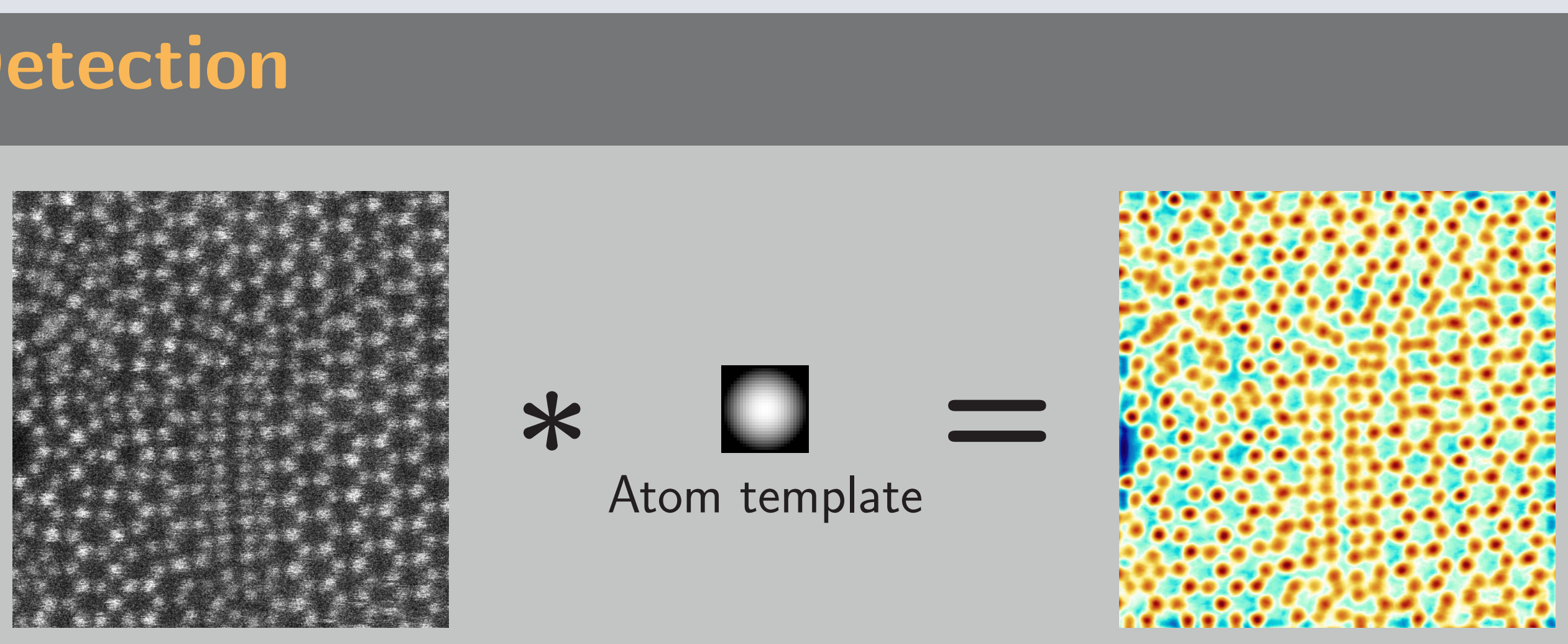
Data Mining Atomically Resolved Images for Material Properties

Sebastian Klaassen
Department of Visualization and Data Analysis
University of Vienna

Introduction

The dataset for challenge 3 of the Smokey Mountain Computational Science and Engineering Conference 2017 consists of a sequence of 49 electron microscopy images. It shows the evolution of Molecules under the electron beam. To study beam-matter interaction, as well as material reconstruction and vacancy formation mechanisms associated with beam induced stress, atoms need to be detected, classified and tracked over time.

