Incremental-Decremental Technique for Delineating Tunnels and Pockets in 3D Digital Images*

R. Gonzalez-Diaz, A. Gonzalez-Molinillo, M.J. Jimenez, J.A. Postigo-Gutierrez

Applied Math Department (I), School of Computer Engineering, University of Seville, Spain
rogodi@us.es, agmolinillo@gmail.com, majiro@us.es, posti85@gmail.com

Abstract In this paper, we combine two complementary techniques for computing homology: Incremental Algorithm for computing AT-models (which consist of an algebraic set of data that provide, in particular, homological information of the given object) is suitable for homology computation in cases in which new cells are added to the existing complex, whereas Decremental Algorithm for computing AT-models is more appropriated in the case that some cells are removed from the complex. Using these algorithms, we are able to describe tunnels and pockets of a 3D digital image (given as a sequence of 2D digital images) in terms of sets of equivalent 1-cycles.

Keywords Cubical complexes, homology computation, incremental-decremental algorithm, AT-models

1 Introduction

Regarding digital images, topological invariants are being widely used to identify patterns in image recognition. One of the main topological invariant developed is the one of homology groups. Many researchers have made significant contributions in this area (for example, [2, 9, 1, 11, 5]).

In this paper, we deal with the problem of delineating tunnels and pockets in 3D digital images. Formally, a tunnel is a homology generator of dim. 1. A formal definition of pockets in macromolecules is given in [4]. The concept of a pocket in that paper is based on an acyclic relation over the set of Delaunay tetrahedra motivated by a continuous flow field. This definition reflects the idea that through a continuous growth process that simultaneously thickens every part of the protein, a pocket becomes a void inaccessible from the outside before it disappears. The definition deliberately excludes shallow valleys or depressions. The problems for delineating tunnels and pockets in 3D digital images is twofold: first, a representative cycle of a homology generator corresponding to a tunnel is a 1D closed path (which does not correspond with the intuitive idea of tunnel). Second, pockets have no translation in the homology of the 3D object.

In this paper, we combine two complementary techniques (incremental and decremental) for computing an algebraic topological datum called AT-model that provides, in particular, the homology of the given object [7, 8, 6]. More concretely, an AT-model for an object is a chain contraction [10] from the chain complex $C(K)$ canonically associated to a combinatorial structure K of the object (for example, K could be a simplicial or cubical complex) to a chain group isomorphic to the homology of the object. A formal definition of AT-model is given in Section 3.

Using these algorithms, we are able to describe tunnels and pockets of a 3D digital image (given as a sequence of 2D digital images) in terms of sets of equivalent 1-cycles. More concretely, using incremental-technique, in a sequence of 2D digital images, we define and compute *horizontal tunnels* which is a set of equivalent 1-cycles that *survive* along the sequence. It is true that all

*Partially supported by Junta de Andalucia (FQM-296 and TIC-02268) and Spanish Ministry for Science and Education (MTM-2006-03722).

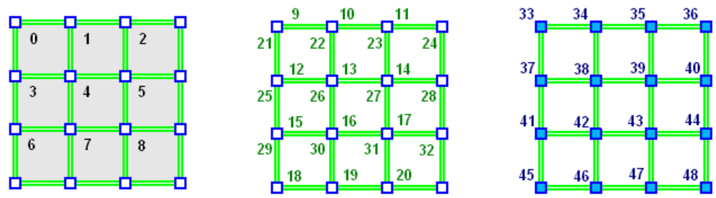


Figure 1: Coding of the cubes in a 3×3 picture (2-cubes on the left, edges on the center and vertices on the right).

the 1-cycles of an horizontal tunnel are representative cycles of a tunnel of the 3D image obtained by concatenating the successive 2D digital images in the sequence. A *frontal pocket* is the set of equivalent 1-cycles that was born at the beginning and died along the sequence. A *back pocket* is the set of equivalent 1-cycles that was born along the sequence and does not die (see Section 4). Relation between the definition of frontal and back pockets in this paper and previous definition of pockets is regarded as future work in Section 5.

