Invariant Representative Cocycles of Cohomology Generators using Irregular Graph Pyramids

Rocio Gonzalez-Diaz[a], Adrian Ion[b,d],
Mabel Iglesias-Ham[b,c], and Walter G. Kropatsch[b]

[a] *Applied Math Department, School of Computer Engineering*
University of Seville, Reina Mercedes Avenue,
CP: 41012 Seville, Spain
rogodi@us.es
[b] *Pattern Recognition and Image Processing Group, Vienna University of Technology,*
Faculty of Informatics, Institute of Computer Aided Automation,
Favoritenstr. 9/1832, A-1040 Vienna, Austria.
{ion,mabel,krw}@prip.tuwien.ac.at
[c] *Pattern Recognition Department, Advanced Technologies Application Center,*
7th Avenue #21812 %218 and 222, Siboney Neighborhood,
Playa, C.P. 12200, Havana City, Cuba.
miglesias@cenatav.co.cu
[d] *Institute for Numerical Simulation*
Faculty of Mathematics and Natural Sciences, University of Bonn,
Wegelerstr. 6, 53115 Bonn, Germany.

Abstract

Structural pattern recognition describes and classifies data based on the relationships of features and parts. Topological invariants, like the Euler number, characterize the structure of objects of any dimension. Cohomology can provide more refined algebraic invariants to a topological space than does homology. It assigns 'quantities' to the chains used in homology to characterize holes of any dimension. Graph pyramids can be used to describe subdivisions of the same object at multiple levels of detail. This paper presents cohomology in the context of structural pattern recognition and introduces an algorithm to efficiently compute representative cocycles (the basic elements of cohomology) in 2D using a graph pyramid. An extension to obtain scanning and rotation invariant cocycles is given.

Keywords: graph pyramids, representative cocycles of cohomology generators

1. Introduction

Image analysis deals with digital images as input to pattern recognition systems with the purpose to extract information about their content, usually objects. Objects appear in images affected by transformations (e.g. rotation, zoom, projection) and noise. Topological features have the ability to ignore changes in the geometry of objects by extracting object properties invariant to elastic transformations. Simple topological features are for example the number of connected components, the number of holes, etc., while more refined ones, like homology and cohomology, characterize holes and their relations.

An example application of topological features is topology simplification, an active field in geometric modeling and medical imaging where high-resolution surfaces are created through iso-surface extraction from volumetric representations, obtained by 3D photography, CT, or MRI. Iso-surfaces often contain many topological errors in the form of tiny handles. These nearly invisible artifacts hinder subsequent operations like mesh simplification, re-meshing, and parametrization. See, for example [1]. Another application is shape description and matching, where persistence and homology of a function defined on a shape have been successfully applied to extract shape features [2].

A 2D image is the result of projecting a 3D scene into the image plane. Often the precise camera parameters are not known and still humans have no problem in correctly interpreting the displayed objects in the image. A 3D object is surrounded by a reflecting surface which itself may split into several smaller but connected patches which can be characterized by their color, their texture or other visual properties. The visible part of the object's surface maps into a region of the image which shows the same adjacencies of patches as the original surface (because *the camera sees the same side of the surface*) although the geometry of the patches may change due to projection, due to camera or object movement or due to deformations of the object. Sometimes a collection of patches is completely surrounded by some other surface patches, e.g. a fancy soccer ball on a Spanish T-shirt (see Fig. 1). Although both the picture of the ball as well as the T-shirt may have a specific patch structure (stripes on shoulder and arms, logo) it is clear which subset of regions forms the ball and which regions belong to the remaining T-shirt. A simplified version could describe the pixels of the T-shirt 1's and the ball's pixels by 0 (see Fig. 1). Then the ball is a *hole* in the T-shirt

a)   b)

Figure 1: a) Original image: ball on T-shirt of world champions 2010 in soccer; b) a segmentation of the original image using pyramids.

as long as the surrounding of the ball is visible and not occluded by other objects. It may be highly difficult to uniquely and reliably identify any of the involved small patches individually under difficult geometric deformations while the overall arrangement of patches forming the ball is mostly invariant to these geometric deformations. How to segment patches into meaningful aggregations has been dealt with in many other segmentation methods and it is not the main emphasis of this paper. We therefore restrict ourselves in the following on binary images with the understanding that each region may be the collection of several subregions belonging together.

Considering 2D binary images, an *object* is defined by a connected set (4-connectivity) of foreground pixels. A *region adjacency graph* (RAG) encodes the adjacency of regions in a partition. The holes in a RAG associated to an object of a 2D binary digital image can be characterized by establishing an equivalence between all the cycles as follows: two cycles are equivalent if one can be obtained from the other by joining to it one or more *degenerate cycles* (cycles with exactly 4 edges). For example, there is only one equivalence class for the foreground (gray pixels) of the digital image in Fig. 2, which represents the unique white hole. This is similar to considering the digital image as a cell complex[1] [3] (see Fig. 2.c). Unfortunately digital images are not 'clean', noise can create many unwanted holes which complicate the correct interpretation. One can ask for the edges we have to delete in order

---

[1]Intuitively a cell complex is defined by a set of 0-cells (vertices) that bound a set of 1-cells (edges), that bound a set of 2-cells (faces), etc.

to 'destroy' a hole. In the example in Fig. 2, it is not enough to delete only one edge. The deletion of the bold edges in Fig. 2.c together with the faces that they bound produces the 'disappearing' of the hole. The set of bold edges in Fig. 2.c define a 1-cocycle, a topological invariant of the respective object. Equivalence classes of such cocycles are the elements of *cohomology*.

To cope with complexity issues that arise when highly complex algorithms must be applied to huge amounts of data, graph pyramids are used. These hierarchical data structures offer possibilities to reduce the amount of data by local operations which can be applied in parallel and which have the enormous advantage to preserve the topological properties of the data. Hence the search for independent cocycles can be correctly done on a fraction of the data at the top of the pyramid. The simplified geometry of these cocycles can then be delineated top-down through the levels of the pyramid by again local processes up to the high accuracy of the base level.

Maybe due to its more abstract nature, lacking a geometric meaning and due to its computation complexity, cohomology has not been yet widely applied to pattern recognition and image processing. This paper is possibly the first attempt to use it in the context of digital images. For this purpose, in this paper we consider the best known environment which are 2D images whereas nD is the ultimate goal. Concepts related to cohomology can have associated interpretations in graph theory. Having these interpretations opens the door for applying classical efficient graph theory algorithms to compute and manipulate these features. Besides, for objects embedded in $\mathbb{R}^3$, homology– a wider used topological invariant, and cohomology groups are isomorphic. But the ring structure presented in cohomology characterizes the relations between 2-holes (cohomology generators of dimension 1), which homology does not. Indeed, dealing with homology and cohomology properties, representative cycles and cocycles, and their computation is quite different and doing this study in 2D gives important insights which should be relevant for extension to nD, $n > 2$. Initial results regarding this work have been presented in [4]. The current paper extends our earlier publication with detailed insights and proofs, and a refinement of the previous method that makes the obtained cocycles scanning and rotation invariant in the case of an identical discretization.

The paper is organized as follows: Sections 2 and 3 recall graph pyramids and cohomology, and make initial connections. In Section 4, preserving-topology properties in irregular graph pyramids are given. Section 5 presents the proposed method. Section 6 uses the properties of the proposed method
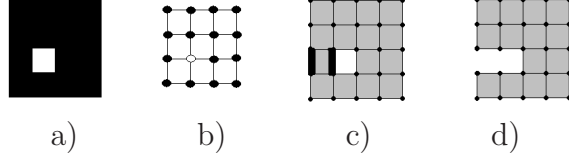
Figure 2: a) A 2D digital image I; b) its RAG; c) a cell complex associated to I (with bold edges, a representative cocycle); and d) the cell complex without the hole.

to extend it and obtain scanning and rotation invariant cocycles. Section 7 concludes the paper.

2. Irregular Graph Pyramids

A graph (see for example [5]) is an ordered pair $G = (V, E)$ comprising a set V of vertices and a set E of edges. Each edge $e \in E$ is incident to two not necessarily distinct vertices $v, w \in V$, written as $e = (v, w)$. An edge $e = (v, w)$ is said to be directed if the pair (v, w) is an ordered pair. An edge is said to be a self-loop if $v = w$. The edge $e = (v, w)$ is called a parallel edge iff $\exists e' \in E, e' \neq e$ s.t. $e' = (v, w)$. A graph is called undirected if none if its edges is directed, and it is called planar if it can be drawn in a plane with no edges crossing (vertices are drawn as points and edges as lines connecting their incident vertices). Given a graph $G = (V, E)$ removing an edge $e \in E$ will result in the graph $G' = (V, E \setminus \{e\})$, contracting the edge $e = (v, w)$ implies removing it and identifying its incident vertices s.t any remaining edge previously incident to v or w is now incident to the unique vertex $v = w$.