Atom Detection

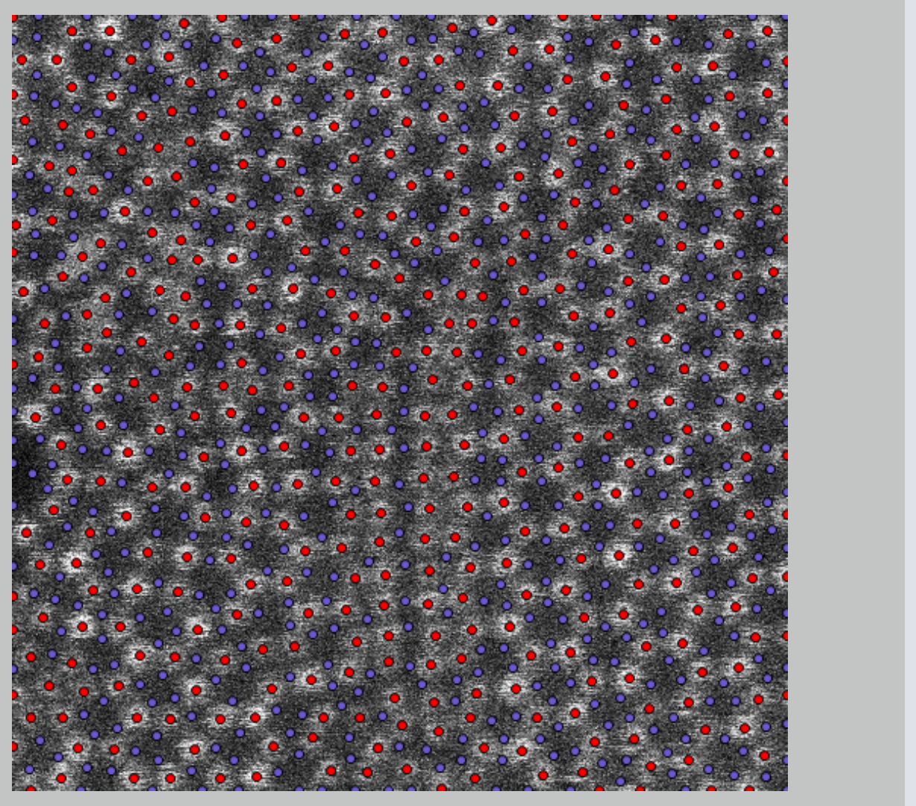


Input image

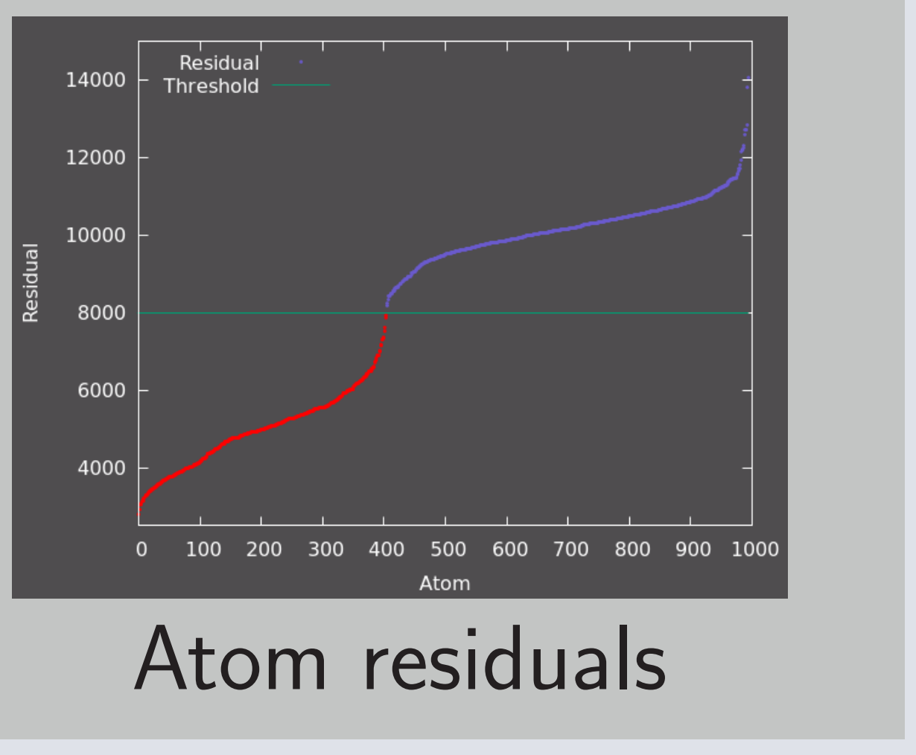
Atom template

Correlation residuals

- We correlate every pixel of the STEM image with a template atom to compute per-pixel dissimilarities (i.e. correlation residuals).
