Computing Fundamental Group of General 3-manifold

Junho Kim[1], Miao Jin[2], Qian-Yi Zhou[3], Feng Luo[4], and and Xianfeng Gu[5]

[1] Dong-Eui University, South Korea, kim.junho@deu.ac.kr
[2] University of Louisiana at Lafayette, USA, mjin@cacs.louisiana.edu
[3] University of Southern California, USA, Qian-YiZhou@usc.edu
[4] Rutgers University, USA, fluo@math.rutgers.edu
[5] Stony Brook University, USA, gu@cs.sunysb.edu

Abstract. Fundamental group is one of the most important topological invariants for general manifolds, which can be directly used as manifolds classification. In this work, we provide a series of practical and efficient algorithms to compute fundamental groups for general 3-manifolds based on CW cell decomposition. The input is a tetrahedral mesh, while the output is symbolic representation of its first fundamental group. We further simplify the fundamental group representation using computational algebraic method. We present the theoretical arguments of our algorithms, elaborate the algorithms with a number of examples, and give the analysis of their computational complexity.

Key words: computational topology, 3-manifold, fundamental group, CW-cell decomposition.

1 Introduction

Topology studies the properties of geometric objects which are preserved under continuous deformation. In biomedical fields, topology has been applied for classification and identification of DNA molecules, and topological changes are considered as indications of some important chemical changes [1, 2]. In CAGD field, a rigorous, robust, and practical method to compute the topologies of general solids, in order to improve the robustness and reliability of CAGD systems required for automation of engineering analysis tasks, is also prefered [3–8].

The computational algorithms for surface topology are mature [9–16], while computing the topologies of 3-manifolds still remains widely open. Both Homology group and fundamental group are important topological invariants for 3-manifold. Although homology group is much easier to compute than fundamental group, the price is that it conveys much less information than fundamental group [17]. In theory, all 3-manifolds can be canonically decomposed to a unique collection of *prime manifolds*, whose topologies are solely determined by their fundamental groups. Since most connected solids in the real world are prime 3-manifolds, computing the fundamental groups for 3-manifolds ia a practical solution to understand their topologies.

To the best knowledge of the author, there is no general practical system in engineering fields, which can verify whether two 3-manifolds are topologically equivalent. To tackle this problem, we present the first efficient algorithms to compute fundamental groups of general 3-manifolds represented by tetrahedral meshes. The computational complexity of our algorithms are greatly reduced by converting an input tetrahedral mesh (i.e., simplicial complex) to a CW complex [18].

2 Related Work

Computational topology has emerged as a very active field [1, 2, 19]. It is beyond the scope of this paper to give a thorough review. Here we only briefly review the most related works.

For a closed 2-manifold surface, the surface can be sliced into a simple polygon, which is called *polygonal schema*. Since the boundary of polygonal schema provides the loops for the fundamental group and the homology group of the surfaces, it has been intensively studied in the field of computational topology. Vegter and Yap [9] present an efficient algorithm on computing a canonical form of a polygonal schema from a closed 2-manifold mesh. From the canonical form they introduce the algorithm on computing the fundamental group for surface meshes. By using Seifert-van Kampen's theorem, Dey and Schipper [10] present a linear time algorithm for computing a polygonal schema of a 2-manifold mesh. Lazarus et al. [13] provide optimal algorithms for computing a canonical polygonal schema of a surface. Erickson and Har-Peled [14] show that it is NP-hard to get an optimal polygonal schema, which has the minimal boundary edge lengths. Colin de Verdière and Lazarus [15] provide the algorithm to find an optimal system of loops among all simple loops obtained from a canonical polygonal schema. Erickson and Whittlesey [20] introduce greedy algorithms to construct the shortest loops in the fundamental group or the first homology group. Yin et al. [16] compute the shortest loop in a given homotopy class by using universal covering space.

We utilize the idea of reducing the problem dimension [10, 11, 21] in the computation of fundamental groups for 3-manifold with CW cell decompositions. In contrast to the algorithms for computing homology groups [11, 21], this work focuses on computing fundamental groups, which convey much more topological information in the case of 3-manifolds. Furthermore, our method is general for 3-manifolds which even can not be embedded in $\mathbb{R}^3$, and is efficient for large tetrahedral meshes reconstructed from medical images.

