Minimum Cycle Bases
Algorithms and Applications

Kurt Mehlhorn
Max-Planck-Institut für Informatik
Saarbrücken
joint work with
C. Gotsmann, R. Hariharan, K. Kaligosi, T. Kavitha, D. Michail, K. Paluch, E. Pyrga

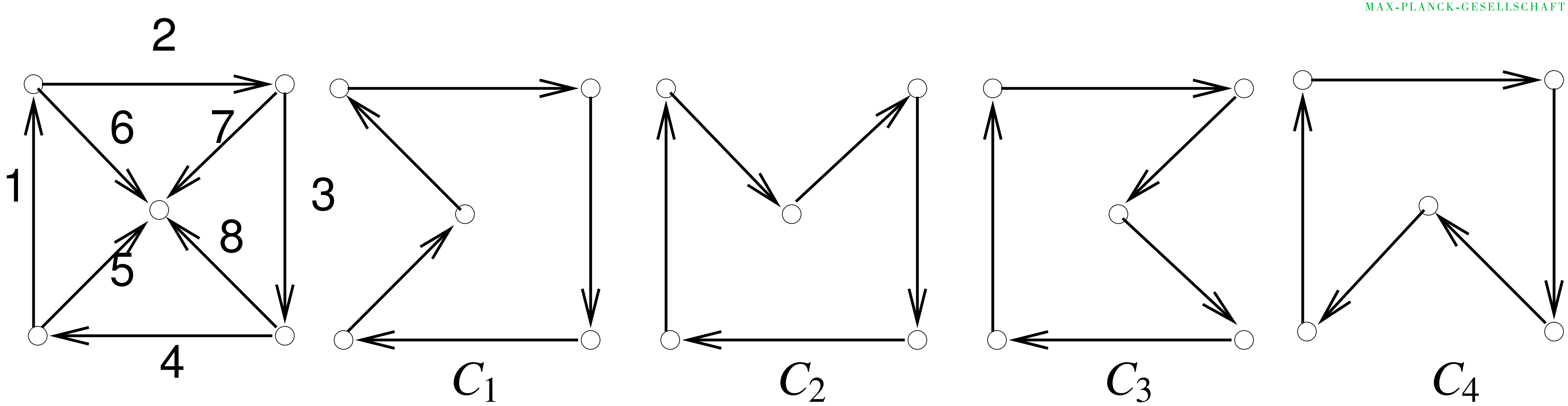
Overview

MAX-PLANCK-GESELLSCHAFT

- Problem Definition
- Motivation
- Undirected and Directed Cycle Basis
  - Algorithmic Approaches: Horton and de Pina
  - Exact and Approximate
- Integral Cycle Basis
- Application to Surface Reconstruction

Slides and papers available at my home page

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

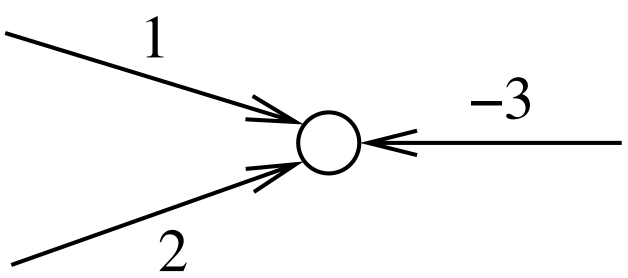
- cycle C = set of edges such that degree of every vertex wrt C is even
- $C = (m(e_1), m(e_2), \ldots, m(e_m)) \in \{0, 1\}^E$
- $m(e_i) = 1$ iff e_i is an element of C

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

- every basis consists of $N = m - (n-1)$ cycles
- *spanning tree basis:*
  - let T be an arbitrary spanning tree
  - for every non-tree edge e,
    e + the T-path connecting the endpoints of e.

