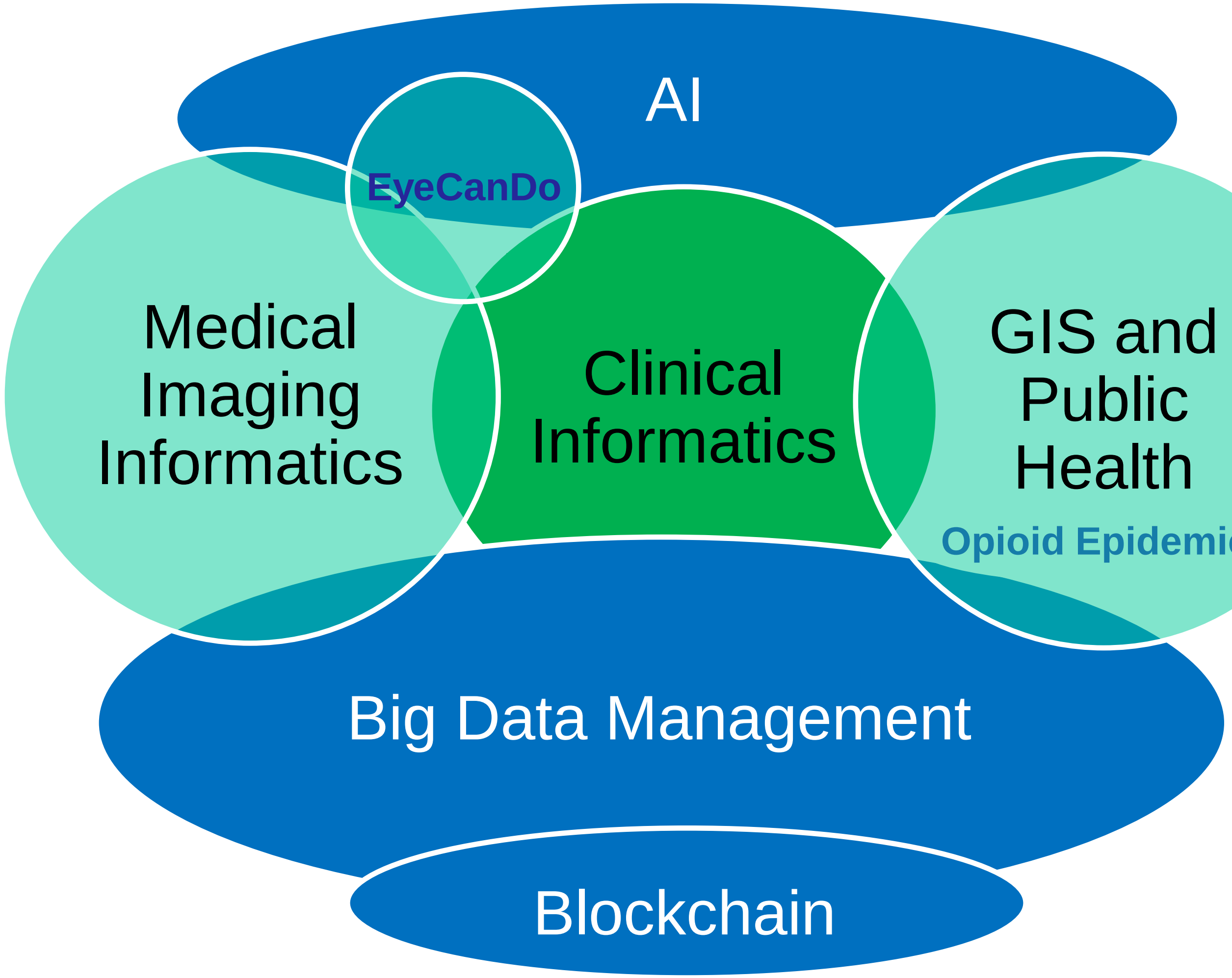
Spatial Data Management at Extreme Scale: from GIS to Medical Imaging

Fusheng Wang

Department of Computer Science
Department of Biomedical Informatics
Stony Brook University

CSE600 Seminar, November 19, 2021

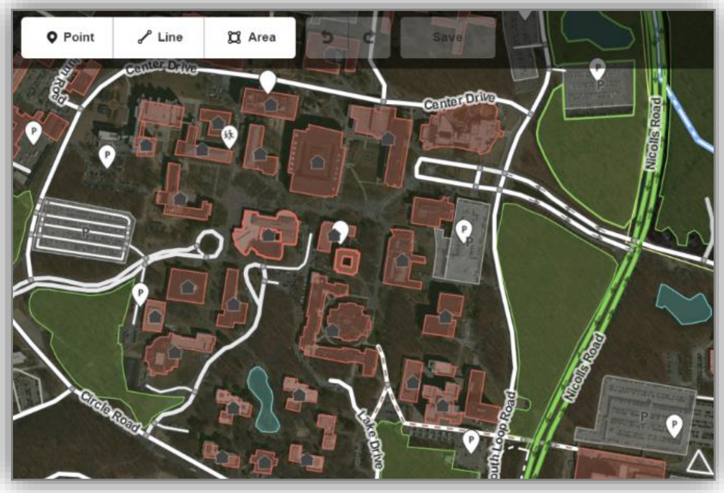

AI
EyeCanDo
Medical Imaging Informatics
Clinical Informatics
GIS and Public Health
Opioid Epidemic
Big Data Management
Blockchain


CSIRE K12 Summer Research Program
CSE393: Intro to Biomedical Informatics
CSE532: Theory of Databases
BMI513: Medical Imaging Informatics

Spatial Big Data: GIS

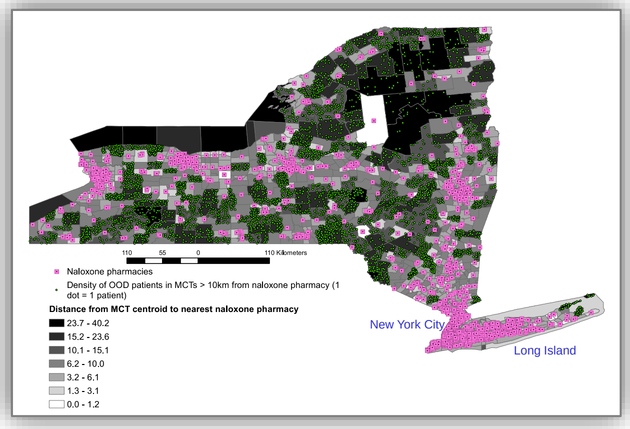
Satellite imagery

Drones



VGI

Smart phone data



GIS for public health

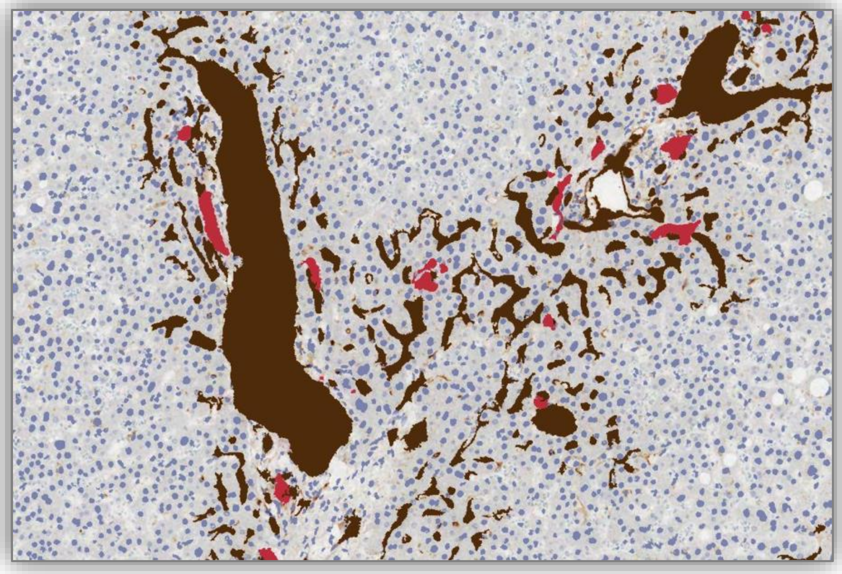
HD map and self-driving

Smart cities

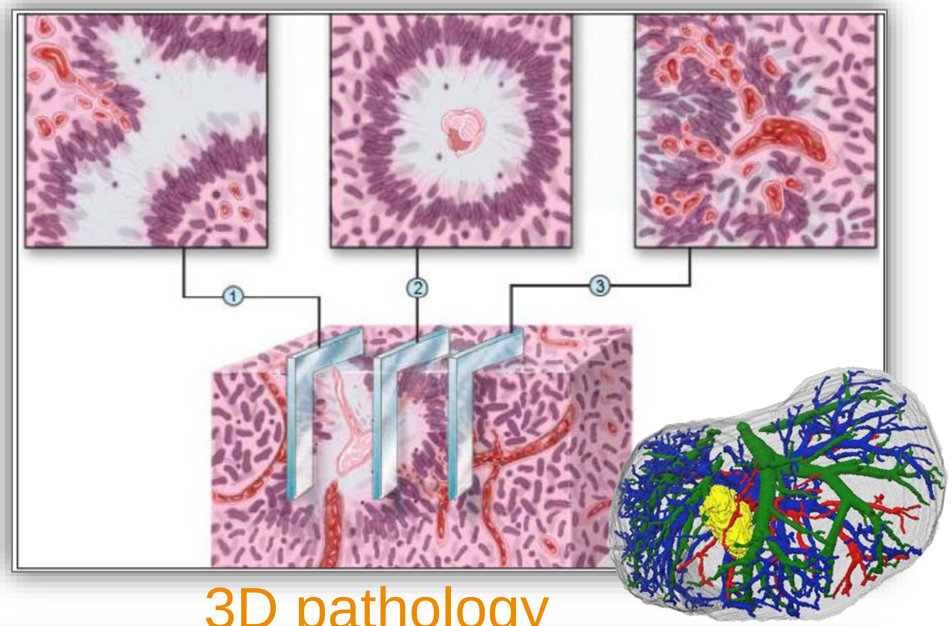
Future manufacturing

Spatial Big Data: Medical Imaging

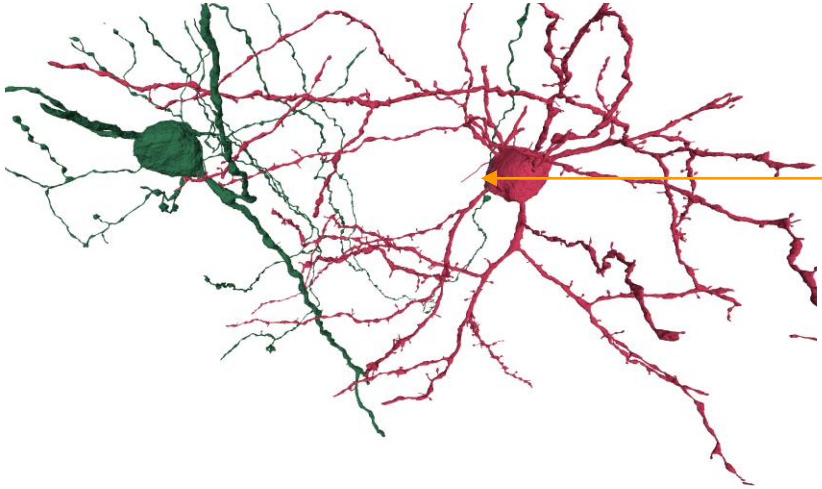
Stony Brook University



2D digital pathology



3D pathology



Brain Atlas

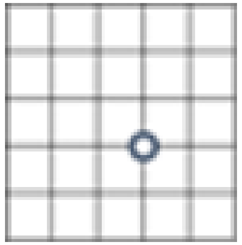
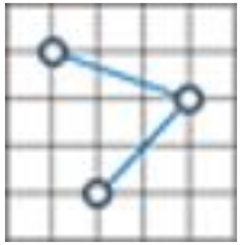
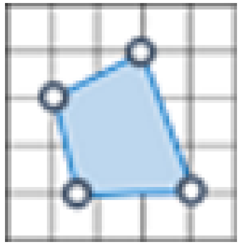
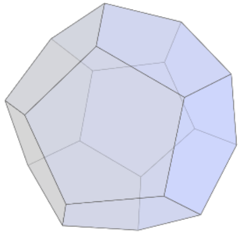
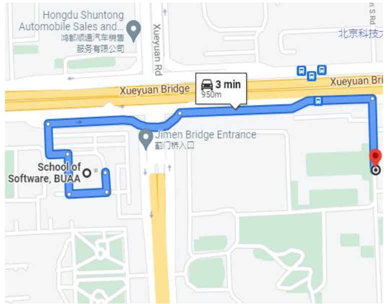
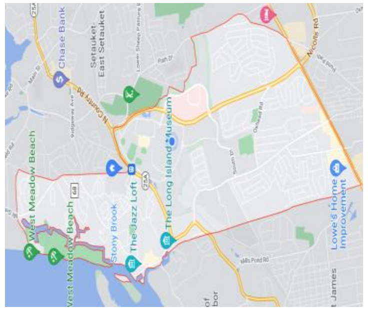
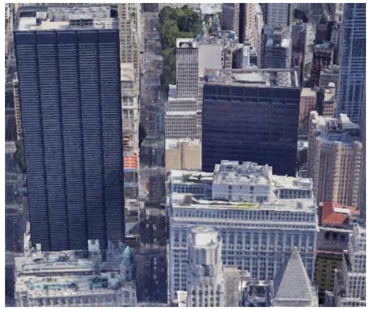
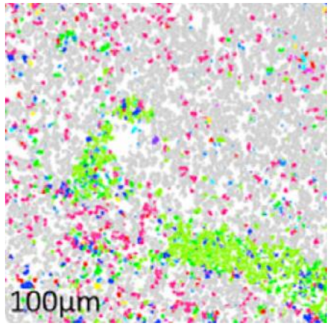
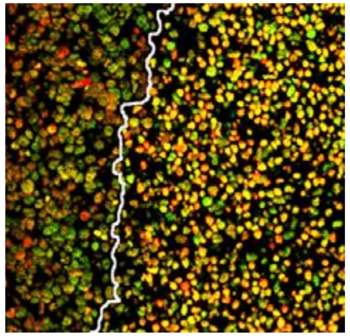
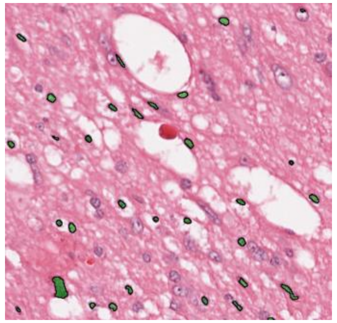
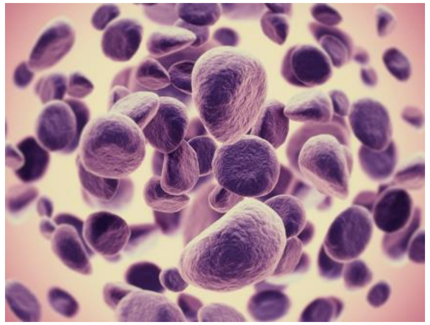
Synapses

Spatial Big Data: The Human Body Atlases at Multi-Scales

Stony Brook University

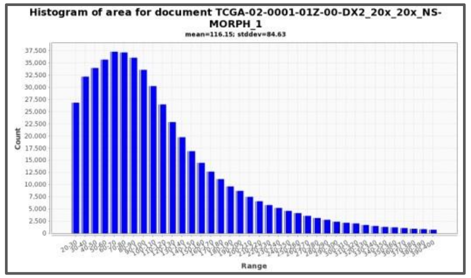
- The NIH Human BioMolecular Atlas Program (HuBMAP) is to develop an open and global platform to map healthy cells in the human body
- The proper functioning of organs and tissues is dependent on the interaction, spatial organization, and specialization of all our cells
- There are 37 trillion cells in an adult human body

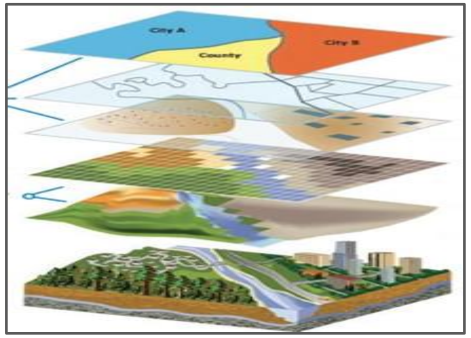
Typical Spatial Objects

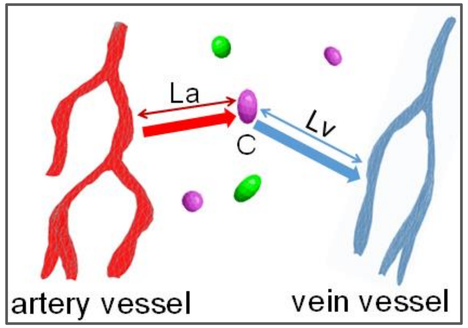
Point	Line string	Polygon	Polyhedron	
				
				GIS
 Biomarker	 Tumor-immune border	 Cells	 Tumor cells	Human body

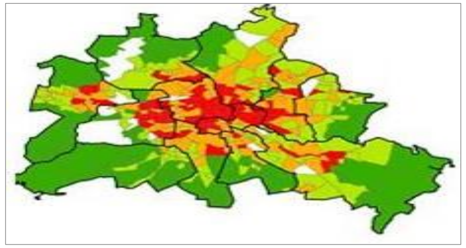
Spatial Queries and Analytics

- Feature based descriptive queries
  - Feature based filtering or feature aggregation
- Spatial relationship based queries
  - Spatial join, window, point-in-polygon
- Distance based queries
  - Nearest neighbors, proximity estimation
- Spatial analysis and mining
  - Find spatial clusters, hotspots, and anomalies
  - Correlation, regression, spatial relationship modeling, e.g., GWR
  - Colocalization