3 Background

In out work, we convert input tetrahedral meshes, namely simplicial complexes, to CW complexes, then compute their fundamental groups, which are represented as symbolic representations. Here we only briefly introduce the concepts directly related with our algorithms. We refer readers to [18, 22, 23] for more details.

3.1 Simplicial Complex

The topology of tetrahedral meshes is typically represented as a simplicial complex: a set of 1, 2, 3 and 4-element subsets of a set of labels, corresponding respectively to the vertices, edges, triangles, and tetrahedrons of the mesh. A simplicial complex can be considered as just the connectivity part of a traditional tetrahedral mesh.

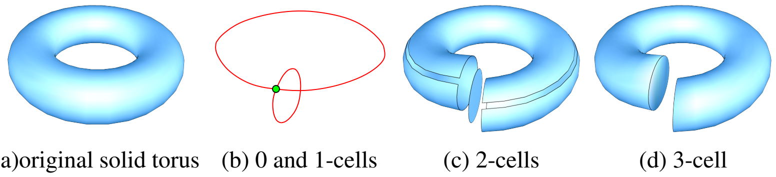

Fig. 1. *CW Complex of a solid torus*. The original solid torus in (a) is decomposed to one 0-cell, shown in (b) with green color, two 1-cells shown in (b) as red arcs, two 2-cells in (c) and one 3-cell in (d).

3.2 CW Complex

CW complex is a generalization of simplicial complex, introduced by J.H.C.Whitehead in 1949. Given a tetrahedral mesh, CW complex forms a topological skeleton of the mesh, which is far more flexible than using simplicial complex representation.

A topological space is called an *n-cell* if it is homeomorphic to $\mathbb{R}^n$. For example, a 0-cell is a point, a 1-cell is a space curve segment, a 2-cell is a surface patch and a 3-cell is a solid. This homeomorphism maps the boundary of a n-cell to the (n-1) sphere.

Given a 3-manifold M with one component, a Hausdorff topological space X is its CW complex if it can be constructed, starting from discrete points, by first attaching 1-cells, then 2-cells, then 3-cells, represented as

$$M^0 \subseteq M^1 \subseteq M^2 \subseteq M^3.$$

Each M^k is called the *k-skeleton*, obtained with attaching k-cells to a M^{k-1} by identifying the boundary of the k-cells with the union of some collection of (k-1)-cells in the complex. For example (shown in Figure 1), to construct a 3-dimensional CW-complex. We begin with the empty set. Then we attach 0-cells by unioning disjoint points into the set (shown in Figure 1(b)). We attach 1-cells by unioning space curve segments whose endpoints lie on these points (shown in Figure 1(b)). We attach 2-cells by unioning surface patches whose boundaries lie on the space curve segments (shown in Figure 1(c)). We attach 3-cells by filling in closed regions bounded by surface (shown in Figure 1(d)).

A particular choice of a collection of skeletons and attaching maps for the cells is called a CW structure on the space, which is not unique in general.

3.3 Fundamental Group

In a topological space X, we mean a *path* as a continuous map $f : I \longrightarrow X$ where I is the unit interval $[0,1]$. Two pathes f_0 and f_1 which share two end points (i.e., $f_0(0) = f_1(0)$ and $f_0(1) = f_1(1)$) are homotopic to each other, if one can be continuously deformed to another in X while two end points are kept during the deformation. All pathes each of which is homotopic to a path f is called a *homotopy class* of f, denoted by $[f]$. Given two paths $f,g : I \to X$ such that $f(1) = g(0)$, there is a *composition* $f \cdot g$ that traverses first f then g, defined by the formula

$$f \cdot g(s) = \begin{cases} f(2s), & 0 \leq s \leq \frac{1}{2} \\ g(2s-1), & \frac{1}{2} \leq s \leq 1 \end{cases}$$

In particular, suppose a path $f : I \to X$ is with the same starting and ending point $f(0) = f(1) = x_0 \in X$, then f is called a *loop*, and the common starting and ending point x_0 is referred to as *base point*.

