

Spatial Logics and Model Checking for Medical Imaging* - Extended Version -

Fabrizio Banci Buonamici
Gina Belmonte

Azienda Ospedaliera Universitaria Senese, Siena, Italy
{F.Banci,G.Belmonte}@ao-siena.toscana.it

Vincenzo Ciancia, Diego Latella ,Mieke Massink
CNR-ISTI, Pisa, Italy

{Vincenzo.Ciancia, Diego.Latella, Mieke.Massink}@cnr.it

Abstract

Recent research on spatial and spatio-temporal model checking provides novel image analysis methodologies, rooted in logical methods for topological spaces. Medical Imaging (MI) is a field where such methods show potential for ground-breaking innovation. Our starting point is SLCS, the *Spatial Logic for Closure Spaces*— Closure Spaces being a generalisation of topological spaces, covering also discrete space structures— and **topochecker**, a model-checker for SLCS (and extensions thereof). We introduce the logical language *ImgQL* (“Image Query Language”). *ImgQL* extends SLCS with logical operators describing *distance* and *region similarity*. The spatio-temporal model checker **topochecker** is correspondingly enhanced with state-of-the-art algorithms, borrowed from computational image processing, for efficient implementation of distance-based operators, namely *distance transforms*. Similarity between regions is defined by means of a *statistical similarity* operator, based on notions from *statistical texture analysis*. We illustrate our approach by means of two examples of analysis of Magnetic Resonance images: segmentation of *glioblastoma* and its *oedema*, and segmentation of *rectal carcinoma*.

keywords: Spatial logics; Closure spaces; Model checking; Medical Imaging; Segmentation; Magnetic Resonance Imaging; Distance Transform; Statistical Texture Analysis

*This work is partially supported by Azienda Ospedaliera Universitaria Senese.

1 Introduction

Computer Science plays a fundamental role in the field of medical image analysis. 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 [31]; *Image Segmentation*, tailored to identify areas that exhibit specific features or functions (such as organs or sub-structures) [40]; *Automatic contouring* of Organs at Risk or target volumes for radiotherapy applications [9]; 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] [74] (examples of indicators are the *mean diffusivity* and the *fractional anisotropy* obtained from Magnetic Resonance (MR) Diffusion-Weighted Images, or the *magnetisation transfer ratio* maps obtained from a Magnetisation Transfer acquisition [29] [53]).

Such kinds of analyses are strictly tied to the spatial features of images.

In this paper we focus on image segmentation, in particular to identify glioblastomas and rectal carcinomas. Glioblastomas are the most common malignant intracranial tumours whereas rectal carcinomas manifest themselves as particular tumours situated at the end of the large intestine. For the treatment of glioblastomas neuroimaging protocols are used before and after treatment to evaluate the effect of treatment strategies and to monitor the evolution of the disease. In clinical studies and routine treatment magnetic resonance images (MRI) are evaluated based mostly on qualitative criteria such as the presence of hyper-intense tissue appearing in the images [58]. The study and development of automatic and semi-automatic segmentation algorithms is aiming at overcoming the current time consuming practise of manual delineation of such tumours and at providing an accurate, reliable and reproducible method of segmentation of the tumour area and related tissues [32].

Segmentation of medical images, and brain segmentation in particular, is nowadays an important topic on its own in many applications in neuroimaging; several automatic and semi-automatic methods have been proposed [52, 30] constituting a very active research area (see for example [32, 35, 68, 78]). One of the technical challenges of the development of automated (brain) tumour segmentation is that lesion areas are only defined through changes in the intensity (luminosity) in the (black & white) images that are *relative* to surrounding normal tissue. Even manual segmentation by experts shows significant variations when intensity gradients between adjacent tissue structures are smooth or partially obscured [58]. Furthermore, there is a considerable variation across images from different patients and images obtained with different MRI scanners.

In this paper we propose a novel approach to image segmentation, namely an interactive, logic based method, supported by *spatial model checking*, tailored to loosely identify a region of interest in MRI on which to focus the analysis of glioblastoma or other types of tumours. This approach is particularly suitable to exploit the *relative* spatial relations between tissues of interest mentioned earlier.

Spatial and Spatio-temporal model checking are the subject of a recent trend in Computer Science (see for instance [28, 42, 18, 19, 22, 61, 43]) that uses specifications written in logical languages describing *space* – called *spatial logics* – to automatically identify patterns and structures of interest in a variety of domains, ranging from signals [61] and images [19, 43] to Collective Adaptive Systems [20, 16, 23].

The research presented in the present paper 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 topology as a mathematical framework for the interpretation of modal logic (see [75] for a thorough introduction). In this context, formulas are interpreted as sets of points of a topological space, and in particular the modal operator \diamond is usually interpreted as the (logical representation of the) topological *closure* operator. A standard reference is the *Handbook of Spatial Logics* [2]. Therein, several spatial logics are described, with applications far beyond topological spaces; such logics treat not only aspects of morphology, geometry and distance, but also advanced topics such as dynamic systems, and discrete structures, that are particularly difficult to deal with from a topological perspective (see, for example [36]). In recent work [18, 19], Ciancia et al. pushed such theoretical development further to encompass directed graphs, resulting in the definition of the *Spatial Logic for Closure Spaces* (SLCS), and a related model checking algorithm. Subsequently, in [17], a spatio-temporal logic, combining *Computation Tree Logic* and the newly defined spatial operators, was introduced; the (extended) model checking algorithm has been implemented in the prototype *spatio-temporal model checker topochecker*¹.

The broader scope of our research interest in the context of medical imaging 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 logic* formulas specifying relevant features, such as texture or similarity, bound together by spatial constraints, for example, proximity, boundary properties, distance, and so on, that increase the significance and signal-to-noise ratio of the obtained results. This is possible by considering such images as instances of (quasi discrete) closure spaces. The tools and methods we introduce can be used both for two-dimensional (2D) and three-dimensional (3D) MI; we remark that modern MRI machines can usually provide 3D data for analysis; however, in standard practice, 3D information is often discarded in favour of 2D (slice by slice) analysis, due to the lack of well-established methods for 3D analysis. Using 3D information may lead to improved accuracy and it is therefore of high interest, in current research, to identify techniques for this purpose.

¹Topochecker: a topological model checker, see <http://topochecker.isti.cnr.it>, <https://github.com/vincenzom1/topochecker>

Original contributions:

This paper details and extends the ideas outlined in [8], providing several further original contributions:

- extension of the spatial logic SLCS to ImgQL, introducing distance-based operators and showing their formal relation to the other spatial logic operators of SLCS. A novel approach to model checking of distance-based operators is provided 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 makes such procedure suitable for the analysis of higher resolution or 3D images;
- introduction of a novel logical connective aimed at estimating similarity between regions. This operator is based on *statistical texture analysis* and is able to classify points of the space based on the similarity between the area where they are localised, and a target region, expressed in logical terms. The connective is specific for medical image analysis. Its embedding shows how such connectives can be integrated into the spatial logic. This provides an example of how other specialised existing algorithms could be introduced and exploited within the spatial logic model checking framework;
- enhancement of the results in the glioblastoma case study first introduced in [8], providing the relevant technical details on the logical specification;
- presentation of a further case study — namely, segmentation of *rectum carcinoma* — showing that the method can also be applied to the segmentation of other types of tumours that are situated in other parts of the body;
- development of efficient model checking algorithms, that are competitive in computational efficiency w.r.t. state-of-the-art (semi-)automatic segmentation approaches. As an additional benefit, logical specifications are transparent, reproducible, accurate, human-readable, and applicable to both 2D and 3D images.

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 work in [8] constituted a first *proof-of-concept* study. In that study **topochecker** was used for the declarative specification of regions in medical images. The model checker was used to automatically and efficiently identify and colour *glioblastoma* and the surrounding *oedema* in MRI scans, on the basis of a declarative definition of the two *regions of interest*, given in terms of their visual appearance. The latter is defined by image features such as proximity,

interconnection, and texture similarity. The input to the model checker consists of a precise, declarative, unambiguous logical specification, that besides being fairly close to the level of abstraction of an expert description of the process, is also remarkably concise, human readable, robust and reproducible.

Related work:

The idea of using model checking, and in particular spatial or spatio-temporal model checking, for the analysis of medical images is relatively recent and there are only a few articles exploring this field so far. In particular, [71] uses spatio-temporal model checking techniques inspired by [42] – pursuing machine learning of the logical structure of image features – for the detection of tumours. In contrast, our approach is more focused on human-intelligible logical descriptions that provide reproducible results. Other interesting work is that in [62] where spatio-temporal meta model checking is used for the analysis of biological processes, with an interesting focus on *multi-scale* aspects.

Among the fully automated approaches that recently are gaining interest are those based on machine learning and deep learning (see for example [4] for a recent review). Although manual segmentation is still the standard for *in vivo* images, this method is expensive and time-consuming, difficult to reproduce and possibly inaccurate due to human error. Machine learning and deep learning approaches have shown promising results in pattern recognition in areas where large, reliable datasets are available and are currently being developed for application in MRI based brain segmentation with the aim to obtain reliable automatic segmentation methods. Deep learning is based on the use of artificial neural networks, consisting of several layers, that can extract a hierarchy of features from raw input data. These methods depend heavily on the availability of large training datasets and the generation of manual ground truth labels, i.e. data sets in which segments of interest are indicated by experts manually in a standard way. This is a complicated task not only because it is very laborious, but also because of the relatively high intra-expert and inter-expert variability of $20\pm 15\%$ and $28\pm 12\%$, respectively, for manual segmentations of brain tumour images [57]. Interactive approaches based on spatial model checking may therefore also be of help to improve the generation of manual ground truth labels in a more efficient, transparent and reproducible way.

Outline:

A technical introduction to spatial logics and distance-based operators is provided in Section 2. Syntax and semantics of the fragment of SLCS we will use in this paper are recalled, as well as the main notions of spatial model checking for the fragment. The definition of a distance operator for ImgQL is presented as well. In Section 3 the logic framework we propose for statistical texture analysis is 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 Logics for Closure Spaces with Distance

In this section, we discuss the background knowledge that we use in the technical developments of the paper and we extend it with notions of *distance*. In particular, we briefly introduce the notion of closure spaces, the fragment of SLCS [18, 19] we use in this paper, the related model checking algorithm, and `topochecker`. We detail the use of so-called *distance operators* in this research line and we extend the logic fragment with a distance operator parametric on the specific notion of distance; we also give an account of the extension of the model checking algorithm necessary for dealing with the distance operator, based on the notion of *distance transform* and its implementation in `topochecker` for two specific notions of distance, namely the *Euclidean* and the *shortest path* distances.

In the sequel we will often make explicit reference to 2D images and their pixels; here we point out that this is done only for the sake of simplicity and that all notions, notations, definitions and results equally apply to 3D images and their voxels (i.e. *volumetric picture elements*, the 3D counterpart of pixels).

2.1 Closure Spaces, Spatial logics and Model Checking

In *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 [2]. 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 *arbitrary* graphs, including e.g. those with a non-transitive / non-symmetric edge relation, 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 [36]) is the key to generalise these results. In this paper we use a fragment of SLCS comprising 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 [3, 76], 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) [26], has been explored in [17]. Some related case studies have been analysed in [23, 21] where the logic caters for spatio-temporal reasoning and model checking. In the present paper, we focus on spatial properties; therefore we restrict our attention to spatial aspects of our framework.

2.1.1 A fragment of SLCS

SLCS is a logic for *space*, where the latter is modelled by means of *closure spaces*. Before introducing the fragment of SLCS we use in the present paper, we recall some basic notions of closure spaces [36, 37].

Definition 1 A closure space is a pair (X, \mathcal{C}) where X is a non-empty set (of points) and $\mathcal{C} : 2^X \rightarrow 2^X$ is a function satisfying the following three axioms:

1. $\mathcal{C}(\emptyset) = \emptyset$;
2. $Y \subseteq \mathcal{C}(Y)$ for all $Y \subseteq X$;
3. $\mathcal{C}(Y_1 \cup Y_2) = \mathcal{C}(Y_1) \cup \mathcal{C}(Y_2)$ for all $Y_1, Y_2 \subseteq X$. •

According to the well known Kuratowski definition, adding the *idempotence axiom* $\mathcal{C}(\mathcal{C}(Y)) = \mathcal{C}(Y)$ for all $Y \subseteq X$ in Definition 1 makes it a definition of topological spaces [37]. Consequently, the latter are a subclass of closure spaces.

