2009

TR-2009007: History Tree Descriptors of Grayscale Images

Gabor T. Herman

T. Yung Kong

Lucas M. Oliveira

History Tree Descriptors of Grayscale Images

Gabor T. Herman[1], T. Yung Kong[2], and Lucas M. Oliveira[3]

[1] Computer Science Ph.D. Program, Graduate Center, City University of New York
gabortherman@yahoo.com
WWW home page: http://www.dig.cs.gc.cuny.edu/~gabor/

[2] Computer Science Department, Queens College, City University of New York
ykong@cs.qc.cuny.edu

[3] Computer Science Ph.D. Program, Graduate Center, City University of New York
lmoliveira@gmail.com

Abstract. We are interested in translating three-dimensional arrays of real numbers (*images*) into simpler structures that nevertheless capture the topological/geometrical essence of the objects in the images. These structures are to be used as descriptors of images in databases. A *foreground history tree structure* (*FHTS*) contains all the information on the relationships between connected components when the image is thresholded at various levels. Unsimplified FHTSs are too sensitive to errors in the image to be good descriptors. We present a method of simplifying FHTSs, which can be shown to be robust in the sense of producing essentially the same simplifications in the presence of small perturbations. We demonstrate the potential applicability of our methodology to macromolecular databases by showing that the simplified FHTSs can be used to distinguish between two slightly different versions of an adenovirus.

Key words: three-dimensional image, topological descriptor, database, foreground history tree structure, FHTS, macromolecule

1 Introduction and Motivation

High-level structural information about macromolecules is now being organized into databases. These include reconstructions from electron microscopic data; i.e., three-dimensional arrays of real numbers (*images*) that are voxelizations of macromolecular structures. The large size of these arrays, the arbitrary position and orientation of the molecule in the array, and the possibility of non-linear stretching of the range make standard methods of comparison between database entries infeasible. We propose the use of simple descriptors that capture the topological/geometrical essence of the macromolecules in the images for the exploration of such databases. We believe that these descriptors will be useful in the identification and classification of macromolecules. In this paper we define our descriptors, describe efficient computer algorithms for producing them, mathematically investigate their robustness, and provide a sample biological application in which they are used to differentiate two versions of an adenovirus.

2 Foreground History Tree Structures

2.1 Adjacency Relations, Spels, and Images

We use the term *adjacency relation* to mean an irreflexive symmetric binary relation (i.e., a set $\boldsymbol{\kappa}$ of ordered pairs such that if $(a, b) \in \boldsymbol{\kappa}$ then $a \neq b$ and $(b, a) \in \boldsymbol{\kappa}$). The members of the pairs that belong to any adjacency relation we are using will be called *spels*. (As in, e.g., [2], "spel" is an abbreviation of "spatial element", and we think of spels as generalizations of pixels and voxels.) We use the term *grayscale image* or, more briefly, the term *image*, to mean a real-valued function whose domain is a nonempty set of spels.

In the practical work described in Section 4, we use the "6-adjacency" relation [2, p. 16] on $\mathbb{Z}^3$ as our adjacency relation, and use grayscale images whose domain is the finite set $\{(x, y, z) \in \mathbb{Z}^3 \mid 0 \leq x \leq 274, 0 \leq y \leq 274, 0 \leq z \leq 274\}$.

Let $\boldsymbol{\kappa}$ be an adjacency relation. We say that two disjoint sets of spels $\mathcal{S}_1$ and $\mathcal{S}_2$ are *$\boldsymbol{\kappa}$-adjacent* if there exist $s_1 \in \mathcal{S}_1$ and $s_2 \in \mathcal{S}_2$ such that $(s_1, s_2) \in \boldsymbol{\kappa}$. We call a sequence $s_0, \ldots, s_l$ of $l + 1$ spels a *$\boldsymbol{\kappa}$-path* if $l = 0$ or if $l \geq 1$ and $(s_i, s_{i+1}) \in \boldsymbol{\kappa}$ for $0 \leq i < l$. We say that a set $\mathcal{S}$ is *$\boldsymbol{\kappa}$-connected* if for all $s, s' \in \mathcal{S}$ there exists a $\boldsymbol{\kappa}$-path $s_0, \ldots, s_l$ such that $s_0 = s$, $s_l = s'$, and $s_i \in \mathcal{S}$ for $0 \leq i \leq l$.

Let $\mathcal{I} : \mathcal{S} \to \mathbb{R}$ be any image, let $\tau \in \mathbb{R}$, and let $s \in \mathcal{S}$. Then $\mathcal{C}_{\boldsymbol{\kappa}}(s, \mathcal{I}, \tau)$ will denote the set of all $s' \in \mathcal{S}$ for which there exists a $\boldsymbol{\kappa}$-path $s_0, \ldots, s_l$ such that $s_0 = s$, $s_l = s'$, and $\mathcal{I}(s_i) \geq \tau$ for $0 \leq i \leq l$. Note that $\mathcal{C}_{\boldsymbol{\kappa}}(s, \mathcal{I}, \tau) = \emptyset$ if $\tau > \mathcal{I}(s)$, and $s \in \mathcal{C}_{\boldsymbol{\kappa}}(s, \mathcal{I}, \tau)$ if $\tau \leq \mathcal{I}(s)$. We write $\mathcal{C}_{\boldsymbol{\kappa}}(s, \mathcal{I})$ to denote the set $\mathcal{C}_{\boldsymbol{\kappa}}(s, \mathcal{I}, \mathcal{I}(s))$. Readily, if $t \in \mathcal{C}_{\boldsymbol{\kappa}}(s, \mathcal{I})$, then $\mathcal{I}(t) \geq \mathcal{I}(s)$ and either $\mathcal{C}_{\boldsymbol{\kappa}}(t, \mathcal{I}) = \mathcal{C}_{\boldsymbol{\kappa}}(s, \mathcal{I})$ or $\mathcal{C}_{\boldsymbol{\kappa}}(t, \mathcal{I}) \subsetneq \mathcal{C}_{\boldsymbol{\kappa}}(s, \mathcal{I})$ according to whether $\mathcal{I}(t) = \mathcal{I}(s)$ or $\mathcal{I}(t) > \mathcal{I}(s)$.

2.2 Some Terminology and Notation Associated with Rooted Trees

Let $\trianglelefteq$ be any partial order on a finite set $\mathbf{V}$. Then we will say that $\trianglelefteq$ is *treelike* if for each $\mathbf{v} \in \mathbf{V}$ the restriction of $\trianglelefteq$ to $\{\mathbf{u} \in \mathbf{V} \mid \mathbf{u} \trianglelefteq \mathbf{v}\}$ is a total order, and, in addition, there exists an element $\mathbf{r} \in \mathbf{V}$ such that $\mathbf{r} \trianglelefteq \mathbf{v}$ for all $\mathbf{v} \in \mathbf{V}$.

We assume the reader is familiar with the concept of a rooted tree (as defined in, e.g., [1, Appendix B.5.2]). Let $\boldsymbol{\mathcal{T}}$ be any rooted tree. We write $\mathbf{Nodes}(\boldsymbol{\mathcal{T}})$ to denote the (finite) set of all nodes of $\boldsymbol{\mathcal{T}}$, write $\mathbf{root}(\boldsymbol{\mathcal{T}})$ to denote the root of $\boldsymbol{\mathcal{T}}$, and write $\mathbf{Leaves}(\boldsymbol{\mathcal{T}})$ to denote the set of all leaves of $\boldsymbol{\mathcal{T}}$. For $\mathbf{v} \in \mathbf{Nodes}(\boldsymbol{\mathcal{T}})$, we write $\mathbf{Children}_{\boldsymbol{\mathcal{T}}}(\mathbf{v})$ to denote the set of all the children of $\mathbf{v}$ in $\boldsymbol{\mathcal{T}}$, and if $\mathbf{v} \neq \mathbf{root}(\boldsymbol{\mathcal{T}})$ then we write $\mathbf{parent}_{\boldsymbol{\mathcal{T}}}(\mathbf{v})$ to denote the parent of $\mathbf{v}$ in $\boldsymbol{\mathcal{T}}$; we write $\boldsymbol{\mathcal{T}}[\mathbf{v}]$ to denote the subtree of $\boldsymbol{\mathcal{T}}$ that is rooted at $\mathbf{v}$.