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

MAX-PLANCK-GESELLSCHAFT

- let T be an arbitrary spanning tree $N =$ 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,
  - cycles in $\mathscr{B}$ are independent
  - they span all cycles: for any cycle C, we have

$$C = \sum_{e \in N \cap C} \lambda_e \cdot C_e$$

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

Pf: $C - \sum_{e \in N \cap C} \lambda_e \cdot C_e$ 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*

Motivation I

MAX-PLANCK-GESELLSCHAFT

- 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 small basis (uniform weights) or a minimum weight basis (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

Algorithmic Approach 1: Horton

- compute a sufficiently large set of 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 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

Algorithmic Approach 1: Horton

- compute a sufficiently large set of 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 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

  Horton set: for any edge $e = (a, b)$ and vertex v take the cycle $C_{e,v}$ consisting of e and the shortest paths from v to a and b.

  $O(nm)$ cycles, Gaussian elimination on a $nm \times m$ matrix

  running time $O(nm^3)$ or $O(nm^\omega)$

Algorithmic Approach 2: de Pina

- construct basis iteratively, assume partial basis is $\{C_1, \ldots, C_i\}$
- compute a vector S orthogonal to $C_1, \ldots, C_i$.
- find a cheapest cycle C having a non-zero component in the direction S, i.e., $\langle C, S \rangle \neq 0$
- add C to the partial basis

- C is **not** the cheapest cycle independent of the partial basis
- it is the shortest vector with a component in direction S.
- correctness
  - alg computes a basis
  - alg computes a minimum weight basis, because every basis must contain a cycle which has a non-zero component in direction S
  - and alg adds the cheapest such cycle

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 $O(m^3)$.
- further speed-up: update in bulk
  - update $S_{N/2+1}$ to S_N only after computation of C_1 to $C_{N/2}$
  - and use this idea recursively
  - now fast matrix multiplication and inversion can be used for update

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 $e \in S$ changes sides, and edges $e \notin S$ do not
  - more precisely: for $e = (v, w) \in S$ have (v^0, w^1) and (v^1, w^0)
  - and for $e = (v, w) \notin S$ have (v^0, w^0) and (v^1, w^1)
- 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

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 algorithms

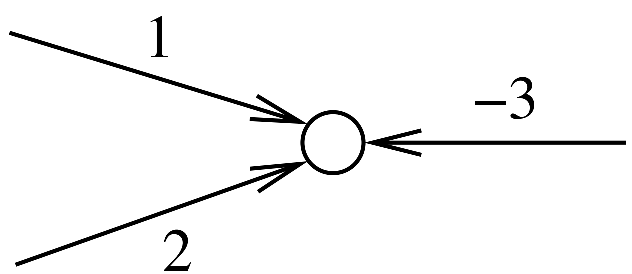
Implementation

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

MAX-PLANCK-GESELLSCHAFT

- 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 $\rightarrow$ 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

MAX-PLANCK-GESELLSCHAFT

- A $2k-1$ approximation can be computed in time $O(kmn^{1+1/k} + mn^{(1+1/k)(\omega-1)})$ Kavitha/Mehlhorn/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 $(2k-1)w(MCB)$
- shortest cycle multiset has weight at most $w(MCB)$
- more involved argument: joint weight is bounded by $(2k-1)w(MCB)$

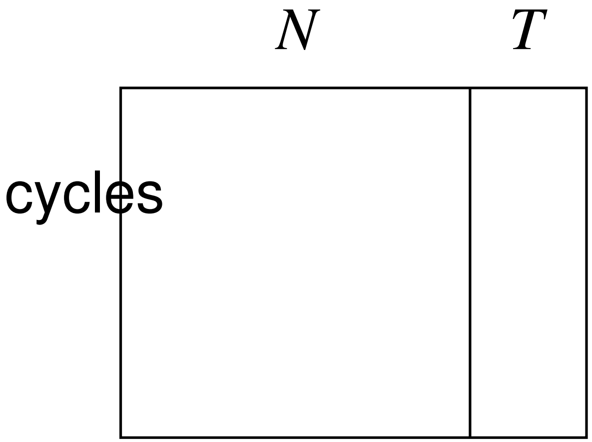
open problem: better approximation algorithms

Integral Basis

- a basis is integral if every cycle is an integral linear combination . . .