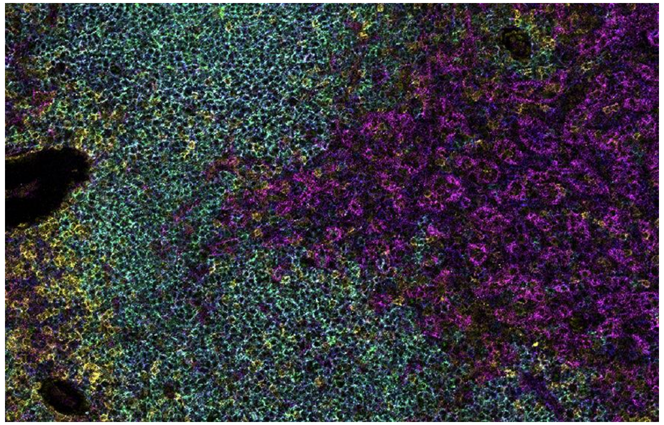




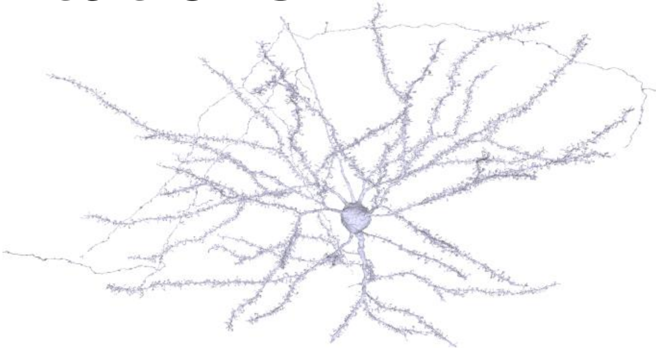


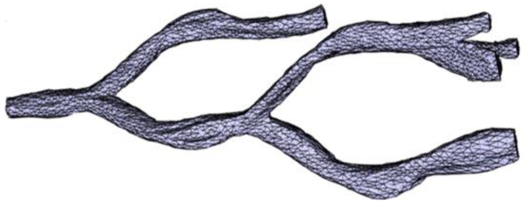
Challenges

- Explosion of spatial data
  - Billions of geo-tagged tweets, hundreds of millions of polygons in OSM, 37 trillions of cells per person
  - High I/O and communication cost for data processing

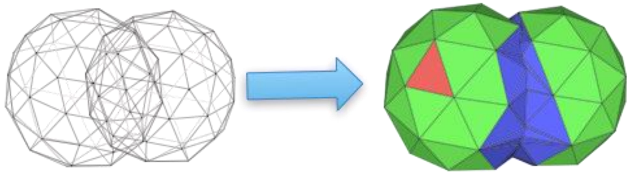


- Complex structures and representations
  - Arbitrary shapes
  - Bifurcations in blood vessel
  - Multiple levels of detail (LOD)





- High computational complexity
  - Polygons/polyhedrons with many edges/faces
  - Heavy duty geometric computation



Spatial Database Management Systems (SDBMS)

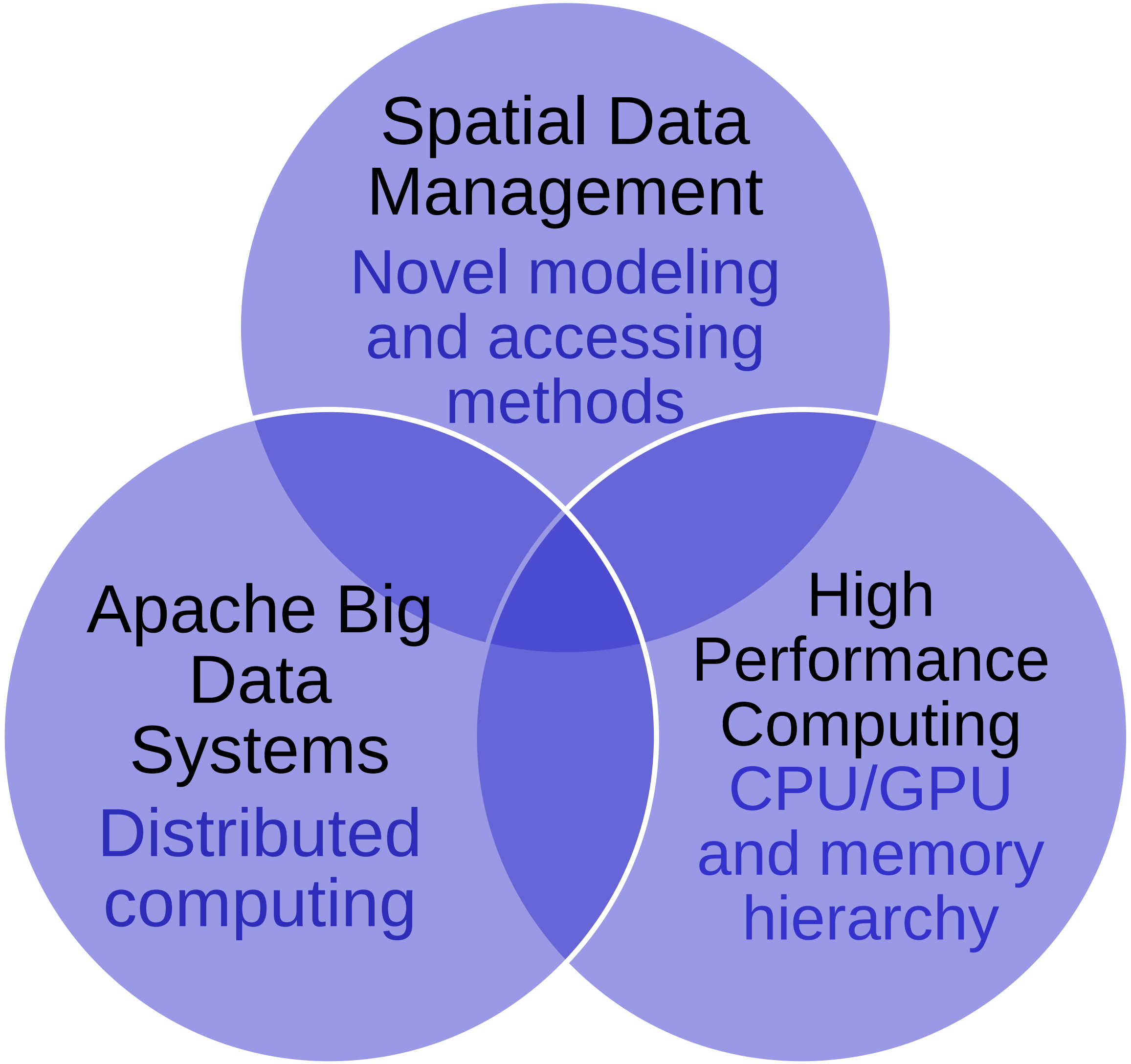
Postgres	Oracle	MySQL	DB2	MongoDB
PostGIS	Oracle Spatial&Graph	MySQL Spatial	DB2 Spatial Extender	MongoDB Geospatial
				

- Spatial data type extensions
- Multi-dimensional indexing methods
- Limitations: limited scalability; data injection problem

```
CREATE TABLE locations (name VARCHAR, coord GEOMETRY);

INSERT INTO locations VALUES (
    ('Point', 'POINT(0 1)');,
INSERT INTO locations VALUES (
    ('Point', 'POINT(3 0))');
INSERT INTO locations VALUES (
    ('Point', 'POINT(1 2))');

CREATE INDEX loc_idx ON locations USING GIST (coord);

SELECT name FROM geometries
WHERE ST_DISTANCE(coord, 'Point(1 1)') < 2.0;
```

Apache Big ("Non-Spatial") Data Systems

Vision of Spatial Big Data Computing



Scalable and Efficient Spatial Big Data Systems

[ICDE21] SPEAR

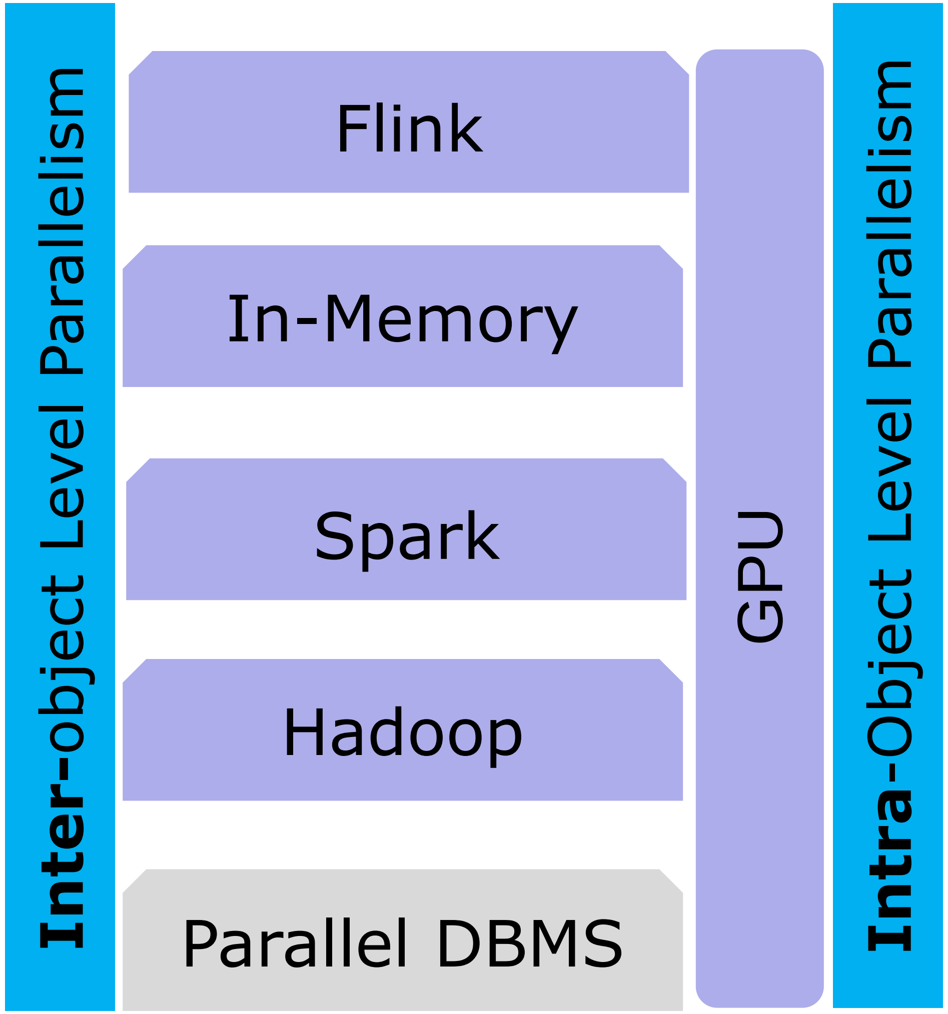
[SIGSPATIAL18] [VLDB2019] 3DPro iSPEED

[MDM21] IDEAL

[SIGSPATIAL17] SparkGIS

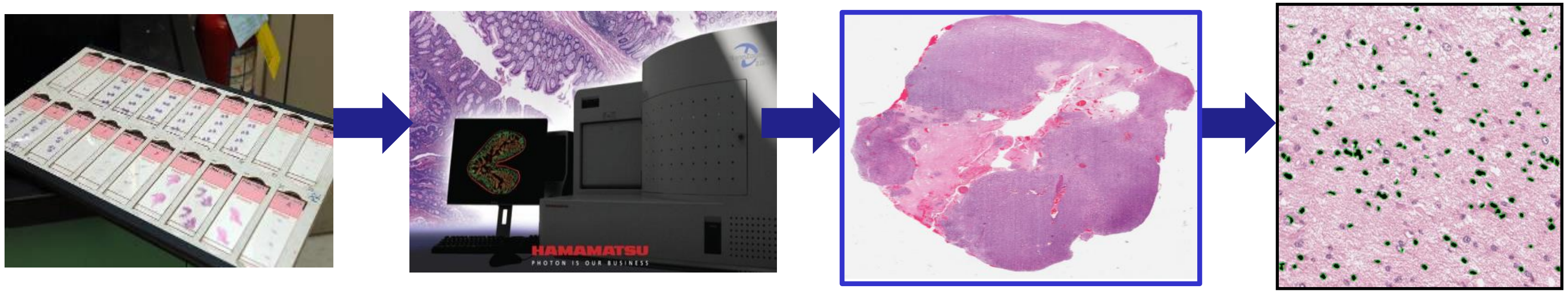
[VLDB13] [SIGSPATIAL13] Hadoop-GIS *Spatial Big Data Solutions*

[JPI12] [JPI13] PAIS



[VLDB12] [BIGDATA20] PixelBox

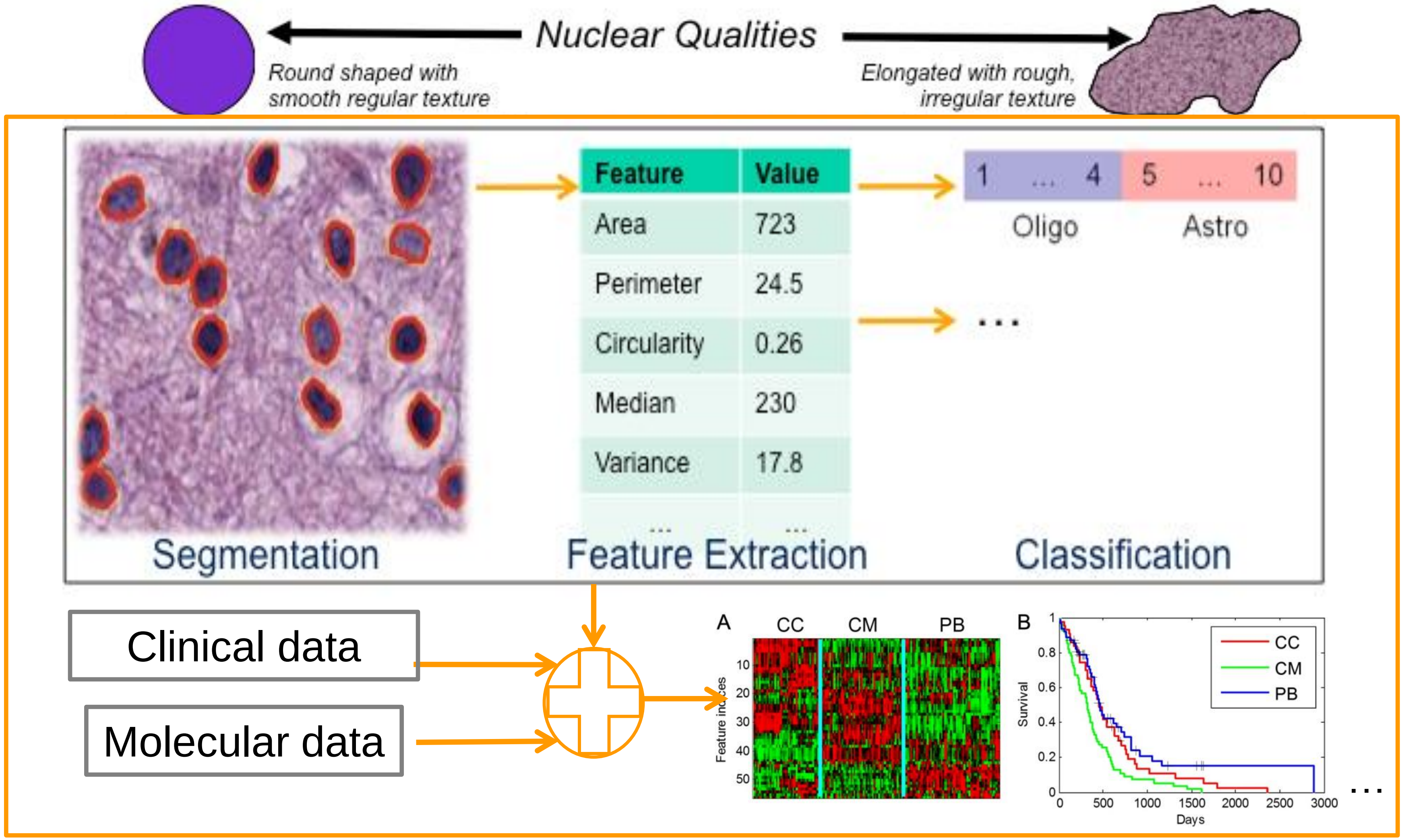
Use Case: Digital Pathology



Glass Slides | Scanning | Whole Slide Images | Image Analysis

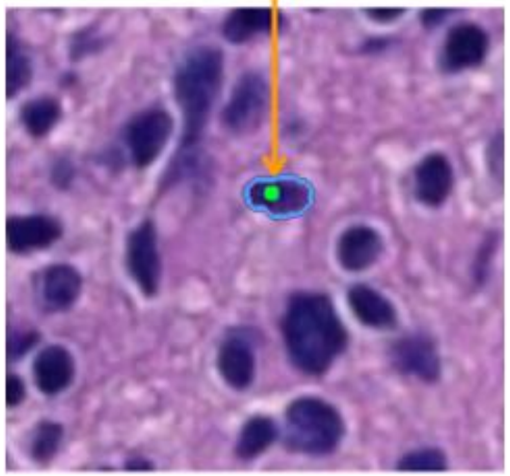
- Pathology images contain rich information to understand diseases and support diagnosis
- FDA approved review and interpretation of digital surgical pathology slides in 2017
- Analyzing pathology images can help to understand diseases and support diagnosis, e.g.:
  - Automated classification of diseases (cells or regions), CAD
  - Integrative translational research by integrating phenotypic imaging data, genetic signatures and clinical outcomes

