

TR-QC-02-2017

Spatial Model Checking for Medical Imaging - Preliminary Version

Revision: 1.0; Mar 10, 2017

Author(s): Fabrizio Banci Buonamici, Gina Belmonte, Vincenzo Ciancia, Diego Latella, Mieke Massink

Publication date: Mar 10, 2017

Funding Scheme: Small or medium scale focused research project (STREP)

Topic: ICT-2011 9.10: FET-Proactive 'Fundamentals of Collective Adaptive Systems' (FOCAS)

Project number: 600708

Coordinator: Jane Hillston (UEDIN)

e-mail: Jane.Hillston@ed.ac.uk

Fax: +44 131 651 1426

Part. no.	Participant organisation name	Acronym	Country
1 (Coord.)	University of Edinburgh	UEDIN	UK
2	Consiglio Nazionale delle Ricerche – Istituto di Scienza e Tecnologie della Informazione "A. Faedo"	CNR	Italy
3	Ludwig-Maximilians-Universität München	LMU	Germany
4	Ecole Polytechnique Fédérale de Lausanne	EPFL	Switzerland
5	IMT Lucca	IMT	Italy
6	University of Southampton	SOTON	UK
7	Institut National de Recherche en Informatique et en Automatique	INRIA	France



Contents

1	Introduction	1
2	Background	3
2.1	Spatial logics and model checking	3
2.2	Distance operators	4
2.3	Texture analysis	4
3	A spatial logic for medical imaging	6
3.1	Spatial Models of Medical Images	6
3.2	Syntax and semantics	7
3.3	Model checking	9
4	Examples	11
4.1	Example: segmentation of glioblastoma	12
4.2	Example: segmentation of rectal carcinoma	20
4.3	Validation	25
5	Conclusions and future work	28

1 Introduction

Formal verification of properties of Collective Adaptive Systems (CAS) is a challenging subject. The huge number of considered entities introduces a gap with classical finite-state methods as the number of states grows exponentially. Approximation methods, such as mean-field or fluid-flow approximation, have been proposed to mitigate this aspect (see [9, 8, 48]). Another relevant issue is that of spatial distribution of the considered entities. Entities composing a CAS are typically located, and moving, in a physical or logical space. Collective behaviour is driven by interaction, which is frequently based on proximity. This makes spatial aspects more prominent in the case of CAS than in classical concurrent systems, and leads to the introduction of spatial properties in formal verification of CAS.

Spatio-temporal model checking is a recent trend in Computer Science (see for instance [26, 38, 18, 19, 21, 56, 39]) that uses specifications written in logical languages describing *space* – called *spatial logics* – to automatically identify patterns and graphical structures of interest in a variety of domains, ranging from signals and images to CAS. The research presented in this work stems from the *topological* approach to spatial logics, dating back to the work of Alfred Tarski, who first recognised the possibility of reasoning on space using the mathematical framework of topology as an interpretation environment for *modal logic* (see [69] for a thorough introduction). In this context, formulas are interpreted as sets of points of a topological space, and in particular $\diamond\phi$ is usually interpreted as the points that lie in the *closure* of the interpretation of ϕ . A standard reference is the *Handbook of Spatial Logics* [1]. Therein, several spatial logics are described, with applications far beyond topological spaces; such logics treat not only aspects of morphology, geometry, distance, but also advanced topics such as dynamic systems, and discrete structures, that are particularly difficult to deal with from a topological perspective. In recent work [18, 19], this theoretical development was pushed further to encompass directed graphs¹, resulting in the study of the approach of *spatial logics for closure spaces*. Subsequently, in [17], a spatio-temporal logic, combining *Computation Tree Logic* and the newly defined spatial operators, was introduced. A model checking algorithm has been implemented in the prototype *spatio-temporal*

¹Such work was carried on in the setting of the QUANTICOL FP7 600708 EU-FET project, dealing with efficient analysis methods for CAS.

model checker `topochecker`², with applications to case studies drawn from the domain of *smart cities* [20, 16, 22].

The contribution of Computer Science to the field of medical image analysis is increasingly significant, and will play a key role in future healthcare. Computational methods are currently in use for several different purposes, such as: *Computer-Aided Diagnosis* (CAD), aiming at the classification of areas in images, based on the presence of signs of specific diseases [29]; *Image Segmentation*, tailored to identify areas that exhibit specific features or functions (such as organs or sub-structures) [36]; *Automatic contouring* of Organs at Risk (OAR) or target volumes (TV) for radiotherapy applications [10]; *Indicators finding*, that is, the identification of indicators, computed from the acquired images, enabling early diagnosis, or understanding of microscopic characteristics of specific diseases, or help in the identification of prognostic factors to predict a treatment output [15] [68] (examples of indicators are the *mean diffusivity* and the *fractional anisotropy* obtained from Magnetic Resonance (MR) Diffusion-Weighted Images, or the *magnetization transfer ratio* maps obtained from a Magnetization Transfer acquisition [27] [50]).

Such kinds of analyses are strictly tied to the spatial features of images. In a first *proof-of-concept* study [7], `topochecker` was used for the declarative specification of regions in medical images. The model checker was used to automatically - and efficiently, taking a few seconds to complete the task - identify and colour *glioblastoma* and the surrounding *oedema* in Magnetic Resonance Imaging (MRI) scans, defining the two *regions of interest* in terms of their visual appearance, defined by image features such as proximity, interconnection, and texture similarity. Input to the model checker consists of a precise, declarative, unambiguous logical specification, that besides being fairly close to level of abstraction of an expert description of the process, is also remarkably concise; for example, the specification of the glioblastoma analysis is less than 30 lines long.

Identification of glioblastoma can be located in the context of automatic and semi-automatic methods for segmentation of glioblastoma, that constitute an active research area (see for example [30, 33, 62, 72]); therefore, our work in this case study will be completed in companion papers aimed at clinical validation of the methodology. However, the broader scope of our research line is to enable the *declarative description* and *automatic* or *semi-automatic*, efficient identification of regions in images (such as tumours, infiltrations, organs at risk, lesions, etc.) using spatial formulas specifying relevant features, such as texture or similarity, bound together by spatial constraints, e.g., proximity, boundary properties, distance, etc., that increase the significance and signal-to-noise ratio of the obtained results. The planned developments will also include means for interactive refinement of analysis, based on visual fine-tuning of specific values (e.g. thresholds or distances) that may have a *non-linear* effect on the results of complex queries, with transformative effect on methods that require human interaction - e.g. *interactive segmentation* in preparation for surgery, or *contouring* for radiotherapy planning.

This paper is an extended version of [7], enhancing the results in the glioblastoma case study, presenting a new example (segmentation of *rectum carcinoma*), providing the relevant technical details on the procedures, and detailing the logical language that is used (including distance-based operators and a novel connective aimed at *statistical texture analysis*). Texture analysis, distance, and reachability in space can be freely combined as high-level logical operators with a clear and well-defined topological semantics. The interplay of these aspects is the key to obtain our experimental results. The first technical contribution of this work is the approach to model checking of distance-based operators, based on so-called *distance transforms*, that forms the basis for the definition an efficient algorithm to solve the model checking problem. Asymptotic time complexity of the procedure we propose is linear or quasi-linear, depending on the kind of distance used. This result dramatically improves on simpler model checking algorithms that have quadratic complexity with respect to the size of the problem, making the approach prohibitively expensive with higher resolution images. The second technical contribution is a novel operator for *statistical texture analysis* that is able to classify

²Topochecker: *a topological model checker*, see <http://topochecker.isti.cnr.it>, <https://github.com/vincenzoml/topochecker>

points of the space based on the similarity between the area where they are localised, and a target region, also expressed in logical terms. We consider texture analysis a promising research area for spatial logics and model checking; in this work we focus on statistical methods, as they are robust to affine transformations and provide good results in the case studies that we present, and leave open for future work the investigation in this context of other forms of texture analysis that may be useful in different applications.

A technical introduction to spatial logics, distance-based operators and statistical texture analysis is provided in Section 2. In Section 3, the syntax and semantics of our logic are presented. In Section 4, the two case studies are presented in detail, including, where available, a first assessment of validation. Some concluding remarks are given in Section 5.

2 Background

In this section, we discuss the background knowledge that we use in the technical developments of the paper. In particular, we briefly introduce the topic of spatial logics and spatial model checking, and we detail the use of so-called *distance operators* in such research line. Furthermore, we provide an outline of *statistical texture analysis*, that we employ as a starting point for the introduction of texture analysis operators in spatial logics for imaging applications.

2.1 Spatial logics and model checking

In this section we briefly introduce the subject of *spatial model checking* via *closure spaces*, that forms the core of our declarative approach to medical image analysis. The interested reader is invited to consult [19] for a more comprehensive description of the matter.

Much attention has been devoted in Computer Science to formal verification of process behaviour through *modal* logics and *model checking* (see e.g. [4] and references therein). Modal logics also have a *spatial* interpretation. In so-called *spatial logics*, modal operators are interpreted using the concept of *neighbourhood* in a topological space, enabling one to reason about *points* of the space using familiar concepts such as proximity, distance, or reachability. A comprehensive reference for these theoretical developments is [1]. Transferring the results in the field to applications, and in particular to model checking, requires one to use *finite* models. However, finite topological spaces are not satisfactory in this respect; for instance, they cannot encode *directed* graphs, that may be the object of spatial reasoning in several applications (for instance, consider the graph of roads in a town, including the one-way streets). Extending topological spaces to *closure spaces* (see [34]) is the key to generalise these results; this concept was exploited in [18, 17, 19] to define a spatio-temporal model checking algorithm. A prototype model checker named `topochecker` has been developed and used in several case studies in the context of Collective-Adaptive Systems, in particular for smart transportation [16] and bike sharing systems [20]. The tool analyses spatial models – either in the form of graphs with atomic proposition in nodes, or of digital images – and is able to assign truth values to points that satisfy given formulas. This is done in linear time with respect to the product of the size of the model and the length of spatial formulas.

The employed spatial logic SLCS (see Section 3 for more details) is a modal logic with an operator, called *near*, interpreted as proximity, and the *surrounded* connective, which is a spatial variant of the classical temporal *weak until* operator, able to characterise unbounded areas of space, based on their boundary properties. The *surrounded* connective is similar in spirit to the spatial *until* operator for topological spaces discussed by Aiello and van Benthem in [2, 70], although it is interpreted in closure spaces. Several derived operators may be defined, among which, notably, variants of the notion of *reachability* in space. The combination of SLCS with temporal operators from the well-known branching time logic CTL (Computation Tree Logic) [24], has been explored in [17] (and used in case studies); the logic caters for spatio-temporal reasoning and spatio-temporal model checking. We do

not consider temporal aspects in this work, as we reason on images that are “snapshots” of a patient situation (see 5).

2.2 Distance operators

For medical imaging applications, in this work we add distance-based analysis capabilities to `topochecker`. Distance operators can be added to spatial logics in various ways (see [45] for an introduction). Distances are very often expressed using the real numbers \mathbb{R} . Typically, one considers operators of the form $\mathcal{D}^{e(z)}\phi$ where $e(z)$ is a constraint parameterised by a free variable z , and ϕ is a formula denoting a spatial property. The intended semantics is that point x is a model of $\mathcal{D}^{e(z)}\phi$ if and only if there is a point y satisfying ϕ such that the *distance* d from x to y satisfies the constraint $e(d)$. Logics of *metric spaces* have been introduced in [46]; therein, the constraint $e(z)$ can only be of the form $z \leq k$, where $k \in \mathbb{R}_{\geq 0}$ (*distance at most k*), or $k_1 \leq z \leq k_2$ (*distance included between k_1 and k_2*)³. The latter is called “doughnut operator” in [45]. Notably, the doughnut operator cannot be expressed just using $z \leq k$ in combination with boolean operators.

Spatial models in [19] are *quasi-discrete closure spaces*, that is, a representation of finite graphs in terms of an extension of topological closure operators. In this context, it is natural to consider *shortest path distance*. However, other notions of distance can be more appropriate. For example, *sampling* a multi-dimensional Euclidean space is often done using a *regular grid*, where nodes of a graph are arranged on multiples of a chosen *unit interval* that may vary along each dimension of the space. Nodes are connected by edges using a chosen notion of adjacency (e.g. in 2-dimensional space, one typically uses four or eight neighbours per point, but any choice may be reasonable, depending on the application context). Such graphs can then be *weighted* by associating to each edge the Euclidean distance between the nodes it connects. Graphs with nodes in an Euclidean space and weighted by Euclidean distance are known as *Euclidean graphs* and are naturally equipped with both Euclidean distance between nodes and (weighted) shortest-path distance – which is also called *Chamfer distance* in the particular case of Euclidean graphs with nodes arranged on a regular grid. Let us restrict our attention to the case of regular grids, which is of interest in MI. Euclidean and Chamfer distances obviously divert, no matter how fine is the grid or how many neighbours are chosen in the adjacency relation, unless *all* the pairs of nodes are linked by an edge.

In MI, voxels – with an application-dependent choice of adjacency – form an Euclidean graph, and Euclidean distance is the reference distance between two voxels. Therefore, in this context, Euclidean distance is considered *error-free*, and Chamfer distance is considered an approximation of it. The chosen adjacency determines the precision-efficiency trade-off of the computed distance: the more adjacent voxels are considered, the more precise is the approximation, at the expenses of generating graphs with larger out-degrees. This is illustrated in Figure 1, where we highlight the points satisfying $\mathcal{D}^{z>k}\phi$ in red, for a binary image with only one point satisfying ϕ , located in the centre of the image. In Figure 1a, the result is computed using Euclidean distance. In Figures 1b and 1d, this is done using Chamfer distance with two different adjacency relations; on the right, the characteristic pattern of the percentage error with respect to Euclidean distance are shown. The percentage error $\delta(x)$ for Chamfer distance d is defined for each voxel x as $\delta(x) = \frac{|d_{eucl}(y,x) - d(y,x)|}{d_{eucl}(y,x)}$, where y is the central point of the image.

2.3 Texture analysis

A novel kind of logical operators that we introduce in this work is related to *Texture Analysis* (TA). TA can be used for finding and analysing patterns in medical images, including some that are imperceptible to the human visual system. Patterns in images are entities characterised by brightness, colour, shape, size, etc. TA includes several techniques and has proved promising in a large number of applications in the field of medical imaging [43, 51, 13, 25]; in particular it has been used in *Computer Aided Diagnosis*

³See also [55, 56] for examples of application of such connectives in spatio-temporal signal analysis.