The paper is organized as follows: in Section 2 we describe our coding system for the cubical structure extracted directly from the digital image; Section 3 is devoted to the explanation of the Incremental-Decremental algorithms for computing homology; such algorithms can be suitably combined to hole tracking purposes, as it is shown in Section 4; finally, we draft some plans for future in Section 5.

2 Digital Images and Cubical Complexes

A *cubical complex* Q in $\mathbf{R}^3$, is given by a finite collection of q-cubes such that a 0-cube is a vertex, a 1-cube is an edge, a 2-cube is a filled square and a 3-cube is a filled cube; together with all their faces and such that the intersection between two of them is either empty or a face of each of them. A *proper face* of $c \in Q$ is a face of c whose dimension is strictly less than the one of c. A *facet* of c is a proper face of c of maximal dimension. A facet of c is *free* if it is not a proper face of any other cube of Q. A *maximal cube* of Q is a cube that is not a proper face of any other cube of Q.

In order to provide an efficient computation of our algorithms on cubical complexes, we establish a coding system by which each cube of the corresponding cubical complex is coded by an integer. This way, the manipulation of the cubes is really efficient (in time and space) since one can get the different relationships between them by basic operations with integers. We numerate each cube in the whole picture, beginning from the 2-cubes, followed by the horizontal edges, the vertical edges and finally by the vertices. See an example of coding in Fig. 1 for a 3×3 grid of pixels.

This coding system performed in 2D pictures can be extended to nD.

3 Incremental-Decremental Algorithms for Computing Homology

In this section, we recall two different algorithms for computing homology that are somehow complementary to each other. Incremental Algorithm for computing AT-models [7, 8] is suitable for the homology computation in cases in which new cells are added to the existing complex, whereas Decremental Algorithm for computing AT-models [6] is more appropriated in the case that some cells are deleted and hence, the homological information needs to be updated.

For any graded set $S = \{S_q\}_q$, one can consider formal sums of elements of S_q, which are called *q-chains*, and which form abelian groups with respect to the component-wise addition (mod 2). These groups are called *q-chain groups* and denoted by $C_q(S)$. The collection of all the chain groups associated to S is denoted by $\mathcal{C}(S) = \{C_q(S)\}_q$ and called also chain group, for simplicity.

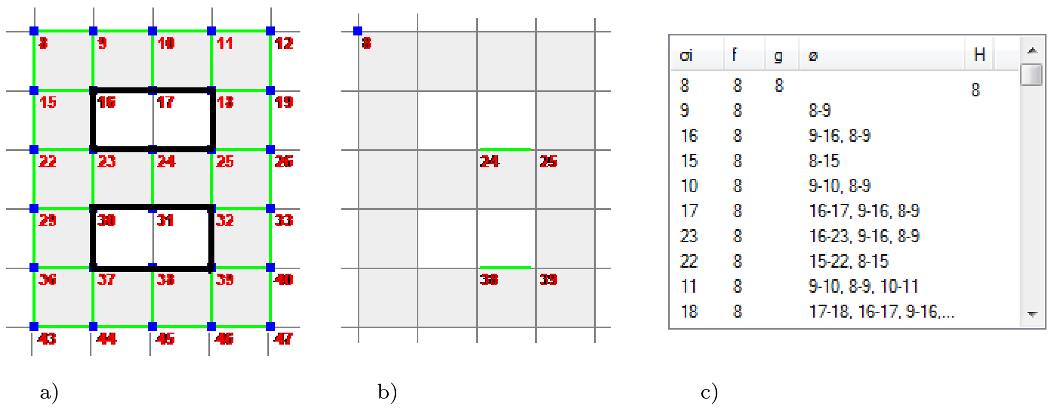


Figure 2: An example of execution of Incremental Algorithm for computing AT-models. a) The input cubical complex (only the labels of the vertices are shown). b) The cubes in H'. c) The table with the information of f', g' and ϕ'.

Let $\{s_1, \dots, s_m\}$ be the elements of a set S_q for a fixed q. Given two q-chains $c_1 = \sum_{i=1}^{m} \alpha_i s_i$ and $c_2 = \sum_{i=1}^{m} \beta_i s_i$, where $\alpha_i, \beta_i \in \mathbf{Z}/2$ for $i = 1, ..., m$, the expression $\langle c_1, c_2 \rangle$ refers to $\sum_{i=1}^{m} \alpha_i \cdot \beta_i$ mod 2. For example, fixed i and j, the expression $\langle c_1, s_i \rangle$ is α_i and $\langle s_i, s_j \rangle$ is 1 if $i = j$ and 0 otherwise.