Example: Distinguishing Characteristics in Glioblastoma

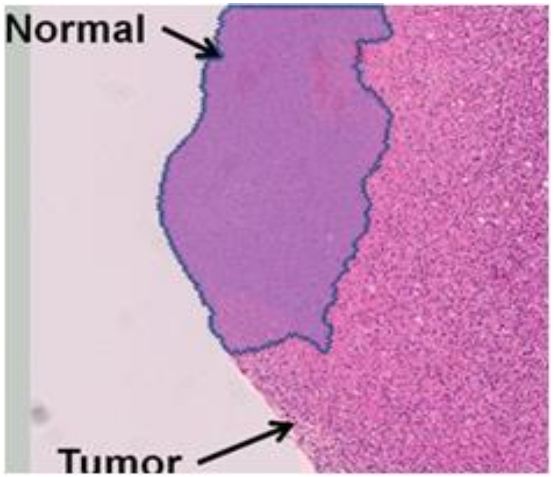


Spatial Queries in Pathology Imaging (2D)

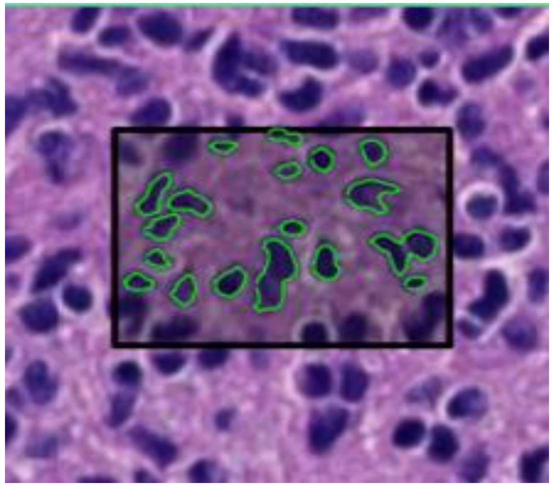
POINT



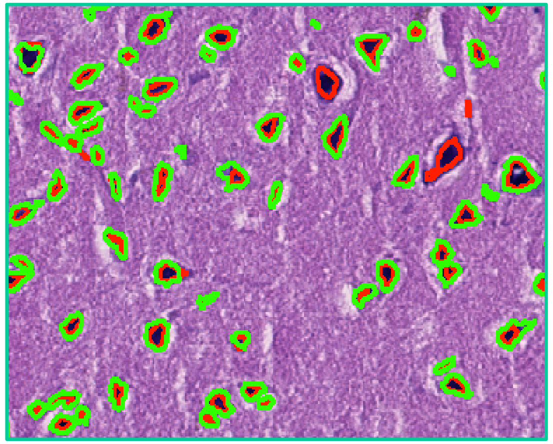
CONTAINMENT



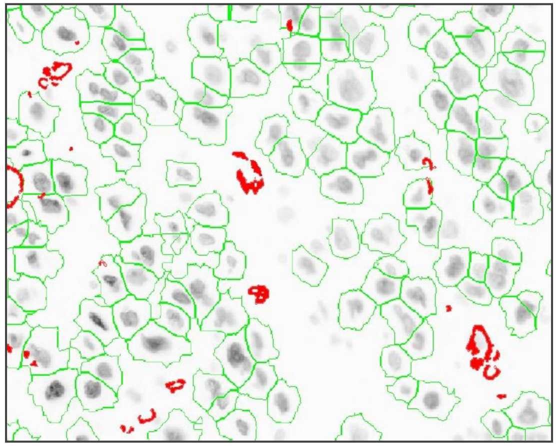
WINDOW



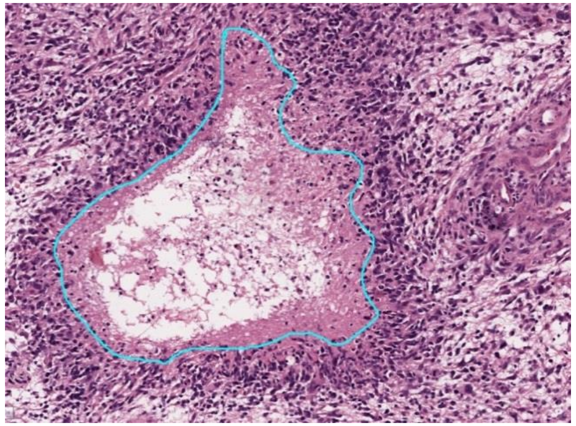
SPATIAL JOIN



NEAREST NEIGHBOR



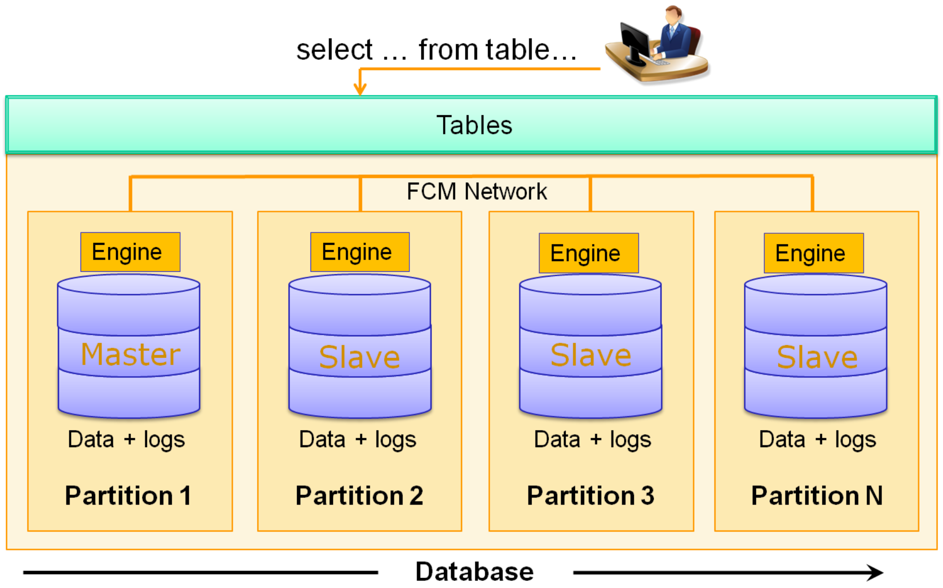
DENSITY



Spatial Data Management: Traditional Approach: Parallel SDBMS

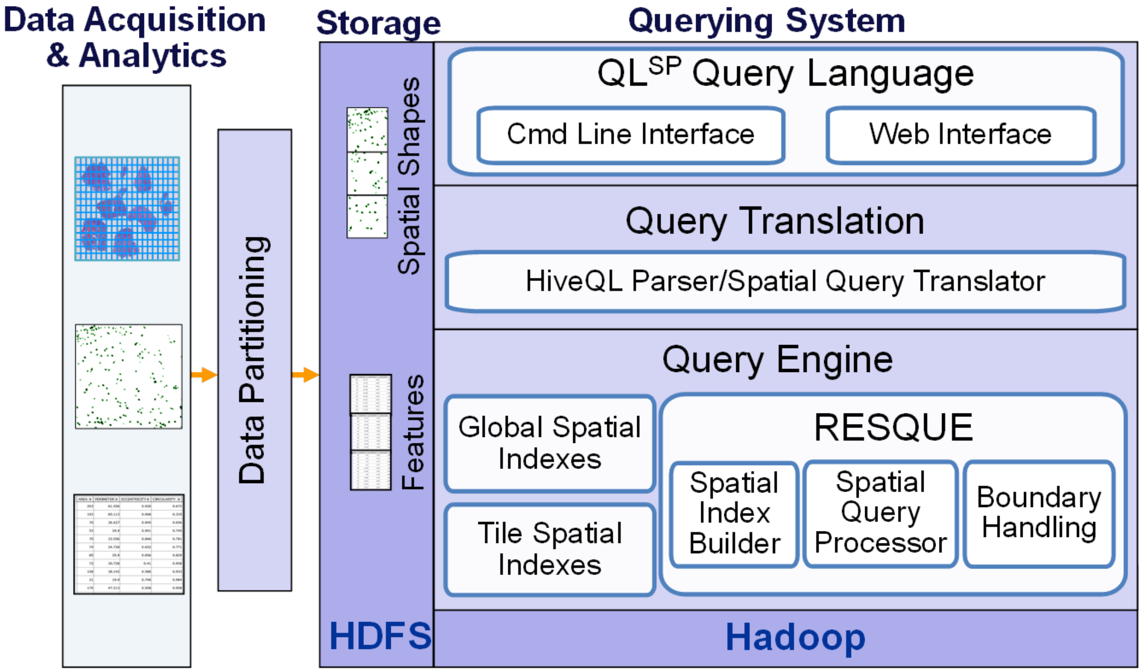
- Shared nothing architecture through partitioning to increase I/O bandwidth via parallel data access
- ORDBMS with spatial data types and access methods
- Comprehensive model and expressive query language
- Partitioning based scale out is possible but very difficult and expensive for many nodes
- Lack of support of complex 3D data types
- Data loading is a major bottleneck
- DB systems not optimized for computational intensive operations

[JPI12, JPI11]



MapReduce Based 2D Spatial Queries: Hadoop-GIS

- Hybrid query engine of MapReduce and database – on-demand query engine
- Data skew aware spatial data partitioning for parallelism
- On-demand indexing
- Declarative spatial queries (Hive^{SP}) and translation into MapReduce



Hadoop-GIS

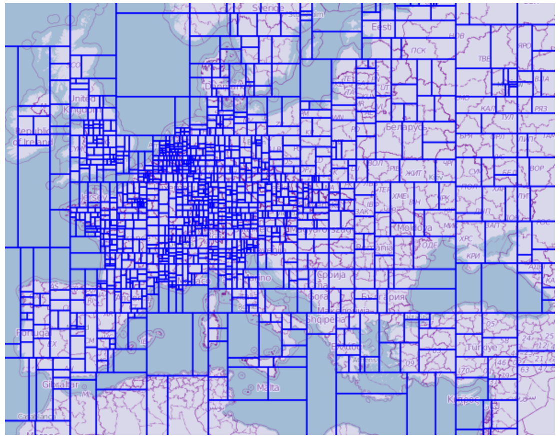
Spatial Big Data Solutions

Hadoop-GIS: A High Performance Spatial Data Warehousing System over MapReduce

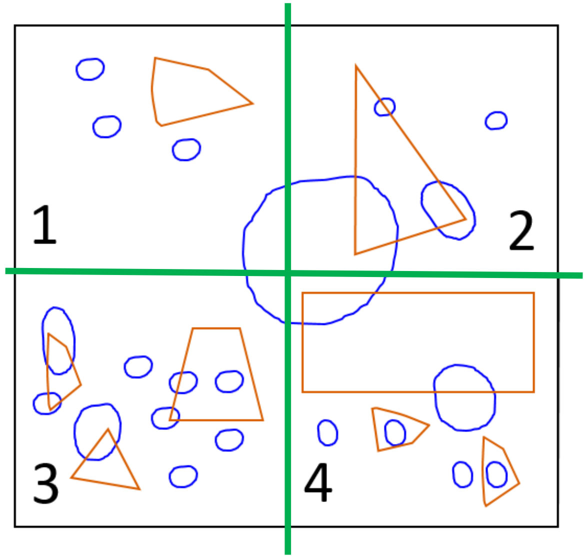
Most cited paper in VLDB 2013

[VLDB13, SIGSPATIAL14, SIGSPATIAL13, SIGSPATIAL12]

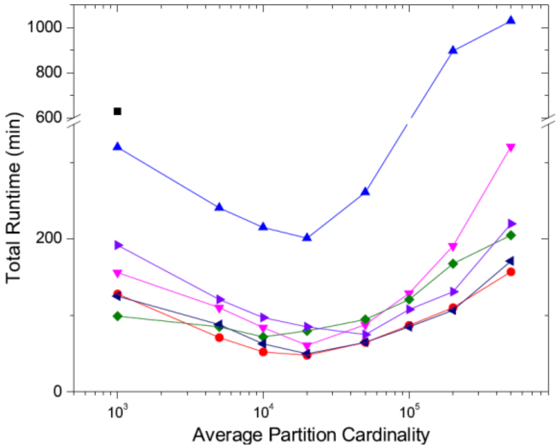
Data Parallelism: Spatial Data Partitioning



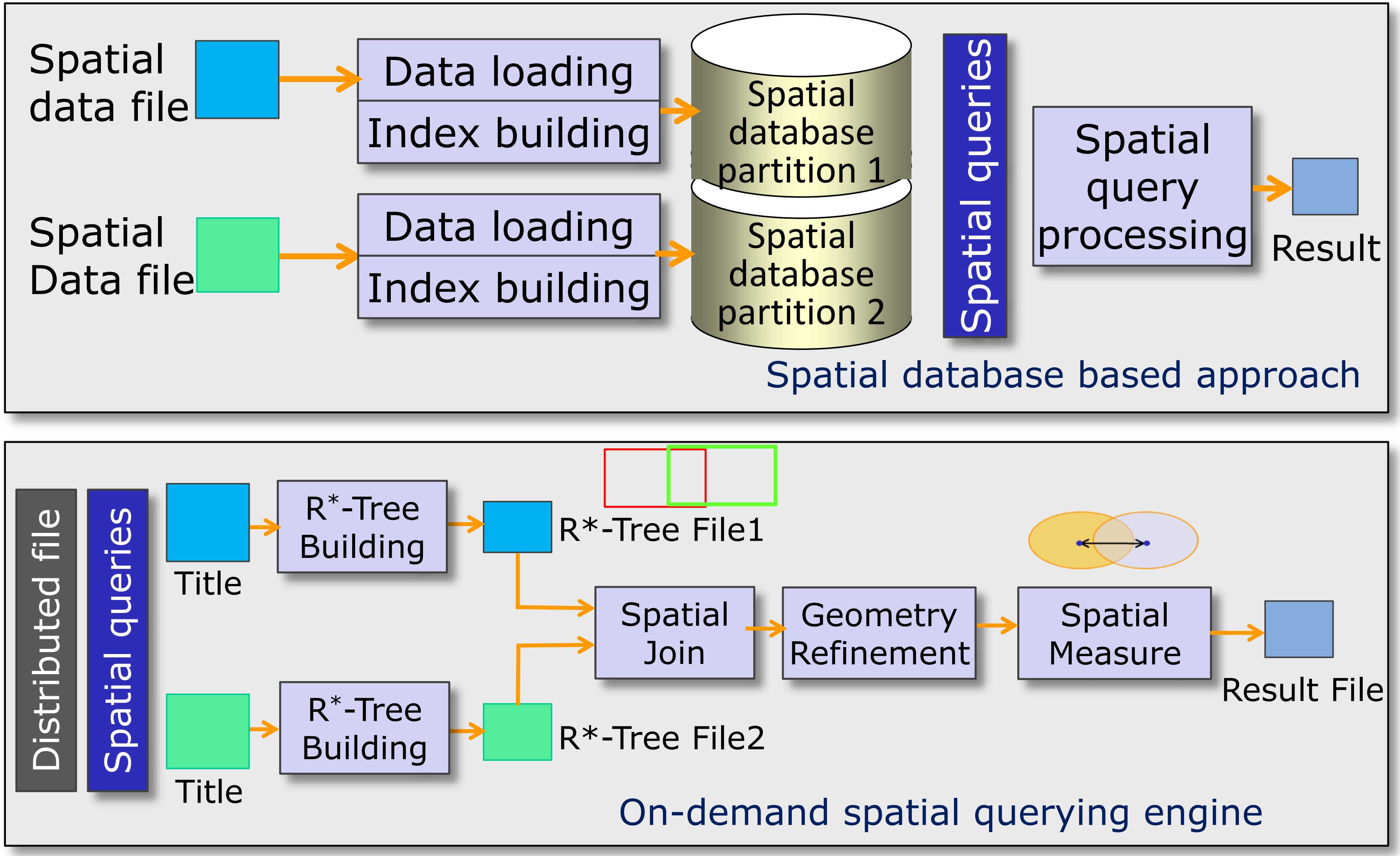
- Effective partitioning is critical for task parallelization and load balancing
  - Data skew
  - Criteria: balanced distribution, granularity, overlapping, impact of queries
- Multiple partitioning algorithms
- Query cost-model based approach for partitioning



$$
\begin{aligned}
C(\theta) &= C_{idx} + C_{flt} + C_{ref} + C_{cst} \\
&\approx \alpha log(\theta)(|R| + |S|) + \theta\gamma \frac{|R|}{\theta} log(\frac{|R|}{\theta}) \\
&+ \theta\omega(1+\beta)\frac{|S|}{\theta} log(\frac{|R|}{\theta}) + \delta(1+\theta\beta) \\
&+ \eta + \theta\zeta
\end{aligned}
$$



