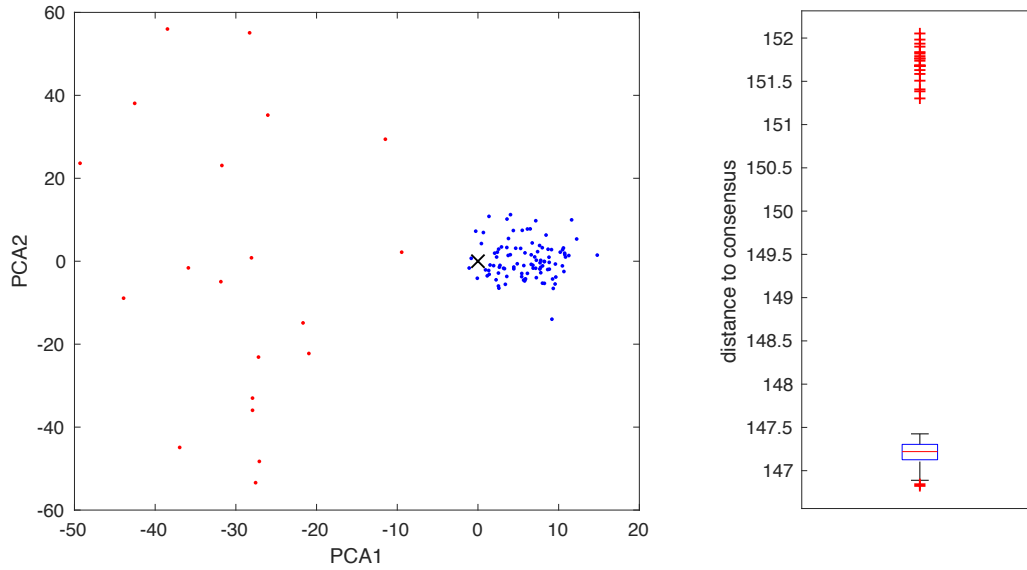


Figure 5.2 — PCA (left) and boxplot (right) visualisations of a comparison between 120 simulated response maps. The blue and red observations are coming from two distinct uniform distributions. The consensus is marked with a black cross in the scatter plot.

imported image and mask should be converted back to (at least) 32-bit floating point and rounded prior to any further processing.

Compared to the digital phantoms, the lung cancer CT image offers a more realistic setting to benchmark applications of image filters as part of a radiomics analysis. Image filtering is assessed as part of two different image processing configurations, see Table 5.2. The main difference between the configurations is that one performs image processing and filtering within each image slice (*i.e.* $2D$), whereas the other operates in $3D$. Features are still computed over the entire region of interest, as mentioned in the IBSI 1 reference manual. The filtering configurations are then shown in Table 5.3.

Parameter	Configuration A	Configuration B
slice-wise ($2D$) or as volume ($3D$)	$2D$	$3D$
interpolation	no	yes
resampled voxel spacing (mm)	—	$1 \times 1 \times 1$
interpolation method	—	tricubic spline
intensity rounding	—	nearest integer
ROI interpolation method	—	trilinear
ROI partial mask volume	—	0.5
re-segmentation		
range (HU)	$[-1000, 400]$	$[-1000, 400]$
outlier filtering	none	none
image filters		
filters	see Table 5.3	see Table 5.3
boundary condition	mirror	mirror

v5: Updated the 2D-test.

Table 5.2 — Image processing configurations for obtaining reference values for intensity-based statistical features and Intensity Histogram (IH) features after image filtering in the lung cancer CT image. The different processing steps are described in the IBSI reference manual⁵¹. Note that the image should not be cropped. Also note the use of a higher order (spline) resampling method instead of linear interpolation. ROI: Region Of Interest; HU: Hounsfield Unit.