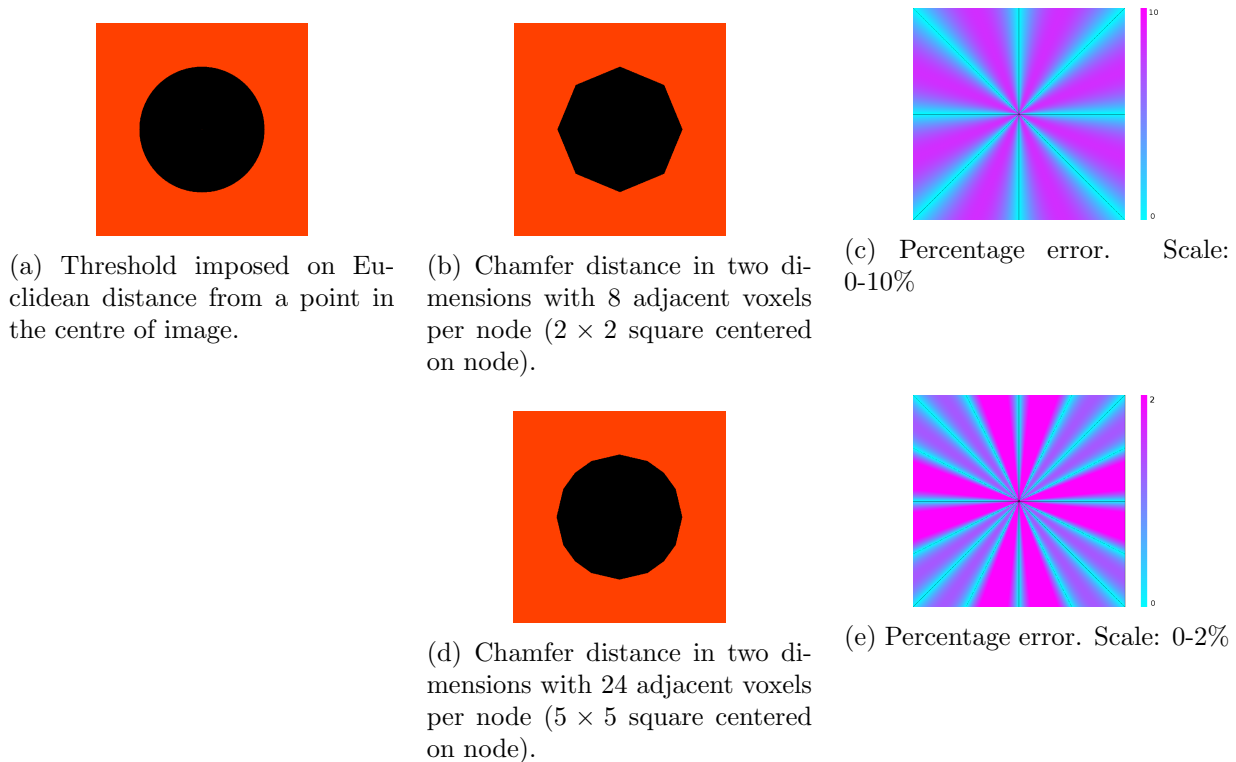


Figure 1: Percentage error of Chamfer distance.

[71, 40, 42] and for classification or segmentation of tissues or organs [14, 60, 59]. In TA, image textures are usually characterised by estimating some descriptors in terms of quantitative features. Typically, such features fall into three general categories: syntactic, statistical, and spectral [43]. Our work is mostly focused on *statistical* approaches to texture analysis. Statistical methods consist of extracting a set of statistical *descriptors* from the distributions of local features in a neighbourhood of each voxel. In particular, we use *first order* statistical methods, that are statistics based on the probability density function (PDF) of the intensity values of the voxels of parts, or the whole, of an image. The statistical distribution of an area A of the image is approximated as the *histogram* H of the grey levels of voxels belonging to A , defined as follows. Given a minimum value m , a maximum value M , and a positive number of *bins* k , let $\Delta = (M - m)/k$. H is a function associating to each *bin* $i \in [1, k]$ the number of voxels that have intensity in the (half-open) interval $[(i - 1) * \Delta + m, i * \Delta + m)$.

The classical statistical approach to TA makes use of statistical indicators of the local distribution of image intensity around each voxel, such as *mean*, *variance*, *skewness*, *kurtosis*, *entropy* [63]. Although such indicators ignore the relative spatial placement of adjacent voxels, statistical operators are useful in MI as their application is invariant under transformations of the image. In particular, first order operators are, by construction, invariant under *affine transformations* (rotation and scaling), which is necessary when analysing several images acquired in different conditions. Nevertheless it is possible to construct features using first order operators, keeping some spatial coherence but losing at least partially the aforementioned invariance [67].

In this work, we introduce a spatial logical operator that compares image regions, in order to classify points that belong to sub-areas in the image where the statistical distribution of the intensity of voxels is similar to that of a chosen reference region. The technique we use improves over classical statistical TA based on first-order statistics, by analysing statistical distributions *as a whole*, whereas classical methods rely on the extraction of specific indicators from distributions. Several similarity measures exist (see [54]), that can be used to compare histograms of images. In particular, as a starting point, we use the *cross-correlation* function (also called *Pearson's correlation coefficient*), that is often

used in the context of *image retrieval*, but is also popular in other computer vision tasks. In MI, cross-correlation is also frequently used for *image co-registration* ([11])⁴.

The cross-correlation cc of two histograms h_1 and h_2 with k bins is defined as follows

$$cc = \frac{\sum_{i=1}^k (h_1(i) - \bar{h}_1) (h_2(i) - \bar{h}_2)}{\sqrt{\sum_{i=1}^k (h_1(i) - \bar{h}_1)^2} \sqrt{\sum_{i=1}^k (h_2(i) - \bar{h}_2)^2}}$$

where \bar{h}_1 and \bar{h}_2 are the means of h_1 and h_2 respectively ($\bar{h} = \frac{1}{k} \sum_{i=1}^k h(k)$).

The value of cc is *normalised* so that $-1 \leq cc \leq 1$. The value $cc = 1$ indicates that h_1 and h_2 are *perfectly correlated* (that is, $h_1 = ah_2 + b$, with $a > 0$); the value $cc = -1$ indicates *perfect anti-correlation* (that is, $h_1 = ah_2 + b$, with $a < 0$). On the other hand, $cc = 0$ indicates no correlation. Note that normalisation makes the value of cc undefined for constant histograms, having therefore standard deviation of 0; in terms of statistics, a variable with such standard deviation is only (perfectly) correlated to itself. This special case is handled by letting $cc = 1$ when both histograms are constant, and $cc = 0$ when only one of the two is constant.

3 A spatial logic for medical imaging

The spatial logic SLCS was proposed in [18, 19] to serve the needs of spatio-temporal verification of Collective-Adaptive Systems. We recall that SLCS is a modal logic interpreted on points of the space, featuring Boolean values and connectives, and the spatial modal operators *near* and *surrounded*. Here, we present the extension of SLCS that we use for medical image analysis. We start by defining formal models corresponding to medical images, then we introduce additional SLCS operators for distance-based thresholding and for statistical texture analysis, and finally we describe our model checking algorithm.

3.1 Spatial Models of Medical Images

Given the two overlapping, but different research lines aimed at applying spatial model checking to CAS and MI, the definition of formal *models* for the proposed extension of SLCS has been driven by the differently specialised kinds of spatial structures that are currently handled by `topochecker`. Spatial models introduced in [18, 19] are based on the mathematical concept of a *closure space* (generalising that of a topological space), encompassing in particular directed or undirected finite graphs. Such standpoint can be further enhanced by considering weighted graphs (see e.g. [56]). Shortest-path distance is naturally defined on such structures. In the weighted case, the length of a finite path is the sum of the weights of all its arcs. In the unweighted case, one just assumes that the “weight” of each arc is 1.

In the following, we restrict our attention to undirected graphs. When such models are instantiated to Medical Images, they take the shape of *anisotropic* grids, with nodes embedded in a multi-dimensional Euclidean space (very often two- or three-dimensional). Such structures are treated by `topochecker` as *Euclidean graphs* (see Section 2.2). As we shall see, our logical language includes different logical operators for Euclidean and shortest-path distances, mostly by the dual nature of our work, aiming also at CAS (more on this in the conclusions).

Formally, the information needed to represent a medical image in the model checker is encoded by a so-called *quasi-discrete closure model* enriched with metric and quantitative information.

Definition 1. Given a finite or countable set P of *proposition letters*, a *quasi-discrete metric closure model* is a structure $((X, \mathcal{C}_{\mathcal{R}}), v, d)$, where X is a finite set, \mathcal{R} is a symmetric relation on X , $\mathcal{C}_{\mathcal{R}}$:

⁴In image processing, the problem of *co-registration* is that of mapping two images coming from different sources to the same spatial domain, by finding transformations of the considered images that make given image features coincide.

F ::=	TT	true
	FF	false
	p	proposition letter
	p \bowtie n	atomic constraint
	!F	not
	F₁ & F₂	and
	F₁ F₂	or
	NF	closure
	IF	interior
	F₁ S F₂	surrounded
	MDDT(F, \bowtie n)	shortest-path distance
	EDT(F, \bowtie n)	Euclidean distance
	SCMP(p₁, F₁, rad, \bowtie thr, vmin, vmax, nbins)(p₂, F₂)	statistical similarity

Figure 2: Syntax of the spatial logic that we use in this work.

$\mathcal{P}(X) \rightarrow \mathcal{P}(X)$ is the quasi-discrete closure operator uniquely identified by the relation $\mathcal{R} \subseteq X \times X$ (or equivalently, (X, \mathcal{R}) is a finite graph⁵), $v : X \times P \rightarrow \mathbb{R}_{\geq 0}$ is a quantitative valuation of proposition letters, and $d : X \times X \rightarrow \mathbb{R}_{\geq 0}$ is a *distance* function, obeying to the well-known axioms of metric spaces (namely, *identity of indiscernibles*, *symmetry* and *triangular inequality*). A weight function is considered implicitly defined by such information, associating to each arc $(x, y) \in \mathcal{R}$ the weight $w(x, y) = d(x, y)$.

In Definition 1, the distance d is meant to denote Euclidean distance, whereas shortest-path distance is implicitly defined by the relation \mathcal{R} . Recall that a medical image is composed of voxels, that are parallelepipeds sharing vertices, edges, or faces with neighbours. Euclidean distance is therefore determined for a medical image by the anisotropic voxel dimensions, that are recorded in images by the machines capturing them. The chosen adjacency relation \mathcal{R} is instead use-case-dependent. A frequent choice in MI – and the default in `topochecker` – is to let the neighbours of voxel x to be (x itself and) all the voxels that have at least one vertex in common with x (also known as *Moore neighbourhood*, or *8-adjacency* in two dimensions, and *26-adjacency* in three dimensions); another possibility is to consider (x itself and) those voxels that have a face in common with x (*Von Neumann neighbourhood* or *4-adjacency* in two dimensions); other possibilities include e.g., larger neighbourhoods (see Section 2.2), or neighbourhoods inducing planar connectivity in three-dimensional spaces.

3.2 Syntax and semantics

In the remainder of the paper, we will represent logical operators using the concrete notation of `topochecker`. Here, we introduce the syntax and semantics of the logic, assuming Definition 1.

Definition 2. The syntax of the spatial logics for MI is defined by the grammar in Figure 2, where p, p_1, p_2 range over a finite set of proposition letters P , \bowtie is a comparison operator (one of $=, <, >, \leq, \geq$), and $n, rad, thr, vmin, vmax, nbins$, are (floating point) numeric constants.

Truth values and Boolean operators have the usual intended semantics. Furthermore, with respect to classical presentations of logical formalisms, we employ a mild generalisation of the language of atomic propositions, that we already used in the research line of [20]. By Definition 1, the valuation of atomic proposition letters is *quantitative*. More precisely, it associates to each proposition letter

⁵The presentation of graphs by quasi-discrete closure spaces is convenient in order to borrow concepts from topology; the reader may just think of the underlying graph (X, \mathcal{R}) ; formally, the closure operator is defined as $\mathcal{C}(A \subseteq X) = \{y \in X \mid \exists x \in A. (x, y) \in \mathcal{R}\}$. Conversely, a binary relation \mathcal{R} is uniquely identified by any *quasi-discrete* closure space. See [19] for more details.

and each point a value in the domain of non-negative real numbers. However, the logic itself is *not* quantitative: each formula and atomic proposition is meant to denote a set of points. This is achieved by letting the syntax of atomic propositions consist of a simple constraint language using equality and inequality between proposition letters and real-valued constants. For example, formula $p > k \ \& \ q < h$ denotes all points x of the space such that the value of p is greater than k and the value of q is less than h . Atomic predicate p in isolation is a shorthand for $p = 1$. Such extension is conservative and does not interfere⁶ with the theory of SLCS.

Operator N (“near”) corresponds to the closure operator of topological spatial logics, that is, voxel x satisfies NF whenever there is some voxel y satisfying F with $(x, y) \in \mathcal{R}$. Using imaging terminology, x is in such situation whenever it belongs to the *dilation* of the set of voxels satisfying F , that is, whenever it either satisfies F , or it is adjacent to a voxel satisfying it. Operator I is a derived one, defined as $IF = !N!F$. Voxel x satisfies IF whenever it satisfies F and for all y with $(x, y) \in \mathcal{R}$, also y satisfies F . Such operation corresponds to the *erosion* of the set of voxels satisfying F (or to its *interior*, in topological terminology). Operator S is the *surrounded* operator. The formal semantics of this operator is discussed in detail in [19]. Briefly, voxel x satisfies F_1SF_2 if and only if it satisfies F_1 and, for each path (of adjacent voxels) x_1, \dots, x_n , with $x_1 = x$ and x_n not satisfying F_1 , there is a point x_i , with $i > 1$, satisfying F_2 , and furthermore, all of x_1, \dots, x_{i-1} satisfy F_1 . Informally speaking, x belongs to a set of voxels satisfying F_1 which is separated from points not satisfying F_1 (if there are any) by a barrier of voxels satisfying F_2 .

Operators MDDT and EDT are the distance-based operators. Let us first discuss EDT (*Euclidean Distance Transform*). Let F be a formula satisfied by set of voxels S . Let x be an arbitrary voxel of the image, and define the *distance from S to x* as $d(S, x) = \min_{y \in S} d(y, x)$. Formula $EDT(F, \bowtie n)$ is true at voxel x when $d(S, x) \bowtie n$ (e.g., formula $EDT(F, \leq 10)$ is true at x when $d(S, x) \leq 10$). Operator MDDT (*Modified Dijkstra Distance Transform*) is interpreted in the same way, but it uses shortest-path distance instead of Euclidean distance.

Finally, SCMP is the operator for statistical texture analysis. Consider the formula $F = SCMP(p_1, F_1, rad, \bowtie thr, vmin, vmax, nbins)(p_2, F_2)$, with F_1, F_2 formulas, and \bowtie a binary comparison operator. Informally speaking, for each voxel x , such formula computes a similarity score between the local distribution, in a neighbourhood of x , of the quantitative values of proposition p_1 , and the global distribution of the quantitative values of proposition p_2 on voxels satisfying F_2 . Formula F_2 may be, for example, a formula denoting a small region drawn by hand by an expert, with the aim of identifying larger areas that are similar to it. Such similarity score is compared to the threshold thr using operator \bowtie , and the Boolean result obtained is the truth value of F at point x .

More precisely, F is true at voxel x when x satisfies⁷ F_1 , and the condition $cc \bowtie thr$ holds, where cc is the statistical *cross-correlation* between two *histograms* h_1 and h_2 , having in common the number of *bins* $nbins$, the minimum value $vmin$, and the maximum value $vmax$ (see Section 2.3 for more details). Namely, h_1 is the histogram of the quantitative values of proposition p_1 in the area of radius rad and centre x , whereas h_2 is the histogram of the quantitative values of proposition p_2 in the voxels of the whole image that satisfy F_2 . The minimum and maximum values are aimed at improving the resolution of histograms, by excluding rare peaks in the signal, that may be due to artefacts in acquisition and would result in a high number of empty bins.

