MAX-PLANCK-GESELLSCHAFT

Cycle Bases in Graphs
Structure, Algorithms, Applications, Open Problems

Kurt Mehlhorn
Max-Planck-Institut für Informatik
Saarbrücken

based on survey (under construction)

M. Geyer, Ch. Liebchen, T. Kavitha, D. Michail, R. Rizzi, G. Wünsch, K. Zweig

Motivation

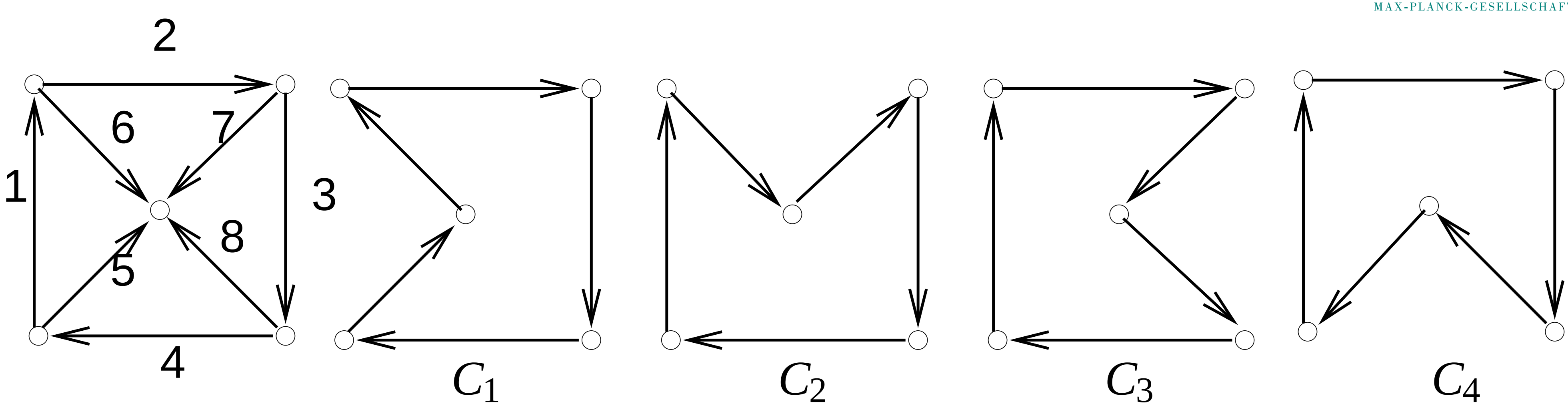
- graphs without cycles are boring
- cycles in graphs play an important role in many applications, e.g., network analysis, biology, chemistry, periodic scheduling, surface reconstruction
- cycle bases are a compact representation of the set of all cycles
- cycle bases raise many interesting mathematical and algorithmic problems

Overview

- Structural Results
  - Directed, Undirected, Integral, Strictly Fundamental Bases
  - The Arc-Cycle Matric and its Determinant
  - General Weight Bounds
- Minimum Weight Cycle Bases: Complexity and Algorithms
  - Undirected and Directed Cycle Basis: Polynomial Time
  - Strictly Fundamental: APX-hard
  - Integral: ???
- Applications
  - Network Analysis
  - Periodic Time Tabling
  - Buffer: Surface Reconstruction

Slides available at my home page
Survey paper should be available within the next two months

Cycle Basis



- $\mathscr{B} = \{C_1, C_2, C_3, C_4\}$ is a directed cycle basis
- vector representation: $C_1 = (0,1,1,1,1,-1,0,0)$, entries = edge usages
- $D = (1,1,1,1,0,0,0,0) = (C_1 + C_2 + C_3 + C_4)/3$ computation in $\mathbb{Q}$
- weight of basis: $w(\mathscr{B}) = 3w(e_1) + 3w(e_2) + \ldots + 2w(e_5) + 2w(e_6) + \ldots$
- undirected basis: $C_1 = (0,1,1,1,1,1,0,0)$ ignore directions
- $D = C_1 \oplus C_2 \oplus C_3 \oplus C_4$ computation in $\mathbb{Z}_2$

Undirected Cycle Basis: Formal Definition

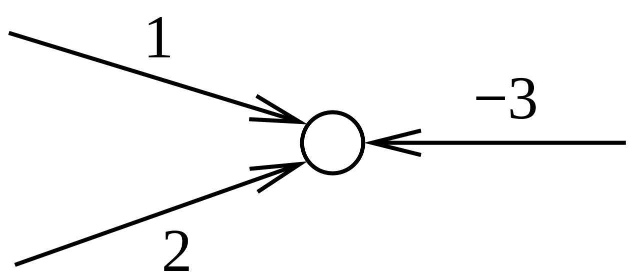
- $G = (V,E)$ undirected graph

- cycle = set C of edges such that degree of every vertex wrt C is even
- $C = (m(e_1), m(e_2), \ldots, m(e_m)) \in \{0,1\}^E$
- $m(e_i) = 1$ iff e_i is an element of C

- cycle space = set of all cycles
- addition of cycles = componentwise addition mod 2
  = symmetric difference of edge sets

The Directed Case

- $G = (V, E)$ directed graph
- cycle space = vector space over $\mathbb{Q}$.
- element of this vector space, $C = (m(e_1), m(e_2), \ldots, m(e_m)) \in \mathbb{Q}^E$
- $m(e_i)$ multiplicity of e_i
- constraint
  - take $|m(e_i)|$ copies of e_i
  - reverse direction if $m(e_i) < 0$
  - then inflow = outflow for every vertex



- a simple cycle in the underlying undirected graph gives rise to a vector in $\{-1, 0, +1\}^E$.

The Spanning Tree Basis

- let T be an arbitrary spanning tree $\qquad N =$ non-tree edges
- for every non-tree edge e,

  $C_e = e +\ T$- path connecting the endpoints of e
- $\mathscr{B} = \{C_e;\, e \in N\}$ is a basis $\qquad$ dimension of cycle space $\nu := N := m - n + 1$
  - cycles in $\mathscr{B}$ are independent
  - they span all cycles: for any cycle C, we have $C = \sum_{e\in N} \lambda_e \cdot C_e$

$$\lambda_e = \begin{cases} +1 & \text{if } C \text{ and } C_e \text{ use } e \text{ with identical orientation} \\ -1 & \text{if } C \text{ and } C_e \text{ use } e \text{ with opposite orientation} \\ 0 & \text{otherwise} \end{cases}$$

Pf: $C - \sum_{e\in N} \lambda_e \cdot C_e$ is a cycle and contains only tree edges.

- *minimum weight spanning tree basis* is NP-complete (Deo et. al., 82)
- spanning tree basis is *integral*

Weight of a Basis

w, weight function on the arcs

weight of a cycle = sum of the weight of its arcs

weight of a basis = sum of the weights of its cycles

uniform weights: $w(a) = 1$ for all arcs a

Applications I

- analysis of cycle space has applications in electrical engineering, biology, chemistry, periodic scheduling, surface reconstruction, graph drawing. . .
- in these applications, it is useful to have a basis of small cardinality (uniform weights) or small weight (non-uniform weights)
- analysis of an electrical network (Kirchhof's laws)
  - for any cycle C the sum of the voltage drops is zero
  - sufficient: for every cycle C in a cycle basis ....
  - number of non-zero entries in equations = size of cycle basis
  - computational effort is heavily influenced by size of cycle basis
  - electrical networks can be huge (up to a 100 millions of nodes), Infineon

Network Analysis

- consider a network with nonlinear resistors, i.e., voltage drop is a nonlinear function of current (not necessarily monotonic), and some number of independent current sources
- voltage drop v_a at arc a, current i_a through i_a: $v_a = f_a(i_a)$
- constraints

$$\sum_{a \in C} f_a(i_a) = 0 \qquad \text{for any cycle } C \qquad (1)$$

$$\text{current into } v = \text{current out of } v \qquad \text{for any vertex } v \qquad (2)$$

$$i_a = \text{const} \qquad \text{for current source arcs} \qquad (3)$$

- constraints (1) are numerically hard, (2) are easy
- it suffices to enforce (1) for the circuits in a basis
- number of terms in (1) = total cardinality of cycle basis
- computational effort is heavily influenced by size of cycle basis
- electrical networks can be huge (millions of nodes), Infineon

The Zoo of Cycle Bases I

MAX-PLANCK-GESELLSCHAFT

- Let $G = (V, A)$ be a directed graph and let $\mathscr{B}$ be a basis of its directed cycle space. $\mathscr{B}$ is called a
- directed cycle basis: always
- undirected cycle basis: if (after ignoring edge directions) it is a undirected cycle basis of the underlying undirected graph.
- integral cycle basis: if every directed cycle is an integral linear combination of the cycles in $\mathscr{B}$
- strictly fundamental cycle basis: if there is a spanning tree T such that $\mathscr{B}$ is the set of fundamental cycles with respect to T

Thm (Liebchen/Rizzi)

- this is a hierarchy, e.g., any integral basis is an undirected basis
- In general, higher-up classes are strictly larger.
- In general, higher-up classes have better minimum weight bases

The Zoo of Cycle Bases II: Hierarchy

- $\mathscr{B}$ be a basis of directed cycle space. $\mathscr{B}$ is called a
- directed cycle basis: always
- undirected cycle basis: if (after ignoring edge directions) it is a undirected cycle basis of the underlying undirected graph.
- integral cycle basis: if every directed cycle is an integral linear combination of the cycles in $\mathscr{B}$
- strictly fundamental cycle basis: if there is a spanning tree T such that $\mathscr{B}$ is the set of fundamental cycles with respect to T

- any strictly fundamental basis is integral already shown

The Zoo of Cycle Bases II: Hierarchy

- $\mathscr{B}$ be a basis of directed cycle space. $\mathscr{B}$ is called a
- directed cycle basis: always
- undirected cycle basis: if (after ignoring edge directions) it is a undirected cycle basis of the underlying undirected graph.
- integral cycle basis: if every directed cycle is an integral linear combination of the cycles in $\mathscr{B}$
- strictly fundamental cycle basis: if there is a spanning tree T such that $\mathscr{B}$ is the set of fundamental cycles with respect to T