Given any relation $R \subseteq X \times X$, function $\mathcal{C}_R : 2^X \rightarrow 2^X$ with $\mathcal{C}_R(Y) = Y \cup \{x \mid \exists y \in Y. y R x\}$ satisfies the axioms of Definition 1 thus making (X, \mathcal{C}_R) a closure space. The class of closure spaces generated by binary relations on the set of points represent a very interesting subclass of closure spaces, known as *quasi-discrete* closure spaces. Quasi-discrete closure spaces include discrete structures like graphs, where each graph (X, R) with set of nodes X and set of the edges R is in one to one correspondence with closure space (X, \mathcal{C}_R) . Clearly, *finite* closure spaces are quasi-discrete closure spaces.

The following definition is instrumental for the definition of *paths* over quasi-discrete closure spaces.

Definition 2 A continuous function $f : (X_1, \mathcal{C}_1) \rightarrow (X_2, \mathcal{C}_2)$ is a function $f : X_1 \rightarrow X_2$ such that, for all $Y \subseteq X_1$, we have $f(\mathcal{C}_1(Y)) \subseteq \mathcal{C}_2(f(Y))$. •

In the definition below $(\mathbb{N}, \mathcal{C}_{Succ})$ is the closure space of natural numbers with the *successor* relation: $(n, m) \in Succ \Leftrightarrow m = n + 1$.

Definition 3 A path π in (X, \mathcal{C}_R) is a continuous function $\pi : (\mathbb{N}, \mathcal{C}_{Succ}) \rightarrow (X, \mathcal{C}_R)$. •

In the sequel we will let π, π', π_1, π_2 denote paths; the elements of \mathbb{N} will be called *indexes* in the context of paths.

A quasi-discrete closure space (X, \mathcal{C}_R) , can be used as the basis for a mathematical model of a 2D digital image; X represents the finite set of *pixels* and R is the reflexive and symmetric *adjacency* relation between pixels [38]. We note in passing that several different adjacency relations can be used. For instance in the *orthogonal* adjacency relation (sometimes called von Neumann adjacency) only pixels which share an edge count as adjacent, so that each pixel is adjacent to (itself and) four other pixels; on the other hand, in the *orthodiagonal* adjacency relation pixels are adjacent as long as they share at least either an edge or a corner, so that each pixel is adjacent to (itself and) eight other pixels.

$\Phi ::=$	p	[ATOMIC PREDICATE]
	$\neg \Phi$	[NEGATION]
	$\Phi_1 \wedge \Phi_2$	[CONJUNCTION]
	$\mathcal{N} \Phi$	[NEAR]
	$\Phi_1 \mathcal{S} \Phi_2$	[SURROUNDED]

Figure 1: Syntax of the fragment of SLCS.

Pixels are usually associated with specific attributes, such as colours and/or colour-intensity. We model this by assuming that a set A of point *attribute names* is given and by enriching (X, \mathcal{C}_R) with an *attribute evaluation* function $\mathcal{A} : X \times A \rightarrow V$ from points and their attributes to some value set V such that $\mathcal{A}(x, a) \in V$ is the *value* of attribute a of point x .

For given set P of *atomic predicates* p , the syntax of the fragment of SLCS we use in this paper is given in Figure 1.

Informally, it is assumed that space is modelled by a set of points; each atomic predicate $p \in P$ models a specific *feature* of *points* and is thus associated with the set of points which have this feature. A point x satisfies $\mathcal{N} \Phi$ if a point satisfying Φ can be reached from x in at most one (closure) step, i.e. if x is *near* (or *close*) to a point satisfying Φ . A point x satisfies $\Phi_1 \mathcal{S} \Phi_2$ if it satisfies Φ_1 and in any path π rooted in x (i.e. such that $\pi(0) = x$) and passing through a point $\pi(\ell)$ *not* satisfying Φ_1 , there is a point $\pi(j)$ before or at ℓ (i.e. $0 < j \leq \ell$) that satisfies Φ_2 . In other words, x belongs to an area satisfying Φ_1 and one cannot *escape* from such an area without hitting a point Φ_2 , i.e. x is *surrounded* by Φ_2 . Finally, the fragment includes logical negation (\neg) and conjunction (\wedge).

The above description is formalised by the definition of *model* and *satisfaction relation*:

Definition 4 A closure model is a tuple $((X, \mathcal{C}), \mathcal{A}, \mathcal{V})$ consisting of a closure space (X, \mathcal{C}) , a valuation $\mathcal{A} : X \times A \rightarrow V$, assigning to each point and attribute the value of the attribute at that point, and a valuation $\mathcal{V} : P \rightarrow 2^X$ assigning to each atomic predicate the set of points where it holds. •

In the sequel, we assume that an atomic predicate p can be bound to an assertion α , the latter stating a property of attributes, and we use the syntax $p := \alpha$ for atomic predicate definitions, to this purpose. Assertions are standard Boolean expressions, e.g. comparisons of the form $a \geq c$, for $c \in V$, and compositions thereof; we refrain from specifying the actual syntax of assertions, and we assume valuation \mathcal{A} be extended in the obvious way in order to evaluate assertions, e.g. $\mathcal{A}(x, a \geq c) = \mathcal{A}(x, a) \geq c$.

Definition 5 Satisfaction $\mathcal{M}, x \models \Phi$ at point $x \in X$ in model $\mathcal{M} = ((X, \mathcal{C}), \mathcal{A}, \mathcal{V})$ is defined by induction on the structure of formulas, as in Figure 2. •

In Figure 3 a simple finite closure model is shown for which the orthogonal adjacency relation is assumed. All the points satisfying atomic predicate p are

$$\begin{array}{llll}
\mathcal{M}, x \models p \in P & \Leftrightarrow & x \in \mathcal{V}(p) & \\
\mathcal{M}, x \models \neg \Phi & \Leftrightarrow & \mathcal{M}, x \models \Phi \text{ does not hold} & \\
\mathcal{M}, x \models \Phi_1 \wedge \Phi_2 & \Leftrightarrow & \mathcal{M}, x \models \Phi_1 \text{ and } \mathcal{M}, x \models \Phi_2 & \\
\mathcal{M}, x \models \mathcal{N}\Phi & \Leftrightarrow & x \in \mathcal{C}(\{y \mid \mathcal{M}, y \models \Phi\}) & \\
\mathcal{M}, x \models \Phi_1 \mathcal{S} \Phi_2 & \Leftrightarrow & \mathcal{M}, x \models \Phi_1 \text{ and} & \\
& & \text{for all paths } \pi \text{ and indexes } \ell: & \\
& & \pi(0) = x \text{ and } \mathcal{M}, \pi(\ell) \models \neg \Phi_1 & \\
& & \text{implies} & \\
& & \text{there exists index } j \text{ such that:} & \\
& & 0 < j \leq \ell \text{ and } \mathcal{M}, \pi(j) \models \Phi_2 &
\end{array}$$

Figure 2: Semantics of the fragment of SLCS; whenever $p := \alpha$ is a definition for p , we assume $x \in \mathcal{V}(p)$ if and only if $\mathcal{A}(x, \alpha)$ yields the truth-value *true*.

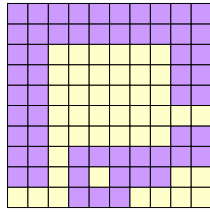


Figure 3: An example model; the points satisfying atomic predicate p are shown in violet, those satisfying q are shown in yellow.

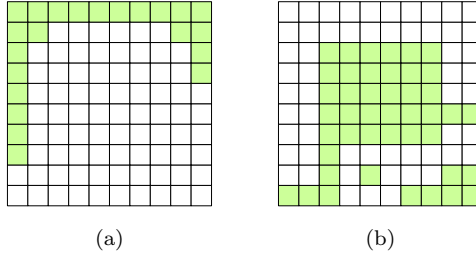


Figure 4: The points in Figure 3 satisfying $\neg\mathcal{N}q$ (a) and those satisfying $q\mathcal{S}p$ (b) are shown in green.

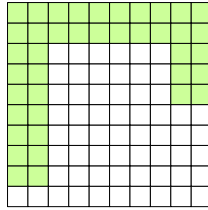


Figure 5: The points in Figure 3 satisfying $p\mathcal{T}\neg(\mathcal{N}q)$ are shown in green.

shown in violet whereas those satisfying q are shown in yellow (no point satisfies $p \wedge q$ in this example). Figure 4a shows in green the points that satisfy $\neg\mathcal{N}q$, while Figure 4b shows in green the points satisfying $q\mathcal{S}p$ (i.e., all q -points that are surrounded by p -points; note that, in the example, these are exactly *all* q -points).

A number of useful *derived operators* are defined in Figure 6. A few words of explanation are worth for the \mathcal{T} operator, while we refer the reader to [19] for a general discussion on SLCS derived operators. A point satisfies $\Phi_1 \mathcal{T} \Phi_2$ if and only if it lays in an area $Y_1 \subseteq X$ the points of which satisfy Φ_1 and Y_1 “touches” a non-empty area Y_2 , the points of which satisfy Φ_2 ; for this reason, sometimes we call the FROM-TO operator “touches”. With reference to Figure 3, Figure 5 shows in green the points satisfying $p\mathcal{T}\neg(\mathcal{N}q)$. Another pattern, that may be used for filtering noise in images, is formula $\mathcal{N}\mathcal{I}\Phi$. The effect of such a formula is to capture the *regular* region [49] included in the set of points satisfying Φ ; point x satisfies $\mathcal{N}\mathcal{I}\Phi$ if and only if it is adjacent to at least one point y satisfying Φ which, in turn, is not adjacent to points satisfying $\neg\Phi$. The effect of such a filter is to eliminate small regions, e.g. those consisting of a single point, when these are considered noise or artefacts.

\perp	$\triangleq p \wedge \neg p$	[FALSE]
\top	$\triangleq \neg \perp$	[TRUE]
$\Phi_1 \vee \Phi_2$	$\triangleq \neg(\neg\Phi_1 \wedge \neg\Phi_2)$	[DISJUNCTION]
$\mathcal{I}\Phi$	$\triangleq \neg(\mathcal{N}\neg\Phi)$	[INTERIOR]
$\Phi_1 \mathcal{R} \Phi_2$	$\triangleq \neg(\neg\Phi_2 \mathcal{S} \neg\Phi_1)$	[REACHABILITY]
$\Phi_1 \mathcal{T} \Phi_2$	$\triangleq \Phi_1 \wedge ((\Phi_1 \vee \Phi_2) \mathcal{R} \Phi_2)$	[FROM-TO]
$\mathcal{E}\Phi$	$\triangleq \Phi \mathcal{S} \perp$	[EVERYWHERE]
$\mathcal{F}\Phi$	$\triangleq \neg(\mathcal{E}\neg\Phi)$	[SOMEWHERE]
$\Phi_1 \mathcal{S} \Phi_2$	$\triangleq (\Phi_1 \mathcal{S} \Phi_2) \wedge \neg \mathcal{E} \Phi_1$	[STRONG SURROUNDED]

Figure 6: Some derived operators.

2.1.2 Model checking SLCS

In this section we will briefly recall model checking of SLCS [18, 19] over finite models. Note that, in the context of the present paper, we are concerned with so-called *global* model checking, i.e. a procedure that, given a finite model and a logic formula, returns the set of *all* points in the model satisfying the formula [25]. We will focus on the surrounded operator only and we will describe the related section of the model checking algorithm by means of an example. Model checking algorithms for the other operators of the fragment is a matter of standard routine. We will also provide a brief description of `topochecker`.

Given a finite closure model $\mathcal{M} = ((X, \mathcal{C}_R), \mathcal{A}, \mathcal{V})$ and a formula Φ , the model checking algorithm returns all those points $x \in X$ such that $\mathcal{M}, x \models \Phi$. For a formula $\Phi_1 \mathcal{S} \Phi_2$ the algorithm, roughly speaking, first identifies areas of *bad* points, that is points that can reach a point satisfying $\neg\Phi_1$ without passing by a point satisfying Φ_2 ; then returns the points that satisfy Φ_1 and that are not bad. A sketch of the fragment of the model checking algorithm related to $\Phi_1 \mathcal{S} \Phi_2$ is given in Figure 7.