- Pixels are iterated in order of increasing correlation residual. A pixel is marked as an atom center, if it is at least 14 pixels away from other atom centers.
- We choose a threshold to separate true atom centers from noise.

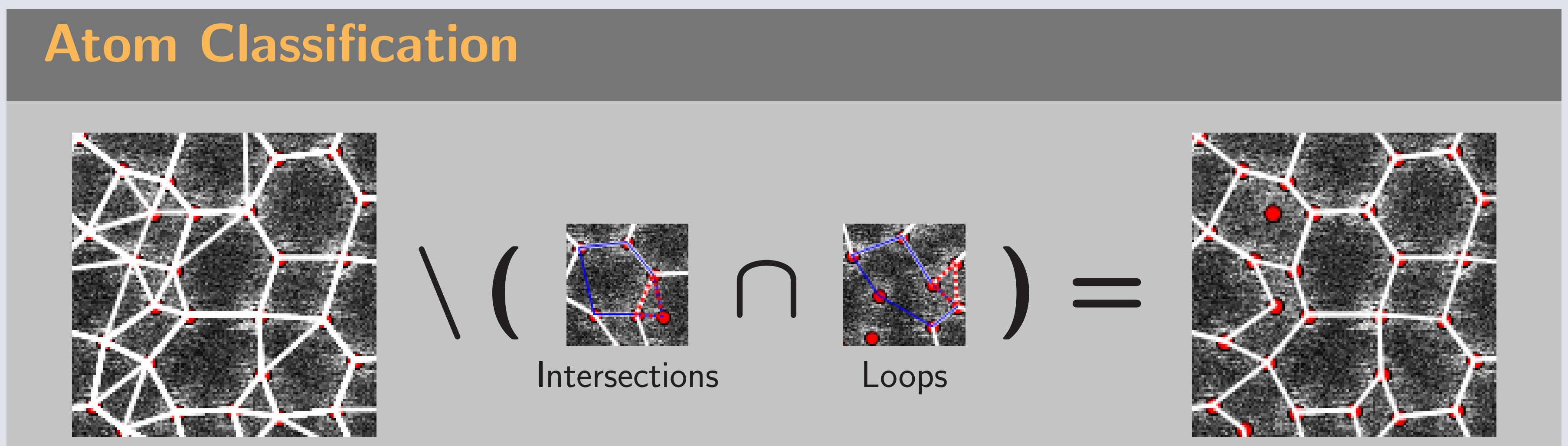


Results (frame 1)



