Characterizing obstacle-avoiding paths using cohomology theory

Paweł Dłotko[1], Walter G. Kropatsch[2] and Hubert Wagner[1]

[1] Institute of Computer Science, Jagiellonian University, Poland
hubert.wagner@ii.uj.edu.pl and dlotko@ii.uj.edu.pl,
[2] Pattern Recognition and Image Processing Group, Vienna University of Technology, Austria
krw@prip.tuwien.ac.at

Abstract. In this paper, we investigate the problem of analyzing the shape of obstacle-avoiding paths in a space. Given a d-dimensional space with holes, representing obstacles, we ask if certain paths are equivalent, informally if one path can be continuously deformed into another, within this space. Algebraic topology is used to distinguish between topologically different paths. A compact yet complete *signature* of a path is constructed, based on cohomology theory. Possible applications include assisted living, residential, security and environmental monitoring. Numerical results will be presented in the final version of this paper.

Keywords: obstacle-avoidance, cohomology generators, trajectory planning problem

1 Introduction.

In the recent years, there has been growing interest in topics such as assisted living, residential, security and environmental monitoring [1, 2]. This is closely related to the area of remote sensing, which aims at delivering a description of the chosen aspects of the sensed environment by aggregating information from an array of sensors.

The information gathered by individual sensors ranges from visual data (Visual Sensor Networks [3]) to the presence of smoke in the air. Visual Sensor Networks are the most closely related to the computer vision field. In this paper we treat the sensors in an abstract way, therefore the method should be applicable in a number of settings.

One important question that arises is how to arrange such sensors. In [1], which largely inspired us to write this paper, straight laser beams are used as sensors. Prompted by some of the questions posed in the summary of that paper, we consider the following questions. How does this scenario generalize to sensors of different shapes? Can we generalize these concepts to higher dimensions (the original considerations were done in 2D)?

As a simple example, consider a network of paid highways. Since exact tracking the movement of each vehicle is prohibitively expansive, simplified measurements have to be performed. Gates serve as sensors, enabling us to roughly estimate the movement of the vehicle. While we fail to capture the precise *geometry* of the path of the vehicle, we are able to capture what we consider the *topology*.

This is closely related to the recent concept of *minimal sensing*, where sensors are very limited in their capabilities. In such a setting, sensors are typically unable to capture the actual geometry of the space. See [2] and references therein, to see how this problem was tackled, often using algebraic topology.

While the above example is trivial and can be described with basic graph algorithms, the situation is much more interesting (and challenging) in higher dimensions. Since our approach is based on algebraic topology, especially cohomology theory, it is dimension-independent.

Our additional aim is to to expose cohomology theory to the CAIP community. We believe that the mathematical robustness and intuitivity make it an interesting tool, which can be applied more generally.

The paper is structured as follows: In Section 2 a rigorous formulation of the considered problem is presented. In Section 3 the complexes used in this paper are discussed. In Section 4 an intuitive introduction to homology and cohomology theory is given. In Section 5 the main result of this paper is stated. In Section 6 an algorithm to compute signature of a given path is presented. Finally in Section 7 the conclusions are drawn.

2 Problem formulation.

We analyze movement, from point S to T, of a number of agents in a known space. Simply put, we ask how to place sensors, so that we are able to describe the *topology* of each path, based only on how it intersects these sensors. We encode these intersections as a *signature*, which is sufficient to discriminate between paths having different topology (more precisely: homology). We will prove that the sensors need to be placed in the support of *cohomology generators*.

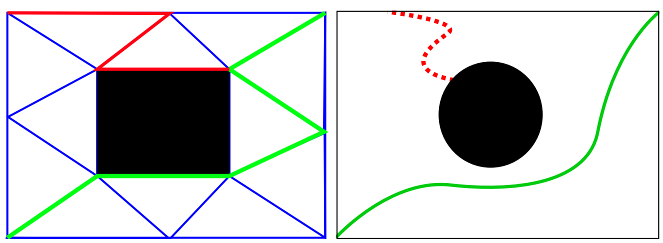
The problem of analyzing paths of moving agents in a 2-dimensional space, in the presence of obstacles and linear (beam) sensors was introduced in [1]. We present a variation for a d-dimensional, orientable space, where "sensors" are represented by certain $(d-1)$-dimensional hypersurfaces (possibly with self-intersections). For the $2-$dimensional case the difference is that our sensors can have arbitrary shape and are allowed to intersect. While the idea of a sensor of arbitrary shape might seem contrived, imagine that such a sensor is actually composed of a number of small sensing units covering a given hypersurface.