Below, we give a brief description of how the algorithm works, using the graphs shown in Figure 8. Let us consider the model of Figure 8a as input model, where points are represented by coloured squares and the adjacency relation is the orthogonal one. In this example we assume that the set of atomic predicates is the set $\{pink, yellow, white\}$ — represented in the figure in the obvious way—and that $\mathcal{V}(p) \cap \mathcal{V}(p') = \emptyset$ whenever $p \neq p'$. Suppose the input formula is $yellow \mathcal{S} pink$. The result of the assignment $\text{Bad} := \text{TempBad}$ of the first iteration of the **repeat** is shown in Figure 8b, where all nodes that satisfy $\neg(yellow \vee pink)$ are marked blue. Note that this blue-colouring is *not* part of the model; we use it at a “meta-level” and only for describing the behaviour of the algorithm; the same will apply to points marked in cyan in the sequel. In Figure 8c the (only two) *yellow* points in the closure of the points indicated in blue, are shown in cyan²; these are the points to be selected for being added to TempBad in the first iteration of the **repeat**. The new value of TempBad , resulting

²In the caption, such a closure is abbreviated by $\mathcal{C}_R(\text{blue})$ for space reasons.

Input:

A model $\mathcal{M} = ((X, \mathcal{C}_R), \mathcal{A}, \mathcal{V})$
 and a formula $\Phi_1 \mathcal{S} \Phi_2$;

Output:

The set of points in X satisfying $\Phi_1 \mathcal{S} \Phi_2$;

Step 1:

TempBad := $\{x \in X | \mathcal{M}, x \models \neg(\Phi_1 \vee \Phi_2)\}$;

Step 2:

repeat

Bad := TempBad;

TempBad := $\text{Bad} \cup (\{x \in X | \mathcal{M}, x \models \Phi_1\} \cap \mathcal{C}_R(\text{Bad}))$;

until TempBad = Bad;

Step 3:

return $\{x \in X | \mathcal{M}, x \models \Phi_1\} \setminus \text{Bad}$.

Figure 7: Sketch of the model checking algorithm for $\Phi_1 \mathcal{S} \Phi_2$.

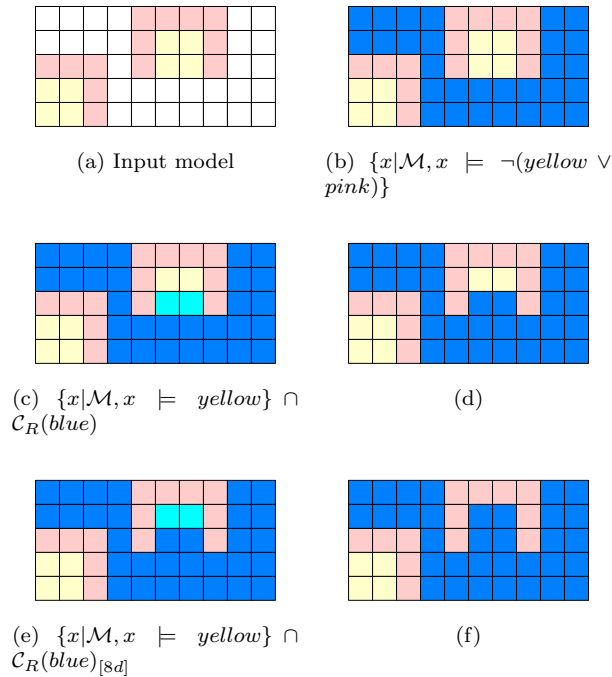


Figure 8: Model checking $\text{yellow} \mathcal{S} \text{pink}$.

p	\neg	\wedge	\mathcal{N}	\mathcal{S}	\perp	\top	\vee	\mathcal{I}	\mathcal{R}	\mathcal{T}
[p]	!	&	N	S	FF	TT		I	R	T

Table 1: `topochecker` syntax.

from the assignment, consists of all blue points of Figure 8d. The body of the `repeat` is executed now with the new value of `TempBad`. In Figure 8e the (again only two) *yellow* points in the closure of the set of points in blue are shown in cyan. Note that such a closure refers to the model of Figure 8d; this is abbreviated in Figure 8e as $\mathcal{C}_R(\text{blue})_{[8d]}$. The new value of `TempBad`, resulting from the assignment, consists of all points indicated in blue in Figure 8f. The body of the `repeat` is executed now for the third time and this results in no change in the value of `TempBad`: the fixed point is reached, the `repeat` block is exited and the points satisfying *yellow Spink* are the four *yellow* points in the bottom-left corner of Figure 8f. In [19] it has been shown that, for any finite closure model $\mathcal{M} = ((X, \mathcal{C}_R), \mathcal{A}, \mathcal{V})$ and SLCS formula Φ of size k , the model checking procedure terminates in $\mathcal{O}(k \cdot (|X| + |R|))$ steps³. We refer the reader to the above mentioned paper for further details on SLCS and its model checking algorithm.

The tool `topochecker` is a global model checker, capable of analysing models specified as weighted graphs, RGB images, or grayscale medical images. In the case of medical images, which is of interest in this work, the tool takes as input a file describing the spatial model to be analysed, the formulas to be checked, and a colour associated with each formula. The spatial model is described in the form of a set of images, whose intensity values are associated by the user to differently named attributes for subsequent usage in formulas. The output of the tool consists of a region of interest (ROI) for each formula to be checked, that is, an image where the specific region where such formula holds is coloured according to the user-specified colour.

Table 1 shows the correspondence between SLCS operators (top) and their syntax in `topochecker` (bottom). The syntax for assertions (extending the syntax for atomic predicates) is `[a \bowtie c]` where `a` is an attribute name, \bowtie is a comparison operator (one of `=`, `<`, `>`, `<=`, `>=`) and `c` is a (floating point) constant. In `topochecker`, (unnamed) assertions can be used in place of atomic predicates; moreover names can be given to complex formulas, by means of *formula definitions* as in the example shown below:

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

formula `N (N [FLAIR>1.7])` is given the name `adipose` that can be used in other formulas; the formula exemplifies using assertion `FLAIR>1.7` in place of an atomic predicate.

³The size of a formula is given by the number of operators in the formula: $size(p) = 1$; $size(\neg\Phi) = size(\mathcal{N}\Phi) = 1 + size(\Phi)$; $size(\Phi_1 \wedge \Phi_2) = size(\Phi_1 \mathcal{S} \Phi_2) = 1 + size(\Phi_1) + size(\Phi_2)$.

In general, names introduced by formula definitions may also have parameters such as

Let $f(f_1, f_2, \dots, f_n) = F$

where F is a formula that can use names f_1, f_2, \dots, f_n , that are instantiated to formulas when f is invoked.

The spatial model checking algorithm is entirely run in central memory, aiming at fast *interactive* usage. The algorithm proceeds by induction on the structure of formulas, and uses memoization to cache the intermediate result on each sub-formula, so that when the same sub-formula is used more than once, results are reused. The cache is also stored on-disk, leveraging incremental design of complex formulas across different model checking sessions. The tool is implemented in the functional programming language OCaml⁴, which provides a good balance between declarative features and computational efficiency. The main loop of the algorithm has been carefully written to avoid memory allocation and garbage collection in most cases, via the use of the “bigarray” data type, providing direct access to memory locations and memory-mapping of large files stored on the hard drive (such as medical images). All the arrays needed for the computation are statically allocated prior to model checking execution. Such optimisations result in competitive execution times, such as those described in Section 4.3. Indeed, since no memory swap takes place, one should take into account the memory requirements of each analysis, that could render in-memory execution unfeasible; however, the size of a typical medical image is in the order of some megabytes, which is orders of magnitude smaller than the available memory on modern computers (some gigabytes); in our experiments, the algorithm never ran out of memory.

2.2 Incorporating Distance

Models of space as well as spatial logics can be extended with notions of *distance* (see e.g. [49, 50, 60, 61]). Distances are very often expressed using the non-negative real numbers $\mathbb{R}_{\geq 0}$, like the Euclidean distance on continuous space.

For quasi-discrete closure spaces, especially when used as a representation of finite graphs, it is natural to consider *shortest path distance*, where a path between two nodes is a sequence of consecutive edges connecting the first node to the second, and its length is given by the sum of the lengths of such edges. The length of an edge is often taken to be 1; however, other notions of distance can be more appropriate. For example, *sampling* a multi-dimensional Euclidean space is often done using a *regular grid*, that is, a graph in which the nodes 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. the orthogonal or orthodiagonal adjacency relations discussed before, but any choice may be reasonable, depending on the application context). Such graphs can then be *weighted* by associating to each edge

⁴See <http://www.ocaml.org>.

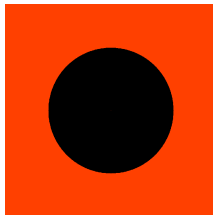
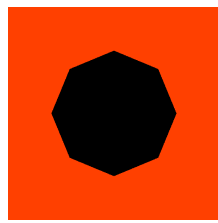


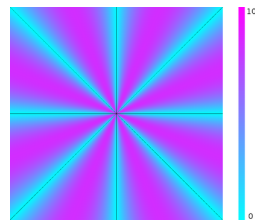
Figure 9: Threshold imposed on Euclidean distance from a point in the centre of image.

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, which is the case of interest for MI. In two-dimensional imaging, pixels—with an application-dependent choice of adjacency—form an Euclidean graph, and Euclidean distance is the reference distance between (the centres of) two pixels.

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* pairs of nodes are linked by an edge (labelled with the Euclidean distance between the end-points of the edge). Therefore, in this context, Euclidean distance is considered *error-free*, and Chamfer distance is considered an approximation of the former. The chosen adjacency relation determines the precision-efficiency trade-off of the computed distance: the more pixels are considered adjacent, the more precise is the approximation, at the expenses of generating graphs with larger out-degrees. This is illustrated in Figures 9 and 10. In the first figure we show a two-dimensional, rectangular image where all and only points at a Euclidean distance larger than a given threshold k from the centre of the figure are coloured in red. In Figure 10a the points in red are those at a Chamfer distance larger than k from the centre; in particular, orthodiagonal adjacency has been used (each pixel has 8 other adjacent pixels). Figure 10b shows the percentage of error for each pixel with respect to the Euclidean distance, in a scale from 0 to 10%. Finally, in Figure 10c we use Chamfer distance, the same threshold k and an adjacency relation where each pixel has 24 other adjacent pixels (i.e. the pixels that are adjacent to any pixel form a 5×5 square centred in the pixel). Figure 10d shows the percentage of error w.r.t. the Euclidean distance, in a scale from 0 to 2%. The percentage error $\delta(x)$ for Chamfer distance d_C is defined for each pixel x as $\delta(x) = \frac{|d_E(y,x) - d_C(y,x)|}{d_E(y,x)}$, where y is the central point of the image and d_E denotes the Euclidean distance.



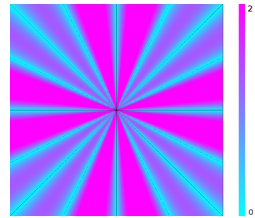
(a) Chamfer distance with 8 adjacent pixels per node (3×3 square centred on node).



(b) Percentage error. Scale: 0-10.



(c) Chamfer distance with 24 adjacent pixels per node (5×5 square centred on node).



(d) Percentage error. Scale: 0-2.

Figure 10: Percentage error of Chamfer distance.

2.2.1 SLCS with distance operators

In this section we extend the fragment of SLCS presented in Section 2.1.1 with logic *distance* operators. We first introduce the notion of *distance closure spaces* and, to that purpose we recall the well-known notion of metric space:

Definition 6 A metric space is a pair (X, d) where X is a non-empty set (of points) and $d : X \times X \rightarrow \mathbb{R}_{\geq 0}$ is function that satisfies the following axioms, for all $x, y, z \in X$:

1. $d(x, y) = 0$ iff $x = y$ [IDENTITY OF INDISCERNIBLE];
2. $d(x, y) = d(y, x)$ [SYMMETRY];
3. $d(x, z) \leq d(x, y) + d(y, z)$ [TRIANGLE INEQUALITY].