The *cubical chain complex* associated to the cubical complex Q is the collection $\mathcal{C}(Q) = \{C_q(Q), \partial_q\}_q$ where:

(a) each $C_q(Q)$ is the corresponding chain group generated by the q-cubes of Q;

(b) the boundary operator $\partial_q : C_q(Q) \to C_{q-1}(Q)$ connects two immediate dimensions. The boundary of a q-cube is the formal sum of all its facets. It is extended to q-chains by linearity.

Roughly speaking, the homology groups of a cubical chain complex will be a chain group whose elements are equivalence classes of *cycles* (vertices in dim. 0, closed paths in dim. 1), such that if one cycle can be obtained from another by continuous deformation through the object, then they are *homologous* (or equivalent). For example, two vertices are homologous if there exists a path through the object between them. Formally, a q-*cycle* is a q-chain a such that $\partial_q(a)$ is zero.

From now on, we will omit subscripts on behalf of simplicity.

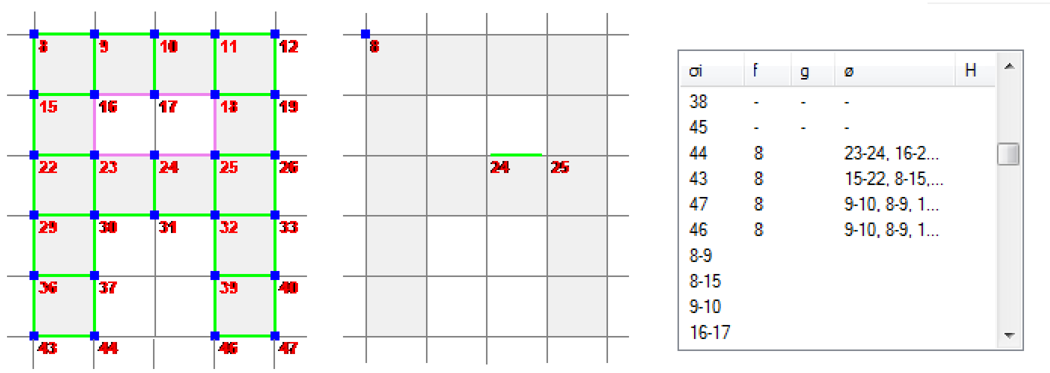
Given a cubical complex Q, an algebraic-topological model (*AT-model* [7, 8]) for Q is a set of data (Q, H, f, g, ϕ), such that:

- H is a subset of Q. In 2D, H can only have points and edges: each point of H represents a connected component of Q and each edge represents a "gap" (i.e., a background connected component) of Q. In 3D, H can contain vertices, edges and 2-cubes: each point of H represents a connected component, each edge represents a "tunnel" and each 2-cube represents a "void" (i.e. a connected component of the background inaccessible from the outside).

- f is a map from $\mathcal{C}(Q)$ to $\mathcal{C}(H)$. This map provides the equivalence relation between cycles (that is, if two cycles, a and b, are equivalent, then $f(a) = f(b)$).

- g is a map from $\mathcal{C}(H)$ to $\mathcal{C}(Q)$. For each cube c in H, $g(c)$ is a representative cycle of a class of homology.

- ϕ is a map from $\mathcal{C}(Q)$ to $\mathcal{C}(Q)$. This map can be seen as a kind of boundary inverse. For example, if c is a vertex, then $\phi(c)$ is the path from c to the vertex $v \in H$ homologous to c.



a) b) c)

Figure 3: An example of execution of Decremental Algorithm for computing AT-models. a) The output cubical complex (only the labels of the vertices are shown) after deleting some cubes from Fig. 2.a. b) The cubes in H'. c) The table with the information of f', g' and ϕ'.

3.1 Incremental Algorithm for Computing AT-Models

An algorithm for computing AT-models using the incremental technique [2] appears in [7, 8].