- any integral basis is an undirected basis:

  if $C = \sum_{C_i \in \mathscr{B}} \lambda_i C_i$ with $\lambda_i \in \mathbb{Z}$, the same equation holds $\bmod 2$

The Zoo of Cycle Bases II: Hierarchy

- $\mathscr{B}$ be a basis of directed cycle space. $\mathscr{B}$ is called a
- directed cycle basis: always
- undirected cycle basis: if (after ignoring edge directions) it is a undirected cycle basis of the underlying undirected graph.
- integral cycle basis: if every directed cycle is an integral linear combination of the cycles in $\mathscr{B}$
- strictly fundamental cycle basis: if there is a spanning tree T such that $\mathscr{B}$ is the set of fundamental cycles with respect to T

- any undirected basis is a directed basis:

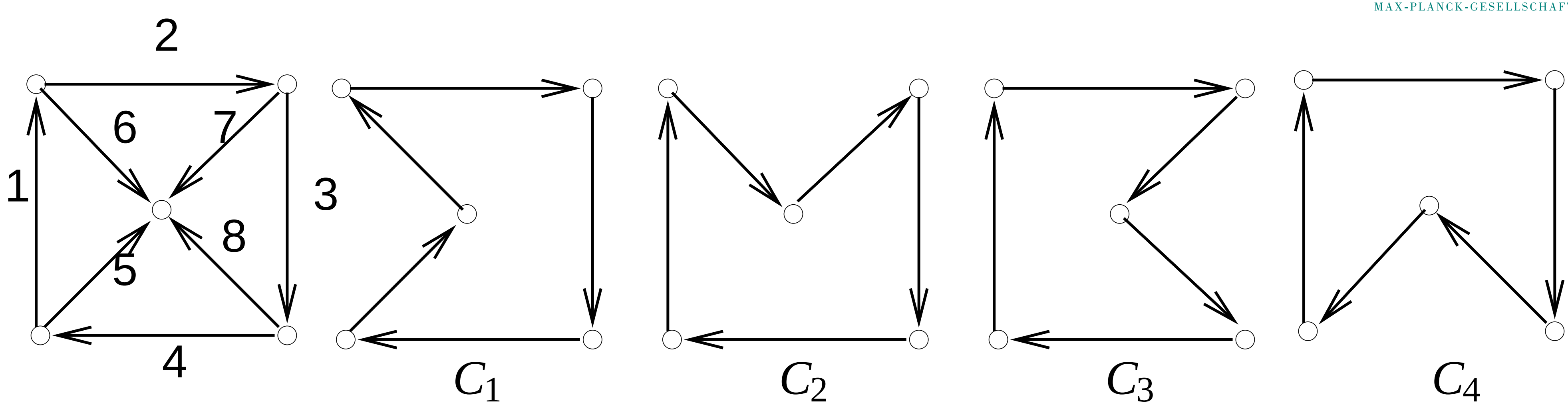
  if a set of cycles is depedendent over $\mathbb{Q}$, then over $\mathscr{F}_2$

  if $\sum_i \lambda_i C_i = 0$ with $\lambda_i \in \mathbb{Z}$, not all even, then this is also nontrivial over $\mathscr{F}_2$

Proof Technique for Strict Hierarchy

- let X and Y be two of the types with X “above” Y
- invent a graph G and a weight function w
- invent a basis $\mathscr{B}$ of G
- show that $\mathscr{B}$ is a (unique minimum weight) basis of type X
- show that $\mathscr{B}$ is not of type Y

Cycle Basis



- $\mathscr{B} = \{C_1, C_2, C_3, C_4\}$ is a directed cycle basis
- vector representation: $C_1 = (0,1,1,1,1,-1,0,0)$, entries = edge usages
- $D =$ the cycle consisting of the four outer edges
- $D = (1,1,1,1,0,0,0,0) = (C_1 + C_2 + C_3 + C_4)/3$
- $\mathscr{B}$ is not an integral basis

Open Problem on Hierarchy

- Let X and Y be two classes with Y $\subseteq$ X:

  derive a good bound for

$$\max_{G,w} \frac{\text{cost of minimum weight basis of type Y}}{\text{cost of minimum weight basis of type X}}$$

- the only known result of this kind is (see below):

$$\max_{G,w} \frac{\text{cost of minimum weight integral basis}}{\text{cost of minimum weight basis}} \leq \log n$$

Simple Properties

- G consists of components $G_1, G_2, \ldots$

  a minimum weight (directed, undirected) cycle basis of G is obtained by combining optimal bases of the components

- there is a minimum weight (directed, undirected) cycle basis consisting only of simple cycles
  - assume $C \in B$ is nonsimple
  - thus $C = C_1 + C_2$ with $w(C_i) \leq w(C)$
  - coefficient of C in representation of either C_1 or C_2 is non-zero (otherwise, $\mathscr{B} - C$ is a basis)
  - thus either $\mathscr{B} - C + C_1$ or $\mathscr{B} - C + C_2$ is a basis.
  - weight does not increase

- Open Problem: does either property hold for integral basis?
- Open Problem: a combinatorial characterization of integral bases

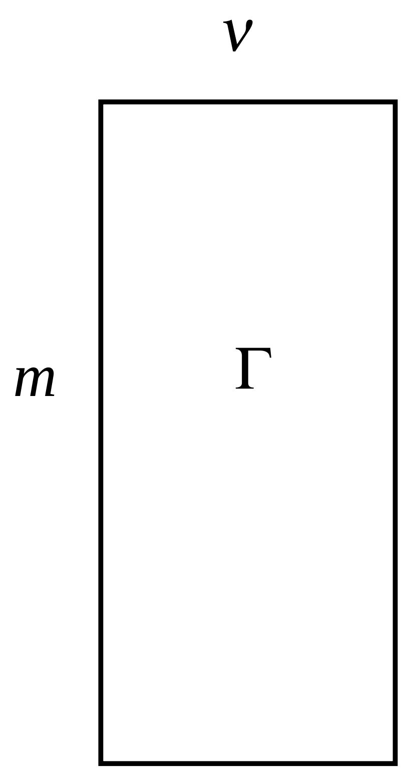
The Arc-Cycle Matrix

- $m \times \nu$ matrix Γ, $\qquad m = \nu + n - 1$
- rows are indexed by arcs, columns are indexed by cycles
- Γ corresponds to a basis $\mathscr{B}$ iff the equation

$$\chi_C = \Gamma x_C$$

  has a solution for the characteristic vector χ_C of any cycle C.
- square submatrices of Γ are of particular interest

- Thm (Liebchen): Up to sign, all nonsingular square submatrices of Γ have the same determinant.

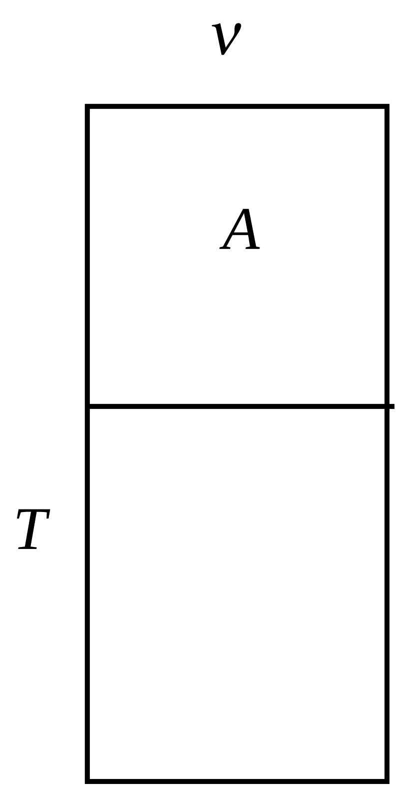


The Arc-Cycle Matrix II

- $m \times \nu$ matrix Γ, $m = \nu + n - 1$
- rows are indexed by arcs, columns are indexed by cycles
- Let T be a set of $n - 1$ edges
- The square submatrix corresponding to the edges not in T is non-singular iff T is a spanning tree
  - Let Φ be the arc-cycle matrix for the fundamental basis with respect to T. Then $\Phi = \Gamma R$ for some R and hence $I = AR$.

    Thus A is nonsingular. Also $\Gamma = \Phi R^{-1} = \Phi A$.
  - Assume T contains a cycle, say C. Then

$$\chi_C = \Gamma x_C \quad \text{and hence} \quad \mathbf{0} = A x_C$$



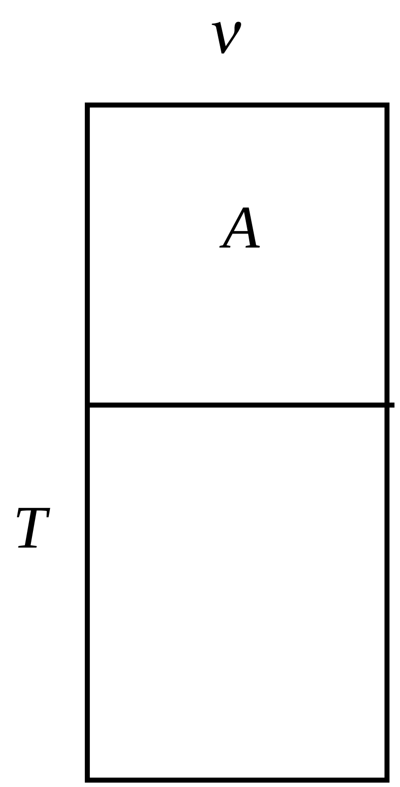
The Arc-Cycle Matrix III

- $m \times \nu$ matrix Γ, $m = \nu + n - 1$
- rows are indexed by arcs, columns are indexed by cycles
- Let T and T' be spanning trees,