Recall that if $\mathbf{x} \in \mathbf{Nodes}(\boldsymbol{\mathcal{T}})$ and $\mathbf{y} \in \mathbf{Nodes}(\boldsymbol{\mathcal{T}}[\mathbf{x}])$ then $\mathbf{x}$ is said to be an *ancestor* of $\mathbf{y}$ in $\boldsymbol{\mathcal{T}}$, and $\mathbf{y}$ is said to be a *descendant* of $\mathbf{x}$ in $\boldsymbol{\mathcal{T}}$. We write $\mathbf{x} \preceq_{\boldsymbol{\mathcal{T}}} \mathbf{y}$ or $\mathbf{y} \succeq_{\boldsymbol{\mathcal{T}}} \mathbf{x}$ to mean that $\mathbf{x}, \mathbf{y} \in \mathbf{Nodes}(\boldsymbol{\mathcal{T}})$ and $\mathbf{x}$ is an ancestor of $\mathbf{y}$ in $\boldsymbol{\mathcal{T}}$. We write $\mathbf{x} \prec_{\boldsymbol{\mathcal{T}}} \mathbf{y}$ or $\mathbf{y} \succ_{\boldsymbol{\mathcal{T}}} \mathbf{x}$ to mean that $\mathbf{x} \preceq_{\boldsymbol{\mathcal{T}}} \mathbf{y}$ but $\mathbf{x} \neq \mathbf{y}$. If $\mathbf{x} \prec_{\boldsymbol{\mathcal{T}}} \mathbf{y}$ then $\mathbf{x}$ is said to be a *proper* ancestor of $\mathbf{y}$ in $\boldsymbol{\mathcal{T}}$, and $\mathbf{y}$ a *proper* descendant of $\mathbf{x}$ in $\boldsymbol{\mathcal{T}}$. Evidently, $\preceq_{\boldsymbol{\mathcal{T}}}$ is a treelike partial order on $\mathbf{Nodes}(\boldsymbol{\mathcal{T}})$. It is also easy to verify that if $\trianglelefteq$ is any treelike partial order on a finite set $\mathbf{V}$, then there exists a rooted tree $\boldsymbol{\mathcal{T}}_{\trianglelefteq}$ such that $\mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\trianglelefteq}) = \mathbf{V}$ and the partial order $\preceq_{\boldsymbol{\mathcal{T}}_{\trianglelefteq}}$ is $\trianglelefteq$.

We write $\mathbf{x}{\Downarrow_{\mathcal{T}}}$ to denote the set $\{\mathbf{y} \in \mathbf{Nodes}(\mathcal{T}) \mid \mathbf{y} \preceq_{\mathcal{T}} \mathbf{x}\}$ and write $\mathbf{x}{\downarrow_{\mathcal{T}}}$ to denote the set $\mathbf{x}{\Downarrow_{\mathcal{T}}} \setminus \{\mathbf{x}\}$. We similarly write $\mathbf{x}{\Uparrow_{\mathcal{T}}}$ to denote the set $\{\mathbf{y} \in \mathbf{Nodes}(\mathcal{T}) \mid \mathbf{y} \succeq_{\mathcal{T}} \mathbf{x}\}$ and write $\mathbf{x}{\uparrow_{\mathcal{T}}}$ to denote the set $\mathbf{x}{\Uparrow_{\mathcal{T}}} \setminus \{\mathbf{x}\}$.

If $\emptyset \neq \mathbf{S} \subseteq \mathbf{Nodes}(\mathcal{T})$ then we write $\bigwedge_{\mathcal{T}} \mathbf{S}$ to denote the *closest common ancestor* of $\mathbf{S}$, by which we mean the node $\mathbf{v}$ of $\mathcal{T}$ such that $\mathbf{v}{\Downarrow_{\mathcal{T}}} = \bigcap_{\mathbf{u} \in \mathbf{S}} \mathbf{u}{\Downarrow_{\mathcal{T}}}$.

A node $\mathbf{v}$ of $\mathcal{T}$ is said to be *critical* if $|\mathbf{Children}_{\mathcal{T}}(\mathbf{v})| \neq 1$; thus $\mathbf{v}$ is a critical node if and only if either $\mathbf{v} \in \mathbf{Leaves}(\mathcal{T})$ or $|\mathbf{Children}_{\mathcal{T}}(\mathbf{v})| \geq 2$. We write $\mathbf{Crit}(\mathcal{T})$ to denote the set of all critical nodes of $\mathcal{T}$. We write $\mathcal{T}^{\mathbf{crit}}$ to denote the rooted tree whose set of nodes is $\mathbf{Crit}(\mathcal{T})$, in which the ancestors of each node $\mathbf{c} \in \mathbf{Crit}(\mathcal{T})$ are just the critical ancestors of $\mathbf{c}$ in $\mathcal{T}$. We write $\mathbf{LCN}(\mathcal{T})$ to denote the *lowest critical node* of $\mathcal{T}$, by which we mean the root of $\mathcal{T}^{\mathbf{crit}}$. If $\mathbf{v} \in \mathbf{Nodes}(\mathcal{T})$ then we say $\mathbf{a}$ is the *immediate proper critical ancestor* of $\mathbf{v}$ in $\mathcal{T}$ if $\mathbf{a} \in \mathbf{Crit}(\mathcal{T})$, $\mathbf{a} \prec_{\mathcal{T}} \mathbf{v}$, and $\{\mathbf{c} \in \mathbf{Crit}(\mathcal{T}) \mid \mathbf{a} \prec_{\mathcal{T}} \mathbf{c} \prec_{\mathcal{T}} \mathbf{v}\} = \emptyset$. Readily, each $\mathbf{v} \in \mathbf{LCN}(\mathcal{T}){\uparrow_{\mathcal{T}}}$ has a unique immediate proper critical ancestor, which we denote by $\mathbf{IPCA}_{\mathcal{T}}(\mathbf{v})$.

2.3 Definition of a $\boldsymbol{\kappa}$-Foreground History Tree Structure ($\boldsymbol{\kappa}$-FHTS); Essential Isomorphism

Let $\boldsymbol{\kappa}$ be any adjacency relation. Then a ***κ**-foreground history tree structure* or ***κ**-FHTS* is a pair $(\mathcal{T}, \ell)$ for which there exists a collection $\mathcal{C}$ of nonempty finite $\boldsymbol{\kappa}$-connected sets of spels such that:

1. $\bigcup \mathcal{C} \in \mathcal{C}$ (i.e., $\mathcal{C}$ has an element that is a superset of every element of $\mathcal{C}$).
2. For all $\mathbf{u}, \mathbf{v} \in \mathcal{C}$, if $\mathbf{u} \not\supseteq \mathbf{v}$ and $\mathbf{v} \not\supseteq \mathbf{u}$ then the sets $\mathbf{u}$ and $\mathbf{v}$ are disjoint and are not $\boldsymbol{\kappa}$-adjacent.
3. ℓ is a real-valued function on $\mathcal{C}$ such that, for all $\mathbf{u}, \mathbf{v} \in \mathcal{C}$, $\ell(\mathbf{u}) < \ell(\mathbf{v})$ whenever $\mathbf{u} \supsetneq \mathbf{v}$.
4. $\mathcal{T}$ is the rooted tree such that $\mathbf{Nodes}(\mathcal{T}) = \mathcal{C}$ and, for all $\mathbf{u}, \mathbf{v} \in \mathcal{C}$, $\mathbf{u} \prec_{\mathcal{T}} \mathbf{v}$ if and only if $\mathbf{u} \supsetneq \mathbf{v}$.

Condition 1 implies that $\bigcup \mathcal{C}$ is a finite $\boldsymbol{\kappa}$-connected set; since $\bigcup \mathcal{C}$ is finite, $\mathcal{C}$ is a finite collection. Moreover, conditions 1 and 2 imply that the restriction of $\supseteq$ to $\mathcal{C}$ is a treelike partial order. It follows that if $\mathcal{C}$ is *any* collection of nonempty finite $\boldsymbol{\kappa}$-connected sets that satisfies conditions 1 and 2, and ℓ any function that satisfies condition 3, then there will exist a unique $\boldsymbol{\kappa}$-FHTS $(\mathcal{T}, \ell)$ for which $\mathbf{Nodes}(\mathcal{T}) = \mathcal{C}$ and $\mathbf{root}(\mathcal{T}) = \bigcup \mathcal{C}$.

If each of $\mathfrak{F} = (\mathcal{T}, \ell)$ and $\mathfrak{F}' = (\mathcal{T}', \ell')$ is a $\boldsymbol{\kappa}$-FHTS, then we write $\mathfrak{F}' \sqsubseteq \mathfrak{F}$ to mean that $\mathbf{Nodes}(\mathcal{T}') \subseteq \mathbf{Nodes}(\mathcal{T})$ and ℓ' is the restriction of ℓ to $\mathbf{Nodes}(\mathcal{T}')$. For any $\boldsymbol{\kappa}$-FHTS $(\mathcal{T}, \ell)$ and any $\mathbf{V} \subseteq \mathbf{Nodes}(\mathcal{T})$ such that $\mathbf{LCN}(\mathcal{T}){\Downarrow_{\mathcal{T}}} \not\subseteq \mathbf{V}$, there is a $\boldsymbol{\kappa}$-FHTS $(\mathcal{T}', \ell') \sqsubseteq (\mathcal{T}, \ell)$ such that $\mathbf{Nodes}(\mathcal{T}') = \mathbf{Nodes}(\mathcal{T}) \setminus \mathbf{V}$; this $\boldsymbol{\kappa}$-FHTS $(\mathcal{T}', \ell')$ will be denoted by $(\mathcal{T}, \ell) - \mathbf{V}$.