- spanning tree basis is integral

- Liebchen and Rizzi: characterization theorem



- $T =$ any spanning tree, $N =$ non-tree edges
- basis is integral iff determinant of square matrix is one
- value of determinant does not depend on choice of T

- integral cycle bases are relevant for integer linear programming

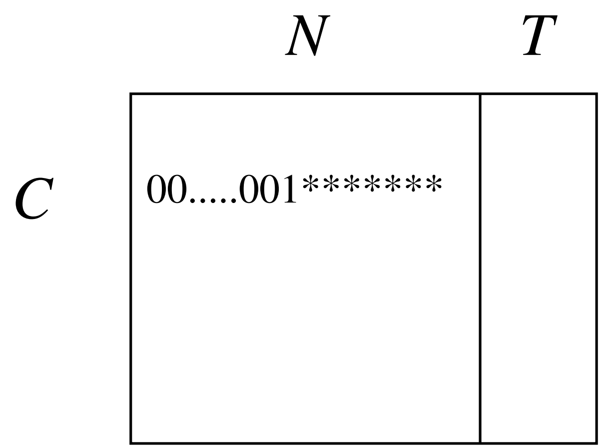
- open problem: is minimum integral cycle basis in P?

Approximation Alg for Integral Basis

- 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):
  - view paths of degree two nodes as superedges
  - find short cycle of $2\log n$ superedges
  - add cycle to basis and delete the heaviest superedge from the graph

- weight of cycle is at most $2\log n$ times weight of deleted edges
- edges in superedge: add all but one to spanning tree