  A indexed by the edges not in T,

  A' indexed by the edges not in T'

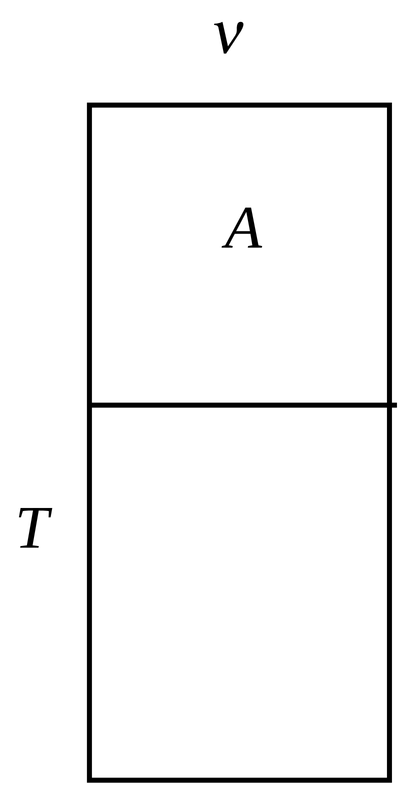
  - Let Φ be the arc-cycle matrix for the fundamental basis with respect to T. Then $\Phi A = \Gamma$.
  - Restriction to rows of A': $\Phi' A = A'$
  - Φ is totally unimodular: $\pm \det A = \det A'$



Characterization of Cycle Basis in Terms of Γ

- $m \times \nu$ matrix Γ, $m = \nu + n - 1$
- rows are indexed by arcs, columns are indexed by cycles
- let $D = \det A$ be the determinant of the nonsingular square submatrices (up to sign)
- let C be any cycle, then

$$\chi_C = \Gamma x_C \quad \text{and hence} \quad x_C = A^{-1}\chi'_C$$



- Thm (Liebchen): $\mathscr{B}$ is
  - directed basis iff $D \neq 0$
  - undirected basis iff D is odd
  - integral basis iff D is one
- Open Problem: combinatorial characterization of integral basis

Small Weight Integral Bases

- Thm (Rizzi): Every digraph has an integral basis of weight $2W \log n$, where W is the total weight of the edges

- Fact: every graph of minimum degree 3 contains a cycle of length at most $2 \log n$. grow a breadth first tree

- Kavitha's algorithm (07):

  - while G is not a tree
    - view paths of degree two nodes as superedges
    - find cycle of $2 \log n$ superedges, call it C
    - add C to basis and delete its heaviest superedge from the graph

Small Weight Integral Basis II

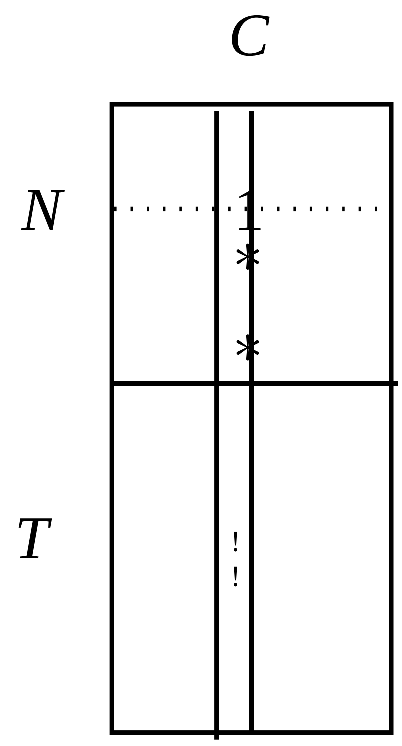
- while G is not a tree
  - view paths of degree two nodes as superedges
  - find cycle of $2\log n$ superedges
  - add it to basis and delete its heaviest superedge from the graph

- weight of cycle is at most $2\log n$ times weight of deleted edges
- thus $w(\mathscr{B}) \leq (2\log n)W$

Small Weight Integral Basis III

- while G is not a tree
  - view paths of degree two nodes as superedges
  - find cycle of $2 \log n$ superedges
  - add it to basis and delete its heaviest superedge from the graph

- we construct spanning tree as we go along
- classify one deleted edge as a nontree edge, all others as tree edges
- above dotted line: previously deleted nontree edges
- C uses no edge above dotted line
- thus the square matrix corresponding to the non-tree edges is lower diagonal with ones on the diagonal; hence basis is integral.



More on Absolute Weight Bounds

- every graph has an integral basis of weight $O(W \log n)$
- (Horton) every graph has an integral basis of size $O(n^2)$
  - by induction on the number of nodes
- there are graphs with $2n$ edges such that every basis has size $\Omega(n \log n)$
  - 4-regular graph with girth $\Omega(\log n)$
- so nonlinear size is required for very sparse graphs and linear size suffices for very dense graphs
- open problem: what happens for $m \in \omega(n) \cap o(n^2)$?
- open problem: bounds on the size of fundamental bases

Algorithms and Complexity

- minimum weight directed cycle basis: polynomial time
- minimum weight undirected cycle basis: polynomial time
- minimum weight strictly fundamental cycle basis: APX-hard, i.e., if $P \neq NP$, no constant-factor approximation
  - NP-completeness was shown by Deo et al.
  - APX-hardness was shown by Rizzi
- minimum weight integral basis: nothing is known
  - not known to be in P
  - clearly in NP
  - not known to be NP-complete
  - no nontrivial exact algorithm

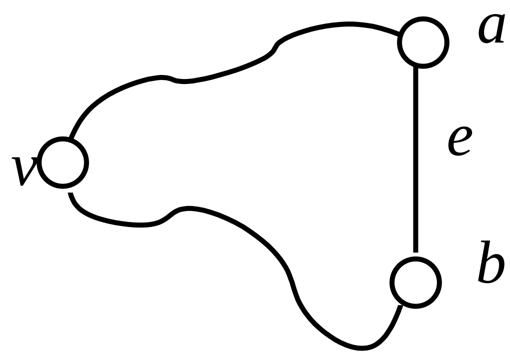
Algorithmic Approach 1: Horton

- compute a sufficiently large set of cycles, e.g., all simple cycles
- sort them by weight
- initialize $\mathscr{B}$ to empty set
- go through the cycles C in order of increasing weight
- add C to $\mathscr{B}$ if it is independent of $\mathscr{B}$
- use Gaussian elimination to decide independance
- in order to make the approach efficient, one needs to identify a small set of cycles which is guaranteed to contain a minimum basis

The Horton Set of Cycles

- for any edge $e = (a, b)$ and vertex v take the cycle

$$C_{e,v} = e + \text{shortest paths from } v \text{ to } a \text{ and } b$$



- $O(nm)$ cycles, Gaussian elimination on a $m \times nm$ matrix
- running time (Horton, Golynski/Horton): $O(nm^3)$ or $O(nm^\omega)$
- a smaller set suffices (Mehlhorn/Michail): v belongs to a feedback vertex set and a and b are in different subtrees of shortest path tree T_v.
- open problem: a candidate set of size $o(nm)$

Algorithmic Approach 2: de Pina

- construct basis iteratively, assume partial basis is $\{C_1, \ldots, C_i\}$
- compute a vector S orthogonal to $C_1, \ldots, C_i$, i.e.,

  $$\langle C_j, S \rangle = 0 \text{ for } 1 \le j \le i.$$
- find a cheapest cycle C with $\langle C, S \rangle \neq 0$
- set C_{i+1} to C and in this way extend the partial basis

- C is **not** the cheapest cycle independent of the partial basis

Algorithmic Approach 2: de Pina

- construct basis iteratively, assume partial basis is $\{C_1,\ldots,C_i\}$
- compute a vector S orthogonal to $C_1, \ldots, C_i$, i.e.,
  $\langle C_j, S\rangle = 0$ for $1 \le j \le i$.
- find a cheapest cycle C with $\langle C, S\rangle \neq 0$
- set C_{i+1} to C and in this way extend the partial basis

- C is **not** the cheapest cycle independent of the partial basis
- correctness
  - alg computes a basis, since C_{i+1} is linearly independent from the previous C_j's
  - alg computes a minimum weight basis, since every basis must contain a C with $\langle C, S\rangle \neq 0$ and alg adds the cheapest such C

Algorithmic Approach 2: de Pina

- construct basis iteratively, assume partial basis is $\{C_1, \ldots, C_i\}$
- compute a vector S orthogonal to $C_1, \ldots, C_i$, i.e.,
  $\langle C_j, S\rangle = 0$ for $1 \leq j \leq i$.
- find a cheapest cycle C with $\langle C, S\rangle \neq 0$
- set C_{i+1} to C and in this way extend the partial basis

- C is **not** the cheapest cycle independent of the partial basis
- correctness
  - alg computes a basis, since C_{i+1} is linearly independent from the previous C_j's
  - alg computes a minimum weight basis, since every basis must contain a C with $\langle C, S\rangle \neq 0$ and alg adds the cheapest such C
- efficiency
  - make each iteration efficient
  - make iterations profit from each other

More Details

- partial basis $C_1, \ldots, C_i$,   vectors in $\{0,1\}^E$
- compute $S \in \{0,1\}^E$ orthogonal to $C_1, \ldots C_i$
  - amounts to solving a linear system of equations, namely

$$\langle S, C_j \rangle = 0 \mod 2 \text{ for } 1 \leq j \leq i$$

  - time bound for this step is $O(m^\omega)$ per iteration (Gaussian elimination) and $O(m^{1+\omega})$ in total
  - this can be brought done to $O(m^\omega)$ total time, see next slide
- determine a minimum weight cycle C with $\langle S, C \rangle \neq 0$
  - see next but one slide
- add it to the basis and repeat