ID	Filter	A	B	Filter parameters
1.A — 1.B	none	×	×	—
2.A — 2.B	mean	×	×	<ul style="list-style-type: none"> • 2D filter, support $M = 5$ voxels • 3D filter, support $M = 5$ voxels
3.A — 3.B	LoG	×	×	<ul style="list-style-type: none"> • 2D filter, scale $\sigma^* = 1.5$ mm, filter size cutoff 4σ • 3D filter, scale $\sigma^* = 1.5$ mm, filter size cutoff 4σ
4.A — 4.B	Laws	×	×	<ul style="list-style-type: none"> • 2D filter, L5E5 energy map, distance $\delta = 7$ voxels • 2D rotation invariance, max pooling • 3D filter, L5E5E5 response map • 3D rotation invariance, max pooling • energy map, distance $\delta = 7$ voxels
5.A — 5.B	Gabor	×	×	<ul style="list-style-type: none"> • 2D modulus response map • $\sigma^* = 5$ mm, $\lambda^* = 2$ mm, $\gamma = 3/2$ • 2D rotation invariance, $\Delta\theta = \pi/8$, average pooling • 2D modulus response map • $\sigma^* = 5$ mm, $\lambda^* = 2$ mm, $\gamma = 3/2$ • 2D rotation invariance, $\Delta\theta = \pi/8$, average pooling • average 2D responses over orthogonal planes
6.A — 6.B	Daubechies 3	×	×	<ul style="list-style-type: none"> • 2D filter, undecimated LH map – 1st level • 2D rotation invariance, average pooling • 3D filter, undecimated LLH map – 1st level • 3D rotation invariance, average pooling
7.A — 7.B	Daubechies 3	×	×	<ul style="list-style-type: none"> • 2D filter, undecimated HH map – 2nd level • 2D rotation invariance, average pooling • 3D filter, undecimated HHH map – 2nd level • 3D rotation invariance, average pooling
8.A — 8.B	Simoncelli	×	×	<ul style="list-style-type: none"> • 2D filter, B map – 1st level • 3D filter, B map – 1st level
9.A — 9.B	Simoncelli	×	×	<ul style="list-style-type: none"> • 2D filter, B map – 2nd level • 3D filter, B map – 2nd level
10.A — 10.B	Riesz-transformed Simoncelli	×	×	<ul style="list-style-type: none"> • 2D filter, B map – 1st level • $\mathbf{l} = (0, 2)$ • 3D filter, B map – 1st level • $\mathbf{l} = (0, 2, 0)$
11.A — 11.B	Riesz-transformed Simoncelli	×	×	<ul style="list-style-type: none"> • 2D filter, B map – 1st level • $\mathbf{l} = (0, 2)$ • aligned by structure tensor, $\sigma_{\text{tensor}} = 1\text{mm}$ • 3D filter, B map – 1st level • $\mathbf{l} = (0, 2, 0)$ • aligned by structure tensor, $\sigma_{\text{tensor}} = 1\text{mm}$

Table 5.3 — Filters and parameters for the configurations A (2D) and B (3D) defined in Table 5.2. These settings are used to determine reference values for radiomics features computed from filter response maps. Note that 2D and 3D rotation invariance for Laws and undecimated wavelet filters (e.g. Daubechies 3) is estimated using equivariant right angle rotational representation for separable filters (Appendix A). For Gabor filters, 2D rotation invariance is estimated by rotating the the filter kernel over multiple orientations in the image plane. The padding method can be freely chosen.

Features will be computed based on the response maps of each of the filters applied to the CT image. That is, in contrast to the previous section where the objective is to arrive at a standard filter implementation, here we aim to define reference values for radiomics features. The features that are being assessed are listed below in Section 5.2.1.

Consensus on reference values will be assessed in the same manner as previously⁵¹. In short, for each team, the most recent values for each feature will be collected. This value is then limited

to three significant digits. The mode of the values across software implementations is then used as a tentative reference value. The level of consensus on the validity of each reference value is then quantified, by: (I) counting the number of teams that submitted a value matching the tentative reference value with a tolerance margin (to be determined); and (II) assessing the proportion of matching teams.

Then, we will assign four consensus levels on the first measure: *weak*: less than three; *moderate*: three to five; *strong*: six to nine; *very strong*: ten or more.

The second measure is used to assess the stability of the consensus. A tentative reference value will be considered valid if and only if at least moderate consensus can be reached by an absolute majority (> 50%) of teams that contribute the value.

The compliance check list developed previously⁵¹ will be used to check whether image processing (aside from filtering) is correctly performed.

5.2.1 Features

Only part of the radiomics features standardised previously need to be reported. This is done for two reasons. First, to facilitate the submission process. And secondly to ensure that sufficient statistical power can be reached as we found that the majority of teams did not implement all features previously⁵¹.