The input of that algorithm is a cubical complex Q (associated, for example, to a 2D digital image; see Fig. 2, on the left), an AT-model for Q, (Q, H, f, g, ϕ), and a cube c' such that its facets are in Q. The output is an AT-model $(Q \cup \{c'\}, H', f', g', \phi')$ for $Q \cup \{c'\}$, where:

C.1. If $f(\partial(c')) = 0$ then a homology class was born.

$H' := H \cup \{c'\}$; $f'(c') := c'$; $g'(c') := c' + \phi(\partial(c'))$; $\phi'(c') := 0$;
for each $\sigma \in Q$: $f'(\sigma) := f(\sigma)$; $\phi'(\sigma) := \phi(\sigma)$; and $g'(\sigma) := g(\sigma)$ if $\sigma \in H$.

C.2. If $f(\partial(c')) \neq 0$ then a homology class died.

Let $c \in Q$ such that $\langle c, f(\partial(c'))\rangle = 1$ then $H' := H \setminus \{c\}$; $f'(c') := 0$; $\phi'(c') := 0$;
for each $\sigma \in Q$: $f'(\sigma) := f(\sigma) + \langle c, f(\sigma)\rangle f(\partial(c'))$; and $\phi'(\sigma) := \phi(\sigma) + \langle c, f(\sigma)\rangle (c' + \phi(\partial(c'))$.

Observe that H and H' differ in exactly one element.

The computational cost for obtaining the AT-model $(Q \cup \{c'\}, H', f', g', \phi')$ for $Q \cup \{c'\}$ from the AT-model (Q, H, f, g, ϕ) for Q, is $\mathcal{O}(m^2)$.

Now, to obtain an AT-model for a cubical complex without having any previous AT-model computed, we consider an ordering on all the cubes of Q: $\{c_1, \dots, c_m\}$ and apply the formulas above for $i = 1$ to m. In this ordering, we put first all the free facets of Q together with all their faces in increasing dimension (this is equivalent to add first the boundary of the foreground of the image). The rest of the cubes are added later in increasing dimension. Moreover, when running the algorithm, if Case 2 occurs, then the selected $c \in Q$ will be the one with greatest index among all the candidates. Given a cubical complex Q, the computational cost for computing an AT-model for Q is $\mathcal{O}(m^3)$ where m is the number of cubes of Q ([7, 8]). See Fig. 2 as an example of execution of the algorithm.

Therefore, observe that if we have an AT-model AT_1 for a cubical complex Q and we add a new cube c' later, it is more efficient to compute an AT-model AT_2 for $Q \cup \{c'\}$ using AT_1 than to directly compute AT_2.

3.2 Decremental Algorithm for Computing AT-Models

An algorithm for computing AT-models "decrementally" appears in [6].

The input of that algorithm is a cubical complex Q, an AT-model for Q, (Q, H, f, g, ϕ), and a maximal cube c' of Q. The output is an AT-model $(Q \setminus \{c'\}, H', f', g', \phi')$ for $Q \setminus \{c'\}$, where:

Figure 4: A sequence of 10 binary digital pictures corresponding to a trabecular bone

C.1. If there exists $c \in Q$ such that $\langle c', g(c) \rangle = 1$ then a homology class died.

1.1. If $c' \in H$ and $\langle c', g(c') \rangle = 1$ then $H' := H \setminus \{c'\}$;
for each $\sigma \in Q \setminus \{c'\}$: $f'(\sigma) := f(\sigma) + \langle c', f(\sigma) \rangle c'$; $\phi'(\sigma) := \phi(\sigma) + \langle c', \phi(\sigma) \rangle g(c')$; and $g'(\sigma) := g(\sigma) + \langle c', g(\sigma) \rangle g(c')$ if $\sigma \in H'$.

1.2. If $c' \notin H$ then $H' := H \setminus \{c\}$;
for each $\sigma \in Q \setminus \{c'\}$: $f'(\sigma) := f(\sigma) + \langle c, f(\sigma) \rangle c$; $\phi'(\sigma) := \phi(\sigma) + \langle c', \phi(\sigma) \rangle g(c)$; and $g'(\sigma) := g(\sigma) + \langle c', g(\sigma) \rangle g(c)$ if $\sigma \in H'$.