Faster Implementation

- maintain partial basis $C_1, \ldots, C_{i-1}$,   vectors in $\{0,1\}^E$
- plus basis $S_i, \ldots S_N$ of orthogonal space
- iteration becomes:
  - intialize S_1 to S_N to unit vectors (S_i to i-th unit vector)
  - in i-th iteration, compute C_i such that $\langle S_i, C_i \rangle = 1 \mod 2$
  - update S_j, $j > i$, as $S_j = S_j - \, < S_j, C_i > S_i$
  - update step makes S_j orthogonal to C_i and maintains orthogonality to C_1 to C_{i-1}.
  - update step has time $O(m^2)$, total time $O(m^3)$.
- total time for updates can be brought done to $O(m^\omega)$

Yet Faster Implementation (KMM)

MAX-PLANCK-GESELLSCHAFT

- update in bulk   a generally useful technique
- $S_{N/2+1}$ to S_N are only needed in "second half" of computation, i.e., for computing $C_{N/2+1}$ to C_N
- update $S_{N/2+1}$ to S_N only after computation of C_1 to $C_{N/2}$
  - $(S'_{N/2+1},\ldots,S'_N) = (S_{N/2+1},\ldots,S_N) - (S_1,\ldots,S_{N/2}) \times R$, R unknown
  - we want $\langle S'_{N/2+i}, C_j \rangle = 0$ for $1 \le i, j \le N/2$
  - we know $\langle S_i, C_j \rangle = \delta_{ij}$ for $1 \le j \le i \le N/2$
  - multiply the equality above by $(C_1,\ldots,C_{N/2})^T$ and obtain

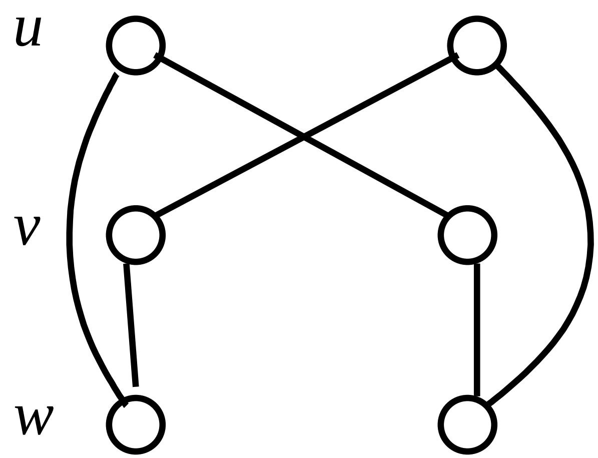
$$\mathbf{0} = (C_1,\ldots,C_{N/2})^T \times (S_{N/2+1},\ldots,S_N) - U \times R$$

  - U is upper diagonal with ones on the diagonal, solve for R
- update corresponds to a few matrix multiplies and matrix inversions
- use this idea recursively,   total time $O(m^\omega)$

Computing Cycles

determine a minimum weight cycle C with $\langle S, C\rangle \neq 0 \mod 2$, i.e., a minimum weight cycle using an odd number of edges in S.

- consider a graph with two copies of V, vertices v^0 and v^1.
- edges $e \in S$ change sides, and edges $e \notin S$ do not
- for any v, compute minimum weight path from v^0 to v^1.
- time $O(m + n \log n)$ for fixed v,
- time $O(nm + n^2 \log n)$ per iteration, i.e., for all v
- $O(nm^2 + n^2 m \log n)$ overall



$(u, v) \in S$,
$(v, w) \notin S$,
$(u, w) \notin S$

can be improved to $O(nm^2/\log n + n^2 m)$ by restricting search to Horton set

Improved Search for Cycle (MM)

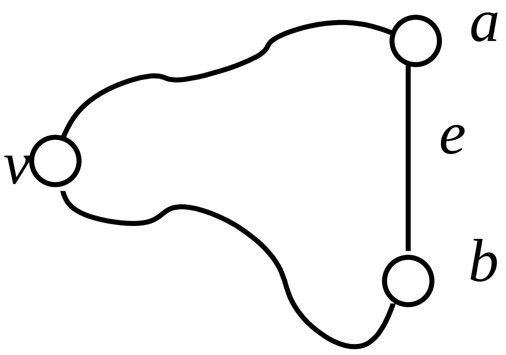
- idea: find cheapest $C \in$ Horton Set with $\langle S,C \rangle = 1$ instead of cheapest C with $\langle S,C \rangle = 1$

- precomputation: for each v, compute shortest path tree T_v ONCE

- in each iteration, i.e., once the S of the iteration is known
  - for each v do:
    - label a in T_v with $\langle S, p_a \rangle$
    - for any edge $e = (a,b)$, compute $\langle S, C_{v,e} \rangle$ as

$$\langle S, p_a \rangle + \langle S, e \rangle + \langle S, p_b \rangle$$

      in time $O(1)$
    - $O(m)$ per v, $O(mn)$ per iteration



History

Type	Authors	Approach	Running time
undirected	Horton, 87	Horton	$O(m^3 n)$
	de Pina, 95	de Pina	$O(m^3 + mn^2 \log n)$
	Golinsky/Horton, 02	Horton	$O(m^\omega n)$
	Berger/Gritzmann/de Vries, 04	de Pina	$O(m^3 + mn^2 \log n)$
	Kavitha/Mehlhorn/Michail/Paluch, 04	de Pina	$O(m^2 n + mn^2 \log n)$
	Mehlhorn/Michail, 07	Horton-Pina	$O(m^2 n / \log n + mn^2)$
directed	Kavitha/Mehlhorn, 04	de Pina	$O(m^4 n)$ det, $O(m^3 n)$ Monte Carlo
	Liebchen/Rizzi, 04	Horton	$O(m^{1+\omega} n)$
	Kavitha, 05	de Pina	$O(m^2 n \log n)$ Monte Carlo
	Hariharan/Kavitha/Mehlhorn, 05	de Pina	$O(m^3 n + m^2 n^2 \log n)$
	Hariharan/Kavitha/Mehlhorn, 06	de Pina	$O(m^2 n + mn^2 \log n)$ Monte Carlo
	Mehlhorn,Michail 07	Horton-Pina	$O(m^3 n)$ det, $O(m^2 n)$ Monte Carlo

open problem: faster algs, the S's are only used to guarantee independence

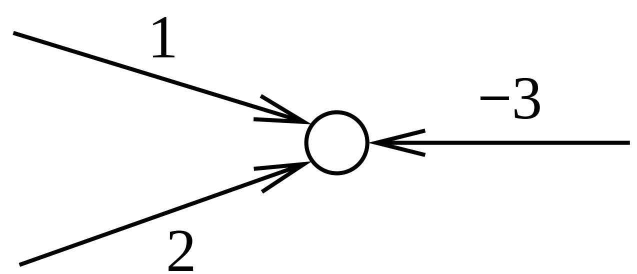
Implementation

MAX-PLANCK-GESELLSCHAFT

- our best implementation uses a blend of de Pina and Horton's approach
- plus heuristics for fast cycle finding
- much, much faster than the pure algorithms
- implementation available from Dimitris Michail
- for details, see M/Michail: Implementing Minimum Cycle Basis Algorithms (JEA)
- open problem: better implementation and/or algorithm that can handle Infineon's graphs

The Directed Case

- $G = (V, E)$ directed graph
- cycle space = vector space over $\mathbb{Q}$.
- element of this vector space, $C = (m(e_1), m(e_2), \ldots, m(e_m)) \in \mathbb{Q}^E$
- $m(e_i)$ multiplicity of e_i
- constraint
  - take $|m(e_i)|$ copies of e_i
  - reverse direction if $m(e_i) < 0$
  - then inflow = outflow for every vertex



- a simple cycle in the underlying undirected graph gives rise to a vector in $\{-1, 0, +1\}^E$.

The Directed Case: algorithmic Approaches

- in principle, as in the undirected case
- but the steps are much harder to realize as we now work over the field $\mathbb{Q}$ and no longer over $\mathscr{F}_2$.
- entries of our matrices become large integers and hence
  cost of arithmetic becomes non-trivial
- finding a minimum cost path with non-zero dot-product $\langle C, S \rangle$ becomes non-trivial
- use of modular arithmetic, randomization, and a variant of Dijkstra's algorithm
- details, see papers

Approximation Algorithms

$2k-1$ approximation in time $O(kmn^{1+1/k}+mn^{(1+1/k)(\omega-1)})$ Kavitha/M/Michail 07

- let $G'=(V,E')$ be a $2k-1$ spanner of G size $O(n^{1+1/k})$
- for any $e \in E \setminus E'$: e + shortest path in E' connecting its endpoints
- plus minimum cycle basis of G'
- weight of each family is bounded by $(2k-1)w(MCB)$
- more involved argument: joint weight is bounded by $(2k-1)w(MCB)$

open problem: better approximation algorithms, avoid use of matrix multiplication, how well can you do in linear time?