Surface Reconstruction

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

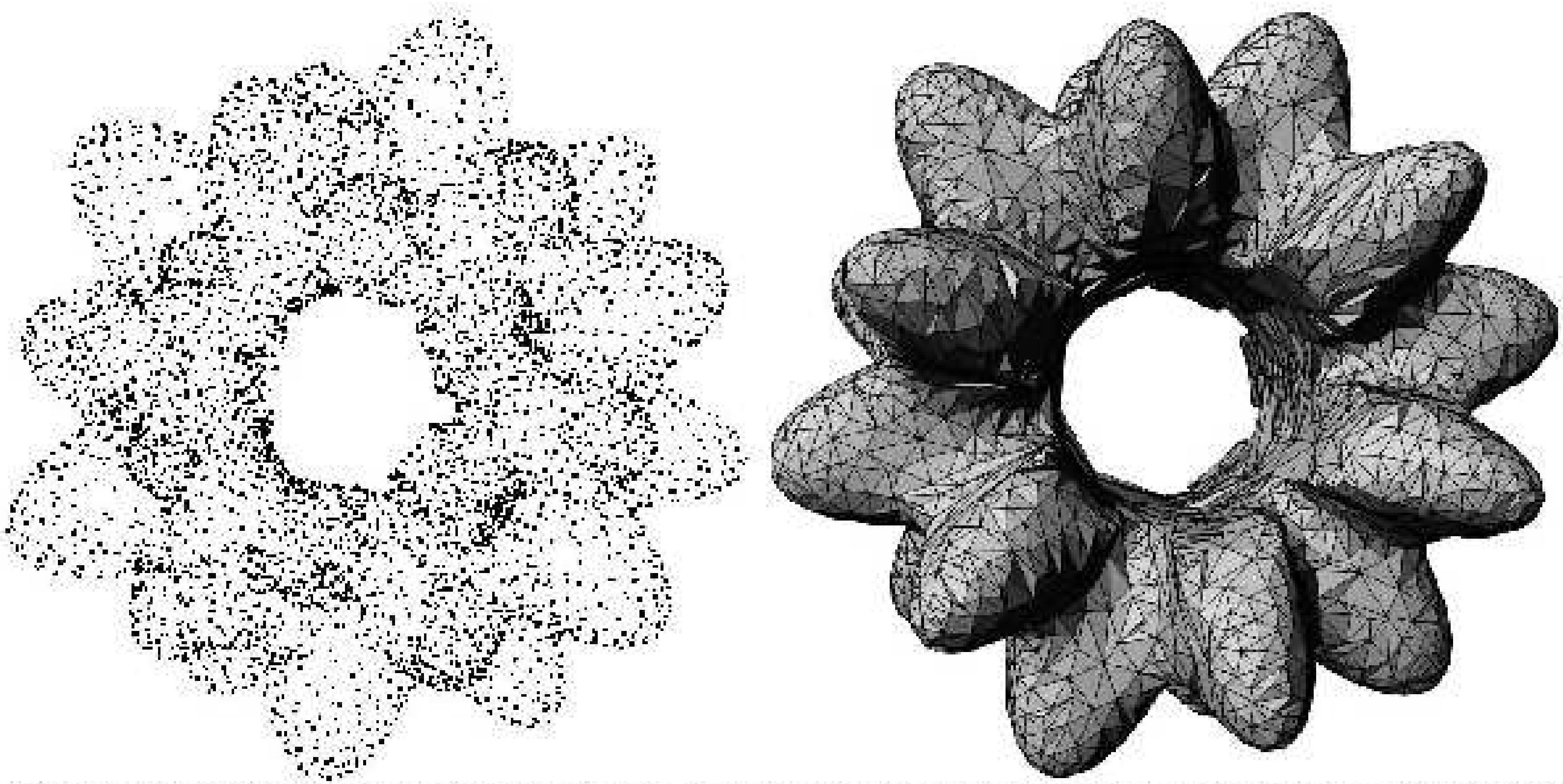
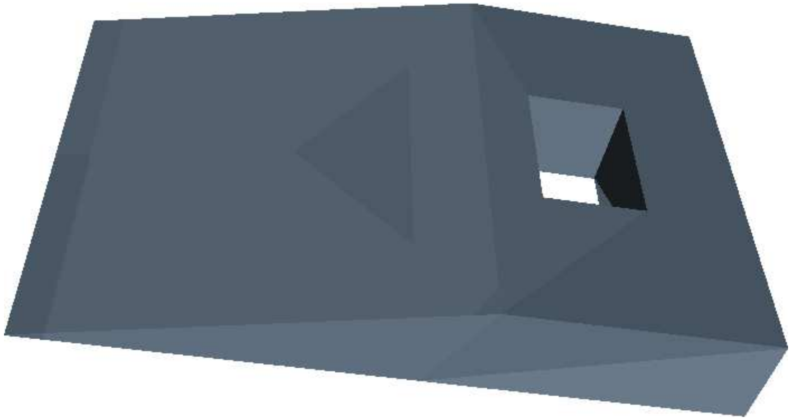
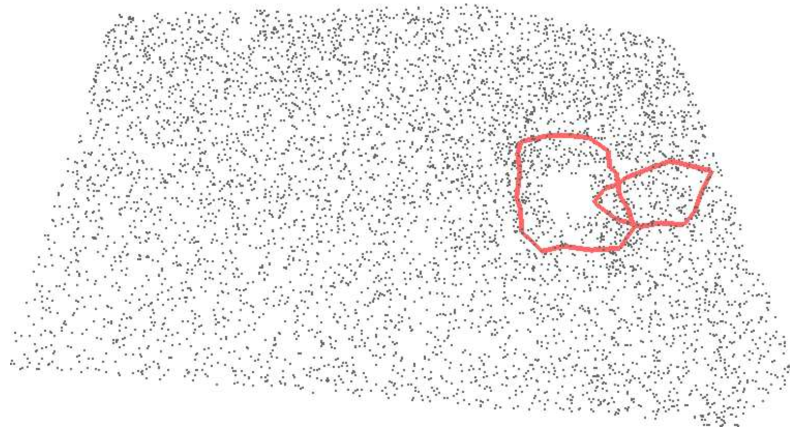


Figure 8: Reconstruction of the 7,371 point "bumpy torus" model. Parameters used were k=7, t=10, d=10 and no simulation of simplicity.