C.2. If $c' \notin \text{Im}\, g$ then a homology class was born.
Let $c \in Q$ such that $c \notin H$ then $H' := H \cup \{c\}$; $g'(c) := \partial(c')$;
for each $\sigma \in Q \setminus \{c'\}$: $f'(\sigma) := f(\sigma) + \langle c', \phi(\sigma) \rangle c$; and $\phi'(\sigma) := \phi(\sigma) + \langle c', \phi(\sigma) \rangle \phi(\partial(c'))$.

Again, observe that H and H' differ in exactly one element.

The computational cost for obtaining the AT-model $(Q \setminus \{c'\}, H', f', g', \phi')$ for $Q \setminus \{c'\}$ from the AT-model (Q, H, f, g, ϕ) for Q, is $\mathcal{O}(m^2)$.

Therefore, if we have an AT-model AT_1 for a cubical complex Q and we remove a cube c' from Q later, it is more efficient to compute an AT-model AT_2 for $Q \setminus \{c'\}$ using AT_1 than to directly compute AT_2.

See Fig. 3 as an example of execution of Decremental Algorithm for computing AT-models. In this example, the input is an AT-model for a cubical complex, (Q, H, f, g, ϕ), representing a 2D image (Fig. 3a) and a list of cubes to eliminate from Q: $\{c_1, \dots c_n\}$ (the cubes to be eliminated from Q correspond to the cubes to be deleted in Fig.2.a to obtain Fig. 3.a) in decreasing dimension. It has to be satisfied that $Q \setminus \{c_1, \dots c_n\}$ is again a cubical complex. The output of the algorithm is the set (Q', H', f', g', ϕ') for $Q' = Q \setminus \{c_1, \dots, c_n\}$, in a table form (see Fig. 3 on the right).

4 Delineating Tunnels and Pockets in 3D Digital Images using Incremental-Decremental Technique

For 3D images, it is easy to define and count connected components and voids or cavities. The concept of a tunnel is a bit more delicate. Think in a hollow torus. It has one connected component, one void and two tunnels (one tunnel that passes through the hole of the torus and another one (less intuitive) that surrounds the cavity). Roughly speaking, intrusions or pockets can be defined as regions in the complement with limited accessibility from the outside. A geometrically rigorous definition of pocket using concepts of Morse theory appears in [4].

The problems for delineating tunnels and pockets in 3D digital images is twofold: first, a representative cycle of a homology generator corresponding to a tunnel is a 1D closed path. Second, pockets have no translation in the homology of the 3D object.

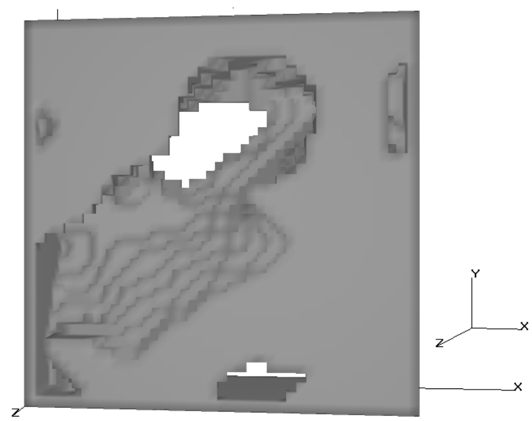


Figure 5: A piece of a trabecular bone in 3D perspective.

Our aim in this section is to describe tunnels and pockets in terms of sets of equivalent 1-cycles. For doing this, given a sequence of 2D digital images (for example, the sequence in Fig. 4 of the trabecular bone in Fig. 5), we compute an AT-model for each 2D digital image of the sequence as follows: let A and B two consecutive images in the sequence. Suppose we have computed an AT-model for A. Then, in order to obtain an AT-model for B, do the following steps:

Step 1. Apply Decremental Algorithm for computing AT-models to obtain an AT-model for $A \setminus B$ from the AT-model for A.

Step 2. Apply Incremental Algorithm for computing AT-models to obtain an AT-model for B from the AT-model for $A \setminus B$.