On-Demand Spatial Query Engine



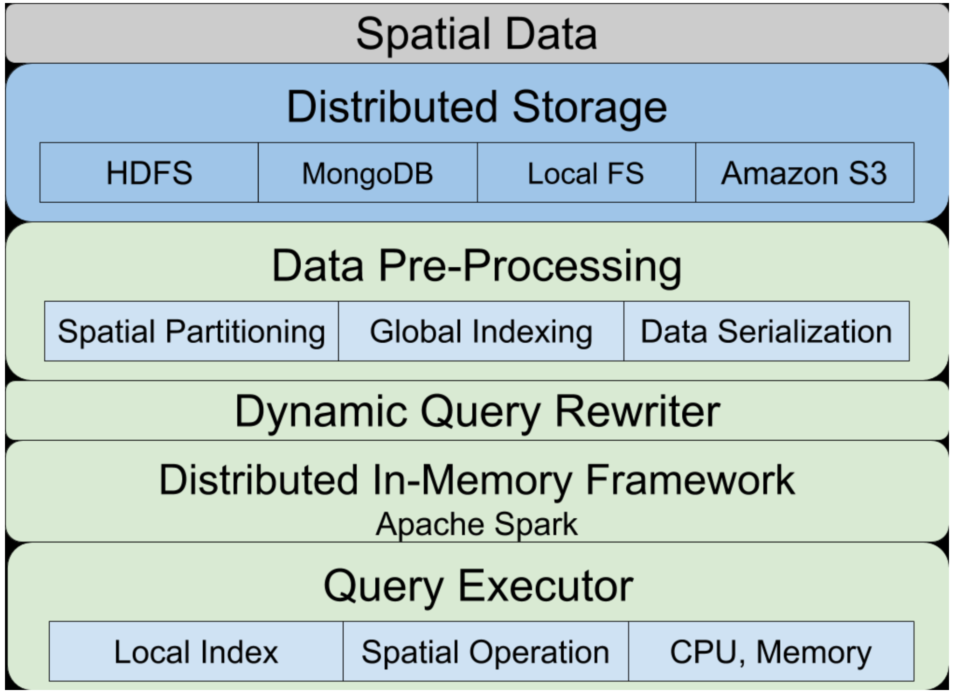
Example: 2-way spatial join

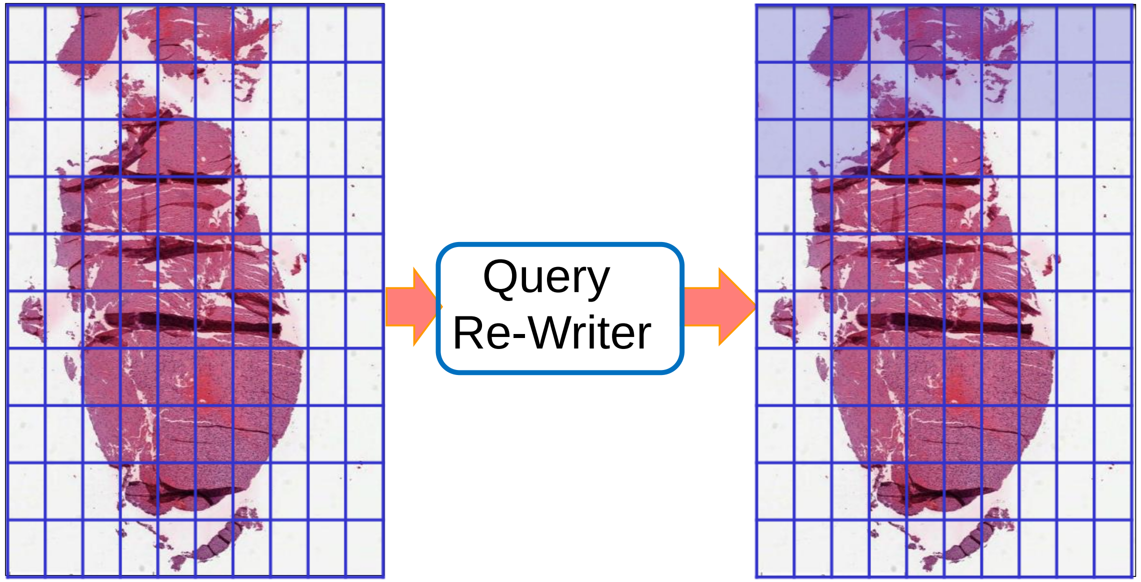
In-Memory Spatial Query Processing with Spark

- Hadoop comes with high I/O cost due to inter-job data movement
- Apache Spark - Industry standard for large scale in-memory processing
  - Minimizes IO by keeping data in memory – if possible
  - Maximizes performance by iteratively processing in-memory data
- Spatial data processing with Spark?
  - Spark requires extensive tuning to avoid "Out Of Memory" exceptions
  - Spark based 2D spatial querying systems often fail to complete jobs when the data are too big

SparkGIS: Resource Aware Efficient In-Memory Spatial Data Processing

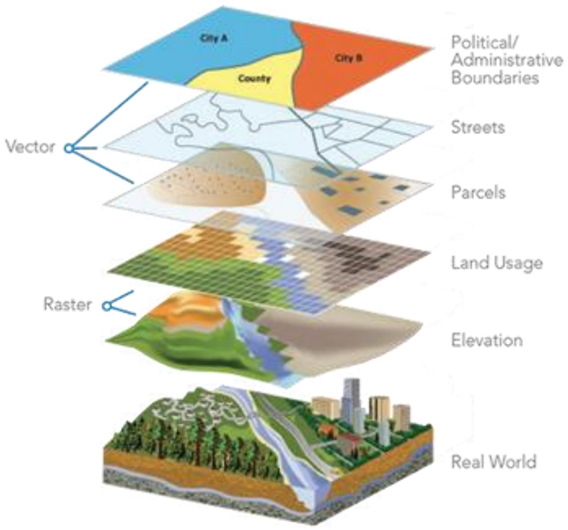
- Spark based spatial data management (SIGSPATIAL'17)
  - Take advantage of distributed memory to store and process spatial data
  - Resource aware query rewriting to break a large query into a pipeline of smaller queries based on partitions

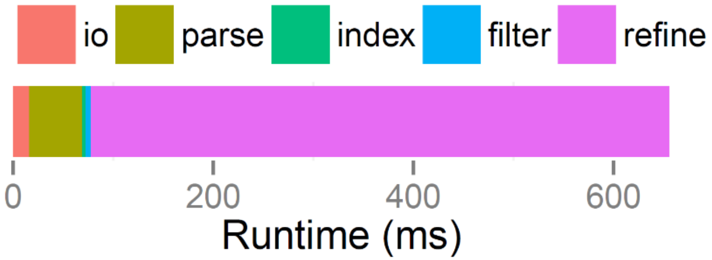




GPU Accelerated Spatial Queries

- Spatial cross-matching or spatial overlay is heavily computational driven
- GPU accelerated spatial cross matching
  - Massively parallel
  - Cheap and highly available
- Intra-Object Level Parallelism
  - Single Instruction Multiple Data (SIMD)

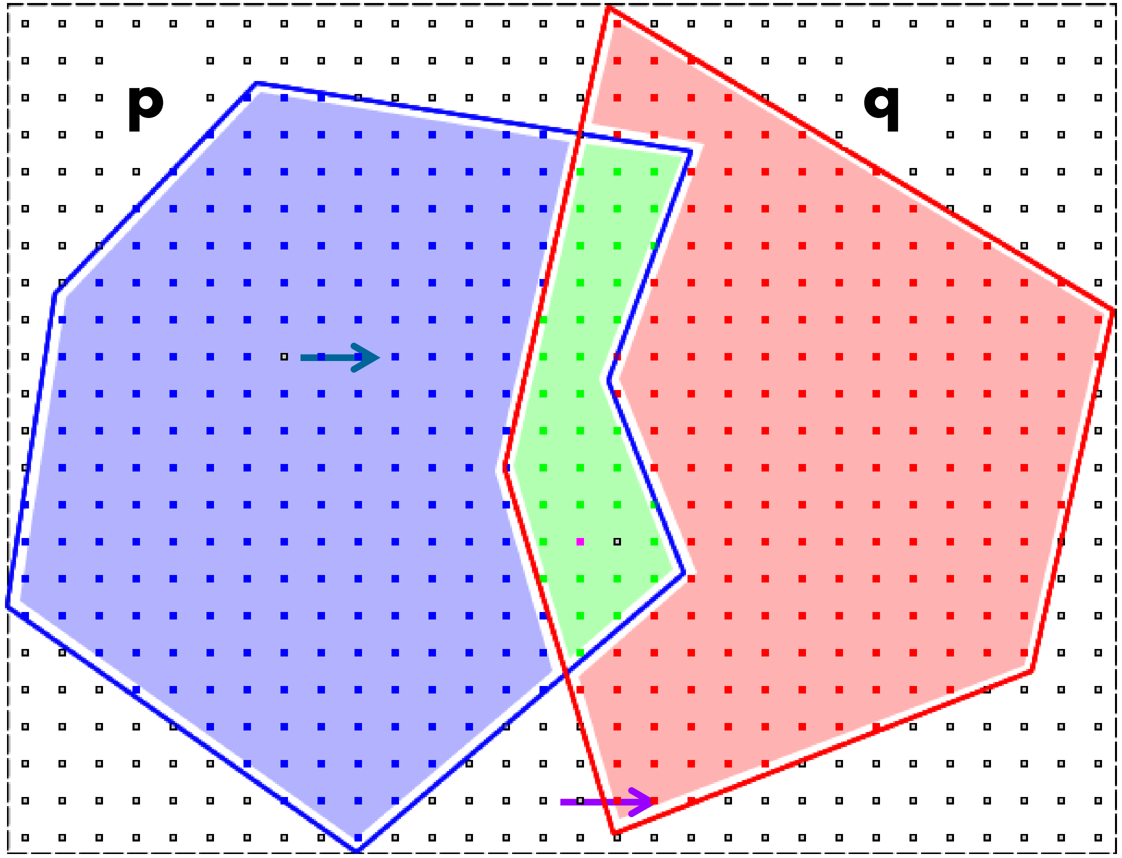




90+% time on computation

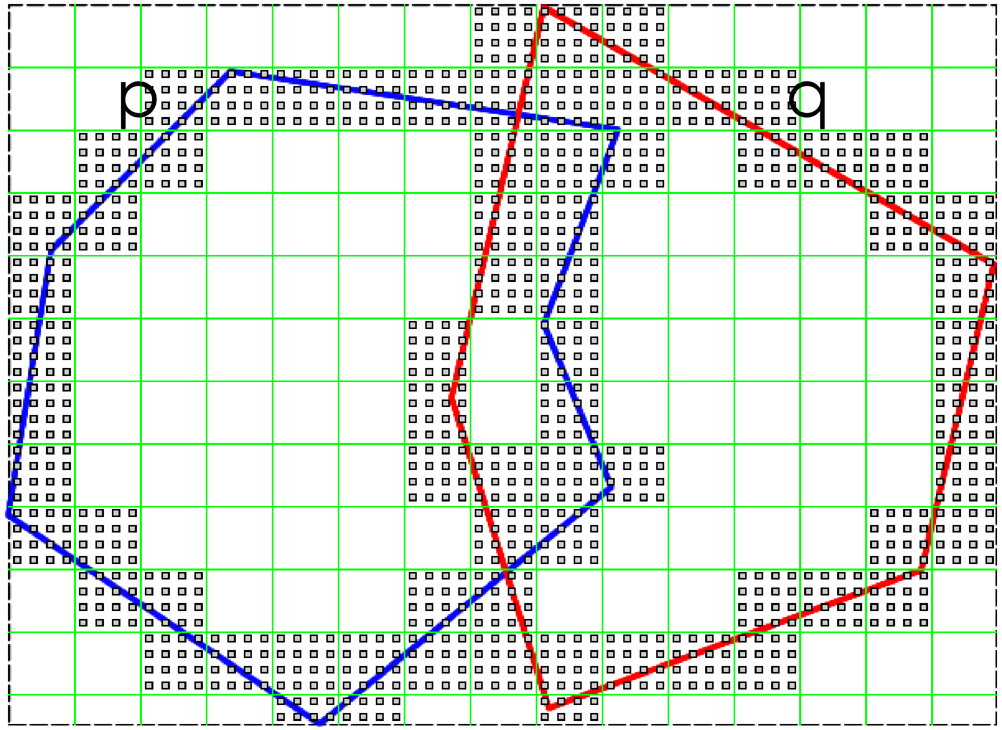
Exploit SIMD Data Parallelism: Monte-Carlo

- Monte-Carlo approach (a basic method)
- Perfect data parallelism, but high compute intensity when polygons are relatively large



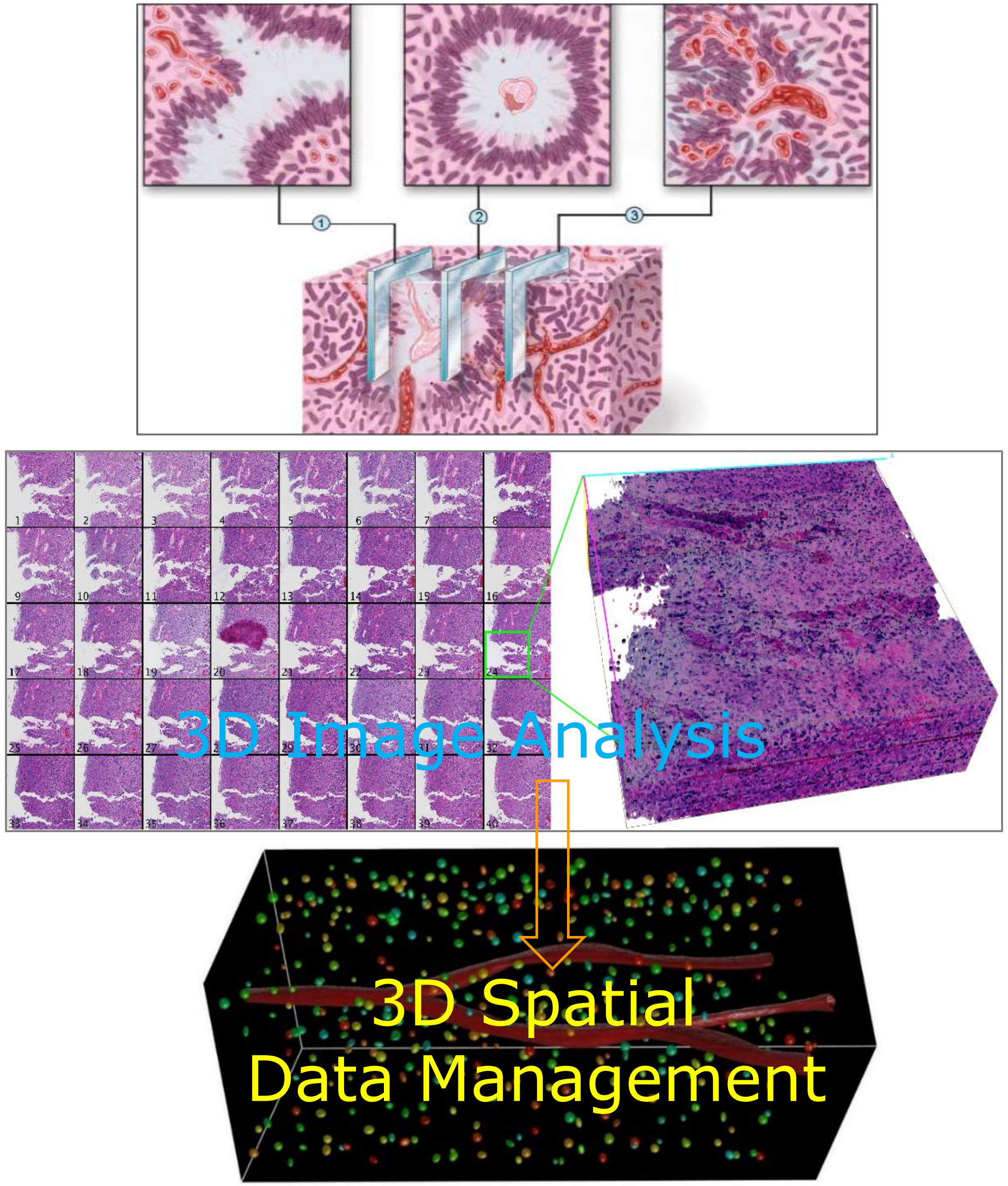
PixelBox: Combine Box and Pixel Testing

- First apply region scheme to finish testing of large regions
- Then apply per-pixel testing for small sub-boxes
- Preserve high data parallelism and low compute intensity
- Speed up: on a single GPU (512 cores): 120X



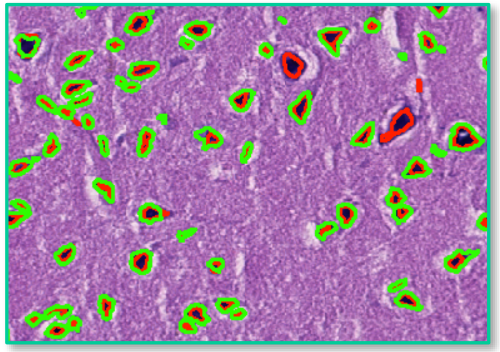
3D Spatial Big Data: 3D Digital Pathology

- 2D views highly depend on the locations and angles of the cutting planes
- 3D reconstruction will preserve much accurate spatial architecture
- Convert pixel/voxel information to 3D micro-anatomic objects
  - Explosion of 3D data and complex objects
  - High I/O and communication cos for data processing
- Hadoop/Spark: distributed file systems for data storage, data shuffling on I/O - a major bottleneck

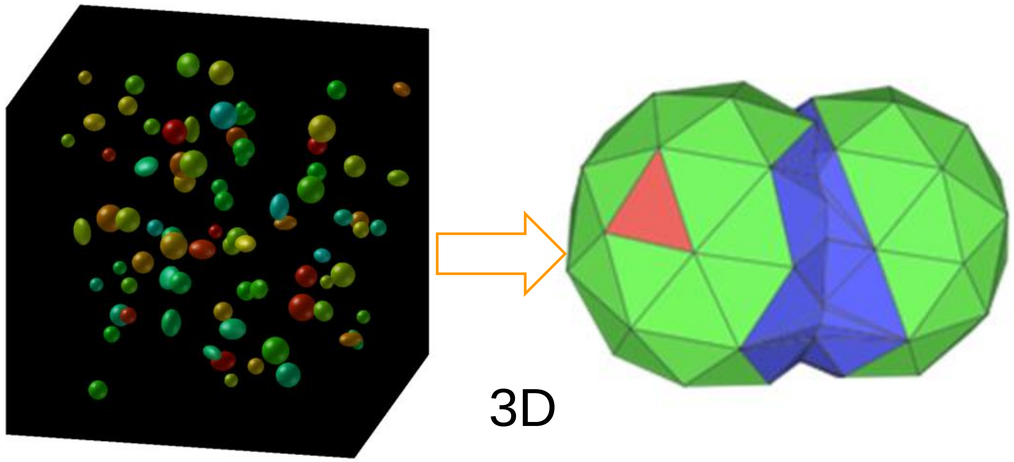


Example Query: Spatial Join/Cross-Matching

- 3D spatial join: compare two sets of spatial objects
- High quality image analysis algorithms are essential to support biomedical research and diagnosis
  - Validate algorithms with human annotations
  - Compare and consolidate different algorithm results
- e.g.: what are the distances and overlap ratios between the 3D objects from two algorithms?



