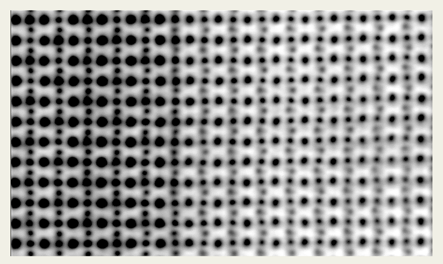
Dictionary Methods for Microscopy Image Analysis

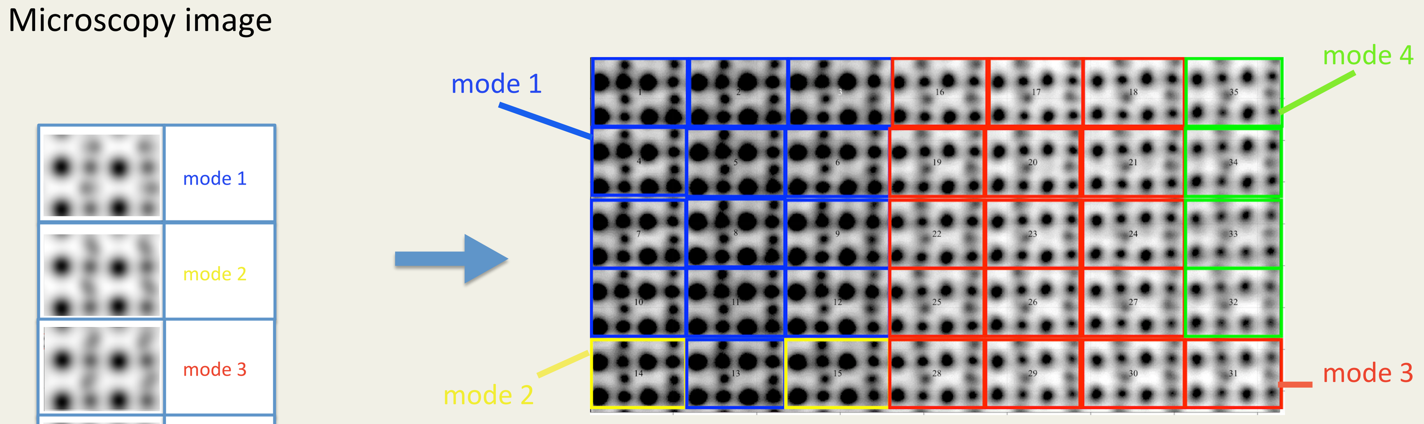
Students: Hang Cheung, Haoran Shu (CUHK)
Mentors: Dr. R. Archibald, Dr. A. Borisevich, Dr. E. D’Azevedo, Dr. K. Wong, Dr. J. Yin
Contact: henryshu1994@gmail.com, henryc9473@gmail.com

Problem Setting

Given an image of microscopy, and a set of dictionary items (of different modes of atoms), we need to design a pipeline to identify each atom in the microscopy with a mode in the dictionary.



