



浙江大學
ZHEJIANG UNIVERSITY

Introduction to atom_catcher

-a statistical analysis & visualization tool for atomically resolved images

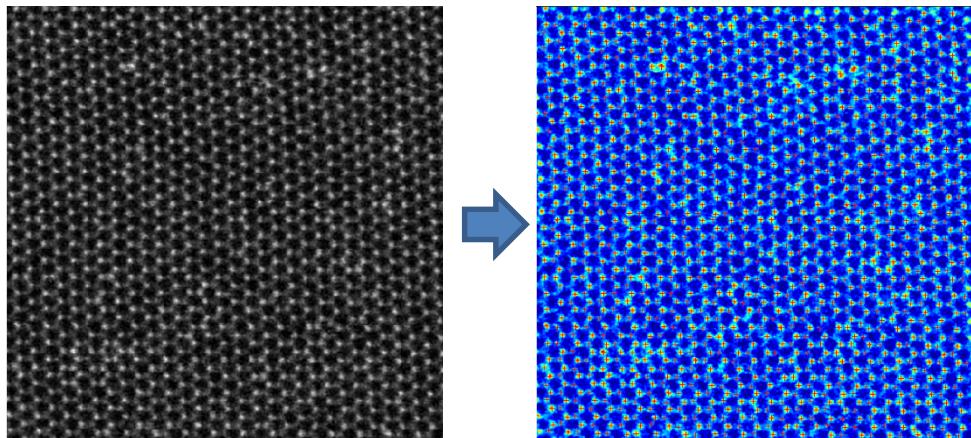
Author : Zhen Fei

Contents

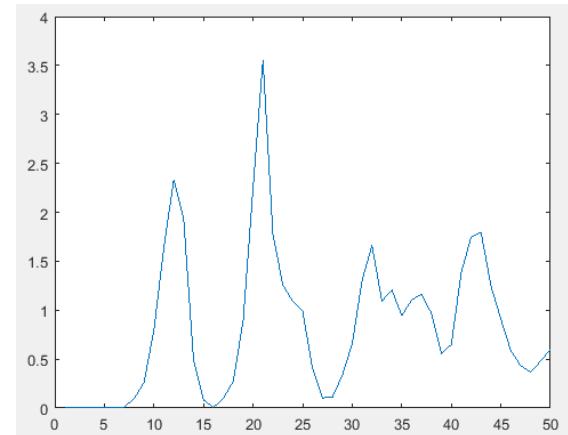
- **What? -features**
 - peak finding & pair function
 - manually add & delete atoms
 - atom intensity histogram with Gaussian mixture model
 - atomic network with alternating sites
 - determining atom type by both intensity and atom site & alloy degree
 - position relaxation(mass center method & Gaussian fitting method)
 - save and reload data
- **How? -workflow**
- **Why? -principles**

Features

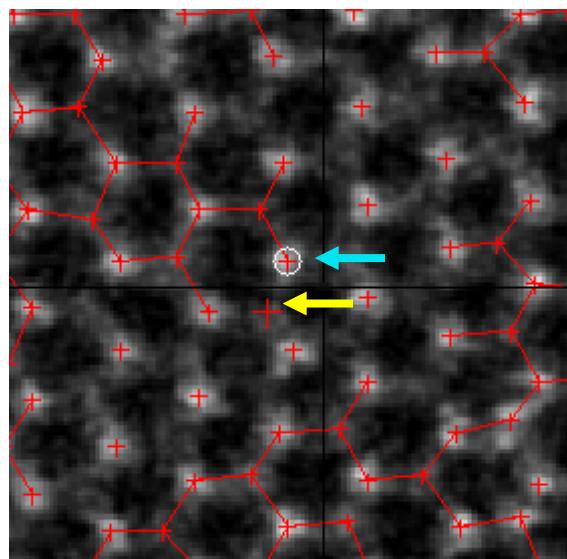
-peak finding & pair function



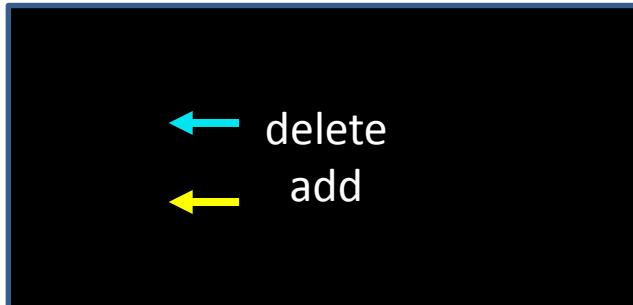
Robust and efficient peak finding



Pair correlation function

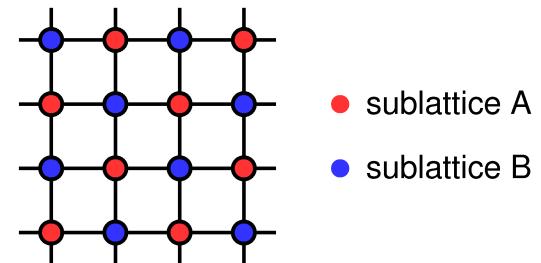
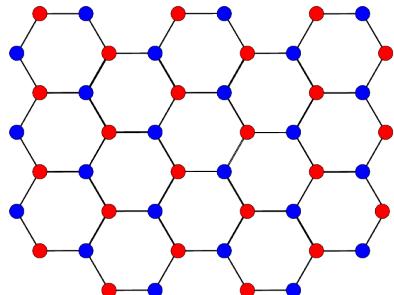
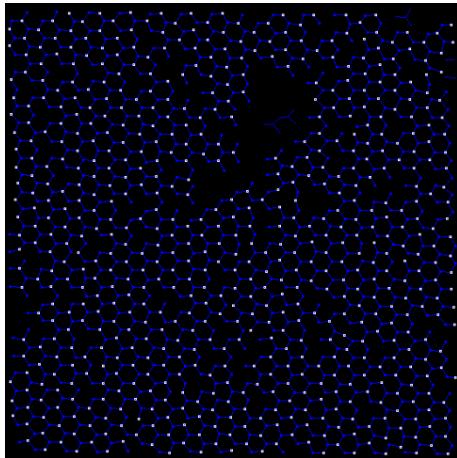