These features are listed below with their permanent identifiers.

1. Mean intensity (Q4LE)
2. Intensity variance (ECT3)
3. Intensity skewness (KE2A)
4. (Excess) intensity kurtosis (IPH6)
5. Median intensity (Y12H)
6. Minimum intensity (1GSF)
7. 10th intensity percentile (QG58)
8. 90th intensity percentile (8DWT)
9. Maximum intensity (84IY)
10. Intensity interquartile range (SALO)
11. Intensity range (2OJQ)
12. Intensity-based mean absolute deviation (4FUA)
13. Intensity-based robust mean absolute deviation (1128)
14. Intensity-based median absolute deviation (N72L)
15. Intensity-based coefficient of variation (7TET)
16. Intensity-based quartile coefficient of dispersion (9S40)
17. Intensity-based energy (N8CA)
18. Root mean square intensity (5ZWQ)

In addition some diagnostic features may be reported. Configurations A and B are similar to those used in IBSI 1. However the use of a higher order interpolation method precludes the direct use of diagnostic features from the previous work. Most features are based on the ROI intensity mask after interpolation and re-segmentation, since these features will be different from those previously established:

1. Number of voxels in the ROI mask, before interpolation.
2. Number of voxels in the ROI intensity mask, after interpolation and re-segmentation.
3. Mean intensity in the ROI intensity mask after interpolation and re-segmentation.
4. Maximal intensity in the ROI intensity mask after interpolation and re-segmentation.
5. Minimal intensity in the ROI intensity mask after interpolation and re-segmentation.

Chapter 6

Conclusion

In this document we detail various convolutional image filters to standardise their implementation in radiomics software. Such filters are used to convert medical images into response maps that emphasise characteristics such as edges, blobs or directional structures. Quantifying such characteristics using standardised radiomics software will open the way to their reproducible assessment and validation, and their translation from bench to bedside.

Acknowledgments

This work was partially supported by the Swiss National Science Foundation (grant 205320_179069), the Swiss Personalized Health Network (IMAGINE and QA4IQI projects) and the Canada CIFAR AI Chairs program.

Bibliography

- [1] V. Andrearczyk, J. Fageot, V. Oreiller, X. Montet, and A. Depeursinge. Exploring local rotation invariance in 3D CNNs with steerable filters. In *Medical Imaging with Deep Learning*, MIDL, 2019.
- [2] V. Andrearczyk, J. Fageot, V. Oreiller, X. Montet, and A. Depeursinge. Local rotation invariance in 3D CNNs. *Medical Image Analysis*, to appear.
- [3] A. P. Apte, A. Iyer, M. Crispin-Ortuzar, R. Pandya, L. V. van Dijk, E. Spezi, M. Thor, H. Um, H. Veeraraghavan, J. H. Oh, A. Shukla-Dave, and J. O. Deasy. Technical note: Extension of CERR for computational radiomics: A comprehensive MATLAB platform for reproducible radiomics research. *Med. Phys.*, 45(8):3713–3720, 2018.
- [4] Y. Balagurunathan, Y. Gu, H. Wang, V. Kumar, O. Grove, S. Hawkins, J. Kim, D. B. Goldgof, L. O. Hall, R. A. Gatenby, and R. J. Gillies. Reproducibility and Prognosis of Quantitative Features Extracted from CT Images. *Translational oncology*, 7(1):72–87, 2014.
- [5] E. J. Bekkers, M. W. Lafarge, M. Veta, K. A. Eppenhof, J. P. Pluim, and R. Duits. Roto-translation covariant convolutional networks for medical image analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 440–448. Springer, 2018.
- [6] F. Bianconi and A. Fernández. Evaluation of the effects of Gabor filter parameters on texture classification. *Pattern Recognition*, 40(12):3325–3335, 2007.
- [7] M. Bogowicz, R. T. H. Leijenaar, S. Tanadini-Lang, O. Riesterer, M. Pruschy, G. Studer, J. Unkelbach, M. Guckenberger, E. Konukoglu, and P. Lambin. Post-radiochemotherapy PET radiomics in head and neck cancer - the influence of radiomics implementation on the reproducibility of local control tumor models. *Radiother. Oncol.*, 125(3):385–391, 2017.
- [8] N. Chenouard and M. Unser. 3D Steerable Wavelets in Practice. *IEEE Transactions on Image Processing*, 21(11):4522–4533, 2012.
- [9] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, L. Tarbox, and F. Prior. The cancer imaging archive (TCIA): maintaining and operating a public information repository. *J. Digit. Imaging*, 26(6):1045–1057, 2013.
- [10] T. S. Cohen and M. Welling. Group Equivariant Convolutional Networks. *CoRR*, abs/1602.0, 2016.
- [11] I. Daubechies. *Ten lectures on wavelets*, volume 61. SIAM, 1992.
- [12] A. Depeursinge. Multi-Scale and Multi-Directional Biomedical Texture Analysis: Finding the Needle in the Haystack. In *Biomedical Texture Analysis: Fundamentals, Tools and Challenges*, Elsevier-MICCAI Society Book series, pages 29–53. Elsevier, 2017.