Microscopy image



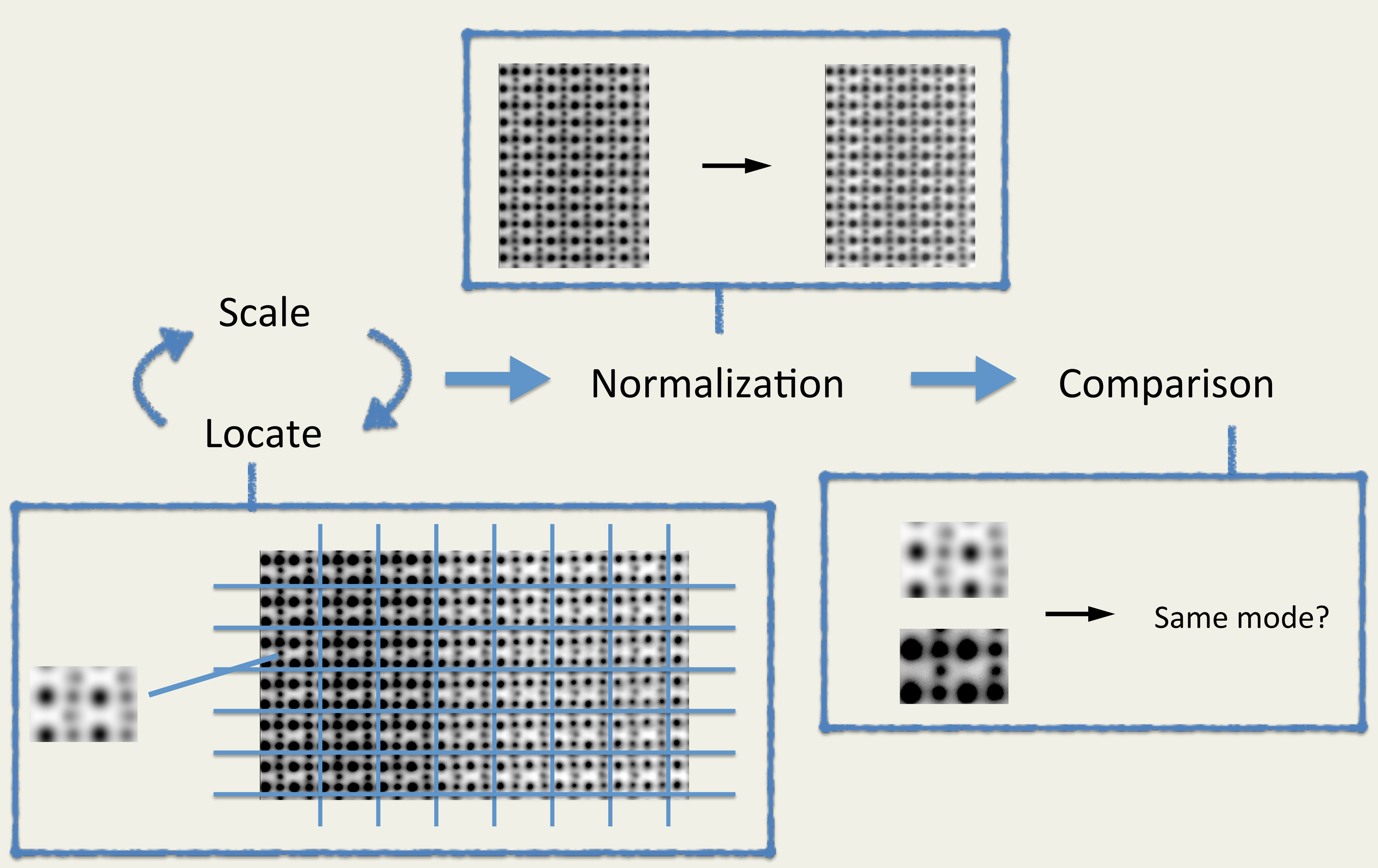
Dictionary items

Ideal Output

Workflow

The workflow of our pipeline consists of mainly three steps (as illustrated below).

Since the sizes of microscopy elements and dictionary items might be different, we first normalize the scale and slice the microscopy into pieces; then we normalize the brightness and contrast of microscopy elements with dictionary items before we compare them to gain optimal matches.

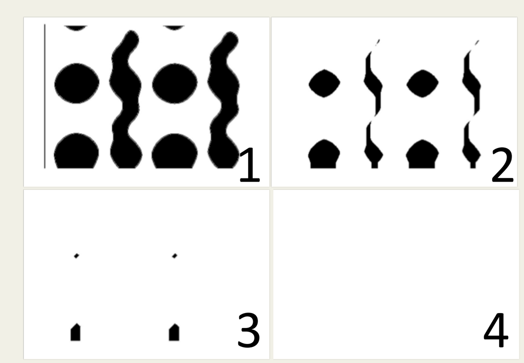


Algorithm

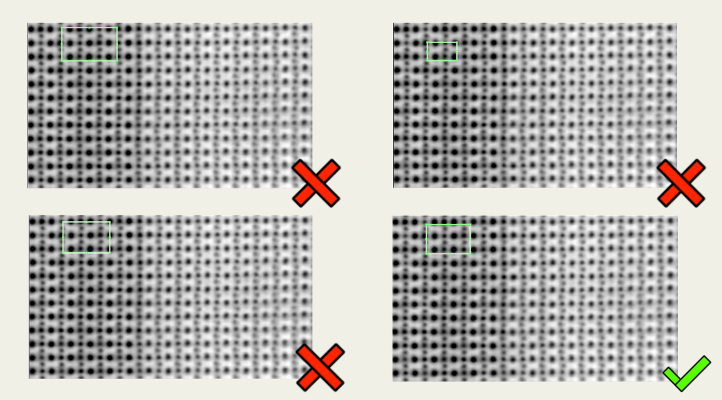
We assume that the elements in a microscopy are evenly and periodically distributed (as they usually should be) and that the atom cores are irrelevant since their shape and size are mode-invariant.