We write $(\mathcal{T}, \ell)^{\mathbf{crit}}$ to denote $(\mathcal{T}, \ell) - (\mathbf{Nodes}(\mathcal{T}) \setminus \mathbf{Crit}(\mathcal{T}))$. In other words, $(\mathcal{T}, \ell)^{\mathbf{crit}} = (\mathcal{T}^{\mathbf{crit}}, \ell^{\mathbf{crit}})$, where $\ell^{\mathbf{crit}}$ is the restriction of ℓ to $\mathbf{Crit}(\mathcal{T})$.

We define $\mathrm{depth}_{(\mathcal{T},\ell)}(\mathbf{v}) = (\max_{\mathbf{y} \in \mathbf{Leaves}(\mathcal{T}[\mathbf{v}])} \ell(\mathbf{y})) - \ell(\mathbf{v})$ for every node $\mathbf{v}$ of $\mathcal{T}$. Note that $\mathrm{depth}_{(\mathcal{T},\ell)^{\mathbf{crit}}}(\mathbf{c}) = \mathrm{depth}_{(\mathcal{T},\ell)}(\mathbf{c})$ for all $\mathbf{c} \in \mathbf{Crit}(\mathcal{T})$.

We say that two $\boldsymbol{\kappa}$-FHTSs $\mathfrak{F}_1 = (\boldsymbol{\mathcal{T}}_1, \ell_1)$ and $\mathfrak{F}_2 = (\boldsymbol{\mathcal{T}}_2, \ell_2)$ are *essentially isomorphic* if $\boldsymbol{\mathcal{T}}_1^{\mathbf{crit}}$ and $\boldsymbol{\mathcal{T}}_2^{\mathbf{crit}}$ are isomorphic trees. Thus $\mathfrak{F}_1$ and $\mathfrak{F}_2$ are essentially isomorphic if and only if there exists a mapping $\theta : \mathbf{Crit}(\boldsymbol{\mathcal{T}}_1) \to \mathbf{Crit}(\boldsymbol{\mathcal{T}}_2)$ such that $\theta[\mathbf{Crit}(\boldsymbol{\mathcal{T}}_1)] = \mathbf{Crit}(\boldsymbol{\mathcal{T}}_2)$ and, for all $\mathbf{c}, \mathbf{c}' \in \mathbf{Crit}(\boldsymbol{\mathcal{T}}_1)$, $\mathbf{c} \preceq_{\boldsymbol{\mathcal{T}}_1} \mathbf{c}'$ if and only if $\theta(\mathbf{c}) \preceq_{\boldsymbol{\mathcal{T}}_2} \theta(\mathbf{c}')$. (The latter property implies that θ is 1-to-1.) Any such θ will be called an *essential isomorphism* of $\mathfrak{F}_1$ to $\mathfrak{F}_2$; if θ also satisfies $\ell_1(\mathbf{x}) = \ell_2(\theta(\mathbf{x}))$ for all $\mathbf{x} \in \mathbf{Crit}(\boldsymbol{\mathcal{T}}_1)$, then we say that θ is *level-preserving*.

2.4 A Bijective Correspondence Between Images with Finite $\boldsymbol{\kappa}$-Connected Domains and $\boldsymbol{\kappa}$-FHTSs

Let $\mathcal{S}$ be any nonempty finite $\boldsymbol{\kappa}$-connected set of spels. Then we associate each image $\mathcal{I} : \mathcal{S} \to \mathbb{R}$ with the $\boldsymbol{\kappa}$-foreground history tree structure $\mathbf{FHTS}_{\boldsymbol{\kappa}}(\mathcal{I})$ that is defined by $\mathbf{FHTS}_{\boldsymbol{\kappa}}(\mathcal{I}) = (\boldsymbol{\mathcal{T}}_{\mathcal{I}}, \ell_{\mathcal{I}})$, where:

(a) $\mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\mathcal{I}}) = \{\mathcal{C}_{\boldsymbol{\kappa}}(s, \mathcal{I}) \mid s \in \mathcal{S}\}$ and, for all $\mathbf{u}, \mathbf{v} \in \mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\mathcal{I}})$, we have that $\mathbf{u} \preceq_{\boldsymbol{\mathcal{T}}_{\mathcal{I}}} \mathbf{v}$ if and only if $\mathbf{u} \supseteq \mathbf{v}$.
(b) For all $s \in \mathcal{S}$, $\ell_{\mathcal{I}}(\mathcal{C}_{\boldsymbol{\kappa}}(s, \mathcal{I})) = \mathcal{I}(s)$. ($\ell_{\mathcal{I}}$ is well defined by this condition, because $\mathcal{I}(s) = \mathcal{I}(s')$ whenever $\mathcal{C}_{\boldsymbol{\kappa}}(s, \mathcal{I}) = \mathcal{C}_{\boldsymbol{\kappa}}(s', \mathcal{I})$.)

Note that a $\boldsymbol{\kappa}$-FHTS with these two properties exists, since conditions 1 and 2 in the definition of a $\boldsymbol{\kappa}$-FHTS hold when $\mathcal{C} = \{\mathcal{C}_{\boldsymbol{\kappa}}(s, \mathcal{I}) \mid s \in \mathcal{S}\}$.

Conversely, we associate each $\boldsymbol{\kappa}$-FHTS $\mathfrak{F} = (\boldsymbol{\mathcal{T}}, \ell)$ with the image $\mathcal{I}_{\mathfrak{F}}$ that we now define. For each spel $s \in \mathbf{root}(\boldsymbol{\mathcal{T}})$, conditions 2 and 4 imply that, among the elements of $\mathbf{Nodes}(\boldsymbol{\mathcal{T}})$ that contain s, there must be a smallest (i.e., a node that is a descendant in $\boldsymbol{\mathcal{T}}$ of every node that contains s); that element will be denoted by $\mathbf{node}_{\boldsymbol{\mathcal{T}}}(s)$. We define $\mathcal{I}_{\mathfrak{F}}$ to be the image whose domain is $\mathbf{root}(\boldsymbol{\mathcal{T}})$, and which satisfies $\mathcal{I}_{\mathfrak{F}}(s) = \ell(\mathbf{node}_{\boldsymbol{\mathcal{T}}}(s))$ for all $s \in \mathbf{root}(\boldsymbol{\mathcal{T}})$.

It is not difficult to prove that $\mathcal{I}_{\mathbf{FHTS}_{\boldsymbol{\kappa}}(\mathcal{I})} = \mathcal{I}$ for any image $\mathcal{I}$ whose domain is finite and $\boldsymbol{\kappa}$-connected, and that $\mathbf{FHTS}_{\boldsymbol{\kappa}}(\mathcal{I}_{\mathfrak{F}}) = \mathfrak{F}$ for every $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}$. Thus $\mathcal{I} \mapsto \mathbf{FHTS}_{\boldsymbol{\kappa}}(\mathcal{I})$ and $\mathfrak{F} \mapsto \mathcal{I}_{\mathfrak{F}}$ are mutually inverse bijections.

With some rather trivial modifications, the algorithm described in [5, pp. 265-7] can be used to construct $\mathbf{FHTS}_{\boldsymbol{\kappa}}(\mathcal{I})$. In the next section we describe how we simplify $\mathbf{FHTS}_{\boldsymbol{\kappa}}(\mathcal{I})$ to produce $\boldsymbol{\kappa}$-FHTSs that are less sensitive to small errors in the image $\mathcal{I}$, and which may therefore be better descriptors of the image.

3 The $(\boldsymbol{\lambda}, \boldsymbol{k})$-Simplification of a $\boldsymbol{\kappa}$-FHTS

If $\mathfrak{F}_0 = (\boldsymbol{\mathcal{T}}_0, \ell_0)$ is any $\boldsymbol{\kappa}$-FHTS, then for every positive real value λ and every nonnegative integer $k < |\mathbf{root}(\boldsymbol{\mathcal{T}}_0)|$ we define the (λ, k)-*simplification* of $\mathfrak{F}_0$ to be the $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_3$ that can be obtained from $\mathfrak{F}_0$ in three steps, as follows:

Step 1: Prune $\mathfrak{F}_0$ by removing nodes of size $\leq k$, to produce $\mathfrak{F}_1 \sqsubseteq \mathfrak{F}_0$.
Step 2: Prune $\mathfrak{F}_1$ by removing branches of length $\leq \lambda$, to produce $\mathfrak{F}_2 \sqsubseteq \mathfrak{F}_1$.
Step 3: Collapse δ-equivalence classes of $\mathfrak{F}_2^{\mathbf{crit}}$ for $\delta \leq \lambda$, to produce the final $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_3 \sqsubseteq \mathfrak{F}_2^{\mathbf{crit}}$.