- [13] A. Depeursinge and J. Fageot. Biomedical Texture Operators and Aggregation Functions: A Methodological Review and User’s Guide. In *Biomedical Texture Analysis: Fundamentals, Tools and Challenges*, Elsevier-MICCAI Society Book series, pages 55–94. Elsevier, 2017.
- [14] A. Depeursinge, J. Fageot, and O. S. Al-Kadi. Fundamentals of Texture Processing for Biomedical Image Analysis: A General Definition and Problem Formulation. In *Biomedical Texture Analysis: Fundamentals, Tools and Challenges*, Elsevier-MICCAI Society Book series, pages 1–27. Elsevier, 2017.
- [15] A. Depeursinge, Z. Puspoki, J. Ward, and M. Unser. Steerable wavelet machines (SWM): Learning moving frames for texture classification. *IEEE Transactions on Image Processing*, 26(4), 2017.
- [16] A. Depeursinge, J. Fageot, V. Andrearczyk, J. P. Ward, and M. Unser. Rotation Invariance and Directional Sensitivity: Spherical Harmonics versus Radiomics Features. In *Machine Learning in Medical Imaging*, pages 107–115, Cham, 2018. Springer International Publishing.
- [17] Y. Dicente Cid, J. Castelli, R. Schaer, N. Scher, A. Pomoni, J. Prior, and A. Depeursinge. QuantImage: An Online Tool for High-Throughput 3D Radiomics Feature Extraction in PET-CT. In *Biomedical Texture Analysis: Fundamentals, Tools and Challenges*, Elsevier-MICCAI Society Book series, pages 349–377. Elsevier, 2017.
- [18] Y. Dicente Cid, H. Müller, A. Platon, P.-A. Poletti, and A. Depeursinge. 3-D Solid Texture Classification Using Locally-Oriented Wavelet Transforms. *IEEE Transactions on Image Processing*, 26(4):1899–1910, 2017.
- [19] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 184–199, Cham, 2014. Springer International Publishing.
- [20] P. Dutilleul. An Implementation of the “algorithme à trous” to Compute the Wavelet Transform. In J.-M. Combes, A. Grossmann, and P. Tchamitchian, editors, *Wavelets*, pages 298–304. Springer, Berlin, Heidelberg, 1990.
- [21] S. Echegaray, S. Bakr, D. L. Rubin, and S. Napel. Quantitative Image Feature Engine (QIFE): an Open-Source, Modular Engine for 3D Quantitative Feature Extraction from Volumetric Medical Images. *Journal of Digital Imaging*, pages 1–12, 2017.
- [22] M. Eickenberg, G. Exarchakis, M. Hirn, and S. Mallat. Solid harmonic wavelet scattering: Predicting quantum molecular energy from invariant descriptors of 3D electronic densities. In *Advances in Neural Information Processing Systems*, pages 6540–6549, 2017.
- [23] Y.-H. D. Fang, C.-Y. Lin, M.-J. Shih, H.-M. Wang, T.-Y. Ho, C.-T. Liao, and T.-C. Yen. Development and evaluation of an open-source software package “CGITA” for quantifying tumor heterogeneity with molecular images. *BioMed research international*, 2014:248505, 2014.
- [24] J. J. Foy, K. R. Robinson, H. Li, M. L. Giger, H. Al-Hallaq, and S. G. Armato. Variation in algorithm implementation across radiomics software. *J. Med. Imaging*, 5(4):044505, 2018.
- [25] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.
- [26] M. Hatt, F. Tixier, L. Pierce, P. E. Kinahan, C. C. Le Rest, and D. Visvikis. Characterization of PET/CT images using texture analysis: the past, the present. . . any future? *Eur. J. Nucl. Med. Mol. Imaging*, 44(1):151–165, 2017.