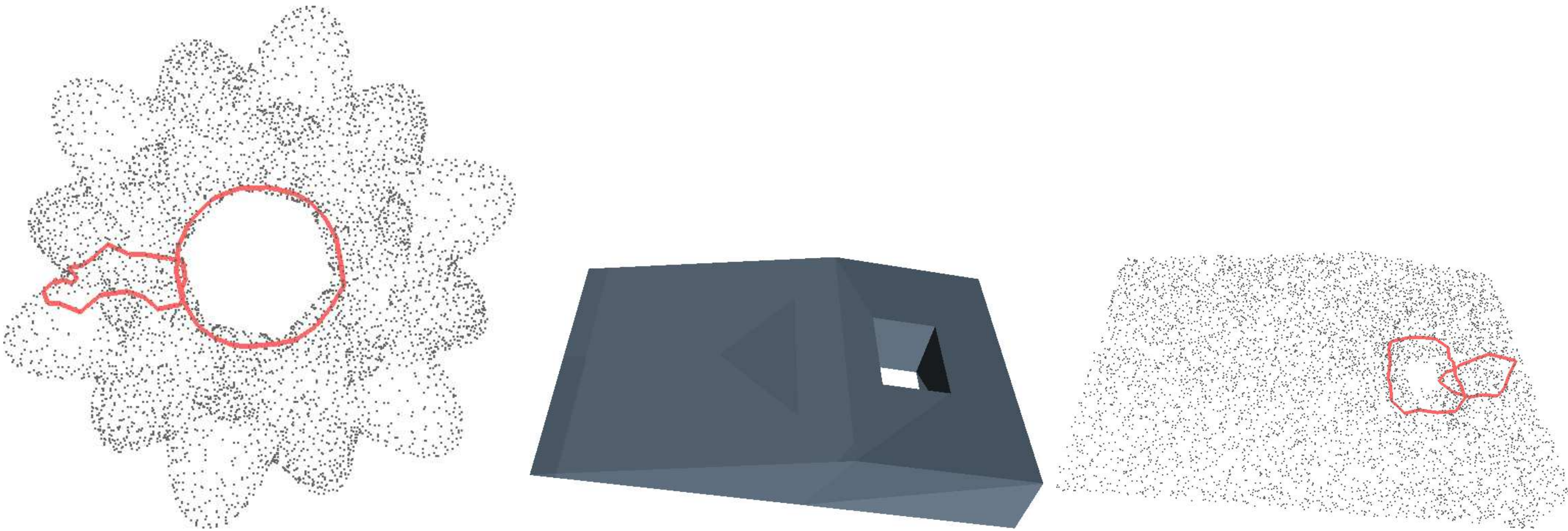
for this talk; point cloud comes from a surface of genus one

Beyond Smooth Surfaces: Cocone Reconstruction





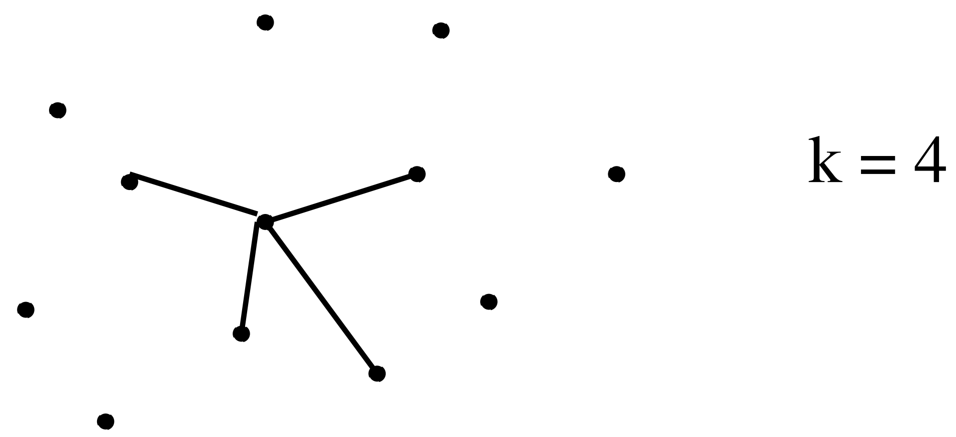
Beyond Smooth Surfaces: Genus Detection I