For example, formula $SCMP(p_1, TT, 10.0, \geq 0.7, 200, 2000, 100)(TT, p_2)$ is true at voxels centred in a region – of radius 10.0 – where the distribution of the values of p_1 has cross-correlation greater

⁶The machinery that we described can be made compatible with classical presentations, as follows: given a set \hat{P} of proposition letters, with quantitative valuation $\hat{v} : \hat{P} \times X \rightarrow \mathbb{R}_{\geq 0}$, the set of atomic propositions (in the classical sense) P is a countable set, consisting of the constraints with free variables in \hat{P} , and the (classical, Boolean) valuation function $v : P \rightarrow 2^X$ is just evaluation of constraints, which makes use of the assignment \hat{v} .

⁷The usage of F_1 as a restriction in SCMP is redundant from a mathematical standpoint: the definition of SCMP could achieve the same effects if F_1 was used in a logical conjunction, instead than as an argument of the SCMP operator; however, the computation of SCMP is resource-intensive, and the considered regions are typically small and well-specified; F_1 is used as a hint to the model checker, to reduce analysis time.

than 0.7 with the distribution of the values of p_2 in the whole image. In this case, cross-correlation is computed using 100 bins, and taking into account only values between 200 and 2000.

Remark 1. *We do not include examples of formulas in this section, as we will present many in the detailed walk-through of our experiments in Section 4. Therein, some logical patterns are frequently used. The “from-to” operator (see [19] for a detailed explanation of this pattern) is defined as $T(\mathbf{F}_1, \mathbf{F}_2) \triangleq \mathbf{F}_1 \& ((\mathbf{F}_1 | \mathbf{F}_2) \mathbf{R} \mathbf{F}_2)$, where \mathbf{R} is the reachability operator. Such operator, dual to \mathbf{S} , is defined as $\mathbf{F}_1 \mathbf{R} \mathbf{F}_2 \triangleq !(\mathbf{F}_2 \mathbf{S} ! \mathbf{F}_1)$. Briefly, a voxel satisfies $T(\mathbf{F}_1, \mathbf{F}_2)$ if and only if it lies in an area A satisfying \mathbf{F}_1 , and A touches the area B satisfying \mathbf{F}_2 , that cannot be empty. For this reason, in Section 4, we shall suggestively call the T operator “touch” for the sake of readability. Two more patterns are used to filter noise in results. One is formula \mathbf{NIF} , where \mathbf{F} is a formula. The effect of such formula is to capture the regular region included in \mathbf{F} . Voxel x satisfies \mathbf{NIF} if and only if it is adjacent to at least one voxel y satisfying \mathbf{F} which, in turn, is not adjacent to voxels satisfying $!\mathbf{F}$. The effect of such a filter is to eliminate small regions, e.g. those consisting of one voxel, when these are considered noise or artefacts. Another (similar in spirit) form of filtering is in use in our examples to smoothen regions using a predefined distance. Consider formula $\mathbf{EDT}(!(\mathbf{EDT}(!\mathbf{F}, < \mathbf{k})), < \mathbf{k})$ (or, similarly, with \mathbf{MDDT} in place of \mathbf{EDT} , when the underlying graph is undirected, as in medical images). The voxels satisfying it are those that are closer than k to at least one voxel satisfying \mathbf{F} , and in turn distant at least k from voxels not satisfying $!\mathbf{F}$. Informally speaking, this eliminates areas of diameter smaller than k satisfying \mathbf{F} .*

3.3 Model checking

All the formulas expressible in our spatial logical language may be verified automatically on medical images. Except for the shortest-path distance operator – currently implemented by a quasi-linear procedure – the algorithm that we use has linear complexity in time and space, with respect to the product of the number of voxels in the considered image and the length of the formula (total number of operators and constants). We note in passing that in the more general case of graphs, the size of models for our complexity estimates is the sum of the number of edges and vertices in the graph; indeed, for medical imaging, once the type of connectivity is chosen, the number of edges is proportional to the number of vertices by a constant factor, therefore one can consider as size for asymptotic reasoning just the number of vertices.

The model checking algorithm used by `topochecker` is a *global* one, that is, even though truth of formulas is defined on single voxels, a run of the algorithm returns a truth value for each voxel, considering all voxels at once. This design choice has proven apt to spatial reasoning, in particular in medical imaging, as typical usage requires identification of areas of an image, rather than just checking whether single voxels satisfy certain requirements. The algorithm proceeds by induction on the structure of formulas. At each sub-formula \mathbf{F} , a new image layer is created, where all the points satisfying \mathbf{F} are coloured. The algorithm used for the fragment of the logic presented in [18, 19] is described in detail therein; the remaining connectives are \mathbf{SCMP} and the distance operators, that we explain in the following. For \mathbf{SCMP} , the definition is implemented directly; this yields a linear algorithm that analyses the metric ball of specified radius for each voxel.

For distance-based operators, generally speaking, the time complexity of simple model checking algorithms is quadratic in the size of the space (see [56] for an example). However, given a multi-dimensional image equipped with Euclidean distance, global spatial model checking of formulas $\mathbf{EDT}(\mathbf{F}, \bowtie \mathbf{n})$, or $\mathbf{MDDT}(\mathbf{F}, \bowtie \mathbf{n})$ can be done in linear time (quasi-linear time, respectively) with respect to the number of points of the space. This is achieved via so-called *distance transforms*, that are one of the subjects of topology and geometry in computer vision [44], and are extensively used in modern image processing. Consider a multi-dimensional image. The outcome S of computing the truth value of \mathbf{F} on each point of a model is a binary (multi-dimensional) image. The binary value stored in each point corresponds to the truth value of \mathbf{F} at that point. From this data, it is possible to define a transformed image, called the *distance transform* of S , such that in every point x , a value $d_x \in \mathbb{R}_{\geq 0}$ is

stored. The value of d_x is the distance from S to x . There exist both exact and approximate algorithms for computing distance transforms, that are usually classified by their asymptotic complexity, computational efficiency, and possibility of parallel execution. In particular, there are effective linear-time algorithms [53, 31].

Model checking of formula $\text{EDT}(\mathbf{F}, \bowtie \mathbf{n})$ (resp. $\text{MDDT}(\mathbf{F}, \bowtie \mathbf{n})$) is done in one pass over the considered structure (therefore, in linear time), selecting the quantitative values v of the distance transform that satisfy $v \bowtie \mathbf{n}$. The study of efficient algorithms for computing distance transforms is a well-established research line in Computer Graphics (see [23] for an outline of the current state-of-the-art). In `topochecker`, two of these algorithms are currently implemented.

For Euclidean distances, `topochecker` uses the linear algorithm that was proposed by Maurer in [53]. Such algorithm computes Euclidean distance transforms on anisotropic multi-dimensional grids (such as medical images); it has linear complexity, its run-time is predictable, and it is among the most efficient algorithms for the purpose [32]. The general idea of the algorithm is to proceed by induction on the number of dimensions of the image. The distance transform problem in $n + 1$ dimensions is reduced to the problem in n dimensions by a technique that relies on multi-dimensional Voronoi maps. For space reasons, we omit further details, and we refer the interested reader to [23], that also studies the algorithm from a theoretical point of view. The specification described therein was closely followed in our implementation.

For shortest-path distances over arbitrary directed graphs, `topochecker` employs a variant of the well-known Dijkstra shortest-path algorithm, called “modified Dijkstra distance transform” in [37]. The pseudo-code of the algorithm is presented in Figure 3. The standard Dijkstra algorithm uses a priority queue sorted by distance from a root node. The queue is initialised to the root node of the considered graph, whose priority is set to 0. In the modified version, when computing the distance transform from a set of nodes identified by formula \mathbf{F} , the queue is initialised with all the nodes that satisfy \mathbf{F} and have an outgoing edge reaching a point not satisfying \mathbf{F} ; all such nodes have priority 0. The algorithm then proceeds as the standard algorithm. As a result, after termination, each node of the graph is labelled with the least shortest-path distance from the nodes satisfying \mathbf{F} , as required by the specification. Asymptotic run-time of this procedure is not linear but quasi-linear due to the usage of a priority queue. In this respect, research is still active to optimise the procedure in specific cases (see e.g. [66]). However, the effective run-time of the algorithm is highly dependent on the structure of the considered graph and the chosen implementation of data structures; in our tests on Euclidean graphs, this procedure is typically faster than computing the Euclidean distance transform using Maurer’s algorithm, although a precise comparison of efficiency between the two algorithms is obviously implementation dependent, and also depends on the precision-efficiency trade-off given by the chosen adjacency relation.

The model checker `topochecker` uses *memoization* of model checking of sub-formulas, so that each sub-formula is visited exactly once for each model. The cache is stored on-disk, leveraging incremental design of complex formulas. The language used by the tool permits parametric abbreviations for formulas, such as $\text{Let } \mathbf{f}(\mathbf{a}, \mathbf{b}, \mathbf{c}, \dots) = \mathbf{F}$, where \mathbf{F} is a formula that can use names $\mathbf{a}, \mathbf{b}, \mathbf{c}$, that are instantiated to formulas when \mathbf{f} is invoked. Model definitions using medical images are introduced in `topochecker` by associating an arbitrary number of proposition names to files containing medical images⁸, as follows:

```
Model "med:img1=file1.nii,img2=file2.nii,...";
```

For example, the quantitative (floating point) value of proposition `img1` at each voxel is equal to the intensity value of the same voxel in `file1.nii`. For this to work, all the loaded images must have the same voxel coordinates (e.g., coming from the same machine and type of acquisition, or after manual

⁸The model loader of `topochecker` currently supports the NIFTI (Neuroimaging Informatics Technology Initiative) file format⁹ (version 1 and 2). In this work, images downloaded from *Radiopaedia.org* in *jpeg* format, and *dicom* images have been converted to NIFTI-1.

```

Function MDDT(G,Phi)

Input:  graph G=(V,E,W) with positive weights given by function W,
        set of nodes V, set of arcs E; formula Phi

Output: vector indexed by nodes in V, containing the shortest-path
        distance of each node from the points satisfying Phi

var L := Empty Ordered Set of Vertices
var D := Vector of Numbers indexed by nodes in V, initialised to infinity

for n in V
  if Phi is true at n
    D[n] := 0
    for m with (n,m) in E
      if Phi is false at m
        insert n in L with priority 0

while (L is not empty)
  pop element n with lowest priority from L
  for m with (n,m) in E
    if (D[n]+W(n,m) < D[m])
      D[m] := D[n]+W(n,m)
      insert or update m in L with priority D[m]

return D

```

Figure 3: Pseudo-code for the *Modified Dijkstra Distance Transform*.

resampling). No resampling is currently done in `topochecker`. Such operations require particular care, and might otherwise introduce artefacts (such as *aliasing*) in intermediate analysis phases, that might be difficult to detect in the final result.

4 Examples

In this section, we illustrate our approach by means of two examples of segmentation in *Magnetic-Resonance* (MR) images, introduced below.

1. Glioblastoma (GBM) tumour and oedema segmentation in images obtained using the MR-FLAIR (*Fluid-attenuated inversion recovery*) pulse sequence; this analysis is carried out in two dimensions.
2. Rectal cancer segmentation in images obtained using the MR-T2w (*T2 weighted*) pulse sequence, and ADC (*Apparent Diffusion Coefficient*) maps. This analysis is carried out in three dimensions.

Remark 2. *Before going into details, it is worth emphasizing that the work we present in this section is aimed at providing an introduction to the analysis capabilities of our logical language, rather than complete case studies. For instance, consider our glioblastoma specification, which is rather concise, consisting of a less than 30 lines long logical specification, and a simple preprocessing step. Although such procedure was successfully tested on five images from different sources and acquired in very different conditions, this is certainly not sufficient to validate our example as a glioblastoma segmentation methodology for future clinical usage. Indeed, given the encouraging preliminary validation results (mentioned in Section 4.3), some work in progress – targeted to clinical applications and the medical community – aims at improving the method, eliminating corner cases in the formulas as much as*

possible, making it robust to different acquisition conditions and properly validating it. More generally speaking, clinical experimentation is the next step in our research program, with respect to the current work.

In this light, we note that numeric thresholds and other parameters (e.g., the number of nested \mathbb{N} constructs in some formulas, the number of bins for statistical analysis, etc.) have been chosen by the medical physicist in charge of the analysis, by expert knowledge on the matter and in some cases by trial-and-error. The values that we use might prove stable in clinical validation (and this is the purpose of the preliminary normalisation of images that we use), but this is not yet to be taken for granted, or even to be expected in more general situations. Instead, parameter calibration on a per-image or per-study basis will be an important subject in our future research. Such calibration may be fully automatic (e.g., through machine learning techniques), but this is just one possibility. It would also make sense to adopt a semi-automatic approach (which is also frequent in state-of-the-art techniques, see e.g. [30, 33, 62, 72]), involving human interaction with an expert to merely calibrate the parameters, rather than performing a full manual segmentation, in order to save a large part of the time (and costs) required for preparation to radiotherapy or surgery.

4.1 Example: segmentation of glioblastoma

In this example we detail the specification of an analysis aimed at the segmentation of glioblastoma and oedema in MR-FLAIR images. The procedure is non-trivial, but every passage is justified by morphological and spatial considerations on the arrangement of parts of the brain.

GBMs are intracranial tumours composed of typically poorly-marginated, diffusely infiltrating necrotic masses. Even if the tumour is totally resected, it usually recurs, either near the original site, or at more distant locations within the brain. GBMs are localised to the cerebral hemispheres and grow quickly to various sizes, from only a few centimetres, to lesions that cover a whole hemisphere. Infiltration beyond the visible tumour margin is always present. In MR T2/FLAIR images GBMs appear *hyperintense* and surrounded by *vasogenic oedema*¹⁰.

Being able to segment tumour and oedema in medical images can be of immediate use for *automatic contouring* applications in radiotherapy and, in perspective, it can be helpful in detecting the invisible infiltrations in Computer-Aided Diagnosis applications.

Segmentation of GBM according to our method is performed in three steps:

1. a preprocessing step (not using `topochecker`), aimed at normalisation of images, to make the choice of thresholds in our experiment applicable to different images;
2. brain segmentation, to limit the area of the image where the tumour is searched for;
3. tumour and oedema segmentation, which is the stated goal of this example.

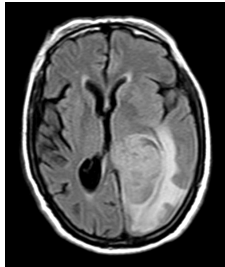
Preprocessing Histograms of grey levels of images¹¹ of the same body part may differ from each other for inter-patient or inter-scanner differences or depending from the actual acquisition volume (Figure 4) or the file format used to store the image¹².

Scaling the intensity of images so that the mean is 1 yields good results on different images. This is accomplished by dividing the intensity of each voxel by the average of the intensity levels of all the *significant* voxels in the image. A voxel is considered significant when it does not belong to the

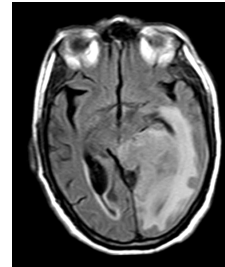
¹⁰Vasogenic oedema is an abnormal accumulation of fluid from blood vessels, which is able to disrupt the blood brain barrier and invade extracellular space

¹¹Histograms of images are constructed as maps from intervals of intensity values to natural numbers representing the number of voxels in the image with corresponding intensity level (see Section 2). The histograms that we show have been normalised so that the area below the curve is 1.