3 Representing spaces with holes.

In this section we present some theory related to computational topology, used later in the paper. For simplicity the concept of *simplicial complex* is used to represent the space. The definition of simplicial complex can be found in [4]. Imagine that a simplicial complex is a decomposition of the space into a set of simplices, that is vertices, edges, triangles etc. In general, $n-$simplex is a convex hull of $n+1$ points lying in general position. The number n is the *dimension* of a simplex S and is denoted by $dim(S)$. We assume that vertices of a simplicial complex are uniquely enumerated with integers, allowing to index each simplex with the set of its vertices. Each simplex in the simplicial complex has an orientation (this is discussed in details in [5]). In the implementation presented in [6], enumeration of vertices of complex $\mathcal{K}$ is used in orienting the edges and higher dimensional simplices. For instance every edge E is oriented from its higher vertex to lower vertex. From now on the orientation of all simplices in the complex is assumed to be fixed. A subset of simplices is chosen to represent the obstacles. During the computation of cohomology, the interior of obstacles is removed from the complex. Later by $\mathcal{K}$ we will denote the complex after this removal.

There are two vertices chosen in our complex, marked as S and T from Source and Target. An oriented path is the formal sum of edges joining those points with $+1, -1$ coefficients, which induce orientation.

The goal is to provide an efficient algorithm to describe and distinguish paths from S to T, which avoid all the obstacles[3]. An example of a $2-$dimensional simplicial complex can be found in Figure 1(a).



(a) (b)

Fig. 1. a) Simple example of a complex. Obstacles are marked with black, paths with green (solid). b) Graphical representation of complexes that we will use for clarity of images. Imagine that the complex is very finely subdivided, but paths and generators are still composed of edges of the complex, which is not displayed. Cohomology generator is depicted as the red (dotted) curve. In both cases point S is placed in the lower left, and point T in the upper right corner of the picture.

4 Cohomology theory.

In this section an intuitive exposition of homology and cohomology theory is given. For a full introduction consult [5]. Both homology and cohomology groups give a compact description of topology of a simplicial complex.

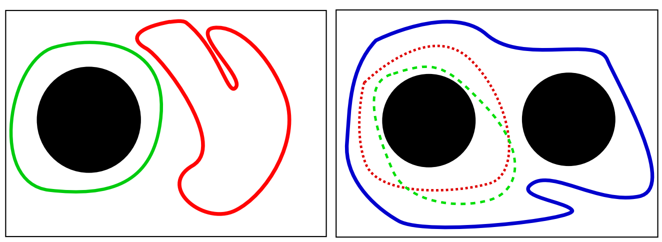
[3] Note that the number of homologically different paths is unbounded for non-trivial cases.

In homology theory one uses a concept of *chain*, being a formal sum of simplices with integer coefficients. A group of chains of dimension n is denoted by $C_n(\mathcal{K}) := \{\sum_{S \in \mathcal{K}, dim(S)=n} \alpha_S S\}$. A boundary operator $\partial : C_n \to C_{n-1}$ is then introduced for a simplex $S = [v_0, \ldots, v_n]$:

$$\partial S = \sum_{i=0}^{n} (-1)^i [v_0, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n] \tag{1}$$

and extended linearly to $C_n(\mathcal{K})$. As an example, let us calculate the boundary of a full triangle: $\partial[0,1,2] = [1,2] - [0,2] + [0,1]$.

A group of n dimensional *cycles* $Z_n(\mathcal{K}) := \{c \in C_n(\mathcal{K}) \mid \partial c = 0\}$. In short, a cycle is a chain whose boundary vanishes. A group of $n-$ dimensional *boundaries* $B_n(\mathcal{K}) := \{c \in C_n(\mathcal{K}) | \exists d \in C_{n+1}(\mathcal{K}) | \partial d = c\}$. The idea behind cycles and boundaries is presented in Figure 2(a).



(a) (b)

Fig. 2. a) Right chain is a cycle and a boundary. Left cycle surrounds a hole, so it is not a boundary b) Red (dotted) and green (dashed) cycles are homologous. Blue (bold) is not homologous with any of them. Red (or green) and blue cycles constitute a homology basis.

It is straightforward to verify from Formula 1 that $\partial\partial = 0$. Therefore we have $B_n(\mathcal{K}) \subset Z_n(\mathcal{K})$ and we can define the *homology group* $H_n(\mathcal{K})$ as a classes of cycles which are not boundaries, namely $H_n(\mathcal{K}) := Z_n(\mathcal{K})/B_n(\mathcal{K})$. Two n-cycles c_1 and c_2 such that $c_1 - c_2 \in B_n(\mathcal{K})$ are said to be *homologous*. By homology generators we mean any representants of classes of cycles that generate $H_n(\mathcal{K})$. In absence of *torsions* the rank of homology group can be interpreted as number of holes in the considered space. Idea of homology groups is given in Figure 2(b).