Summary

- cycle basis are useful in many contexts: analysis of electrical networks, periodic scheduling, surface reconstruction
- significant progress was made over the past five years
- many open questions (structural, algorithmic) remain
- in the remaining time, I tell you about an unexpected application

An Unexpected Application: Surface Reconstruction

MAX-PLANCK-GESELLSCHAFT

given a point cloud P in $\mathscr{R}^3$ reconstruct the underlying surface S

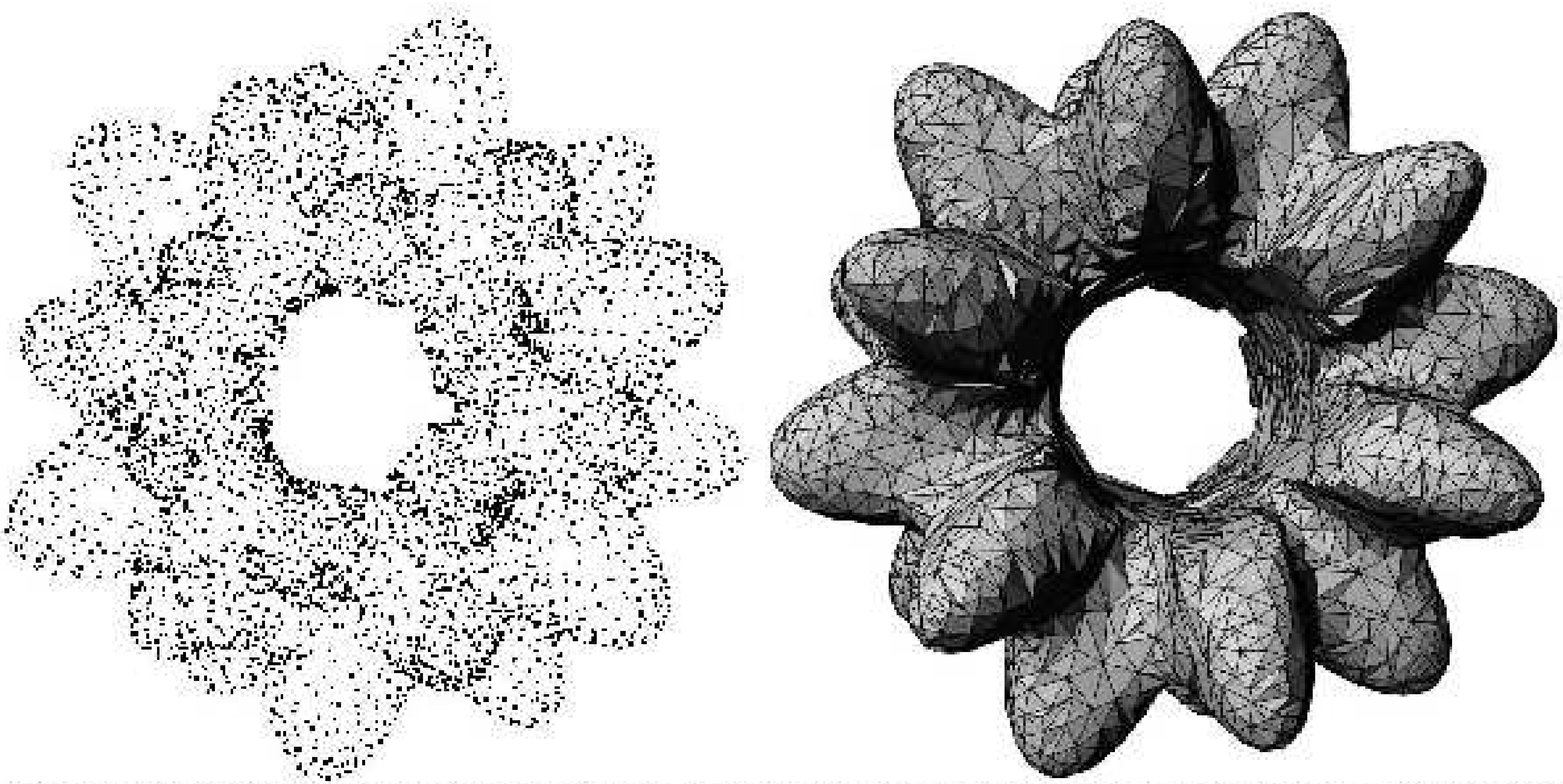
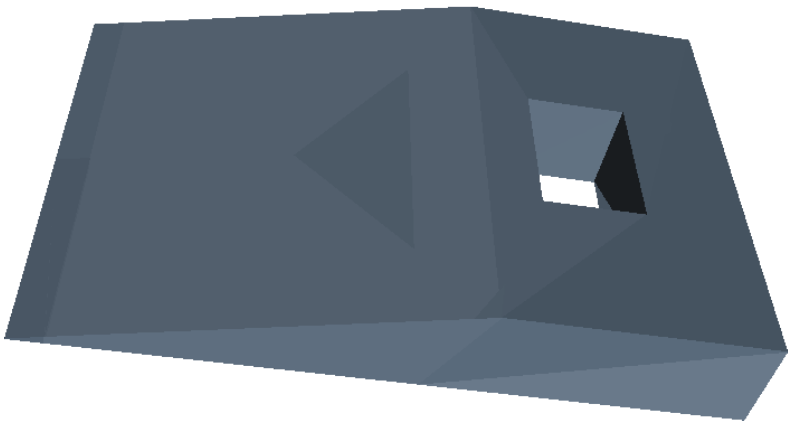
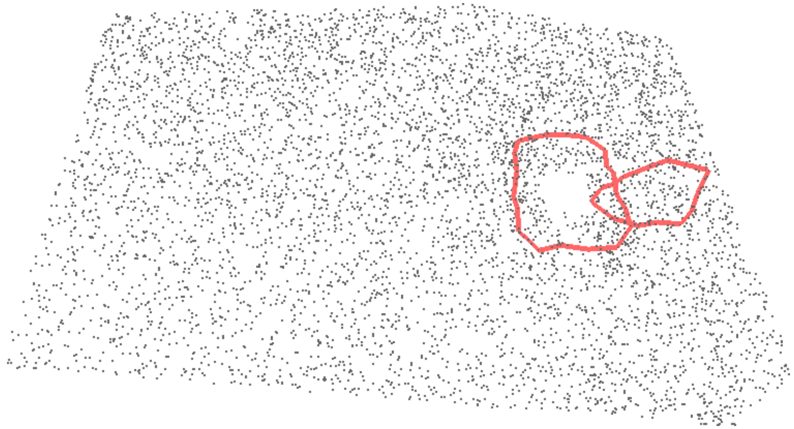


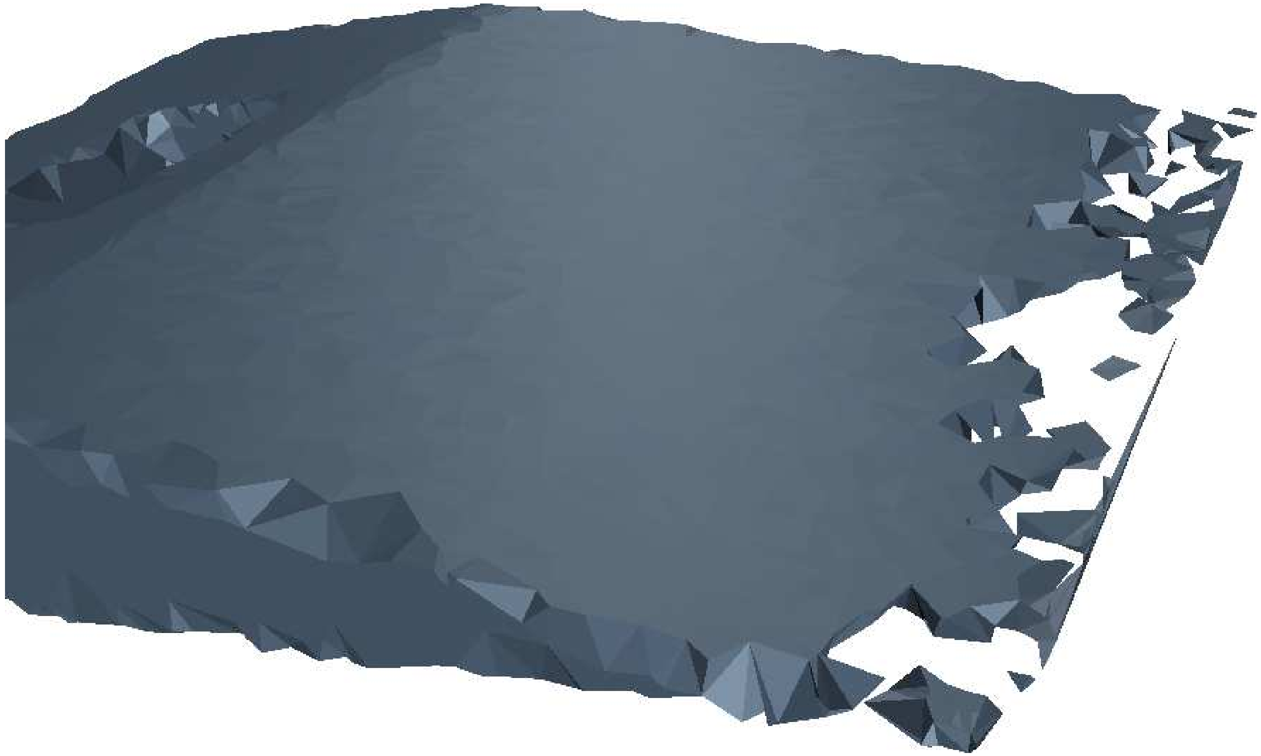
Figure 8: Reconstruction of the 7,371 point "bumpy torus" model. Parameters used were k=7, t=10, d=10 and no simulation of simplicity.