The final $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_3$ has the property that every node in its tree is critical.

Step 1 is essentially the same as the first step of the simplification method of [5]. It is defined as follows: The result of pruning $\mathfrak{F}_0 = (\boldsymbol{\mathcal{T}}_0, \ell_0)$ by removing nodes of size $\leq k$ is $\mathfrak{F}_0 - \{\mathbf{v} \in \mathbf{Nodes}(\boldsymbol{\mathcal{T}}_0) \mid |\mathbf{v}| \leq k\}$. (As usual, $|\mathbf{v}|$ denotes the cardinality of the set $\mathbf{v}$—i.e., the number of spels in $\mathbf{v}$.) Note that the result is just $\mathfrak{F}_0$ itself if $k = 0$. Steps 2 and 3 will be defined in Subsections 3.1 and 3.2.

While our simplification method is somewhat similar to the method of [5], it has the robustness property that is stated in Theorem 1 below (which the method of [5] does not have). A rough statement of this property is that if $\mathcal{I}$ is any image, and $\mathcal{I}'$ is any image that is sufficiently close to $\mathcal{I}$, then our simplification method can be used to reduce $\mathbf{FHTS}_{\boldsymbol{\kappa}}(\mathcal{I}')$ to a $\boldsymbol{\kappa}$-FHTS that is essentially isomorphic to $\mathbf{FHTS}_{\boldsymbol{\kappa}}(\mathcal{I})$. We now introduce some terminology that will be used in our precise statement of this result.

Let $\mathcal{I}$ be any image whose domain is a finite $\boldsymbol{\kappa}$-connected set $\mathcal{S}$ of spels, and let $(\boldsymbol{\mathcal{T}}, \ell) = \mathbf{FHTS}_{\boldsymbol{\kappa}}(\mathcal{I})$. Then, for any $\lambda > 0$ and any integer $k \geq 0$, we say that the image $\mathcal{I}$ is (λ, k)-*good* if the following conditions both hold:

(a) There is no $s \in \mathcal{S}$ for which $|\mathcal{C}_{\boldsymbol{\kappa}}(s, \mathcal{I})| \leq k$.
(b) There are no $s, t \in \mathcal{S}$ for which $\mathcal{C}_{\boldsymbol{\kappa}}(s, \mathcal{I}), \mathcal{C}_{\boldsymbol{\kappa}}(t, \mathcal{I}) \in \mathbf{Crit}(\boldsymbol{\mathcal{T}})$, $t \in \mathcal{C}_{\boldsymbol{\kappa}}(s, \mathcal{I})$, and $\mathcal{I}(s) < \mathcal{I}(t) \leq \mathcal{I}(s) + \lambda$. (Equivalently, there are no $\mathbf{u}, \mathbf{v} \in \mathbf{Crit}(\boldsymbol{\mathcal{T}})$ for which $\mathbf{u} \succ_{\boldsymbol{\mathcal{T}}} \mathbf{v}$ and $\ell(\mathbf{u}) \leq \ell(\mathbf{v}) + \lambda$.)

$\mathcal{I}$ is sure to be (λ, k)-good for all sufficiently small positive λ and $k = 0$. If $\mathcal{I}$ is (λ, k)-good, then $\mathcal{I}$ is (λ', k')-good whenever $0 < \lambda' \leq \lambda$ and $0 \leq k' \leq k$.

If $\mathcal{I}$ is (λ, k)-good, then the (λ, k)-simplification of $\mathbf{FHTS}_{\boldsymbol{\kappa}}(\mathcal{I})$ is just $\mathbf{FHTS}_{\boldsymbol{\kappa}}(\mathcal{I})^{\mathbf{crit}}$. Moreover, if the $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_{\text{simp}} = (\boldsymbol{\mathcal{T}}_{\text{simp}}, \ell_{\text{simp}})$ is the (λ, k)-simplification of a $\boldsymbol{\kappa}$-FHTS, and $\mathcal{I} = \mathcal{I}_{\mathfrak{F}_{\text{simp}}}$, then $\mathcal{I}$ is (λ, k)-good and each node of $\boldsymbol{\mathcal{T}}_{\text{simp}}$ is critical, so that $\mathcal{I}$ satisfies the following much stronger version of (b):

(b′) There are no $s, t \in \mathcal{S}$ for which $t \in \mathcal{C}_{\boldsymbol{\kappa}}(s, \mathcal{I})$ and $\mathcal{I}(s) < \mathcal{I}(t) \leq \mathcal{I}(s) + \lambda$.

For any image $\mathcal{I} : \mathcal{S} \to \mathbb{R}$ and any $\epsilon > 0$, we use the term ϵ-*perturbation* of $\mathcal{I}$ to mean an image $\mathcal{I}' : \mathcal{S} \to \mathbb{R}$ such that $|\mathcal{I}(x) - \mathcal{I}'(x)| \leq \epsilon$ for all $x \in \mathcal{S}$.

We now state the above-mentioned robustness property of our method:

Theorem 1. *Let $\boldsymbol{\kappa}$ be any adjacency relation and $\mathcal{I}$ any grayscale image whose domain is finite and $\boldsymbol{\kappa}$-connected. Let $\epsilon > 0$ be a real value and $k \geq 0$ an integer such that the image $\mathcal{I}$ is $(4\epsilon, k)$-good, and let $\mathcal{I}'$ be any ϵ-perturbation of $\mathcal{I}$. Then the $(2\epsilon, k)$-simplification of $\mathbf{FHTS}_{\boldsymbol{\kappa}}(\mathcal{I}')$ is essentially isomorphic to $\mathbf{FHTS}_{\boldsymbol{\kappa}}(\mathcal{I})$.*

Our simplification method was in fact designed with Theorem 1 in mind; it was one of our main design goals to make this theorem true. We will outline a proof of the theorem in Subsection 3.3. But first we give a precise definition of Steps 2 and 3 of (λ, k)-simplification. The next two subsections will do this.

3.1 Pruning by Removing Branches of Length $\leq \lambda$

For any $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_{\text{in}} = (\boldsymbol{\mathcal{T}}_{\text{in}}, \ell_{\text{in}})$ and any $\lambda > 0$, the result of pruning $\mathfrak{F}_{\text{in}}$ by removing branches of length $\leq \lambda$ is the $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_{\text{out}} = (\boldsymbol{\mathcal{T}}_{\text{out}}, \ell_{\text{out}}) \sqsubseteq \mathfrak{F}_{\text{in}}$

that is produced by Algorithm 1 below. Although $\mathfrak{F}_{\text{out}}$ depends on the sequence $\mathsf{leaf}[1], \ldots, \mathsf{leaf}[\mathsf{n}]$, which is not uniquely determined by $\mathfrak{F}_{\text{in}}$ if there are distinct leaves $\mathbf{x}$ and $\mathbf{y}$ of $\boldsymbol{\mathcal{T}}_{\text{in}}$ for which $\ell_{\text{in}}(\mathbf{x}) = \ell_{\text{in}}(\mathbf{y})$, we will see that $\mathfrak{F}_{\text{out}}$ is uniquely determined by $\mathfrak{F}_{\text{in}}$ and λ *up to a level-preserving essential isomorphism.*

Algorithm 1: Prune by Removing Branches of Length $\leq \lambda$

inputs : a $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_{\text{in}} = (\boldsymbol{\mathcal{T}}_{\text{in}}, \ell_{\text{in}})$; a positive real value λ
output: a $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_{\text{out}} = (\boldsymbol{\mathcal{T}}_{\text{out}}, \ell_{\text{out}})$

begin
  $\mathsf{n} \longleftarrow |\,\mathbf{Leaves}(\boldsymbol{\mathcal{T}}_{\text{in}})|$;
  Sort $\mathbf{Leaves}(\boldsymbol{\mathcal{T}}_{\text{in}})$ into a sequence $\mathsf{leaf}[1], \ldots, \mathsf{leaf}[\mathsf{n}]$
    such that $\ell(\mathsf{leaf}[1]) \leq \cdots \leq \ell(\mathsf{leaf}[\mathsf{n}])$;
  $(\boldsymbol{\mathcal{T}}, \ell) \longleftarrow \mathfrak{F}_{\text{in}}$;
  for $\mathsf{j} \longleftarrow 1$ **to** $\mathsf{n} - 1$ **do**
    $\mathsf{a} \longleftarrow \mathbf{IPCA}_{\boldsymbol{\mathcal{T}}}(\mathsf{leaf}[\mathsf{j}])$;
    if $\ell(\mathsf{leaf}[\mathsf{j}]) - \ell(\mathsf{a}) \leq \lambda$ **then** $(\boldsymbol{\mathcal{T}}, \ell) \longleftarrow (\boldsymbol{\mathcal{T}}, \ell) - (\mathsf{leaf}[\mathsf{j}]{\Downarrow_{\boldsymbol{\mathcal{T}}}} \cap \mathsf{a}{\uparrow_{\boldsymbol{\mathcal{T}}}})$;
  $\mathfrak{F}_{\text{out}} \longleftarrow (\boldsymbol{\mathcal{T}}, \ell)$;
end