Given a decomposition of an object or image into regions a region adjacency graph (RAG) is an undirected graph that encodes the adjacency of regions in a partition. A vertex is associated to each region, vertices of neighboring regions are connected by an undirected edge. Classical RAGs do not contain any self-loops or parallel edges. An *extended region adjacency graph* (eRAG) is a RAG that contains the so-called *pseudo edges*, which are self-loops and parallel edges used to encode neighborhood relations to a cell completely enclosed by one or more other cells [6]. The *dual* graph of an eRAG G is called a *boundary graph* (BG) and is denoted by $\bar{G}$ (G is said to be the *primal* graph of $\bar{G}$). The edges of $\bar{G}$ represent the boundaries (borders) of the regions encoded by G, and the vertices of $\bar{G}$ represent points
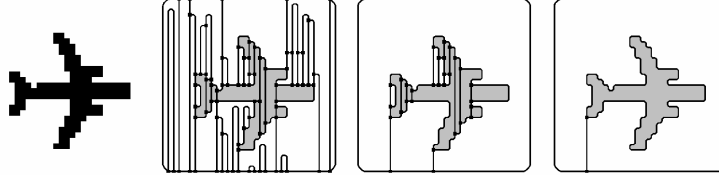
5

Figure 3: A digital image I, and boundary graphs $\bar{G}_6$, $\bar{G}_{10}$ and $\bar{G}_{15}$ of the pyramid of I.

where boundary segments meet. G and $\bar{G}$ are planar graphs. There is a one-to-one correspondence between the edges of G and the edges of $\bar{G}$, which induces a one-to-one correspondence between the vertices of G and the 2D cells (will be denoted by $faces^2$) of $\bar{G}$. The dual of $\bar{G}$ is again G. The following operations are equivalent: edge contraction in G with edge removal in $\bar{G}$, and edge removal in G with edge contraction in $\bar{G}$.

A (dual) irregular graph pyramid [7, 8, 6, 9] is a stack of successively reduced planar graphs $P = \{(G_0, \bar{G}_0), \ldots, (G_n, \bar{G}_n)\}$. Each level $(G_k, \bar{G}_k)$, $0 < k \leq n$, is obtained by first contracting edges in G_{k-1} (removal in $\bar{G}_{k-1}$), if their end vertices have the same label (regions should be merged), and then removing edges in G_{k-1} (contraction in $\bar{G}_{k-1}$) to simplify the structure. The contracted and removed edges are said to be *contracted* or *removed* (sometimes called *removal* edges) in $(G_{k-1}, \bar{G}_{k-1})$. In each G_{k-1} and $\bar{G}_{k-1}$, contracted edges form trees called *contraction kernels*. One vertex of each contraction kernel is called a *surviving vertex* and is considered to have 'survived' to $(G_k, \bar{G}_k)$. The vertices of a contraction kernel in level $k - 1$ form the *reduction window* of the respective surviving vertex v in level k. The *receptive field* of v is the (connected) set of vertices from level 0 that have been 'merged' to v over levels $0, \ldots, k$. The *equivalent contraction kernel (ECK)* of a vertex v is the tree obtained by replacing v and all is descendants with their corresponding contraction kernels. The vertices of the ECK of v form the receptive field of v.

Algorithm 1 gives the main steps used to build a graph pyramid. In Line 5 the operations are performed on both graphs (adjacency and boundary) even if for simplicity only one of the graphs is mentioned in each step. Line 5.ii removes self-loops bounding an empty face in the adjacency graph. Line 5.iii simplifies the region boundaries in the boundary graph.

---

[2]Not to be confused with the vertices of the dual of a RAG (sometimes also denoted by the term *faces*).

Algorithm 1 Build (dual) graph pyramid

Input: image I /*pixels labeled 'object' or 'background'*/

1: $(G_0, \bar{G}_0) = ((V_0, E_0), (\bar{V}_0, \bar{E}_0))$
/*V_0 associates a vertex to every pixel. E_0 connects vertices corresponding to 4-connected pixels. G_0 and $\bar{G}_0$ are dual.*/
2: $k = 0$
3: **repeat**
4:     /*select edges to contract*/
     $T = \emptyset$
     for all $v \in G_k$ **do**
       i. select an edge $(v, w) \in G_k$ with v, w having the same label
       ii. $T \leftarrow T \cup (v, w)$ /*add edge*/
5:     **if** $T \neq \emptyset$ **then**
     /*region merging, easier described in the adjacency graph:*/
     i. $(G', \bar{G}') \leftarrow$ contract edges T of G_k (removal in $\bar{G}_k$)
     /*simplification, easier described in the boundary graph:*/
     ii. $(G'', \bar{G}'') \leftarrow$ contract pending trees in $\bar{G}'$ (removal in G')
     iii. $(G_{k+1}, \bar{G}_{k+1}) \leftarrow$ contract one distinct edge incident to each vertex of degree 2 in $\bar{G}''$ (removal in G'')
6:     $k \leftarrow k + 1$
7: **until** $T = \emptyset$

Output: Graph pyramid $P = \{(G_0, \bar{G}_0), \ldots, (G_{k-1}, \bar{G}_{k-1})\}$.

---

3. Homology, Cohomology and Integral Operators

We refer to [10] for an introduction to homology and cohomology.

Intuitively, homology characterizes the holes of any dimension (i.e. connected components, 1-dimensional holes, etc.) of an n-dimensional object. It defines the concept of *generators* which, for example for 2D objects are similar to closed paths of edges surrounding holes. More general, k-dimensional manifolds surrounding $(k + 1)$-dimensional holes are generators [10], and define equivalence classes of $(k+1)$-holes. Cohomology arises from the algebraic dualization of the construction of homology. It manipulates groups of homomorphisms to define equivalence classes. Intuitively, cocycles (the invariants computed by cohomology), represent the sets of elements (e.g. edges) to be removed to destroy certain holes. See Fig. 2.c for an example cocycle.

0-cells	$\{v_1, v_2, v_3, v_4\}$
1-cells	$\{e_1, e_2, e_3, e_4, e_5, e_6\}$
2-cells	$\{f_1\}$
1-chain	$e_1 + e_3$
1-boundary	$\partial f_1 = e_1 + e_2 + e_5$
1-cycle	$a = e_3 + e_4 + e_5$
1-cycle	$b = e_1 + e_2 + e_3 + e_4$
homologous cycles	a and b; since $a = b + \partial f_1$

Figure 4: Example of cell, chain, boundary and cycle.

A *homeomorphism* is a bijective continuous function between two spaces, that has a continuous inverse function. They are the mappings which preserve all the topological properties of a given space. Two spaces with a homeomorphism between them are called *homeomorphic*, and from a topological viewpoint they are the same.

Two continuous functions from one topological space to another are called *homotopic* if one can be 'continuously deformed' into the other. Two spaces X and Y are *homotopy equivalent* if there are maps $f : X \to Y$ and $g : Y \to X$ such that gf is homotopic to id_X and fg is homotopic to id_Y. Observe that if two spaces are homeomorphic then they are homotopic. A *homotopy invariant* is a topological property which is invariant under homotopy.

A *p*-dimensional *cell* (or *p*-cell, for short) is a topological space that is homeomorphic to the *p*-dimensional ball B^p. A 0-cell is homeomorphic to a point, a 1-cell to an arc and a 2-cell to a disk. Roughly speaking, a *cell- (or CW-) complex* is built by gluing together the basic building blocks called cells.

Figures 4 and 5 illustrate the following abstract concepts.

3.1. Homology

The notion of *p-chain* is defined as a formal sum of *p*-cells. The chains are considered over $\mathbb{Z}/2$ coefficients i.e. a *p*-cell is either present in a *p*-chain (coefficient 1) or absent (coefficient 0) - any cell that appears twice vanishes. The set of *p*-chains form an abelian group called the *p-chain group* C_p. This group is generated by all the *p*-cells. The *boundary operator* is a set of homomorphisms $\{\partial_p : C_p \to C_{p-1}\}_{p \geq 0}$ connecting two consecutive dimensions. By linearity, the boundary of any *p*-chain is defined as the formal

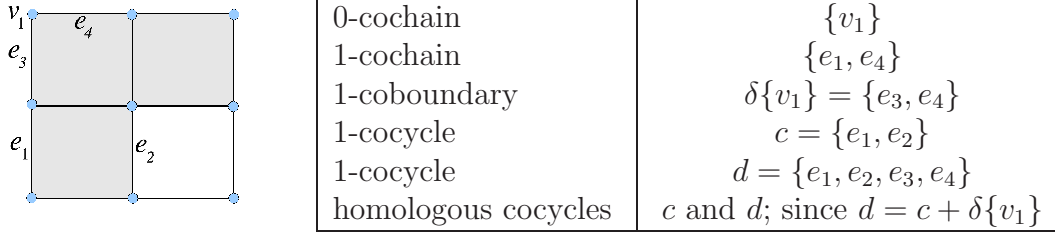| 0-cochain	$\{v_1\}$
1-cochain	$\{e_1, e_4\}$
1-coboundary	$\delta\{v_1\} = \{e_3, e_4\}$
1-cocycle	$c = \{e_1, e_2\}$
1-cocycle	$d = \{e_1, e_2, e_3, e_4\}$
homologous cocycles	c and d; since $d = c + \delta\{v_1\}$

Figure 5: Example of cochain, cocycle and coboundary.

sum of the boundaries of each p-cell that appears in the chain. The boundary of 0-cells (i.e. points) is always 0. A *chain complex* is the set of all the chain groups connected by the boundary operator: $\cdots \overset{\partial_{p+1}}{\to} C_p \overset{\partial_p}{\to} C_{p-1} \to \cdots \overset{\partial_1}{\to} C_0 \overset{\partial_0}{\to} 0$.

A p-chain σ is called a *p-cycle* if $\partial_p(\sigma) = 0$. If $\sigma = \partial_{p+1}(\mu)$ for some $(p+1)$-chain μ then σ is called a *p-boundary*. Two p-cycles a and a' are *homologous* if there exists a p-boundary b such that $a = a' + b$. Denote the groups of p-cycles and p-boundaries by Z_p and B_p respectively. For each p, $\partial_{p-1}\partial_p = 0$. In other words, all p-boundaries are p-cycles ($B_p \subseteq Z_p$). Define the pth *homology group* to be the quotient group $H_p = Z_p/B_p$, for all p. Each element of H_p is a class obtained by adding each p-boundary to a given p-cycle a. Then a is a *representative p-cycle* of the homology class $a + B_p$.