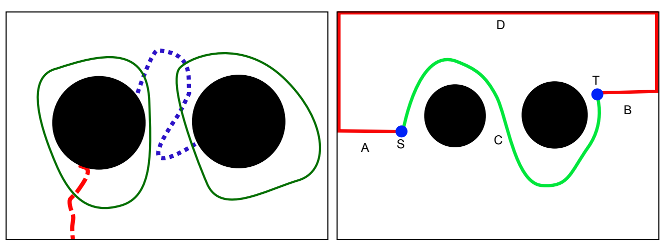
In this paper we restrict ourselves to connected simplicial complexes $\mathcal{K}$ which are torsion-free in dimension one (i.e. after the obstacles are removed from the complex, the resulting complex is connected and torsion free). Torsions in homology mean that elements of a homology group have finite order (they generate a subgroup $\mathbb{Z}_p$ of homology group for $p \in \mathbb{Z}$ being the order of an element).

For a formal introduction to the cohomology theory consult [5], for an intuitive introduction consult [6]. Further in the paper we need a concept of *n-cochain*

c^* being a map assigning any chain $c \in Z_n(\mathcal{K})$ a number[4] $\langle c^*, c\rangle \in \mathbb{Z}$. A group of $n-$cochains is denoted as $C^n(\mathcal{K})$. Dually to homology, a so-called *coboundary operator* $\delta : C^n(\mathcal{K}) \to C^{n+1}(\mathcal{K})$ is introduced. It is defined as $\langle \delta c^*, c\rangle = \langle c^*, \partial c\rangle$ for every $c^* \in C^{n-1}(\mathcal{K})$ and $c \in C_n(\mathcal{K})$. Again, cochain c^* is a *cocycle* if $\delta c^* = 0$. Cochain c^* is a *coboundary* if there exists a cochain $d^* \in C^{n-1}(\mathcal{K})$ such that $\delta d^* = c^*$. Cocycles are denoted as $Z^n(\mathcal{K})$, and coboundaries as $B^n(\mathcal{K})$. Finally, *cohomology group* is defined as the quotient $H^n(\mathcal{K}) := Z^n(\mathcal{K})/B^n(\mathcal{K})$.

It might appear that for torsion-free spaces all (co)homology computations could be performed with $\mathbb{Z}_p$ coefficients for $p \in \mathbb{Z}$, $p \geq 2$. This is not the case. Without going into details: we must use $\mathbb{Z}$ coefficients to handle the case of paths crossing certain cohomology generators np-times for $n \in \mathbb{Z}$.

For our purposes it is sufficient to consider cohomology group basis in dimension one. For torsion-free spaces, there is a straightforward correspondence between homology and cohmology group generators (see Theorem 4.8, [7]). Theorem 4.8 states that for any set of cycles representing homology generators $h_1, \ldots, h_n$ there exist dual cohomology generators $h^1, \ldots, h^n$ such that $\langle h^i, h_j\rangle = \delta_{ij}$. This theorem allows us to use the so-called "cutting analogy" to describe a cohomology basis. In fact, in the considered case the generator h^i, for $i \in \{1, \ldots, n\}$, can be seen as a fence that blocks any cycles in the class of h_i. This idea is illustrated in Figure 3(a). The concept of the presented "cut analogy" was developed in the so-called *Discrete Geometrical Approach* to Maxwell's equations [6].



(a) (b)

Fig. 3. a) The "cut analogy". When one cuts a complex along the red (dashed) cohomology generator, the left homology class vanishes. Cutting along blue (dotted) generator makes the right homology class vanish. b) Completion of chain c.

With the algorithm described in [6], we obtain cohomology generators (represented as a set of pairs (edge, integer)) of any simplicial complex. Note that

[4] Operation $\langle c^*, c\rangle$ is called evaluation of a cocycle c^* on a cycle c. In order to compute $\langle c^*, c\rangle$, note that the set of maps $\{S^* | \langle S^*, K\rangle = \delta_{SK}$ for any $K \in \mathcal{K}\}_{S\in\mathcal{K}}$ constitutes a basis of $C^n(\mathcal{K})$. Therefore every cochain c^* is equal to $\sum_{S\in\mathcal{K}} \alpha_S S^*$ for $dim\ S = n$. Then for a chain $c = \sum_{S\in\mathcal{K}} \beta_S S$ we have $\langle c^*, c\rangle = \langle \sum_{S\in\mathcal{K}} \alpha_S S^*, \sum_{S\in\mathcal{K}} \beta_S S\rangle = \sum_{S\in\mathcal{K}} \alpha_S \beta_S$.

cohomology generators are allowed to intersect. See the Borromean Rings phenomenon in [8] for an example of a 3-dimensional space, where it is impossible to find a non-intersecting cohomology basis.