HERE: point cloud comes from a surface of genus one

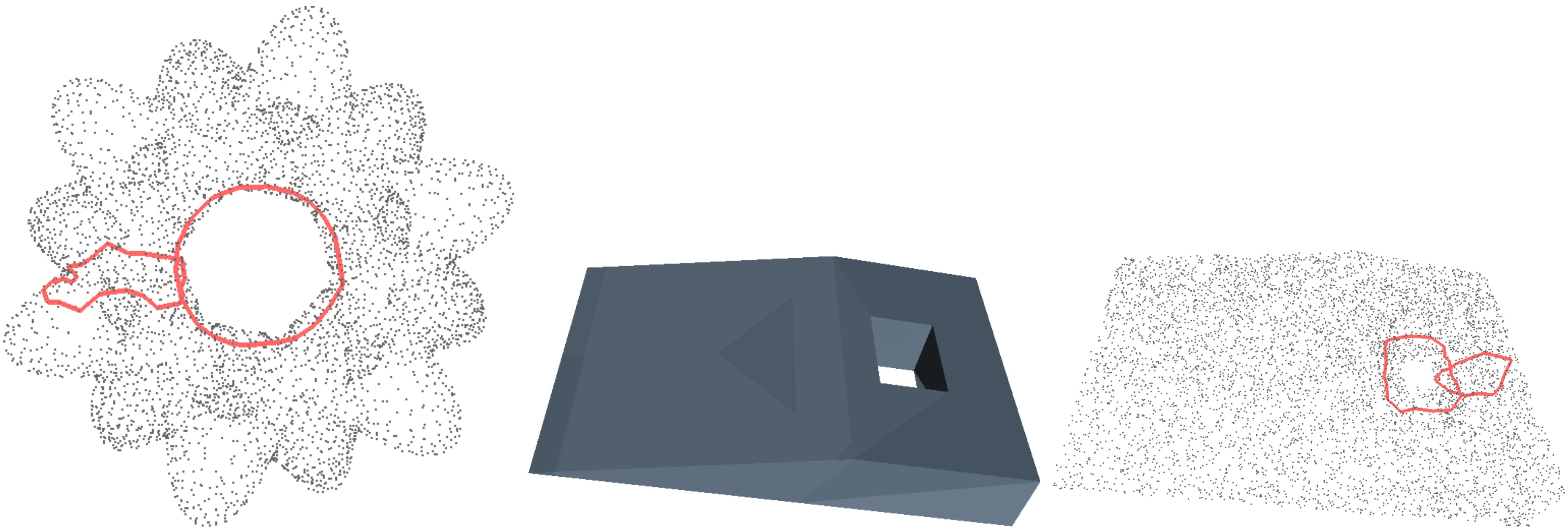
Beyond Smooth Surfaces: Cocone Reconstruction







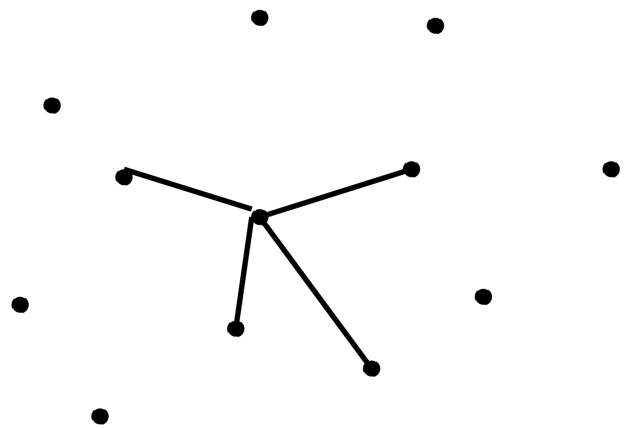
Beyond Smooth Surfaces: Genus Detection I



- genus g of a closed surface = sphere + g handles
- examples are genus one surfaces, i.e., homeomorphic to a torus
- genus detection: compute $2g$ cycles spanning the space of non-trivial cycles

MCBs in Nearest Neighbor Graph

- Nearest Neighbor Graph G_k on P (k integer parameter)
  - connect u and v is v is one the k points closest to u and vice versa



$k = 4$

  - easy to construct

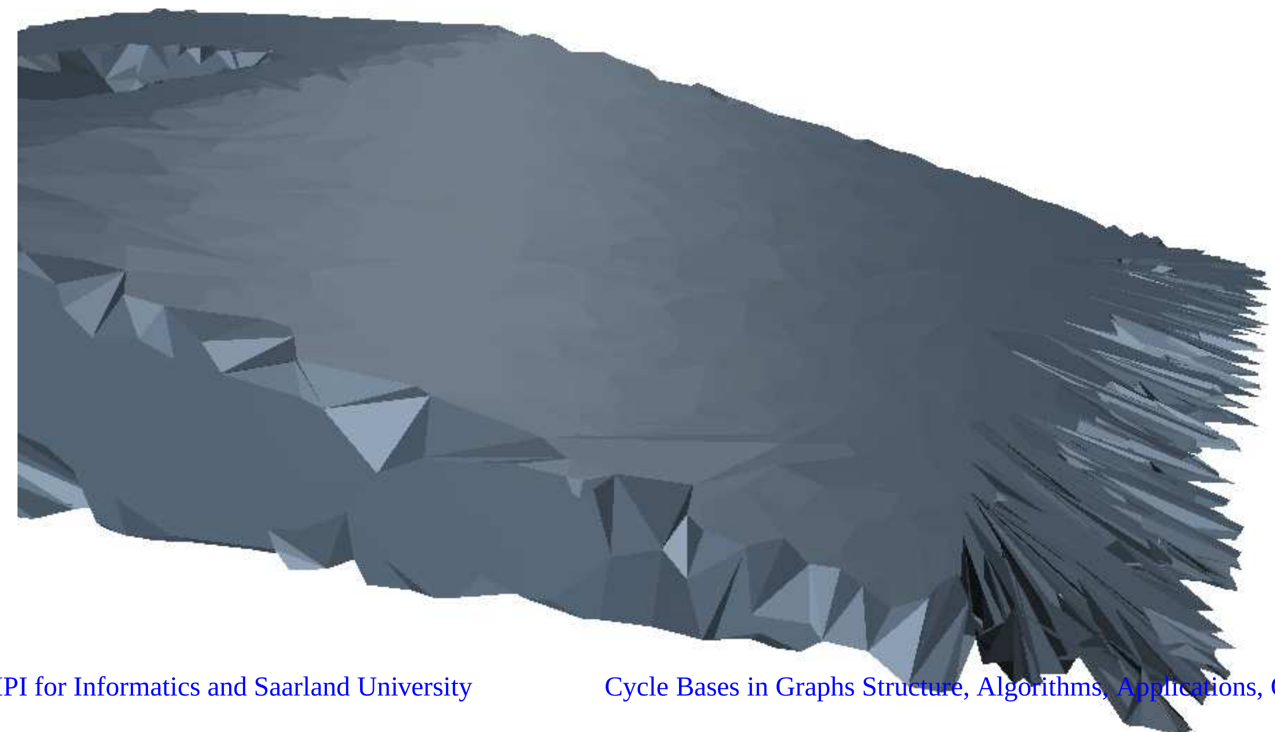
- Theorem (Gotsman/Kaligossi/Mehlhorn/Michail/Pyrga 05): if S is smooth, P is sufficiently dense, and k appropriately chosen:

  MCB of $G_k(P)$ consists of short (lenght at most $2k+3$) and long (length at least $4k+6$) cycles. There are $2g$ long cycles

  Moreover, the short cycles span the space of trivial cycles and the long cycles form a homology basis.

Beyond Smooth Surfaces: Reconstruction

- Tewari/Gotsman/Gortler have an algorithm to reconstruct genus one surfaces if a basis for the trivial cycles of $G_k(P)$ is known.
- our algorithm computes a basis for the trivial cycles of $G_k(P)$
- together the algorithms reconstruct genus one surfaces
- algorithm constructs a genus one triangulation of P
- open problem: geometric guarantee, not just topological guarantee



Tutte's Algorithm for Drawing a Planar Graph

- G is a 3-connected planar graph
- place the nodes of the outer face on the vertices of a convex polygon
- relax the graph, i.e., put every nonboundary node into the center of gravity of its neighbors

- produces a planar embedding with all faces nondegenerate
- algorithmically: amounts to solving a linear system either directly or iteratively
- for every vertex not on the boundary: $x_v = \sum_{w \in N(v)} x_w / \deg(v)$
- or alternatively $\sum_{w \in N(v)} (x_w - x_v) = 0$

Drawings on the Torus I

- goal: given a map (graph + cyclic ordering on the edges incident to any vertex) of genus one, embed it into the torus
- with every (directed edge) (v,w) associate a variable z_{vw}: the vector from v to w in the embedding
- constraints:

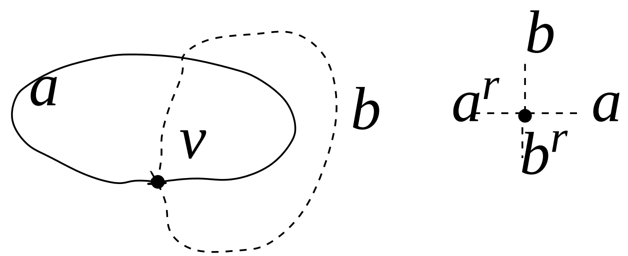
(symmetry)	$z_{vw} = -z_{wv}$ for all $(v,w) \in E$.
(center of gravity)	for all $v \in V$: $\sum_{w \in N(v)} z_{vw} = 0$.
(face sums)	for all faces f: $\sum_{e \in \delta f} z_e = 0$.

- E variables (since$z_{vw} = -z_{wv}$), $V+F$ equations
- (Euler's formula): $F-E+V = 2-2g = 0$ and hence $E = V+F$.
- two equations are redundant: one vertex and one face equation
- solution space is two-dimensional

  compute two linearly independent solutions, assign an arbitrary vertex to the origin, and compute x- and y-coordinates of the other vertices using the solutions

Drawings on the Torus II

- a map of genus one: one vertex v, two undirected edges a and b, one face



- with every (directed edge) (v,w) associate a variable z_{vw}: the vector from v to w in the embedding
- constraints:

(symmetry) $z_a = -z_{a^R}$ and similarly for b

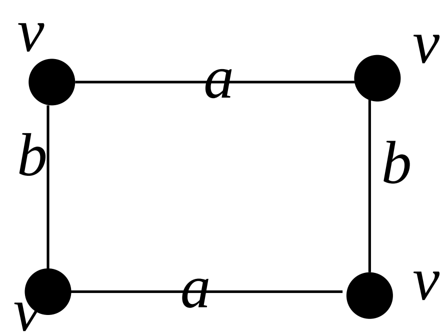
(center of gravity) $z_a + z_b + z_{a^R} + z_{b^R} = 0$

(face sums) one face: a, b, a^R, b^R.