-manually add and delete atoms



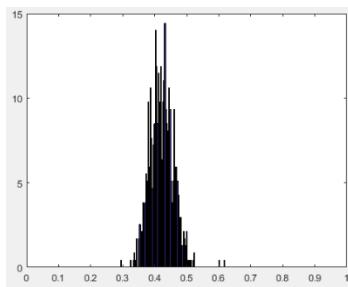
Features

-atomic network with alternating sites



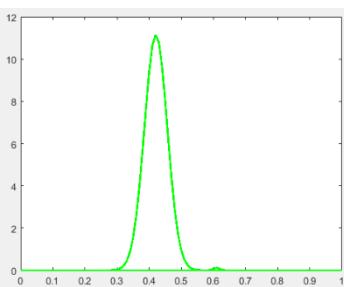
- sublattice A
- sublattice B

-atom intensity histogram with Gaussian mixture model

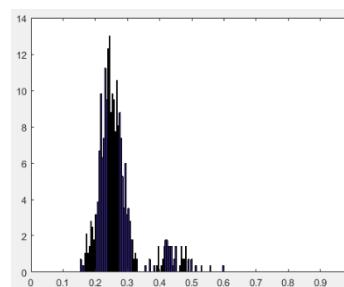


Gaussian mixture distribution with 2 components in 1 dimensions
Component 1:
Mixing proportion: 0.002766
Mean: 0.6104

Component 2:
Mixing proportion: 0.997234
Mean: 0.4209

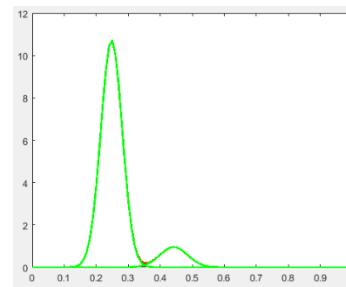


Vacancy concentration & doping concentration



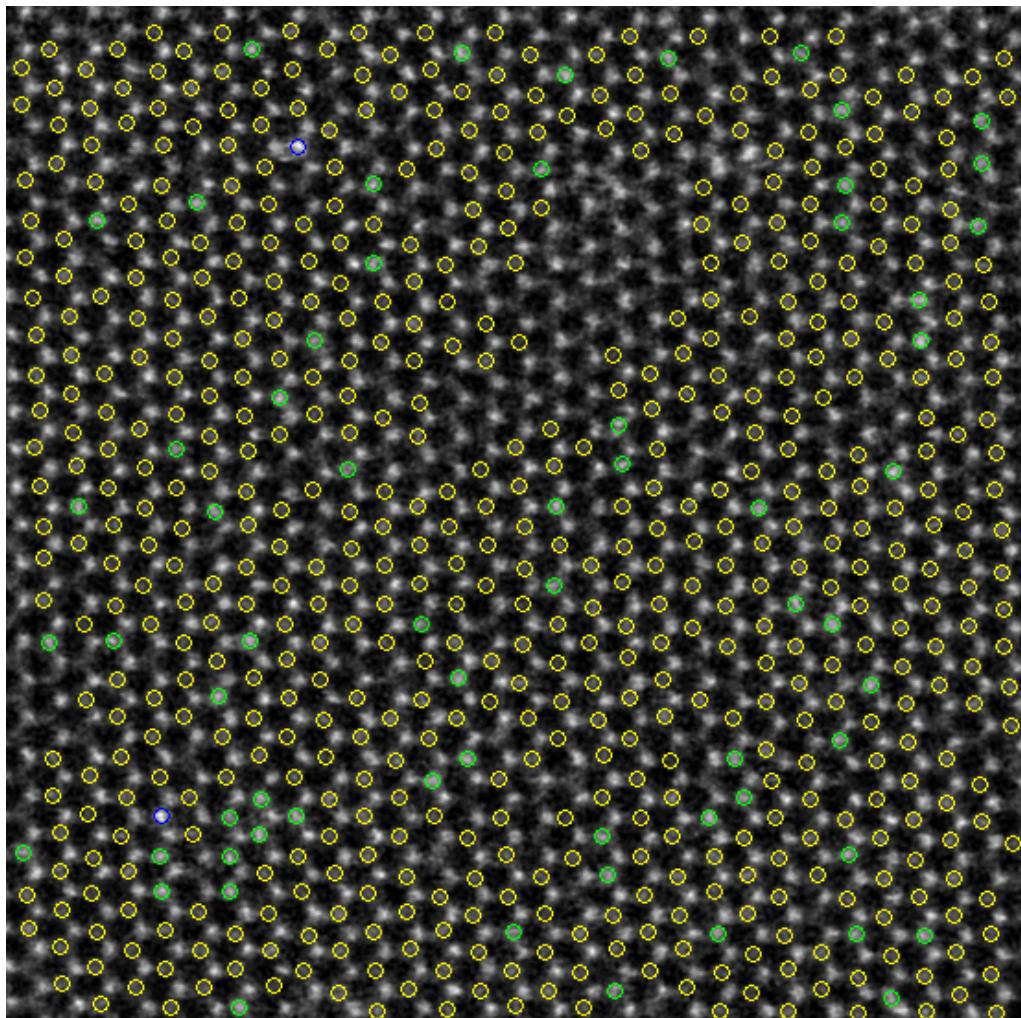
Gaussian mixture distribution with 2 components in 1 dimensions
Component 1:
Mixing proportion: 0.896705
Mean: 0.2480