Following the idea of persistence of homology classes (see [3]), consider two consecutive images in the sequence, A and B. Let $(Q_A, H_A, f_A, g_A, \phi_A)$ and $(Q_B, H_B, f_B, g_B, \phi_B)$ be the AT-models for A and B obtained, respectively. We say that an element $a \in H_A \cap H_B$ is a *surviving homology class*. If $a \in H_B \setminus H_A$ then a is a *born homology class* in H_B. If $a \in H_A \setminus H_B$ then a is a *death homology class* in H_A.

This way, a *horizontal tunnel* will be a set of the cycles representing a homology class that survives along the sequence. A *frontal pocket* will be a set of the cycles representing a homology

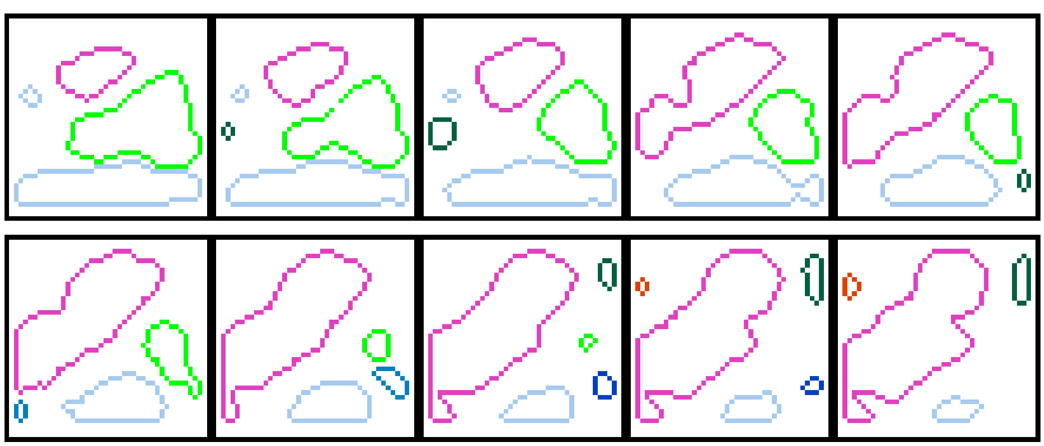


Figure 6: Representative cycles of the holes of the sequence computed using incremental-decremental technique.

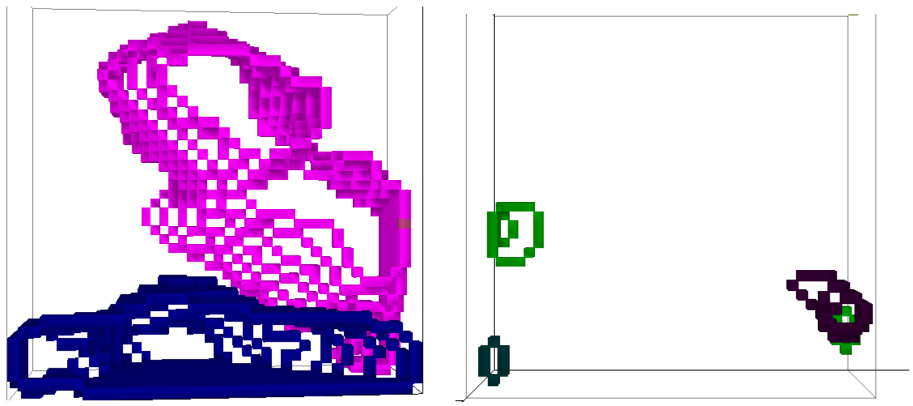


Figure 7: Representative cycles of the tunnels (on the left) and cavities (on the right) in a 3D perspective

class that was born at the beginning and died in one of the images along the sequence and a *back pocket* will be a set of cycles representing a homology class that was born along of the image of the sequence and survives. Observe that a homology class that was born along the sequence and died later is a cavity. See, for example, Fig. 6 and 7 where there are two tunnels (7 on the left), two frontal pockets (8 on the left), two back pockets (8 on the right) and four cavities (7 on the right).

5 Future Work

Relation between the definition of frontal and back pockets in this paper and previous definition of pockets is regarded as future work. Tunnels and pockets in any direction in 3D digital pictures is regarded as future work.