For $2 \leq i \leq |\,\mathbf{Leaves}(\boldsymbol{\mathcal{T}}_{\text{in}})|$, let $\mathfrak{F}_i$ and $\boldsymbol{\mathcal{T}}_i$ respectively denote the $\boldsymbol{\kappa}$-FHTS $(\boldsymbol{\mathcal{T}}, \ell)$ and its tree $\boldsymbol{\mathcal{T}}$ at the end of the $i - 1^{\text{st}}$ iteration of Algorithm 1's **for** loop—so that $\mathfrak{F}_{\text{out}} = \mathfrak{F}_{|\,\mathbf{Leaves}(\boldsymbol{\mathcal{T}}_{\text{in}})|}$. Let $\mathfrak{F}_1 = \mathfrak{F}_{\text{in}}$ and $\boldsymbol{\mathcal{T}}_1 = \boldsymbol{\mathcal{T}}_{\text{in}}$. It is not hard to show that, for $2 \leq i \leq |\,\mathbf{Leaves}(\boldsymbol{\mathcal{T}}_{\text{in}})|$, $\mathbf{Leaves}(\boldsymbol{\mathcal{T}}_i) \subseteq \mathbf{Leaves}(\boldsymbol{\mathcal{T}}_{i-1})$, $\mathbf{Crit}(\boldsymbol{\mathcal{T}}_i) \subseteq \mathbf{Crit}(\boldsymbol{\mathcal{T}}_{i-1})$, $\text{depth}_{\mathfrak{F}_i}(\mathbf{u}) = \text{depth}_{\mathfrak{F}_{\text{in}}}(\mathbf{u})$ for every $\mathbf{u}$ in $\mathbf{Nodes}(\boldsymbol{\mathcal{T}}_i)$, and, for each $\mathbf{v}$ in $\mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\text{in}}) \setminus \mathbf{Nodes}(\boldsymbol{\mathcal{T}}_i)$, $\mathbf{v}{\Uparrow_{\boldsymbol{\mathcal{T}}_{\text{in}}}} \cap \mathbf{Nodes}(\boldsymbol{\mathcal{T}}_i) = \emptyset$.

For each $\mathbf{v} \in \mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\text{in}})$, we write $\mathsf{lastLeaf}(\mathbf{v})$ to denote the leaf of $\boldsymbol{\mathcal{T}}_{\text{in}}[\mathbf{v}]$ that appears last in the sequence $\mathsf{leaf}[1], \ldots, \mathsf{leaf}[\mathsf{n}]$, and write $\mathsf{Path}(\mathbf{v})$ to denote the set $\{\mathbf{x} \in \mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\text{in}}) \mid \mathbf{v} \preceq_{\boldsymbol{\mathcal{T}}_{\text{in}}} \mathbf{x} \preceq_{\boldsymbol{\mathcal{T}}_{\text{in}}} \mathsf{lastLeaf}(\mathbf{v})\}$. Note that $\mathsf{lastLeaf}(\mathbf{v})$ and $\mathsf{Path}(\mathbf{v})$ might not be uniquely determined by $\boldsymbol{\mathcal{T}}_{\text{in}}$ and λ.

Let $\mathbf{U}^{\lambda}$ denote the set $\{\mathbf{v} \in \mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\text{in}}) \mid \text{depth}_{\mathfrak{F}_{\text{in}}}(\mathbf{v}) > \lambda\}$, and let $\mathbf{V}^{\lambda}$ denote the set $\{\mathbf{v} \in \mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\text{in}}) \setminus \mathbf{U}^{\lambda} \mid \mathbf{v}{\downarrow_{\boldsymbol{\mathcal{T}}_{\text{in}}}} \subseteq \mathbf{U}^{\lambda}\}$. Then $\mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\text{in}}) = \mathbf{U}^{\lambda} \cup \bigcup_{\mathbf{v} \in \mathbf{V}^{\lambda}} \mathbf{v}{\Uparrow_{\boldsymbol{\mathcal{T}}_{\text{in}}}}$ and it is not hard to show that $\mathbf{U}^{\lambda} \subsetneq \mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\text{out}})$. Hence:

$$\mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\text{out}}) = \mathbf{U}^{\lambda} \cup \bigcup_{\mathbf{v} \in \mathbf{V}^{\lambda}} (\mathbf{v}{\Uparrow_{\boldsymbol{\mathcal{T}}_{\text{in}}}} \cap \mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\text{out}})) \tag{1}$$

If $\text{depth}_{\mathfrak{F}_{\text{in}}}(\mathbf{root}(\boldsymbol{\mathcal{T}}_{\text{in}})) > \lambda$, then $\mathbf{root}(\boldsymbol{\mathcal{T}}_{\text{in}}) \in \mathbf{U}^{\lambda}$ and so $\mathbf{root}(\boldsymbol{\mathcal{T}}_{\text{in}}) \notin \mathbf{V}^{\lambda}$; in this case, let $\mathbf{V}_1^{\lambda} = \{\mathbf{v} \in \mathbf{V}^{\lambda} \mid \text{depth}_{\mathfrak{F}_{\text{in}}}(\mathbf{v}) + \ell_{\text{in}}(\mathbf{v}) - \ell_{\text{in}}(\mathbf{parent}_{\boldsymbol{\mathcal{T}}_{\text{in}}}(\mathbf{v})) > \lambda\}$. If $\text{depth}_{\mathfrak{F}_{\text{in}}}(\mathbf{root}(\boldsymbol{\mathcal{T}}_{\text{in}})) \leq \lambda$ (so that $\mathbf{U}^{\lambda} = \emptyset$), let $\mathbf{V}_1^{\lambda} = \{\mathbf{root}(\boldsymbol{\mathcal{T}}_{\text{in}})\} = \mathbf{V}^{\lambda}$. Then it is readily confirmed that $\mathbf{v}{\Uparrow_{\boldsymbol{\mathcal{T}}_{\text{in}}}} \cap \mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\text{out}}) = \mathsf{Path}(\mathbf{v})$ for all $\mathbf{v} \in \mathbf{V}_1^{\lambda}$, and $\mathbf{v}\Uparrow_{\boldsymbol{\mathcal{T}}_{\mathrm{in}}} \cap \mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\mathrm{out}}) = \emptyset$ for all $\mathbf{v} \in \mathbf{V}^{\lambda} \setminus \mathbf{V}_1^{\lambda}$. This and (1) imply that:

$$\mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\mathrm{out}}) = \mathbf{U}^{\lambda} \cup \bigcup_{\mathbf{v} \in \mathbf{V}_1^{\lambda}} \mathsf{Path}(\mathbf{v}) \tag{2}$$

$\mathbf{U}^{\lambda}$ is determined by $\mathfrak{F}_{\mathrm{in}}$ and λ. For every $\mathbf{v} \in \mathbf{V}_1^{\lambda}$ we see from (2) that no node in $\mathsf{Path}(\mathbf{v}) \setminus \{\mathsf{lastLeaf}(\mathbf{v})\}$ is a critical node of $\boldsymbol{\mathcal{T}}_{\mathrm{out}}$, and it is evident that $\ell_{\mathrm{out}}(\mathsf{lastLeaf}(\mathbf{v})) = \ell_{\mathrm{in}}(\mathsf{lastLeaf}(\mathbf{v}))$ is determined by $\mathfrak{F}_{\mathrm{in}}$. Hence $\mathfrak{F}_{\mathrm{out}}$ is uniquely determined up to a level-preserving essential isomorphism by $\mathfrak{F}_{\mathrm{in}}$ and λ.

The value of $\mathrm{depth}_{\mathfrak{F}_{\mathrm{in}}}(\mathbf{v})$ can be computed for every node $\mathbf{v}$ of $\boldsymbol{\mathcal{T}}_{\mathrm{in}}$ during a single bottom-up traversal of the tree. Thus (2) provides the basis for a somewhat more efficient algorithm whose output is the same as that of Algorithm 1 up to a level-preserving essential isomorphism.

3.2 Collapsing δ-Equivalence Classes for $\delta \leq \lambda$

For any $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_{\mathrm{in}} = (\boldsymbol{\mathcal{T}}_{\mathrm{in}}, \ell_{\mathrm{in}})$ and any $\lambda > 0$, the result of collapsing δ-equivalence classes of $\mathfrak{F}_{\mathrm{in}}$ for $\delta \leq \lambda$ is defined to be the $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_{\mathrm{out}} = (\boldsymbol{\mathcal{T}}_{\mathrm{out}}, \ell_{\mathrm{out}}) \sqsubseteq \mathfrak{F}_{\mathrm{in}}$ that is produced by Algorithm 2 below.