Definition 1 (Fundamental Group). *The set of all homotopy classes $[f]$ of loops $f : I \to X$ at the base point x_0 is a group with respect to the product $[f][g] = [f \cdot g]$, which is called the* fundamental group *of X at the base point x_0, and denoted as $\pi_1(X, x_0)$.*

If X is path connected, then for any base points $x_0, y_0 \in X$, the fundamental groups $\pi_1(X, x_0)$ and $\pi_1(X, y_0)$ are isomorphic, therefore, we can omit the base point and denote the fundamental group as $\pi_1(X)$.

We represent the fundamental group as $< S; R >$. It is a free group generated by S, called the *generator* and represented as a set of non-commutative symbols. R is called the *relation*, represented as words formed using these symbols.

4 Algorithm

Given a 3-manifold represented by a simplicial complex (a tetrahedral mesh) M, our goal is to compute its fundamental group $\pi_1(M)$, represented as generators and relations $\langle S; R \rangle$. Considering that the number of the simplexes in M is in general high such that direct computation is prohibitively expensive, our algorithms will be built on CW complex representation of M instead of simplex representation. So the first step of our algorithms is to compute the CW cell decomposition of the input tetrahedral mesh M.

Then the following lemmas give the keys of the next steps of our algorithms for computing the generators and relations of M, which tell that they only depend on the 2-skeleton M^2 and the 1-skeleton M^1 of the CW complex representation of M. We refer readers to Appendix for the proof.

Lemma 1. *The fundamental group $\pi_1(M)$ is isomorphic to the fundamental group $\pi_1(M^2)$.*

Lemma 2. *The fundamental group $\pi_1(M^1)$ is a free group (only generators, no relations),*

$$\pi_1(M^1) = \langle \gamma_1, \gamma_2, \cdots, \gamma_n \rangle.$$

Suppose $M^2 = M^1 \bigcup \{\sigma_1^2, \sigma_2^2, \cdots, \sigma_{n_2}^2\}$, and σ_i^2 is 2-cell, then the boundary of each 2-cell $\partial \sigma_i^2$ is a loop in M^1. The fundamental group of M^2 has the form

$$\pi_1(M^2) = \langle \gamma_1, \gamma_2, \cdots, \gamma_n; [\partial \sigma_1^2], [\partial \sigma_2^2], \cdots [\partial \sigma_{n_2}^2] \rangle$$

where $[\partial \sigma_i^2]$ is the homotopy class of $\partial \sigma_i^2$ in $\pi_1(M^1)$, represented by a word formed by γ_k's.

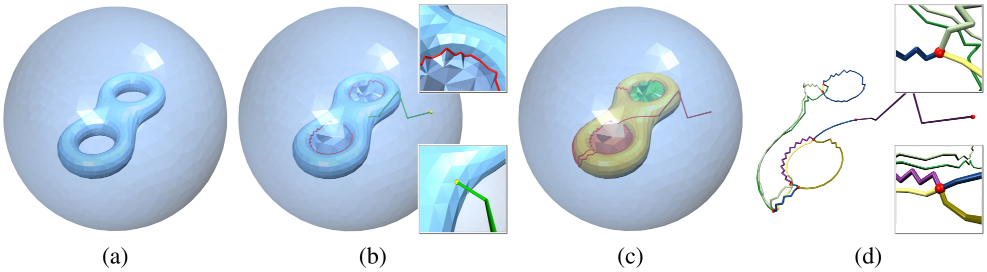

Fig. 2. CW cell decomposition: (a) an input non trivial tetrahedral mesh, obtained by removing a solid two hole torus from a solid sphere; (b) the 2-skeleton of the tetrahedral mesh; (c) the different 2-cells in the 2-skeleton illustrated with different colors; (d) the 1-skeleton of the tetrahedral mesh; vertices in the 1-skeleton whose valence is greater than two, belong to the 0-skeleton.

4.1 Computing CW Complex

Suppose M is a 3-manifold represented by a tetrahedral mesh. Our goal is to compute a CW complex

$$M^0 \subseteq M^1 \subseteq M^2 \subset M^3,$$

where M^k is the k-skeleton, obtained by attaching k-cells to M^{k-1}.