- two variables, no constraint
- two independent solutions:

$$x_a = 1,\ x_b = 0 \qquad y_a = 0,\ y_b = 1$$

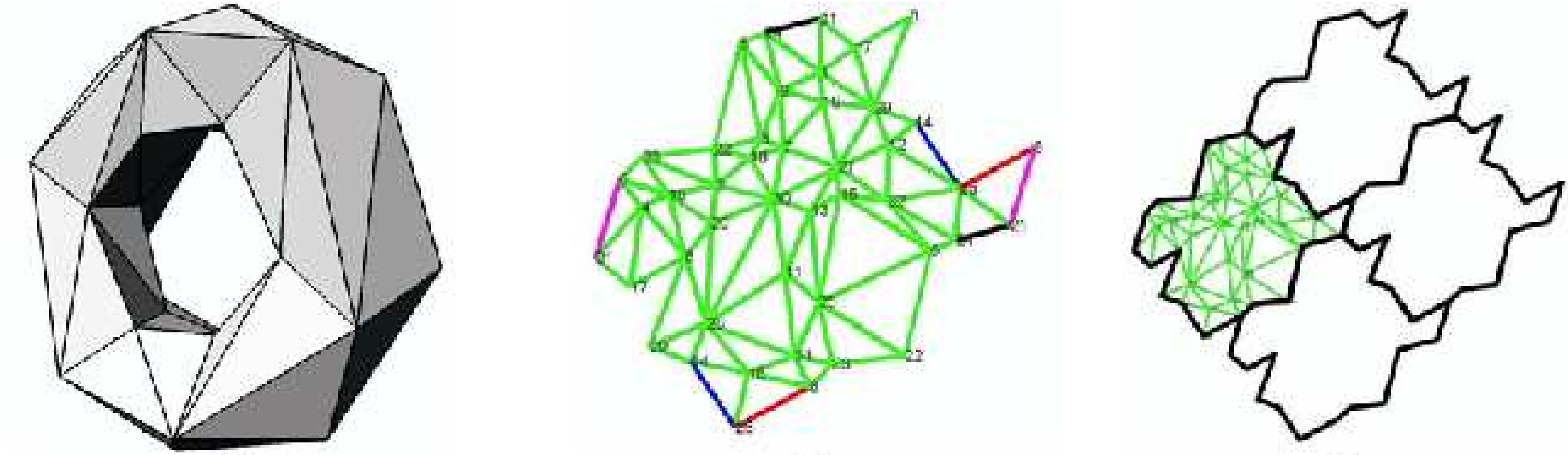
- after identification, this is a perfect drawing on the torus



Drawings on the Torus III

method generalizes Tutte's method

Gortler/Cotsman/Thurston: for a 3-connected map of genus one, the method produces an embedding with nondegenerate and disjoint faces



(a) (b) (c)

Figure 7: Parameterization of a torus containing 32 vertices and 64 faces. (a) 3D torus. (b) Parameterization of the torus to the plane using two harmonic one-forms generated with uniform weights. Vertices are numbered. The color coded edges along the boundary correspond. (c) Double periodic tiling of the plane using the drawing in (b).

Surface Reconstruction

given a point cloud P in $\mathbb{R}^3$ reconstruct the underlying surface S

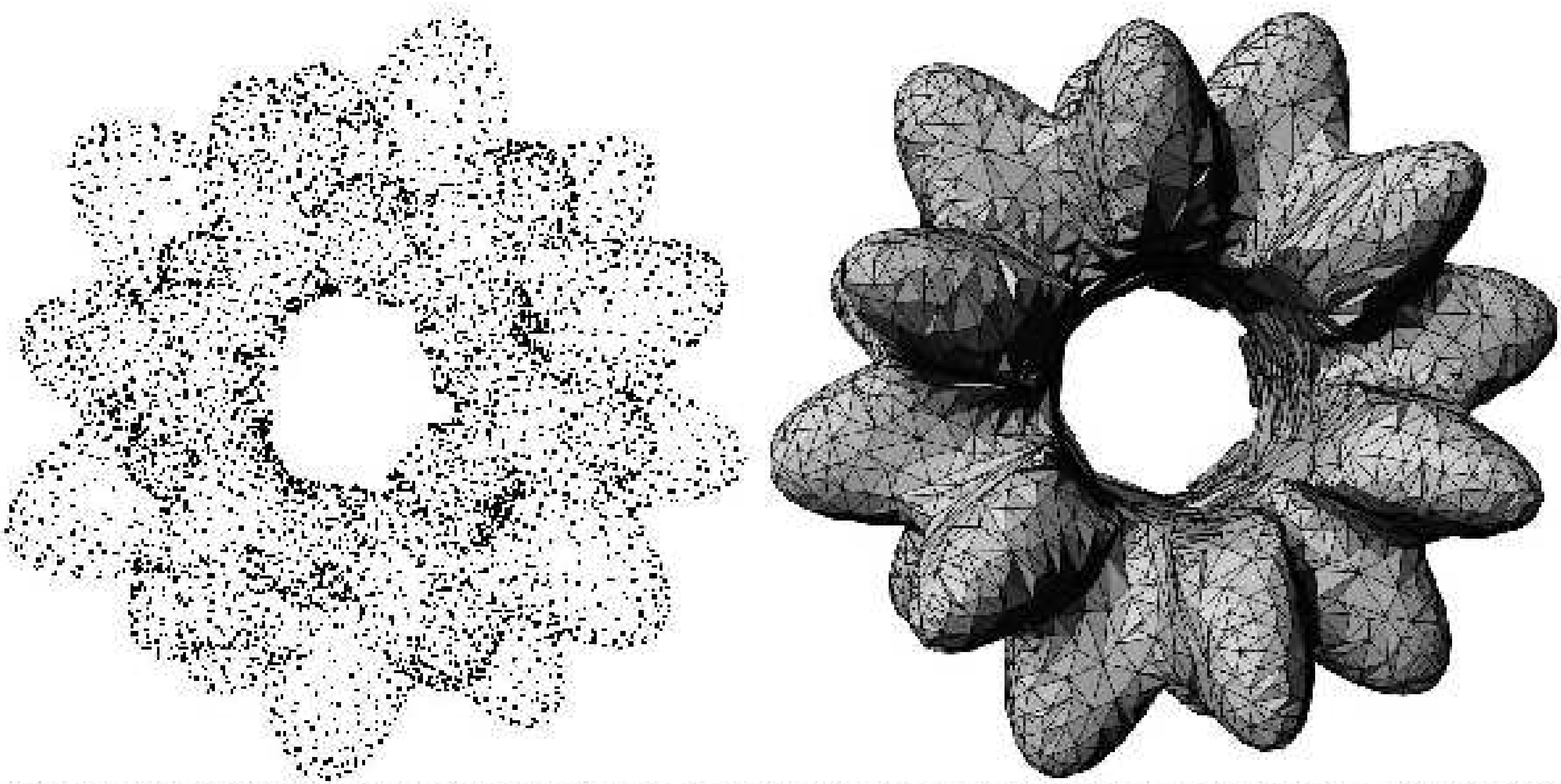


Figure 8: Reconstruction of the 7,371 point "bumpy torus" model. Parameters used were $k=7$, $t=10$, $d=10$ and no simulation of simplicity.

for this talk; point cloud comes from a surface of genus one

Reconstruction of Surfaces of Genus One

- P, point cloud (sampled from unknown surface S of genus one)
- Gotsman et al. suggest the following strategy:
  1. map P to the torus
  2. triangulate the embedded point set, say Delaunay
  3. lift triangulation to the original point set in three-space
- step one must preserve local structure (as in graph embedding)

Reconstruction of Surfaces of Genus One

- P, point cloud (sampled from unknown surface S of genus one)
- Gotsman et al. suggest the following strategy:
  1. map P to the torus
     - G_k symmetric nearest neighbor graph: (v, w) is an edge if w is one of the k-closest points to v and vice-versa.
     - use G_k instead of a genus-one-mesh in the embedding alg.
     - enforce face-sum-constraint for an appropriate (???) set of cycles
  2. triangulate the embedded point set, say Delaunay
  3. lift triangulation to the original point set in three-space
- step one must preserve local structure (as in graph embedding)

Which Cycles?

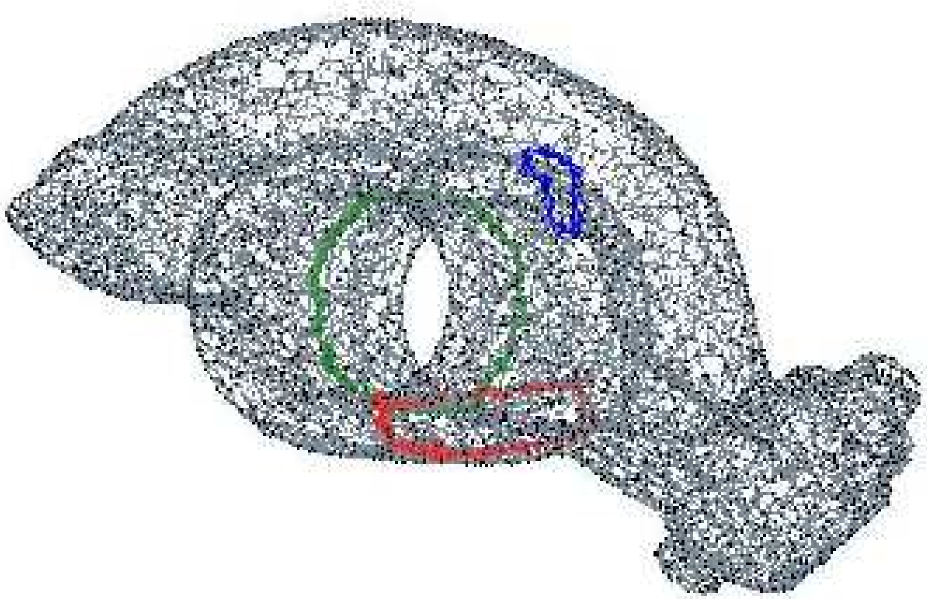


Figure 2: Three MCB cycles on a KNNG of a point cloud: trivial (blue) and non-trivial (red and green). The first should be closed and the latter two not.