Component 2:
Mixing proportion: 0.103295
Mean: 0.4421



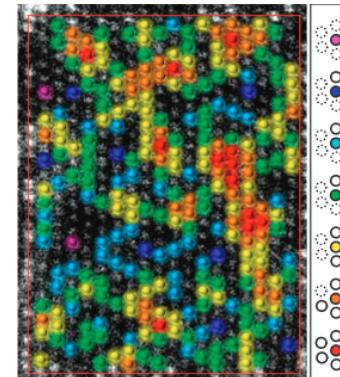
Features

-determining atom type by both intensity and atom site & alloy degree



Visualization of different atoms

Calculating alloy degree

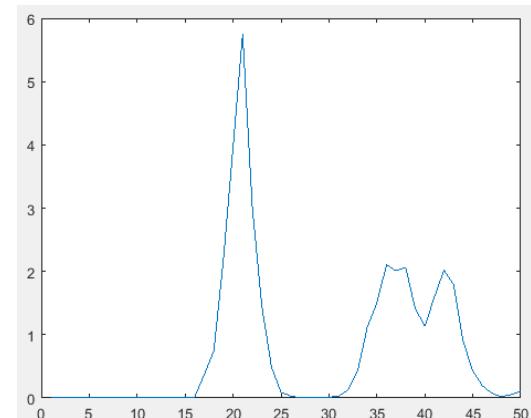


$$P_{\text{observed}} = \frac{\sum_{i=0}^6 (i \times N_{W-iMo})}{6 \times N_W},$$

$$P_{\text{random}} = \frac{N_{Mo}}{N_W + N_{Mo}},$$

$$J_W = \frac{P_{\text{observed}}}{P_{\text{random}}} \times 100\%$$

Dumcenco, Dumitru O., et al. *Nature communications* 4 (2013): 1351.

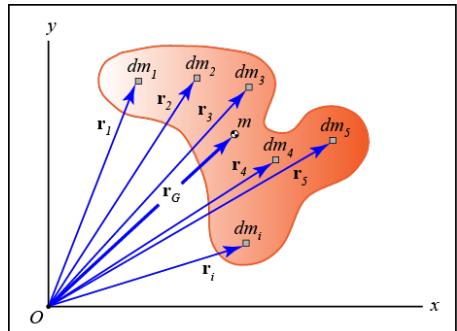


Pair function for sublattice

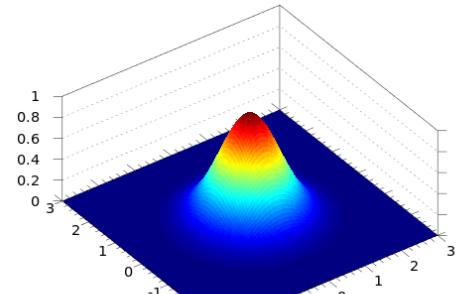
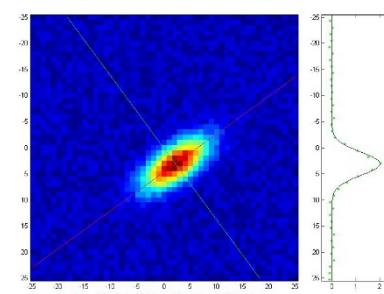
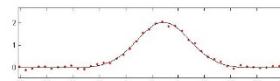
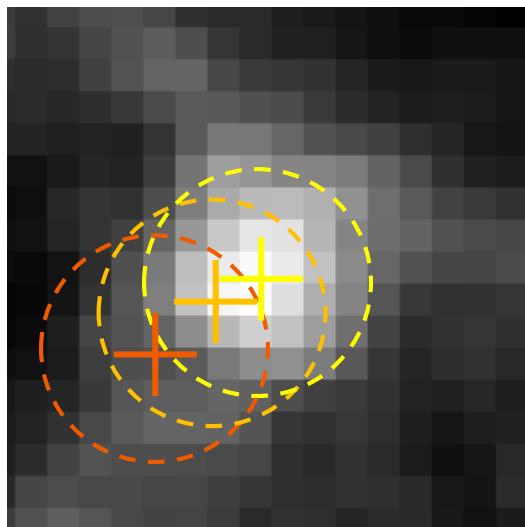
Features

-position relaxation

Iteration through finding mass center



$$x_{cm} = \frac{\sum_{i=1}^N m_i x_i}{M} \quad y_{cm} = \frac{\sum_{i=1}^N m_i y_i}{M}$$



$$f(x, y) = A \exp \left(- \left(\frac{(x - x_o)^2}{2\sigma_x^2} + \frac{(y - y_o)^2}{2\sigma_y^2} \right) \right)$$

-save and reload data

All data saved in ***.mat automatically after you finish the last subroutine---'alloy_degree.m'

 [data26-Dec-2016 135547.mat](#)

To retrieve the data you saved, first run the subroutine---'reload_data.m', then click your ***.mat

Contents

- **What? -features**
- **How? -workflow**
 - overview
 - step by step
- **Why? -principles**