- [27] E. P. Hernandez-Cerdan, P. Isotropic and Steerable Wavelets in N Dimensions. A multiresolution analysis framework. *The Insight Journal*, 1:1–20, 2016.
- [28] R. Kakarala and D. Mao. A theory of phase-sensitive rotation invariance with spherical harmonic and moment-based representations. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 105–112. IEEE, 2010.
- [29] P. Lambin, R. T. Leijenaar, T. M. Deist, J. Peerlings, E. E. de Jong, J. van Timmeren, S. Sanduleanu, R. T. Larue, A. J. Even, A. Jochems, Y. van Wijk, H. Woodruff, J. van Soest, T. Lustberg, E. Roelofs, W. van Elmpt, A. Dekker, F. M. Mottaghy, J. E. Wildberger, and S. Walsh. Radiomics: the bridge between medical imaging and personalized medicine. *Nature Reviews Clinical Oncology*, 14(12):749–762, 2017.
- [30] K. I. Laws. Rapid Texture Identification. In *24th Annual Technical Symposium*, volume 238, pages 376–381. SPIE, 1980.
- [31] G. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt, and A. O’Leary. Pywavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237, 2019.
- [32] S. G. Mallat. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.
- [33] National Cancer Institute Clinical Proteomic Tumor Analysis Consortium (CPTAC). Radiology data from the clinical proteomic tumor analysis consortium glioblastoma multiforme [CPTAC-GBM] collection, 2018.
- [34] C. Nioche, F. Orlhac, S. Boughdad, S. Reuzé, J. Goya-Outi, C. Robert, C. Pellot-Barakat, M. Soussan, F. Frouin, and I. Buvat. LIFEx: A freeware for radiomic feature calculation in multimodality imaging to accelerate advances in the characterization of tumor heterogeneity. *Cancer Res.*, 78(16):4786–4789, 2018.
- [35] N. Petkov and P. Kruizinga. Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: Bar and grating cells. *Biological Cybernetics*, 76(2):83–96, 1997.
- [36] J. Portilla and E. P. Simoncelli. A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients. *International Journal of Computer Vision*, 40(1):49–70, 2000.
- [37] P. Prasanna, P. Tiwari, and A. Madabhushi. Co-occurrence of Local Anisotropic Gradient Orientations (CoLIAGe): Distinguishing tumor confounders and molecular subtypes on MRI. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8675 LNCS, pages 73–80. Springer Verlag, 2014.
- [38] Z. Qian, D. N. Metaxas, and L. Axel. Extraction and Tracking of MRI Tagging Sheets Using a 3D Gabor Filter Bank. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 711–714. IEEE, 2006.
- [39] P. M. Szczypiński and A. Klepaczko. MaZda – A Framework for Biomedical Image Texture Analysis and Data Exploration. In *Biomedical Texture Analysis*, pages 315–347. Elsevier, 2017.
- [40] H. Tamura, S. Mori, and T. Yamawaki. Textural Features Corresponding to Visual Perception. *IEEE Transactions on Systems, Man and Cybernetics*, 8(6):460–473, 1978.
- [41] M. Unser. Image Processing I and II. Ecole polytechnique fédérale de Lausanne (EPFL), 2012.

