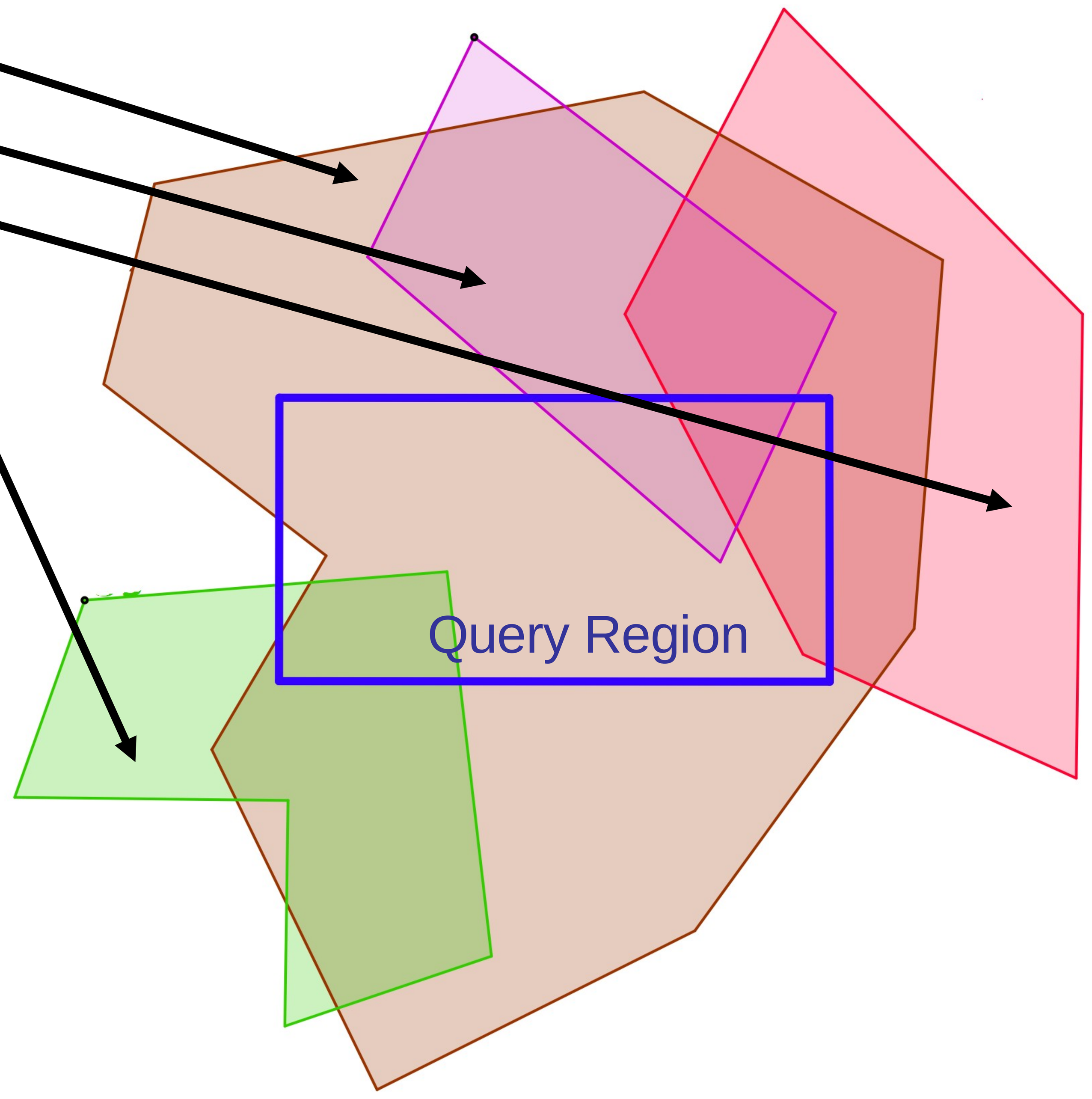
Persistent Spatial Data Structures

Stuart A. MacGillivray and Bradford G. Nickerson
Faculty of Computer Science, University of New Brunswick, Fredericton, New Brunswick, Canada