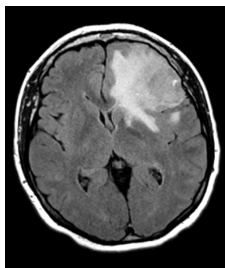
¹²For instance, *jpeg* images, as downloaded from *Radiopaedia.org*, typically use 8-bit precision (typical range 0-255) (see Figure 4) whereas *dicom* images saved by scanners typically use 12 or 16-bit (for MR images typical range is 0-4096 or 0-65536, respectively) (see Figure 5).



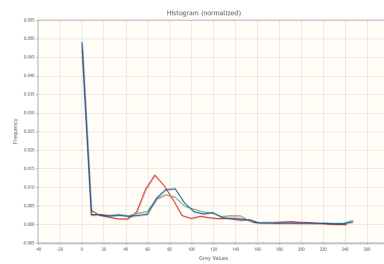
(a) Case courtesy of Dr. Ahmed Abd Rabou, Radiopaedia.org, rID: 22779.



(b) A different slice of the acquisition in Figure 4a.

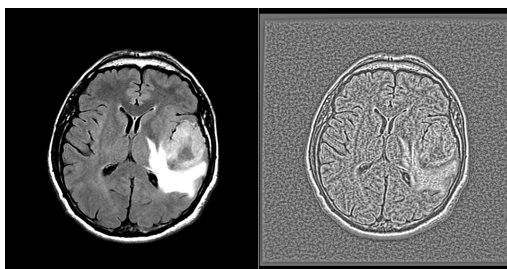


(c) Case courtesy of A.Prof Frank Gaillard, Radiopaedia.org, rID: 5292.

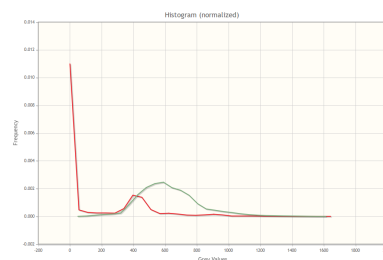


(d) Histograms of Figure 4a (blue), 4b (green), 4c (red).

Figure 4: Slice of MR-FLAIR brain acquisition of different patients and corresponding histogram.

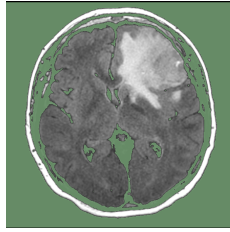


(a) A slice of MR acquisition of brain and its equalized version.

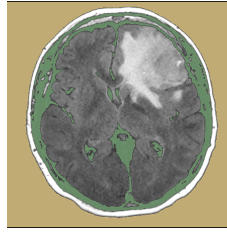


(b) Histograms of grey levels of the original (red) and equalized (green) version of image in 5a

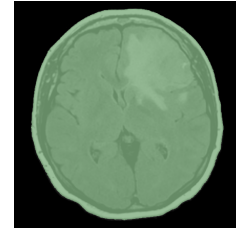
Figure 5: Effect of histogram equalization.



(a) Effect of thresholding using the first part of histogram (green).



(b) Sub-mask that touches the border of the image (yellow).



(c) Mask (green) of image excluding the background.

Figure 6: Finding the mask for normalisation.

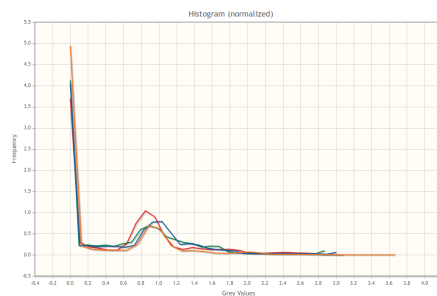


Figure 7: Histograms of normalised version of images in Figure 4a (blue), Figure 4b (green), Figure 4c (red) and Figure 5a (orange).

background. Significant voxels are selected using a mask (consisting of the green area in Figure 6c). We note in passing that *equalization of histograms* is another form of normalisation, frequently used for texture analysis ([41]). We do not use this method as it changes the relationship between grey levels of different structures in the image (as shown in Figure 5), that we use rather prominently for differentiating different tissues; normalisation of image intensity is sufficient for our purposes.

To build the mask in Figure 6c we start from the observation that the background (corresponding to the air surrounding the head of a patient) is darker than the rest of the image, so it mostly contributes to the initial part of the histogram. This situation is witnessed in the histogram by a peak close to 0. A cutting level is thus selected for each image as the value immediately following such peak. By a threshold using such level, it is possible to isolate the background, by separating it from the head (Figure 6a). Note that the obtained mask also includes cerebrospinal fluid (CSF) and bone. The part of the mask that touches the boundary of the whole image is then selected (Figure 6b) and inverted (that is, it is used to *exclude* the external background), obtaining the mask in Figure 6c, which is finally used to select the significant voxels to compute the mean value for normalisation. Figure 7 shows the histograms of images after normalisation.

Brain segmentation Using `topochecker`, we perform a segmentation of the brain to limit the search area of the tumour. This improves the accuracy of the output (e.g., avoiding areas in bone marrow or skull) and reduces computing time.

Brain segmentation in MR images is an important topic on its own in many applications in neuroimaging; several methods have been proposed and research is still ongoing (see e.g. [49, 28]). In contrast, we use a simple method, tailored to loosely identify a region in MR FLAIR images on which to focus the analysis of GBM. In the process below, we shall fix some thresholds; by virtue of the preprocessing step that we employ, these do not need to be changed for each image.

Intuitively, the model of a patient head that we use to segment the brain in MR FLAIR images is defined as follows

- Darker voxels in the head belong to CSF and bones.
- Brighter voxels belong to adipose tissue surrounding the head and bone marrow. Tumour and oedema are also brighter than the surrounding tissues.
- The brain region is composed by white matter, grey matter, tumour and oedema.
- The brain (excluding tumour) has intermediate intensities and is mainly surrounded by CSF.

The model definition in `topochecker` is as follows.

```
Model "med:FLAIR=GBM-49-NORM.nii";
Let reach(a,b) = !(b S !a);
Let touch(a,b) = a & reach(a|b,b);
```

`GBM-49-NORM.nii` is the NIfTI image of the MR FLAIR acquisition shown in Figure 4c. In the rest of the analysis, it is referred to using the atomic proposition symbol `FLAIR`. The definition of the derived operators `reach` and `touch` comes from Remark 1.

We start by looking for the external border of the head, consisting of skin and adipose tissue. Adipose tissue in the normalised MR FLAIR images has intensity above 1.7. Boundaries of regions in medical images may fail to be closed, due to the presence of occasional thin structures of different intensity (e.g. the bright connections to the venous sinuses that are present in the CSF that surrounds the brain). However, we shall use the `S` operator to identify areas that are surrounded by the adipose tissue; this requires the boundary to be closed. Therefore we use nested applications of the `near` operator `N` to enlarge the boundary in order to close it. In this case, two applications are sufficient (see Remark 2).

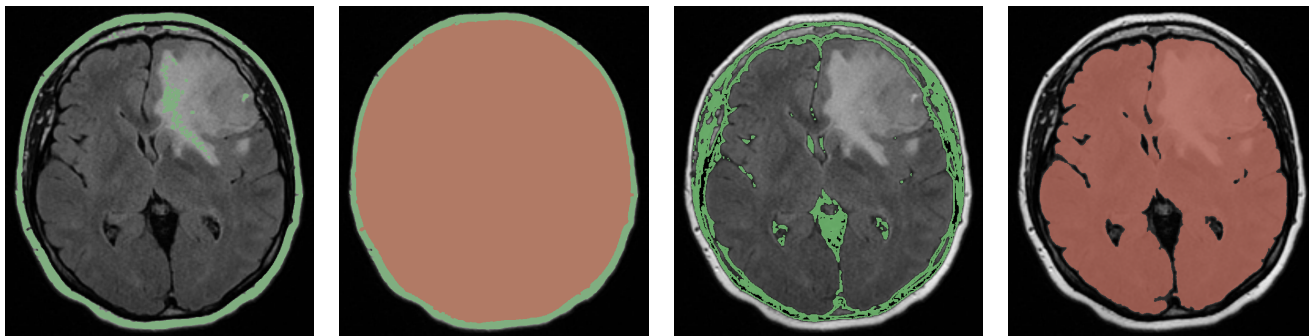
```
Let adipose = N (N [FLAIR>1.7]);
```

In Figure 8a we show the output of `topochecker` on formula `adipose`, where the voxels satisfying the formula are shown in green. Part of the oedema and the bone marrow are segmented together with the adipose tissue. However, this is sufficient to identify the head as the region within the `adipose` border. This is done by the following formulas:

```
Let brainTHR = [FLAIR>0.4];
Let head1 = [FLAIR>0.98] & [FLAIR<1.7];
Let head2 = head1 & (!touch(head1,adipose));
Let external = (!head2) S adipose;
Let head3 = !(external);
Let head3A = (!((!adipose) S head3)) & brainTHR;
Let head = (head3A S head3) | head3;
```

We comment on the most important definitions. Formula `head2` identifies a region which is certainly contained in the head (ensured by the chosen thresholds for `head1`) and does not touch the adipose border. This is aimed at removing artefacts that are present after thresholding `adipose`, due to the continuous intensity decay from the brighter adipose region to the darker background. Formula `external` identifies the part of the image which lies *outside* `adipose` (more precisely, the part of the image which is on the “opposite side” of `adipose` with respect to `head2`). Therefore, formula `head3` characterises the region *inside* `adipose` (on the same side of `head2`). Finally, formula `head` (the brown area in Figure 8b) is used to filter out small holes of variable size (corresponding to the green dots in Figure 8a) consisting of oedema and bone marrow that are erroneously included by formula `adipose` in the previous step.

The next step is to identify the CSF as the dark voxels within the head region. Again, we will need to use formula `CSF` in an application of `S`, therefore we enlarge it using nested `N` operators.



(a) Identification of adipose tissue surrounding the head (formula `adipose`). (b) Identification of the head surrounded by the adipose tissue (formula `head`). (c) Identification of the dark cerebrospinal fluid (CSF) within the head (formula `CSF`). (d) Final brain segmentation (formula `brain`).

Figure 8: `topochecker` output of brain segmentation in the FLAIR MR slice of a brain affected by glioblastoma seen in Figure 4c.

```
Let CSF1 = [FLAIR>0.06] & [FLAIR<0.33] & head;
Let CSF = N (N CSF1);
```

Voxels satisfying formula `CSF` are coloured in green in Figure 8c. Finally, we proceed with segmentation of the brain, as follows:

```
Let flt(a) = MDDT(!(MDDT(!a,<1)),<1);

Let brain1 = brainTHR & ((head&(!CSF)) S CSF) ;
Let brain11 = brain1 & [FLAIR<0.8];
Let brain2 = flt(brain11);
Let brain3 = touch(brainTHR,brain2);
Let brain4 = touch(!CSF,brain3);
Let brain = brain4 S brain3;
```

The brain region is initially (formula `brain1`) defined as the part of the previously identified region `brainTHR` that is included in the part of the head which is not CSF, but is surrounded by CSF. The brighter bone marrow is removed by a further threshold (`brain11`). Then we remove regions with radius smaller than 1mm (formula `brain2`), using the technique explained in Remark 1. Small regions in MRI scans are more sensitive to noise, and in this particular analysis we are interested in identifying the main tumoral area.

Region `brain3` enlarges `brain2` with the voxels of intermediate intensities (`brainTHR`) that *touch* it. Finally, small holes in `brain3` are filled, obtaining region `brain`. Such artefacts are defined as those voxels (`brain4`) that do not belong to CSF, *touch* `brain3` (that could otherwise be an empty region), and are surrounded by `brain3`. In Figure 8d we show the final output of brain segmentation by colouring in brown voxels satisfying formula `brain`.

GBM segmentation In the final part of our analysis, we identify tumour and oedema regions. Since in MR FLAIR, GBM and oedema are hyperintense areas, and the oedema is brighter than the tumour, we start by using thresholds that provide a rough segmentation of the image:

```
Let ut = [FLAIR >= 1.47] & [FLAIR < 2.4] & brain;
Let lt = [FLAIR > 1.17] & [FLAIR < 1.53] & brain;
```

```
Let lt1 = flt(lt);
Let ut1 = flt(ut);
```

In Figure 9b we show in brown formula `ut` (“upper threshold”) that refers to the oedema, and in green formula `lt` (“lower threshold”) that refers to the tumour. Moreover, we elide regions in `ut` and `lt` whose radius is smaller than 1mm, by formulas `ut1` and `lt1`, illustrated, using the same colours, in Figure 9c.

An important constraint, that drastically reduces noise in the output of our analysis, is the *a priori* knowledge that oedema and tumour are very close to each other. We implement this idea using the distance operator as follows:

```
Let utdst = MDDT (ut1,<=2.0);
Let tum1 = touch(lt1,utdst);
Let oed1 = ut1 & reach(utdst,tum1);
```

In this example, we used shortest-path distance as an approximation of Euclidean distance, for the sake of execution speed, as high accuracy for the distance is less important in this particular case. We first define the region at distance less than 2mm from `ut1` (formula `utdst`); then select sub-regions of `lt1` that *touch* `utdst` (formula `tum1`) and sub-regions of `ut1` that can reach formula `tum1` (formula `oed1`). The result is shown in Figure 9d.

Regions denoted by formulas `tum1` and `oed1` are certainly part of the tumour and the oedema, respectively. However, this is not sufficient to cover the whole tumour and oedema. We finalise our analysis by using statistical texture analysis:

```
Let tum2 = (SCMP (x,brain,10,>0.4,0,3.3,100) (x,tum1));
Let oed2 = (SCMP (x,brain,10,>0.4,0,3.3,100) (x,oed1));
Let tum3 = flt(tum2 | tum1);
Let oed3 = flt(oed2 | oed1);
Let tumor = touch(tum3,oed3);
Let oedema = touch(oed3,tum3);
```

Regions `tum2` and `oed2` are sub-regions of the outcome of formula `brain` having a similar histogram (that is, cross-correlation greater than 0.4) to `tum1` and `oed1`, respectively. In the definition of `tum3` and `oed3`, smaller areas (of radius smaller than 1mm) are filtered using the `flt` construct that we explained above.

Finally, `tumor` and `oedema` are defined as being inter-reachable. Figure 9e illustrates the areas defined by `tum3` and `oed3`. Figure 9f is the final output of segmentation.

For completeness, we show the code outputting the resulting images in `topochecker`. Output is saved in image `GBM-seg.nii`, colours for formulas are as specified in the first parameter of the `Check` instructions; a *colour palette* (mapping 8 to brown and 7 to green) has been applied to display the images.

```
Output GBM-seg.nii
Check "8" oedema;
Check "7" tumor;
```

The whole analysis presented in this section has been carried out in 2 dimensions. The same approach also works in three dimensions, with minor modifications. Figure 10 shows some slices of the segmentation of MR-FLAIR acquisition of the patient in Figure 5a, using `topochecker` on the whole 3D volume image. Minor modifications to the model checking session presented in this section were required. For space reasons, we omit the details. However, in Section 4.2 we detail rectum carcinoma segmentation, that has been carried out in 3 dimensions for accuracy reasons.