Note that $\mathbf{root}(\boldsymbol{\mathcal{T}})$ does not change at any iteration of Algorithm 2's **for** loop. Also note that $\min\{\ell(\mathbf{v}) - \ell(\mathbf{parent}_{\boldsymbol{\mathcal{T}}}(\mathbf{v})) \mid \mathbf{v} \in \mathbf{root}(\boldsymbol{\mathcal{T}})\uparrow_{\boldsymbol{\mathcal{T}}}\} \geq \delta[\mathsf{j}]$ at the start of each iteration of the **for** loop. It follows that $\mathbf{root}(\boldsymbol{\mathcal{T}}_{\mathrm{out}}) = \mathbf{root}(\boldsymbol{\mathcal{T}}_{\mathrm{in}})$, and that every non-root node $\mathbf{v}$ of $\boldsymbol{\mathcal{T}}_{\mathrm{out}}$ satisfies $\ell_{\mathrm{out}}(\mathbf{v}) - \ell_{\mathrm{out}}(\mathbf{parent}_{\boldsymbol{\mathcal{T}}_{\mathrm{out}}}(\mathbf{v})) > \lambda$.

In Step 3 of (λ, k)-simplification, collapsing δ-equivalence classes for $\delta \leq \lambda$ is done with $\mathfrak{F}_2^{\mathbf{crit}}$ as the input $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_{\mathrm{in}}$, where $\mathfrak{F}_2$ is the result of pruning a $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_1$ by removing branches of length $\leq \lambda$. In this case $\mathbf{Leaves}(\boldsymbol{\mathcal{T}})$ does not change at any iteration of Algorithm 2's **for** loop, and at the end of each iteration $\boldsymbol{\mathcal{T}}$ still has the property that every node is critical. This implies that $\mathbf{Leaves}(\boldsymbol{\mathcal{T}}_{\mathrm{out}}) = \mathbf{Leaves}(\boldsymbol{\mathcal{T}}_{\mathrm{in}})$, and that each node of $\boldsymbol{\mathcal{T}}_{\mathrm{out}}$ is critical.

Algorithm 2: Collapse δ-Equivalence Classes for $\delta \leq \lambda$

inputs : a $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_{\mathrm{in}} = (\boldsymbol{\mathcal{T}}_{\mathrm{in}}, \ell_{\mathrm{in}})$; a positive real value λ
output: a $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_{\mathrm{out}} = (\boldsymbol{\mathcal{T}}_{\mathrm{out}}, \ell_{\mathrm{out}})$

```
begin
    (T, ℓ) ⟵ F_in;
    DiffSet ⟵ {ℓ(y) − ℓ(x) | x ≺_T y and ℓ(y) − ℓ(x) ≤ λ};
    Sort DiffSet into a strictly increasing sequence δ[1] < ··· < δ[|DiffSet|];
    for j ⟵ 1 to |DiffSet| do
        (T, ℓ) ⟵ (T, ℓ) − {v ∈ root(T)↑_T | ℓ(v) − ℓ(parent_T(v)) = δ[j]};
    F_out ⟵ (T, ℓ);
end
```

We now explain why we call this process "collapsing δ-equivalence classes for $\delta \leq \lambda$". For every $\boldsymbol{\kappa}$-FHTS $\mathfrak{F} = (\boldsymbol{\mathcal{T}}_{\mathfrak{F}}, \ell_{\mathfrak{F}})$ and every positive δ, we write $\rho_{\delta}^{\mathfrak{F}}$ to denote the binary relation on $\mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\mathfrak{F}})$ such that $(\mathbf{u}, \mathbf{v}) \in \rho_\delta^{\mathfrak{F}}$ if and only if $\mathbf{u} \preceq_{\boldsymbol{\mathcal{T}}_{\mathfrak{F}}} \mathbf{v}$ and $\ell_{\mathfrak{F}}(\mathbf{v}) - \ell_{\mathfrak{F}}(\mathbf{u}) \leq \delta$, and we define δ-*equivalence* on $\mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\mathfrak{F}})$ to be the equivalence relation that is the symmetric transitive closure of $\rho_\delta^{\mathfrak{F}}$. Let $\mathfrak{F}^0 = (\boldsymbol{\mathcal{T}}^0, \ell^0)$ be the $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_{\text{in}} = (\boldsymbol{\mathcal{T}}_{\text{in}}, \ell_{\text{in}})$ and, for $1 \leq i \leq |\mathsf{DiffSet}|$, let $\mathfrak{F}^i = (\boldsymbol{\mathcal{T}}^i, \ell^i)$ be the $\boldsymbol{\kappa}$-FHTS $(\boldsymbol{\mathcal{T}}, \ell)$ at the end of the i^{th} iteration of Algorithm 2's **for** loop. Then, for $1 \leq i \leq |\mathsf{DiffSet}|$, $\mathbf{Nodes}(\boldsymbol{\mathcal{T}}^i) \subseteq \mathbf{Nodes}(\boldsymbol{\mathcal{T}}^{i-1})$ and it is readily confirmed that, for each equivalence class $\mathbf{C}$ of the $\delta[i]$-equivalence relation on $\mathbf{Nodes}(\boldsymbol{\mathcal{T}}^{i-1})$, the set $\mathbf{Nodes}(\boldsymbol{\mathcal{T}}^i)$ contains just one element of $\mathbf{C}$, namely the element which is an ancestor in $\boldsymbol{\mathcal{T}}^{i-1}$ of every element of $\mathbf{C}$.

When $\mathsf{DiffSet}$ is large, Algorithm 2 is slow; indeed, execution of line 6 involves traversing the tree $\boldsymbol{\mathcal{T}}$, so the **for** loop performs $|\mathsf{DiffSet}|$ tree-traversals. Algorithm 3 is then a much more efficient way to produce the same output $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_{\text{out}}$ from $\mathfrak{F}_{\text{in}}$ for Step 3 of (λ, k)-simplification. Algorithm 3 labels each $\mathbf{v} \in \mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\text{in}}) \setminus \mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\text{out}})$ with the value of $\delta[\mathsf{j}]$ at the iteration of Algorithm 2's **for** loop that removes $\mathbf{v}$, and labels each $\mathbf{v} \in \mathbf{Nodes}(\boldsymbol{\mathcal{T}}_{\text{out}})$ with a value that exceeds λ. It does a top-down traversal of $\boldsymbol{\mathcal{T}}_{\text{in}}$, during which the **repeat** ... **until** loop of the procedure `labelDescendants` is executed once for each proper descendant of the root that is not a leaf, to compute the label of that node from the labels of its proper ancestors. `labelDescendants` gives every leaf of $\boldsymbol{\mathcal{T}}_{\text{in}}$ a label that exceeds λ, beacuse it assumes that no leaf of $\boldsymbol{\mathcal{T}}_{\text{in}}$ should be removed—i.e., it assumes that $\mathbf{Leaves}(\boldsymbol{\mathcal{T}}_{\text{out}}) = \mathbf{Leaves}(\boldsymbol{\mathcal{T}}_{\text{in}})$. As mentioned above, this assumption is correct for Step 3 of (λ, k)-simplification.

Algorithm 3: A Faster Version of Algorithm 2 for Simplification Step 3.

inputs : a $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_{\text{in}} = (\boldsymbol{\mathcal{T}}_{\text{in}}, \ell_{\text{in}})$; a positive real value λ
output: a $\boldsymbol{\kappa}$-FHTS $\mathfrak{F}_{\text{out}} = (\boldsymbol{\mathcal{T}}_{\text{out}}, \ell_{\text{out}})$

begin
  $(\boldsymbol{\mathcal{T}}, \ell) \longleftarrow \mathfrak{F}_{\text{in}}$;
  $\mathbf{root}(\boldsymbol{\mathcal{T}})$.label $\longleftarrow$ some value that exceeds λ;
  foreach c $\in \mathbf{Children}_{\boldsymbol{\mathcal{T}}}(\mathbf{root}(\boldsymbol{\mathcal{T}}))$ **do** `labelDescendants`(c,$\boldsymbol{\mathcal{T}}$,ℓ,λ);
  $\mathfrak{F}_{\text{out}} \longleftarrow (\boldsymbol{\mathcal{T}}, \ell) - \{\mathbf{v} \in \mathbf{Nodes}(\boldsymbol{\mathcal{T}}) \mid \mathbf{v}.\mathsf{label} \leq \lambda\}$;
end

Procedure `labelDescendants`($node, \boldsymbol{\mathcal{T}}, \ell, \lambda$)

if $node \in \mathbf{Leaves}(\boldsymbol{\mathcal{T}})$ **then**
  $node$.label $\longleftarrow$ some value that exceeds λ;
else
  ancestor $\longleftarrow$ $node$;
  repeat
    ancestor $\longleftarrow$ $\mathbf{parent}_{\boldsymbol{\mathcal{T}}}$(ancestor);
    $node$.label $\longleftarrow$ $\ell(node) - \ell$(ancestor);
  until ($node$.label $> \lambda$ **or** $node$.label $\leq$ ancestor.label);
  foreach c $\in \mathbf{Children}_{\boldsymbol{\mathcal{T}}}(node)$ **do** `labelDescendants`(c,$\boldsymbol{\mathcal{T}}$,ℓ,λ);

3.3 A Brief Outline of a Proof of Theorem 1

We now assume the hypotheses of Theorem 1 are satisfied, and explain how the theorem can be proved.