Since the chains are considered over $\mathbb{Z}/2$ coefficients, the chain groups are vector spaces and the boundary operators are linear operators. The cycle and and boundary groups are just the kernel and image of such operator. The homology group is a quotien space.

3.2. Cohomology

Cohomology groups are constructed by turning chain groups into groups of homomorphisms and boundary operators into their dual homomorphisms. Define a *p-cochain* as a homomorphism $c : C_p \to \mathbb{Z}/2$. We can see a p-cochain as a binary mask of the set of p-cells: imagine you order all p-cells in the complex. (let's say we have n p-cells, and call this ordered set S_p). Then a p-cochain c is a binary mask of n values in $\{0,1\}^n$, where n is the number of p-cells in the complex. When no confusion can arise, we will identify the p-cochain c with the set S of p-cells that are evaluated to 1 by c.

The p-cochains form the set C^p which is a group. The boundary operator defines a dual set of homomorphisms, the *coboundary operator* $\{\delta^p : C^p \to C^{p+1}\}_{p \geq 0}$, such that $\delta^p(c) = c\partial_{p+1}$ for any p-cochain c. Since the coboundary
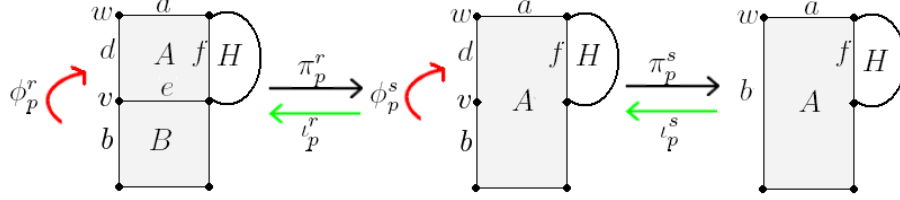
9

Figure 6: The cell complex K, K' and K''; and the homomorphisms $\phi_p^r, \pi_p^r, \iota_p^r$ and $\phi_p^s, \pi_p^s, \iota_p^s$.

operator runs in a direction opposite to the boundary operator, it raises the dimension. Its kernel is the group of *cocycles* and its image is the group of *coboundaries*. A p-cochain c is a p-cocycle if $\delta^p c (= c\partial_{p+1}) : C_{p+1} \to \mathbb{Z}/2$ is the null homomorphism. A p-cochain d is a p-coboundary if there exists a $(p-1)$-cochain e such that $d = \delta^{p-1} e (= e\partial_p)$. Two p-cocycles c and c' are *cohomologous* if there exists a p-coboundary d such that $c = c' + d$. The pth *cohomology group* is defined as the quotient of p-cocycle modulo p-coboundary groups, $H^p = Z^p/B^p$, for all p. Each element of H^p is a class obtained by adding each p-coboundary to a given p-cocycle c. Then c is a *representative p-cocycle* of the cohomology class $c + B^p$, denoted by $[c]$.

Definition 3.1. *A set of p-cocycles $\{c_1, \ldots, c_n\}$ is a basis of representative p-cocycles if:*

- *Any other p-cocycle c can be written as a linear combination of the p-cocycles of the set plus a p-coboundary, that is:*

$$c = \sum_{i=1}^{n} \lambda_i c_i + \delta^{p-1} e$$

  where $\lambda_i = 0, 1$ and e is a $(p-1)$-cochain.

- *None of the p-cocycles in the set can be written as a linear combination of the rest plus a p-coboundary (minimality).*

Remark 3.1. *If $\{c_1, \ldots, c_n\}$ is a basis of representative p-cocycles then $\{[c_1], \ldots, [c_n]\}$ is a basis of the pth cohomology group of the object. Each $[c_i]$ is a cohomology generator.*

10

	ϕ_p^r	π_p^r
e	B	$a + f + d$
B	0	0
A	0	A
other p-cell σ	0	σ

	ϕ_p^s	π_p^s
v	d	w
d	0	0
b	0	b
other p-cell σ	0	σ

Figure 7: The homomorphisms ϕ_p^r, π_p^r and ϕ_p^s, π_p^s.

3.3. Integral Operator

Starting from a chain complex, $\cdots \overset{\partial_2}{\to} C_1 \overset{\partial_1}{\to} C_0 \overset{\partial_0}{\to} 0$, take a q-cell σ and a $(q+1)$-chain α. An *integral operator* [11] is defined as the set of homomorphisms $\{\phi_p : C_p \to C_{p+1}\}_{p \geq 0}$ such that $\phi_q(\sigma) = \alpha$, $\phi_q(\mu) = 0$ if μ is a q-cell different to σ, and for all $p \neq q$ and any p-cell γ we have $\phi_p(\gamma) = 0$. It is extended to all p-chains by linearity.

An integral operator $\{\phi_p : C_p \to C_{p+1}\}_{p \geq 0}$ satisfies the *chain-homotopy property* iff $\phi_p \partial_{p+1} \phi_p = \phi_p$ for each p. For ϕ_p satisfying the chain-homotopy property, define $\pi_p = id_p + \phi_{p-1} \partial_p + \partial_{p+1} \phi_p$ where $\{id_p : C_p \to C_p\}_{p \geq 0}$ is the identity. Define $im\pi_p = \{b \in C_p \mid \exists a \in C_p \text{ s.t. } \pi_p(a) = b\}$. Then, $\cdots \overset{\partial_2}{\to} im\pi_1 \overset{\partial_1}{\to} im\pi_0 \overset{\partial_0}{\to} 0$ is a chain complex and $\{\pi_p : C_p \to im\pi_p\}_{p \geq 0}$ is a *chain equivalence* [10]. Its chain-homotopy inverse is the inclusion map $\{\iota_p : im\pi_p \to C_p\}_{p \geq 0}$. Integral operators satisfying the chain-homotopy property can be seen as a kind of inverse boundary operator: They raise the dimension and satisfy the nilpotent condition $\phi_{p+1} \phi_p = 0$ for all p. Although, in general, $\phi_{p-1} \partial_p \neq id_p$ and $\partial_{p+1} \phi_p \neq id_p$, what happens is $\phi_p \partial_{p+1} \phi_p = \phi_p$ for all p (which would be equivalent to $x \cdot \frac{1}{x} \cdot x = x$ for $x \in \mathbb{R} \setminus \{0\}$). Consider, for example, the cell complex K in Fig. 6 on the left. The integral operator associated to the removal of the edge e is given by $\phi_1^r(e) = A$. Then, $\pi_1^r(e) = (id_1 + \phi_0^r \partial_1 + \partial_2 \phi_1^r)(e) = e + a + f + d + e = a + f + d$, $\pi_2^r(A) = 0$, $\pi_2^r(B) = A$ ($A + B$ is renamed as A in K') and π_p^r is the identity over the other p-cells of K, $p = 0, 1, 2$. The removal of edge e decreases the degree of vertex v allowing for further simplification.

The following lemma guarantees the correctness of the down projection procedure for computing cocycles given in Section 5. Since graph pyramids offer possibilities to reduce the amount of data by local operations, then the search for independent cocycles can be done on the top of the pyramid. Hence, the following lemma guarantees that these cocycles can be correctly delineated top-down through the levels of the pyramid.

11

Lemma 3.2. *Let $\{\phi_p : C_p \to C_{p+1}\}_{p\geq 0}$ be an integral operator satisfying the chain-homotopy property. The chain complexes $\cdots \overset{\partial_2}{\to} C_1 \overset{\partial_1}{\to} C_0 \overset{\partial_0}{\to} 0$ and $\cdots \overset{\partial_2}{\to} im\pi_1 \overset{\partial_1}{\to} im\pi_0 \overset{\partial_0}{\to} 0$ have isomorphic homology and cohomology groups. If $c : im\pi_p \to \mathbb{Z}/2$ is a representative p-cocycle of a cohomology generator, then $c\pi : C_p \to \mathbb{Z}/2$ is a representative p-cocycle of the same generator.*

Proof. An integral operator that satisfies the chain homotopy property, is a chain homotopy of the identity $\{id_p : C_p \to C_p\}_{p\geq 0}$ to $\{\iota_p\pi_p : C_p \to C_p\}_{p\geq 0}$. Therefore, $\{\pi_p : C_p \to im\pi_p\}_{p\geq 0}$ is a chain equivalence and chain equivalences induce isomorphisms on homology and cohomology (see [10]). $\square$

For example, consider the cell complex K' of Fig. 6. The 1-cochain α^*, defined by the set $\{b, f\}$ of edges of K', is a 1-cocycle which 'blocks' the white hole H (in the sense that all the cycles representing the hole must contain an odd number of edges of α^*). Then $\beta = \alpha^*\pi_1^r$ is defined by the set $\{b, f, e\}$ of edges of K. α and β are both 1-cocycles representing the same white hole H.

4. Preserving Topology in Irregular Graph Pyramids

Considering binary images, an *object* is defined by a connected set (4-connectivity) of foreground pixels. A partition of the whole space (foreground and background) in cells is called a *cell subdivision*. The referred partition could be obtained from any of the planar graphs in every level of the pyramid.

Fix a level $(G_i, \bar{G}_i)$, the *cell complex* associated to the foreground object, called *boundary cell complex*, denoted by K^i, is obtained from $(G_i, \bar{G}_i)$ by taking all faces of $\bar{G}_i$ corresponding to vertices of G_i, whose receptive fields contain (only) foreground pixels, and adding all edges and vertices needed to represent the faces. The p-chain group generated by the p-cells of K^i is denoted by $C_p(K^i)$.

The following lemma guarantees that the local operations applied to build a graph pyramid preserve the topological properties of the initial data.

