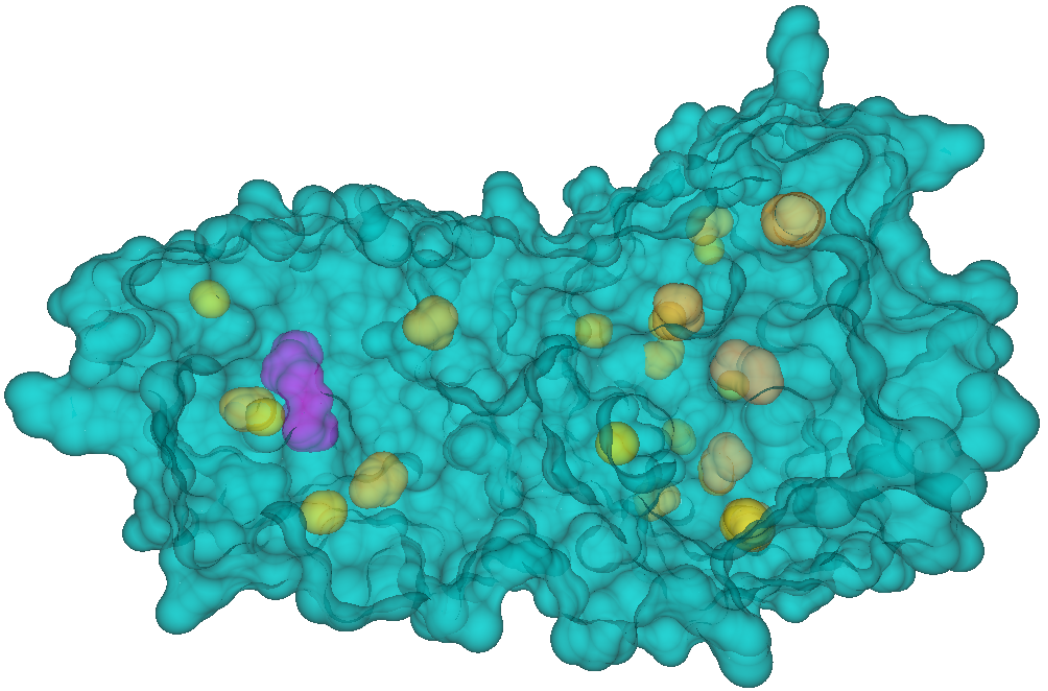


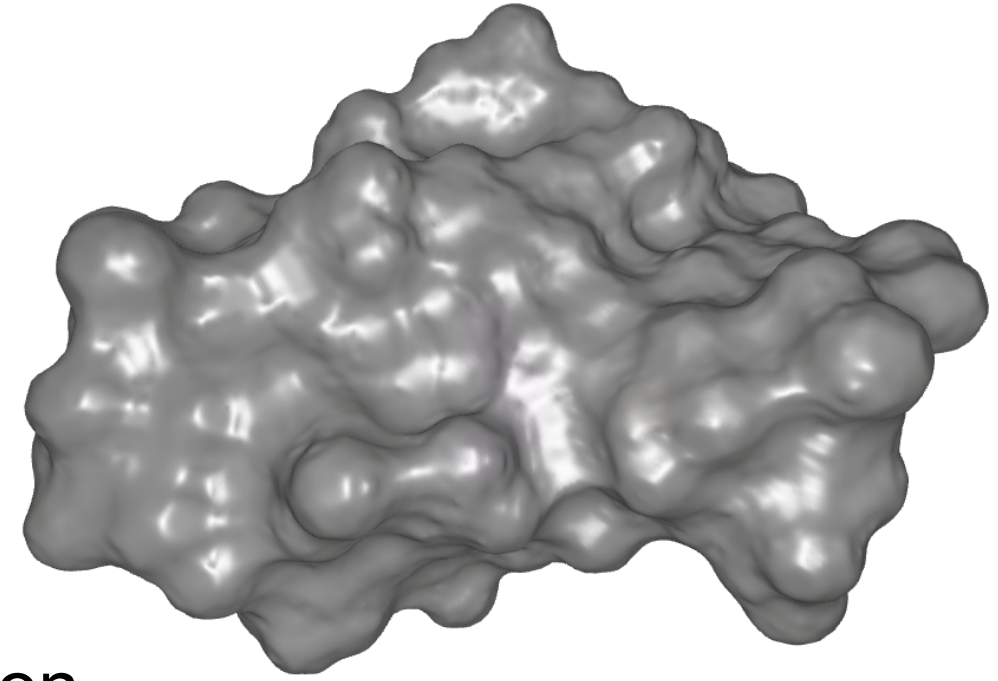
Accelerated Visualization of Transparent Molecular Surfaces in Molecular Dynamics

Adam Jurčík, Jiří Sochor, Barbora Kozlíková (Masaryk University)
Julius Parulek (University of Bergen)

PacificVIS 2016
Taipei, Taiwan

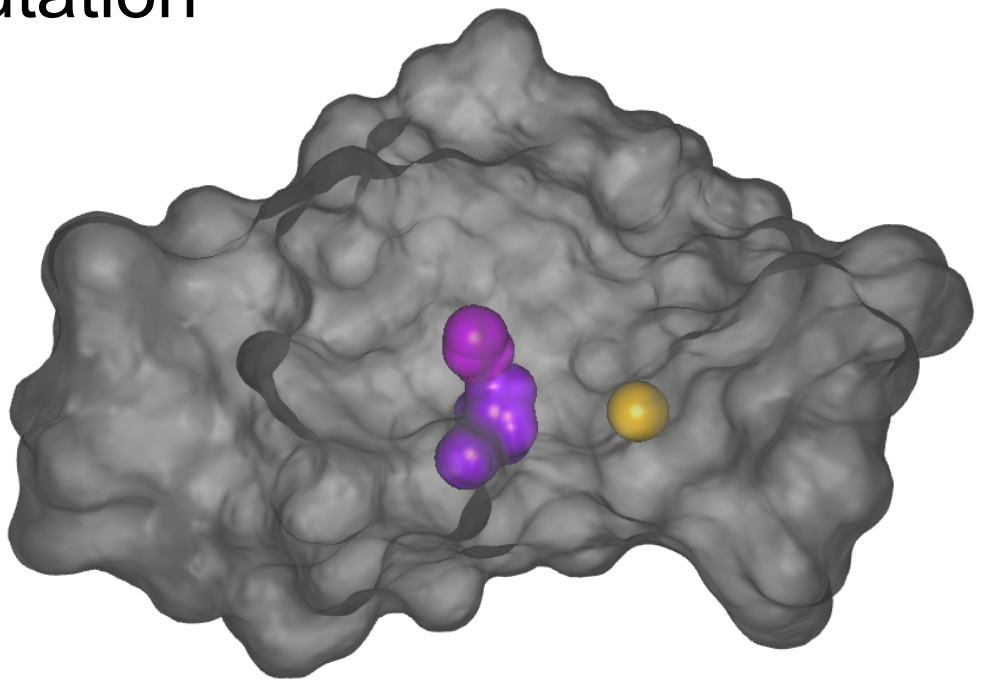
Protein Surfaces in Biochemistry

- Proteins – in all living cells
- Chemical reactions
  - On the surface
  - In the protein – **voids**
- Molecular dynamics
  - Natural motion simulation
  - 50,000 snapshots