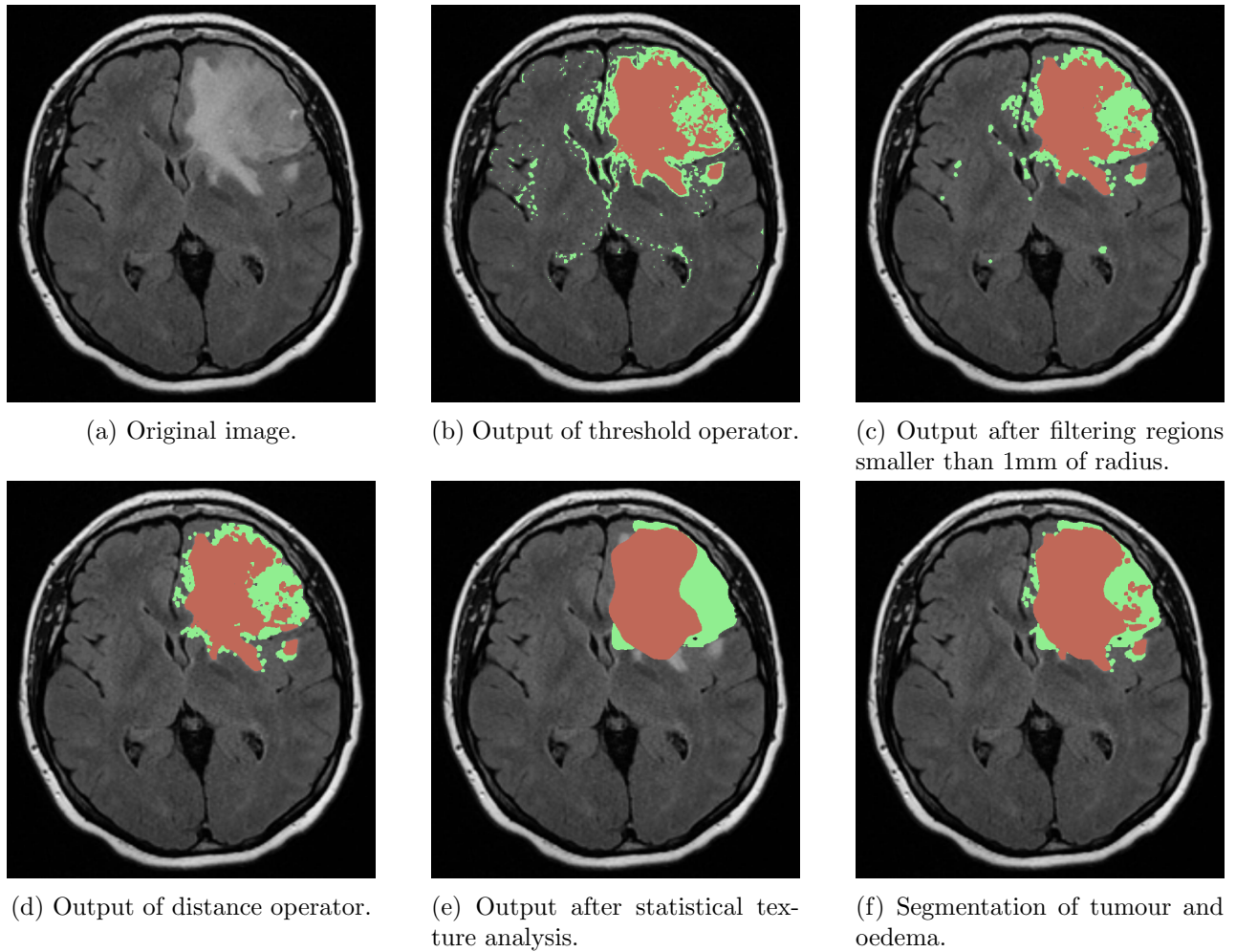


Figure 9: Experimental results of using `topochecker` for segmentation of glioblastoma (green) and oedema (brown) (case courtesy of A.Prof Frank Gaillard, Radiopaedia.org, rID: 5292).

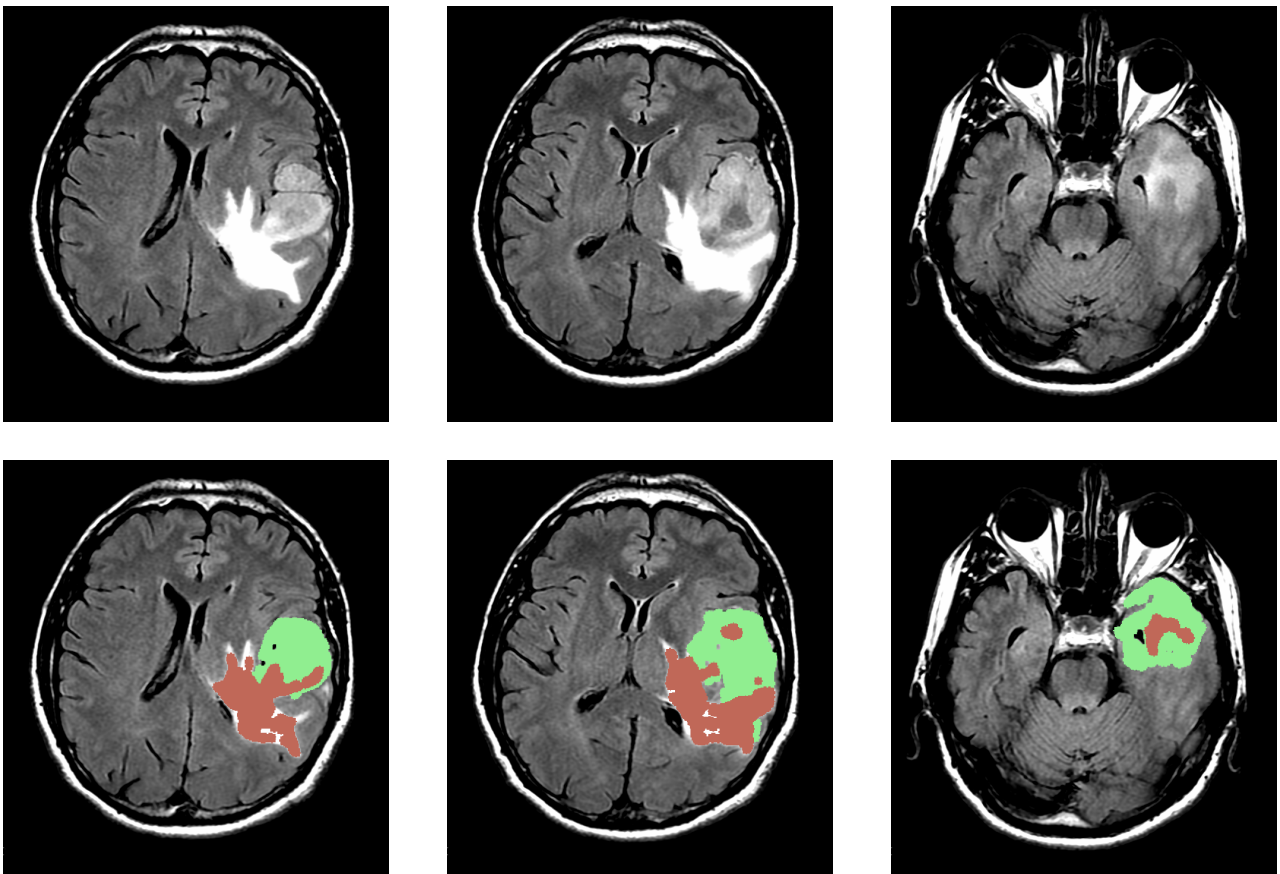


Figure 10: Slices of an image obtained using the method of Section 4 for segmentation of glioblastoma (green) and oedema (brown) on a 3D volume. The top row shows the original slices. The bottom row is the output of segmentation.

4.2 Example: segmentation of rectal carcinoma

In this example we detail an analysis aimed at segmentation of rectal cancer. Rectal carcinoma is a frequent pathology [61] and the survival rate after radical surgery is much improved in case of early diagnosis. Therefore, identifying the tumour by diagnostic imaging has a key role in the output of the treatment of rectal cancer; segmenting the tumour in images is an important step in preparation for radiotherapy. Rectal cancer MR imaging protocols usually include T2w images of the pelvic district, which is considered the key sequence for the diagnosis of rectal cancer. However, several studies have underlined the importance of using DWI (*Diffusion-weighted imaging*) sequences for a more detailed study of the disease [64]. MR-DWI images measure the degree of diffusion of water molecules through imaged tissues. Changes in tissues caused by the growth of a tumour (apoptosis, necrosis, increased vascularity) modify the effective diffusive capacity of water molecules in that area, and DWI is useful to capture this phenomenon. The properties of diffusion are quantified out of DWI images building ADC (*apparent diffusion coefficient*) maps. ADC maps are *hyperintense* in areas where water diffusion is *free* and *hypointense* in areas where water diffusion is *restricted* due to the presence of obstacles. Rectal carcinomas have intermediate grey levels in T2w and are *hypointense* in ADC maps.

Since positioning of *regions of interest* (ROIs) has a considerable influence on tumour ADC values [47], instead of using the T2w images and then co-registering segmentation output to ADC maps, we perform the segmentation of rectal cancer directly on ADC maps for more accurate results.

Differently from Section 4.1, segmentation of rectal cancer is performed using the 3D volume of the image as a whole, rather than considering separate slices. In our experimentation, 3D analysis has yielded better results, as the considered regions are rather small and reasoning simultaneously on different slices maximises the information which is available to each analysis pass. The segmentation process is done in four steps:

1. preprocessing (not using `topochecker`), aimed at normalisation of images;
2. rectum segmentation in T2w images, to limit the area of the image where the tumour is searched; rectum segmentation is done in T2w as the contrast of ADC is not sufficient to properly distinguish organs;
3. co-registration of rectum segmented in T2w to ADC, using patient positioning information that is stored in images by the scanner;
4. tumour segmentation in ADC map, which is the stated goal of this example.

Preprocessing Figure 11 shows one axial and one sagittal view of T2w and ADC acquisitions.

Since the FOV (*Field of View*) of the T2w acquisition lies entirely within the patient body (Figure 11a and Figure 11b), normalisation of the T2w volume is obtained dividing the grey level of each voxel by the average of voxel intensities. For ADC maps instead, a mask is created using a procedure similar to that described in Section 4.1. However, we used the DWI images to obtain the mask, as in the ADC maps the background is very noisy (see Figure 12 – note that DWI and ADC masks have the same coordinate system).

Rectum segmentation The model definition for rectum segmentation defines the atomic proposition symbol T2 which is used throughout the analysis.

```
Model "med:T2=T2-NORM.nii";
```

We slightly change the definition of `flt` (defined in Section 3, and used in Section 4.1 to remove small regions attributed to noise). We consider regions that only appear on one slice as noise, even when these are not filtered out by the previous definition; such regions are thus removed by nested application of I and N on top of the previous definition.

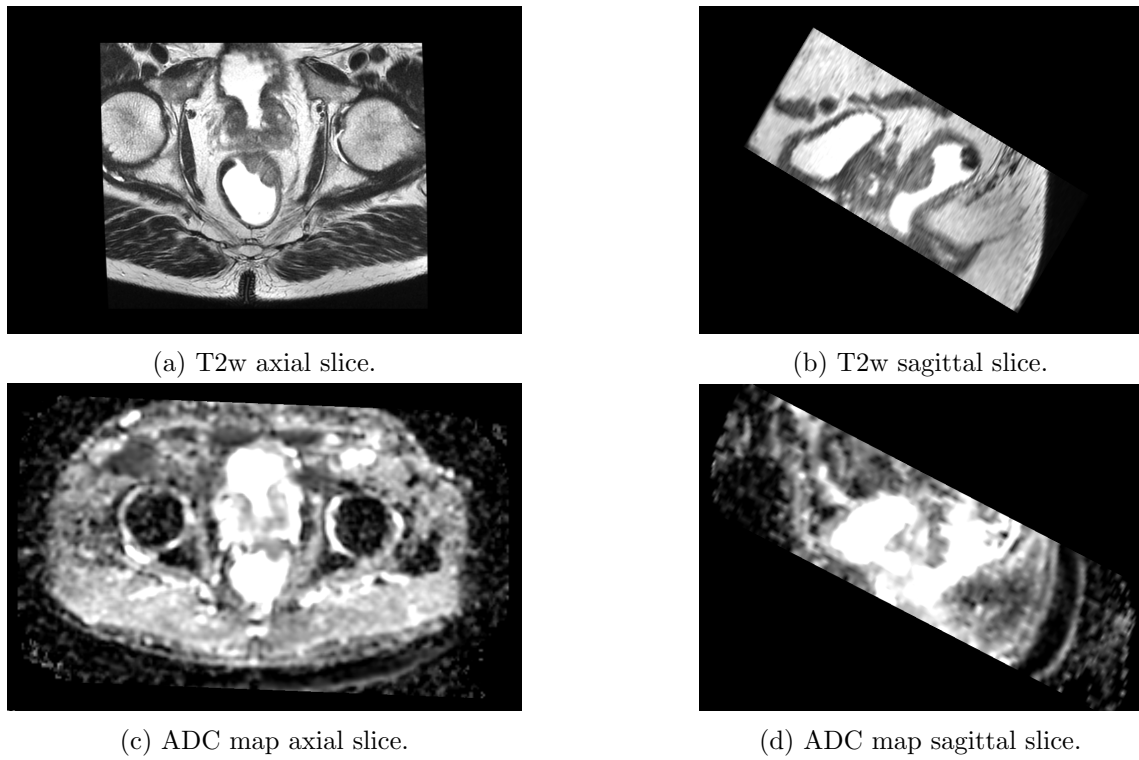


Figure 11: Rectum acquisition

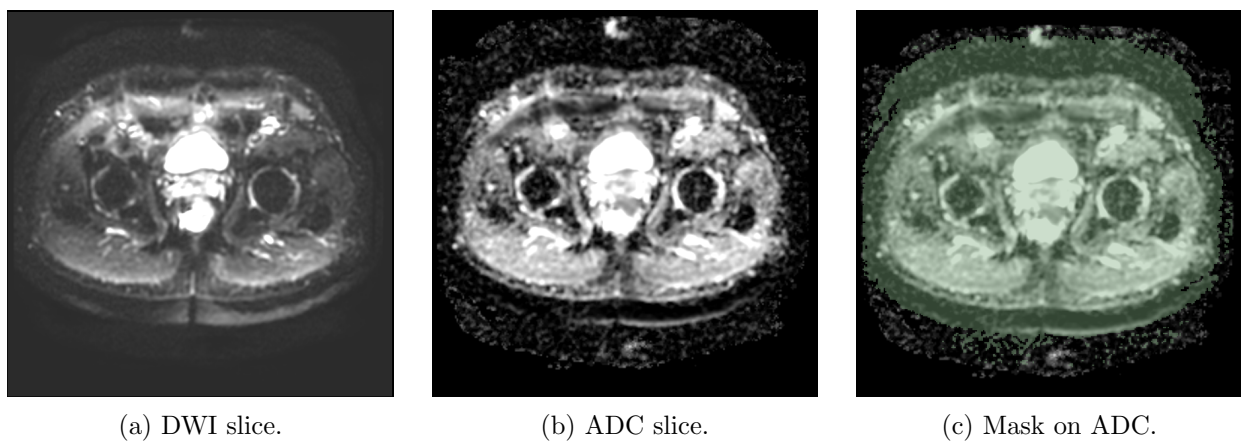


Figure 12: ADC mask

```
Let flt3D(a) = N (I (MDDT(! (MDDT(!a,<1)),<1)));
```

The area corresponding to the rectum in T2w images is identified as the union of a *hyperintense* region, called `hyperT2r`, a *hypointense* region, called `hypoT2r`, and region having intermediate intensity, called `intermT2r`, that are close to each other (detected using the `touch` operator, corresponding to the T connective described in Remark 1).