Lemma 4.1. *The boundary cell complex is well-defined. All the boundary cell complexes of a given irregular dual graph pyramid are cell subdivisions of the same object. Therefore, all these cell complexes are homeomorphic.*

Proof. Our input is a binarized 2D digital image. An object is the 4-connected set of foreground pixels. Since we only remove an edge in $\bar{G}_k$
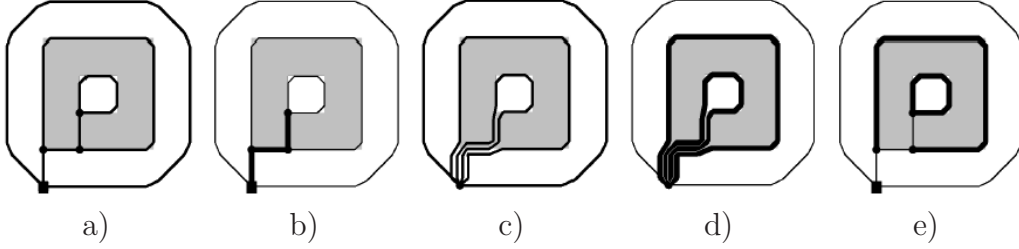
12

Figure 8: a) Boundary cell complex K_4 obtained from the top level of the pyramid, $(G_4, \bar{G}_4)$; b) in bold, spanning tree edges of $\bar{G}_4$; c) homology-generator level, K^H; d) in bold, the self-loops representing the cocycle edges in K^H; e) in bold, the cocycle edges in top level.

when it is in the boundary of two different regions that have the same label (region merging), and contract an edge in $\bar{G}_k$ when it is incident to a vertex of degree 2 or it is a pendant edge (simplification), all the new p-cells created are homeomorphic to p-dimensional balls, $p = 0, 1, 2$. □

As a result of Lemma 4.1, topological invariants computed on different levels of the pyramid are equivalent.

For the purpose of this paper, a new cell complex called *homology-generator level*, is added over the boundary cell complex obtained from the top (last) level of the pyramid (see Fig. 8). This new cell complex is denoted by K^H and it is a set of regions surrounded by a set of self-loops incident to a single vertex. To obtain this cell complex, on the top of the computed pyramid, we compute a spanning tree of the boundary graph of the top level of the pyramid, and contract all the edges that belong to it. Note that K^H is no longer homeomorphic to any K^i, but homotopic.

Lemma 4.2. *The two operations used to construct an irregular graph pyramid: edge removal and edge contraction, are integral operators satisfying the chain-homotopy property.*

Proof. Fix a level $(G_i, \bar{G}_i)$, suppose an edge e in $\bar{G}_i$ is removed. Since $\bar{G}_i$ is planar, then e is in the boundary of two 2-cells (or regions) A and B (see Fig. 6). The integral operator ϕ^r associated to this edge removal is given by $\phi^r_1(e) = B$ (see Fig. 7). Now, suppose that an edge d of $\bar{G}_i$, with a vertex v of degree 2 in its boundary, is contracted (see Fig. 6). The integral operator ϕ^s associated to this edge contraction is given by $\phi^s_0(v) = d$ (see Fig. 7). □

13

Starting from a cell decomposition of an object, its homology studies incidence relations of its subdivision. Cohomology arises from the algebraic dualization of the construction of homology. Both homology and cohomology are homotopy invariants.

Corollary 4.3. *The boundary cell complex of any level of the pyramid and the homology-generator level have isomorphic homology and cohomology groups.*

As a consequence of Lemmas 3.2 and 4.2 we have:

Lemma 4.4. *Fix a level $(G_i, \bar{G}_i)$, suppose an edge e in $\bar{G}_i$, which is in the boundary of a region B, is removed. Let $\phi_1^r(e) = B$ be the integral operator associated to this removal. Let α^* be a 1-cocycle defined by a set of edges S in $\bar{G}_i \setminus \{e\}$. If an odd number of edges of α^* is in B, then $S \cup \{e\}$ defines a 1-cocycle in K^i; otherwise, it is S which defines a 1-cocycle in K^i.*

In terms of embedded graphs, an integral operator maps a vertex/point to exactly one of its incident edges and an edge to exactly one of its incident faces. In every level of a graph pyramid, the contraction kernels make up a spanning forest. A forest composed of k connected components, spanning a graph with n vertices, has k root vertices, $n - k$ other vertices, and also $n - k$ edges. These edges can be oriented toward the respective root such that each edge has a unique starting vertex. Then, integral operators mapping the starting vertices to the corresponding edge of the spanning forest can be defined as follows: $\phi_0(v_i) = e_j$, where e_j is the edge incident to v_i, oriented away from it.

The following lemma guarantees that all integral operators that create homeomorphisms are in fact a combination of the two operations used to construct irregular graph pyramids.

Lemma 4.5. *All integral operators that create homeomorphisms can be represented in a dual graph pyramid. This is equivalent to: given an input image $(G_0, \bar{G}_0)$ and its associated cell complex $Z = \{C_0, C_1, C_2\}$, a cell complex $Z' = \{C_0', C_1', C_2'\}$ with Z, Z' homeomorphic, and Z a refinement of Z' i.e. $C_0' \subseteq C_0$, $C_1' \subseteq C_1$, and $C_2' \subseteq C_2$, then there exists a pyramid P s.t. Z' is the cell complex associated to some level $(G_k, \bar{G}_k), k \geq 0$, of P.*
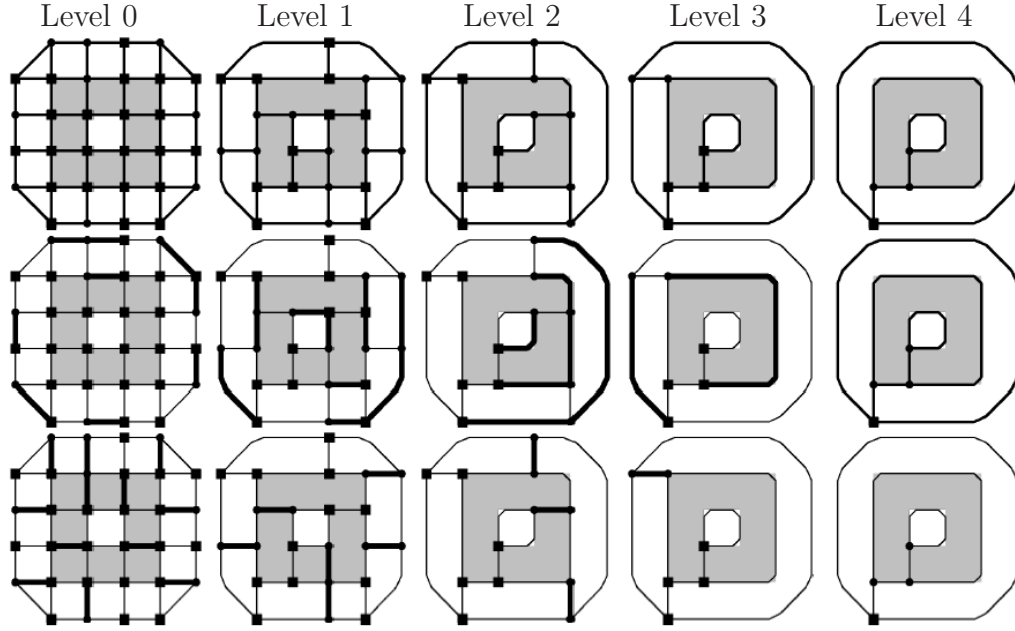
Figure 9: Top row, from left to right: boundary graphs for all levels of the pyramid. Vertices surviving to the next level are drawn with a square. Middle row, in bold: contracted edges in the respective levels. Bottom row: removed edges in the boundary graph, equivalent to contracted edges in the adjacency graph.

5. Representative Cocycles in Irregular Graph Pyramids

A method for efficiently computing representative cycles of homology generators using an irregular graph pyramid is given in [12]. In [13] a novel algorithm for correctly visualizing graph pyramids, including multiple edges and self-loops is given. This algorithm preserves the geometry and the topology of the original image and has been used to produce the images throughout the paper (see Fig. 9)

In this paper, representative cocycles are computed and drawn in the boundary graph of any level of a given irregular graph pyramid. They are computed in the top level and down projected using the described process.

In the homology-generator level (see Fig. 8.c), each self-loop α that surrounds a region of the background (hole of a region R of the foreground) is a representative 1-cycle of a homology generator.

15

Let K^H be the homology-generator level. Without loss of generality, we can suppose that K^H is connected. If not, repeat the following reasoning for each connected component (region) of K^H. Let $\{\alpha_1, \dots, \alpha_n\}$ be the set of the self-loops surrounding a face of the background. Therefore, there are n white holes: $O_1, \dots O_n$ (see Fig. 8.d). Fix i, $i = 1, \dots, n$, α_i is a representative 1-cycle of the homology generator associated to the white hole O_i. Let β be a self-loop surrounding the face f of the foreground (recall that we suppose that K^H is connected) such that α_i is in the boundary of f in K^H. Form the sets $\{\alpha_1, \beta\}, \dots, \{\alpha_n, \beta\}$. Let K^0 denote the boundary cell complex associated to the foreground in $\bar{G}_0$. Let $\{\phi_p : C_p(K^0) \to C_{p+1}(K^0)\}_{p \geq 0}$ be the composition of all integral operators associated with all removals and contractions of edges of the foreground of the boundary graphs of a given irregular graph pyramid. Let $\{\pi_p = id_p + \phi_{p-1}\partial_p + \partial_{p+1}\phi_p : C_p(K^0) \to C_p(K^H)\}_{p \geq 0}$ where $\{\iota_p : C_p(K^H) \to C_p(K^0)\}_{p \geq 0}$ is the inclusion map.