Workflow

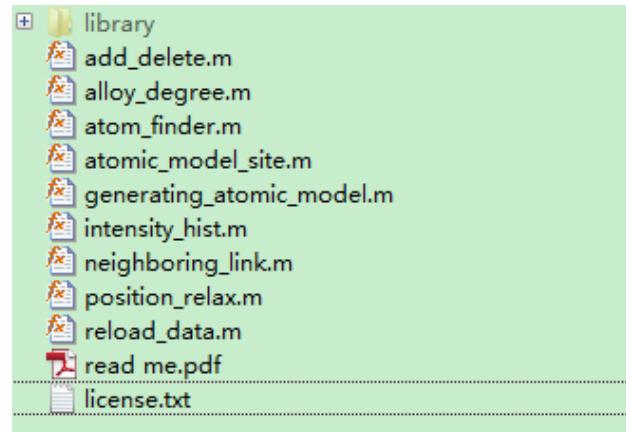
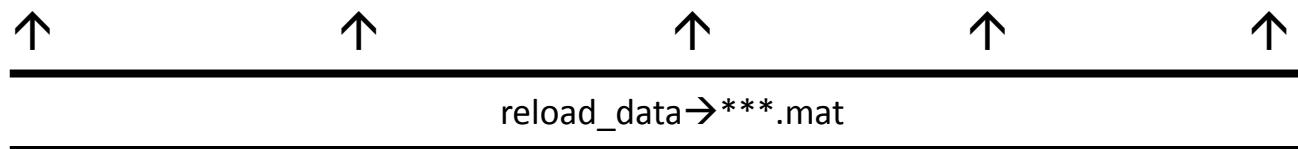
-overview

Requirements: MATLAB R2016b(a)

Files under the atom_catcher folder

add_delete(optional) Position_relax(optional)

↓ ↑ ↑ ↓
atom_finder → neighboring_link → generating_atomic_model → intensity_hist → atomic_model_site → alloy_degree



To run ***.m file, open the corresponding file and press

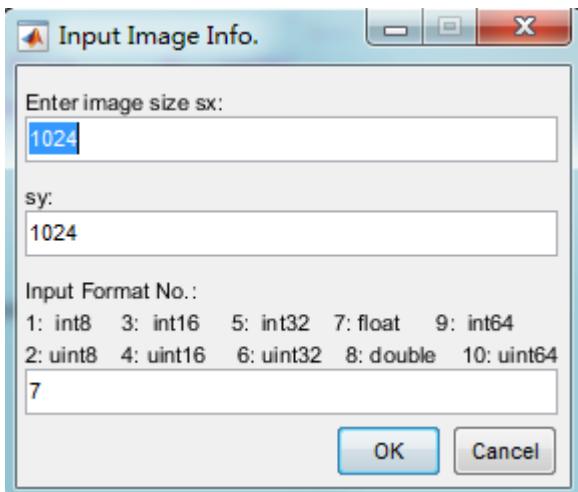


in the tool bar

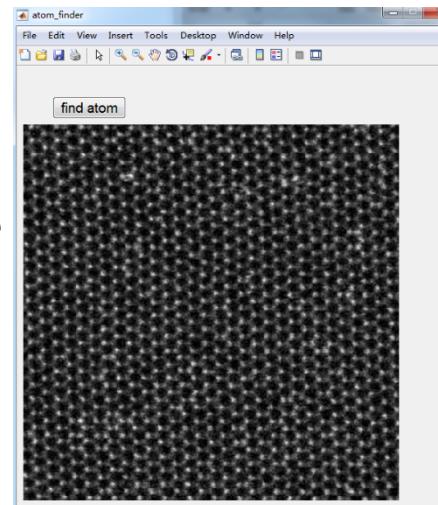
Workflow

-step by step

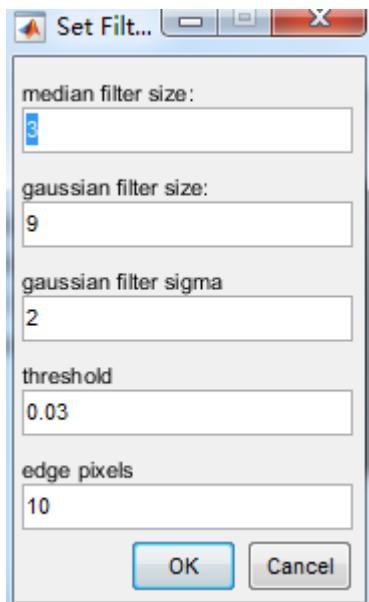
Subroutine1 atom_finder



Choose a ***.dat file to open; specify the size of the image and data type of the pixel value



Click 'find atom' to continue



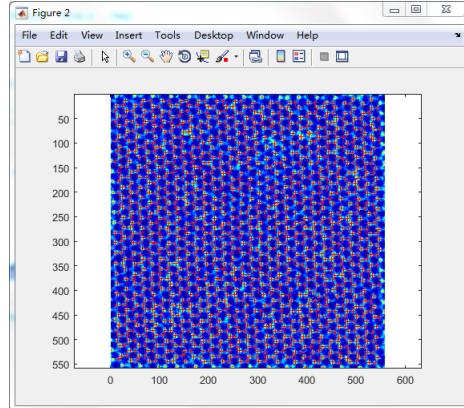
Two filters applied before peak finding(local maxima).

1. median filter: to remove 'salt and pepper' noise; small → more misassigned peaks;-integer number(3 pixels).
2. gaussian filter: to smooth noisy peaks; filter size-integer number(9 pixels); small sigma → less smoothing → more misassigned peaks,-real positive number(2 pixels).
3. Threshold: small threshold → more misassigned peak,-real number 0~1(0.03).
4. Edge pixels: discard edge pixels, integer number(10 pixels)