- [42] M. Unser and D. Van De Ville. Wavelet steerability and the higher-order Riesz transform. *IEEE Transactions on Image Processing*, 19(3):636–652, 2010.
- [43] M. Unser, N. Chenouard, and D. Van De Ville. Steerable Pyramids and Tight Wavelet Frames in $L_2(\mathbb{R}^d)$. *IEEE Transactions on Image Processing*, 20(10):2705–2721, 2011.
- [44] J. J. M. van Griethuysen, A. Fedorov, C. Parmar, A. Hosny, N. Aucoin, V. Narayan, R. G. H. Beets-Tan, J.-C. Fillion-Robin, S. Pieper, and H. J. W. L. Aerts. Computational radiomics system to decode the radiographic phenotype. *Cancer Res.*, 77(21):e104–e107, 2017.
- [45] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. Cohen. 3D steerable CNNs: Learning rotationally equivariant features in volumetric data. In *Advances in Neural Information Processing Systems*, volume 2018-December, pages 10381–10392. Neural information processing systems foundation, 2018.
- [46] M. Weiler, F. A. Hamprecht, and M. Storath. Learning Steerable Filters for Rotation Equivariant CNNs. *CoRR*, abs/1711.0, 2018.
- [47] M. Winkels and T. S. Cohen. Pulmonary nodule detection in CT scans with equivariant CNNs. *Medical Image Analysis*, 55:15–26, 2019.
- [48] L. Zhang, D. V. Fried, X. J. Fave, L. A. Hunter, J. Yang, and L. E. Court. IBEX : An open infrastructure software platform to facilitate collaborative work in radiomics. *Medical Physics*, 42(3):1341–1353, 2015.
- [49] A. Zwanenburg. Radiomics in nuclear medicine: robustness, reproducibility, standardization, and how to avoid data analysis traps and replication crisis. *Eur. J. Nucl. Med. Mol. Imaging*, 46(13):2638–2655, 2019.
- [50] A. Zwanenburg, S. Leger, M. Vallières, and S. Löck. Image biomarker standardisation initiative. *arXiv*, 1612.07003, 2016.
- [51] A. Zwanenburg, M. Vallières, M. A. Abdalah, H. J. W. L. Aerts, V. Andrearczyk, A. Apte, S. Ashrafinia, S. Bakas, R. J. Beukinga, R. Boellaard, M. Bogowicz, L. Boldrini, I. Buvat, G. J. R. Cook, C. Davatzikos, A. Depeursinge, M.-C. Desseroit, N. Dinapoli, C. V. Dinh, S. Echegaray, I. El Naqa, A. Y. Fedorov, R. Gatta, R. J. Gillies, V. Goh, M. Götz, M. Guckenberger, S. M. Ha, M. Hatt, F. Isensee, P. Lambin, S. Leger, R. T. H. Leijenaar, J. Lenkowicz, F. Lippert, A. Losnegård, K. H. Maier-Hein, O. Morin, H. Müller, S. Napel, C. Nioche, F. Orlhac, S. Pati, E. A. G. Pfaehler, A. Rahmim, A. U. K. Rao, J. Scherer, M. M. Siddique, N. M. Sijtsema, J. Socarras Fernandez, E. Spezi, R. J. H. M. Steenbakkers, S. Tanadini-Lang, D. Thorwarth, E. G. C. Troost, T. Upadhaya, V. Valentini, L. V. van Dijk, J. van Griethuysen, F. H. P. van Velden, P. Whybra, C. Richter, and S. Löck. The image biomarker standardization initiative: Standardized quantitative radiomics for High-Throughput image-based phenotyping. *Radiology*, 191145, 2020.

Appendix A

Equivariant right angle rotational representation for separable filters

Right-angle rotations of separable filters can be efficiently obtained using permutations and uni-dimensional filter flipping. The flipping of a $1D$ vector $g_1 \in \mathbb{Z}^n$ can be obtained by multiplication with an $n \times n$ exchange matrix J , where

$$J[i, j] = \begin{cases} 1 & \text{if } j = n - i + 1, \\ 0 & \text{otherwise.} \end{cases}$$

Note that a 0 should be appended to any even-sized filter kernel, to allow for equivalence with methods where rotational invariance is reached through rotating the image, see Sec. 3.3. For instance, $g_1 = (1, 2, 3, 4)$ is first extended to $g_1^* = (1, 2, 3, 4, 0)$, and then flipped by computing $Jg_1^* = (0, 4, 3, 2, 1)$.