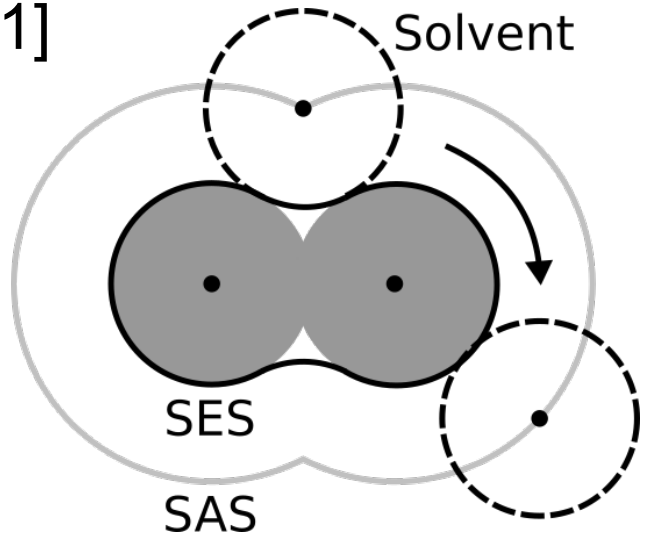
Motivation

- Visualization of inner surfaces
  - Transparency – opacity modulation
  - Interactivity – precomputation
- Transparency in MD
  - 50,000 snapshots
  - Surfaces change
  - Parameters change
  - **No precomputation**

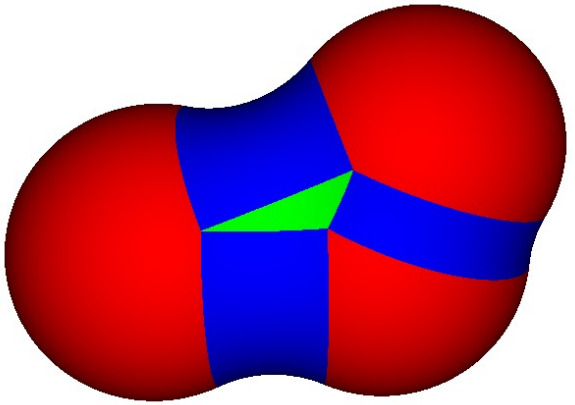


Molecular Surface

- Solvent Accessible [Lee et al. '71]
  - Spherical patches

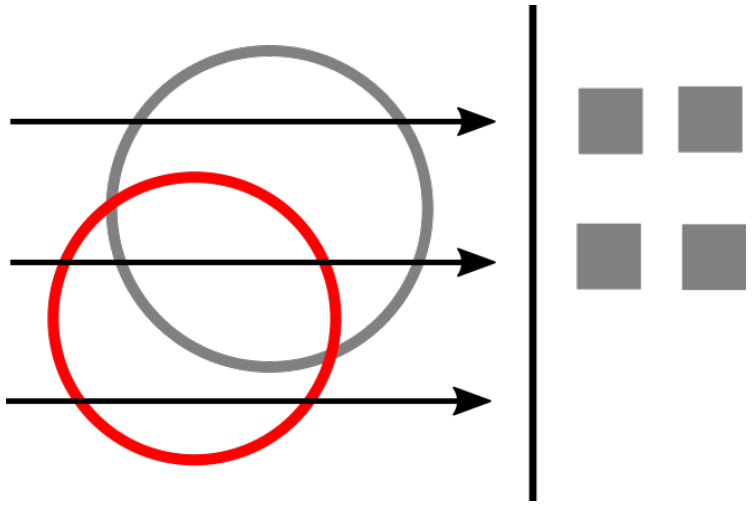


- Solvent Excluded [Connolly '83]
  - Spherical patches
  - Toroidal patches
  - Spherical triangles



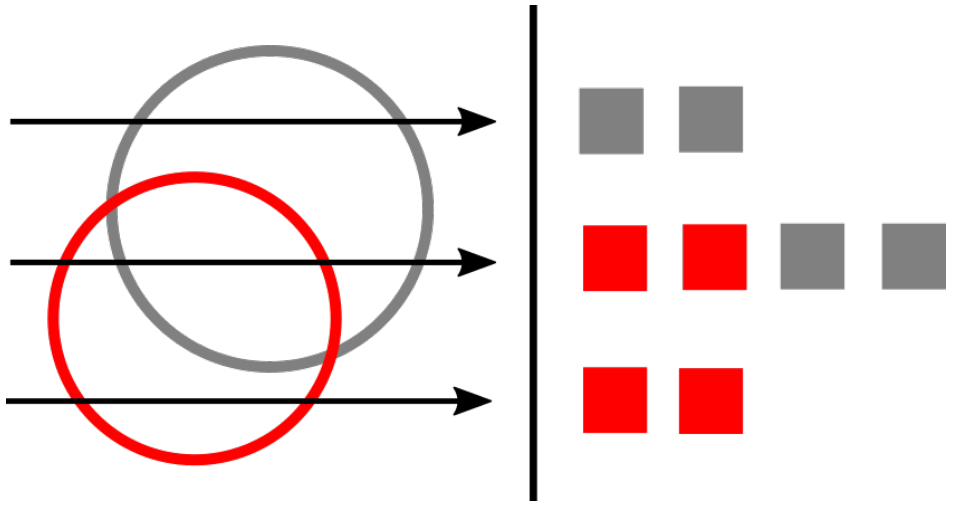
Transparent Molecular Surface

- Molecular surface using order independent transparency [Kauker et al. '13]
  - Use fragments of all atom spheres
  - CSG operations on all fragments



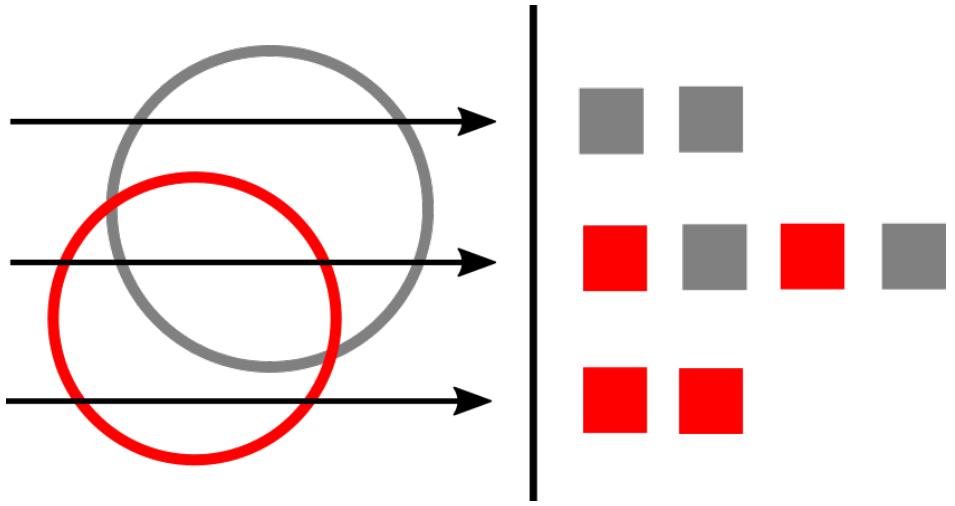
Transparent Molecular Surface

- Molecular surface using order independent transparency [Kauker et al. '13]
  - Use fragments of all atom spheres
  - CSG operations on all fragments



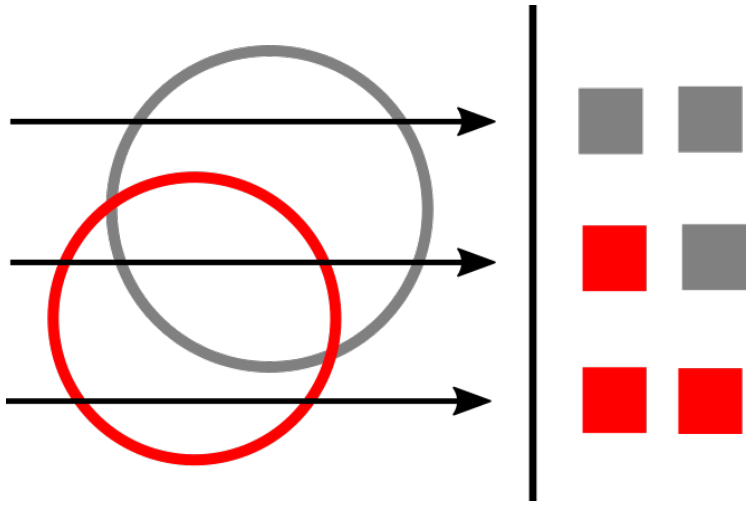
Transparent Molecular Surface

- Molecular surface using order independent transparency [Kauker et al. '13]
  - Use fragments of all atom spheres
  - CSG operations on all fragments