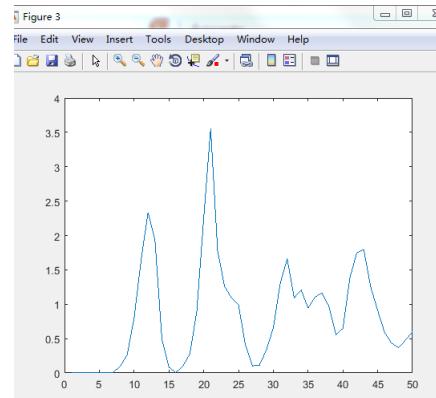
Workflow

-step by step

Subroutine1 atom_finder



Display the peaks found automatically

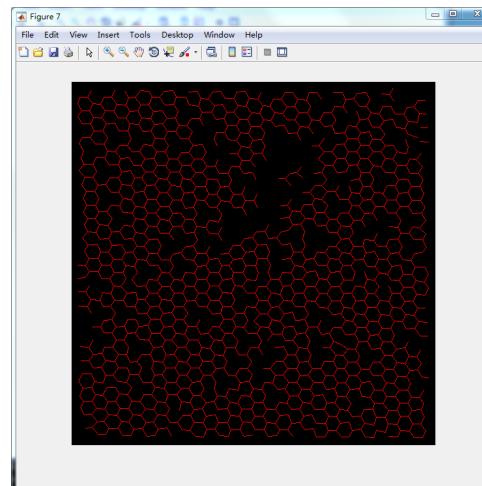


Corresponding pair function, the first peak indicates the nearest atomic bonding length

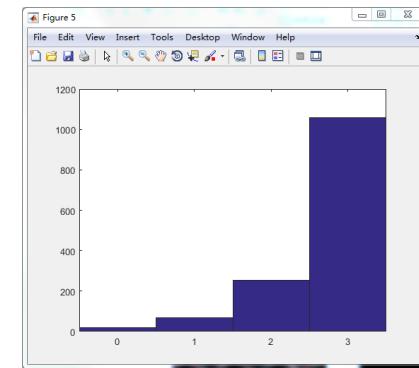
Subroutine2 neighboring_link

```
function []=neighboring_link(varargin)
global X;
global d;
global h2;
global h4;
global G;
if nargin<1)
    %% value setting
    boundary=10;%cutoff value,
    %where the first peak ends according to the pair function
    low=10;high=15;%restriction of bonding between neighboring pairs,
    %the range of the first peak according to the pair function
    %%
end
```

Value setting: according to pair function, before running



Display bondings between neighboring atoms

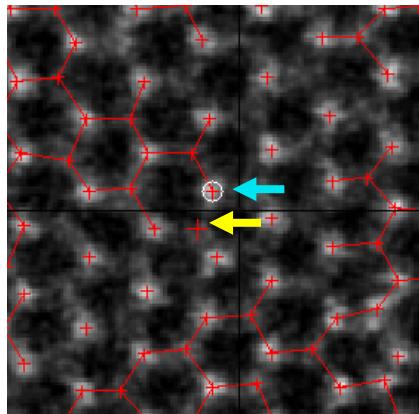


Display frequency of atoms VS neighboring atom number

Workflow

-step by step

Subroutine3 add_delete

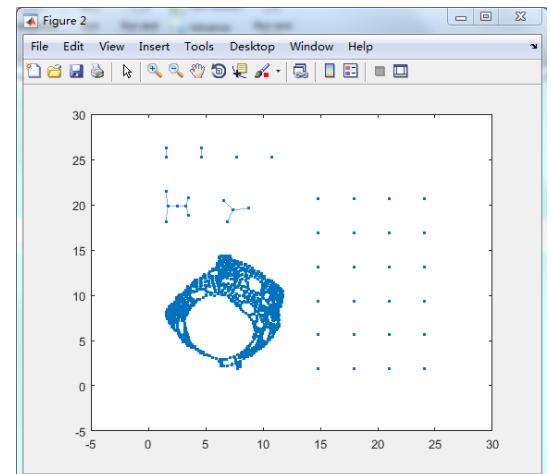
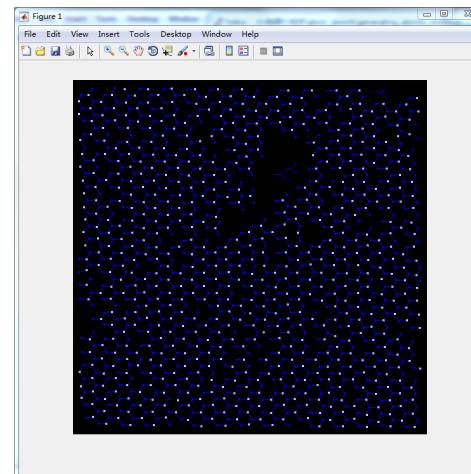


1. Left mouse click- add missing atom
2. Right mouse click- delete misassigned atom
3. Middle mouse click- exit

Subroutine4 generating_atomic_model

```
function []=generating_atomic_model()
global G;
global X;
global d;
global h3
global group;

%% value setting
initial=10;%specify the initial point,
%check cluster in the workspace;default is 1
%%
```



Value setting: random number 0~total point number, before running



Check connectivity of the lattice, graph representation of the lattice model(right figure)