In the following discussion, the vertex, edge, triangle, and tetrahedron refer to simplicial complex. The algorithm starts with an input tetrahedral mesh M, which is equivalent to the 3-skeleton M^3. Initially, we set M^0, M^1, and M^2 as empty sets. Since $M^3 = M^2 \bigcup \{\sigma_1^3, \sigma_2^3, \cdots, \sigma_{n_3}^3\}$, where σ_i^3 is a 3-cell. Suppose Δ^3 is a tetrahedron in M, then Δ^3 must belong to a 3-cell σ_i^3. We merge all the tetrahedra sharing a face with Δ^3 to Δ^3 to form a bigger 3-cell. We keep growing this 3-cell until all the tetrahedra are exhausted or the 3-cell can not be extended further, then the 3-cell is σ_i^3. Then we select another tetrahedron in $M^3 \setminus \sigma_i^3$, and get another 3-cell. We repeat this process, until all tetrahedron are removed. Then what left is the 2-skeleton M^2.

The computation of 2-cells and 1-skeleton M^1 is very similar. We select a triangle $\Delta^2 \in M^2$. By growing Δ^2, we can find a 2-cell. By repeating removing 2-cells from M^2, we obtain M^1. All the vertices in M^1 whose valence is not equal to 2 form 0-skeleton M^0. The connected components of $M^1 \setminus M^0$ are 1-cells.

Algorithm 1 gives the general procedure to get a k-skeleton from a (k+1)-skeleton.

Figure 2 shows an example of the CW cell decomposition from an input non trivial tetrahedral mesh.

4.2 Computing Generators

According to propositions 1 and 2, the generators of $\pi_1(M)$ is equivalent to the the generators of $\pi_1(M^1)$. To compute the generators of $\pi_1(M^1)$, we can treat the 1-skeleton M^1 as a graph G by considering the 0-cells as nodes and 1-cells as edges. Then the generators of $\pi_1(M^1)$ are simply those loops, whose compositions can generate all possible loops in G.

Algorithm 1 Computing CW Complex

```
Set a randomly picking (k+1) simplex $\Delta^{k+1}$ as seed;
Set seed as a (k+1)-cell $\sigma^{k+1}$ and marked;
queue += all the (k+1) simplexes sharing a face with seed;
repeat
  repeat
    Pop a (k+1) simplex $\Delta^{k+1}$ out of queue;
    Let $\tau$ be the common face of $\Delta^{k+1}$ and $\sigma^{k+1}$, where $\partial\Delta^{k+1} \cap \partial\sigma^{k+1} = \tau$.
    if $\Delta^{k+1}$ has not been marked then
      Grow the (k+1)-cell $\sigma^{k+1}$ by including $\Delta^{k+1}$ and $\tau$;
      Mark $\Delta^{k+1}$;
      queue += all the (k+1) simplexes sharing a face with $\Delta^{k+1}$ while not marked;
    end if
  until queue is empty
  Shrink current (k+1)-skeleton by removing this (k+1)-cell $\sigma^{k+1}$, such that
                    $M^{k+1} \leftarrow M^{k+1} - \sigma^{k+1};$
  if Some (k+1) simplexes $\Delta^{k+1}$ in $M^{k+1}$ not marked then
    Set one of them as seed and marked;
    queue += all the (k+1) simplexes sharing a face with seed and not marked;
  end if
until queue is empty
A k-skeleton $M^k$ is obtained by removing all (k+1)-cells from $M^{k+1}$;
```

The algorithm is as following. First, we compute a minimal spanning tree T of G. Then, the set of edges in G is partitioned into the set of edges in T and the set of *non-tree edges* in $G \setminus T$. Let e^T and $e^{\neg T}$ be an edge in T and a non-tree edge in $G \setminus T$. When we union each non-tree edge $e_i^{\neg T}$ with T, a unique loop γ_i is generated in G. From the property of the spanning tree of a graph, the set of $\{\gamma_1, \gamma_2, \cdots, \gamma_n\}$ is the generators for G, where n is the number of non-tree edges in G. Moreover, the set $\{\gamma_1, \gamma_2, \cdots, \gamma_n\}$ can be considered as the generators of $\pi_1(M)$ from the propositions.

4.3 Computing Relations

Since the boundary of each 2-cell is a loop in the 1-skeleton, they must come from the concatenations of the generators of $\pi_1(M^1)$ we just computed, and can be represented as a word w. Considering the boundary loop of any 2-cell can be shrunk to a point in M, the word w must equal to e. Therefore, w is a relation.