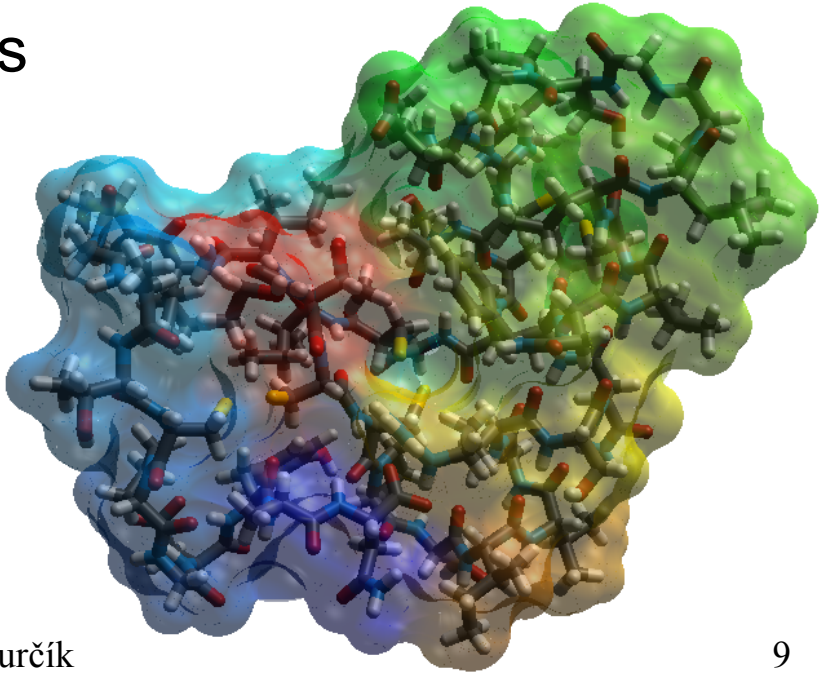
Transparent Molecular Surface

- Molecular surface using order independent transparency [Kauker et al. '13]
  - Use fragments of all atom spheres
  - CSG operations on all fragments



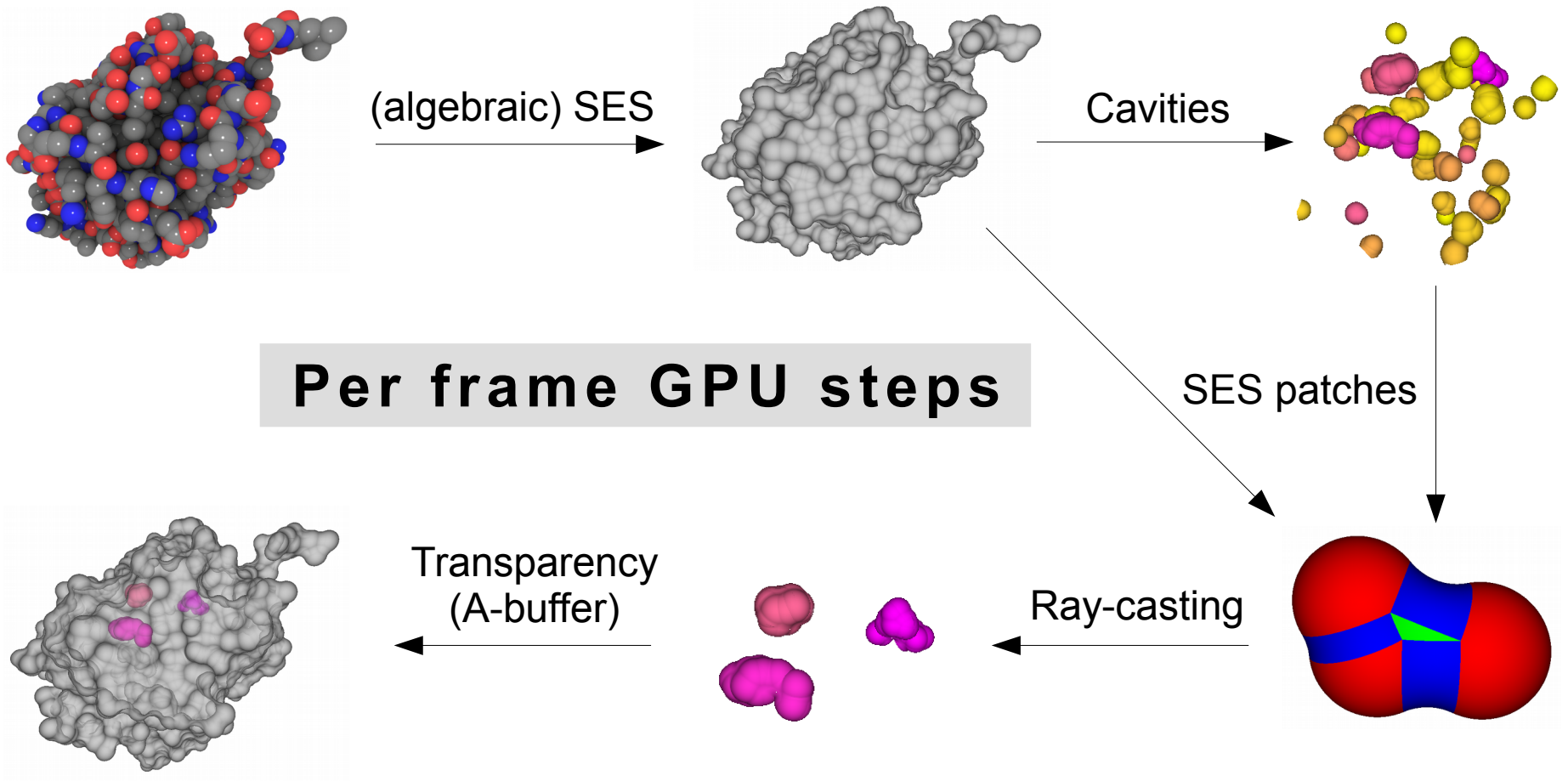
Transparent Molecular Surface

- Molecular surface using order independent transparency [Kauker et al. '13]
  - Correct transparency
  - Performance bottleneck – interaction
    - 6.2 frames/s for ~10000 atoms