A.1 2D case

In the $2D$ case, four versions of a filter can be obtained by right-angle rotations (*i.e.* $0, \frac{\pi}{2}, \pi$ and $\frac{3\pi}{2}$). The permutations and flips required to rotate the filter $g[\mathbf{k}] = g_1[k] \otimes g_2[k]$ are listed in the following, where g^θ is a rotation R_θ by θ of the filter g so that $g^\theta = g(R_\theta \cdot)$.

$$\begin{aligned} g^0 &= g_1 \otimes g_2, \\ g^{\frac{\pi}{2}} &= Jg_2 \otimes g_1, \\ g^\pi &= Jg_1 \otimes Jg_2, \\ g^{\frac{3\pi}{2}} &= g_2 \otimes Jg_1. \end{aligned}$$

A.2 3D case

When considering $3D$ filters, 24 right angle rotations are possible (cube symmetries). We use a (k_1, k_2, k_3) (*i.e.* x, y, z) extrinsic Euler notation (α, β, γ) representing a first rotation by α on the k_3 axis, followed by a rotation by β on the k_2 axis and a last rotation by γ on the k_1 axis. We denote $g^{(\alpha, \beta, \gamma)}$ a $3D$ filter g rotated by such angles so that $g^{(\alpha, \beta, \gamma)} = g(R_\alpha R_\beta R_\gamma \cdot)$. The flips and permutations of the $1D$ vectors required to obtain these 24 rotations of a separable filter

$g[\mathbf{k}] = g_1[k] \otimes g_2[k] \otimes g_3[k]$ are listed in the following.

$$\begin{aligned}
g^{(0,0,0)} &= g_1 \otimes g_2 \otimes g_3, \\
g^{(0,\frac{\pi}{2},0)} &= Jg_3 \otimes g_2 \otimes g_1, \\
g^{(0,\pi,0)} &= Jg_1 \otimes g_2 \otimes Jg_3, \\
g^{(0,\frac{3\pi}{2},0)} &= g_3 \otimes g_2 \otimes Jg_1, \\
g^{(\frac{\pi}{2},0,\frac{\pi}{2})} &= g_2 \otimes g_3 \otimes g_1, \\
g^{(\frac{\pi}{2},0,\frac{3\pi}{2})} &= g_2 \otimes Jg_3 \otimes Jg_1, \\
g^{(\frac{\pi}{2},0,0)} &= g_2 \otimes Jg_1 \otimes g_3, \\
g^{(\pi,0,0)} &= Jg_1 \otimes Jg_2 \otimes g_3, \\
g^{(\frac{3\pi}{2},0,0)} &= Jg_2 \otimes g_1 \otimes g_3, \\
g^{(0,\frac{\pi}{2},\frac{3\pi}{2})} &= Jg_3 \otimes Jg_1 \otimes g_2, \\
g^{(0,\frac{\pi}{2},\pi)} &= Jg_3 \otimes Jg_2 \otimes Jg_1, \\
g^{(0,\frac{\pi}{2},\frac{\pi}{2})} &= Jg_3 \otimes g_1 \otimes Jg_2, \\
g^{(\frac{\pi}{2},\pi,0)} &= Jg_2 \otimes Jg_1 \otimes Jg_3, \\
g^{(\pi,\pi,0)} &= g_1 \otimes Jg_2 \otimes Jg_3, \\
g^{(\frac{3\pi}{2},\pi,0)} &= g_2 \otimes g_1 \otimes Jg_3, \\
g^{(0,\frac{3\pi}{2},\frac{\pi}{2})} &= g_3 \otimes Jg_1 \otimes Jg_2, \\
g^{(0,\frac{3\pi}{2},\pi)} &= g_3 \otimes Jg_2 \otimes g_1, \\
g^{(0,\frac{3\pi}{2},\frac{3\pi}{2})} &= g_3 \otimes g_1 \otimes g_2, \\
g^{(\pi,0,\frac{\pi}{2})} &= Jg_1 \otimes g_3 \otimes g_2, \\
g^{(\frac{3\pi}{2},0,\frac{\pi}{2})} &= Jg_2 \otimes g_3 \otimes Jg_1, \\
g^{(0,0,\frac{\pi}{2})} &= g_1 \otimes g_3 \otimes Jg_2, \\
g^{(\pi,0,\frac{3\pi}{2})} &= Jg_1 \otimes Jg_3 \otimes Jg_2, \\
g^{(\frac{3\pi}{2},0,\frac{3\pi}{2})} &= Jg_2 \otimes Jg_3 \otimes g_1, \\
g^{(0,0,\frac{3\pi}{2})} &= g_1 \otimes Jg_3 \otimes g_2.
\end{aligned}$$