The following gives the algorithm based on the graph G and tree T computed from previous algorithm Sec. 4.2. We first give an arbitrary orientation for each edge e_i in G. The symbol e_i^{-1} represents the opposite direction for the orientation of e_i. Then, we select an arbitrary orientation of the 2-cell boundary and write down the boundary with the sequence of symbols, by using the corresponds between 1-cells and edges in G. Next, we eliminate the symbol e_i (or e_i^{-1}) in the sequence if e_i is an edge in the spanning tree T. Then, each remaining symbol e_i must correspond to a *non-tree edge*

$e_j^{\neg T}$. Finally, we replace each symbol with the generator γ_j, which corresponds to the loop identified by $e_j^{\neg T}$.

4.4 Group Representation Simplification

The group representation obtained from previous procedures has redundancies. In order to simply the presentation, we first remove some redundancies by the following simple algorithm.

1. Sort the relations by their lengths.
2. For each relation with length one, $w = \gamma_k$, that means γ_k is homotopic to a point, then we remove w from the relations and γ_k from the generators, and remove γ_k from all relations.
3. For each relations with length two, $w = \gamma_i \gamma_j$, that means $\gamma_j = \gamma_i^{-1}$, then we remove w from relators, γ_j from the generators, and replace γ_j by γ_i^{-1} in all relations.
4. Repeat step 1 through 3, until the lengths of all relations are greater than two.

Then we use the computational algebraic package GAP [24] for further simplification, which is based on Tietze Transformation program [25] with four elementary Tietze transformations to modify a representation to an isomorphic one.

1. *Adding a relation* If a relation can be derived from the existing relations then it may be added to the presentation without changing the group.
2. *Removing a relation* f a relation in a presentation can be derived from the other relations then it can be removed from the presentation without affecting the group.
3. *Adding a generator* Given a presentation it is possible to add a new generator that is expressed as a word in the original generators.
4. *Removing a generator* If a relation can be formed where one of the generators is a word in the other generators then that generator may be removed.

5 Experimental Results

In this section, we analysis the complexity of our algorithms, then apply to general 3-manifolds to compute their fundamental group.

5.1 Complexity Analysis

Suppose the input 3-manifold has n_1 edges, n_2 triangles and n_3 tetrahedra, then the complexity of the algorithm to convert it to a CW complex is linear, $O(n_1+n_2+n_3)$. Suppose the number of k-cells in the CW complex is m_k, $k = 0,1,2,3$, $m_k < n_k$. In general, $m_k \ll n_k$ and our algorithm ensures to minimize m_k, m_1 equals to the number of the group generators, m_2 equals to the number of relations. Then computing the group generators is in m_1 steps, and computing relations is less than m_2 steps. Therefore, the total complexity is $O(n_1+n_2+n_3)+O(m_1+m_2)$. The complexity of the final step to use Tietze transformations to simplify the group representation is difficult to analyze. Because in theory, finding the simplest representation of a group using Tietze transformation is undecidable. We use the heuristic algorithm, which is linear to the number of generators and relations of the input group.

5.2 General 3-Manifolds

We test our algorithms introduced in Sec. 4 on general 3-manifolds. Due to the page limit, we only list one here. Figure 3 illustrates a complicated 3-manifold, constructed by removing a solid knot and a solid two-hole torus from a solid torus. We compute its fundamental group, which has four generators and three relations, as follows:

$$\begin{aligned}&\langle a,b,c,d;(b^{-1}c^{-1}bc),(a^{-1}b^{-1}dbd^{-1}bab^{-1}),\\&(b^{-1}d^{-1}bd^{-1}ab^{-1}a^{-1}d^{-1}aba^{-1}daba^{-1}b^{-1}ab^{-1}a^{-1}d^{-1}ab^{-1}a^{-1}daba^{-1}d)\rangle\end{aligned}$$

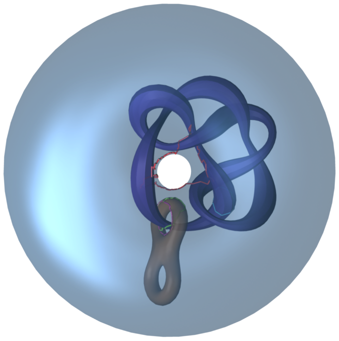

Fig. 3. A complicated 3-manifold is constructed by removing a solid knot and a solid two-hole torus from a solid torus. Four loops are marked with different colors, each of which corresponds a generator that cannot be shrunk to a point in the 3-manifold.

