On the Incremental Computation of Simplicial Homology

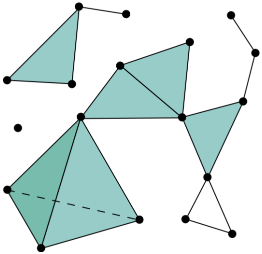
Brian Hamrick

Thomas Jefferson High School for Science and Technology Computer Systems Lab

October 26, 2009

Introduction

- Simplices
- Simplicial Complexes
- Simplicial Homology



http://commons.wikimedia.org/wiki/File:Simplicial_complex_example.svg

Homology

Computed from a chain complex of abelian groups

$$\cdots \xrightarrow{\partial_{n+1}} C_n \xrightarrow{\partial_n} C_{n-1} \xrightarrow{\partial_{n-1}} \cdots \xrightarrow{\partial_4} C_3 \xrightarrow{\partial_3} C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0$$

- $\partial_n \circ \partial_{n+1} = 0$
- $\mathrm{Im}(\partial_{n+1}) \trianglelefteq \mathrm{Ker}(\partial_n)$
- The homology groups are defined as $H_n = \mathrm{Ker}(\partial_n)/\mathrm{Im}(\partial_{n+1})$
- Simplicial Homology

Methods

- Matrix reduction to Smith normal form
- Incremental methods
- Geometric methods

Our Method

- Incremental
- Based on the Mayer-Vietoris Sequence

The Mayer-Vietoris Sequence

An exact sequence relating homology groups. If A and B are two spaces whose interiors cover X, then

$$\cdots \to H_{n+1}(X) \stackrel{\partial_*}{\to} H_n(A \cap B) \stackrel{(i_*, j_*)}{\to} H_n(A) \oplus H_n(B) \stackrel{k_* - l_*}{\to} H_n(X) \stackrel{\partial_*}{\to}$$
$$\stackrel{\partial_*}{\to} H_{n-1}(A \cap B) \to \cdots \to H_0(A) \oplus H_0(B) \stackrel{k_* - l_*}{\to} H_0(X) \to 0$$

is an exact sequence.

Incremental Method

- Idea: Add one simplex at a time, updating homology groups as you go.
- A is our previous simplicial complex K_{i-1}, B is the new simplex σ_i, X is the new simplicial complex K_i.

$$\cdots \to H_{n+1}(K_i) \to H_n(\partial\sigma_i) \to H_n(K_{i-1}) \to H_n(K_i) \to \cdots$$

One Dimensional Complexes

Two operations:

- Add a vertex
- Add an edge
  - Join two components (reduce H_0)
  - Create a cycle (expand H_1)

Two Dimensional Simplices

Three operations:

- Add a vertex
- Add an edge
- Add a triangle

Adding a Triangle

Two cases:

- Part of a cycle – cycle becomes a new generator in H_2.
- Not part of a cycle – boundary defines a new relation in H_1.
    - How to write boundary in terms of the generators?

Keeping Track of the Generators

Introduce the concept of a *fundamental cycle*.

Definition

For an index i, if σ_i was the part of a cycle when it was first added, then a cycle which contains σ_i the minimal positive number of times is called the fundamental cycle S_i.

Theorem

The fundamental cycles of dimensions k form a generating set of the cycle group C_k.

Setup for the Algorithm

Variables to keep track of:

- The fundamental cycles S_i corresponding to σ_i.
- The generators G_j for H_k and their torsion coefficients T_j.
- The representation for each S_i in terms of the generators G_j.

Adding a Simplex

- Determine if the simplex is part of a cycle.
- If it is, add the cycle as a new generator with a torsion coefficient of 0 and as a new fundamental cycle.
- If not, write the boundary in terms of the generators, then compute the quotient of the next smaller homology group by the relation defined by the boundary.

Adding a Simplex

- Determine if the simplex is part of a cycle.
- If it is, add the cycle as a new generator with a torsion coefficient of 0 and as a new fundamental cycle.
- If not, write the boundary in terms of the generators, then compute the quotient of the next smaller homology group by the relation defined by the boundary.

Is σ_i Part of a Cycle?

- σ_i is one-dimensional
  - Path finding problem
- σ_i is two-dimensional
  - Each bounding edge must be cancelled out by a contribution from a neighboring simplex
  - Floodfill

Adding a Simplex

- Determine if the simplex is part of a cycle.
- If it is, add the cycle as a new generator with a torsion coefficient of 0 and as a new fundamental cycle.
- If not, write the boundary in terms of the generators, then compute the quotient of the next smaller homology group by the relation defined by the boundary.

Writing a Cycle in Terms of Generators

- Start with cycle $Z = \sum c_i \sigma_i$.
- Initial representation $Z_0 = 0$.
- To create the next representation Z_{n+1}, take the latest nonzero term of $Z - Z_n$ by index, $c_i \sigma_i$. Then $Z_{n+1} = Z_n + c_i S_i$.
- The index of the last nonzero coefficient of $Z - Z_n$ strictly decreases with each iteration, so eventually $Z = Z_n$.
- Use stored representations of S_i in terms of the generators G_i to translate $Z = \sum d_i S_i$ into $Z = \sum e_i G_i$.

Computing Quotient of Homology Group

- Adding a single relation: $\partial\sigma_i = Z = \sum e_i G_i = 0$.
- Smith Normal Form of $\begin{pmatrix} T_1 & 0 & 0 & \cdots & 0 \\ 0 & T_2 & 0 & \cdots & 0 \\ 0 & 0 & T_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & T_n \\ e_1 & e_2 & e_3 & \cdots & e_n \end{pmatrix}$
- Column operations change generators and thus representations of the fundamental cycles

Performance Analysis

- Approximately one Smith Normal Form computation for each simplex.
- Sparse matrix
- Computation at each step is at most $O(\sum T_i)$ row and column operations
- $O(N)$ row and column operations per update for orientable surfaces