Whenever X is a closure space (X, \mathcal{C}) , $((X, \mathcal{C}), d)$ is called a metric closure space •

Metric functions are easily extended to sets as follows:

Definition 7 Given metric space (X, d) , $x \in X$ and $Y, Z \subseteq X$ we let

1. $d(x, Y) = \inf\{d(x, y) | y \in Y\}$
2. $d(Y, Z) = \inf\{d(y, z) | y \in Y \text{ and } z \in Z\}$

Note that if $Y \neq \emptyset$ is finite, then $\inf\{d(x, y) | y \in Y\} = \min\{d(x, y) | y \in Y\}$; we let $d(x, \emptyset) = \infty$ by definition. •

In the case of quasi-discrete closure spaces, symmetry may turn out to be too much restrictive. This is for instance the case when the relation R underlying the closure operator \mathcal{C}_R is not symmetric. Similarly, the triangle inequality is not fitting well when more qualitative distance “measures” are used, for instance when the codomain of d is composed of only three values, representing *short*, *medium*, and *large* distance respectively. For all these reasons, for quasi-discrete closure spaces we often use the less restrictive notion of *distance space*, where only Axiom 1 of Definition 6 above is required.

Definition 8 A distance closure space is a tuple $((X, \mathcal{C}), d)$ where (X, \mathcal{C}) is a closure space and $d : X \times X \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$ is function satisfying $d(x, y) = 0$ iff $x = y$.

A quasi-discrete distance closure space is a distance closure space $((X, \mathcal{C}_R), d)$ where (X, \mathcal{C}_R) is a quasi-discrete closure space. •

Distance operators can be added to spatial logics in various ways (see [49] for an introduction). For the purposes of the present paper, we extend SLCS with the operator \mathcal{D}^I , where I is an interval of $\mathbb{R}_{\geq 0}$. The intended meaning is that a point x of a distance closure model satisfies $\mathcal{D}^I \Phi$ if its distance from the set of points satisfying Φ falls in interval I . Below we provide the necessary formal definitions.

Definition 9 A distance closure model is a tuple $((X, \mathcal{C}), d, \mathcal{A}, \mathcal{V})$ consisting of a distance closure space $((X, \mathcal{C}), d)$, a valuation $\mathcal{A} : X \times A \rightarrow V$, assigning to each point and attribute the value of the attribute of the point and a valuation $\mathcal{V} : P \rightarrow 2^X$ assigning to each atomic predicate the set of points where it holds. A quasi-discrete distance closure model $((X, \mathcal{C}_R), d, \mathcal{A}, \mathcal{V})$ is a distance closure model where $((X, \mathcal{C}_R), d)$ is a quasi-discrete distance closure space. •

As the definition of d might require the elements of R to be weighted—as in the case of Euclidean graphs—quasi-discrete distance closure models are often enriched with a R -weighting function $\mathcal{W} : R \rightarrow \mathbb{R}$ assigning the weight $\mathcal{W}(x, y)$ to each $(x, y) \in R$.

The satisfaction relation of our fragment of SLCS is extended as expected:

Definition 10 Satisfaction $\mathcal{M}, x \models \Phi$ at point $x \in X$ in distance closure model $\mathcal{M} = ((X, \mathcal{C}), d, \mathcal{A}, \mathcal{V})$ is defined by induction on the structure of formulas, by adding the equation below to those in Figure 2:

$$\mathcal{M}, x \models \mathcal{D}^I \Phi \Leftrightarrow d(x, \{y \mid \mathcal{M}, y \models \Phi\}) \in I$$

Note that the definition of the SLCS distance operator is *parametric* on the specific distance used. The particular meaning of the distance operator is fully characterised by the specific distance d of the underlying distance closure model. In this paper, we will use the Euclidean distance d_E and the Chamfer distance d_C .

We close this section with the definition of an additional set of derived operators shown in Figure 11.

$$\begin{aligned} \mathcal{D}^{<c} \Phi &\triangleq \mathcal{D}^{[0, c)} \Phi \\ \mathcal{D}^{\leq c} \Phi &\triangleq \mathcal{D}^{[0, c]} \Phi \\ \mathcal{D}^{=c} \Phi &\triangleq \mathcal{D}^{[c, c]} \Phi \\ \mathcal{D}^{\geq c} \Phi &\triangleq \mathcal{D}^{[c, \infty)} \Phi \\ \mathcal{D}^{>c} \Phi &\triangleq \mathcal{D}^{(c, \infty)} \Phi \\ \mathcal{J}^{<c} \Phi &\triangleq \mathcal{D}^{<c}(\neg \mathcal{D}^{<c} \neg \Phi) \\ \Phi_1 \mathcal{S}^I \Phi_2 &\triangleq ((\Phi_1 \wedge \neg \Phi_2) \mathcal{S}^I \Phi_2) \wedge \mathcal{D}^I \Phi_2 \end{aligned}$$

Figure 11: Additional derived operators.

Again, with reference to Figure 3, Figure 12 shows in green all the points satisfying $\mathcal{D}^{>2} p$, according to the Chamfer distance.

Intuitively, the $\mathcal{J}^{<c}$ operator can be used as a form of filtering, eliminating small details caused by noise in the fine-scale structure of an image; this method is akin to the nested application of \mathcal{N} and \mathcal{I} described in Section 2.1.1, parameterised with respect to a chosen maximum size c of details to be suppressed. To see this, recall that $\mathcal{I}\Phi$ is defined as $\neg \mathcal{N} \neg \Phi$, therefore $\mathcal{N}\mathcal{I}\Phi$ is the same as $\mathcal{N}(\neg \mathcal{N} \neg \Phi)$, which is very similar to the definition of $\mathcal{J}^{<c}$, with \mathcal{N} replaced by $\mathcal{D}^{<c}$.

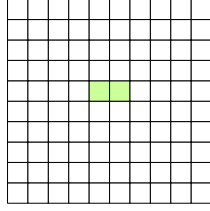


Figure 12: The points in Figure 3 satisfying $\mathcal{D}^{>2}p$ are shown in green.

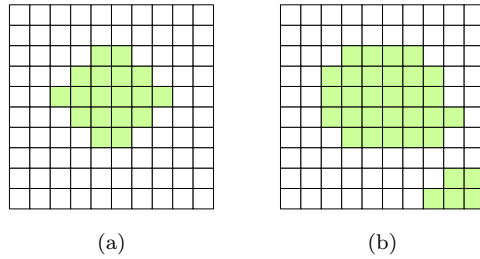


Figure 13: The points in Figure 3 satisfying $\mathcal{J}^{<3}q$ (a) and those satisfying $\mathcal{J}^{<2}q$ (b) are shown in green.

With reference to Figure 3, Figure 13a shows in green the points satisfying $\mathcal{J}^{<3}q$, whereas those satisfying $\mathcal{J}^{<2}q$ are given in Figure 13b.

The *bounded surrounded operator* $\Phi_1 \mathcal{S}^I \Phi_2$ is satisfied by a point x if and only if x satisfies Φ_1 , is strongly surrounded by points satisfying Φ_2 and its distance from such points falls in interval I . Note that, in the first argument of \mathcal{S} , it is required that $\neg\Phi_2$ holds as well; this ensures that all Φ_2 -points are at a distance of at least $\inf I$ from x .

In Figure 14a (Figure 14b, respectively) the points satisfying $q \mathcal{S}^{[2,2]} p$ ($q \mathcal{S}^{[2,3]} p$, respectively) are shown in green. Note that a similar operator has been defined in [61], which turns out to be stronger than $\mathcal{S}^{[a,b]}$, i.e., denoting the former operator by $\hat{\mathcal{S}}^{[a,b]}$, we have that, for all formulas Φ_1, Φ_2 , $(\Phi_1 \wedge \neg\Phi_2) \hat{\mathcal{S}}^{[a,b]} \Phi_2$ implies $\Phi_1 \mathcal{S}^{[a,b]} \Phi_2$.

We close this section pointing out that, for finite models, the operator $\mathcal{D}^{\leq c}$ coincides with the operator $\exists^{\leq c}$ proposed in [66]:

$$\mathcal{M}, x \models \exists^{\leq c} \Phi \Leftrightarrow \exists y. d(x, y) \leq c \text{ and } \mathcal{M}, y \models \Phi$$

and similarly for $\mathcal{D}^{< c}$ and $\exists^{< c}$. Note that, this coincidence does not hold in general, e.g. for Euclidean spaces. Our choice of the specific distance operator is motivated by its natural compatibility with distance transforms, as we illustrate in Section 2.2.2 below.

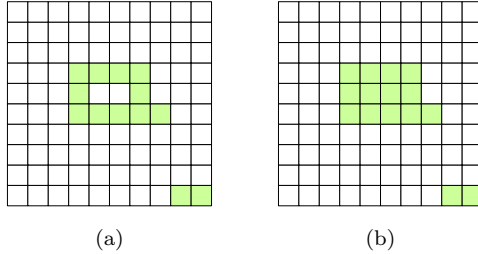


Figure 14: The points in Figure 3 satisfying $q\mathcal{S}^{[2,2]}p$ (a) and those satisfying $q\mathcal{S}^{[2,3]}p$ (b) are shown in green.

2.2.2 Model checking SLCS with distance operators

For distance-based operators, generally speaking, the time complexity of naive model checking algorithms is quadratic in the size of the space (see [61] for an example). However, given a Euclidean graph representing a multi-dimensional image, spatial model checking of formulas $\mathcal{D}^I\Phi$ for Euclidean or Chamfer distance can be done in linear time or 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 [48], and are extensively used in modern image processing and computer graphics [24]. In particular, effective linear-time algorithms have been recently introduced in the literature [56, 33]. Basically, a *distance transform* takes a model \mathcal{M}_{in} as input and produces a model \mathcal{M}_{out} as output as follows. Let \mathcal{M}_{in} be a model $((X, \mathcal{C}_R), d, \mathcal{A}_{in}, \mathcal{V}, \mathcal{W})$ such that every point $x \in X$ has an attribute, say a_{in} , defined on a binary domain, say $\{0, 1\}$ —the value of such an attribute may represent the fact that the point satisfies a given formula Φ or not. The output model will be $\mathcal{M}_{out} = ((X, \mathcal{C}_R), d, \mathcal{A}_{out}, \mathcal{V}, \mathcal{W})$ such that every point $x \in X$ has an attribute, say a_{out} , and $\mathcal{A}_{out}(x, a_{out}) = d(x, \{y \in X | \mathcal{A}_{in}(y, a_{in}) = 1\})$. The fragment of the model checking algorithm related to $\mathcal{D}^I\Phi$ is sketched in Figure 15. During the first step, the points satisfying Φ are marked, so that in the second step each point is annotated with its distance from the set of marked points and, finally, the set of the points with a distance laying in the interval I is returned.

In **topochecker**, two of the standard algorithms for distance transform are currently implemented; one for the Euclidean distance d_E and the other for the Chamfer distance d_C . Correspondingly, two distance operators are provided, with syntax as in Table 2.

For Euclidean distances, **topochecker** uses the linear algorithm that was proposed by Maurer in [56]. Such algorithm computes Euclidean distance transforms on anisotropic multi-dimensional grids (such as 2D and 3D medical images); it has linear complexity, its run-time is predictable, and it is among the most efficient algorithms for the purpose [34]. The general idea of the algo-

Input:

A model $\mathcal{M} = ((X, \mathcal{C}_R), \mathcal{A}, \mathcal{V})$;
 and a formula $\mathcal{D}^I \Phi$;

Output:

The set of points in X satisfying $\mathcal{D}^I \Phi$;

Step 1:

Compute intermediate model $\mathcal{M}' = ((X, \mathcal{C}_R), \mathcal{A}', \mathcal{V})$ such that for all $x \in X$:

$\mathcal{A}'(x, a') = 1$ if $\mathcal{M}, x \models \Phi$ and

$\mathcal{A}'(x, a') = 0$ if $\mathcal{M}, x \not\models \Phi$

Step 2:

Compute intermediate model $\mathcal{M}'' = ((X, \mathcal{C}_R), \mathcal{A}'', \mathcal{V})$ such that for all $x \in X$:

$\mathcal{A}''(x, a'') = d(x, \{y \in X \mid \mathcal{A}'(y, a') = 1\})$;

Step 3:

return $\{x \in X \mid \mathcal{A}''(x, a'') \in I\}$.