Atom residuals

Atom Classification



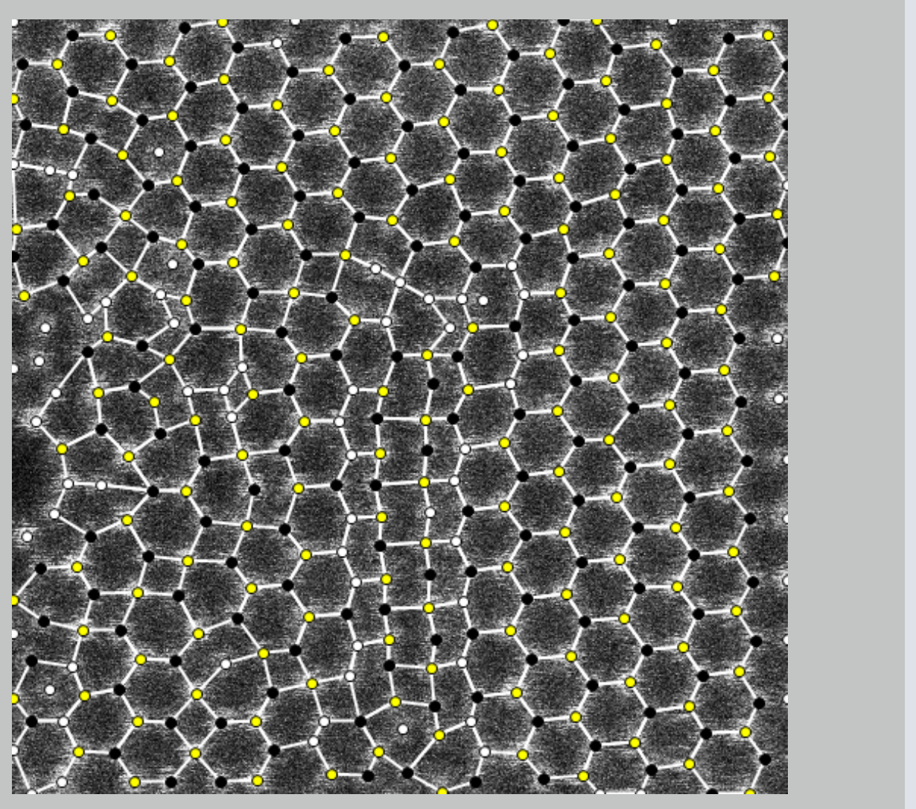
Detected hexagons

Intersections

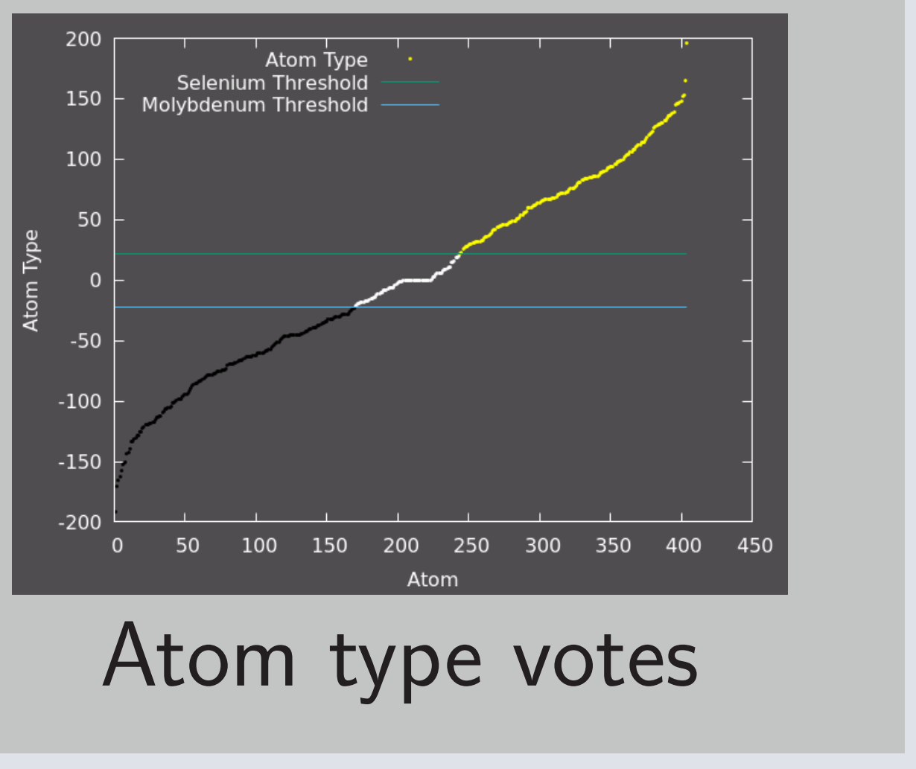
Loops

Valid hexagons

- For each atom, we detect the smallest clockwise and the smallest counter-clockwise molecule (i.e. hexagon).
- Detected hexagons are added in order of uniformity, as long as they don't intersect existing hexagons and their edges don't form length-3 loops with existing hexagons.
- Each molecule casts a vote on the atom type of its atoms based on relative brightness within the molecule.
- The type of an atom is computed from the votes of adjacent molecules.