Let $\mathcal{S}$ be the domain of $\mathcal{I}$ and $\mathcal{I}'$. Let $\mathfrak{F} = (\mathcal{T}, \ell) = \mathbf{FHTS}_{\kappa}(\mathcal{I})$ and $\mathfrak{F}' = (\mathcal{T}', \ell') = \mathbf{FHTS}_{\kappa}(\mathcal{I}')$. Let $\mathfrak{F}_1 = (\mathcal{T}_1, \ell_1)$ be the κ-FHTS that results from pruning $\mathfrak{F}'$ by removing nodes of size $\leq k$, let $\mathfrak{F}_2 = (\mathcal{T}_2, \ell_2)$ be the κ-FHTS that results from pruning $\mathfrak{F}_1$ by removing branches of length $\leq 2\epsilon$, and let $\mathfrak{F}_3 = (\mathcal{T}_3, \ell_3)$ be the κ-FHTS that results from collapsing δ-equivalence classes of $\mathfrak{F}_2^{\mathbf{crit}}$ for $\delta \leq 2\epsilon$. Then $\mathfrak{F}_3$ is the $(2\epsilon, k)$-simplification of $\mathbf{FHTS}_{\kappa}(\mathcal{I}')$, and what we need to show is that there exists an essential isomorphism of $\mathfrak{F}$ to $\mathfrak{F}_3$.

Let $\mathcal{I}^* = \mathcal{I}_{\mathfrak{F}_1}$, so that $\mathfrak{F}_1 = \mathbf{FHTS}_{\kappa}(\mathcal{I}^*)$. Readily, $\mathcal{I}^*$ is the image with domain $\mathcal{S}$ such that, for every $x \in \mathcal{S}$, $\mathcal{I}^*(x)$ is the greatest real value τ for which $|\mathcal{C}_{\kappa}(x, \mathcal{I}', \tau)| \geq k + 1$. Thus $\mathcal{I}^* \leq \mathcal{I}'$, and (since $\mathcal{I}$ is $(4\epsilon, k)$-good and $\mathcal{I}'$ is an ϵ-perturbation of $\mathcal{I}$) it is not hard to verify that $\mathcal{I}^*$ is an ϵ-perturbation of $\mathcal{I}$.

Let $\mathbf{v}$ be any leaf of $\mathcal{T}$, and let s be any spel in $\mathbf{v}$ such that $\mathcal{C}_{\kappa}(s, \mathcal{I}) = \mathbf{v}$. Then we define $\mathcal{M}(\mathbf{v}, \mathcal{I}, \mathcal{I}^*)$ to be the set of all spels t in $\mathcal{C}_{\kappa}(s, \mathcal{I}, \mathcal{I}(s) - 2\epsilon)$ such that $\mathcal{I}^*(t) = \max\{\mathcal{I}^*(w) | w \in \mathcal{C}_{\kappa}(s, \mathcal{I}, \mathcal{I}(s) - 2\epsilon)\}$. ($\mathcal{M}(\mathbf{v}, \mathcal{I}, \mathcal{I}^*)$ depends on $\mathbf{v}$, but not on our choice of s in $\mathbf{v}$ such that $\mathcal{C}_{\kappa}(s, \mathcal{I}) = \mathbf{v}$.) As $(\mathcal{T}_1, \ell_1) = \mathfrak{F}_1 = \mathbf{FHTS}_{\kappa}(\mathcal{I}^*)$, we can show (using the facts that $\mathcal{I}$ is $(4\epsilon, k)$-good and $\mathcal{I}^*$ is an ϵ-perturbation of $\mathcal{I}$) that $\mathcal{C}_{\kappa}(t, \mathcal{I}^*)$ is a leaf of $\mathcal{T}_1$ for all t in $\mathcal{M}(\mathbf{v}, \mathcal{I}, \mathcal{I}^*)$, and that, when $\mathfrak{F}_1$ is pruned by removing branches of length $\leq 2\epsilon$ to produce $\mathfrak{F}_2 = (\mathcal{T}_2, \ell_2)$, just one of the leaves in $\{\mathcal{C}_{\kappa}(t, \mathcal{I}^*) \mid t \in \mathcal{M}(\mathbf{v}, \mathcal{I}, \mathcal{I}^*)\}$ is not deleted; the leaf which remains is a leaf of $\mathcal{T}_2$ that we will denote by $\phi(\mathbf{v})$. This defines a mapping $\phi : \mathbf{Leaves}(\mathcal{T}) \to \mathbf{Leaves}(\mathcal{T}_2)$. We can show that ϕ is a bijection onto $\mathbf{Leaves}(\mathcal{T}_2)$, and that $|\ell_2(\bigwedge_{\mathcal{T}_2}(\phi[\mathbf{L}])) - \ell(\bigwedge_{\mathcal{T}} \mathbf{L})| \leq \epsilon$ whenever $\emptyset \neq \mathbf{L} \subseteq \mathbf{Leaves}(\mathcal{T})$ (which implies that, for all nonempty sets $\mathbf{L} \subseteq \mathbf{L}' \subseteq \mathbf{Leaves}(\mathcal{T})$, $\ell_2(\bigwedge_{\mathcal{T}_2}(\phi[\mathbf{L}])) - \ell_2(\bigwedge_{\mathcal{T}_2}(\phi[\mathbf{L}'])) \leq 2\epsilon$ if and only if $\bigwedge_{\mathcal{T}} \mathbf{L}' = \bigwedge_{\mathcal{T}} \mathbf{L}$).

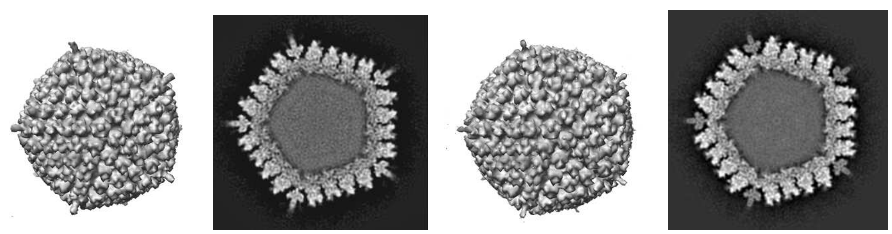
We extend the bijection ϕ to a mapping $\varphi : \mathbf{Crit}(\mathcal{T}) \to \mathbf{Crit}(\mathcal{T}_2)$ by defining $\varphi(\mathbf{u}) = \bigwedge_{\mathcal{T}_2} \phi[\mathbf{Leaves}(\mathcal{T}[\mathbf{u}])]$. Now it is not difficult to establish that: (i) for all $\mathbf{u} \in \mathbf{Crit}(\mathcal{T})$, $\mathbf{Leaves}(\mathcal{T}_2[\varphi(\mathbf{u})]) = \varphi[\mathbf{Leaves}(\mathcal{T}[\mathbf{u}])]$; (ii) for all $\mathbf{x} \in \varphi[\mathbf{Crit}(\mathcal{T})]$, there is no $\mathbf{y} \in \mathbf{x}{\downarrow}_{\mathcal{T}_2} \cap \mathbf{Crit}(\mathcal{T}_2)$ such that $\ell_2(\mathbf{x}) - \ell_2(\mathbf{y}) \leq 2\epsilon$; (iii) for all $\mathbf{x} \in \mathbf{Crit}(\mathcal{T}_2)$, there exists some $\mathbf{z} \in \mathbf{x}{\Downarrow}_{\mathcal{T}_2} \cap \varphi[\mathbf{Crit}(\mathcal{T})]$ such that $\ell_2(\mathbf{x}) - \ell_2(\mathbf{z}) \leq 2\epsilon$. (For all $\mathbf{x} \in \mathbf{Crit}(\mathcal{T}_2)$, $\mathbf{z} = \varphi(\bigwedge_{\mathcal{T}} \varphi^{-1}[\mathbf{Leaves}(\mathcal{T}_2[\mathbf{x}])])$ has the property stated in (iii).) As $\mathfrak{F}_3 = (\mathcal{T}_3, \ell_3)$ is the result of collapsing δ-equivalence classes of $\mathfrak{F}_2^{\mathbf{crit}}$ for $\delta \leq 2\epsilon$, (ii) and (iii) imply that $\varphi[\mathbf{Crit}(\mathcal{T})] = \mathbf{Nodes}(\mathcal{T}_3) = \mathbf{Crit}(\mathcal{T}_3)$. Moreover, (i) implies that $\varphi(\mathbf{u}) \preceq_{\mathcal{T}_2} \varphi(\mathbf{u}')$ if and only if $\mathbf{u} \preceq_{\mathcal{T}} \mathbf{u}'$, for all $\mathbf{u}, \mathbf{u}' \in \mathbf{Crit}(\mathcal{T})$. So φ is an essential isomorphism of $\mathfrak{F}$ to $\mathfrak{F}_3$.