Figure 15: Sketch of the model checking algorithm for $\mathcal{D}^I \Phi$.

$\mathcal{D}^{\boxtimes c}$	$\mathcal{D}^{\boxtimes c}$
based on d_E	based on d_C
EDT($-, \boxtimes c$)	MDDT($-, \boxtimes c$)

Table 2: topochecker syntax for distance operators.

rithm 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. We refer the interested reader to [24], where a theoretical study of the algorithm is provided. 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 [41]. 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 Φ , the queue is initialised with all the nodes that satisfy Φ and have an outgoing edge reaching a node not satisfying Φ ; 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 shortest-path distance from the the set of nodes satisfying Φ , as required by the specification. The 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. [72]). However, the effective run-time behaviour 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.

3 A Logical Framework for Texture Analysis

In this section we define an additional logic operator that, when incorporated in the spatial logic presented in the previous sections, provides a logical framework for 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 distinct combinations of features, such as brightness, colour, shape, size. TA includes several techniques and has proved promising in a large number of applications in the field of medical imaging [47, 54, 13, 27]; in particular it has been used in *Computer Aided Diagnosis* [77, 44, 46] and for classification or segmentation of tissues or organs [14, 65, 64]. 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 [47]. Our work is mostly focused on *statistical* approaches to texture analysis. For two-dimensional images, statistical methods consist of extracting a set of statistical *descriptors* from the

distributions of local features in a neighbourhood of each pixel.

In this paper, we explore the use of *first order* statistical methods, that are statistics based on the probability distribution function of the intensity values of the pixels of parts, or the whole, of an image. The classical first-order statistical approach to TA makes use of statistical indicators of the local distribution of image intensity around each pixel, such as *mean*, *variance*, *skewness*, *kurtosis*, *entropy* [69]. Although such indicators ignore the relative spatial placement of adjacent pixels, statistical operators are useful in MI because their application is invariant under transformations of the image, in particular *affine transformations* (rotation and scaling), which is necessary when analysing several images acquired in different conditions. It is worth mentioning that current research also focuses on constructing features using first order operators, keeping some spatial coherence, but losing at least partially the aforementioned invariance [73]. The method we propose is an attempt to improve over the classical setting described above, by analysing (the histograms of) statistical distributions directly.

In the following, we introduce a spatial logic 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 pixels is *similar* to that of a chosen reference region. Several similarity measures exist (see [59]), that can be used to compare distributions in 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 in the case of *image co-registration* ([10])⁵.

3.1 A logical operator for statistical similarity

The statistical distribution of an area Y of a black and white image is approximated by the *histogram* h of the grey levels of points (pixels or voxels, for two- and three-dimensional images) belonging to Y , defined as follows. Given a minimum value m , a maximum value M , and a positive number of *bins* k , let $\Delta = (M - m)/k$ and define the histogram h as a function associating to each *bin* $i \in [1, k]$ the number of points that have intensity in the (half-open) interval $[(i - 1) \cdot \Delta + m, i \cdot \Delta + m)$. The minimum value m and the maximum value M are aimed at improving the resolution of histograms, by excluding rare peaks in the image, that may be due to artefacts in acquisition and would result in a high number of empty bins. A formal definition is given below:

Definition 11 *Given closure model $\mathcal{M} = ((X, \mathcal{C}), \mathcal{A}, \mathcal{V})$, we define function $\mathcal{H} : A \times 2^X \times \mathbb{R} \times \mathbb{R} \times \mathbb{N} \rightarrow (\mathbb{N} \rightarrow \mathbb{N})$ such that for all values $m, M \in \mathbb{R}$, with $m < M$, and $k, i \in \mathbb{N}$, with $k > 0$ and $i \in [1, k]$, letting $\Delta = \frac{M-m}{k}$:*

⁵In 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.

$$\mathcal{H}(a, Y, m, M, k)(i) = |\{y \in Y \mid (i-1)\Delta \leq \mathcal{A}(y, a) - m < i\Delta\}| \quad \bullet$$

So $\mathcal{H}(a, Y, m, M, k)$ is the histogram of the distribution of the values of attribute a of the points in Y , in the interval $[m, M]$ with step Δ . The above definition applies also to quasi-discrete / distance closure models.

In the sequel, for histogram $h : [1, k] \rightarrow \mathbb{N}$ we let $\bar{h} = \frac{1}{k} \sum_{i=1}^k h(i)$ denote the mean of h .

The definition of *cross correlation* between two histograms follows:

Definition 12 Let $h_1, h_2 : [1, k] \rightarrow \mathbb{N}$ be two histograms. The cross correlation of h_1 and h_2 is defined as follows:

$$\mathbf{r}(h_1, h_2) = \frac{\sum_{i=1}^k (h_1(i) - \bar{h}_1) (h_2(i) - \bar{h}_2)}{\sqrt{\sum_{i=1}^k (h_1(i) - \bar{h}_1)^2} \sqrt{\sum_{i=1}^k (h_2(i) - \bar{h}_2)^2}} \quad \bullet$$

The value of \mathbf{r} is *normalised* so that $-1 \leq \mathbf{r} \leq 1$; $\mathbf{r}(h_1, h_2) = 1$ indicates that h_1 and h_2 are *perfectly correlated* (that is, $h_1 = ah_2 + b$, with $a > 0$); $\mathbf{r}(h_1, h_2) = -1$ indicates *perfect anti-correlation* (that is, $h_1 = ah_2 + b$, with $a < 0$). On the other hand, $\mathbf{r}(h_1, h_2) = 0$ indicates no correlation. Note that normalisation makes the value of \mathbf{r} 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 $\mathbf{r}(h_1, h_2) = 1$ when both histograms are constant, and $\mathbf{r}(h_1, h_2) = 0$ when only one of the h_1 or h_2 is constant. We are now ready for embedding the statistical similarity operator $\Delta_{\bowtie c} \left[\begin{smallmatrix} m & M & k \\ \rho & a & b \end{smallmatrix} \right]$ in ImgQL .

Definition 13 Satisfaction $\mathcal{M}, x \models \Phi$ at point $x \in X$ in distance closure model $\mathcal{M} = ((X, \mathcal{C}), d, \mathcal{A}, \mathcal{V})$ is defined by induction on the structure of formulas, by adding the following equation, where $m, M \in \mathbb{R}$, with $m < M$, and $k \in \mathbb{N}$, with $k > 0$:

$$\mathcal{M}, x \models \Delta_{\bowtie c} \left[\begin{smallmatrix} m & M & k \\ \rho & a & b \end{smallmatrix} \right] \Phi \Leftrightarrow \mathbf{r}(h_a, h_b) \bowtie c$$

with:

$$h_a(i) = \mathcal{H}(a, S(x, \rho), m, M, k)(i),$$

$$h_b(i) = \mathcal{H}(b, \{y \mid \mathcal{M}, y \models \Phi\}, m, M, k)(i),$$

$\bowtie \in \{=, <, >, \leq, \geq\}$, and

$S(x, \rho) = \{y \in X \mid d(x, y) \leq \rho\}$ is the sphere of radius ρ centred in x , to the equations in Figure 2. •

So $\Delta_{\bowtie c} \left[\begin{smallmatrix} m & M & k \\ \rho & a & b \end{smallmatrix} \right] \Phi$ compares the region of the image constituted by the sphere of radius ρ centred in x against the region characterised by Φ . The comparison is based on the cross correlation of the histograms of the chosen attributes of (the points of) the two regions, namely a and b and both histograms share the same domain ($[m, M]$) and the same bins ($[1, k]$). In summary, the operator allows to check to which extent the sphere around the point of interest is *statistically similar* to a given region (specified by) Φ .

3.2 Model checking ImgQL with statistical similarity operators

In `topochecker` the statistical similarity operator `SCMP` is provided, with the following syntax

$$\text{SCMP}(\mathbf{a}, \mathbf{fa}, \mathbf{R}, \bowtie c, m, M, k)(\mathbf{b}, \mathbf{fb})$$

corresponding to $\Phi_a \wedge \Delta_{\bowtie c} \left[\begin{smallmatrix} m & M & k \\ \rho & a & b \end{smallmatrix} \right] \Phi_b$ where \mathbf{fa} , \mathbf{fb} and \mathbf{R} represent Φ_a , Φ_b and ρ , respectively, in the language of the tool. For example, formula

$$\text{SCMP}(\mathbf{a}, \text{TT}, 10.0, \geq 0.7, 200, 2000, 100)(\mathbf{b}, \text{TT})$$

is true at voxels centred in a region—of radius 10.0—where the distribution of the values of \mathbf{a} has cross-correlation greater than 0.7 with the distribution of the values of \mathbf{b} in the whole image. In this case, cross-correlation is computed using 100 bins, and taking into account only values between 200 and 2000.

The algorithm we use for implementing the `SCMP` operator is straightforward. An array v_b , sized to the number of bins k , is allocated, initialised to 0 at each index, and the histogram h_b is stored in it, by iterating over all the points y of satisfying Φ , finding the index i of the bin corresponding to the grey level of y , and increasing the corresponding value of $v_b[i]$. An array v_a , sized to the number of bins k , is allocated. For each pixel x , v_a is (re-)initialised to 0 at each index, and all the points y laying in the sphere of radius ρ centred in x are examined; for each y , the index i of its bin is identified, and the value of $v_a[i]$ is increased, so that when all the y have been examined, v_a represents the histogram h_a . The cross-correlation value $\mathbf{r}(h_a, h_b)$ is then computed by simple calculations that are linear in the number of bins k . This algorithm has time complexity proportional to $v \cdot \rho^n \cdot k$, where v is the number of pixels in the image and n the number of dimensions (indeed the number of pixels in an n -dimensional sphere is proportional to ρ^n). Since ρ and k are usually fixed along a given analysis, such algorithm can still be considered “linear” in the size of the image. This basic procedure is amenable to optimisation, for instance by observing that the spheres centred around two different points of the image may share some pixels, therefore the histogram of each one could be computed starting from the histogram of the other, at the expenses of more memory needed to store the histogram of different points. We leave the study of similar optimisations for future work as the algorithm we discussed has proved to be sufficiently fast for our experiments.

4 Using topochecker for MI analysis with ImgQL

MR images are produced using different kinds of *sequences* of magnetic field gradients and radio-waves. Images so obtained are called *weighted* images; these can be further post-processed in various ways. For instance, typical weighted images are those produced using *Fluid-attenuated inversion recovery* pulse sequence

(MR-FLAIR), *T2 weighted pulse sequence* (MR-T2w), or *diffusion weighted images*, whereas the *Apparent Diffusion Coefficient* maps (ADC) are obtained via post-processing of diffusion weighted images. A standard reference for such matters is [11]. In this section, we illustrate our approach on some of these kinds of medical images, using a tutorial-like, step-wise style. We do this by means of two examples of segmentation in MRI, introduced below:

1. Glioblastoma tumour and oedema segmentation in images obtained using MR-FLAIR; this analysis is carried out in two dimensions.
2. Rectal cancer segmentation in images obtained using MR-T2w and ADC. This analysis is carried out in three dimensions.

For glioblastoma, our procedure was successfully tested on five images from different sources, that were acquired in very different conditions. However, validation of the methodology for actual clinical usage requires extensive clinical research. We refer to Section 4.3 for preliminary validation results and a more detailed discussion.

Model definitions using medical images are introduced in `topochecker` by associating an arbitrary number of attribute names to files containing medical images⁶, as follows:

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

where `med` is a keyword indicating the file type, and `img1`, `img2` ... are names of pixel/voxel attributes, for which the related values are drawn from `file1.nii`, `file2.nii` ..., respectively.

For this to work, all the loaded images must have the same voxel coordinates and orientation (e.g., coming from the same machine and type of acquisition, or after manual resampling). No resampling is currently done in `topochecker`.