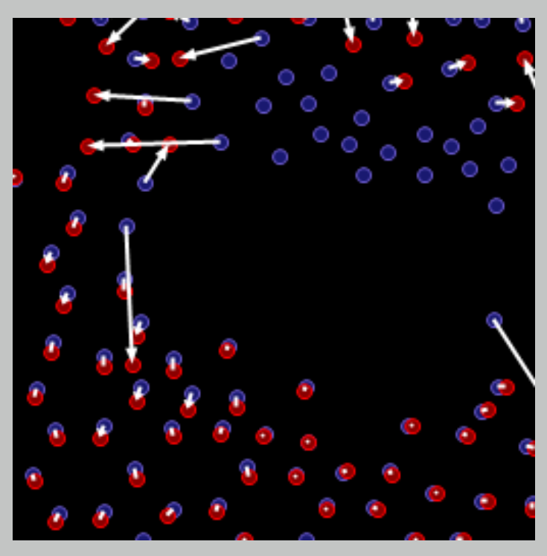
Results (frame 1)



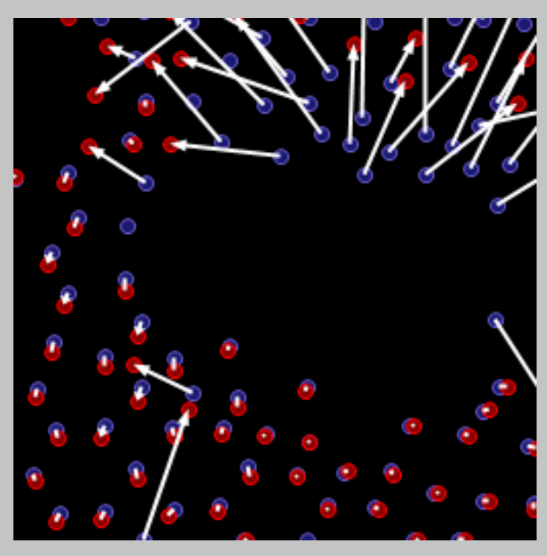
Atom type votes

Atom Tracking

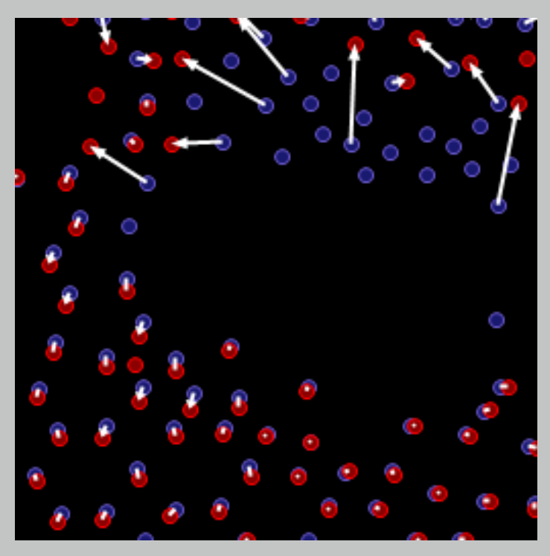
- We track atoms by **finding the best possible mapping** $M[p_0] = p_1$ between all points of the first frame $p_0 \in P_0$ and all points of the second frame $p_1 \in P_1$.
- We start with an **initial mapping**. Since we only search for a local optimum, finding a good initial mapping is crucial.
  - **Proximity Priority mapping** mapps points in order of mapping distance. This method works well in areas of slow atomic motion, but it introduces errors where P_1 points are sparse.
  - **Proximity & Density Priority mapping** mapps points in order of mapping distance and increasing P_1 density. This overcomes the limitations of proximity-priority mapping. We use Proximity & Density Priority as initial mapping.