2D

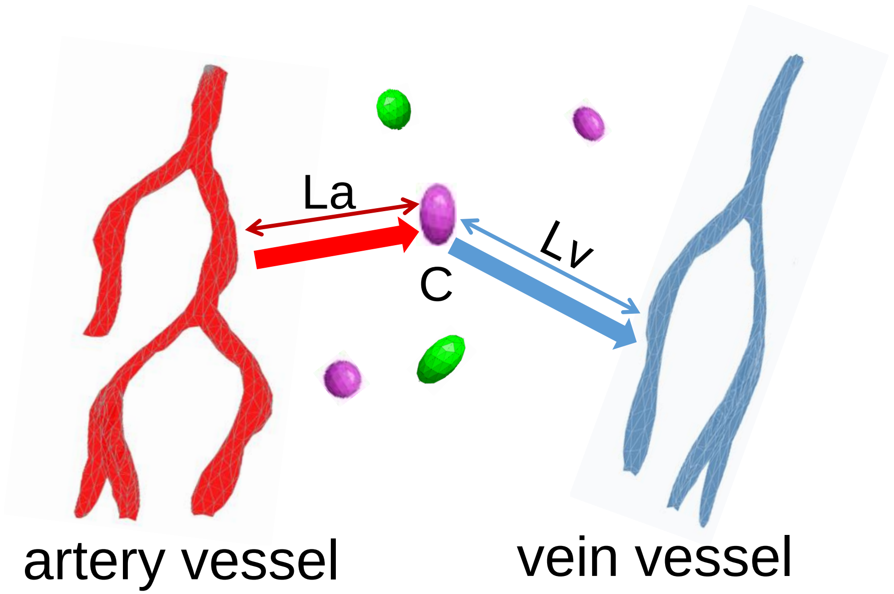


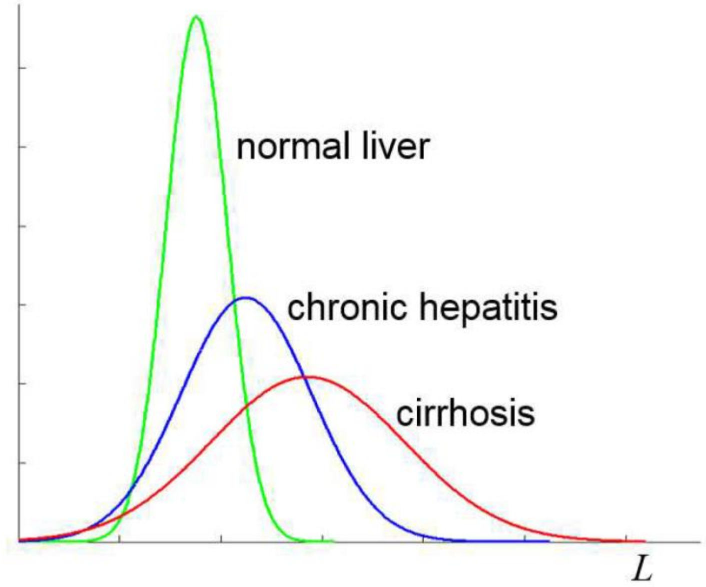
3D

- Containment query is a special case

Example Query: 3D Spatial Proximity Estimation

- Spatial proximity estimation aims to explore inter-objects distribution in 3D space based on distances between neighboring objects
- e.g., for each cell in liver tissue, find the shortest path to its neighboring artery vessel and the shortest path to its neighboring vein vessel, and compute the average and standard deviation (dispersion) of the full path



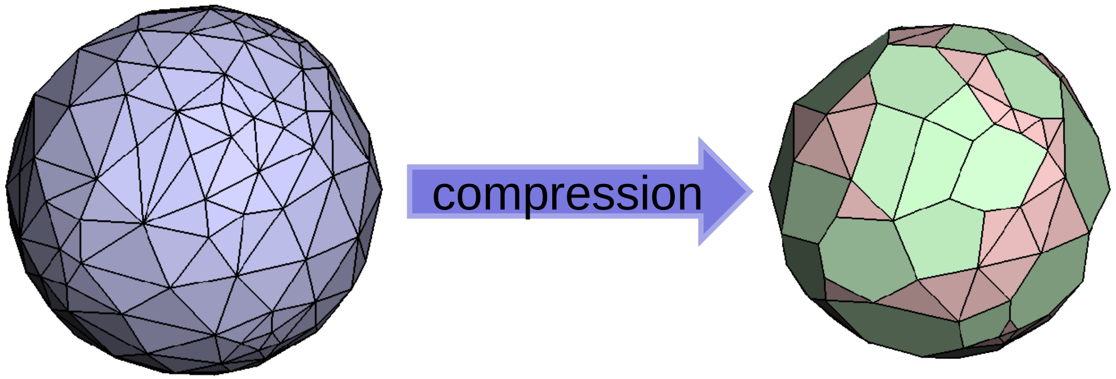


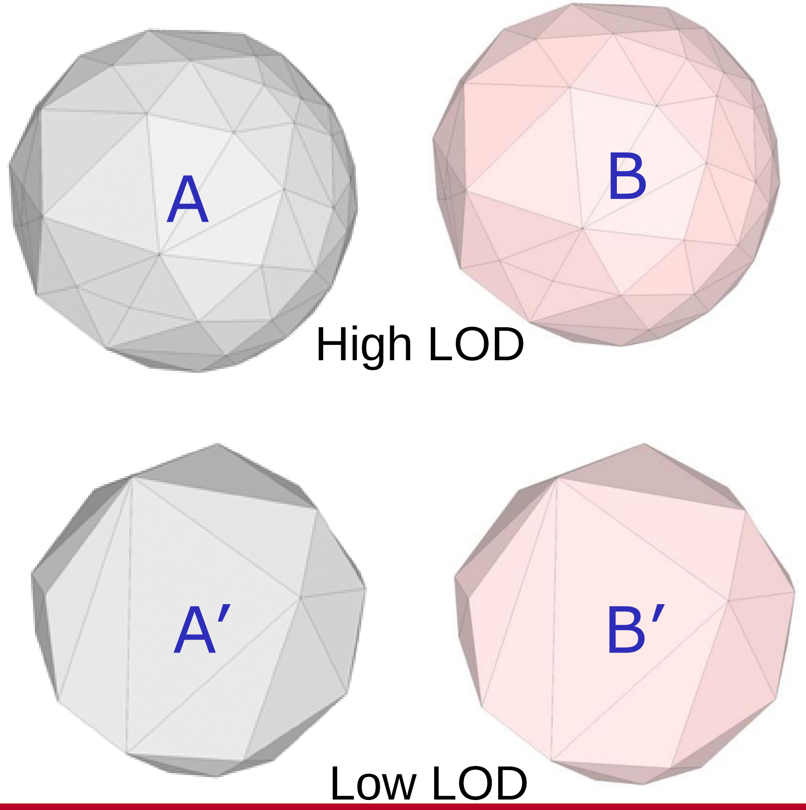
Our Goal: a Highly Efficient and Scalable 3D Querying System: iSPEED

- iSPEED (**i**n-memory **sp**atial query system for thr**ee** **d**imensional spatial data)
- Effective progressive compression for individual 3D objects – reducing data size
- In-memory based data storage and indexing – reducing I/O and communication
- Multi-level spatial indexing – minimizing search space
- On-demand structural indexing – tailored for complex objects
- In-memory based 3D spatial query pipelines highly scalable on Hadoop/Spark – achieving high scalability

3D Compression

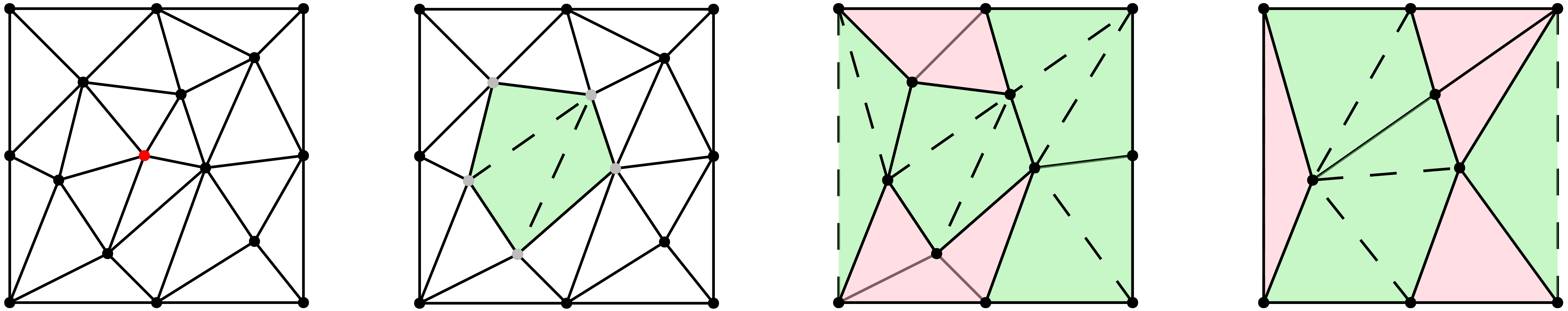
- A polyhedron can be compressed to represent the original object with lower resolution
- Querying objects with at low-resolution representation can significantly advance the query efficiency – balance between accuracy and speed





3D Compression

The compression can be conducted by removing surface elements

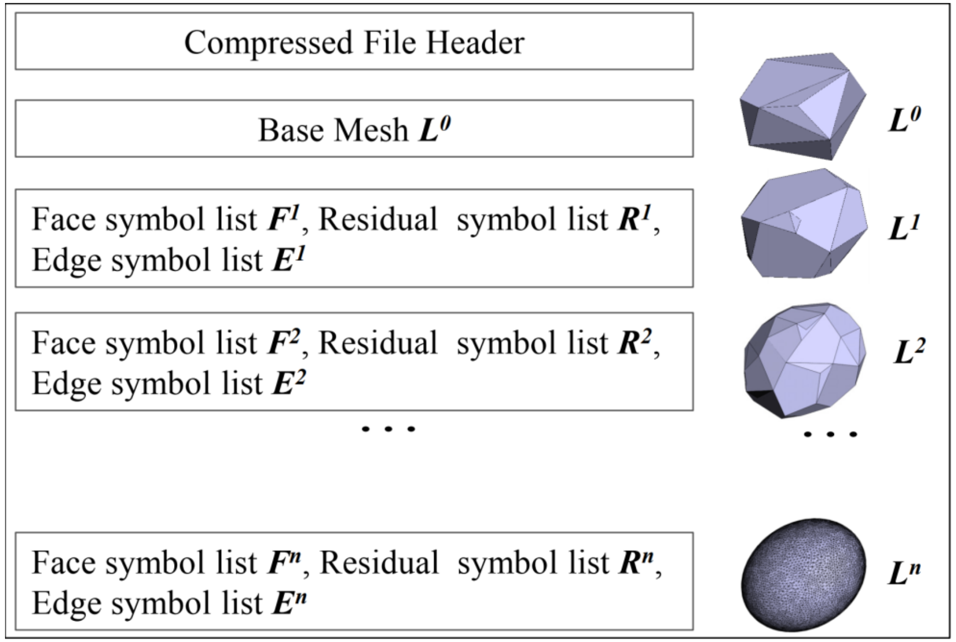


Original mesh — One vertex removed — After one round of decimation — After two rounds of decimation

Maglo, Adrien, et al. "Progressive compression of manifold polygon meshes." Computers & Graphics 36.5 (2012): 349-359.

Progressive 3D Compression

- Multiple LOD polyhedrons can be stored in a single compressed format
- The representation at a specific LOD can be retrieved directly without decompressing all the LODs



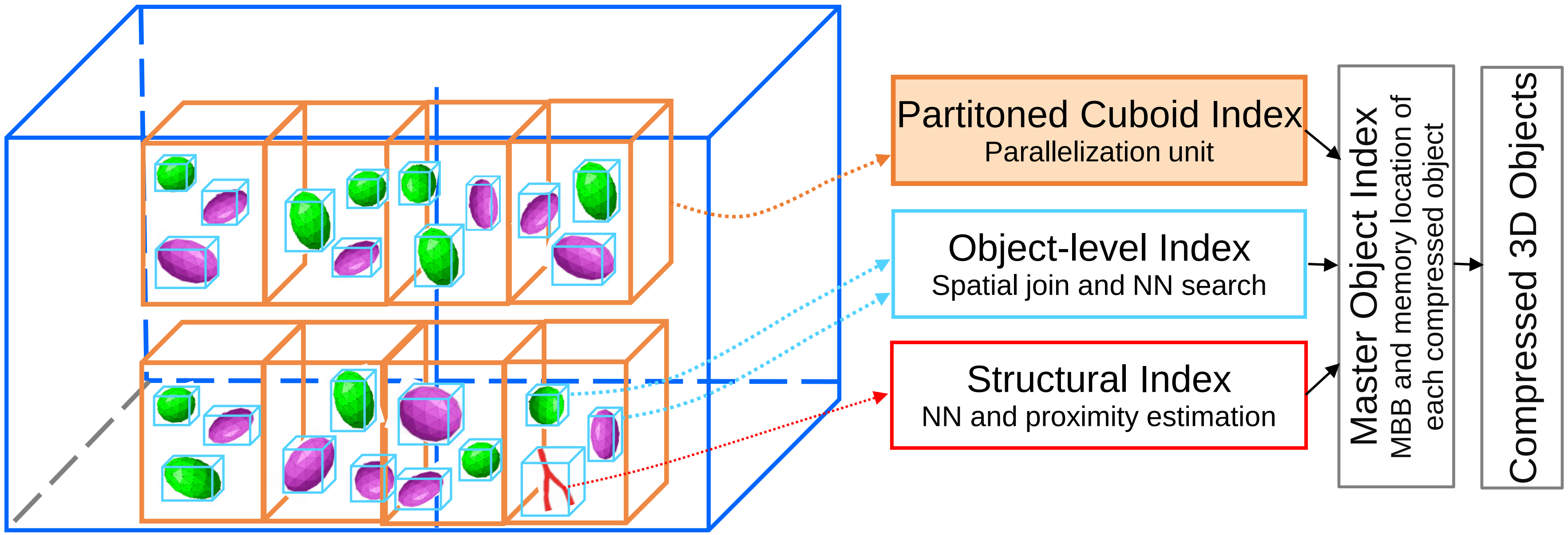
Liang, Yanhui, et al. "ispeed: an efficient in-memory based spatial query system for large-scale 3d data with complex structures." SIGSPATIAL 2017.