Accelerated Transparent MS

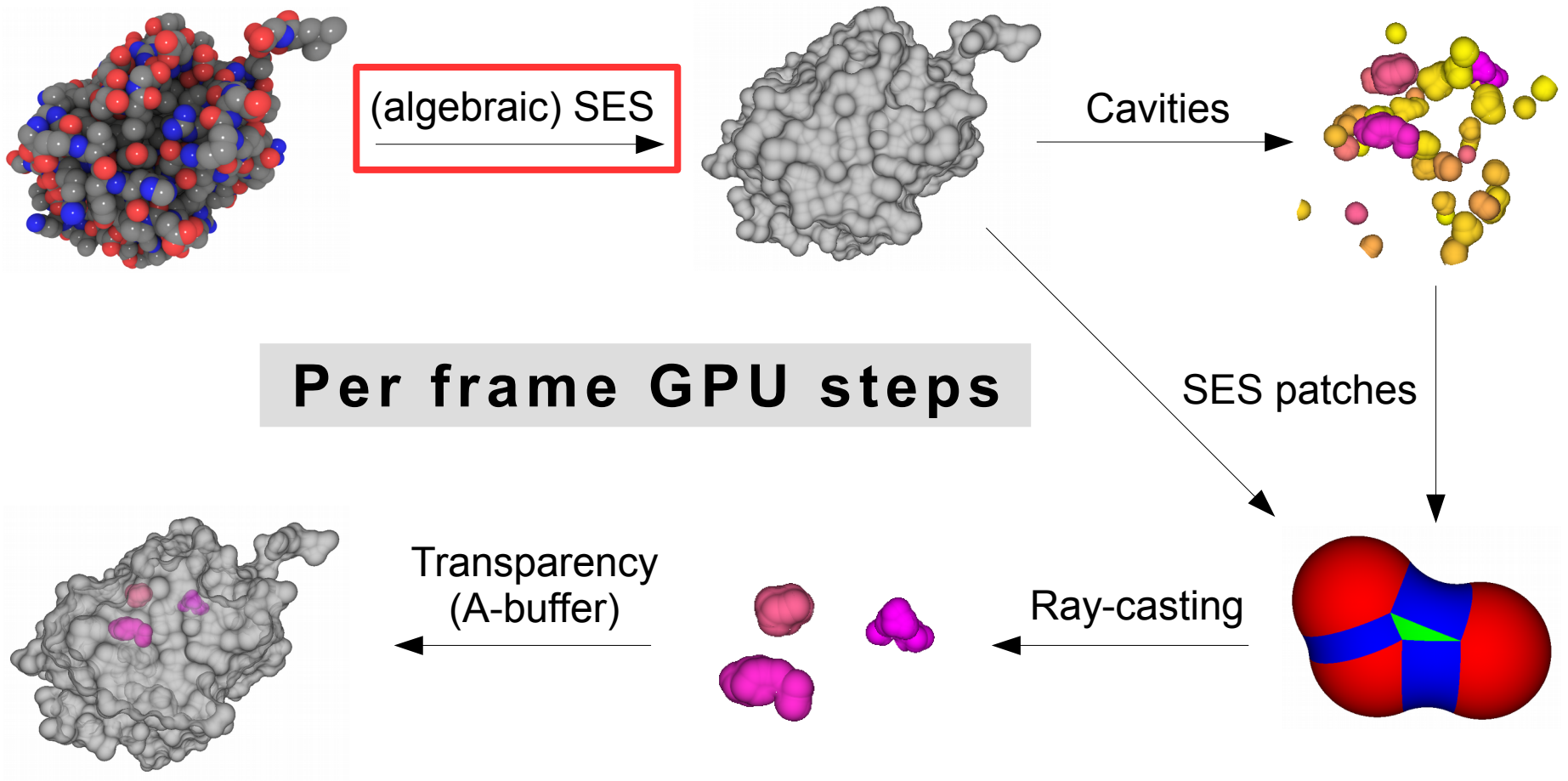
Input: Atom positions



Output: Transparent SES

Accelerated Transparent MS

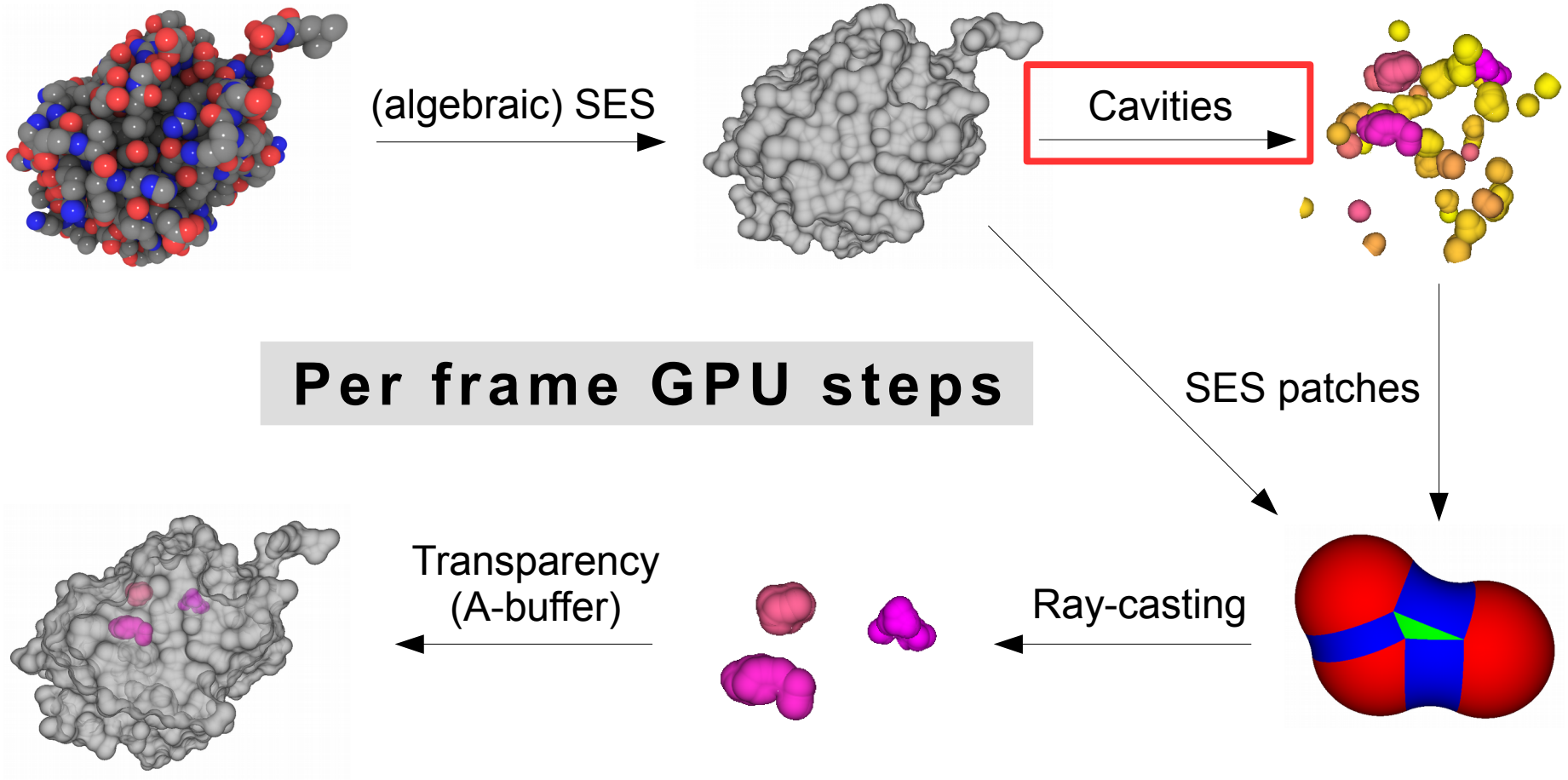
Input: Atom positions



Output: Transparent SES

Accelerated Transparent MS

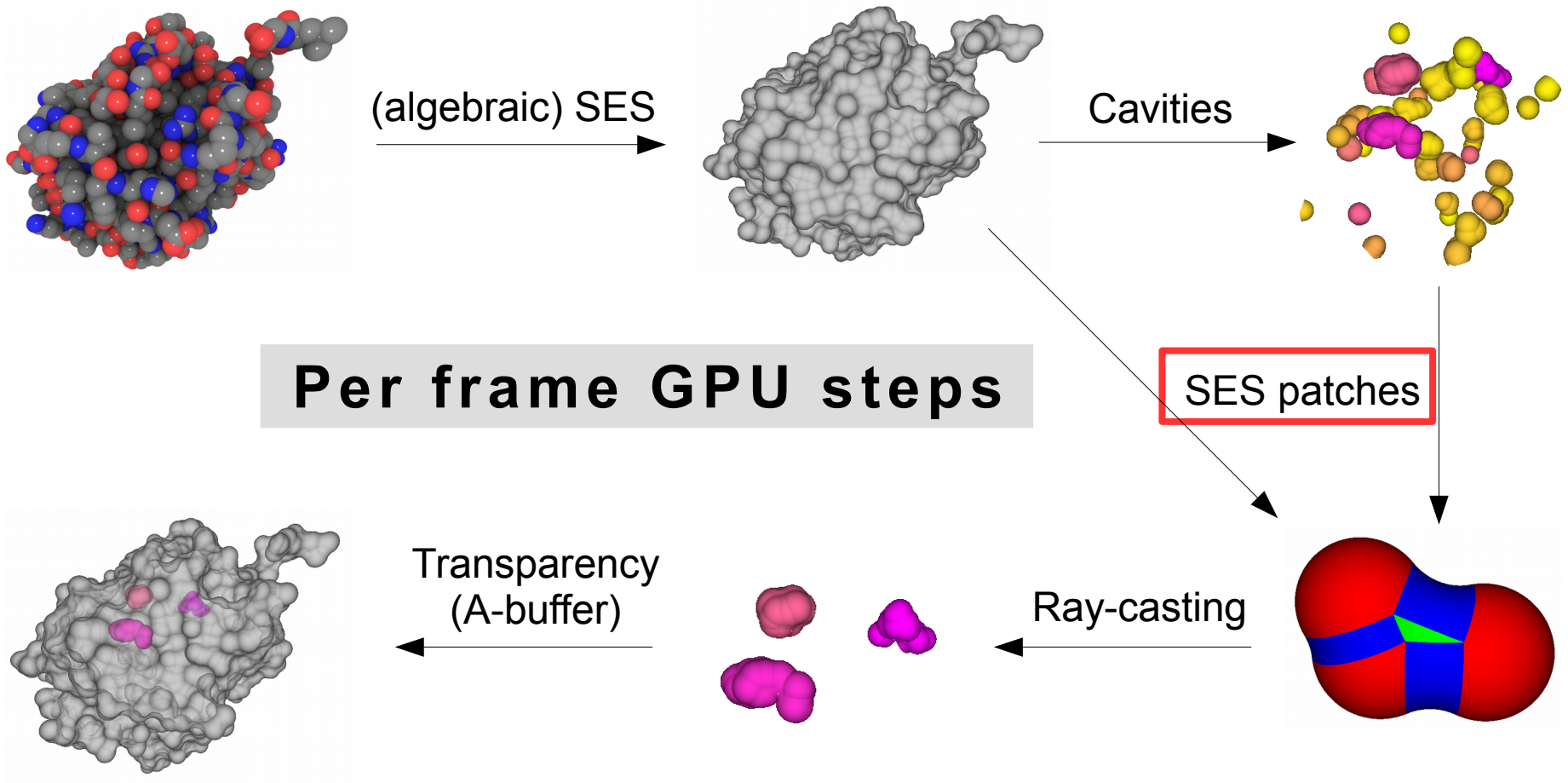
Input: Atom positions



Output: Transparent SES

Accelerated Transparent MS

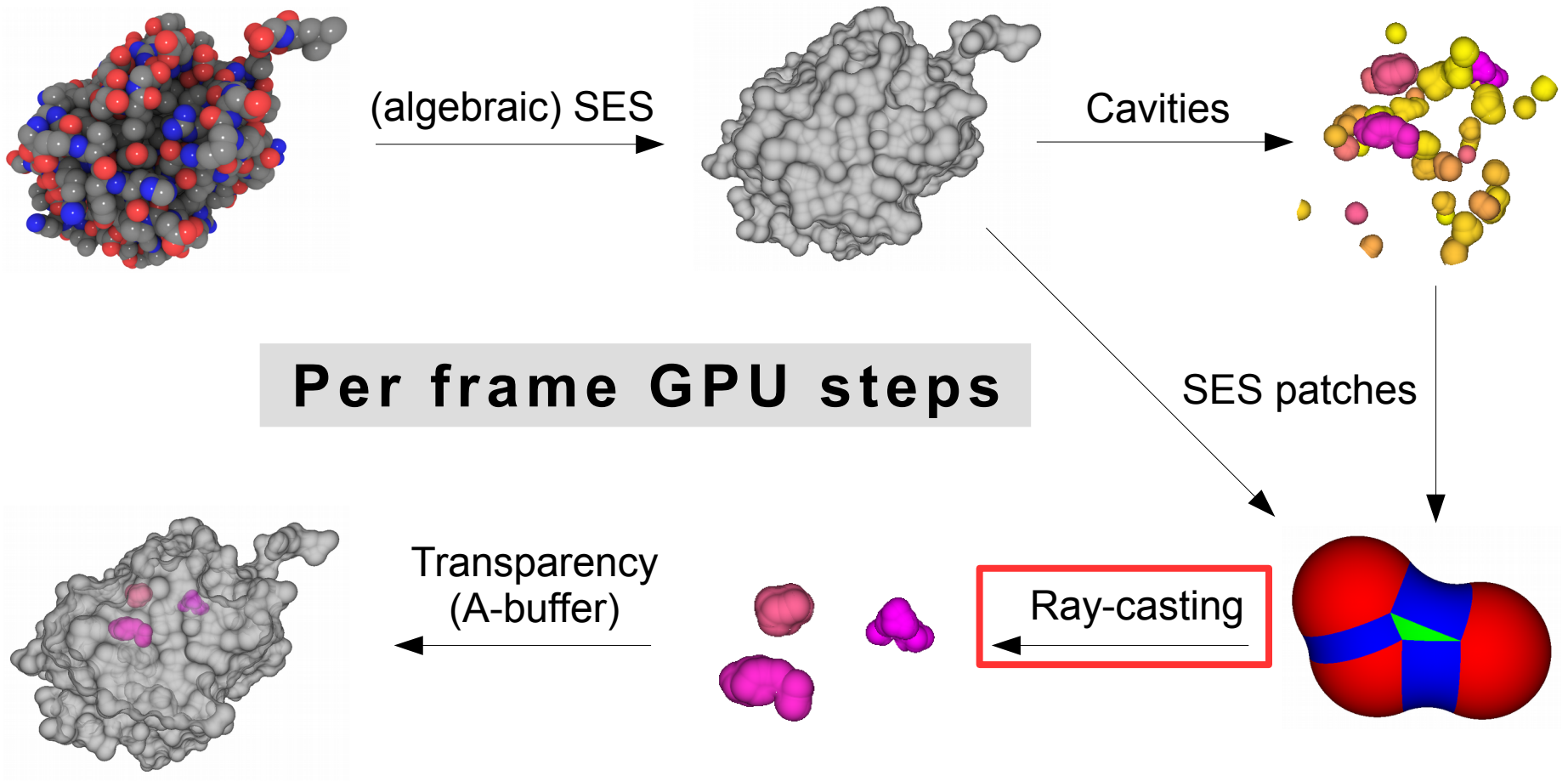
Input: Atom positions



Output: Transparent SES

Accelerated Transparent MS