Initial mapping using Proximity Priority



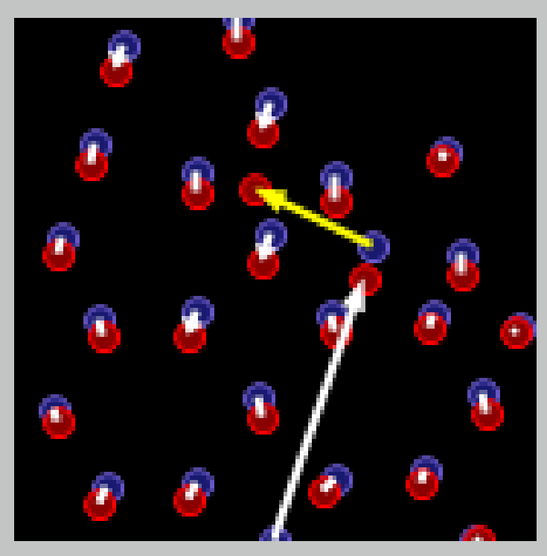
Initial mapping using Proximity & Density Priority



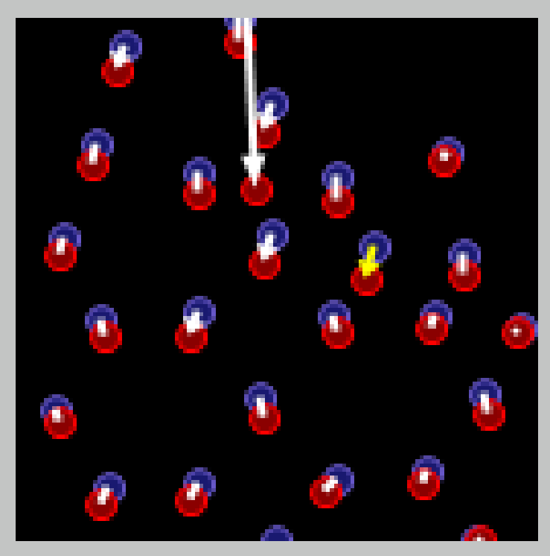
Solution after 64 iterations of iterative refinement

- We improve the initial mapping by **iterative refinement**. Each iteration consists of 3 steps:

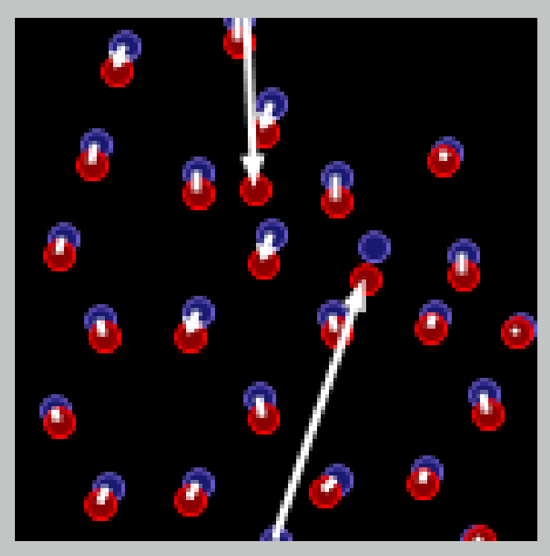
1. Find a point whose mapping seems out of place.
2. Estimate a better mapping for this point.
3. Compare the current mapping (option A) with the estimate (option B) and with a mapping that assumes the point got lost between frames (option C). Pick the option that best improves the solution.