Workflow

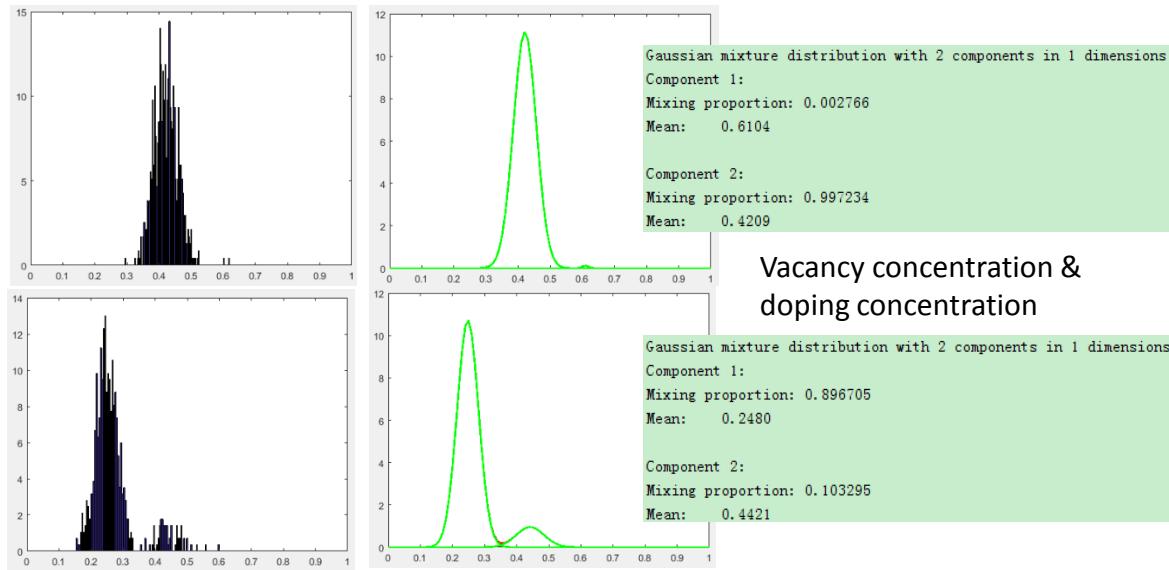
-step by step

Subroutine5 intensity_hist

```
function []=intensity_hist()
global group;
global d;
global X;
global r;

%% value setting
r=4;%radius of the area averaged specified/pixel
%%
```

Value setting: atom radius according to image, before running

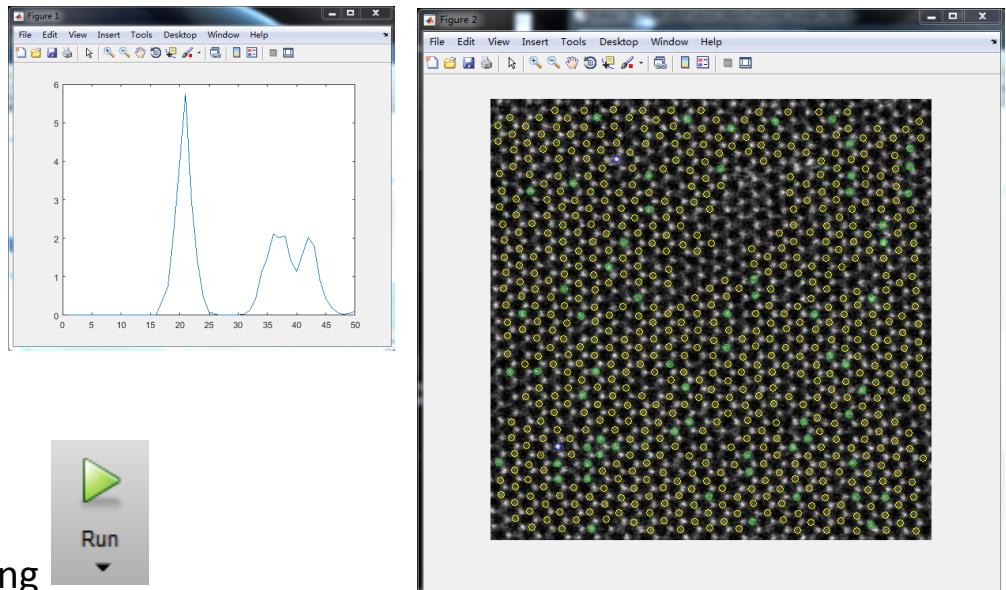


Vacancy concentration & doping concentration

Subroutine6 atomic_model_site

```
function []=atomic_model_site()
global X;
global d;
global h5;
global group;
global r;
global type_count
Sub=[];
%% value setting
j=2;%specify the site you're interested in,
%switch between site 1 and 2;
seperation=[0 0.12 0.34 0.55 1];%specify the intensity range
%for the different atoms on this site(including vacancy),
%here 4 types on this site;
%%
```

Value setting: specify the site(1 or 2) you are concerned; atom classification according to intensity from the last histogram, before running



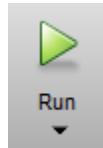
Workflow

-step by step

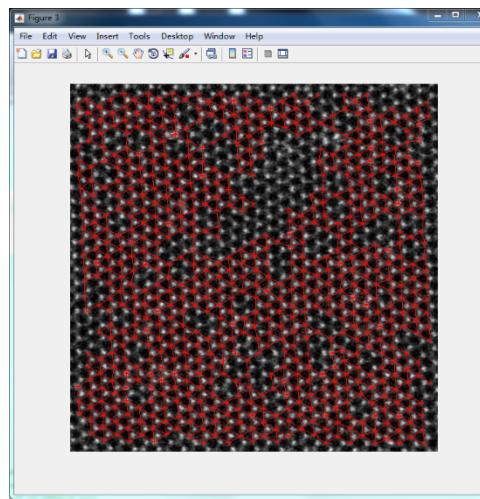
Subroutine7 alloy_degree

```
function []=alloy_degree()
global type_count

evalin('base',' save(''Sub'', ''Sub'')');
Sub=load('Sub.mat');
S=Sub.Sub;
%% value setting
neighboring_link(S,26,16,25);%restriction of forming links
%between neighboring pairs,
%boundary, low and high respectively
%to generate the network H for further processings
%%
```



Value setting: same as subroutine2 according to the pair function of sublattice, before running



```
>> alloy_degree
J23 =
0.8863
J32 =
0.9053
```

Display network of sublattice Alloy degree



Automatically saved ***.mat file including all data

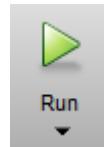
Subroutine8 position_relax

```
function []=position_relax()
global X;
global d;
global r;
%% value setting
selection=2;%selection:1 method massive center;
%selection:2 method gaussian fitting.
%%
```