- 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



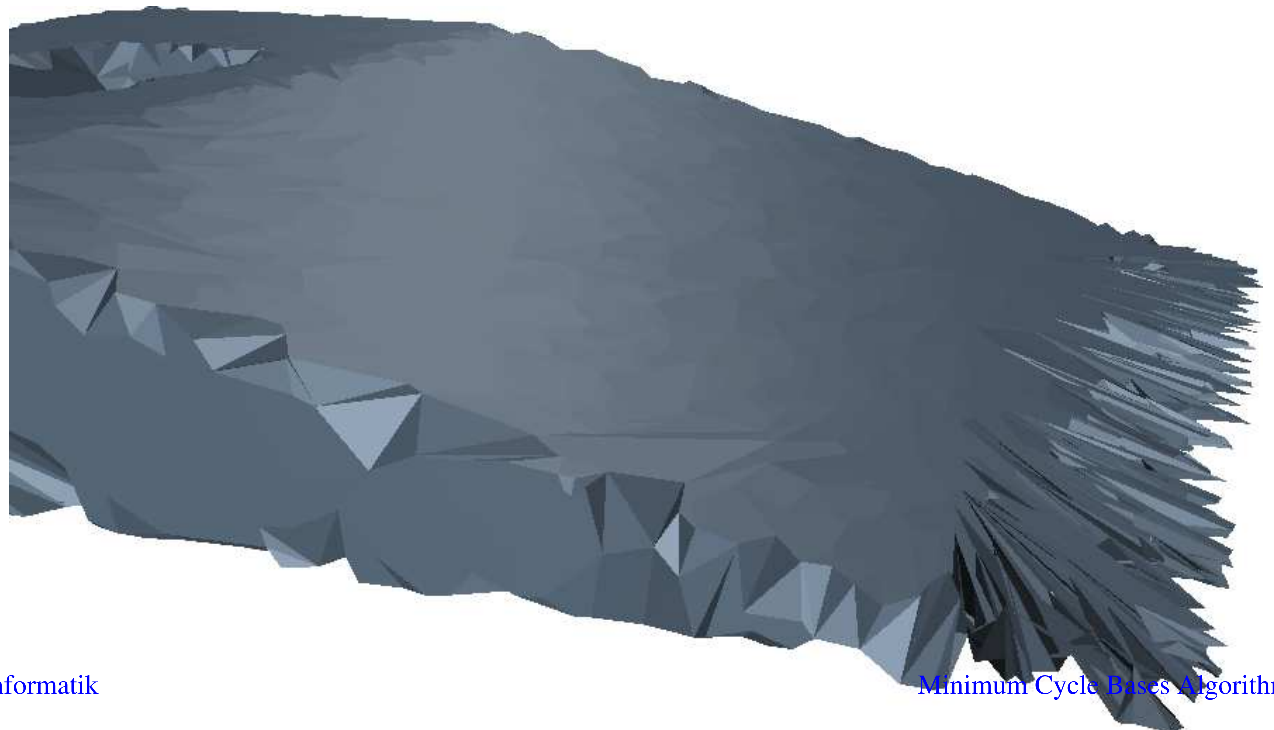
  - easy to construct
- Theorem (Gotsman/Kaligossi/Mehlhorn/Michail/Pyrga 05): if S is smooth, P is sufficiently dense, and k appropriately chosen:

  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



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