5 Path characterization using signatures.

In this section a formal proof of the main result of the paper is provided. Suppose a simplicial complex $\mathcal{K}$ is given. As previously, we assume that $H_1(\mathcal{K})$ is torsion-free and $\mathcal{K}$ itself is connected. Let $h^1, \ldots, h^n$ be cocycles representing first cohomology group generators of $\mathcal{K}$. Moreover, let $h_1, \ldots, h_n$ be the homology generators dual to $h^1, \ldots, h^n$ according to Theorem 4.8 in [7] (they are only needed for the proof). We fix $h^1, \ldots, h^n$ and their dual $h_1, \ldots, h_n$ for the rest of this section. Let $c \in C_1(\mathcal{K})$ be a path from S to T.

Definition 1. *For a path c the vector $S_c = [a_1, \ldots, a_n]$ such that $a_i = \langle h^i, c\rangle$, for $i \in \{1, \ldots, n\}$, is called a* signature *of c.*

In this section we show that paths having the same signature are homologous and, conversely, that paths having different signature are non-homologous. It is necessary to assume that all paths lead from S to T. A signature of a path provides an efficient way of distinguishing non-homologous paths and identifying homologous ones. Let us start with a lemma, the proof of which can be found in [7].

Lemma 1. *Let $c^* \in Z^1(\mathcal{K})$ be a cocycle and let $b \in B_1(\mathcal{K})$ be a boundary. Then $\langle c^*, b\rangle = 0$.*

Let us now define the *completion* of a chain. Let us take any chain A joining point S with the boundary of the complex $\mathcal{K}$, B joining point T with the boundary of a complex and D lying entirely on the boundary of $\mathcal{K}$ joining endpoints of chains A and B. With any path $c \in C_1(\mathcal{K})$ from S to T we can assign a cycle $c \cup A \cup B \cup D$. This cycle is called a *completion* of chain c (see Figure 3(b)).

Now we are ready to give the two main theorems of this paper.

Theorem 1. *Two homologous paths c_1 and c_2 have the same signature, $S_{c_1} = S_{c_2}$.*

Proof. Since c_1 and c_2 are homologous, there exists $b \in C_2(\mathcal{K})$ such that $\partial b = c_1 - c_2$. Therefore $c_1 = c_2 + \partial b$. From Lemma 1 we have, that $\langle h^i, c_1\rangle = \langle h^i, c_2 + \partial b\rangle = \langle h^i, c_2\rangle + \langle h^i, \partial b\rangle = \langle h^i, c_2\rangle + 0 = \langle h^i, c_2\rangle$ for every $i \in \{1, \ldots, n\}$. Therefore $S_{c_1} = S_{c_2}$. □

Theorem 2. *Two non-homologous paths c_1 and c_2 have different signatures, $S_{c_1} \neq S_{c_2}$.*

Proof. Suppose by contrary that c_1 and c_2 are non-homologous and $S_{c_1} = S_{c_2}$. Therefore $d_1 = c_1 \cup A \cup B \cup D$ and $d_2 = c_2 \cup A \cup B \cup D$ are also non-homologous. But $h_1, \ldots, h_n$ is a homology basis dual to cohomology basis $h^1, \ldots, h^n$. Then we have $d_1 = \sum_{i=1}^n \alpha_i h_i + \partial e$ and $d_2 = \sum_{i=1}^n \beta_i h_i + \partial f$ for some $e, f \in C_2(\mathcal{K})$ and $\alpha_i, \beta_i \in \mathbb{Z}$ for $i \in \{1, \dots, n\}$. Since d_1 and d_2 are not homologous there exists an index $i \in \{1, \dots, n\}$ such that $\alpha_i \neq \beta_i$. But from the hypothesis we have $S_{c_1} = S_{c_2}$. It implies, that $S_{d_1} = S_{d_2}$. We have $\langle h^i, d_1 \rangle = \langle h^i, \sum_{i=1}^n \alpha_i h_i \rangle = \alpha_i$ and $\langle h^i, d_2 \rangle = \langle h^i, \sum_{i=1}^n \beta_i h_i \rangle = \beta_i$. Therefore from the hypothesis we have $\alpha_i = \beta_i$ for every $i \in \{1, \dots, n\}$, which gives a contradiction. $\square$

6 Computing the Signature of a path.

In this section we present an algorithm which, for fixed cocycles $h^1, \dots, h^n$, constituting a cohomology basis and a path c from A to B outputs S_c, the signature of c. We assume that simplicial complex is represented as a pointer-based data-structure as in [6]. Moreover, let each edge E of simplicial complex $\mathcal{K}$ be equipped with a vector v of n integers such that $v_E[i] = \langle h^i, E \rangle$ for every $i \in \{1, \dots, n\}$. Let a path c be given as a vector of pointers to edges in $\mathcal{K}$.