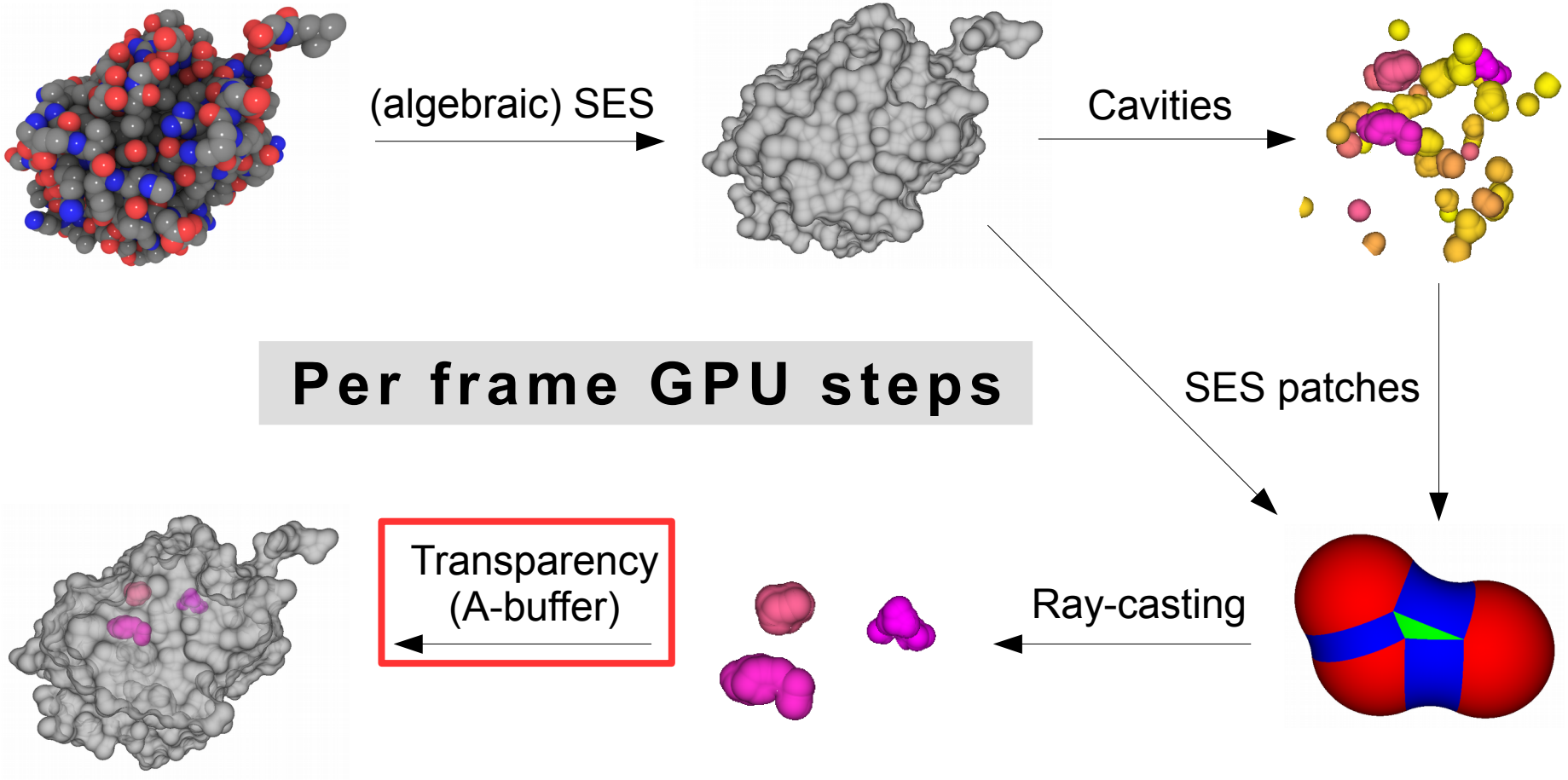
Input: Atom positions



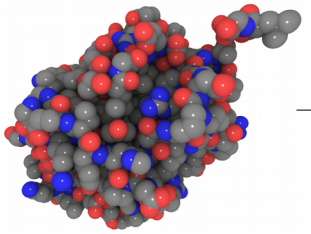
Output: Transparent SES

Accelerated Transparent MS

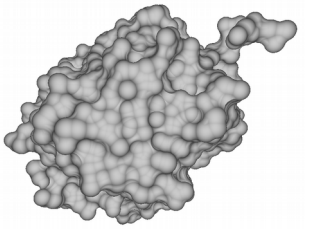
Input: Atom positions



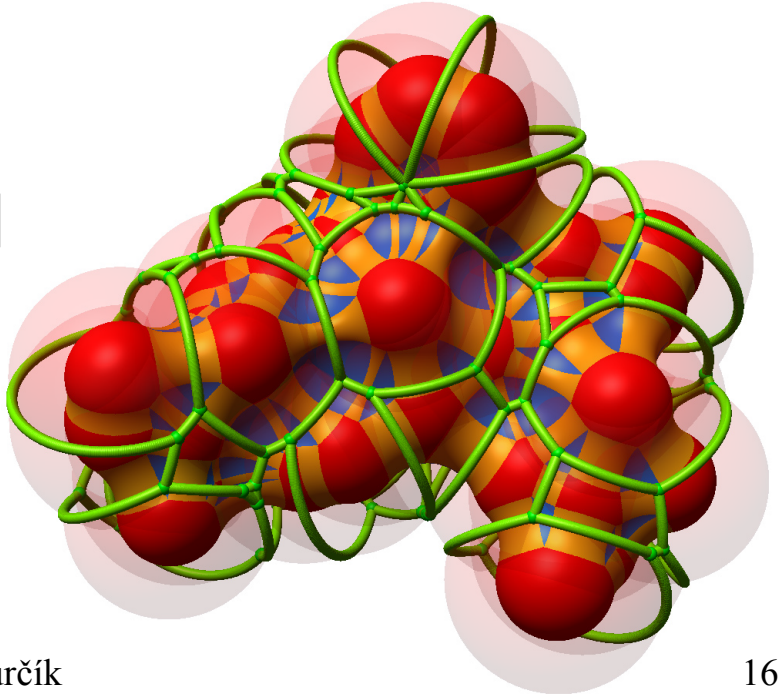
Output: Transparent SES

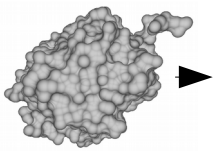


Surface Computation

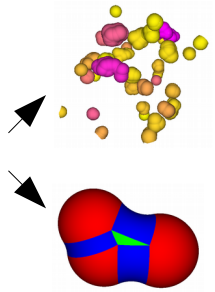


- Contour-buildup algorithm [Totrov et al. '96]
  - Accelerated and **localized** computation