Value setting: selection of position relax method(1 or 2), before running

Subroutine9 reload_data



Run and then load ***.mat

Contents

- **What? -features**
- **How? -workflow**
- **Why? –principles**

-filters in image processing

-kd tree data structure

-pair correlation function

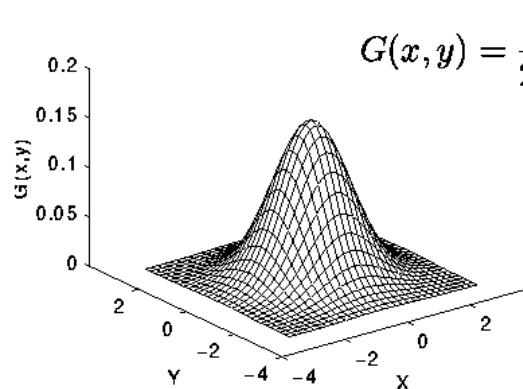
-breadth first search and graph

-Gaussian mixture model

Principles

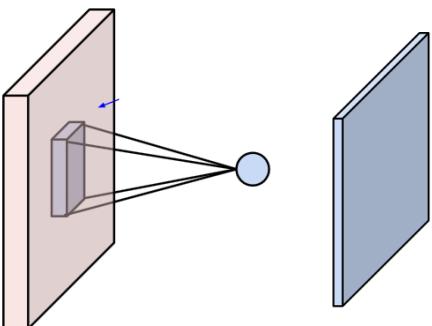
- filters in image processing

Gaussian filter/smoothing/convolution



$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1



convolution

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau \\ = \int_{-\infty}^{\infty} f(t - \tau) g(\tau) d\tau.$$

<http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>

<http://cs231n.github.io/convolutional-networks/>

Median filter

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

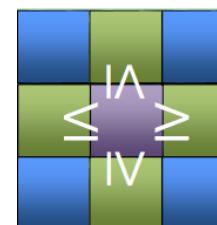
Neighbourhood values:

115, 119, 120, 123, 124,
125, 126, 127, 150

Median value: 124

<http://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm>

Local maxima search

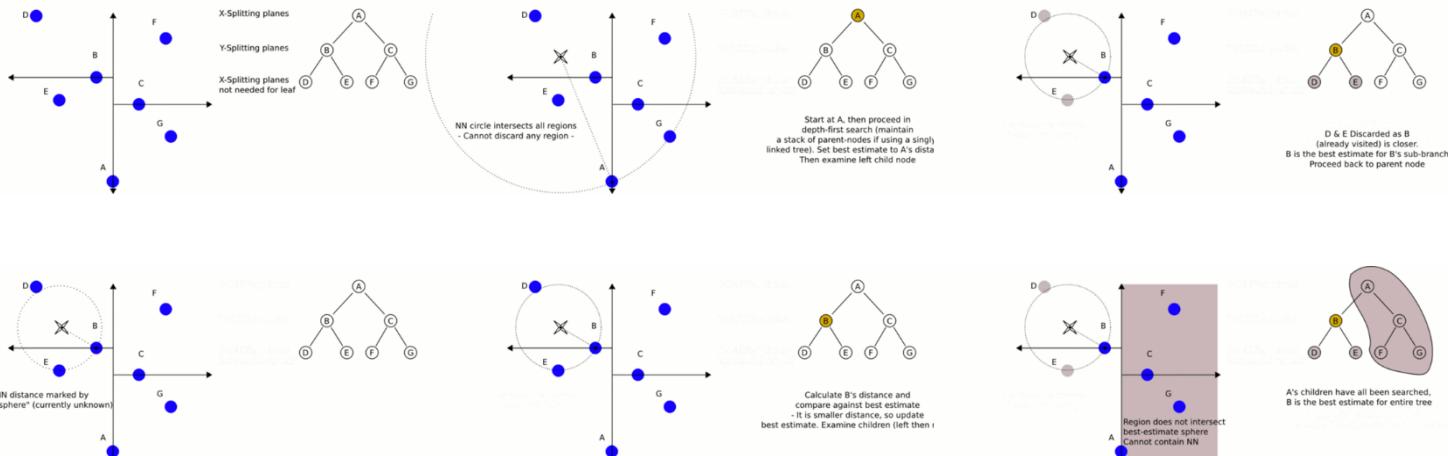
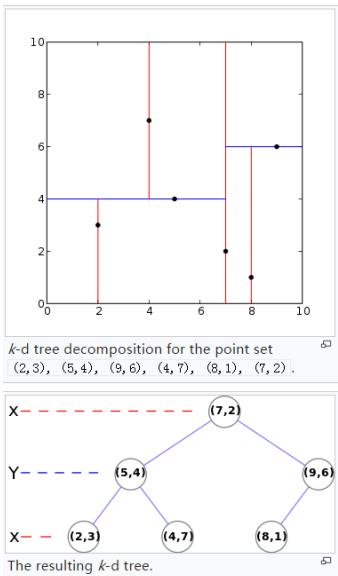


9	3	5	2	4	9	8
7	2	5	1	4	0	3
9	8	9	3	2	4	8
7	6	3	1	3	2	3
9	0	6	0	4	6	4
8	9	8	0	5	3	0
2	1	2	1	1	1	1