6 Conclusions and Future Work

In this paper, we provided a practical tool for computing the topology of general 3-manifolds with their fundamental groups. For the input tetrahedral mesh, we perform the CW cell decomposition to reduce the computational complexities. We also proved that the generators of the fundamental group of a 3-manifold come from its 1-skeleton and the relations come from its 2-cells in the CW complex. We presented the method to simplify the fundamental group representation by algebraic symbolic computations.

In the future, we will apply our algorithms for applications such as isotopy detection, handle and tunnel loops detections [7], DNA molecular structure, path planning in robotics, isotopy surface classification, and collision detection in animation.

Acknowledgments

We thank Tamal K. Dey for the references and advice and Alexander Hulpke for the assistances on using GAP. The project is supported by NSF, NSFC, and Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2008-331-D00510).

References

1. Edelsbrunner, H.: Biological applications of computational topology. In Goodman, J.E., O'Rourke, J., eds.: Handbook of Discrete and Computational Geometry. CRC Press (2004) 1395–1412
2. Moore, E.L.F., Peters, T.J.: Computational topology for geometric design and molecular design. In Ferguson, D., Peters, T., eds.: Mathematics for Industry: Challenges and Frontiers. SIAM (2005) 125–137
3. Hart, J.C.: Using the CW-complex to represent the topological structure of implicit surfaces and solids. In: Proc. Implicit Surfaces '99. (1999) 107–112
4. Amenta, N., Peters, T.J., Russell, A.: Computational topology: Ambient isotopic approximation of 2-manifolds. Theoretical Computer Science **305** (2003) 3–15
5. Abe, K., Bisceglio, J., Peters, T., Russell, A., Ferguson, D., Sakkalis, T.: Computational topology for reconstruction of surfaces with boundary: integrating experiments and theory. In: Shape Modeling and Applications, 2005 International Conference. (2005) 288–297
6. Abe, K., Bisceglio, J., Ferguson, D., Peters, T., Russell, A., Sakkalis, T.: Computational topology for isotopic surface reconstruction. Theoretical Computer Science **365** (2006) 184–198
7. Dey, T.K., Li, K., Sun, J.: On computing handle and tunnel loops. In: 2007 International Conference on Cyberworlds. (2007) 357–366
8. DiCarlo, A., Milicchio, F., Paoluzzi, A., Shapiro, V.: Chain-based representations for solid and physical modeling. IEEE Trans. Automation Science and Engineering **5** (2008) to appear
9. Vegter, G., Yap, C.K.: Computational complexity of combinatorial surfaces. ACM SoCG 1990 (1990) 102–111
10. Dey, T.K., Schipper, H.: A new technique to compute polygonal schema for 2-manifold with application to null-homotopy detection. Discrete & Computational Geometry **14** (1995) 93–110
11. Kaczyński, T., Mrozek, M., Ślusarek, M.: Homology computation by reduction of chain complexes. Computer & Mathematics with Applications **35** (1998) 59–70
12. Dey, T.K., Guha, S.: Computing homology groups of simplicial complexes in R^3. Journal of ACM **45** (1998) 266–287
13. Lazarus, F., Vegterz, G., Pocchiola, M., Verroust, A.: Computing a canonical polygonal schema of an orientable triangulated surface. ACM SoCG 2001 (2001) 80–89
14. Erickson, J., Har-Peled, S.: Optimally cutting a surface into a disk. Discrete & Computational Geometry **31** (2004) 37–59
15. Éric Colin de Verdière, Lazarus, F.: Optimal system of loops on an orientable surface. Discrete & Computational Geometry **33** (2005) 507–534
16. Yin, X., Jin, M., Gu, X.: Computing shortest cycles using universal covering space. The Visual Computer **23** (2007) 999–1004
17. Hempel, J.: 3-Manifolds. AMS Chelsea Publishing (1935)
18. Hatcher, A.: Algebraic Topology. Cambridge University Press (2001)
19. Dey, T.K., Edelsbrunner, H., Guha, S.: Computational topology. In Chazelle, B., Goodman, J.E., Pollack, R., eds.: Advances in Discrete and Computational Geometry. Contemporary Mathematics 223. AMS (1999) 109–143
20. Erickson, J., Whittlesey, K.: Greedy optimal homotopy and homology generators. 16th Annual ACM-SIAM Symposium on Discrete Algorithms (2005) 1038–1046
21. Dey, T.K., Guha, S.: Transforming curves on surfaces. Journal of Computer and System Sciences **58** (1999) 297–325
22. Munkres, J.R.: Elements of Algebraic Topology. Benjamin Cummings (1984)
23. Lyndon, R.C., Schupp, P.E.: Combinatorial Group Theory. Springer (2001)