- Parallelization
  - Multiple CPUs [Lindow et al. '10]
  - Single GPU [Krone et al. '11]

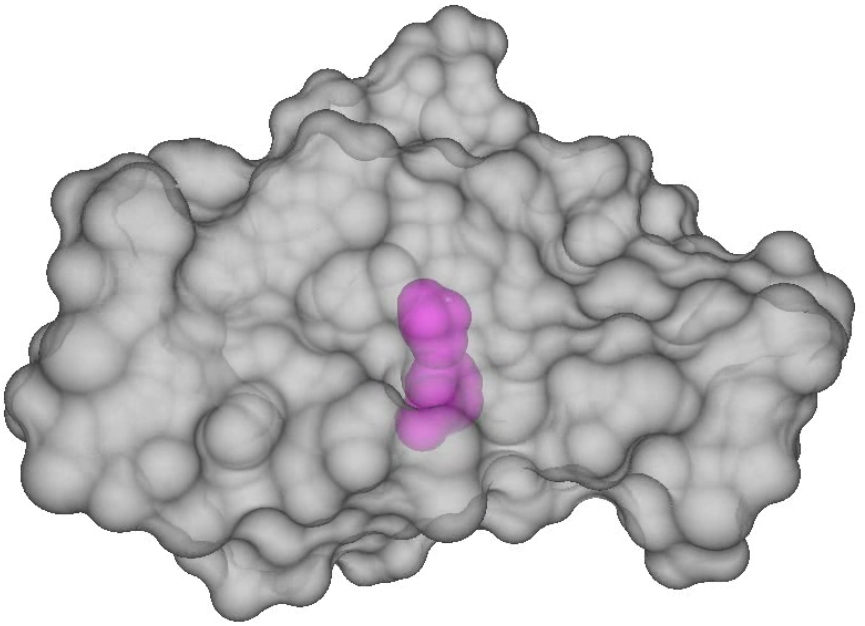


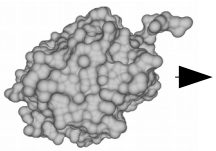


Cavities and patches extraction

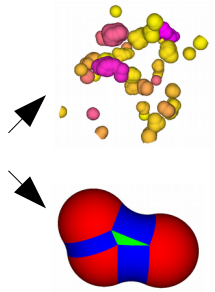


- Observations:
  - Surfaces = isolated connected components (CC)
  - Spherical patches are enclosed with tori
  - Tori connect triangles
- Graph algorithms:
  1) Adjacency list
  2) CC analysis – use BFS
  3) Cycles forming patches