<https://courses.csail.mit.edu/6.006/spring11/lectures/lec02.pdf>

Principles

- kd tree data structure



https://en.wikipedia.org/wiki/K-d_tree

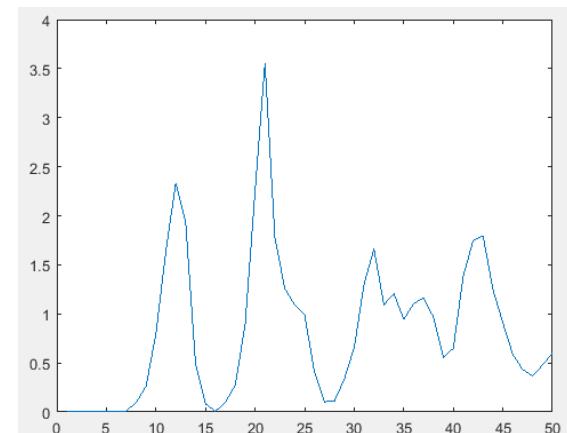
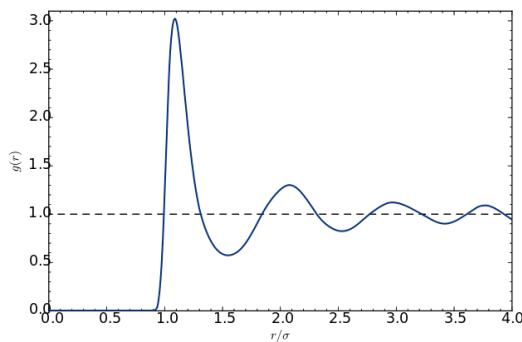
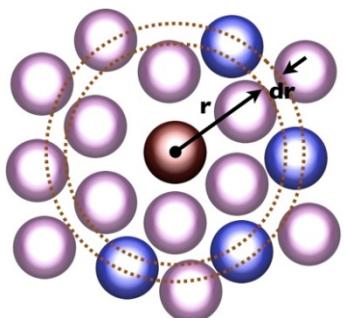
- pair correlation function

Particle density

$$\rho = N/V$$

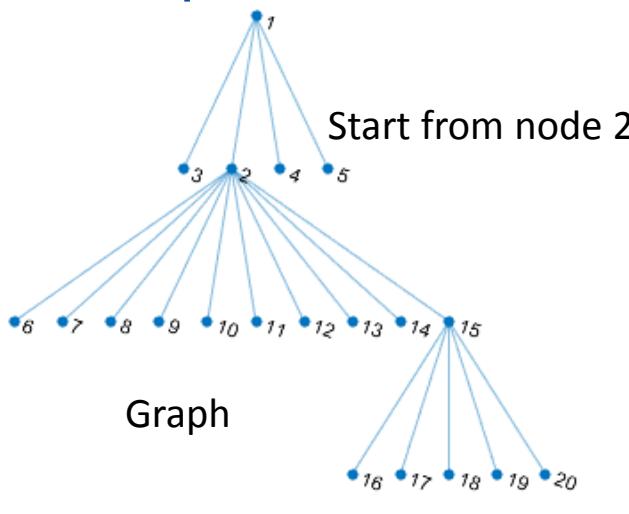
Normalization for sphere shell

$$g(r)_I = 4\pi r^2 \rho dr$$



https://en.wikipedia.org/wiki/Radial_distribution_function

Principles



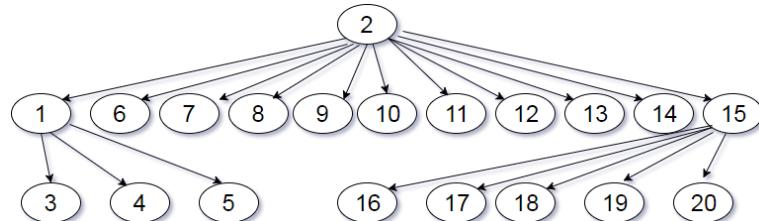
MATLAB Documentation

```
v = bfsearch(G,2)
```

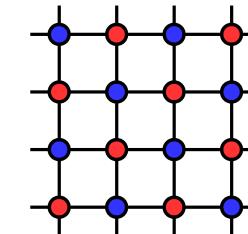
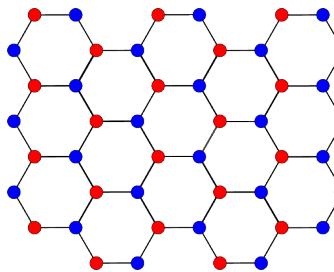
v =

2
1
6
7
8
9
10
11
12
13
14
15
3
4
5
16
17
18
19
20

- breadth first search and graph

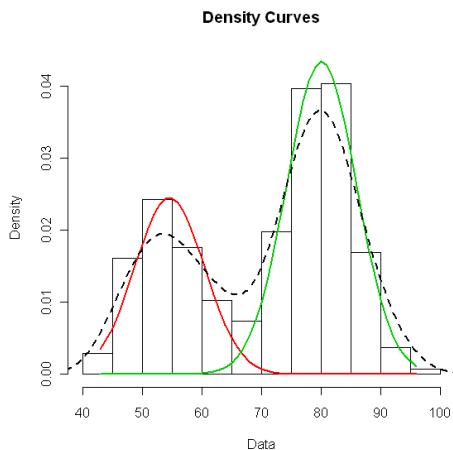


Order in which the nodes are expanded

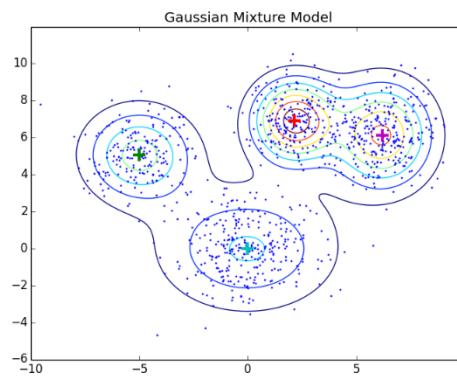


- sublattice A
- sublattice B

- gaussian mixture model



1D dataset



2D dataset

$$f_{X,Y}(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp\left\{\frac{-1}{2(1-\rho^2)} \left[\frac{(x-\mu_x)^2}{\sigma_x^2} - 2\rho \frac{(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} + \frac{(y-\mu_y)^2}{\sigma_y^2} \right]\right\}$$

Linear combination

$$\begin{aligned} f &= f_1 p_1 + f_2 p_2 + f_3 p_3 + \dots \\ p_1 + p_2 + p_3 + \dots &= 1 \end{aligned}$$