Scale + Locate

‘Eating from outside’ strategy to get an approximate scale ratio by comparing the time needed for both images to fade out

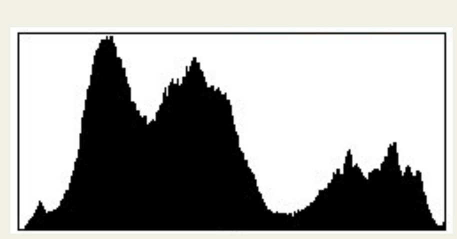


Adopt a brute force search by varying around the approximate scale ratio and scan through

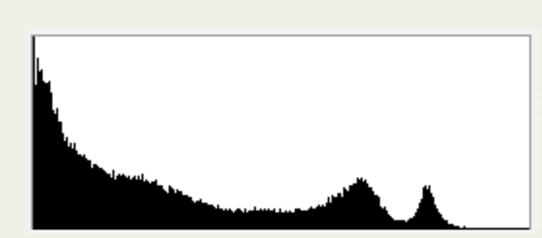


* After ‘eating from outside’, we use a brute force search to get the offset of first element

Normalization



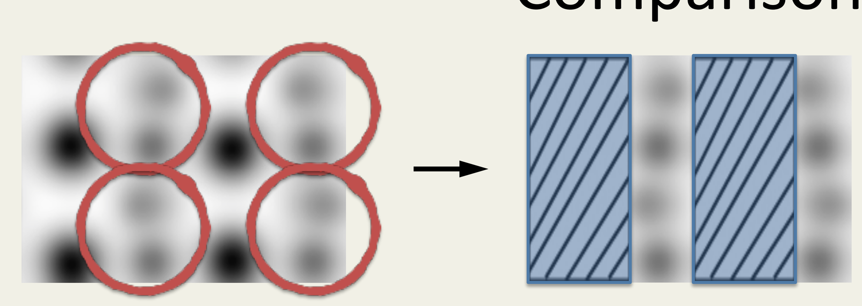
intensity distribution of microscopy
(not real distribution, figure just for illustration)



intensity distribution of dictionary
(not real distribution, figure just for illustration)

* The intensity distribution of microscopy is pixel-wisely normalized to be the same with that of the dictionary item

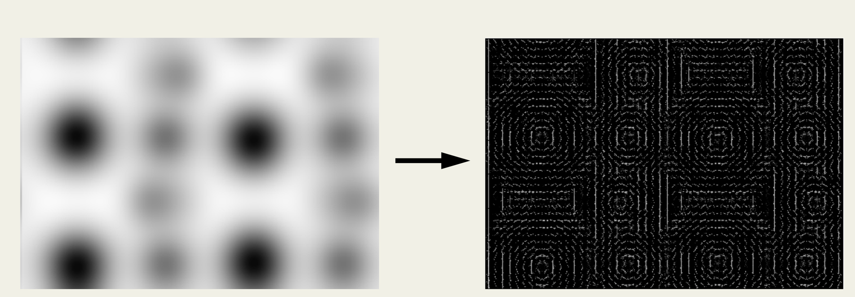
Comparison



Since we are only concerned about the subtle differences in the circled area, we use a mask to shade the cores and do not take them into consideration for comparison.

We use HOG (Histogram of Oriented Gradients) to extract features of the concerned area.