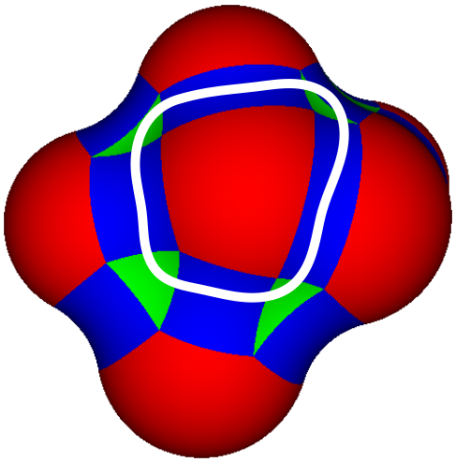




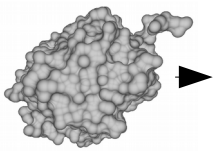
Cavities and patches extraction



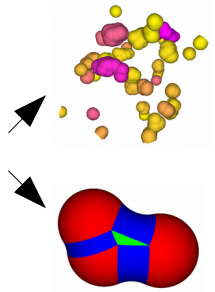
- Observations:
  - Surfaces = isolated connected components (CC)
  - Spherical patches are enclosed with tori
  - Tori connect triangles



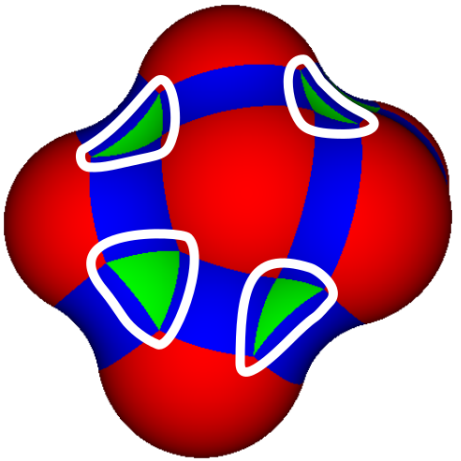
- Graph algorithms:
  1) Adjacency list
  2) CC analysis – use BFS
  3) Cycles forming patches



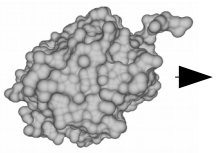
Cavities and patches extraction



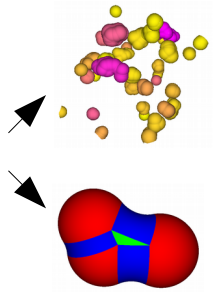
- Observations:
  - Surfaces = isolated connected components (CC)
  - Spherical patches are enclosed with tori
  - Tori connect triangles



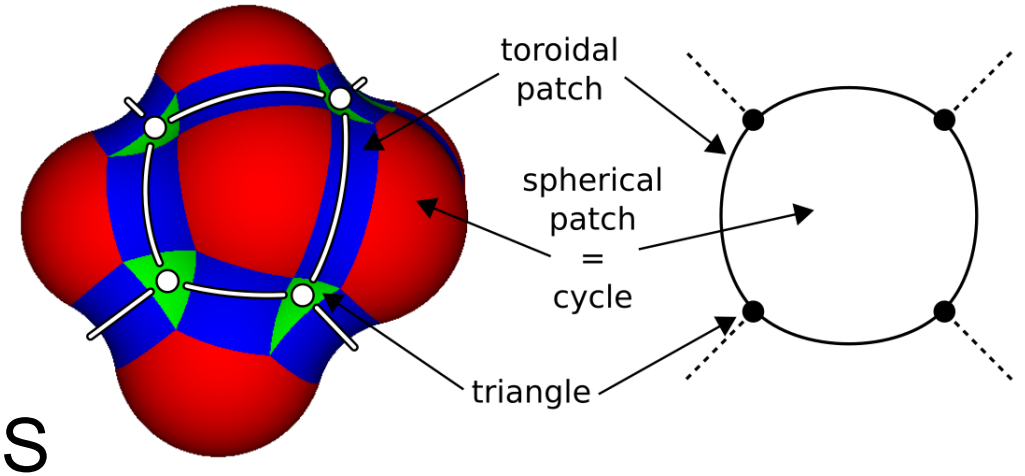
- Graph algorithms:
  1) Adjacency list
  2) CC analysis – use BFS
  3) Cycles forming patches

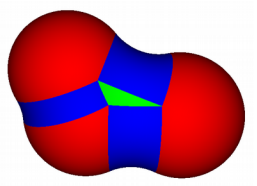


Cavities and patches extraction

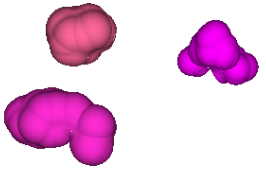


- Observations:
  - Surfaces = isolated connected components (CC)
  - Spherical patches are enclosed with tori
  - Tori connect triangles
- Graph algorithms:
  1) Adjacency list
  2) CC analysis – use BFS
  3) Cycles forming patches

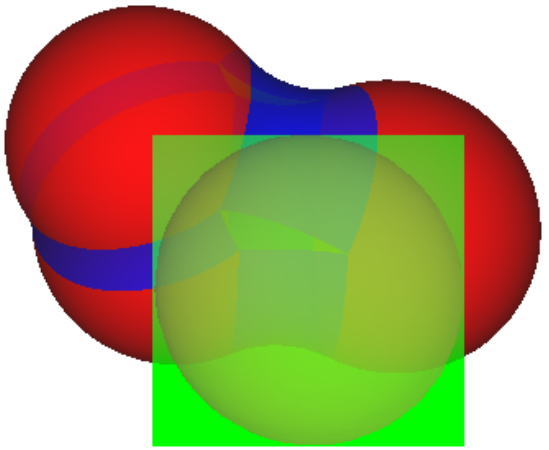




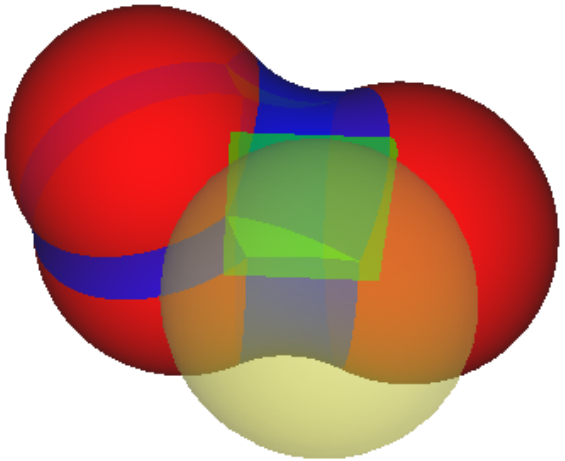
Ray-casting



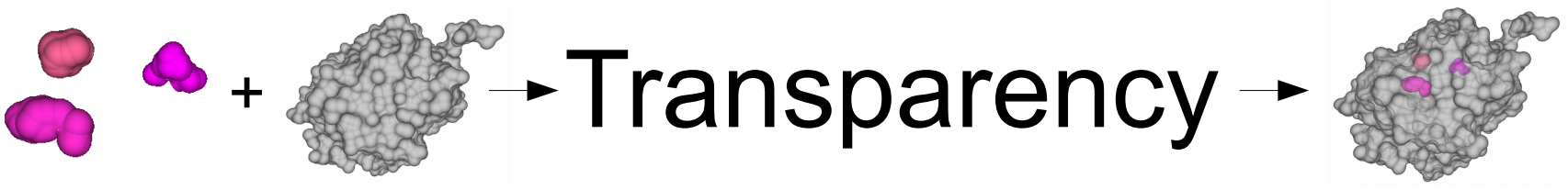
- Individual SES patches
  - OBB splats – geometry shader
  - Less rays – higher performance