The aforementioned hyperintense region is defined as `hyperT2r` below.

```
Let hyperT2=flt3D([T2>1.6]);
Let hyperT2Super = flt3D([T2>2.5]);
Let hyperT2r = touch(hyperT2,hyperT2Super);;
```

The hypointense region `hypoT2r` is defined below as being within 5mm from `hyperT2r`.

```
Let hypoT2 = flt3D([T2>0.17] & [T2<0.5]);

Let hyperT2rS = MDDT(hyperT2r,<5);
Let hypoT2r = touch(hypoT2,hyperT2rS);
```

Finally, the region of intermediate intensity `intermT2r` is defined as follows.

```
Let rectum1S = MDDT(hyperT2r | hypoT2r,<5);

Let intermT2 = flt3D([x>0.9] & [x<1.4]);
Let intermT2r = touch(intermT2,rectum1S);
```

The segmented rectum (formula `rectum` below) is defined as the union of the three regions (the green, brown and red areas in Figure 13); the area is expanded, in formula `rectumS`, to cater for loss of precision that occurs in the co-registration to the ADC map (see the green area in Figure 14).

```
Let rectum = hyperT2r | hypoT2r | intermT2r;
Let rectumS = MDDT(rectum,<5);
```

Co-registration Co-registration between the T2w and the ADC maps makes use of the orientation information stored in dicom header. T2w and DWI images are acquired in the same session. The *dicom* header stores the necessary information to translate image coordinate systems to the scanner (world) coordinate system. More precisely, each voxel has coordinates (i, j, k) within the image and dimension (ps_i, ps_j, ps_k) . In addition, also the correspondent position of the voxel in world coordinates is stored in the header. Using such information, the coordinates (i, j, k) of each voxel within the image is mapped to the position (x, y, z) of the voxel in world coordinates (Figure 15).

In order to co-register the ROI of rectum segmented in T2w to the ADC map, we map the image coordinates $(i, j, k)_{T2}$ of the T2w image to the world coordinates (x,y,z) and back to the image coordinates $(i, j, k)_{ADC}$ of the ADC map. In Figure 16, the green area represents voxels on the ADC map that correspond to voxels in T2w satisfying `rectumS`.

Tumour segmentation Tumour segmentation is finally performed on the ADC map. Below, we load the ADC map (ADC) and the rectum segmented on T2w and co-registered to ADC (ROI). We define formula `rectumS` selecting voxels defined in ROI.

```
Model "med:ADC=ADC-norm.nii,ROI=ROI_T2-2-ADC.nii";

Let rectumS=[ROI>0];
```

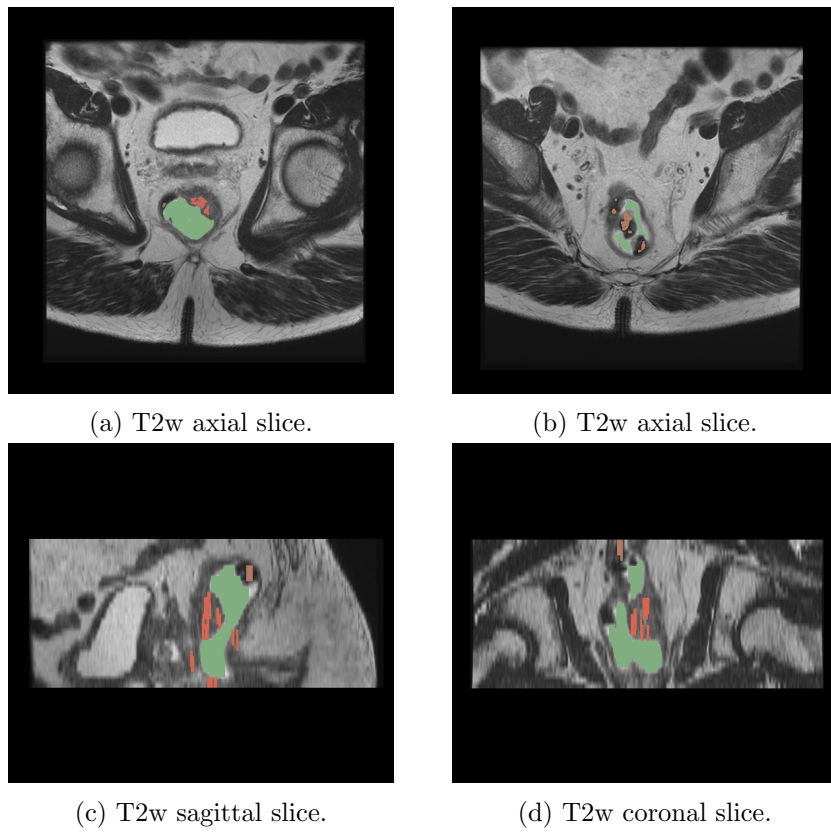


Figure 13: Hyperintense (green), hypointense (brown) and intermediate intensity (red) regions used to segment rectum in T2w.

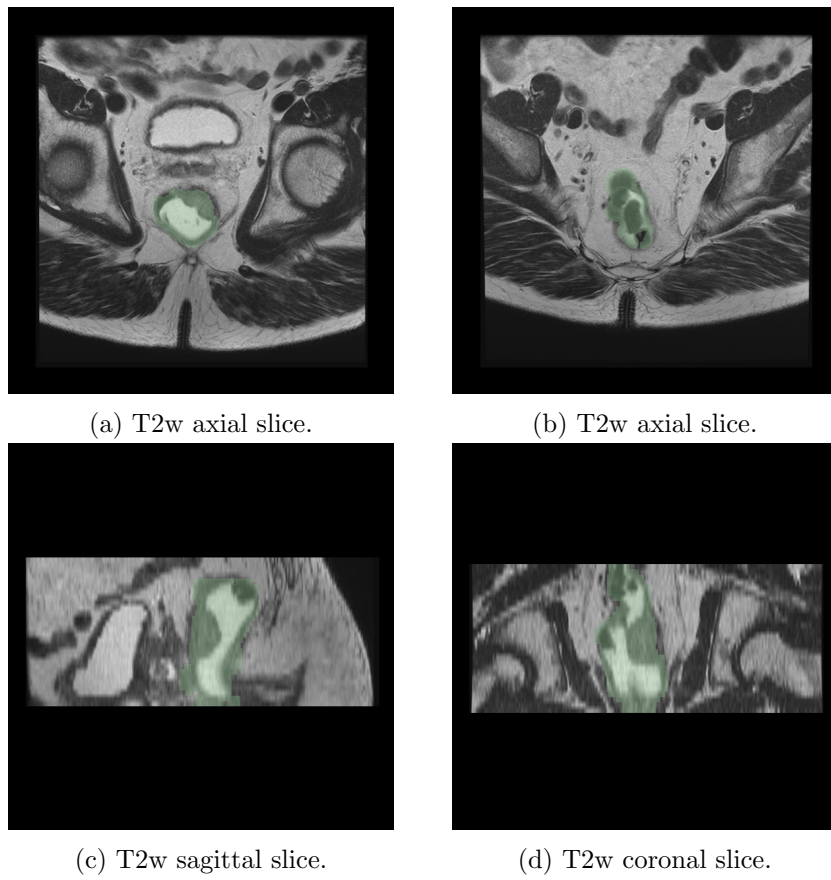


Figure 14: Final output of segmentation of rectum in T2w. In green `rectumS`.

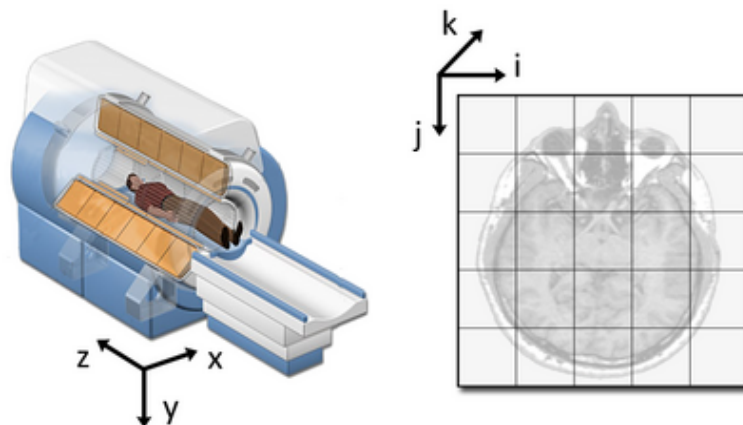


Figure 15: World (x, y, z) and image (i, j, k) coordinate systems. Image based on image shared in https://www.slicer.org/wiki/Coordinate_systems

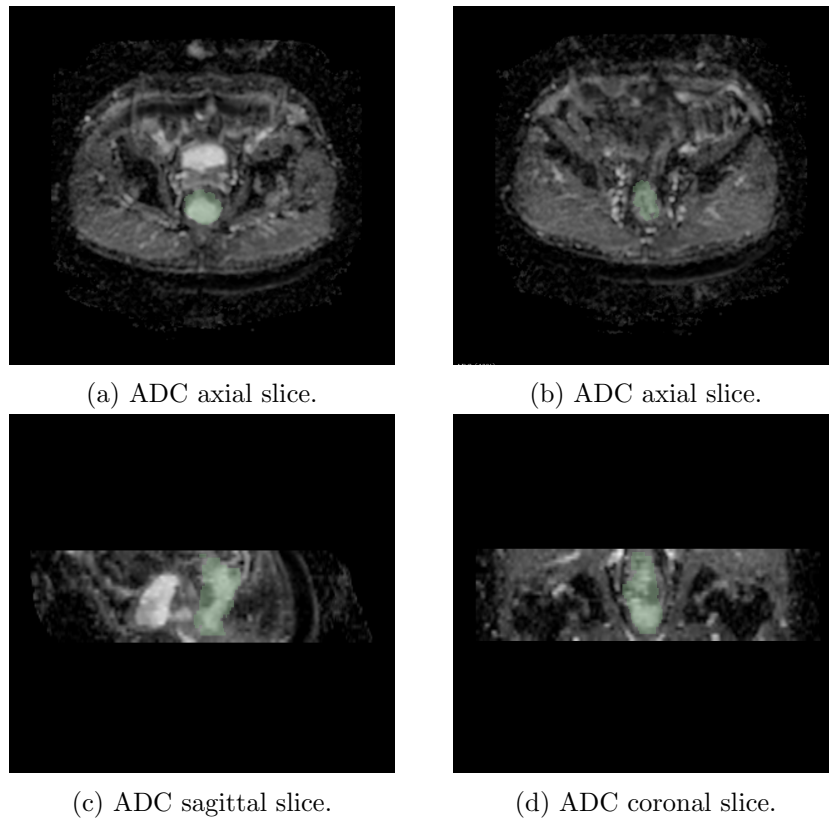


Figure 16: Co-registration of rectum ROI (green) segmented in T2w to ADC map.

We delineate the initial estimate of the tumour based on grey levels (the green area in Figure 17a-17f).

```
Let tumor1 = [ADC>0.96] & [ADC<1.56];
Let tumor2 = flt3D(tumor1);
```

Formula `tumour3`, below, constrains the tumour region to an area that touches the ROI that has been segmented for the rectum (see the green area in Figure 17g,17h,17i)

```
Let tumor3 = touch(tumor2,rectumS);
```

Statistical texture analysis is then used to find regions that are similar to `tumor3` (cross correlation > 0.8). Search is restricted to areas close to `tumor3` (region of radius 20mm around `tumor3` named `tumorSpace`) (Figure 18).

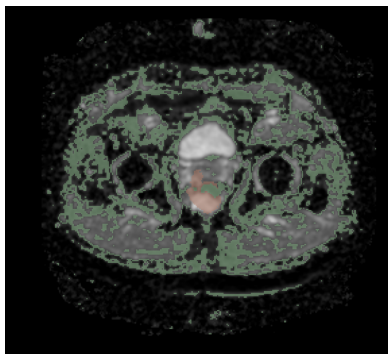
```
Let tumorSpace = MDDT(tumor3,<20);
Let tumorStat = SCMP(ADC,tumorSpace,3,>0.8,0.01,2.7,100) (ADC,tumor3);
```

Finally, the tumour region is the union of `tumor3` and `tumorStat` (Figure 19)

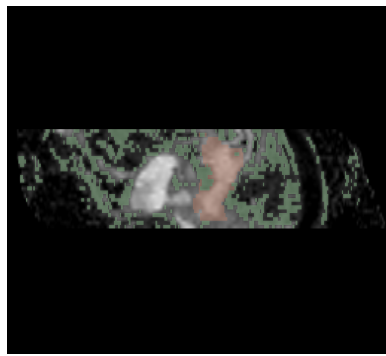
```
Let tumor = tumor3 | tumorStat;
```

4.3 Validation

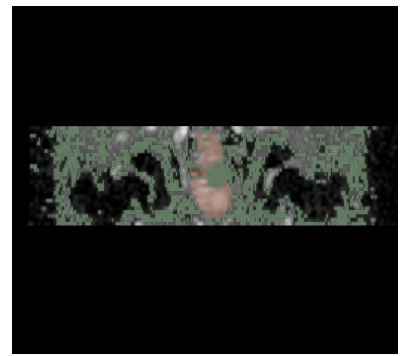
As we already mentioned, clinical validation of segmentation in the two examples that we presented requires a separate clinical study, which is part of the ongoing work in our research program. However, some conclusions can already be drawn from the data we have, both with respect to efficiency and



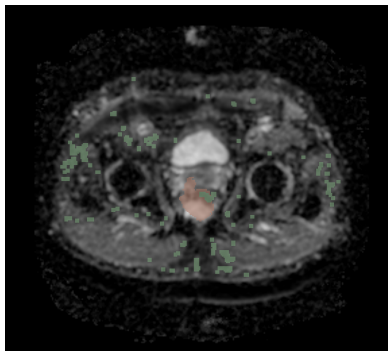
(a) Output of threshold operator on one ADC axial slice.



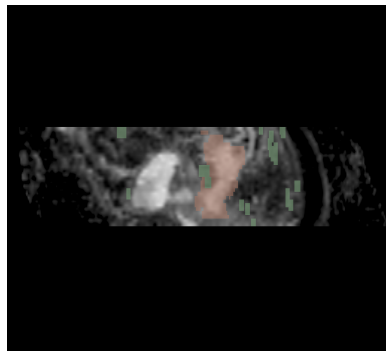
(b) Output of threshold operator on one ADC sagittal slice.