3D Data Compression (cont'd)

- Compression effectiveness
  - Compressed size: base mesh(LOD 0): 1%; all LODs: 3%
  - Computation complexity: linear with the # of vertices
- Compressed file stored in memory and replicated across all nodes
- Error from compression
  - Image analysis itself comes with errors
  - Spatial join query: an error of 0.54% with 90% LOD; 2.72% for 70% LOD
  - In practice, such precision loss is negligible for large data
- Users have an option to trade off between accuracy and speed

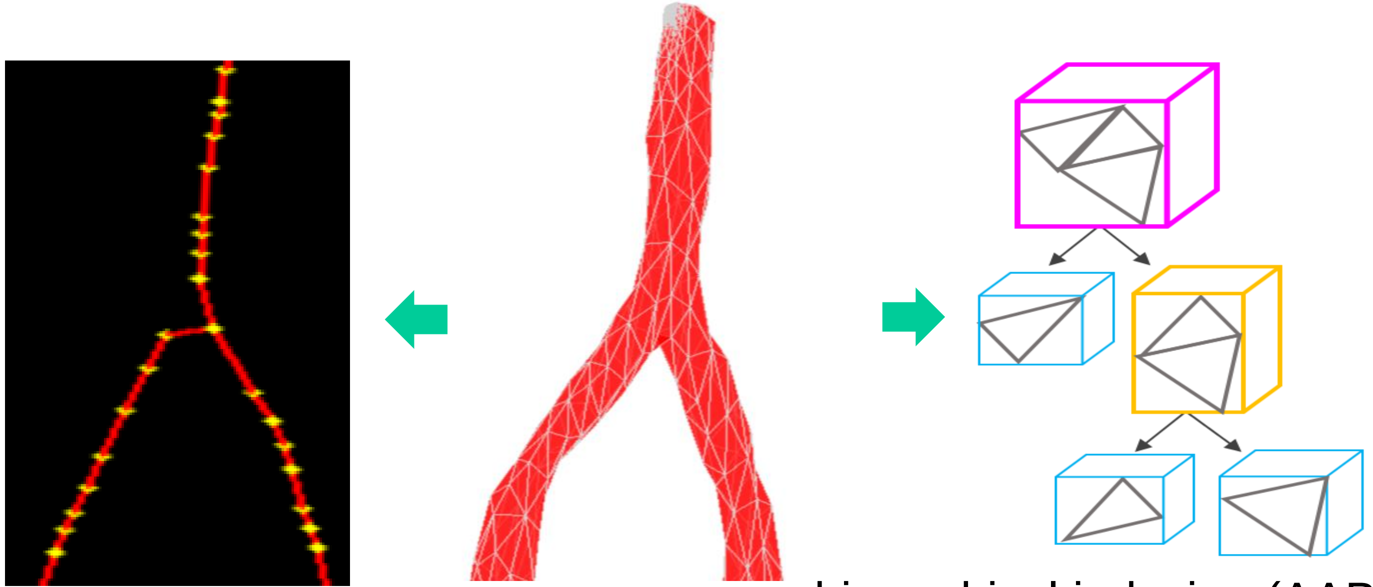
Multi-level Spatial Indexing: Global Indexing

- Multi-level indexing
  - Global space level: cuboid based partitions
  - Inter-object level: R-Tree (or R*-Tree) based spatial indexing of objects within a partition
  - Intra-object level indexing (structural indexing)



Structural Indexing for Complex Objects

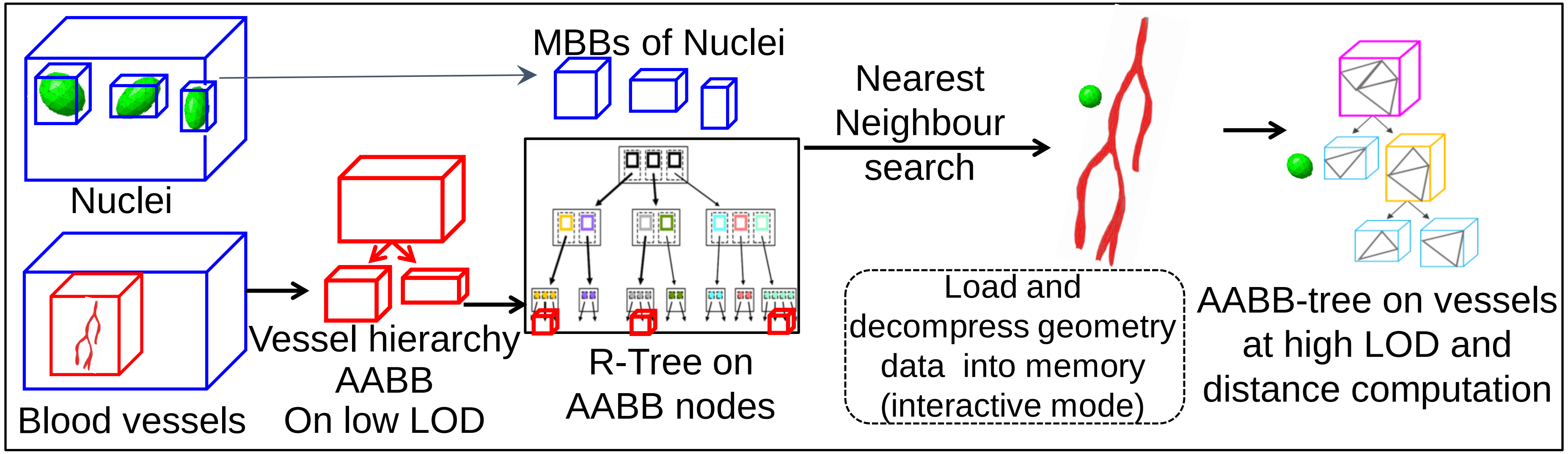
- Complex objects(e.g., vessels) can not be approximated as points or MBBs for distance based queries
- Topological skeleton based: effective shape abstraction to capture the essential topology of complex structures
- Hierarchical tree based: binary tree (axis-aligned bounding box or AABB tree) based hierarchical representation of the MBBs which traverse from the overall object to its subobjects



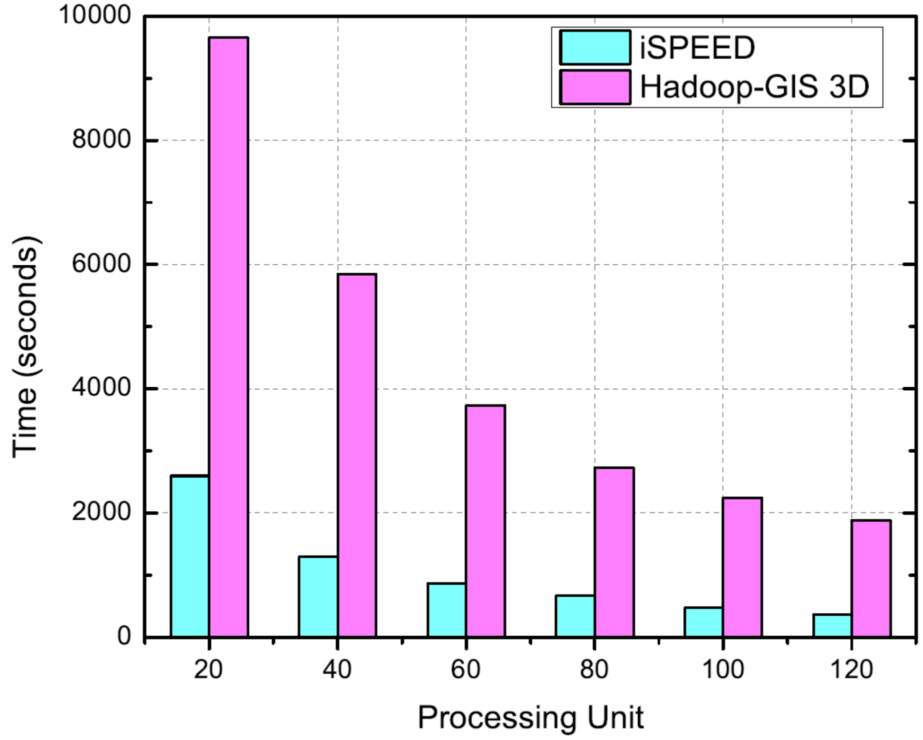
Skeleton based indexing　original vessel　hierarchical indexing (AABB tree)

Proximity Estimation

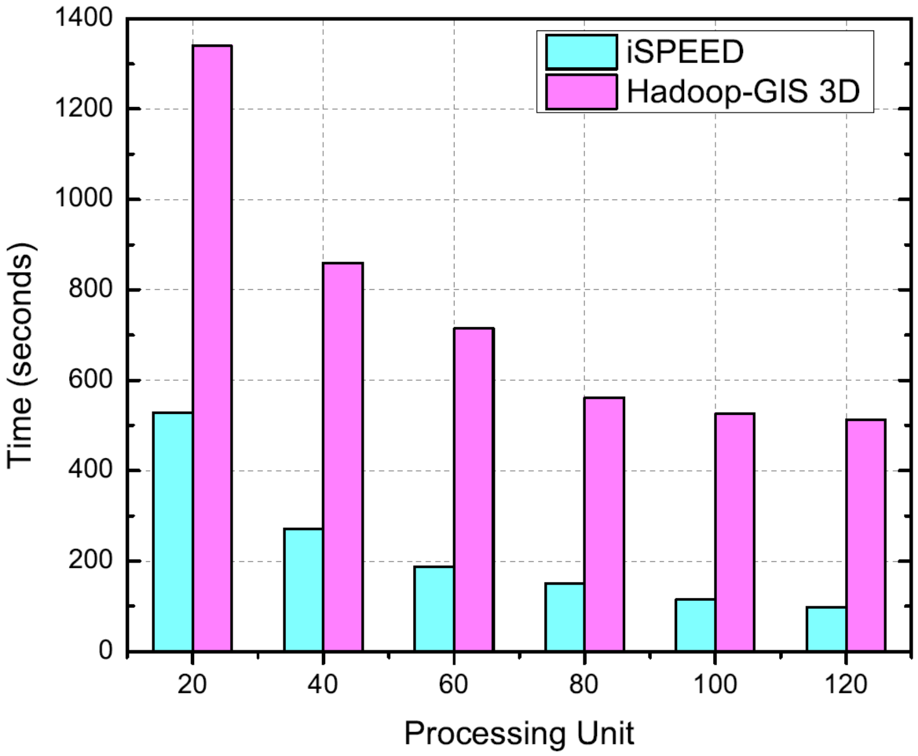
- Three-step proximity estimation query pipeline using structural indexing and R*-tree indexing
  - Use structural index (AABB) of vessels in low LOD
  - Find nearest neighbor using R*-Tree to index MBBs from above structural index
  - Perform accurate calculation of distances using structural index of vessels at high LOD (very few)
- Only small number of vessels with high LOD will be retrieved



Performance Comparison between iSPEED and Hadoop-GIS 3D

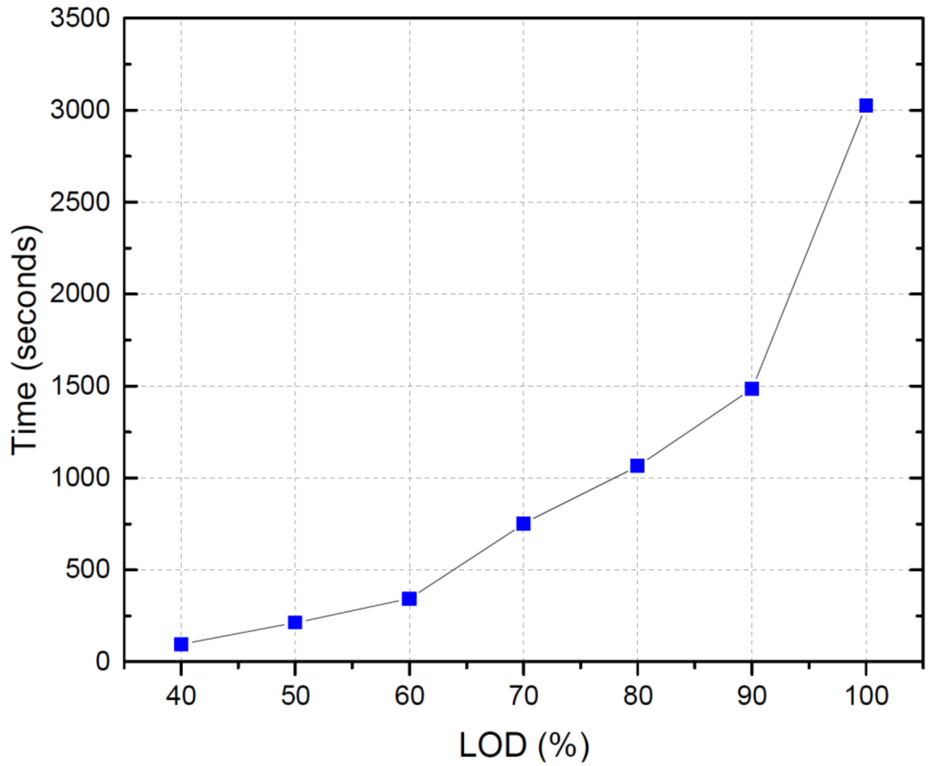


Spatial join

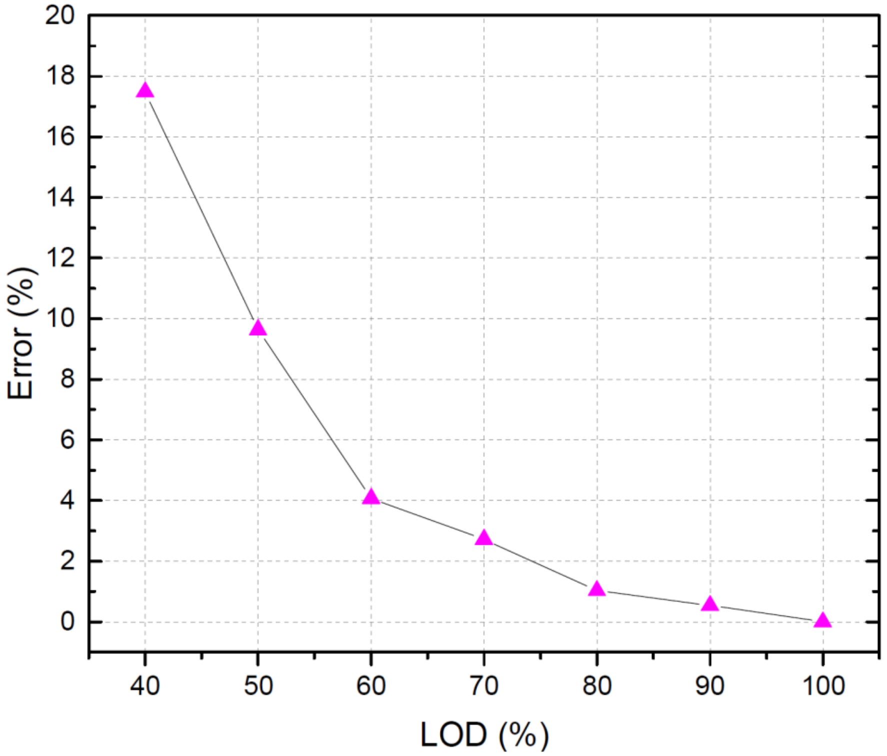


Spatial proximity approximation

Performance and Accuracy vs Level of Detail (Spatial Join)