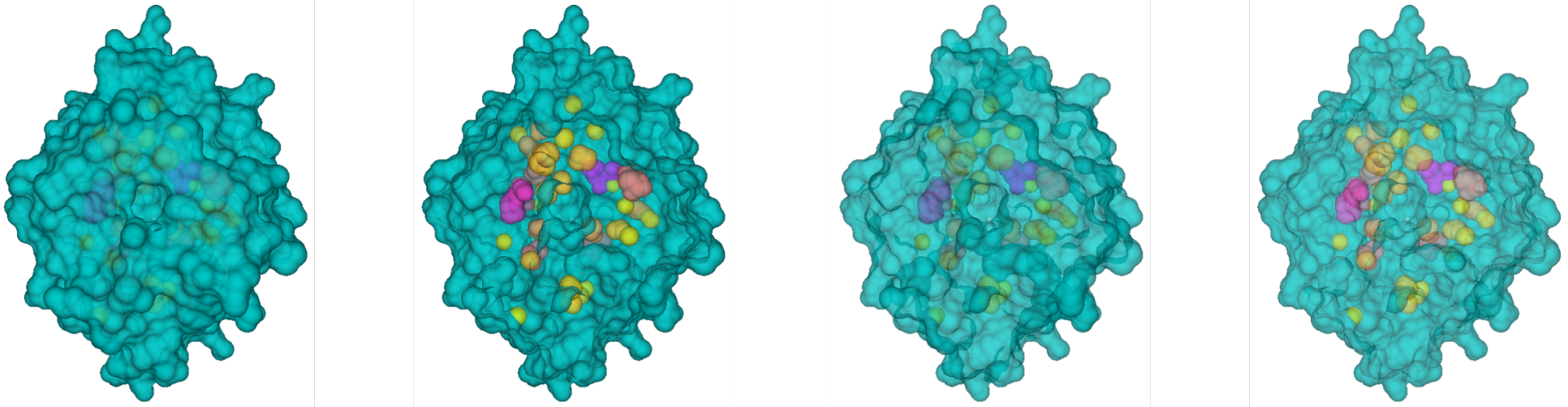
full sphere splat
5.29 ms (~2500 atoms)



OOB splat
< 42% rays, 3.66 ms (~2500 atoms)



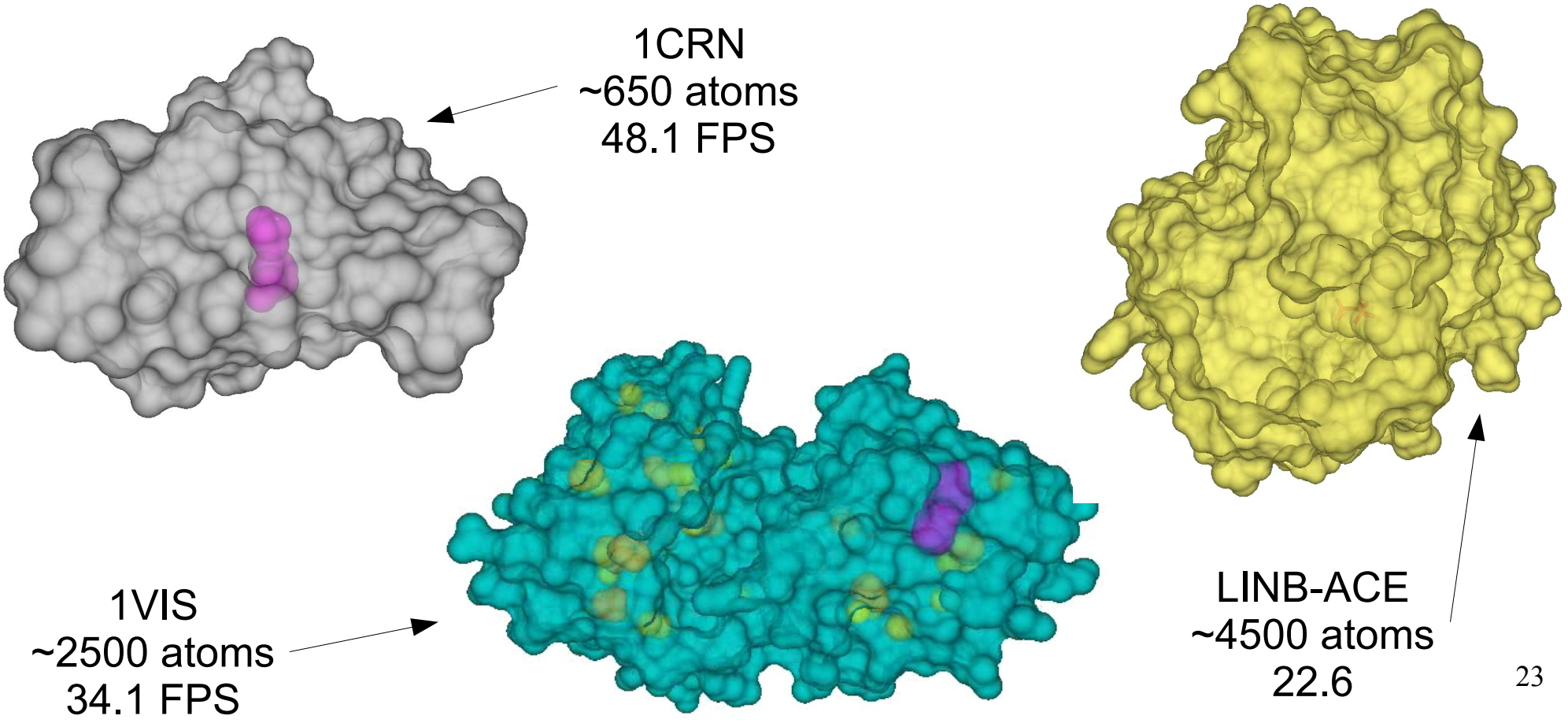
- A-buffer – all surface fragments
- Opacity modulation
  - Overall opacity (O)
  - Entry fragments – opacity suppression (K)



$O = 0.8, K = 1$ $O = 0.8, K = 16$ $O = 0.5, K = 1$ $O = 0.5, K = 16$

Results I

- Transparent SES visualization:
  - Static and dynamic structures – PDB ID



Result I – Video

Results II

- Performance comparison
  - Resolution: 1024 x 768
  - GPU: NVIDIA GF GTX 680

		Our method		Kauker et al.		
PDB ID	Atoms	DL	FPS	DL	FPS	Speedup
1OGZ	~650	12	48.1	117	31.0	**1.55**
1VIS	~2500	15	34.1	135	11.2	**3.04**
4ADJ	~10000	19	15.5	188	6.2	**2.50**

Summary

- Contribution
  - Interactive transparent dynamic SES visualization
- Limitations
  - Transparency perception – interaction
  - Not detecting open void paths
- Future work
  - Detection and coloring of tunnels
  - Experiments with more efficient BFS algorithm

Thank you for your attention!