- imagine a drawing of G_k on S
- want only cycles corresponding to trivial loops and a sufficient number of them
- do not want cycles correspon-ding to nontrivial loops

Which Cycles?

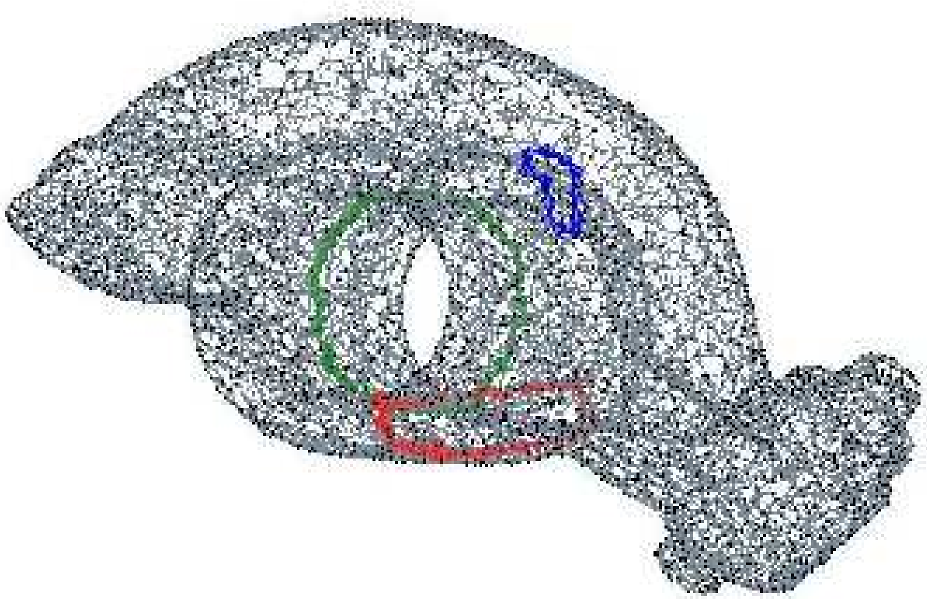


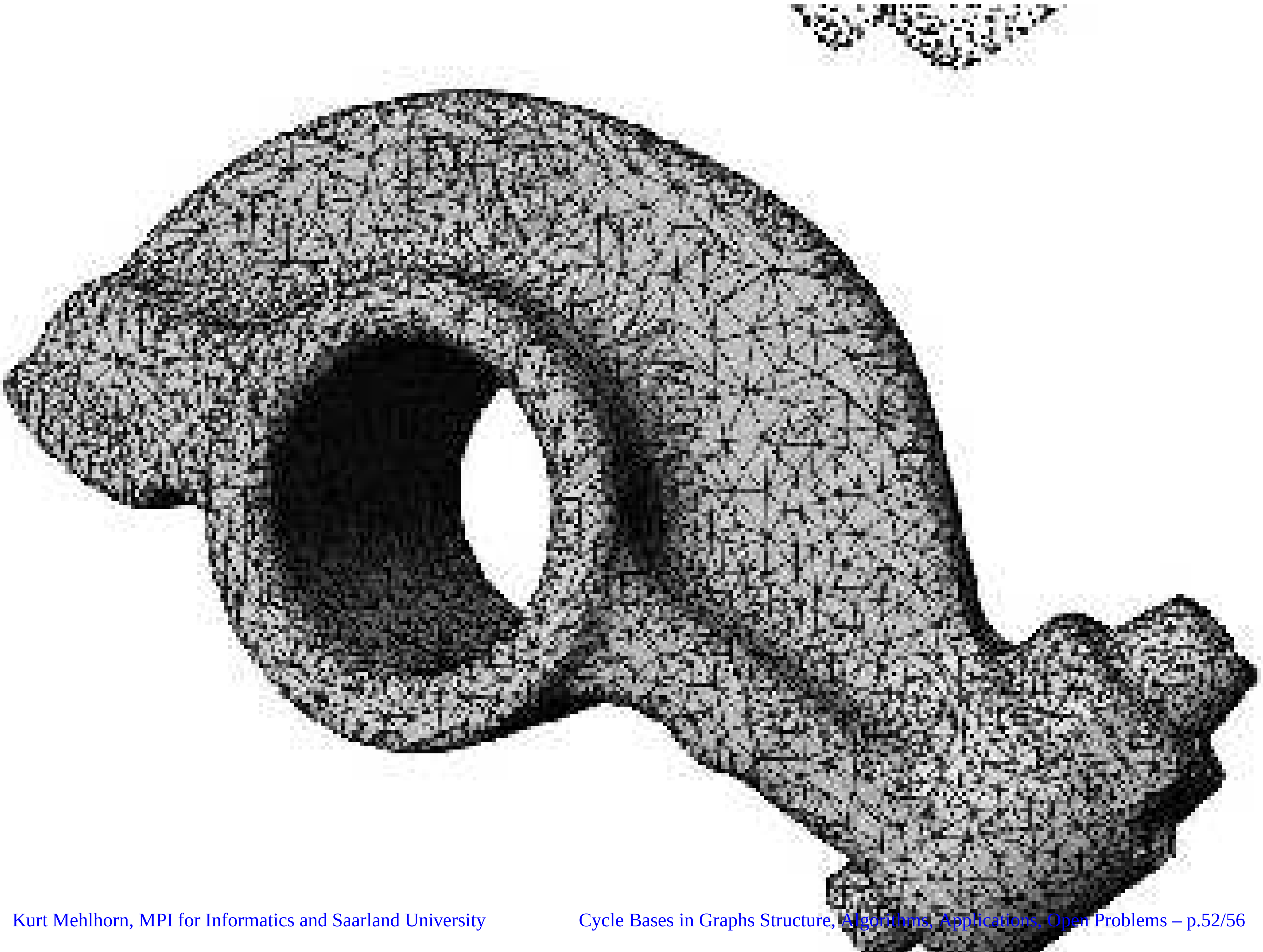
Figure 2: Three MCB cycles on a KNNG of a point cloud: trivial (blue) and non-trivial (red and green). The first should be closed and the latter two not.

- imagine a drawing of G_k on S
- want only cycles corresponding to trivial loops and a sufficient number of them
- do not want cycles corresponding to nontrivial loops

The following seems to work (experiments by Gotsman et al. using our impl.):

- compute a MCB of G_k, uniform edge costs
- in the MCB exactly two cycles are long and all others are short WHY
- the short ones form a basis for the trivial cycles USE THEM

Reconstruction for Rocker Arm



Some Intuition

- G_k has n nodes and m edges, cycle basis has $m - n + 1$ cycles
- every cycle basis must contain at least two cycles corresponding to nontrivial loops (= nontrivial cycles)
- if sample is sufficiently dense, nontrivial cycles are long

Some Intuition

- G_k has n nodes and m edges, cycle basis has $m-n+1$ cycles
- every cycle basis must contain at least two cycles corresponding to nontrivial loops (= nontrivial cycles)
- if sample is sufficiently dense, nontrivial cycles are long
- assume (wishful thinking)
  - G_k contains a mesh M for S, M has m' edges
  - consider the following set of cycles:
    - all but one face of M Euler tells us $f-m'+n=2-2g=0$
    - one cycle for each edge of G_k-M
    - in total, $f-1+(m-m')=m'-n-1+m-m'=m-n-1$ cycles
  - these cycles are independent; let us assume further that they are short (compared to the nontrivial cycles)
  - then there is a cycle basis in which all but two cycles are short

Some Intuition

- G_k has n nodes and m edges, cycle basis has $m-n+1$ cycles
- every cycle basis must contain at least two cycles corresponding to nontrivial loops (= nontrivial cycles)
- if sample is sufficiently dense, nontrivial cycles are long
- assume (wishful thinking)
  - G_k contains a mesh M for S, M has m' edges
  - consider the following set of cycles:
    - all but one face of M Euler tells us $f - m' + n = 2 - 2g = 0$
    - one cycle for each edge of $G_k - M$
    - in total, $f - 1 + (m - m') = m' - n - 1 + m - m' = m - n - 1$ cycles
  - these cycles are independent; let us assume further that they are short (compared to the nontrivial cycles)
  - then there is a cycle basis in which all but two cycles are short
- Thus MCB contains exactly two long cycles and the short cycles in MCB span the trivial cycles

A Theorem

Assume S is smooth, P is dense, and k sufficiently large

- for $x \in S$: $f(x)$:= distance from x to Voronoi diagram of S
- for every $x \in S$ there is a $p \in P$ with $||x - p|| \leq \varepsilon f(x)$
- if $p, q \in P$ and $p \neq q$ then $||p - q|| \geq \delta f(p)$
- $\varepsilon = 0.01$, $\delta = \varepsilon / 10$, k about 100

A Theorem

Assume S is smooth, P is dense, and k sufficiently large

- for $x \in S$: $f(x)$:= distance from x to Voronoi diagram of S
- for every $x \in S$ there is a $p \in P$ with $||x - p|| \leq \varepsilon f(x)$
- if $p, q \in P$ and $p \neq q$ then $||p - q|| \geq \delta f(p)$
- $\varepsilon = 0.01$, $\quad \delta = \varepsilon/10$, $\quad k$ about 100

- (Amena/Bern) G_k contains a mesh for S
- all cycles in the set described above are short: lenght at most $2k + 3$
- all nontrivial cycles are long: length at least least $4k + 6$.
- Theorem: the short cycles in MCB span the space of trivial cycles and MCB contains exactly two long cycles

- experiments work with much large values of ε and much smaller values of k

Open Problems for this Approach to Surface Reconstruction

MAX-PLANCK-GESELLSCHAFT

- guarantees for the triangulation
- extension to surfaces of higher genus
- extension to nonsmooth surfaces
- show that methods works for larger ranges of ε and k
- faster algorithms for MCB
  - smaller set of candidate cycles
  - approximation algorithms
  - further applications

Thank you for your attention