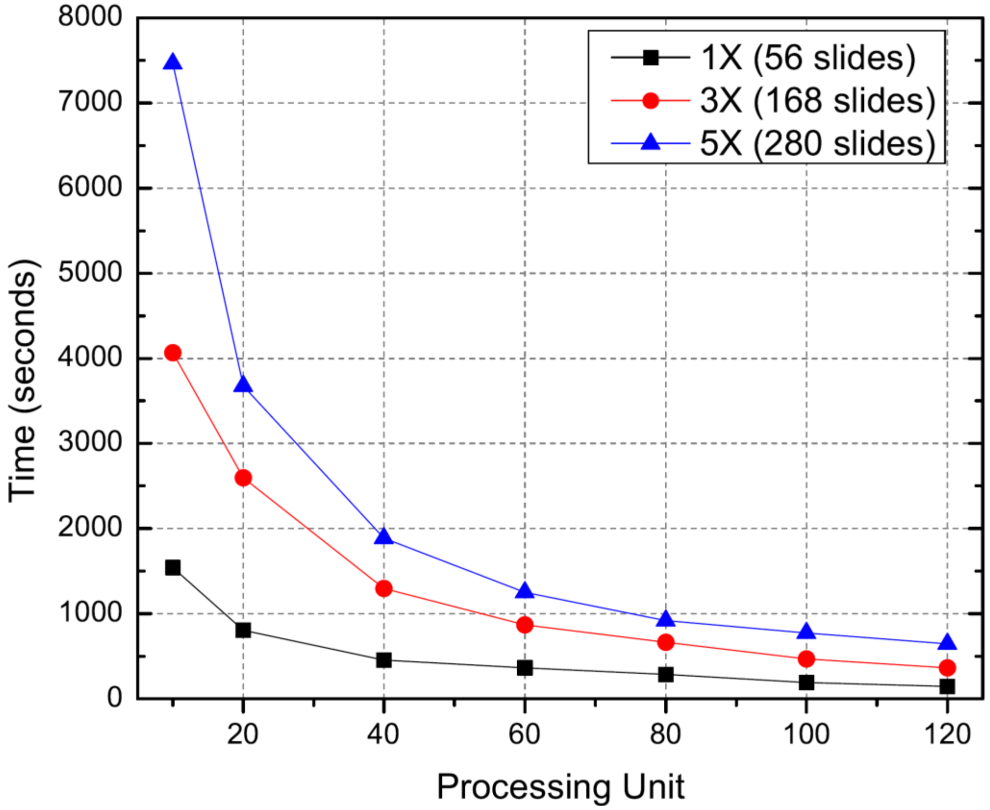


Execution time vs LOD

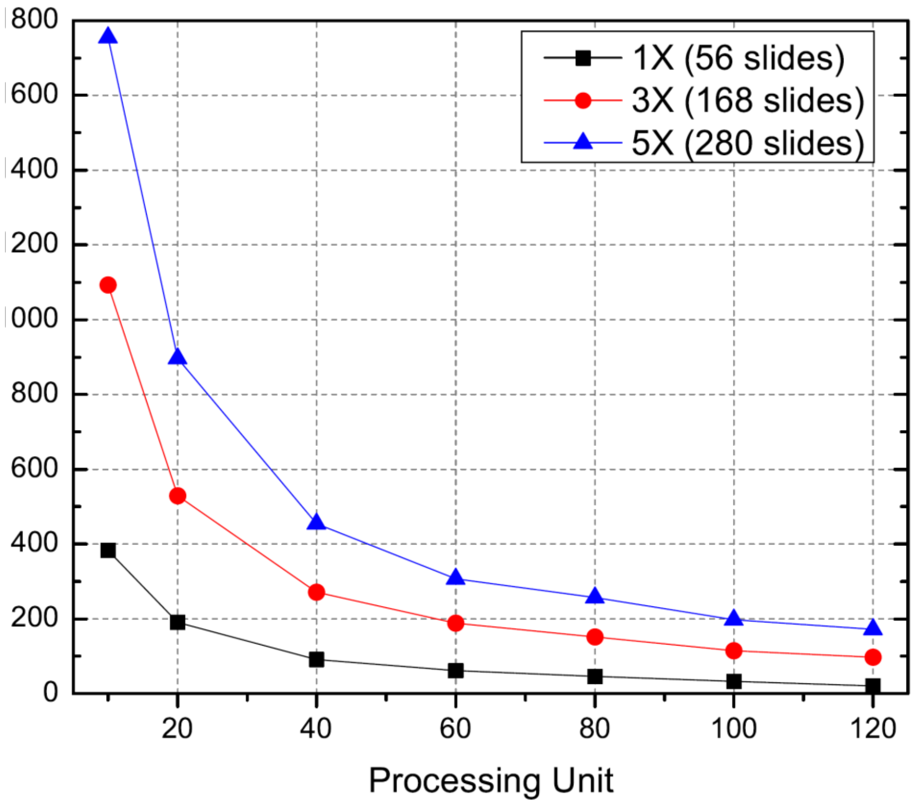


Error vs LOD

Scalability of iSPEED



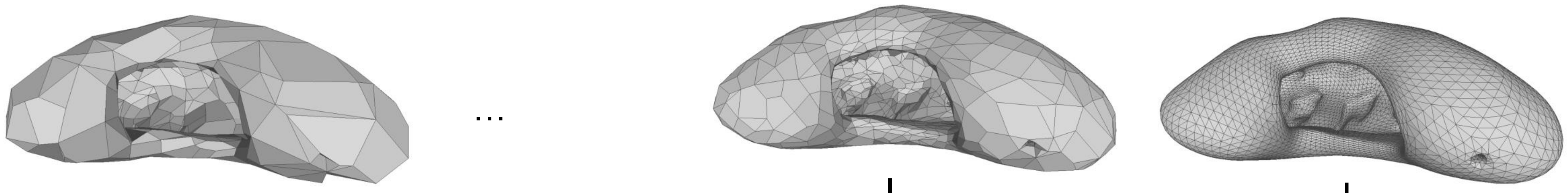
Spatial join



Proximity approximation

3D Compression Revisited

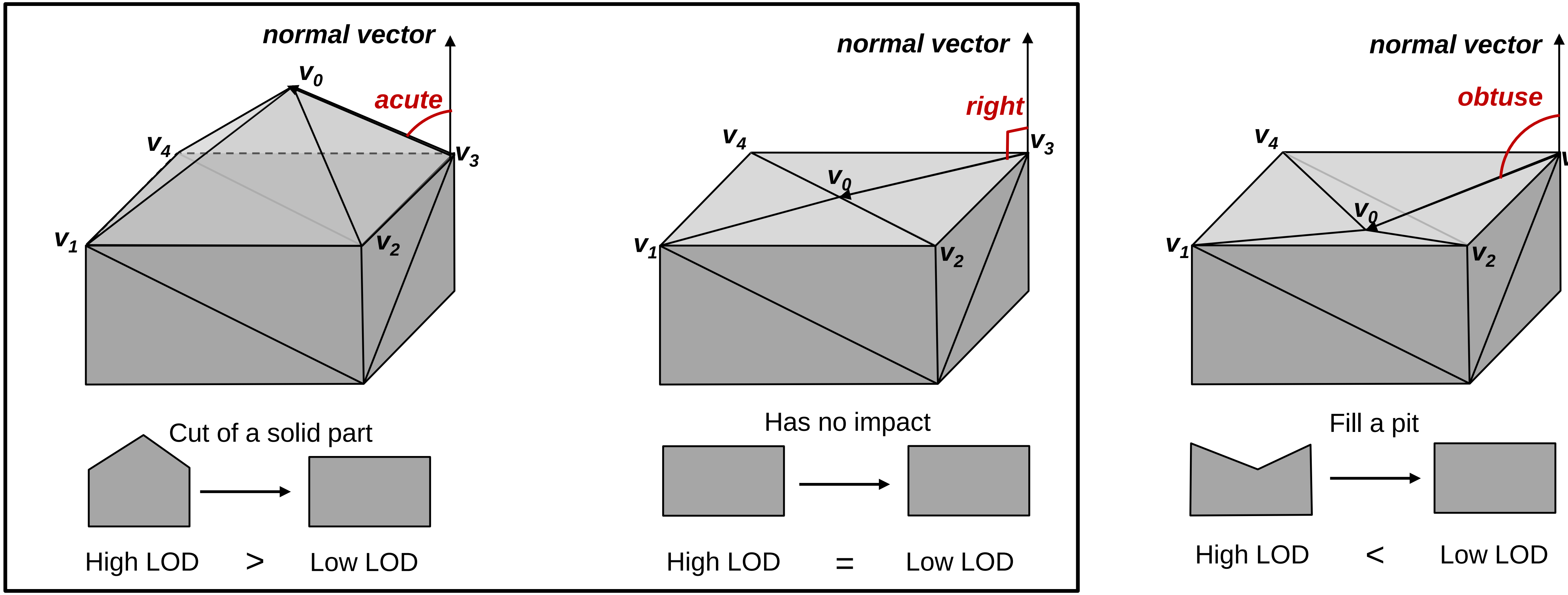
- A polyhedron can be compressed to represent the original object with lower precision
- The compression can be conducted by removing surface elements



L_0 … L_{n-1} L_n

…

3D Compression Revisited: Impacts of Removing a Vertex



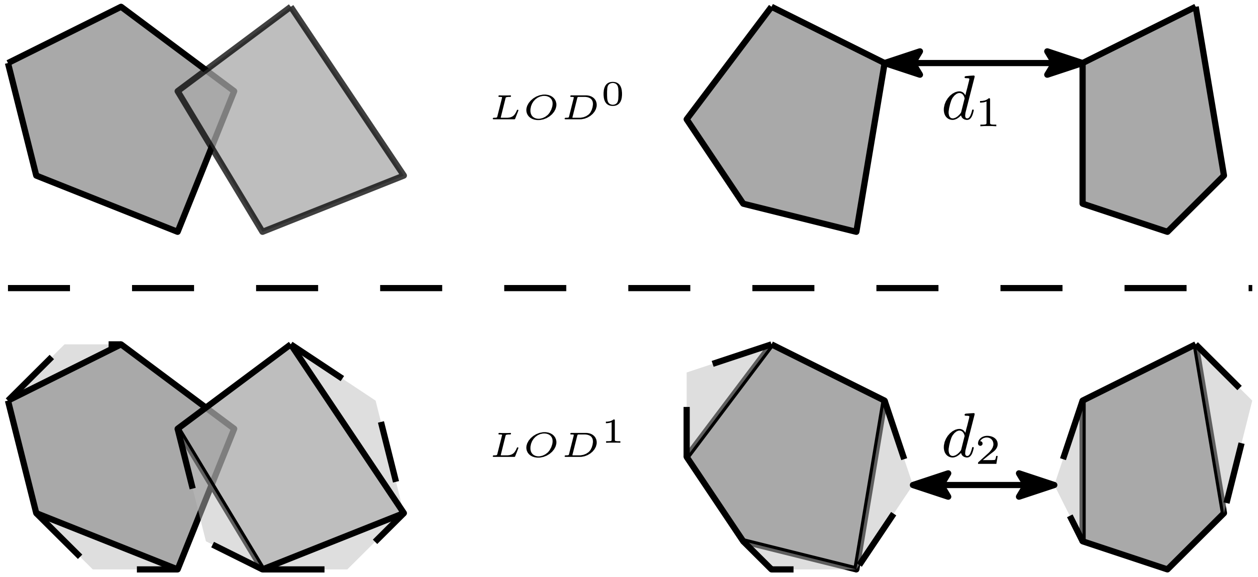
Protruding vertex

Recessing vertex

Progressive Protruding-Vertex Pruning Compression (PPVP)

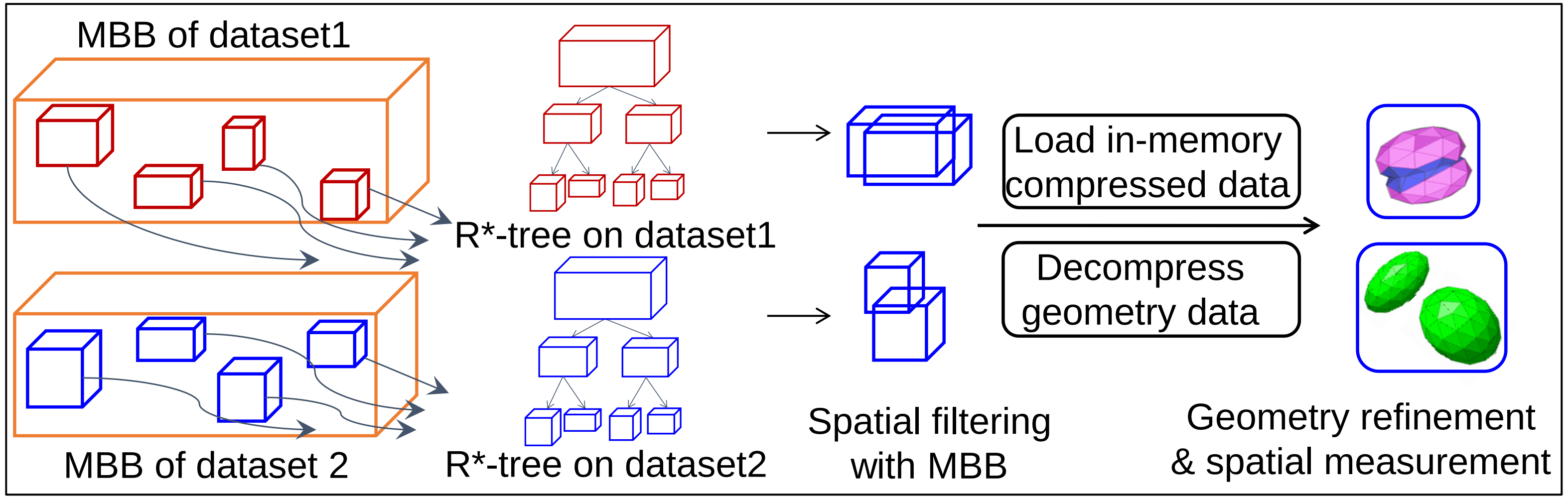
- We enforce that only protruding vertices can be removed: Progressive Protruding-Vertex Pruning Compression
- If a lower LOD of two objects intersect:

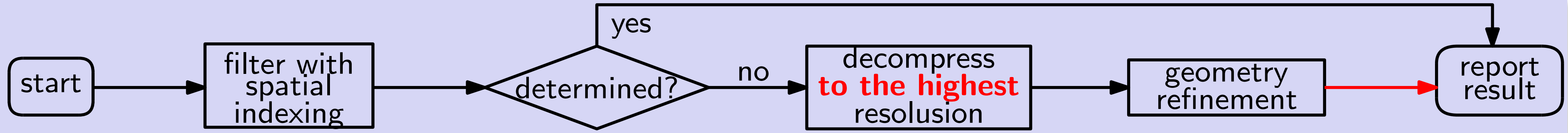
  → the higher LOD will intersect
- The distance between lower LOD is always **>** the distance at higher LOD



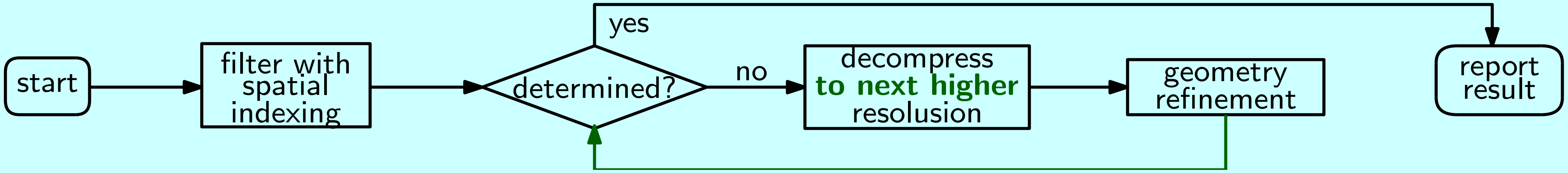
(a) Intersection (b) Distance

Filter-Progressive-Refine Spatial Querying Paradigm





Filter-Refine Paradigm



Filter-Progressive-Refine Paradigm

3DPro: Progressive 3D Spatial Data Representation and Queries

Stony Brook University

- Progressive Protruding-Vertex Pruning Compression

  → reduce storage and I/O

- Shape aware indexing with partitioning for complex geometries like vessels

  → further address shape complexity

- Filter-progressive-refine spatial querying: early return of query results at lower LOD

  → minimize computation complexity

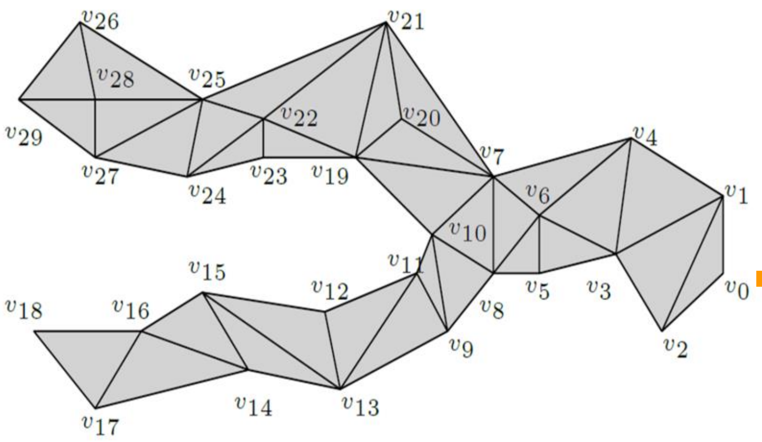
- Parallelized processing at both object level and intra-object level