Proposition 5.1. *The 1-cochain α_i^* defined by the set $\{\alpha_i, \beta\}$ in K^H is a 1-cocycle. Moreover, the set $\{\alpha_1^*, \dots, \alpha_n^*\}$ is a basis of representative 1-cocycles.*

Proof. The set of edges of K^H is the set of the self-loops $\{\alpha_1, \dots, \alpha_n\}$ surrounding a region of the background together with the self-loop β, renamed by α_{n+1}, surrounding the face f.

The 1-cochain α_i^* is a cocycle in K^H iff $\delta_1(\alpha_i^*) = 0$. Since we work with objects embedded in $\mathbb{R}^2$ then α_i can only be in the boundary of two faces. In this case, one face belongs to the background and the other face is f in K^H. Then $\delta_1(\alpha_i^*)(f) = \alpha_i^*(\partial_2(f)) = \alpha_i^*(\alpha_i + \beta + \cdots) = \alpha_i^*(\alpha_i) + \alpha_i^*(\beta) + \alpha_i^*(\cdots) = 1 + 1 + 0 = 0$.

Let us prove minimality. Suppose, for example, that $\alpha_1^* = \alpha_{j_1}^* + \cdots + \alpha_{j_s}^*$ where $1 < j_1 < \dots < j_s \leq n$, $s \geq 1$. Then α_1^* is defined by the set $\{\alpha_{j_1}, \dots, \alpha_{j_s}\}$ if s even, and $\{\alpha_{j_1}, \dots, \alpha_{j_s}, \beta\}$ if s odd, which is a contradiction. $\qquad\square$

We will say that α_i^* is a representative 1-cocycle of the cohomology generator associated to the white hole O_i.

Algorithm 2 gives the proposed method to downproject a cocycle α^* from level k to level $k - 1$. Informally, in the homology-generator level, there is only one face representing the object. Based on the geometric interpretation of cocycles (Section 3), if we remove the edges and the face in-between, we destroy the hole. Then, there is no need to add any other edge to the cocycle to remove the hole. However, when going down in the pyramid, this face

---
Algorithm 2 Down project cocycle

For each connected component (region) of K^H. Let A_k, $k > 0$, denote the set of edges that define a cocycle in $\bar{G}_k$ (the boundary graph in level k). The *down projection* of A_k to $\bar{G}_{k-1}$ is the set of edges $A_{k-1} \subseteq \bar{G}_{k-1}$ that corresponds to A_k i.e. represents the same cocycle. A_{k-1} is computed as $A_{k-1} = A_{k-1}^s \cup A_{k-1}^r$, where A_{k-1}^s denotes the set of surviving edges in $\bar{G}_{k-1}$ that correspond to A_k, and A_{k-1}^r is a subset of removed edges in $\bar{G}_{k-1}$. The following steps show how to obtain A_{k-1}^r:

1. Consider the contraction kernels of G_{k-1} (RAG) whose vertices are labeled with ℓ (the region for which cocycles are computed). The edges of each contraction kernel are oriented toward the respective root - each edge has a unique starting vertex.

2. For each contraction kernel T, from the leaves of T to the root, let e be an edge of T, v its starting point, and E_v the edges in the boundary of the face associated to v: label v with the sum of the number of edges that are in both A_{k-1}^s and in E_v, plus the labels of the children nodes of v.

3. A removal edge of $\bar{G}_{k-1}$ is in A_{k-1}^r if the starting point of the corresponding edge of G_{k-1} is labeled with an odd number.
---

is partitioned. A connection among all the new regions is determined by the contraction kernels of the RAG. When the first partitioning occurs, the contraction kernel will contain one or two nodes corresponding to faces with one surviving cocycle edge in its boundary, and the rest of the nodes will have none. What Algorithm 2 does is to *find the unique path in the contraction kernel joining these two nodes, and take the set of boundary edges between consecutive faces as part of the new cocycle (see Prop. 6.2, 6.3, and their proofs).* Lower levels will update the connections in subsections of the cocycle path. Every subsection will correspond to a partitioned region between two consecutive cocycle edges.

Consider the example in Fig. 8. In the homology-generator level we have $A_5 = \{\alpha, \beta\}$ the representative 1-cocycle of a cohomology generator (self-loops in Fig. 8.d). For down projection in level 4, $A_4 = A_4^s \cup A_4^r$. We have that A_4^s is the surviving edges in bold of top level in Fig. 8.e). In this case, $A_4^r = \emptyset$ because there is no merging of foreground regions from the boundary cell complex obtained from the top level to the homology-generator level.

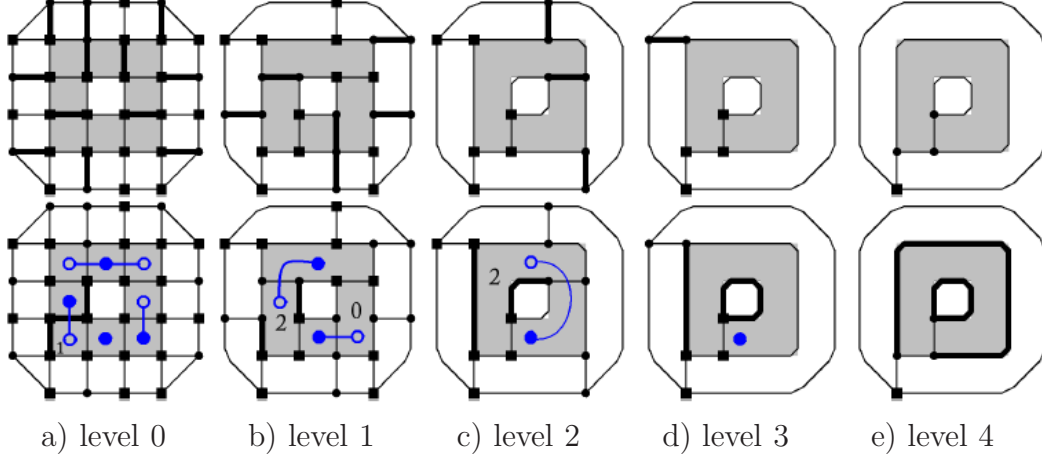| a) level 0 | b) level 1 | c) level 2 | d) level 3 | e) level 4 |

Figure 10: Top row: removed edges in boundary graph. Bottom row, from right to left: the down-projected cocycles in bold. Filled circles on faces, represent surviving vertices from the adjacency graph in foreground regions.

In the example in Fig. 10, the cocycle α^* in level 4 is the set of the two edges in bold (see Fig. 10, bottom row, column d). The down projection from level 4 to 3 are the surviving edges of the cocycle in level 4. This is because there was no contraction in the foreground region. The contractions of the adjacency graph can be seen in the top row of the figure.

In level 2 (Fig. 10, bottom row, column c), the first contraction of foreground in the adjacency graph with a single edge appears. In this case, the leaf node represents a face with an even number (2) of surviving cocycle edges in its boundary, which leads to not adding any other edge to the down-projected cocycle. Only in the base level (column a), one contraction kernel has a leaf node with an odd number (1) of surviving cocycle edges in the boundary. In this case, the corresponding edge in the boundary graph, for the edge connecting the respective node with its father, is added to the cocycle.

Any edge $\bar{G}_0$ that has survived to a higher level k, and was selected as part of the cocycle in $\bar{G}_k$, will belong to the down-projected cocycle in $\bar{G}_0$. In particular, the edges in $\bar{G}_0$ that have survived to be the edges α, β of the cocycle in the homology-generator level, are going to be the *entry* and *exit* point of the cocycle path through the foreground region.

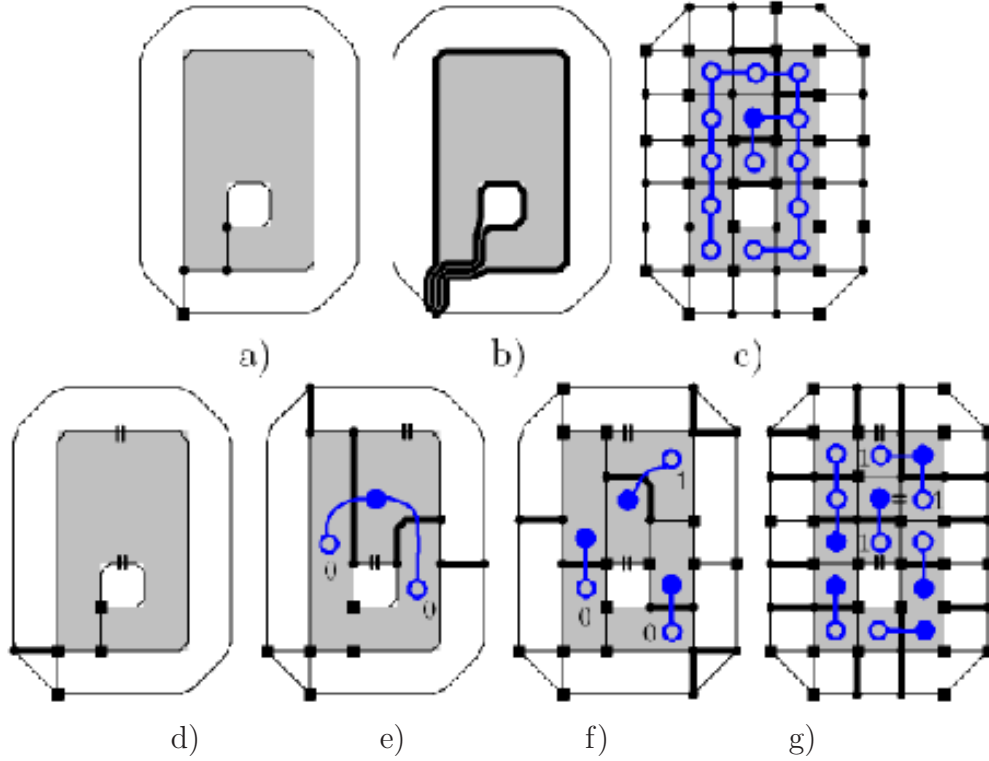In Fig. 11, the space between the outside boundary of the object and the

18

Figure 11: a) Boundary cell complex obtained from the top level of the pyramid, $(G_4, \bar{G}_4)$; b) homology-generator level with cocycles edges in bold; c) ECK of the foreground in RAG, drawn over $\bar{G}_0$ with cocycle edges in bold; d) to g) shows removed edges in bold for levels from 3 to 0. Cocycle edges are marked with two small parallel lines.

hole is bigger, allowing for more possibilities for the paths of the cocycle. The cocycle path in the base is going to converge to the unique path connecting the surviving edges α, β through the ECK (see Fig. 11.c).