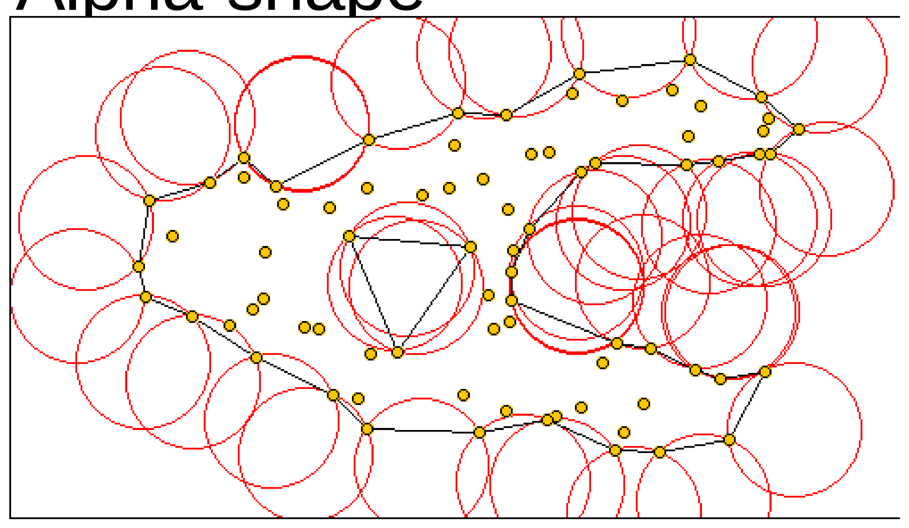
Problem Definition and Motivation

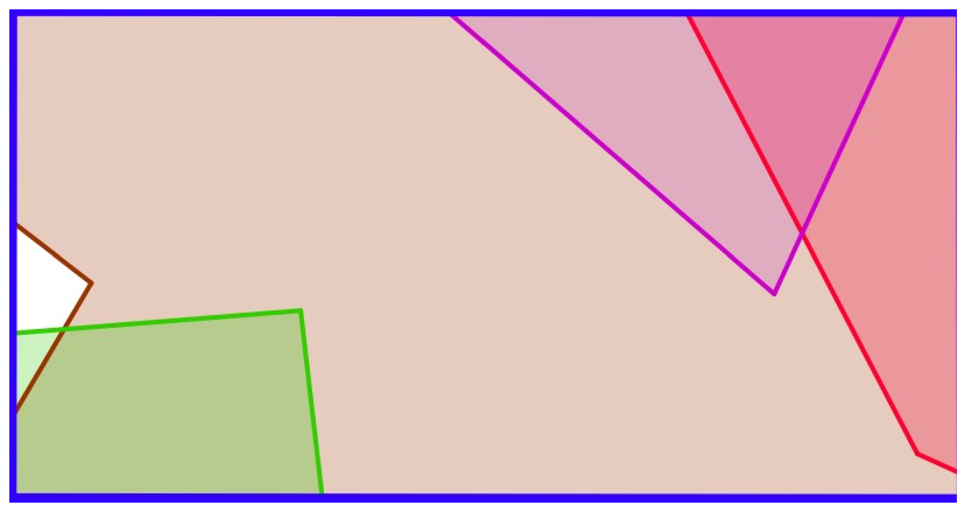
- Sets of spatial data points Si added at distinct time intervals. $T_i = [t^b_i, t^e_i]$ with $t^b_i > t^e_{i-1}$
- Range searches only return most recent data where sets overlap.
- Irregular update regions may require alpha-shapes for definition..
- Motivation: Queries on time-stamped massive data surveys.
- Challenge: Massive data requires I/O model; complex queries.

Set S_0, interval T_0
Set S_1, interval T_1
Set S_2, interval T_2
Set S_3, interval T_3