4.1 Example: segmentation of glioblastoma

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

Normal tissues of the head can be divided into several classes. The outer layer of the head consists of adipose tissue (and skin) surrounding the skull that in turn consists of bone and bone-marrow. The skull encloses the brain

⁶The model loader of `topochecker` currently supports the NIFTI (Neuroimaging Informatics Technology Initiative) file format (<https://nifti.nimh.nih.gov/>, version 1 and 2). In this work, images downloaded from *Radiopaedia.org* in *jpeg* format, and *dicom* images have been converted to NIFTI-1.

tissues. The brain itself is suspended in cerebrospinal fluid (CSF) and isolated from the blood stream. Thresholds in the grey levels of images can be used to single out specific tissues in a medical image; however, in doing so, noise is generated in the form of (small, scattered) regions not belonging to the tissue. The relative positioning of tissues — the so-called *topological information* of the image — plays an important role in suppressing such noise. We will see in the following how such information is encoded by logic formulas in the methodology we propose.

GBMs are intracranial tumours composed of typically poorly-margined, 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-FLAIR images GBMs appear *hyperintense* and surrounded by *vasogenic oedema*⁷.

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.

4.1.1 Preprocessing

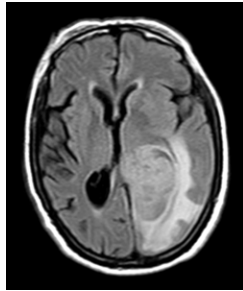
Histograms of grey levels of images⁸ of the same body part may differ from each other due to inter-patient or inter-scanner differences or depending on the actual acquisition volume (Figure 16) or the file format used to store the image⁹.

More uniform results, on different images, can be obtained by dividing the intensity of each pixel by the average of the intensity levels of all the *significant* pixels in the image. A pixel is considered significant when it does not belong to the background. Significant pixels are selected using a Boolean mask (indicated by the green area in Figure 17c). In order to compute such a mask, 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 its histogram. This situation is witnessed in the histogram

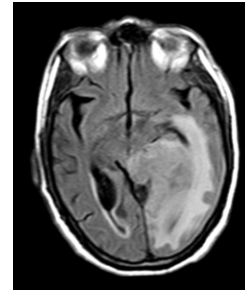
⁷Vasogenic oedema is an abnormal accumulation of fluid from blood vessels, which is able to disrupt the blood-brain barrier and invade extracellular space.

⁸To ease visual comparison, in Figs.16d, 17d, 18b, the histograms that we show are normalised so that the measure of the area below the curve is 1.

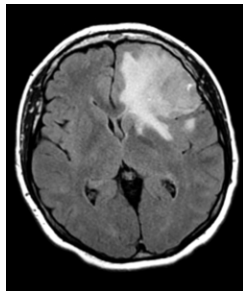
⁹For instance, *jpeg* images, as downloaded from *Radiopaedia.org*, typically use 8-bit precision (typical range 0-255) (see Figure 16) whereas *dicom* images saved by scanners typically use 12 or 16-bit (for MR images, the typical range is 0-4096 or 0-65536, respectively) (see Figure 18).



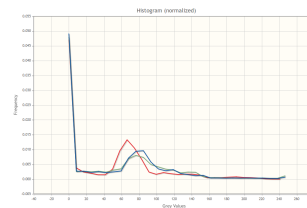
(a) Case courtesy of Dr. Ahmed Abd Rabou, Radiopaedia.org, rID: 22779.



(b) A different slice of the acquisition in Figure 16a.

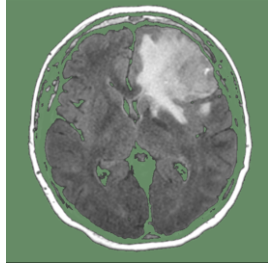


(c) Case courtesy of A.Prof Frank Gaillard, Radiopaedia.org, rID: 5292.

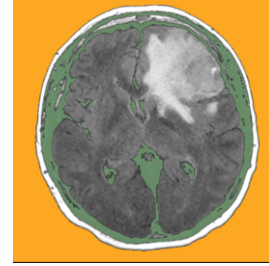


(d) Histograms of Figure 16a (blue), 16b (green), 16c (red).

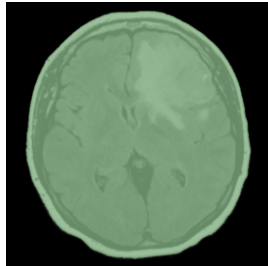
Figure 16: Slice of MR-FLAIR brain acquisition of different patients and corresponding histogram.



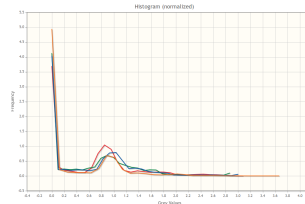
(a) The pixels in Figure 16c with grey levels below a given threshold are shown in green.



(b) The sub-mask that touches the border of the image is shown in orange.



(c) The mask of the image excluding the background is shown in green.



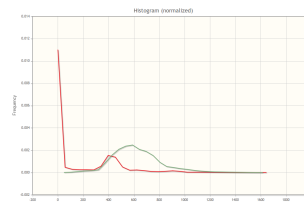
(d) Histograms of normalised version of images in Figure 16a (blue), Figure 16b (green), Figure 16c (red) and Figure 18a (orange).

Figure 17: Finding the mask for normalisation.

by a peak close to 0. A threshold is thus selected for each image as the value immediately following such peak. Using this threshold, it is possible to isolate the background, by separating it from the head (Figure 17a). 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 17b) and its complement, that is, the green area in Figure 17c, is finally used to select the significant pixels to compute the mean value for normalisation. Figure 17d shows the histograms of images after normalisation.

We remark that *equalisation of histograms* is another form of normalisation, frequently used for texture analysis ([45]). We do not use this method as it changes the relationship between grey levels of different structures in the image (as shown in Figure 18), that we use rather prominently for differentiating different tissues; normalisation of image intensity is sufficient for our purposes.

Awaiting for publication approval



(a) A slice of MR acquisition of brain (on the left) and its equalised version (on the right) (Case courtesy of Azienda Ospedaliera Universitaria Senese; image and data processing performed in compliance with EU GRDP 679/2016)

(b) Histograms of grey levels of the original (red) and equalised (green) version of image in 18a.

Figure 18: Effect of histogram equalisation.

4.1.2 Brain segmentation

In this second phase of our method, `topochecker` is used to perform a segmentation of the brain in order 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.

In the process below, we fix some thresholds for identifying different tissues in the brain; note that, thanks to the preprocessing step described above, these can be kept uniform across different images.

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

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

The model definition in `topochecker` is as follows.

```
Model "med:FLAIR=GBM-NORM.nii";
```

GBM-NORM.nii is the normalised NIfTI image of the MR-FLAIR acquisition shown in Figure 16c. By the above declaration, in the rest of the analysis, the relevant attribute, i.e. the normalised grey level, of each pixel in this image is referred to as FLAIR. Formula definitions for general derived operators `reach` and `touch` are given according to Figure 6 (\mathcal{R} , \mathcal{T}); similarly for operator $\mathcal{J}^{<c}$, with reference to Figure 11:

```
Let reach(a,b) = !(b S !a);
Let touch(a,b) = a & reach(a|b,b);
Let flt(a) = MDDT(!(MDDT(!a,<1)),<1);
```

Furthermore, we define also a few operators that serve as macros and that are specifically useful in the segmentation procedure that follows:

```
Let grow(a,b) = (a|b) S (!b);
Let denoise(a) = touch(a,MDDT(!a,>=2));
Let closeTo(a) = MDDT(a,<3.0);
```

Formula `grow(a,b)` is inspired by the image segmentation method of *seeded region growing* [1]. This method starts from a number of *seed points* in the region of interest and examines neighbouring points to decide whether they should be added to the region. We start from points that satisfy property `a` and to which all points satisfying property `b` are added that, together with those satisfying `a`, form a common region that is surrounded by points that do not satisfy `b`. Formula `grow(a,b)` can be used only when it is guaranteed that all points satisfying `a` are also satisfying `b` (but not the other way around).

Let A be the set of points satisfying formula `a`. The formula `denoise(a)` is used to remove small areas from A , as follows: first A is shrunk by 2.0 units; in doing so, some subareas of A may disappear; the areas that do not disappear are restored to their original shape by means of the `touch` operator. This operation is similar to `flt`, but it preserves the contours of the original area A . The formula `closeTo(a)` denotes the points that lay at a distance less than 3.0 units from the set of points satisfying `a`. For this analysis the MDDT Chamfer distance operator uses 4 adjacent pixels per node and the distance units are in millimetres with respect to the actual dimension of the head, i.e. the *real-world* coordinates.

Next we define a number of useful thresholds for the grey levels of the image that are used to obtain a first approximation of different kinds of tissue of interest:

```
Let lowIntsty = [FLAIR < 0.5];
Let medIntsty = [FLAIR > 0.5] & [FLAIR < 1.3];
Let highIntsty = [FLAIR > 1.7];
Let tumIntsty = [FLAIR > 1.17] & [FLAIR < 1.53];
Let oedIntsty = [FLAIR >= 1.47] & [FLAIR < 2.4];
```

We distinguish three general levels of intensity (low, medium, and high), and two specific intensities that are typical for tumour and oedema, respectively.

We are now ready to start the segmentation procedure. First we identify the points that are part of the background of the image. These all have a very low intensity, but there are other points in the image that have low intensity as well. What distinguishes the points of the background from the other low intensity points is that the background area touches the border of the image. In `topochecker`, when loading images, a special atomic proposition named `border` is defined, that is satisfied by points that form the border of an image. This way points of the background are exactly those that satisfy the property `background`:

```
Let background = touch(lowIntsty,[border]);
```

The points that satisfy `background` are shown in red in Figure 19b. The original image is shown in Figure 19a.

The next step is to look for the external border of the head, consisting of skin and adipose tissue. For our purposes, it is sufficient to identify the adipose tissue, since the brain is surrounded by the adipose tissue, which separates it from the skin. Adipose tissue in the normalised MR-FLAIR images has intensity above 1.7, so of high intensity. As before, there may be other points with high intensity in the image, but we exploit the knowledge that adipose is at the external border of the head, and thus close to the background. These points can be found with the following formula:

```
Let adipose = touch(highIntsty,closeTo(background));
```

The points that satisfy `adipose` are shown in green in Figure 19b.

Using the properties `background` and `adipose` it is not difficult to specify the points that are part of the head. These are all those points that are not part of the background or close to adipose tissue.

```
Let head = !(closeTo(adipose) | background);
```

The points that satisfy `head` are the union of the green and red points in Figure 19c (see below).

In the next steps we show how we can distinguish the various tissues within the area of the head, namely the brain and the cerebrospinal fluid (CSF) that contains it. We start from the identification of points that are part of CSF. These are points that are within the head and that have low intensity:

```
Let CSF = lowIntsty & head;
```

The points that satisfy `CSF` are shown in red in Figure 19c.

We proceed with segmentation of the brain in four subsequent steps. As a first approximation we look for the points of the brain with medium intensity within the head (and that are not belonging to CSF). Within this approximation we select some inner areas that are most certainly part of brain tissue and that can serve as a seed from which to ‘grow’ in a more precise way points belonging to the brain. Finally, we remove pieces that have been erroneously identified as part of the brain, but that are actually relatively small areas that are part

of the skull or bone, having a similar intensity as that of the brain. This way we obtain all pixels that are actually part of the brain. The four steps of the specification are given below.

```
Let brainApprox = head & (!CSF) & medIntsty;
Let brainSeed = MDDT(!brainApprox,>10);
Let noisyBrain = grow(brainSeed,head & (!CSF));
Let brain = touch(noisyBrain,brainSeed);
```

The points that satisfy `brainApprox` (`brainSeed`, `noisyBrain`, respectively) are shown in green in Figure 19d (Figure 19e, Figure 19f, respectively). The final result of the brain is shown in Figure 19g.

4.1.3 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 the thresholds we introduced before that provide a rough segmentation of the image shown in Figure 19a:

```
Let tum0 = flt(tumIntsty S (brain|CSF));
Let oed0 = flt(oedIntsty S (brain|CSF));
```

In Figure 19h we show in red the points that satisfy formula `oed0` referring to the oedema, and in green those satisfying formula `tum0` referring to the tumour. These are points that have the selected intensity (`oedIntsty` and `tumIntsty`, respectively) and are part of the brain tissue, i.e. they are surrounded by `brain` or `CSF`. Note that the regions `oed0` and `tum0` are partially overlapping. Moreover, we remove from these identified regions areas whose radius is smaller than 1mm using the `flt` operator defined above.