(c) Output of threshold operator on one ADC coronal slice.



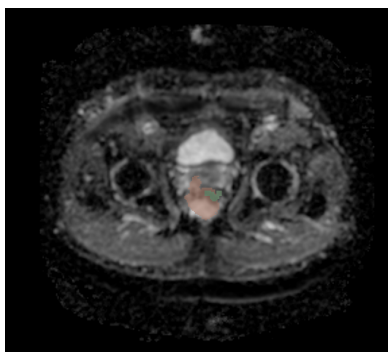
(d) Output of `flt3D` operator on one ADC axial slice.



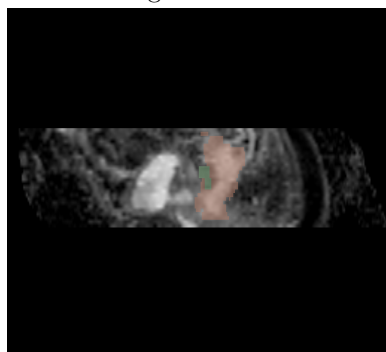
(e) Output of `flt3D` operator on one ADC sagittal slice.



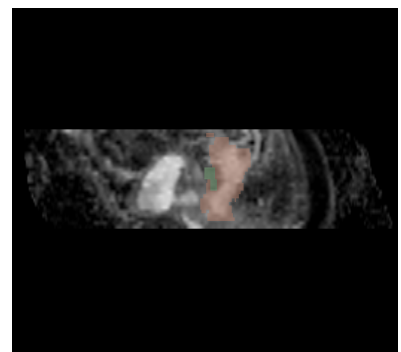
(f) Output of `flt3D` operator on one ADC coronal slice.



(g) Output of `touch` operator on one ADC axial slice.

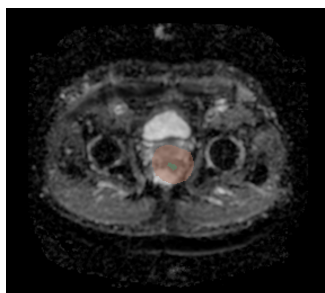


(h) Output of `touch` operator on one ADC sagittal slice.

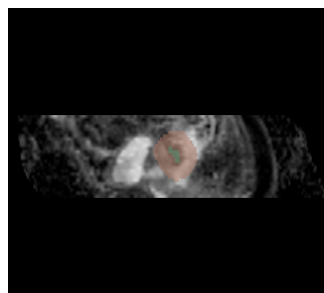


(i) Output of `touch` operator on one ADC coronal slice.

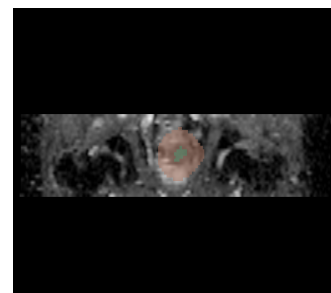
Figure 17: Output of segmentation of the tumour (green) and the rectum (brown).



(a) ADC axial slice.

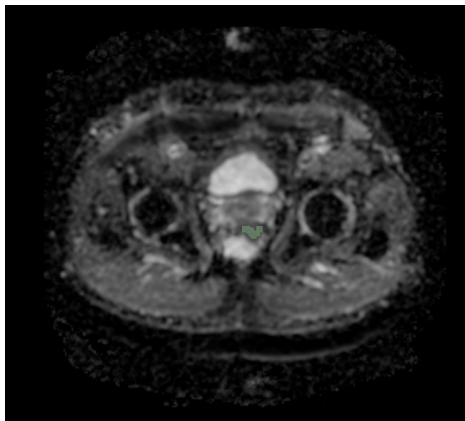


(b) ADC sagittal slice.

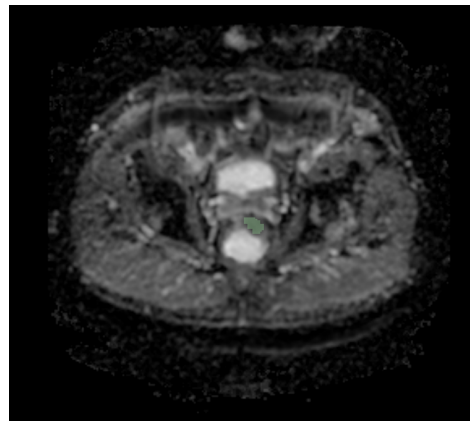


(c) ADC coronal slice.

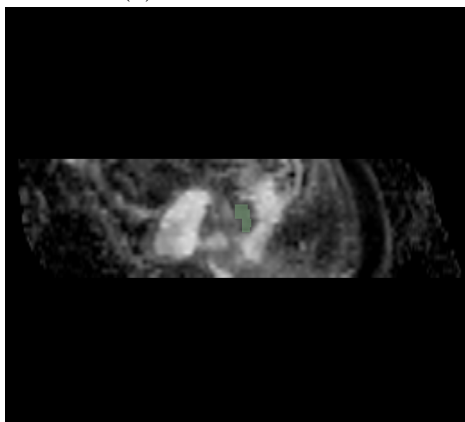
Figure 18: Output of the `SCMP` operator (green) and the searching space `tumorSpace` (brown).



(a) ADC axial slice.



(b) ADC axial slice.



(c) ADC sagittal slice.



(d) ADC coronal slice.

Figure 19: Final output of rectal cancer segmentation.

to accuracy of the obtained results. We do so for the glioblastoma case study, as there is enough literature for a comparison with the state of the art. For rectum carcinoma, less data is available, and clinical testing will be essential in understanding the applicability of the procedure.

Analysis time is proportional to image size. In the glioblastoma example, MRI-FLAIR very often have a slice size of 256×256 voxels, multiplied by 20 – 30 slices. Execution times for analysis of one slice - including preprocessing - on a fairly standard laptop, currently stay below one minute, whereas whole-volume analysis (as done in Figure 10) takes less than 10 minutes. This information may serve as a rough estimate, and indicates that our approach is already in par with the state-of-the art in semi-automatic glioblastoma segmentation procedures (see [33]). We remark that our procedure makes use of a prototype general-purpose model checker, that could be amenable to further optimisation, and that, in contrast with the state of the art, the current procedure is fully automatic due to the normalisation step that we employ (but this may change after clinical validation, see Remark 2).

A preliminary assessment of the quality of the obtained results in the case of glioblastoma was performed for the patient in Figure 10. The patient underwent first surgery and then radiotherapy. We compared our results on the post-surgery FLAIR with target volumes delineated on the pre-treatment CT by one experienced radiotherapist. In particular, we considered the *gross tumour volume* (GTV), i.e. what can be seen or imaged, and the *clinical target volume* (CTV), which contains the GTV, plus a margin for sub-clinical disease spread which therefore cannot be fully imaged [12]. Usually for GBMs the CTV is defined as a 2-2.5 cm isotropic expansion of GTV within the brain. In order to quantify the effectiveness of our segmentation we computed the *Dice coefficient* DC , that we used to measure the morphological similarity between the manual segmentation MS and automatic segmentation AS . The coefficient is defined as $DC = \frac{V(MS \cap AS)}{V(MS) + V(AS)}$, where $V(\cdot)$ is the volume of voxels inside the binary mask. DC ranges from 0 to 1, 0 indicates no overlap and 1 indicates complete overlap.

We co-registered the CT volume to the FLAIR volume and then compared the the GTV contour with the union of oedema and tumour derived by our algorithm and the CTV contour with a 2.5 cm expansion of the union of oedema and tumour derived by our algorithm. We obtained $DS = 0.76$ for GTV and $DS = 0.81$ for CTV. Although a single case does not have clinical significance, these results are very encouraging, and aligned with state-of-the-art methods for automatic and semi-automatic segmentation of GBM [30].

5 Conclusions and future work

Our work is motivated by some considerations about how medical image analysis is carried out. The overall general description of a feature (e.g. the shape and spatial arrangement of parts of an image that exhibit diseases) is often carried out informally, but in a logically structured way (e.g.: “the tumour is lighter than the surrounding brain area, and touches the oedema, whose intensity is a bit darker than the tumour”). Such description is then turned into a series of different analysis passes, performed by specific software tools or by custom programs. The results of such different passes are often integrated by hand or by using hand-crafted scripts. This complex and elaborate process hinders the implementation, and sharing across the medical community, of novel analysis methods that emerge from current research. Logical methods borrowed from Computer Science and in particular the area of *formal methods* could provide means for the unambiguous and precise specification of such procedures. Our current research aims at paving the way and establishing foundational results for such a development to happen.

This work provides a first, promising exploration of logical methods for medical image analysis in the domain of radiotherapy. Logical properties are used as classifiers for points of an image; this can be used both for colouring regions that may be similar to diseased tissues, and therefore being diseased tissue in turn, and for colouring regions corresponding to organs of the human body. Envisaged applications range from *contouring* to *computer-aided diagnosis*.

Our early experiments show that typical analyses carried out using spatial model checking in

medical imaging require careful calibration of numeric parameters (for example, a threshold for the distance between a tumour and the associated oedema, or the size of areas identified by a formula, that are small enough to be considered *noise*, and ought be filtered out). The calibration of such parameters might be done using machine-learning techniques. In this respect, future work could be focused on application, in the context of our research line, of the methodology used in the development of the logic SpaTeL, aimed at signal analysis (see [38, 35, 5, 6]), that pursues *machine learning* of the logical structure of image features. It is worth noting that these methods have been also applied to the detection of tumours in very recent literature [65], whereas [57] is closer to the setting of CAS, applied to biological processes, with an interesting focus on *multi-scale* aspects. However, as our approach is focused on human-intelligible logical descriptions, application of machine learning could as well be focused on the identification of numeric parameters, rather than logic formulas, that may depend on complex features of images.

It is noteworthy that the analysis we designed for glioblastoma segmentation can be used, with mild modifications, also to analyse the whole 3D volume of an image at once. 3D analysis is a relatively new application in medical imaging, leveraging the precision/efficiency trade-off of more classical methods. Furthermore, 3D analysis may be combined with existing applications of *3D printing* in preparation for surgery (see [58]), by providing to practitioners models of a patient's body, with the relevant regions printed in different colours. Such aspects constitute an interesting research line for future work.

Part of our ongoing work consists in identifying novel logical operators that are useful in medical imaging. So far, we only used operators that classify voxels. However, drawing inspiration from the family of *region calculi* (see [1]), one could also classify regions, taking advantage of “collective” observations on sets of voxels that belong to the same area. Some work in this direction is [19], including the definition of operators related to connectedness of regions; further work will be directed to the investigation of properties related to the size of regions, or to their morphological properties. Also, the “distance-bounded surrounded” operator defined in [56] could be useful in medical imaging. A limitation of the model checking algorithm in [56] is its quadratic complexity. Application of distance transforms could yield a linear algorithm also in the case of (variants of) the bounded surrounded operator for the case of images (that is, regular grids), and could be worth further investigation. We recall that **topochecker** is a spatio-temporal model checker. Temporal reasoning could be exploited in future work to consider, for instance, the sequence of acquisitions of a patient in order to reason about the evolution of image features such as tumours, which is very important in radiotherapy applications.

The research line that we present in this paper stems from research in collective adaptive systems and departs from it to direct spatial analysis to medical imaging. We foresee that the novel statistical texture analysis operators and the study of global model checking of distance formulas using distance transforms are of interest when dealing with very large populations that are spread over some spatial structure (e.g. a geographical map). Potential applications include the analysis of statistical properties arising from *gossip protocols* and *disease spreading* models, in which statistical distribution of features in space appears to be relevant.

Our logic is able to predicate on both shortest-path and Euclidean distance at the same time, and **topochecker** implements both operators. Some clarification is needed on this. In MI, shortest-path distances proved useful so far mostly to speed up interactive development; this is mostly implementation-dependent, as the Modified Dijkstra transform that we use (see Section 3.3) currently perform faster than Maurer distance transform in our tests. However, this may change in the near future, as optimised implementations of Euclidean distance transforms could render shortest-path distances unnecessary in the setting of MI. But the definition of both distance operators in our spatial logic is, on the other hand, very relevant for CAS. Even in cases when the nodes of a graph are embedded in an Euclidean space, weights of arcs may be defined in different ways, making shortest-path distance worth being used. Consider e.g. the streets of a town, forming a directed graph (think of one-way streets) embedded in the Euclidean coordinates of the real-world. In such application scenarios, **topochecker** permits one to reason on Euclidean and shortest-path distances in logic formulas at the same time. As the focus of our current research is also to explore the mutual benefits of the research on spatial

analysis of CAS and MI, we consider it worth to keep the two operators in the formal definition of our logic, and to explore the usage of Euclidean graphs in graph-based calculi for CAS such as [3].

The example of glioblastoma that we illustrated in Section 4.1 has immediate practical relevance. As we already mentioned, cleanup and clinical validation of the procedure is in progress. If necessary, the normalisation step that we employ could be improved using state-of-the-art methods (see [52, 49], and the references therein).

Finally, we mention that, in order to leverage the uptake of our research by a wider community, one important step could be to define a public library of pre-defined formula templates, or patterns, for pre-designed formulas, such as parts of the body, that would ease interactive development of complex analyses.

Acknowledgements The authors wish to thank dr. Marco Di Benedetto for suggesting the application of distance transforms to improve the complexity of model checking of formulas with distances, and dr. Valerio Nardone for contributing to the preliminary assessment of our experimental results.

References

- [1] M. Aiello, I. Pratt-Hartmann, and J. van Benthem. *Handbook of Spatial Logics*. Springer, 2007.
- [2] Marco Aiello. *Spatial Reasoning: Theory and Practice*. PhD thesis, Institute of Logic, Language and Computation, University of Amsterdam, 2002.
- [3] Francesco Luca De Angelis and Giovanna Di Marzo Serugendo. A logic language for run time assessment of spatial properties in self-organizing systems. In *2015 IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASO Workshops 2015, Cambridge, MA, USA, September 21-25, 2015*, pages 86–91. IEEE Computer Society, 2015.
- [4] C. Baier and J. Katoen. *Principles of model checking*. MIT Press, 2008.
- [5] E. Bartocci, L. Bortolussi, D. Milios, L. Nenzi, and G. Sanguinetti. *Studying Emergent Behaviours in Morphogenesis Using Signal Spatio-Temporal Logic*, pages 156–172. Springer, 2015.
- [6] Ezio Bartocci, Ebru Aydin Gol, Iman Haghghi, and Calin Belta. A formal methods approach to pattern recognition and synthesis in reaction diffusion networks. *IEEE Transactions on Control of Network Systems*, pages 1–1, 2016.
- [7] Gina Belmonte, Vincenzo Ciancia, Diego Latella, and Mieke Massink. From collective adaptive systems to human centric computation and back: Spatial model checking for medical imaging. In Maurice H. ter Beek and Michele Loreti, editors, *Proceedings of the Workshop on FORmal methods for the quantitative Evaluation of Collective Adaptive SysTems, FORECAST@STAF 2016, Vienna, Austria, 8 July 2016.*, volume 217 of *EPTCS*, pages 81–92, 2016.
- [8] L. Bortolussi, J. Hillston, Latella. D., and M. Massink. Continuous approximation of collective system behaviour: A tutorial. *Perform. Eval.*, 70(5):317–349, 2013.
- [9] Luca Bortolussi and Jane Hillston. Fluid model checking. In *CONCUR 2012 - Concurrency Theory - 23rd International Conference*, volume 7454 of *Lecture Notes in Computer Science*, pages 333–347. Springer, 2012.
- [10] K.K. Brock. *Image processing in radiation therapy*. CRC Press, 2014.
- [11] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM Comput. Surv.*, 24(4):325–376, December 1992.