Alpha-shape

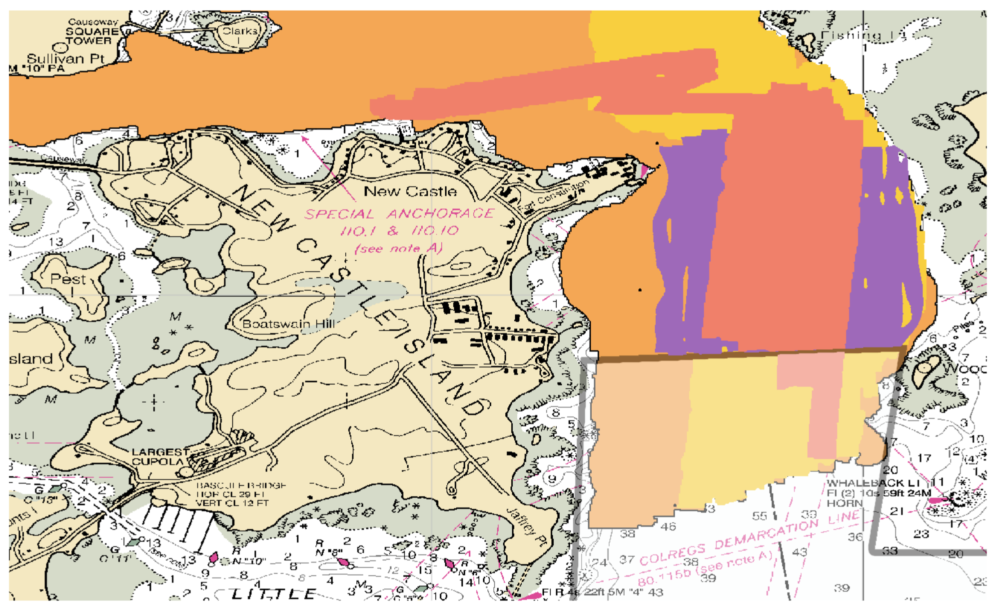




Query $Q=(R,\{T_0,T_1,T_2,T_3\})$

Testing Plans

- Data sets: Shallow Survey 2008 Common Dataset (CCOM/JHC), as well as a synthetic data set.
- SSCD contains >580 GB of survey data; sets of 4-D point data range from 2 to 28 GB.
- Synthetic data set for initial testing randomly generated.

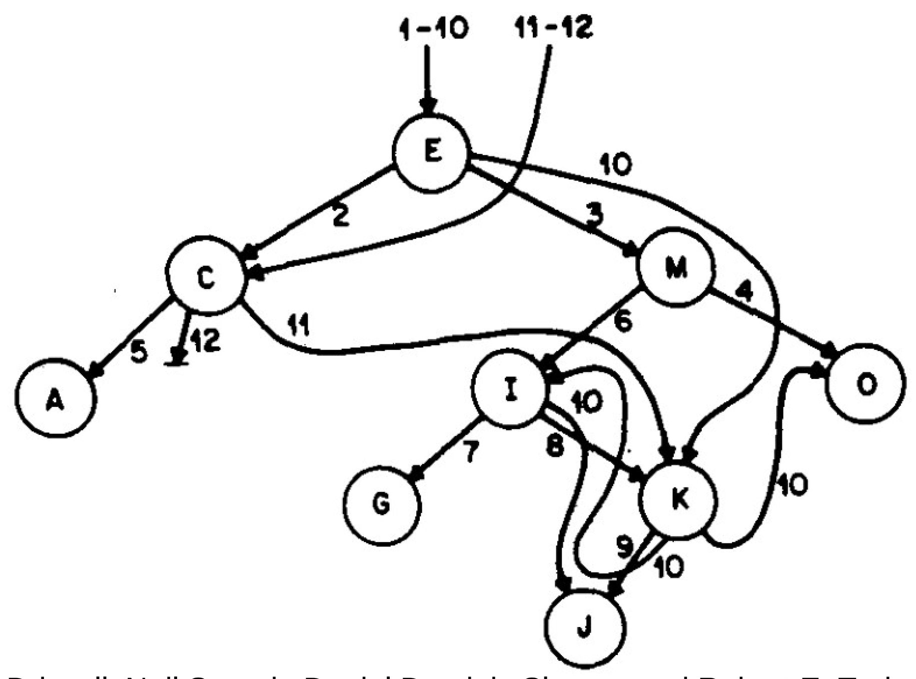


From Shallow Survey 2008. http://www.shallowsurvey2008.org/

Persistent Data Structures

- Persistent data structures maintain versioning history and data, retaining search complexity.
- Alterations tracked between versions of structure.
- Multiple styles of persistence for different methods of data management, e.g. source code version control.
- Partial persistence: Queries on past versions possible, can only modify the most recent version.
- Full persistence: Edits to past versions create alternate branches, forming a tree of versions.
- 'Exclusion' persistence: Queries on any version can ignore any subset of past updates.

Diagram of a 'fat node' approach to partial persistence.



From James R. Driscoll, Neil Sarnak, Daniel Dominic Sleator, and Robert E. Tarjan. Making data structures persistent. J. Comput. Syst. Sci., 38(1):86-124, 1989.

Planned Approaches

- Compare the priority R-tree to the multi-level grid file, along with other spatial data structures..
- Optimal 2-D data structures storing N points require O(N log N/log log N) space for queries to return K points in O(log N + K) time.
- Can persistent structures match this?
- Can updates be performed in O(M log N) time for M points added at interval T_i?
- Can persistent spatial data structures be made I/O-efficient?
- Is 'exclusion' persistence possible, and what time and space bounds constrain it?

References: Peyman Afshani, Lars Arge, and Kasper Dalgaard Larsen. Orthogonal range reporting in three and higher dimensions. In FOCS, pages 149-158. IEEE Computer Society, 2009.
James R. Driscoll, Neil Sarnak, Daniel Dominic Sleator, and Robert E. Tarjan. Making data structures persistent. J. Comput. Syst. Sci., 38(1):86-124, 1989.
Neil Sarnat, Robert E. Tarjan. Planar Point Location Using Persistent Search Trees. CACM, 29(7):669-679, 1986.
Shallow Survey 2008. http://www.shallowsurvey2008.org/

Sponsored by: Faculty of Computer Science UNB; NSERC CRSNG