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 exploit this knowledge using the distance operator `MDDT` as follows:

```
Let oeddst = MDDT (oed0,<=2.0);
Let tum1 = touch(tum0,oeddst);
Let oed1 = oed0 & reach(oeddst,tum1);
```

We first define the region `oeddst` at distance less than 2mm from `oed0`; then select sub-regions of `tum0` that *touch* `oeddst` (formula `tum1`) and sub-regions of `oed0` that can reach points satisfying `tum1` by passing only through points satisfying `oeddst` (formula `oed1`). The result is shown in Figure 19i. Comparing the latter with Figure 19h we can observe that some green areas, located in the left half of the brain, have disappeared. These were points with a similar intensity as that of tumour tissue, but not actually part of it since they were not connected to the tumour. 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.

```
Let tum2 = denoise(tum1);
Let oed2 = denoise(oed1);
```

Figure 19j illustrates the areas defined by `tum2` (green) and `oed2` (red). Compared to Figure 19i this removes a number of small detached areas of oedema that were located in the tumour area and should be considered as noise.

Finally, `tumor` and `oedema` are defined as being inter-reachable. This part could remove some separate areas that have tumour or oedema intensity but should not be considered as such since they are too far apart. In this particular case no such areas were present apparently as can be observed comparing Figure 19j with the final output of the segmentation in Figure 19k.

```
Let tumor = touch(tum2,oed2);
Let oedema = touch(oed2,tum2);
```

As a further result, in Figure 20 we show the final segmentation of tumour and oedema on two other images from a different patient applying the same specification. Fig 20b shows the segmentation of the image in Figure 16a, and Figure 20d shows the one of the image in Figure 16b. The original images are also shown aside of the result in Figure 20 for more convenient comparison.

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 red 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 2D. The same approach also works in 3D, with minor modifications. Figure 21 shows some slices of the segmentation of MR-FLAIR acquisition of the patient in Figure 18a, 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 3D for accuracy reasons.

In the GBM case we have not used the statistical comparison operator, which will instead be used in the example in Section 4.2. In general there may be many different ways to obtain an accurate segmentation. Ideally, these should be robustly working for many different images, both in 2D and in 3D. In future work we plan to investigate this in more detail and compare various variants from the point of view of robustness, accuracy and computational efficiency. Regarding the latter, the 2D analysis of GBM was performed on a standard laptop (equipped with an Intel CORE i7 CPU, and 8 gigabytes of RAM) and performed in a little less than 1 minute, which as a first indication is in line with the current state-of-the-art.

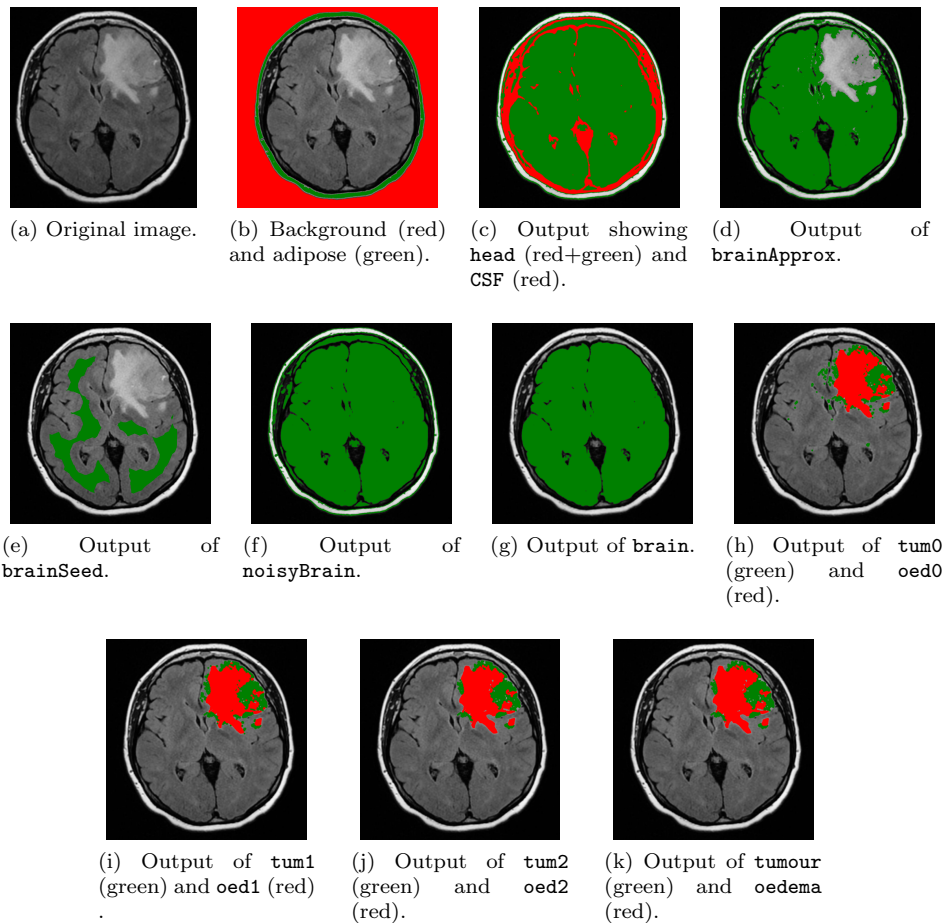


Figure 19: Experimental results of using `topochecker` for segmentation of glioblastoma (green) and oedema (red) (case courtesy of A.Prof Frank Gailard, Radiopaedia.org, rID: 5292).

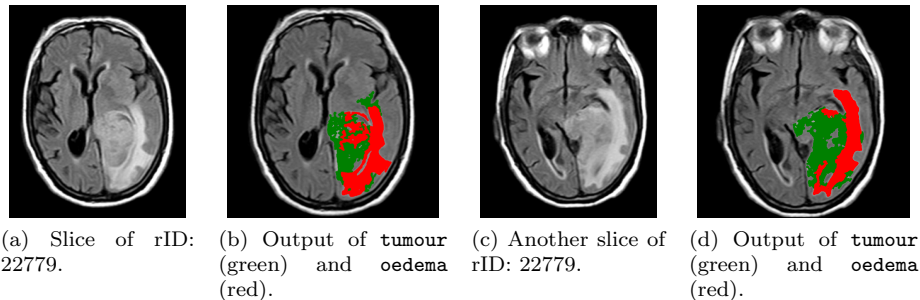


Figure 20: Additional results of using `topochecker` for segmentation of glioblastoma (green) and oedema (red) (Case courtesy of Dr. Ahmed Abd Rabou, Radiopaedia.org, rID: 22779).

Awaiting for
publication
approval

(a)

Awaiting for
publication
approval

(b)

Awaiting for
publication
approval

(c)

Awaiting for
publication
approval

(d)

Awaiting for
publication
approval

(e)

Awaiting for
publication
approval

(f)

Figure 21: Slices of an image obtained using the method we presented 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 (Case courtesy of Azienda Ospedaliera Universitaria Senese; image and data processing performed in compliance with EU GRDP 679/2016).

4.2 Example: segmentation of rectal carcinoma

As a further example we detail an analysis aimed at segmentation of rectal cancer. Rectal carcinoma is a frequent pathology [67] 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. Furthermore, 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 [70]. Briefly, 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 so-called 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 [51], 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 composed of four steps:

1. preprocessing (performed without use of `topochecker`), aimed at normalisation of images;
2. rectum segmentation in T2w images, to limit the area of the image where the tumour is searched for; 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 the ADC map, which is the stated goal of this example.

4.2.1 Preprocessing

Figure 22 shows one axial and one sagittal view of T2w and ADC acquisitions.

Awaiting for
publication
approval

(a) T2w axial slice.

Awaiting for
publication
approval

(b) T2w sagittal slice.

Awaiting for
publication
approval

(c) ADC map axial slice.

Awaiting for
publication
approval

(d) ADC map sagittal slice.

Figure 22: Rectum acquisition (Case courtesy of Azienda Ospedaliera Universitaria Senese; image and data processing performed in compliance with EU GRDP 679/2016).

Awaiting for
publication
approval

Awaiting for
publication
approval

Awaiting for
publication
approval

(a) DWI slice.

(b) ADC slice.

(c) Mask on ADC.

Figure 23: ADC mask (Case courtesy of Azienda Ospedaliera Universitaria Senese; image and data processing performed in compliance with EU GRDP 679/2016)

Since the FOV (*Field of View*) of the T2w acquisition lies entirely within the patient body (Figure 22a and Figure 22b), 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, because in the ADC maps the background is very noisy (see Figure 23 – note that DWI and ADC masks have the same coordinate system).

4.2.2 Rectum segmentation

The model definition for rectum segmentation defines the relevant attribute, i.e. the normalised grey level of each pixel in this image, as `T2`, which is used throughout the analysis.

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

We slightly change the definition of `flt` (defined in Figure 11 (\mathcal{J}), 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. In order to remove such regions, we employ nested application of `I` and `N` on top of the definition of `flt`.

```
Let flt3D(a) = N(I(flt(a)));
```

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 a region having an intermediate intensity, called `intermT2r`, that are close to each other. This is detected using the `touch` operator.

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([T2>0.9] & [T2<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 24); 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 25).

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

Co-registration T2w and DWI images are acquired in the same acquisition session with the patient. Co-registration between the T2w and the ADC maps makes use of the image header, that in medical images contains 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 corresponding 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 26).

In order to co-register the ROI of the 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 27, the green area represents voxels on the ADC map that correspond to voxels in T2w satisfying `rectumS`.

4.2.3 Tumour segmentation

Finally, tumour segmentation is performed on the ADC map. Below, we load the (normalised) 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];
```


Awaiting for
publication
approval

(a) T2w axial slice.

Awaiting for
publication
approval

(b) T2w axial slice.

Awaiting for
publication
approval

(c) T2w sagittal slice.

Awaiting for
publication
approval

(d) T2w coronal slice.

Figure 24: Hyperintense (green), hypointense (brown) and intermediate intensity (red) regions used to segment rectum in T2w (Case courtesy of Azienda Ospedaliera Universitaria Senese; image and data processing performed in compliance with EU GRDP 679/2016).

Awaiting for
publication
approval

(a) T2w axial slice.

Awaiting for
publication
approval

(b) T2w axial slice.

Awaiting for
publication
approval

(c) T2w sagittal slice.

Awaiting for
publication
approval

(d) T2w coronal slice.

Figure 25: Final output of segmentation of rectum in T2w. In green `rectumS` (Case courtesy of Azienda Ospedaliera Universitaria Senese; image and data processing performed in compliance with EU GRDP 679/2016).

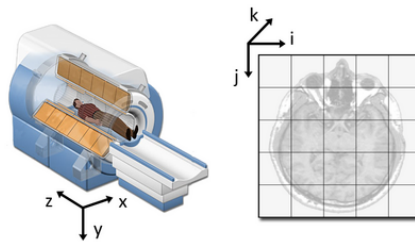


Figure 26: World (x, y, z) and image (i, j, k) coordinate systems. Image based on image shared in https://www.slicer.org/wiki/Coordinate_systems

Awaiting for
publication
approval

(a) ADC axial slice.

Awaiting for
publication
approval

(b) ADC axial slice.

Awaiting for
publication
approval

(c) ADC sagittal slice.

Awaiting for
publication
approval

(d) ADC coronal slice.

Figure 27: Co-registration of rectum ROI (green) segmented in T2w to ADC map (Case courtesy of Azienda Ospedaliera Universitaria Senese; image and data processing performed in compliance with EU GRDP 679/2016).

We delineate the initial estimate of the tumour based on grey levels (the green area in Figure 28a-28f).

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

Formula `tumor3`, 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 28g,28h,28i)

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

Statistical texture analysis is then used to find regions that are similar to `tumor3` (cross correlation > 0.8). The search is restricted to areas close to `tumor3`, i.e. the region of radius 20mm around `tumor3` defined as `tumorSpace` (Figure 29).

```
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` shown in green in Figure 30.

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

4.3 Validation

The work presented in this section is aimed at providing an illustration of the analysis capabilities of our logic-based methodology, rather than providing complete clinical case studies. For instance, consider the 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. Future work 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.