HOG:
1. divide an image into smaller patches
2. calculate the gradients at each pixel
3. generate a feature vector of gradients distribution for each small patch



$$r_{xy} = \frac{\sum_{i=1}^{n}(x(i)-\bar{x})(y(i)-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x(i)-\bar{x})^2 \sum_{i=1}^{n}(y(i)-\bar{y})^2}}$$

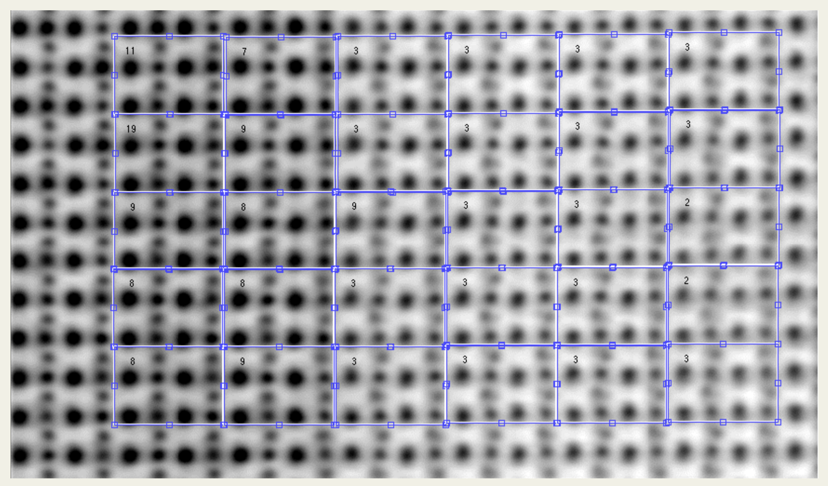
Cross Correlation

Use the cross correlation of HOG feature vectors to represent the similarity between two images.

* This metric is also used in Scale+Locate for determining better alignment

Current Outcome

Using Matlab with VLFeat API for HOG generation, we have successfully built up a working pipeline and the results are shown below.



The numbers at the corner of each unit stands for the number of dictionary item it is matched to.

On average our algorithm takes 110s to run on a MacBook Air (1.8 GHz Intel Core i5, 4 GB 1600 MHz DDR3) for a dictionary of size 25 and a microscopy of size 30 (items). A more detailed complexity analysis will be included in our final report.

Future Work

Based on current result, we plan to work on three aspects in the future: looking for a more accurate comparison method; implementing our algorithm on other platforms, say C/C++/Python; scaling our algorithm via parallelism.

1. SVM/Neural Network: we plan to improve our comparison by Machine Learning with a classifier using HOG features;

3. LSH: we plan to preprocess the dictionary using Locality Sensitive Hashing to reduce comparing time to sub-linear.

Acknowledgements & Reference

Thanks to University of Tennessee Knoxville, JICS, and Oak Ridge National Lab and the Chinese University of Hong Kong.

Reference :
1 : P. F. Felzenszwalb, R. B. Grishick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. PAMI, 2009.

2 : A. Vedaldi and B. Fulkerson. VLFeat Library.
http://www.vlfeat.org/

3 : A. Ng(2016), Machine Learning on Coursera, Retrieved from
https://www.coursera.org/learn/machine-learning

4 : Jisung Yoo, Sung Soo Hwang, Seong Dae Kim, Myung Seok Ki, Jihun ChaCorrigendum to “Scale-invariant template matching using histogram of dominant gradients” [Pattern Recognit. 47/9 (2014) 3006–3018]Pattern Recognition, Volume 47, Issue 12, December 2014, Page 3980

5 : Fatih Porikli, "Integral Histogram: A Fast Way To Extract Histograms in Cartesian Spaces", CVPR, 2005, Proceedings. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Proceedings. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2005, pp. 829-836, doi:10.1109/CVPR.2005.188

6 : J.P. Lewis, Fast Template Matching, Vision Interface 95, Canadian Image Processing and Pattern Recognition Society, Quebec City, Canada, May 15-19, 1995, p. 120-123.

7 : S. Korman, D. Reichman, G. Tsur, S. Avidan, "FAsT-Match: Fast Affine Template Matching”, CVPR 2013, Portland

8: K. Briechle, U. D. Hanebeck, “Template matching using fast normalized cross correlation”, SPIE 4387, Optical Pattern Recognition XII, (20 March 2001); doi: 10.1117/12.421129

9 : E.H. Andelson and C.H. Anderson and J.R. Bergen and P.J. Burt and J.M. Ogden. "Pyramid methods in image processing"

10 : Summed area table. (n.d.). Retrieved July 11, 2016, from https://en.wikipedia.org/wiki/Summed_area_table