In Fig. 11.d the cocycle is made of the surviving cocycle edges from the homology-generator level in Fig. 11.b . The first partition of the foreground is connected by the contraction kernel in Fig. 11.e . Here, one of the regions in the partition contains in its boundary the two surviving cocycle edges, so there is no path to find and no new cocycle edge to add. In Fig. 11.f there is only one edge to add to the cocycle to connect the path of edges, which is identified with the leaf node with label 1.

Notice that from level Fig. 11.f to the one in Fig. 11.g also the selected 'surviving' edges play a role, as edges to be removed at higher levels had to

have been 'surviving' ones at levels below. In this case, the edge we add to the cocycle in level 1 (Fig. 11.f), was a surviving edge in level below that was connecting two contraction kernels. Therefore, the selection of the surviving edges also determine the delineation of the down-projected cocycle together with the contracted ones. The ECK contains the decision of which edges were contracted but also which ones were simplified, determining the unique path.

Proposition 5.2. *1. The down projection of α_i^* is a set of edges 'blocking' the creation of the hole O_i, i.e., given a cycle g homologous to the down projection of the cycle α_i, the down projection of α_i^* contains an odd number of edges of g.*

2. The down projection of α_i^* is always a cocycle. Moreover, the down projection of $\{\alpha_1^*, \ldots, \alpha_n^*\}$ is a basis of representative 1-cocycles.

Proof.

1. The down projection of α_i^*, which is $\alpha_i^* \pi_1$, contains an odd number of edges of g iff $\alpha_i^*(g) = 1$. First, if g is homologous to the down projection of α_i, which is $\iota_1(\alpha_i)$, then there exists a 2-chain b in K_0 such that $g = \iota_1(\alpha_i) + \partial_2(b)$. Second, $\alpha_i^* \pi_1(g) = \alpha_i^* \pi_1(\iota_1(\alpha_i) + \partial_1(b)) = 1$, since $\alpha_i^* \iota_1(\alpha_i) = \alpha_i^*(\alpha_i) = 1$, and $\alpha_i^* \pi_1 \partial_2(b) = 0$ because α_i^* is a cocycle and $\pi_1 \partial_2 = \partial_1 \pi_2$ (since $\{\pi_p : C_p(K^0) \to C_p(K *_H)\}_{p \geq 0}$ is a chain equivalence [10]). So g must contain an odd number of edges of the set that defines α_i^*.

2. Proof of correctness of the down projection algorithm: it is a consequence of Lemma 4.4. □

Example down projections are shown in Fig. 9, 10, and 11.

5.1. Complexity

Let n be the height of the pyramid (number of levels) and v_0, e_0 the number of vertices, respectively edges in the base level, with $n \approx \log v_0$ (logarithmic height). An upper bound for the computation complexity is: $O(v_0 n)$ to build the pyramid; for each foreground component, $O(h)$ in the number of holes h to choose the representative cocycles in the top level; $O(e_0 n)$ to down project each cocycle. The overall computation complexity

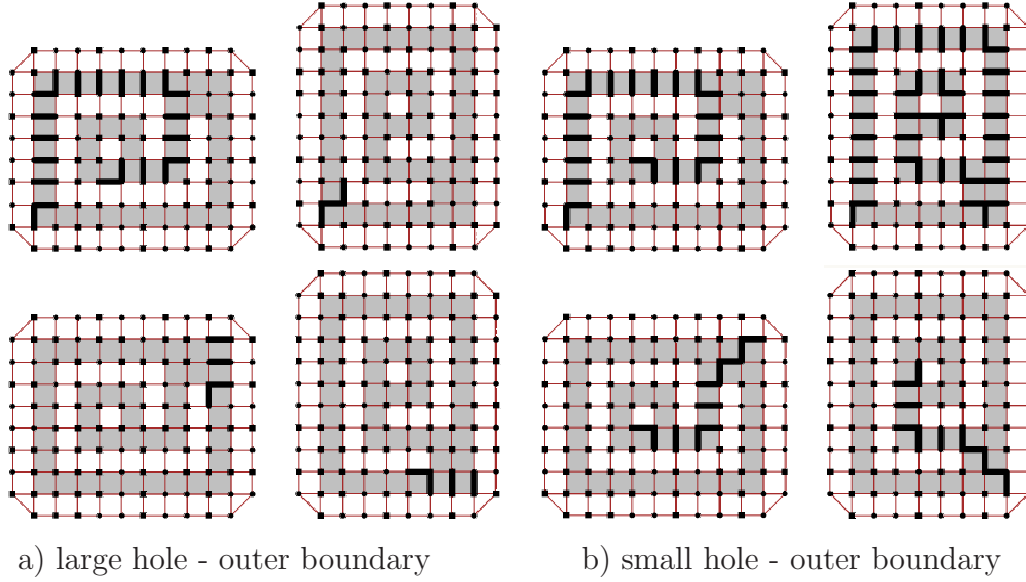a) large hole - outer boundary   b) small hole - outer boundary

Figure 12: Example showing (top) normal and (bottom) rotation invariant cocycles. The cocycles are down-projected starting with the surviving edge of the outer boundary and the surviving edge of the a) large and b) small holes.

is then below $O(v_0 n + c(he_0 n))$, where c is the number of cocycles that are computed and down projected.

Actually not all edges are part of cocycles and not all levels have e_0 number of edges. When building A_{k-1} one can go in linear time over the edges of A_k and consider only the contraction kernels in G_{k-1} for which the surviving vertices in G_k correspond to one of the two faces to which an edge of A_k is incident to. Then, computing A_{k-1} actually takes $|A_k| + \sum_i |T_{k-1}^i|$ number of steps, where $T_{k-1}^i \in G_{k-1}$ are the contraction kernels mentioned before. Thus in practice the complexity of down projecting a cocycle is below $O(e_0 n)$.

6. A First Step Towards Stable Cocycles

If topology is considered in the context of recognition, or a joint extraction of both topological and geometrical features is required, then the location and shape of the extracted topological invariants becomes relevant and is an important way to ensure stability/repeatability. A relevant example is the

work in [14] where handle and tunnel loops are made geometry aware by placing them on "geometrically relevant" positions.

In this section we make a first stept towards obtaining stable cocycles and consider invariance with respect to scanning and rotation of the object. Because we start with a pixel grid and the 4 neighborhood,only rotations with multiple of 90 degrees produce identical discretizations allowing for identical cocycles.

The following properties are required for Prop. 6.4 which gives the parts of the pyramid that the computed cocycles depend on.

Proposition 6.1. *The down-projected cocycles contain only removal edges in the boundary graphs, corresponding to edges in the ECK of the top vertex representing the object in G_n, and the two edges α, β that have survived to $\bar{G}_n$ and where selected as the cocycle $\alpha^* = \{\alpha, \beta\}$.*

Proof. Algorithm 2 starts with two edges $\{\alpha, \beta\}$ in $\bar{G}_n$, and for each level $k = n - 1, \ldots, 0$ it adds only removal edges from $\bar{G}_k$ i.e. edges that where contracted in G_k to merge neighboring regions belonging to the object. $\square$

Proposition 6.2. *The result of down projection (Algorithm 2) does not depend on the selected surviving vertices in G_k.*

Proof. Consider the function $q : T \subseteq G_{k-1} \to \mathbb{N}$, $q(T) = \sum_{v \in T} |A^s_{k-1} \cap E_v|$ (see Alg.2 for the used notation). Every cocycle A_k has an even number of edges from the boundary of any face in $\bar{G}_k$ (Section 3). Then $q(T)$ is also even and the number of vertices v for which $|A^s_{k-1} \cap E_v|$ is odd, is even. For any edge $e \in T = (V, E)$ consider the two connected components (trees) T_1, T_2 of the subgraph $(V, E \setminus \{e\})$ (e is a cut edge of T because T is a tree). The removal edge of $\bar{G}_{k-1}$ corresponding to the edge e is added to the cocycle if $q(T_1)$ and $q(T_2)$ are even, which is independent of the originally chosen surviving vertex, the root of T. $\square$

Proposition 6.3. *A down-projected cocycle does not depend on the order in which edges are removed in the boundary graphs (region merging).*