- [12] Neil G. Burnet. Defining the tumour and target volumes for radiotherapy. *Cancer Imaging*, 4(2):153–161, 2004.
- [13] G. Castellano, L. Bonilha, L.M. Li, and F. Cendes. Texture analysis of medical images. *Clinical Radiology*, 59(12):1061–1069, dec 2004.
- [14] C.-C. Chen, J.S. DaPonte, and M.D. Fox. Fractal feature analysis and classification in medical imaging. *IEEE Transactions on Medical Imaging*, 8(2):133–142, jun 1989.
- [15] G. Chetelat and J. Baron. Early diagnosis of alzheimer’s disease: contribution of structural neuroimaging. *NeuroImage*, 18(2):525–541, 2003.
- [16] V. Ciancia, S. Gilmore, D. Latella, M. Loreti, and M. Massink. Data verification for collective adaptive systems: Spatial model-checking of vehicle location data. In *Eighth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASOW*, pages 32–37. IEEE Computer Society, 2014.
- [17] V. Ciancia, G. Grilletti, D. Latella, M. Loreti, and M. Massink. An experimental spatio-temporal model checker. In *Software Engineering and Formal Methods - SEFM 2015 Collocated Workshops*, volume 9509 of *Lecture Notes in Computer Science*, pages 297–311. Springer, 2015.
- [18] V. Ciancia, D. Latella, M. Loreti, and M. Massink. Specifying and verifying properties of space. In *Theoretical Computer Science - 8th IFIP TC 1/WG 2.2 International Conference, TCS 2014, Rome, Italy, September 1-3, 2014. Proceedings*, volume 8705 of *Lecture Notes in Computer Science*, pages 222–235. Springer, 2014.
- [19] V. Ciancia, D. Latella, M. Loreti, and M. Massink. Model Checking Spatial Logics for Closure Spaces. *Logical Methods in Computer Science*, Volume 12, Issue 4, October 2016.
- [20] V. Ciancia, D. Latella, M. Massink, and R. Pakauskas. Exploring spatio-temporal properties of bike-sharing systems. In *2015 IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASO Workshops*, pages 74–79. IEEE Computer Society, 2015.
- [21] Vincenzo Ciancia, Diego Latella, Michele Loreti, and Mieke Massink. Spatial logic and spatial model checking for closure spaces. In Marco Bernardo, Rocco De Nicola, and Jane Hillston, editors, *Formal Methods for the Quantitative Evaluation of Collective Adaptive Systems - 16th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2016, Bertinoro, Italy, June 20-24, 2016, Advanced Lectures*, volume 9700 of *Lecture Notes in Computer Science*, pages 156–201. Springer, 2016.
- [22] Vincenzo Ciancia, Diego Latella, Mieke Massink, Rytis Paskauskas, and Andrea Vandin. A tool-chain for statistical spatio-temporal model checking of bike sharing systems. In Tiziana Margaria and Bernhard Steffen, editors, *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques - 7th International Symposium, ISoLA 2016, Imperial, Corfu, Greece, October 10-14, 2016, Proceedings, Part I*, volume 9952 of *Lecture Notes in Computer Science*, pages 657–673, 2016.
- [23] Krzysztof Chris Ciesielski, Xinjian Chen, Jayaram K. Udupa, and George J. Grevera. Linear time algorithms for exact distance transform. *Journal of Mathematical Imaging and Vision*, 39(3):193–209, nov 2010.
- [24] Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs, Workshop*, pages 52–71, London, UK, UK, 1982. Springer-Verlag.

- [25] F. Davnall, C. S. P. Yip, G. Ljungqvist, M. Selmi, F. Ng, B. Sanghera, B. Ganeshan, K. A. Miles, G. J. Cook, and V. Goh. Assessment of tumor heterogeneity: an emerging imaging tool for clinical practice? *Insights into Imaging*, 3(6):573–589, oct 2012.
- [26] R. De Nicola, J. Katoen, D. Latella, M. Loret, and M. Massink. Model checking mobile stochastic logic. *Theor. Comput. Sci.*, 382(1):42–70, 2007.
- [27] S. De Santis, M. Drakesmith, S. Bells, Y. Assaf, and D. K. Jones. Why diffusion tensor MRI does well only some of the time: Variance and covariance of white matter tissue microstructure attributes in the living human brain. *NeuroImage*, 89:35–44, apr 2014.
- [28] I. Despotović, B. Goossens, and W. Philips. MRI segmentation of the human brain: Challenges, methods, and applications. *Computational and Mathematical Methods in Medicine*, 2015:1–23, 2015.
- [29] Kunio Doi. Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. *Comput. Med. Imaging Graph.*, 31(4-5):198–211, 2007.
- [30] C. Dupont, N. Betrouni, N. Reyns, and M. Vermandel. On image segmentation methods applied to glioblastoma: State of art and new trends. *IRBM*, 37(3):131–143, jun 2016.
- [31] R. Fabbri, L. Da Fontoura Da Costa, J. C. Torelli, and O. M. Bruno. 2d euclidean distance transform algorithms: A comparative survey. *ACM Comput. Surv.*, 40(1):2:1–2:44, February 2008.
- [32] Ricardo Fabbri, Luciano Da F. Costa, Julio C. Torelli, and Odemir M. Bruno. 2d euclidean distance transform algorithms: A comparative survey. *ACM Comput. Surv.*, 40(1):2:1–2:44, February 2008.
- [33] Even Hovig Fyllingen, Anne Line Stensjøen, Erik Magnus Berntsen, Ole Solheim, and Ingerid Reinertsen. Glioblastoma segmentation: Comparison of three different software packages. *PLOS ONE*, 11(10):e0164891, oct 2016.
- [34] Antony Galton. The mereotopology of discrete space. In Christian Freksa and David M. Mark, editors, *Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science*, volume 1661 of *Lecture Notes in Computer Science*, pages 251–266. Springer Berlin Heidelberg, 1999.
- [35] E.A. Gol, E. Bartocci, and C. Belta. A formal methods approach to pattern synthesis in reaction diffusion systems. In *53rd IEEE Conference on Decision and Control*, pages 108–113, 2014.
- [36] N. Gordillo, E. Montseny, and E. Sobrevilla. State of the art survey on MRI brain tumor segmentation. *Magn. Reson. Imaging.*, 31(8):1426–1438, 2013.
- [37] G. J. Grevera. Distance transform algorithms and their implementation and evaluation. In *Deformable Models*, pages 33–60. Springer Science, 2007.
- [38] R. Grosu, S.A. Smolka, F. Corradini, A. Wasilewska, E. Entcheva, and E. Bartocci. Learning and detecting emergent behavior in networks of cardiac myocytes. *Commun. ACM*, 52(3):97–105, 2009.
- [39] I. Haghghi, A. Jones, Z. Kong, E. Bartocci, R. Grosu, and C. Belta. Spatel: A novel spatial-temporal logic and its applications to networked systems. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, HSCC ’15, pages 189–198, New York, NY, USA, 2015. ACM.

- [40] F. Han, H. Wang, G. Zhang, H. Han, B. Song, L. Li, W. Moore, H. Lu, H. Zhao, and Z. Liang. Texture feature analysis for computer-aided diagnosis on pulmonary nodules. *Journal of Digital Imaging*, 28(1):99–115, aug 2014.
- [41] Robert M Haralick, Karthikeyan Shanmugam, et al. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973.
- [42] T. Heinonen, T. Arola, A. Kalliokoski, P. Dastidar, M. Rossi, S. Soimakallio, J. Hyttinen, and H. Eskola. Computer aided diagnosis tool for the segmentation and texture analysis of medical images. In *IFMBE Proceedings*, pages 274–276. Springer Science, 2009.
- [43] A. Kassner and R. E. Thornhill. Texture analysis: A review of neurologic MR imaging applications. *Am. J. Neuroradiol.*, 31(5):809–816, 2010.
- [44] R. Kimmel, N. Kiryati, and A. M. Bruckstein. Sub-pixel distance maps and weighted distance transforms. *Journal of Mathematical Imaging and Vision*, 6(2):223–233, 1996.
- [45] R. Kontchakov, A. Kurucz, F. Wolter, and M. Zakharyashev. Spatial logic + temporal logic = ? In *Handbook of Spatial Logics*, pages 497–564. Springer, 2007.
- [46] O. Kutz, F. Wolter, H. Sturm, N. Suzuki, and M. Zakharyashev. Logics of metric spaces. *ACM Trans. Comput. Log.*, 4(2):260–294, 2003.
- [47] Doenja M. J. Lambregts, Geerard L. Beets, Monique Maas, Luís Curvo-Semedo, Alfons G. H. Kessels, Thomas Thywissen, and Regina G. H. Beets-Tan. Tumour ADC measurements in rectal cancer: effect of ROI methods on ADC values and interobserver variability. *European Radiology*, 21(12):2567–2574, aug 2011.
- [48] D. Latella, M. Loreti, and M. Massink. On-the-fly pctl fast mean-field approximated model-checking for self-organising coordination. *Science of Computer Programming*, 110:23 – 50, 2015.
- [49] L. Lemieux, G. Hagemann, K. Krakow, and F.G. Woermann. Fast, accurate, and reproducible automatic segmentation of the brain in t1-weighted volume mri data. *Magnetic Resonance in Medicine*, 42(1):127–135, 1999.
- [50] C. Li, J. G. Herndon, F. J. Novembre, and X. Zhang. A longitudinal magnetization transfer imaging evaluation of brain injury in a macaque model of NeuroAIDS. *AIDS Research and Human Retroviruses*, 31(3):335–341, mar 2015.
- [51] R. Lopes, A. Ayache, N. Makni, P. Puech, A. Villers, S. Mordon, and N. Betrouni. Prostate cancer characterization on MR images using fractal features. *Med. Phys.*, 38(1):83, 2011.
- [52] Anant Madabhushi and Jayaram K. Udupa. New methods of MR image intensity standardization via generalized scale. *Medical Physics*, 33(9):3426–3434, aug 2006.
- [53] C.R. Maurer, Rensheng Qi, and V. Raghavan. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):265–270, Feb 2003.
- [54] Kouros Meshgi and Shin Ishii. Expanding histogram of colors with gridding to improve tracking accuracy. In *MVA*, 2015.
- [55] L. Nenzi and L. Bortolussi. Specifying and monitoring properties of stochastic spatio-temporal systems in signal temporal logic. In *8th International Conference on Performance Evaluation Methodologies and Tools, VALUETOOLS 2014, Bratislava, Slovakia, December 9-11, 2014*. ICST, 2014.

- [56] L. Nenzi, L. Bortolussi, V. Ciancia, M. Loreti, and M. Massink. Qualitative and quantitative monitoring of spatio-temporal properties. In *Runtime Verification - 6th International Conference, RV 2015 Vienna, Austria, September 22-25, 2015. Proceedings*, volume 9333 of *Lecture Notes in Computer Science*, pages 21–37. Springer, 2015.
- [57] O. Prvu and D. Gilbert. A novel method to verify multilevel computational models of biological systems using multiscale spatio-temporal meta model checking. *PLoS ONE*, 11(5):1–43, 05 2016.
- [58] F. Rengier, A. Mehndiratta, H. von Tengg-Kobligk, C. M. Zechmann, R. Unterhinninghofen, H.-U. Kauczor, and F. L. Giesel. 3d printing based on imaging data: review of medical applications. *International Journal of Computer Assisted Radiology and Surgery*, 5(4):335–341, 2010.
- [59] D. Rodriguez Gutierrez, A. Awwad, L. Meijer, M. Manita, T. Jaspan, R. A. Dineen, R. G. Grundy, and D. P. Auer. Metrics and textural features of MRI diffusion to improve classification of pediatric posterior fossa tumors. *American Journal of Neuroradiology*, 35(5):1009–1015, dec 2013.
- [60] N. Sharma, A. Ray, S. Sharma, K.K. Shukla, S. Pradhan, and L. Aggarwal. Segmentation and classification of medical images using texture-primitive features: Application of BAM-type artificial neural network. *J Med Phys*, 33(3):119, 2008.
- [61] Rebecca L. Siegel, Kimberly D. Miller, and Ahmedin Jemal. Cancer statistics, 2016. *CA: A Cancer Journal for Clinicians*, 66(1):7–30, 2016.
- [62] V.R. Simi and Justin Joseph. Segmentation of glioblastoma multiforme from MR images – a comprehensive review. *The Egyptian Journal of Radiology and Nuclear Medicine*, 46(4):1105–1110, dec 2015.
- [63] G.N. Srinivasan and G. Shobha. Statistical texture analysis. In *Proceedings of World Accademy of Science, Engineering and Technology*, volume 36, pages 1264–1269, dec 2012.
- [64] Yiqun Sun, Tong Tong, Sanjun Cai, Rui Bi, Chao Xin, and Yajia Gu. Apparent diffusion coefficient (ADC) value: A potential imaging biomarker that reflects the biological features of rectal cancer. *PLoS ONE*, 9(10):e109371, oct 2014.
- [65] A. Sundstrom, E. Grabocka, D. Bar-Sagi, and B. Mishra. Histological image processing features induce a quantitative characterization of chronic tumor hypoxia. *PLoS ONE*, 11(4):1–30, 04 2016.
- [66] Mikkel Thorup. Undirected single-source shortest paths with positive integer weights in linear time. *J. ACM*, 46(3):362–394, May 1999.
- [67] B. M. Tijms, P. Series, D. J. Willshaw, and S. M. Lawrie. Similarity-based extraction of individual networks from gray matter MRI scans. *Cerebral Cortex*, 22(7):1530–1541, aug 2011.
- [68] A. T. Toosy. Diffusion tensor imaging detects corticospinal tract involvement at multiple levels in amyotrophic lateral sclerosis. *J. Neurol. Neurosurg. Psychiatry*, 74(9):1250–1257, 2003.
- [69] J. van Benthem and G. Bezhanishvili. Modal logics of space. In *Handbook of Spatial Logics*, pages 217–298. Springer, 2007.
- [70] Johan van Benthem and Guram Bezhanishvili. Modal logics of space. In *Handbook of Spatial Logics*, pages 217–298. Springer, 2007.
- [71] B.J. Woods, B. D. Clymer, T. Kurc, J. T. Heverhagen, R. Stevens, Orsdemir A., O. Bulan, and M. V. Knopp. Malignant-lesion segmentation using 4d co-occurrence texture analysis applied to dynamic contrast-enhanced magnetic resonance breast image data. *J. Magn. Reson. Imaging*, 25(3):495–501, 2007.

- [72] Ying Zhu, Geoffrey S. Young, Zhong Xue, Raymond Y. Huang, Hui You, Kian Setayesh, Hiroto Hatabu, Fei Cao, and Stephen T. Wong. Semi-automatic segmentation software for quantitative clinical brain glioblastoma evaluation. *Academic Radiology*, 19(8):977–985, aug 2012.