  → high throughput

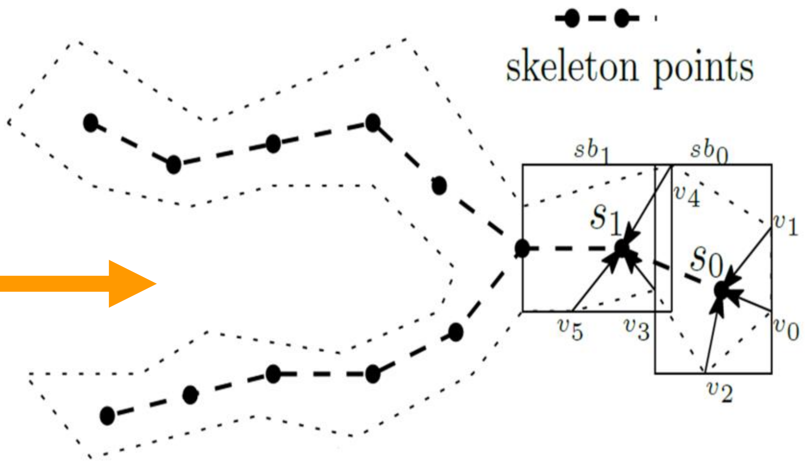
- In-memory based data management

  → minimize I/O

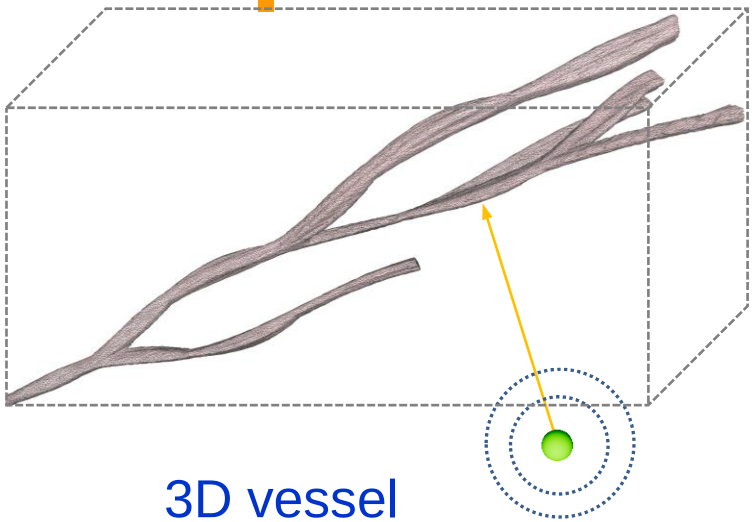
Shape Aware Indexing with Partitioning



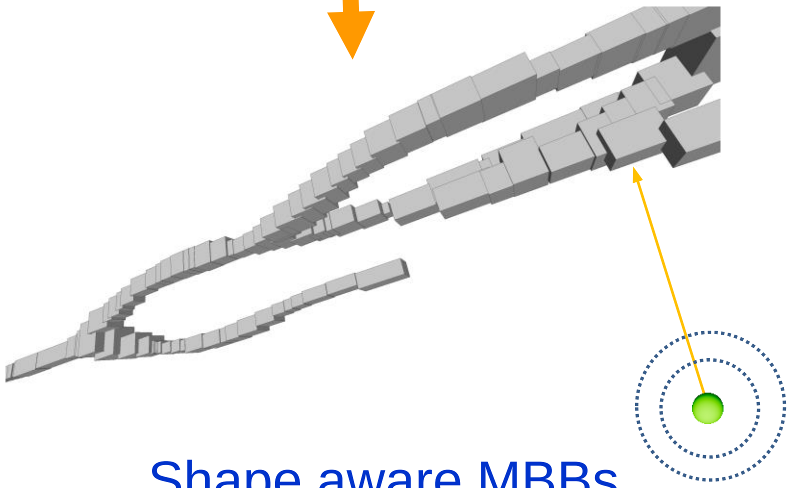
3D vessel meshes



Skeleton of vessels



3D vessel

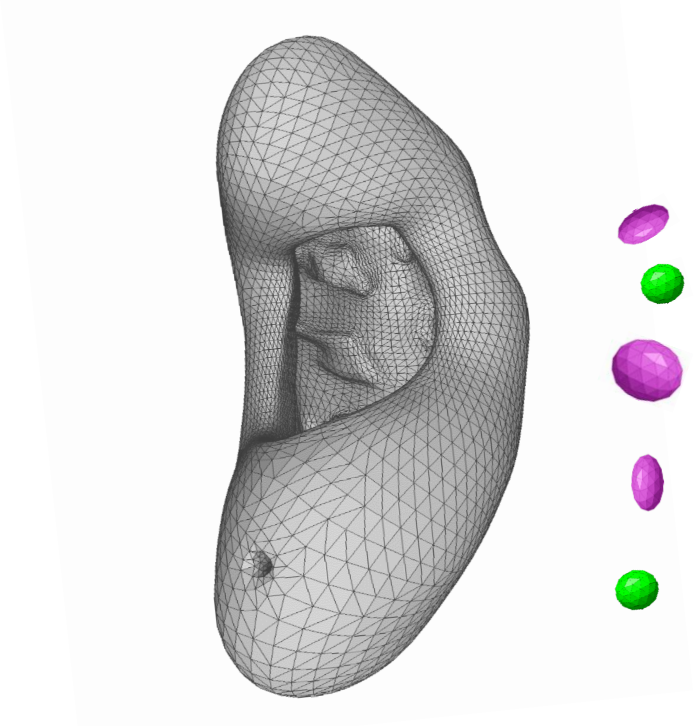


Shape aware MBBs

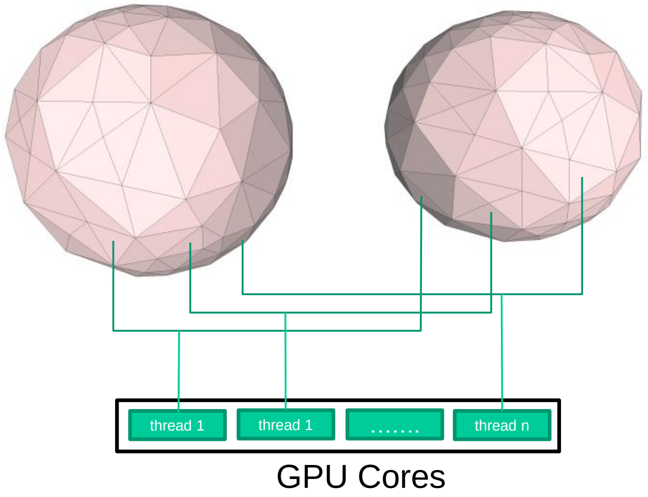
GPU Acceleration for Collison Detection/Intersection

Stony Brook University

- Inter-object parallelism: each object (cell) can be evaluated independently, ideal for GPU
- Intra-object parallelism: each object has multiple surfaces, and distance computation can be evaluated for each surface



Object based parallelization (CPU or GPU)



Surface level parallelization

Performance

Stony Brook University

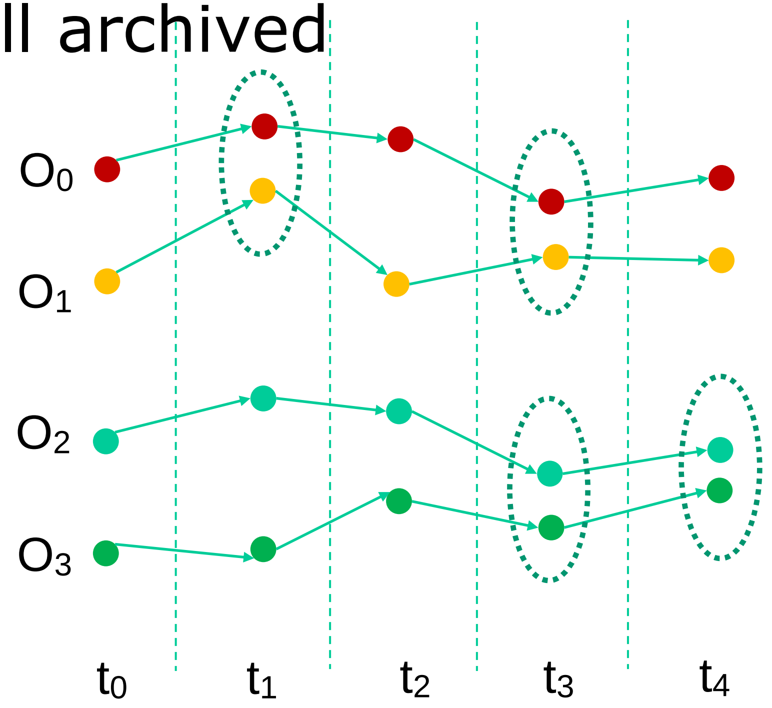
- 4.4X, 13.8X, 880X, 37.4X, 264X faster

Query	Datasets	Test ID	Paradigm	Brute-force	Partition	AABB	GPU	Partition+GPU
Intersect	Nuclei-Nuclei	I-NN	FR	356.0	355.7	338.2	340.4	N/A
			FPR	**84.8**	**86.4**	**82.7**	**80.7**	N/A
Nearest Neighbor	Nuclei-Nuclei	N-NN	FR	2264.0	2268.9	516.9	267.9	N/A
			FPR	**893.8**	**893.1**	**306.6**	**164.1**	N/A
	Nuclei-Vessel	N-NV	FR	151630.0	1649.8	108799.9	62506.1	392.8
			FPR	**24968.1**	**422.2**	**21025.6**	**10202.0**	**172.3**
Within	Nuclei-Nuclei	W-NN	FR	2253.7	2249.0	480.2	250.8	N/A
			FPR	**108.2**	**108.5**	**74.7**	**60.5**	N/A
	Nuclei-Vessel	W-NV	FR	25056.8	645.1	11197.3	9827.0	196.3
			FPR	**8458.8**	**111.6**	**1948.7**	**2990.1**	**95.1**

Table 1: The execution time (Seconds) of tests for three queries with different datasets and accelerating approaches

GLINT: GPU-based Real-Time Contact Tracing

- Contact tracing is an essential tool to control the spread of infectious diseases like COVID-19
- A person is considered at risk if the person is within a specific distance (spatial constraint) for a specific period of time (temporal constraint)
- To achieve accurate retrieval of human traces, the locations should be sampled frequently (sub-second level) but not all archived
  - 12.7TB data with 1M people sampled per second



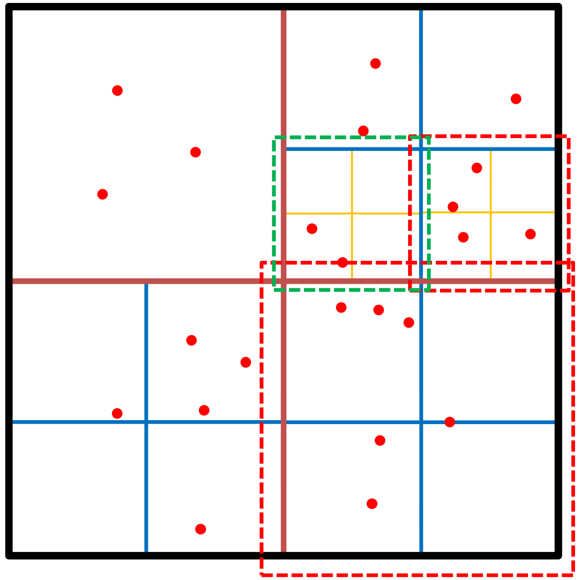
GLINT: a GPU-based real-time contact tracing system that can achieve sub-second response for contact tracing data processing for a population at tens of millions scale

[SIGSPATIL2021]

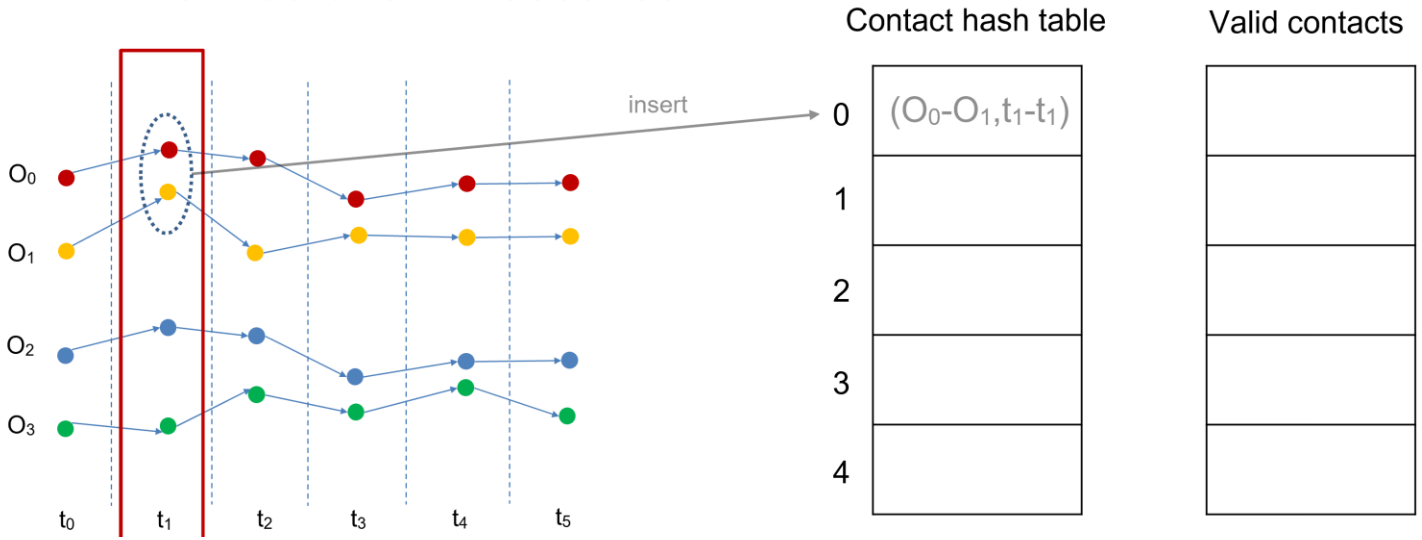
Stony Brook University

GLINT

- Dynamic indexing of moving objects using an adaptive partitioning schema on GPU with very low overhead
- GPU optimized refining
- Temporal evaluation with GPU-based hash table

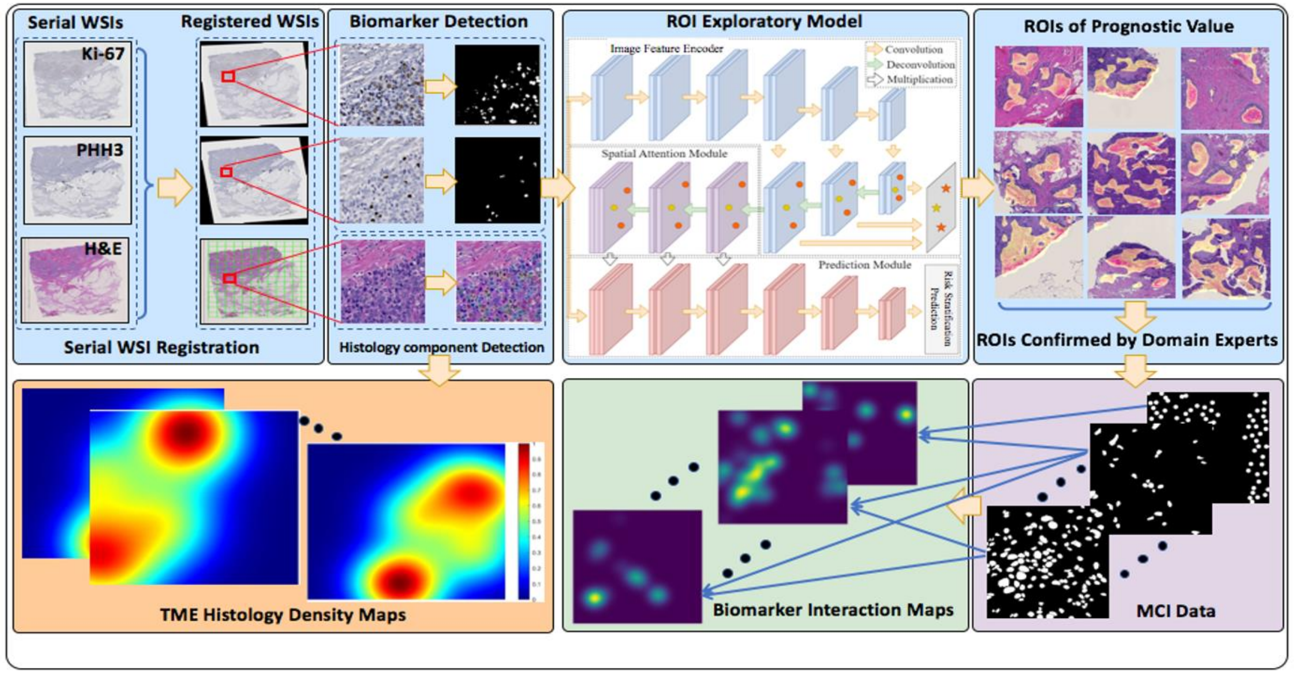


- Minimum duration: 2 seconds



Ongoing Project: Multi-Modal Multl-Scale Integrated Spatial and Image Analytics

- Understand complex relationships of cells and their functions
- Incorporate spatial knowledge into images for multi-modal predictions



Integrate spatial big data management, high performance computing, computational geometry, computer vision and machine learning

Source Codes

- PAIS: http://bmidb.cs.stonybrook.edu/pais/

  Parallel spatial database for whole slide imaging
- Hadoop-GIS: http://bmidb.cs.stonybrook.edu/hadoopgis/index

  Hadoop based spatial querying system
- SparkGIS: http://bmidb.cs.stonybrook.edu/sparkgis/index

  Spark based spatial querying system
- iSPEED: http://bmidb.cs.stonybrook.edu/ispeed/index

  3D spatial querying system
- 3DPro: http://bmidb.cs.stonybrook.edu/3dpro/index

  Progressive 3D querying system
- IDEAL: http://bmidb.cs.stonybrook.edu/ideal/index

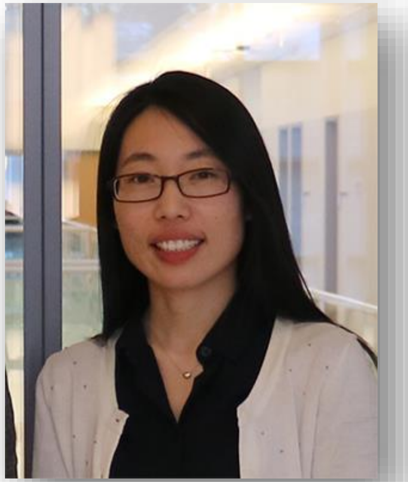
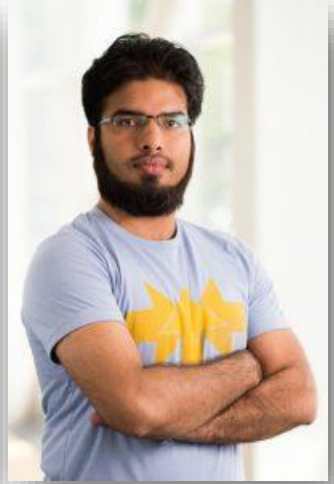
  Hybrid vector-raster model for complex polygons
- GLINT: http://bmidb.cs.stonybrook.edu/glint/index

  GPU-based contact tracing

Team

Ph.D. students (current and past)



Collaborators

Jun Kong

GSU and Emory

Questions?

fusheng.wang@stonybrook.edu

http://www.cs.stonybrook.edu/~fuswang

Look for students working on advanced projects,
Ph.D. student to work on big spatial big data systems