Proof. Consider the proof of Prop. 6.2. What Algorithm 2 does is to select additional edges from the contraction kernel T, to connect the vertices v with odd $|A^s_{k-1} \cap E_v|$. In a tree, there is a unique path connecting any two vertices. Denote by K the ECK of the vertex in G_n corresponding to the
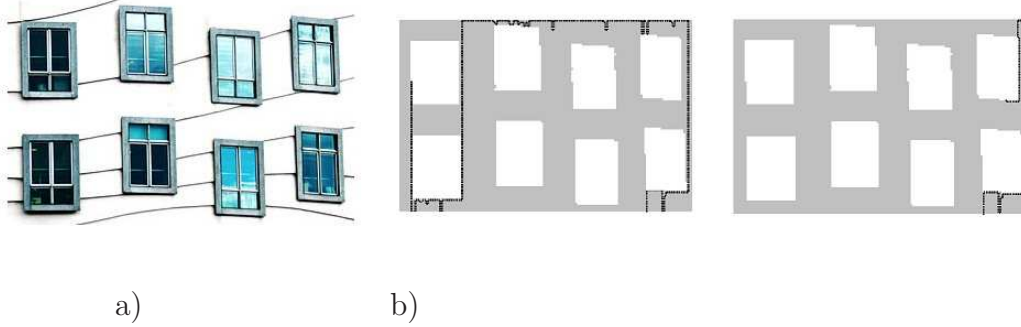
22

a)   b)

Figure 13: a) original image; b) in bold, the paths in the RAG G_0 associated to the down-projected cocycles related to the holes representing the top-left and top-right windows.

object. Algorithm 2 returns α and β, plus the set of edges of $\bar{G}_0$ corresponding to the path in K that connects the two vertices whose corresponding faces in $\bar{G}_0$ have α and β in their boundary. The ECK of a vertex does not depend on the order of the intermediate steps [15]. $\square$

The following property is an immediate result of Properties 6.1, 6.2, and 6.3.

Proposition 6.4. *The cocycles computed by Algorithm 2 depend only on the cocycle $\{\alpha, \beta\}$ chosen in the top level, $(G_n, \bar{G}_n)$, and on the ECK of the vertex in G_n corresponding to the face describing the object in $\bar{G}_n$.*

The following property results from Prop. 6.4 and motivates the modification proposed in the rest of this section.

Proposition 6.5. *If the ECK of the vertex representing the object in the adjacency graph of the homology-generator level, and the edges that survive to be in the boundary of the corresponding face in the boundary graph, are scanning and rotation invariant, we will obtain scanning and rotation invariant cocycles.*

In the following we will consider the necessary additions to the pyramid building process, to ensure that computed cocycles do not depend on the scanning and rotation of the object. We follow Prop. 6.5 and consider the ECK and the boundary edges of the top level.
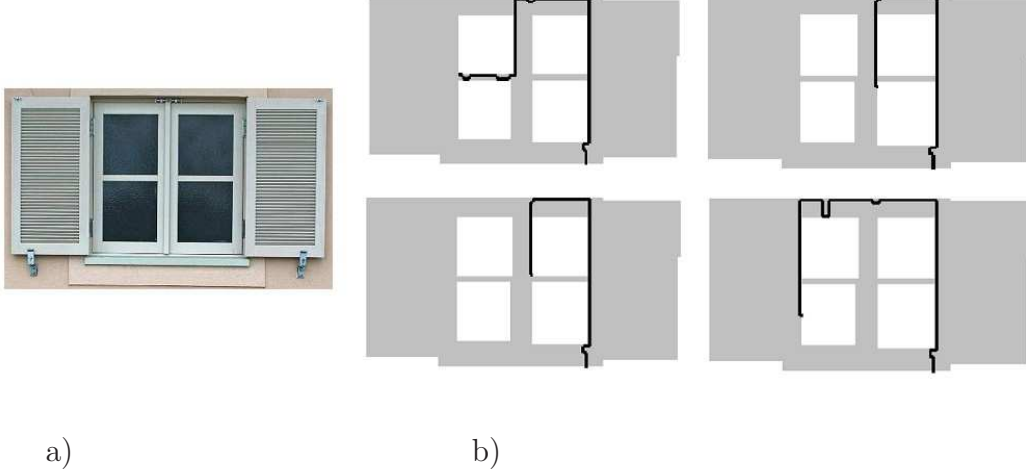
23

a)   b)

Figure 14: a) original image; b) in bold, the paths in the RAG G_0 associated to the down-projected cocycles.

6.1. Invariant ECK and surviving edges

As the edges to be removed in $\bar{G}_k$ are 'locally' chosen, in a binary image like the ones used to represent our objects, there is no local structure and a random or scanning/orientation dependent direction is taken (Line 4 of Algorithm 1).

To 'add structure' and create an ordering for selecting edges to be removed and contracted:

1. Compute a *spanning tree* of the subgraph $O \subset G_0$ corresponding to our object in the base level. Mark the *edges of $\bar{G}_0$ corresponding to the edges of the spanning tree to be the removed* ones.
2. Create a *strict ordering between any two edges*. This ordering is used to select surviving edges during simplification, and thus controls the choice of edges in the homology-generator level.

The tree. Given the graph O corresponding to our object, and a vertex $s \in O$, we define $d(v) : O \to \mathbb{N}$ to be the number of edges of the shortest path connecting v and s in O i.e. the geodesic distance between the two pixels corresponding to v and s, using the 4 neighborhood. Note that a vertex s can be obtained in a rotation invariant manner for example by using an automatic shape orientation method [16] and then selecting the top, left-most vertex.

24

Stable ECK (in G_0). Every vertex $v \in G_0$, $v \neq s$, labels the edge $(v, v') \in G_0$ with $d(v') = d(v) - 1$ as 'to contract'. If v has more than one neighbor v' with $d(v') = d(v) - 1$ the neighbor that minimizes the angle $\widehat{SVV'}$, and in case of angle equality, the one that has a clockwise orientation of SVV' is chosen. S, V, and V' are the points in $\mathbb{Z}^2$ corresponding to the centers of the pixels represented by s, v, and v'. The edges in $\bar{G}_0$ corresponding to the edges of G_0 labeled as 'to contract', are marked as 'to remove'.

Region boundary simplification (in $\bar{G}_k$). In Line 5.iii) of Algorithm 1, from any chain of edges bounded by at least one vertex of degree 2, one edge will survive and all others will be contracted. To choose to surviving edge, assign to each edge $e \in \bar{G}_0$ bounding a cell of the object, the value $f(e) = min\{d(v_1), d(v_2)\}$ where v_1, v_2 are the vertices of G_0 corresponding to the two faces of $\bar{G}_0$ to which e is incident to. Faces not part of the object are ignored. When choosing the edge to survive i.e. not contract, the edges are sorted using the following (transitive) relation between any two edges e and e':

- $f(e)$ vs. $f(e')$;

- if $f(e) = f(e')$ then use the orientation of $c(e) \, S \, c(e')$ vs. the orientation of $c(e') \, S \, c(e)$, where $c(e), c(e') \in \mathbb{R}^2$ are the centers of the edges e and e', and $S \in \mathbb{Z}^2$ is the center of the pixel used to define the rotation invariant tree;

- if $c(e) \, S \, c(e')$ are collinear, the Euclidean distance between $c(e)$ and S vs. the Euclidean distance between $c(e')$ and S.

Homology-generator level. When building the homology-generator level, all edges of $\bar{G}_{n-1}$ bounding the face of the object are sorted based on the criteria above. Edges are selected in inverse order and used to create the spanning tree to be contracted. Edges not bounding the face corresponding to the object are added in random order.

Fig. 12 shows an example object and its computed cocycles with and without the rotation invariant pyramid. Fig. 13 and 14 show the paths in the RAG associated to the down-projected cocycles of the test images in [12]. Fig. 15 is another example showing the path in the RAG G_0 associated to the down-projected cocycle related to the hole associated to the ball. Finally, Fig 16 shows the down-projected cocycles computed on a image from the 2010 World Cup final in South Africa.

Figure 15: In bold, the path in the RAG G_0 associated to the down-projected cocycle.

6.2. Discussion

Besides stability with respect to geometric transformations, one could consider additional criteria like the minimality of the obtained cocycles w.r.t. additional measures, like for example the number of edges of each cocycle, the sum of the number of edges of a basis of representative cocycles, the length of the path in $\mathbb{R}^2$ passing through the support squares of the pixels having at least one cocycle in their boundary[3], etc.

It has been shown for homology generators [17] that in general the problem of computing minimal representative cycles is NP-hard. Nevertheless, in certain cases, like computing $(n-1)$-cycles for n dimensional objects, finding minimal cycles is not NP-hard. For cocycles such a study does not exist yet, but considering the relation between homology and cohomology, similar results can be expected. In the case of 2D objects, the problem of finding cocycles with minimal number of edges can be related to the problem of finding in the RAG shortest paths that connect vertices adjacent to different holes – this problem can be solved in $n \log n$ if using Dijkstra's algorithm [18].

Real life objects are typically obtained by using scanning devices of different type: 3D scanners, video cameras, CT, MRI, etc. One common issue

---

[3]Would give Euclidean rotation robust cocycles.

in all these cases is the presence of noise and being robust w.r.t. the possible deformations of a real object. A possible solution could be to define a robust basis of cocycles based on a function like the eccentricity transform [19] which is known to be robust w.r.t. noise and deformations. The eccentricity transform associates to each point of a shape the geodesic distance to the point furthest away. For a given starting point the geodesic distance function defines behind holes a set of points called the *cut locus* which can be reached in the same distance on multiple paths (going on both sides of the hole). In many cases, cutting a shape along these sets can produce a shape with less holes, which gives the same geodesic distance function for the same starting point. The eccentricity transform can be interpreted as the maximum over multiple geodesic distance propagations initiated at each point of the shape. As cocycles can be seen as 'cuts into the object' that 'kill' a hole, a cut which has the minimum effect on the eccentricity transform, could provide an avenue for selecting robust cocycles.