4 Demonstration of Potential Biological Applicability

To illustrate the potential usefulness of our simplified FHTSs in identifying structural differences between macromolecules, we looked for two structures that are very similar, but not identical. Appropriate data sets were kindly provided by Roberto Marabini of the Universidad Autónoma de Madrid.

These data sets originate from the work of San Martín et al. [4], which investigated some biological questions associated with adenoviruses. These viruses are responsible for a large number of diseases in humans such as gastrointestinal and respiratory infections, but can also be used in gene therapy and vaccine delivery [3]. They have an icosahedral shape with a diameter of approximately 900 Å. At each of the 12 vertices of the icosahedron there is a substructure referred to as a *penton*, and the rest of the surface of the icosahedron consists of 240 *hexons*. To reflect this, our simplified FHTSs of these viruses would be expected to have 252 leaves, one for each penton or hexon. This is indeed the case, as we will see.

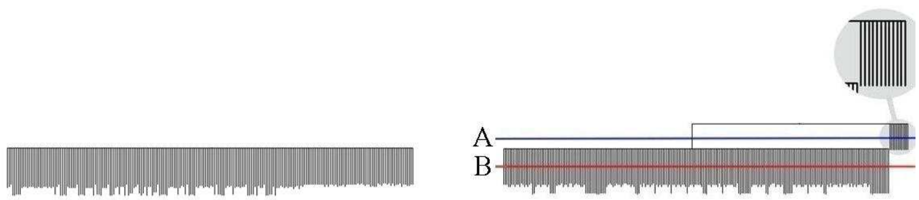
In the course of their work, San Martín et al. [4] produced a *mutant* version of the *wildtype* version of the adenovirus they were investigating. The two are identical except for a change in a protein (called IIIa). Surface renderings and central cross-sections of the two versions are shown in Fig. 1. We now describe how, in spite of their great similarity, the two versions can be distinguished from each other by an obvious topological difference between their simplified FHTSs.



(a) (b) (c) (d)

Fig. 1: Adenovirus. Surface rendering (a) and central cross-section (b) of the wildtype version. Surface rendering (c) and central cross-section (d) of the mutant version.

Each version of the virus studied by San Martín et al. [4] was represented by a grayscale volume image on a $275 \times 275 \times 275$ array of sample points. We further quantized the gray levels in each of these images to a set of just 256 equally spaced values represented by the integers $0, \ldots, 255$, where 0 corresponded to the minimum and 255 the maximum gray value in the original image. For each resulting image $\mathcal{I}$, we constructed $\mathbf{FHTS}_{\boldsymbol{\kappa}}(\mathcal{I})$ using 6-adjacency as our adjacency relation $\boldsymbol{\kappa}$, and computed the (λ, k)-simplification of $\mathbf{FHTS}_{\boldsymbol{\kappa}}(\mathcal{I})$ for various choices of λ and k. We found that $\lambda = 10$ and $k = 799$ were good choices that yielded topologically different simplified FHTSs for the two versions of the virus, as shown in Fig. 2. Here each simplified FHTS has 252 leaves, corresponding to the 12 pentons and 240 hexons. For the wildtype version, the root is the parent of all 252 leaves; see Fig. 2(a). For the mutant version, the root is the parent of the 12 leaves that correspond to pentons, but is the grandparent of the 240 leaves that correspond to hexons; see Fig. 2(b). This indicates that for the mutant version of the virus there is a substantial range of threshold levels (such as level A in Fig. 2(b)) at which the pentons are disconnected from each other and from the hexons, but the hexons are connected to each other; for the wildtype version there is no such range of threshold values. Interestingly, San Martín et al. [4] do not mention this difference between the two versions of the virus, although they do point out that in images of the mutant version pentons have lower gray values than hexons. (The latter can be seen in Fig. 1(d), and is also indicated by Fig. 2(b); when the image of the mutant virus is thresholded at the gray level B in Fig. 2(b), the hexons are observable but the pentons are not.)



(a) (b)

Fig. 2: (λ, k)-simplifications of FHTSs of wildtype (a) and mutant (b) adenoviruses, where $\lambda = 10$ and $k = 799$. In (a), the root (represented by the horizontal line segment) is the parent of all 252 leaves of the tree. In (b), the root (represented by the horizontal line segment above line A) is the parent of the rightmost 12 leaves, which correspond to pentons, but is the grandparent of the other 240 leaves, which correspond to hexons. (The representation of simplified FHTSs $(\boldsymbol{\mathcal{T}}, \ell)$ that we use here has the following properties: each non-leaf node of $\boldsymbol{\mathcal{T}}$ is represented by a horizontal line segment and, whenever a node $\mathbf{p}$ is the parent of a node $\mathbf{c}$, the edge of $\boldsymbol{\mathcal{T}}$ from $\mathbf{p}$ to $\mathbf{c}$ is represented by a vertical line segment of length proportional to $\ell(\mathbf{c}) - \ell(\mathbf{p})$ whose upper endpoint lies on the horizontal segment that represents $\mathbf{p}$; if $\mathbf{c}$ is not a leaf, then the lower endpoint of that vertical segment lies on the horizontal segment that represents $\mathbf{c}$.)

So our simplified FHTSs may possibly have revealed a previously unobserved difference between the mutant and the wildtype versions of the virus: in the mutant version, there is a substantial range of threshold values at which the hexons are connected to each other, but no penton is connected to a hexon or to another penton. To investigate whether this is a genuine difference between the two versions of the virus or merely a difference between the specific volume images from which we produced our FHTSs, we carried out a further study.

Ideally, we would have compared simplified FHTSs of, say, 10 independently reconstructed volume images of each version, but such data were not available to us. So we conducted the following approximation of such a study. For each version of the virus, we randomly selected 2000 out of 3000 available projection images, and used them to reconstruct a volume image on a $275 \times 275 \times 275$ array of points. This was repeated 10 times. For each of the 20 resulting volume images, we produced a simplified FHTS using the above-mentioned parameters. In each of the 10 simplified FHTSs of the mutant adenovirus, the root had 13 children, 12 corresponding to the pentons and the 13^{th} being the root of a subtree whose leaves corresponded to the hexons, as in Fig. 2(b). This was not true of the simplified FHTSs of the wildtype adenovirus; they were all similar to Fig. 2(a).

These results provide some evidence to support the hypothesis that images of the mutant version of the virus can indeed be distinguished from images of the wildtype version by the existence in the former (but not the latter) of a substantial range of threshold values with the above-mentioned properties. However, the evidence is based on very little data and is therefore quite tenuous. More investigation would be needed to confirm the hypothesis.

In any event, this example illustrates how our simplified FHTSs may reveal interesting structural differences between two similar macromolecules.

5 Conclusions

FHTSs can be used as descriptors of grayscale images, but unsimplified FHTSs are too sensitive to errors in the image. A robust method of simplifying FHTSs has been presented. The simplified FHTSs are potentially useful for the exploration of macromolecular databases.

Acknowledgments. We thank Roberto Marabini and Homeira Pajoohesh for discussions regarding this paper. The work was supported by Award number R01HL070472 from the National Heart, Lung, and Blood Institute. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Heart, Lung, and Blood Institute or the National Institutes of Health.

References

1. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2^{nd} ed. MIT Press, Cambridge, Mass. (2001)
2. Herman, G.T.: Geometry of Digital Spaces. Birkhäuser, Boston (1998)
3. Russell, W.C.: Update on Adenovirus and its Vectors. J. Gen. Virol. 81, 2573–2604 (2000)
4. San Martín, C., Glasgow, J.N., Borovjagin, A., Beatty, M.S., Kashentseva, E.A., Curiel, D.T., Marabini, R., Dmitriev, I.P.: Localization of the N-Terminus of Minor Coat Protein IIIa in the Adenovirus Capsid. J. Mol. Biol. 383, 923–934 (2008)
5. Sarioz, D., Kong, T.Y., Herman, G.T.: History Trees as Descriptors of Macromolecular Structures. In: Bebis, G. et al. (eds.) ISCV 2006. LNCS, vol. 4291, pp. 263–272. Springer, Heidelberg (2006)