However, some conclusions can already be drawn from the data we have, both with respect to efficiency and 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 the size of the image (the algorithm is linear). In the glioblastoma example, MR-FLAIR very often have a slice size of 256×256

Awaiting for
publication
approval

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

Awaiting for
publication
approval

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

Awaiting for
publication
approval

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

Awaiting for
publication
approval

(d) Output of `f1t3D` operator on one ADC axial slice.

Awaiting for
publication
approval

(e) Output of `f1t3D` operator on one ADC sagittal slice.

Awaiting for
publication
approval

(f) Output of `f1t3D` operator on one ADC coronal slice.

Awaiting for
publication
approval

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

Awaiting for
publication
approval

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

Awaiting for
publication
approval

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

Figure 28: Output of segmentation of the tumour (green) and the rectum (brown) (Case courtesy of Azienda Ospedaliera Universitaria Senese; image and data processing performed in compliance with EU GRDP 679/2016).

Awaiting for
publication
approval

(a) ADC axial slice.

Awaiting for
publication
approval

(b) ADC sagittal slice.

Awaiting for
publication
approval

(c) ADC coronal slice.

Figure 29: Output of the `SCMP` operator (green) and the searching space `tumorSpace` (brown) (Case courtesy of Azienda Ospedaliera Universitaria Senese; image and data processing performed in compliance with EU GRDP 679/2016).

Awaiting for
publication
approval

(a) ADC axial slice.

Awaiting for
publication
approval

(b) ADC axial slice.

Awaiting for
publication
approval

(c) ADC sagittal slice.

Awaiting for
publication
approval

(d) ADC coronal slice.

Figure 30: Final output of rectal cancer segmentation in green (Case courtesy of Azienda Ospedaliera Universitaria Senese; image and data processing performed in compliance with EU GRDP 679/2016).

pixels, multiplied by 20 – 30 slices. As a rough estimate, the execution time for the analysis of a single 1024×1024 -voxels slice—including preprocessing—on a standard laptop (equipped with an Intel CORE i7 CPU, and 8 gigabytes of RAM), currently stays below one minute. This information, although not being a fully-fledged benchmark, provides a first indication that, efficiency-wise, our approach is in par with the state-of-the art in semi-automatic glioblastoma segmentation procedures (see [35]). We remark that our procedure makes use of a prototype general-purpose model checker, that could be amenable to further optimisation, e.g. employing specialised, well-known flood-filling algorithms for images for model checking the *surrounded* connective — in place of the current graph-theoretical method.

A preliminary assessment of the quality of the obtained results in the case of glioblastoma was performed for the patient in Figure 21. The patient underwent first surgery and then radiotherapy. We compared our results on the post-surgery MR-FLAIR with target volumes delineated on the pre-treatment Computed Tomography (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 glioblastomas 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 $DC = \frac{2V(MS \cap AS)}{V(MS) + V(AS)}$, where $V(a)$ is the volume of a , that is, the number of voxels that belong a ; DC ranges from 0 to 1, 0 indicates no overlap and 1 indicates complete overlap.

The CT volume was co-registered to the FLAIR volume. Then, we considered the region R obtained as the union of the oedema and tumour regions, as found using our method. We compared R to the GTV contour, and furthermore we compared R , expanded by 2.5cm (as explained above) to the CTV contour. We obtained $DC = 0.76$ for GTV and $DC = 0.81$ 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 glioblastoma [32].

In [7], a variant of the method we described was assessed on a dataset of 7 patients affected by GBM, that have undergone radiotherapy. The obtained results, and average execution time per patient on the same machine used for the experiments in the current paper, were in line with the state-of-the-art.

Finally, we note that numeric thresholds and other parameters (e.g., the number of nested 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, on the basis of expert knowledge on the matter and in some cases by trial-and-error. The values that we used 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. [32, 35, 68, 78]), 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.

5 Conclusions and future work

This work provides a first, promising exploration of logical methods for declarative medical image analysis in the domain of radiotherapy. A declarative approach makes analysis transparent, reproducible, human-readable, exchangeable, and permits domain experts who are not technicians to understand the specifications. Such advantages are akin to those obtained in other domains, such as the application of the *Structured Query Language* (SQL) in the field of databases, or the introduction of query languages (XPath, XSLT, ...) in semi-structured data management.

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 logic ImgQL is able to predicate on both shortest-path and Euclidean distance at the same time, and *topochecker* implements both operators. In MI, shortest-path distances proved useful so far mostly to speed up interactive development; this is implementation-dependent, as the Modified Dijkstra transform that we use (see Section 2.2.2) currently performs faster than Maurer distance transform in our tests. We also considered the embedding of specific operators for MI in ImgQL such as an operator for texture analysis based on first order statistical methods. Other options and operators could be considered following a similar approach, providing a way to include state-of-the-art analysis techniques that can be conveniently combined using the spatial operators of the logic.

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 [63]), by providing practitioners with models of a patient's body, with the relevant regions, identified by our method, printed in different colours. Such aspects constitute a further interesting line of research 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 indi-

vidual pixels or voxels. However, drawing inspiration from the family of *region calculi* (see [2]), 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 [61] could be useful in medical imaging. A limitation of the model checking algorithm in [61] is its quadratic complexity. We have shown that the application of distance transforms yields a *linear* algorithm for a weaker variant of the bounded surrounded operator for the case of images (that is, regular grids).

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.

Our 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 performed 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 [42, 39, 5, 6]), that pursues *machine learning* of the logical structure of image features. We emphasise that such a development, if implemented, would be a radical improvement in application of machine learning to medical imaging. It can be framed under the recent research trend on *explainable artificial intelligence*, as it would yield a procedure that can explain in terms of a human readable language the methodology that a machine learning algorithm extrapolates from data. Our topological approach to spatial logics would be a key enabling technique for this purpose, as the formulas obtained in the SpaTeL approach are not meant to be intelligible by humans. It is worth noting that machine learning and deep learning methods have also been applied to the detection of tumours in very recent literature [71, 4]. On the other hand, our application of machine learning could as well be focused simply on the identification of numeric parameters, rather than logic formulas, that may depend on complex features of images.

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. The normalisation step that we employ could be improved using state-of-the-art methods (see [55, 52], and the references therein).

Planned future developments 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 significant impact on methods that require human interaction—e.g. *interactive segmentation* in preparation for surgery, or *contouring* for radiotherapy planning.

6 Acknowledgments

The authors wish to thank dr. Marco Di Benedetto for suggesting the application of distance transforms to reduce 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] Rolf Adams and Leanne Bischof. Seeded region growing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16:641–647, 1994.
- [2] M. Aiello, I. Pratt-Hartmann, and J. van Benthem. *Handbook of Spatial Logics*. Springer, 2007.
- [3] Marco Aiello. *Spatial Reasoning: Theory and Practice*. PhD thesis, Institute of Logic, Language and Computation, University of Amsterdam, 2002.
- [4] Z. Akkus, A. Galimzianova, A. Hoogi, D. L. Rubin, and B. J. Erickson. Deep learning for brain mri segmentation: State of the art and future directions. *Journal of Digital Imaging*, 30:449–459, 2017.
- [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] G. Belmonte, V. Ciancia, D. Latella, M. Massink, M. Biondi, G. De Otto, V. Nardone, G. Rubino, E. Vanzi, and F. Banci Buonamici. A topological method for automatic segmentation of glioblastoma in mr flair for radiotherapy - ESMRMB 2017, 34th annual scientific meeting. *Magnetic Resonance Materials in Physics, Biology and Medicine*, 30(S1):437, oct 2017.
- [8] 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.
- [9] K.K. Brock. *Image processing in radiation therapy*. CRC Press, 2014.
- [10] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM Comput. Surv.*, 24(4):325–376, December 1992.
- [11] Robert W. Brown, Yu-Chung N. Cheng, E. Mark Haacke, Michael R. Thompson, and Ramesh Venkatesan, editors. *Magnetic Resonance Imaging*. John Wiley & Sons Ltd, apr 2014.
- [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, Stephen Gilmore, Gianluca Grilletti, Diego Latella, Michele Loreti, and Mieke Massink. Spatio-temporal model checking of vehicular movement in public transport systems. *International Journal on Software Tools for Technology Transfer*, Jan 2018.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999. ISBN 0-262-03270-8.
- [26] 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.
- [27] 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.
- [28] R. De Nicola, J. Katoen, D. Latella, M. Loreti, and M. Massink. Model checking mobile stochastic logic. *Theor. Comput. Sci.*, 382(1):42–70, 2007.
- [29] 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.
- [30] 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.

- [31] 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.
- [32] 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.
- [33] 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.
- [34] 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.
- [35] 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.
- [36] 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.
- [37] Antony Galton. A generalized topological view of motion in discrete space. *Theor. Comput. Sci.*, 305(1-3):111–134, 2003.
- [38] Antony Galton. Discrete mereotopology. In Claudio Calosi and Pierluigi Graziani, editors, *Mereology and the Sciences: Parts and Wholes in the Contemporary Scientific Context*, pages 293–321. Springer International Publishing, 2014.
- [39] 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.
- [40] 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.
- [41] G. J. Grevera. Distance transform algorithms and their implementation and evaluation. In *Deformable Models*, pages 33–60. Springer Science, 2007.
- [42] 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.

- [43] 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.
- [44] 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.
- [45] Robert M Haralick, Karthikeyan Shanmugam, et al. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973.
- [46] 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.
- [47] A. Kassner and R. E. Thornhill. Texture analysis: A review of neurologic MR imaging applications. *Am. J. Neuroradiol.*, 31(5):809–816, 2010.
- [48] 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.
- [49] R. Kontchakov, A. Kurucz, F. Wolter, and M. Zakharyashev. Spatial logic + temporal logic = ? In *Handbook of Spatial Logics*, pages 497–564. Springer, 2007.
- [50] O. Kutz, F. Wolter, H. Sturm, N. Suzuki, and M. Zakharyashev. Logics of metric spaces. *ACM Trans. Comput. Log.*, 4(2):260–294, 2003.
- [51] 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.
- [52] 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.
- [53] 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.

- [54] 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.
- [55] Anant Madabhushi and Jayaram K. Udupa. New methods of MR image intensity standardization via generalized scale. *Medical Physics*, 33(9):3426–3434, aug 2006.
- [56] 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.
- [57] G. P. Mazzara, R. P. Velthuisen, J. L. Pearlman, H. M. Greenberg, and H. Wagner. Brain tumor target volume determination for radiation treatment planning through automated mri segmentation. *Int. J. Radiat. Oncol. Biol. Phys.*, 59(1):300–312, 2004.
- [58] B. H. et. al. Menze. The multimodal brain tumor image segmentation benchmark (brats). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, 2015.
- [59] Kouros Meshgi and Shin Ishii. Expanding histogram of colors with griding to improve tracking accuracy. In *MVA*, 2015.
- [60] 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, VALUE-TOOLS 2014, Bratislava, Slovakia, December 9-11, 2014*. ICST, 2014.
- [61] 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.
- [62] 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.
- [63] 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.
- [64] 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.

- [65] 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.
- [66] Mikhail Sheremet, Frank Wolter, and Michael Zakharyashev. A modal logic framework for reasoning about comparative distances and topology. *Ann. Pure Appl. Logic*, 161(4):534–559, 2010.
- [67] Rebecca L. Siegel, Kimberly D. Miller, and Ahmedin Jemal. Cancer statistics, 2016. *CA: A Cancer Journal for Clinicians*, 66(1):7–30, 2016.
- [68] 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.
- [69] G.N. Srinivasan and G. Shobha. Statistical texture analysis. In *Proceedings of World Academy of Science, Engineering and Technology*, volume 36, pages 1264–1269, dec 2012.
- [70] 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.
- [71] 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.
- [72] Mikkel Thorup. Undirected single-source shortest paths with positive integer weights in linear time. *J. ACM*, 46(3):362–394, May 1999.
- [73] 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.
- [74] 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.
- [75] J. van Benthem and G. Bezhanishvili. Modal logics of space. In *Handbook of Spatial Logics*, pages 217–298. Springer, 2007.
- [76] Johan van Benthem and Guram Bezhanishvili. Modal logics of space. In *Handbook of Spatial Logics*, pages 217–298. Springer, 2007.
- [77] 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.

- [78] 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.