Finding associations between concepts in cohomology and graph theory will open the door for applying existing efficient algorithms (e.g. shortest path). The following lemma can be seen as a first step in this direction.

Lemma 6.6. *Any set of foreground edges in the boundary graph $\bar{G}_0$, associated to a path in the RAG G_0, connecting a hole O_i of the object with the (outside) background face, is a representative 1-cocycle cohomologous to the down projection of the 1-cocycle α_i^*. It blocks any generator that would surround the hole.*

In other words, consider the down projection [12] of α_i and β in $\bar{G}_0$: the 1-cycles $\iota_1(\alpha_i) = a$ and $\iota_1(\beta) = b$, respectively. Take any edge $e_a \in a$ and $e_b \in b$. Let f_a, f_b be faces of K^0, the boundary cell complex associated to the foreground in G_0 having e_a, respectively e_b, in their boundary. Let $v_0, v_1, \ldots, v_n$ be a simple path of vertices in G_0 s.t. all vertices are labeled as foreground. v_0 is the vertex associated to f_a, and v_n to f_b. Consider the set of edges $c = \{e_0, \ldots, e_{n+1}\}$ of $\bar{G}_0$, where $e_0 = e_a$, $e_{n+1} = e_b$, and e_ℓ, $\ell = 1 \ldots n$, is the common edge of the regions in $\bar{G}_0$ associated with the vertices v_{i-1} and v_i. c defines a 1-cocycle cohomologous to the down projection of the 1-cocycle α_i^*.

Proof. c is a 1-cocycle iff $c\partial_2$ is the null homomorphism. First, $c\partial_2(f_\ell) = c(e_\ell + e_{\ell+1}) = 1 + 1 = 0$. Second, if f is a 2-cell of K^0, $f \neq f_\ell$, $\ell = 0, \ldots, n$, then, $c\partial(f) = 0$. To prove that the cocycles c and $\alpha_i^* \pi_1$ (the down projection of α_i^* to the base level of the pyramid) are cohomologous, is equivalent to prove that $c\iota_1 = \alpha_i^*$. We have that $c\iota_1(\alpha) = c(e_b) = 1$ and $c\iota_1(\beta) = c(e_a) =$
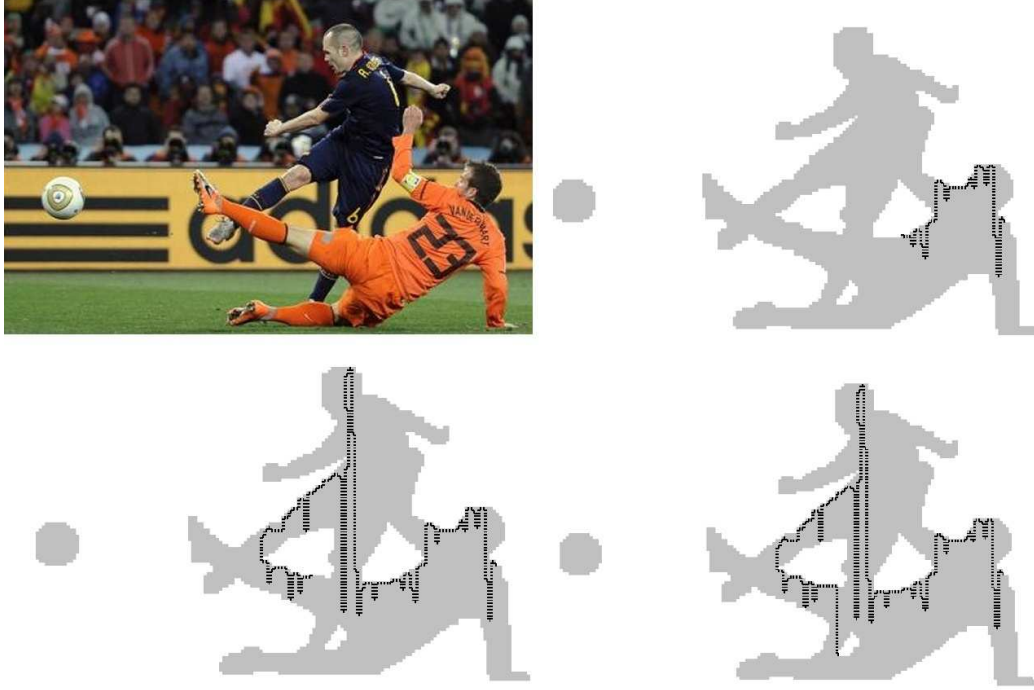
Figure 16: On the top-left, original image (a image from the 2010 World Cup final in South Africa); in bold, down-projected cocycles.

1. Finally, $c\iota_1$ over the remaining self-loops of the boundary graph of the homology-generator level is null. Therefore, $c\iota_1 = \alpha_i^*$. □

7. Conclusion

This paper considers cohomology in the context of graph pyramids. Representative cocycles are computed at the reduced top level and down projected to the base level corresponding to the original image. Connections between cohomology and graph theory are proposed, considering the application of cohomology in the context of classification and recognition. The current paper extends the previous work with detailed insights and proofs, and a refinement of the previous method that makes the obtained cocycles scanning and rotation invariant. Extension to higher dimensions, where cohomology has a richer algebraic structure than homology, and complete

cohomology - graph theory associations are proposed for future work. For this last task, we could consider nD generalized map pyramids [20].

8. Acknowledgements

References

[1] Z. J. Wood, H. Hoppe, M. Desbrun, P. Shroder, Removing excess topology from isosurfaces, ACM Trans. Graph. 23 (2) (2004) 190–208.

[2] M. Allili, D. Corriveau, Topological analysis of shapes using morse theory, Comput. Vis. Image Underst. 105 (3) (2007) 188–199.

[3] A. Hatcher, Algebraic topology, Cambridge University Press, 2002.

[4] R. Gonzalez-Diaz, A. Ion, M. Iglesias-Ham, W. G. Kropatsch, Irregular graph pyramids and representative cocycles of cohomology generators, in: A. Torsello, F. Escolano, L. Brun (Eds.), Proceedings of the 7th IAPR-TC15 International Workshop on Graph-based Representations in Pattern Recognition, Vol. 5534 of LNCS, Springer, Venice, Italy, 2009, pp. 263–272.

[5] R. Diestel, Graph Theory, no. 173 in Graduate Texts in Mathematics, Springer, 1997.

[6] W. G. Kropatsch, Building irregular pyramids by dual graph contraction, IEE-Proc. Vision, Image and Signal Processing 142 (6) (1995) 366–374.

[7] W. G. Kropatsch, C. Reither, D. Willersinn, G. Wlaschitz, The dual irregular pyramid, in: Proceedings CAIP'93, Computer Analysis of Images and Patterns, Lecture Notes in Computer Science, Vol. 719, Dmitry Chetverikov and Walter G. Kropatsch editors, Springer, 1993, pp. 31–40.

[8] D. Willersinn, W. G. Kropatsch, Dual graph contraction for irregular pyramids, in: ICPR1994: Proceedings of the 12th IAPR International

Conference on Pattern Recognition, Vol. 3, Shmuel Peleg, Shimon Ullman, and Yehezkel Yeshurun, editors, IEEE Comp.Soc., 1994, pp. 251–256.

[9] W. G. Kropatsch, Y. Haxhimusa, Z. Pizlo, G. Langs, Vision pyramids that do not grow too high, Pattern Recognition Letters 26 (3) (2005) 319–337.

[10] J. R. Munkres, Elements of Algebraic Topology, Addison-Wesley, 1993.

[11] R. Gonzalez-Diaz, M. J. Jimenez, B. Medrano, H. Molina-Abril, P. Real, Integral operators for computing homology generators at any dimension, in: 13th Iberoamerican Congress on Pattern Recognition (CIARP 2008), Vol. 5197 of LNCS, Springer, 2008, pp. 356–363.

[12] S. Peltier, A. Ion, W. G. Kropatsch, G. Damiand, Y. Haxhimusa, Directly computing the generators of image homology using graph pyramids, Image and Vision Computing 27 (7) (2009) 846–853.

[13] M. Iglesias-Ham, A. Ion, W. G. Kropatsch, E. B. García, Delineating homology generators in graph pyramids, in: 13th Iberoamerican Congress on Pattern Recognition (CIARP 2008), Vol. 5197 of LNCS, Springer, 2008, pp. 576–584.

[14] T. K. Dey, K. Li, J. Sun, D. Cohen-Steiner, Computing geometry-aware handle and tunnel loops in 3d models, ACM Trans. Graph. 27 (3) (2008) 1–9.

[15] W. G. Kropatsch, Equivalent contraction kernels to build dual irregular pyramids, Advances in Computer Science Advances in Computer Vision (1997) 99–107.

[16] J. D. Zunic, L. Kopanja, J. E. Fieldsend, Notes on shape orientation where the standard method does not work, Pattern Recognition 39 (5) (2006) 856–865.

[17] C. Chen, D. Freedman, Hardness results for homology localization, in: Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2010, pp. 1594–1604.

[18] M. J. Atallah (Ed.), Algorithms and Theory of Computation Handbook, 1st Edition, CRC-Press, 1998.

[19] A. Ion, W. G. Kropatsch, E. Andres, Euclidean eccentricity transform by discrete arc paving, in: D. Coeurjolly, I. Sivignon, L. Tougne, F. Dupont (Eds.), 14th IAPR International Conference on Discrete Geometry for Computer Imagery (DGCI), Vol. LNCS 4992 of Lecture Notes in Computer Science, Springer, Lyon, France, 2008, pp. 213–224.

[20] C. Grasset-Simon, G. Damiand, P. Lienhardt, nd generalized map pyramids: Definition, representations and basic operations, Pattern Recogn. 39 (4) (2006) 527–538.