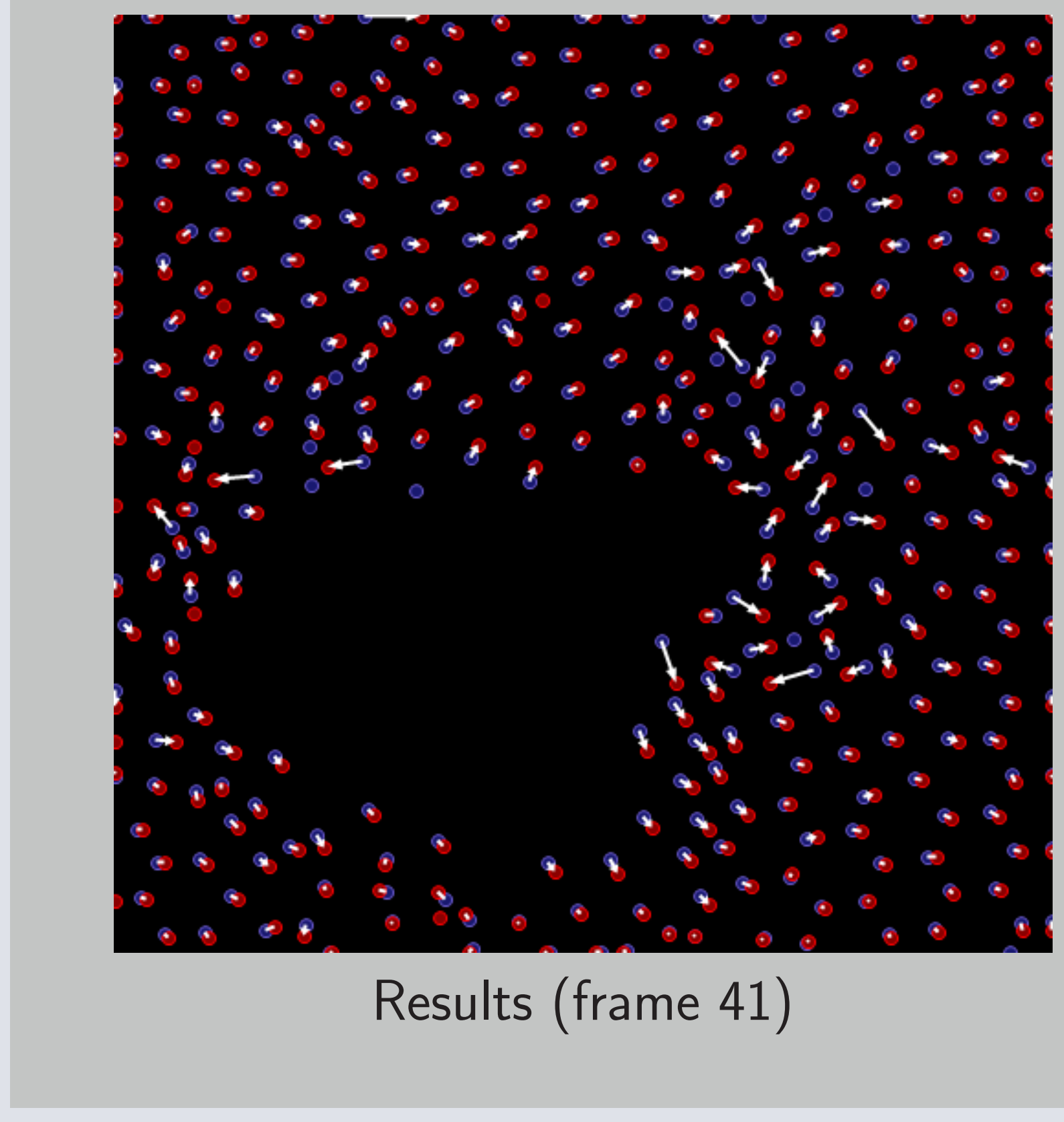
Option A
Keep current mapping



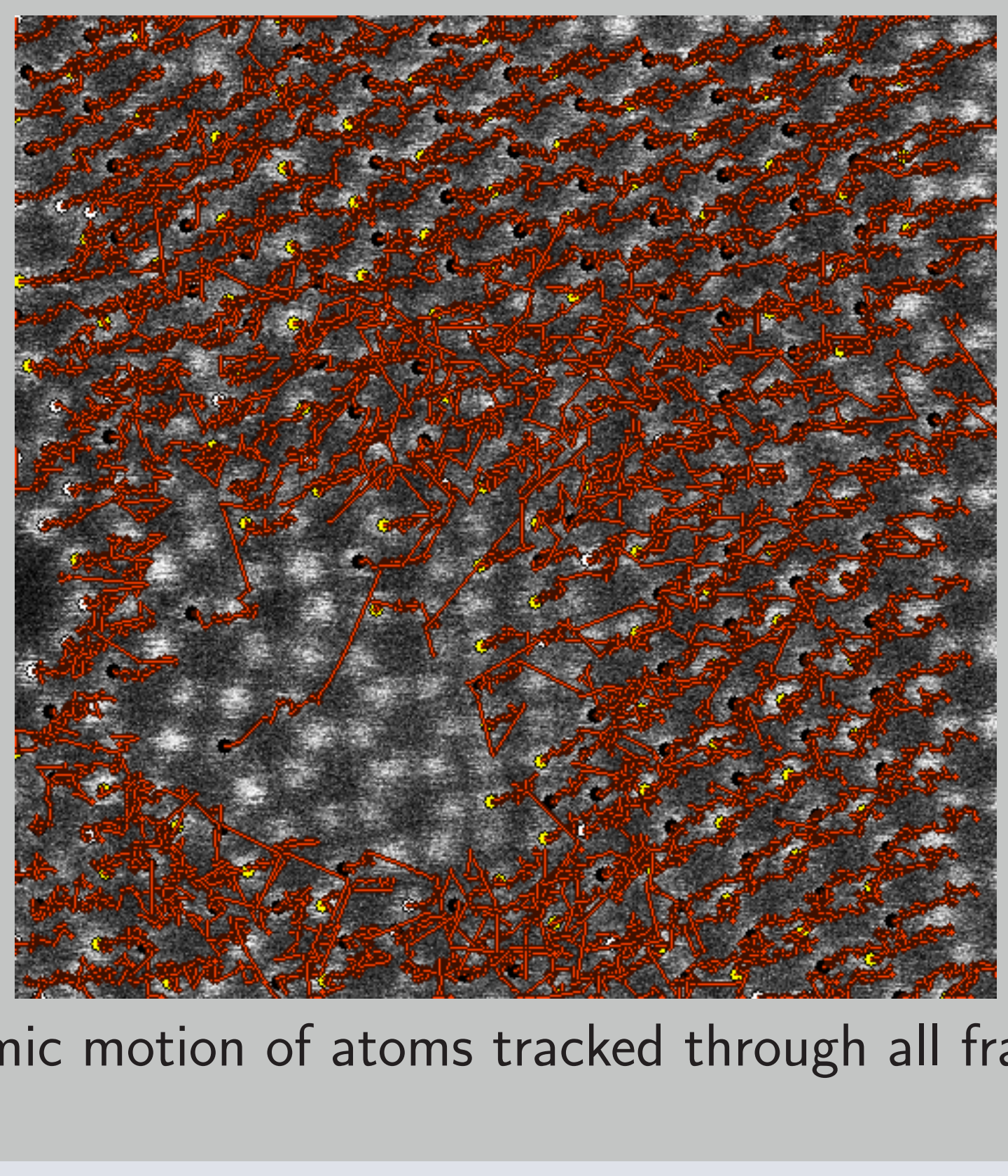
Option B
Replace with estimate



Option C
Remove mapping



Results (frame 41)



Atomic motion of atoms tracked through all frames

Results

- We detect between 343 and 412 atoms on each of the 49 frames.
- 283 atoms are tracked throughout the full animation.
- All algorithms have been implemented in C++

Algorithm	Runtime
Detecting atoms in all 49 frames	17 seconds
Classifying atoms in the first frame	160 milliseconds
Tracking atoms between all 49 frames	7.5 seconds