It remains to resolve the subtlety of orientation of simplices versus an orientation of a path c. The path is oriented from point S to T. Let us define $o(c, E)$ in the following way: $o(c, E) := 1$ if orientation of c is the same as orientation of E and -1 otherwise. Now we list the algorithm. Also, see Figure 4 for a visual example. Note that this two-dimensional example is very simple and can be solved with basic tools, but our method works for general dimension.

Algorithm 1 Computing signature of a path

Input: path c, simplicial complex $\mathcal{K}$ with cohomology generators $h^1, \dots, h^n$
Output: s - signature of path c

1: Let v be the vector encoding the intersections of c with cohomology generators
2: Let s be an n-tuple
3: **for** $i \in \{1, \dots, n\}$ **do**
4: $\quad s[i] \leftarrow \sum_{E \in c} o(c, E) v_E[i]$
5: **return** s

7 Conclusions

The ideas presented in this paper generalize the approach using "laser beams" presented in [1]. We use topological tools to distinguish between different obstacle-avoiding paths, based only on their intersections with selected *sensors*. The usage of algebraic topology enables us to use sensors of arbitrary shape and abstract away from the actual geometry of the space. Topological information (cohomology generators and their intersections with paths) sufficiently represents the space. Additionally, the usage of algebraic topology makes our method dimension-independent, which extends the area of applications.

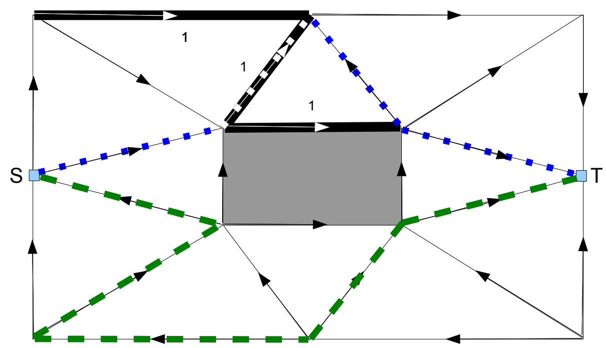


Fig. 4. We use the presented procedure to compute $s[1]$ for the blue (dotted) path. $v_E[1]$ is nonzero only for the edges in the support of the cohomology generator. Therefore $s[1] = 1$, as the orientation of this path is the same as the orientation of cohomology generator (bold black). As for the green (dashed) path $s[1] = 0$, since the path do not cross the cohomology generator. Blue (dotted) and green (dashed) paths are clearly non-homologous.

Acknowledgments.

P.D. was partially supported by MNiSW grant N N206 625439, H.W. by the Austrian Science Fund (FWF) grant no. P20134-N13 and Foundation for Polish Science IPP Programme "Geometry and Topology in Physical Models". P.D. and H.W. are partially funded by the Google Research Awards Program.

References

1. B. Tovar, F. Cohen and S. M. LaValle, "Sensor Beams, Obstacles, and Possible Paths", Proc. Workshop on the Algorithmic Foundations of Robotics (WAFR), 2008
2. V. de Silva and R. Ghrist, "Coverage in sensor networks via persistent homology", Alg. & Geom. Topology, 7, 339-358, 2007.
3. S. Soro and W. Heinzelman, "A Survey of Visual Sensor Networks", Advances in Multimedia vol. 2009, 2009
4. H. Edelsbrunner, Geometry and Topology for Mesh Generation, Cambridge University Press, 2001, ISBN 9780521793094.
5. A. Hatcher, Algebraic Topology, available online, Cambridge University Press 2002.
6. P. Dłotko, R. Specogna, "Efficient cohomology computation for electromagnetic modeling", CMES: Computer Modeling in Engineering & Sciences, Vol. 60, No. 3, 2010, pp. 247-278.
7. P. Dłotko, R. Specogna, "Critical analysis of the spanning tree techniques", SIAM Journal of Numerical Analysis (SINUM), Vol. 48, No. 4, 2010, pp. 1601-1624.
8. P. R. Cromwell, E. Beltrami and M. Rampichini, "The Borromean Rings", Mathematical Intelligencer 20 no 1 (1998) 5362.