24. GAP: ((Groups, Algorithms, Programming - a System for Computational Discrete Algebra) http://www-gap.dcs.st-and.ac.uk/)
25. Robertson, E.F.: Tietze transformations with weighted substring search. Journal of Symbolic Computation **6** (1988) 59–64

Appendix:

Proof. From the CW complex definition, $M^3 = M^2 \bigcup\{\sigma_1^3, \cdots, \sigma_{n_3}^3\}$. Let $\bar{M}^2$ be the tubular neighborhood of M^2 (i.e., a thickened M^2),

$$U_1 = \bar{M}^2 \bigcup \{\sigma_1^3, \cdots, \sigma_{n_3-1}^3\},\ U_2 = \sigma_{n_3}^3,$$

then both U_1 and U_2 are path connected. $U_1 \cap U_2$ is a tubular neighborhood of the boundary of U_2, which can retract to a sphere. Therefore, $U_1 \cap U_2$ is path connected. Both $\pi(U_2)$ and $\pi(U_1 \cap U_2)$ are trivial. By applying Seifert-van Kampen Theorem [18], we get $\pi(M^3) = \pi(U_1)$. This shows that the fundamental group is preserved by removing a 3-cell. We can repeat this process to remove all 3-cells, then we get $\pi(M^3) = \pi(\bar{M}^2)$. Because M^2 is the deformation retract of $\bar{M}^2$, therefore $\pi(M^2) = \pi(\bar{M}^2) = \pi(M^3)$.

Proof. By induction. Suppose $n_2 = 0$, because the 1-skeleton M^1 is a graph, therefore, its fundamental group is a free group [18], $\pi(M^1) = \langle \gamma_1, \gamma_2, \cdots, \gamma_n \rangle$, where γ_i's are independent loops of the graph.

Suppose the proposition holds for $n_2 < k$. Now assume $n_2 = k$, $M^2 = M^1 \bigcup\{\sigma_1^2, \sigma_2^2, \cdots, \sigma_k^2\}$. Let $\bar{M}^1$ be the tubular neighborhood of M^1,

$$U_1 = \bar{M}^1 \bigcup \{\sigma_1^2, \sigma_2^2, \cdots, \sigma_{k-1}^2\}, U_2 = \sigma_k^2,$$

then both U_1 and U_2 are path connected. $U_1 \bigcap U_2$ is a topological annulus, retracts to ∂U_2. $\pi(U_2)$ is trivial, therefore the loop ∂U_2 is homotopic to a point in U_2. ∂U_2 is a loop in U_1, we use $[\partial U_2]$ to denoted its homotopy class in $\pi(U_1)$, which can be represented as an element in $\pi(M^1)$, namely, a word formed by γ_k's. By assumption,

$$\pi(U_1) = \langle \gamma_1, \gamma_2, \cdots, \gamma_n; [\partial\sigma_1^2], [\partial\sigma_2^2], \cdots [\partial\sigma_{k-1}^2] \rangle$$

According to Seifert-van Kampen Theorem [18], $\pi(M^2)$ can be obtained by inserting $[\partial U_2]$ to the relations of $\pi(U_1)$, therefore

$$\pi(M^2) = \langle \gamma_1, \gamma_2, \cdots, \gamma_n; [\partial\sigma_1^2], [\partial\sigma_2^2], \cdots [\partial\sigma_k^2] \rangle$$