We think that the potential of the method will be in 3D+time images, where the computation and tracking of tunnels is important for segmentation of non-rigid moving objects.

Another possible application of the method is in morphology under topological control. For example, we could control which pockets are converted to tunnels when a closing operation is performed over an object.

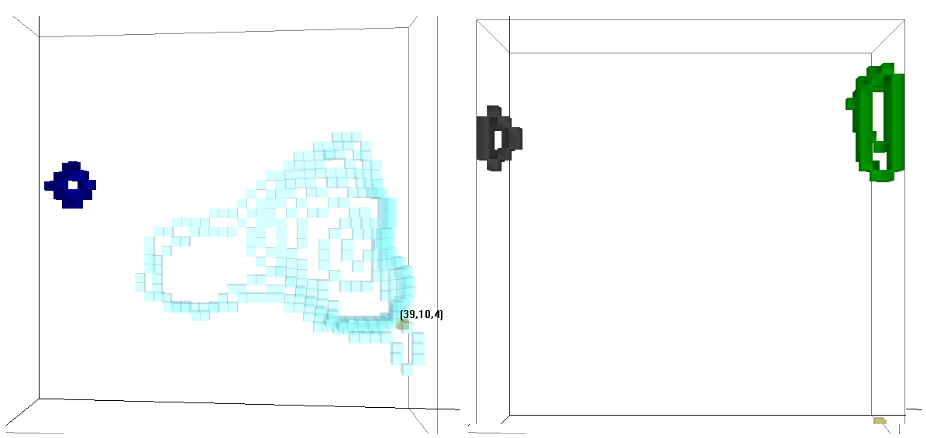


Figure 8: Representative cycles of the frontal pockets (on the left) and back pockets (on the right) in a 3D perspective

References

[1] S. Biasotti, L. De Floriani, B. Falcidieno, P. Frosini, D. Giorgi, C. Landi, L. Papaleo, and M. Spagnuolo. Describing shapes by geometrical-topological properties of real functions. *ACM Comput. Surv.*, 40(4):1–87, 2008.

[2] C. J. A. Delfinado and H. Edelsbrunner. An incremental algorithm for betti numbers of simplicial complexes on the 3–sphere. *Comput. Aided Geom. Design*, 12:771–784, 1995.

[3] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. In *FOCS '00: Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, page 454, Washington, DC, USA, 2000. IEEE Computer Society.

[4] Herbert Edelsbrunner, Michael Facello, and Jie Liang. On the definition and the construction of pockets in macromolecules. *Discrete Appl. Math.*, 88(1-3):83–102, 1998.

[5] R. Gonzalez-Diaz, M. J. Jimenez, B. Medrano, and P. Real. Chain homotopies for object topological representations. *Discrete Appl. Math.*, 157(3):490–499, 2009.

[6] R. Gonzalez-Diaz, B. Medrano, P. Real, and J. Sanchez-Pelaez. Simplicial perturbation techniques and effective homology. In *CASC 2006: Proceedings of the 9th international workshop on Computer algebra in scientific computing*, volume LNCS 4194, pages 166–177, Berlin, Heidelberg, 2006. Springer-Verlag.

[7] R. Gonzalez-Diaz and P. Real. Towards digital cohomology. In *DGCI 2003: Discrete Geometry for Computer Imagery*, volume LNCS 2886, pages 92–101, Berlin, Heidelberg, 2003. Springer-Verlag.

[8] Rocio Gonzalez-Diaz and Pedro Real. On the cohomology of 3d digital images. *Discrete Appl. Math.*, 147(2-3):245–263, 2005.

[9] T. Kaczynski, K. Mischaikow, and M. Mrozek. *Computational Homology, Applied Mathematical Sciences 157.* Springer, 2004.

[10] J.R. Munkres. *Elements of Algebraic Topology.* Addison-Wesley, 1984.

[11] Samuel Peltier, Adrian Ion, Walter G. Kropatsch, Guillaume Damiand, and Yll Haxhimusa. Directly computing the generators of image homology using graph pyramids. *Image Vision Comput.*, 27(7):846–853, 2009.