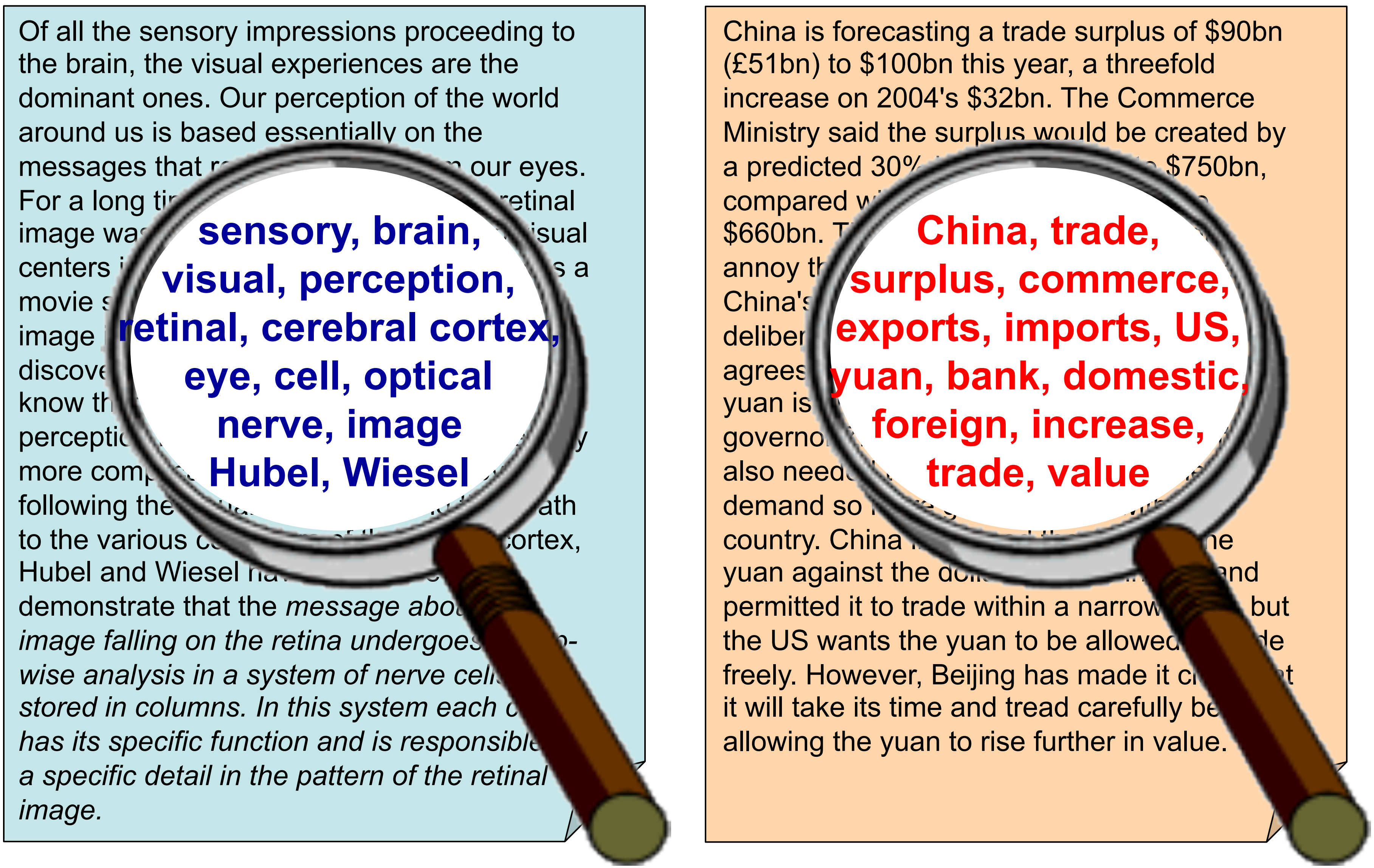
Part 1: Bag-of-words models

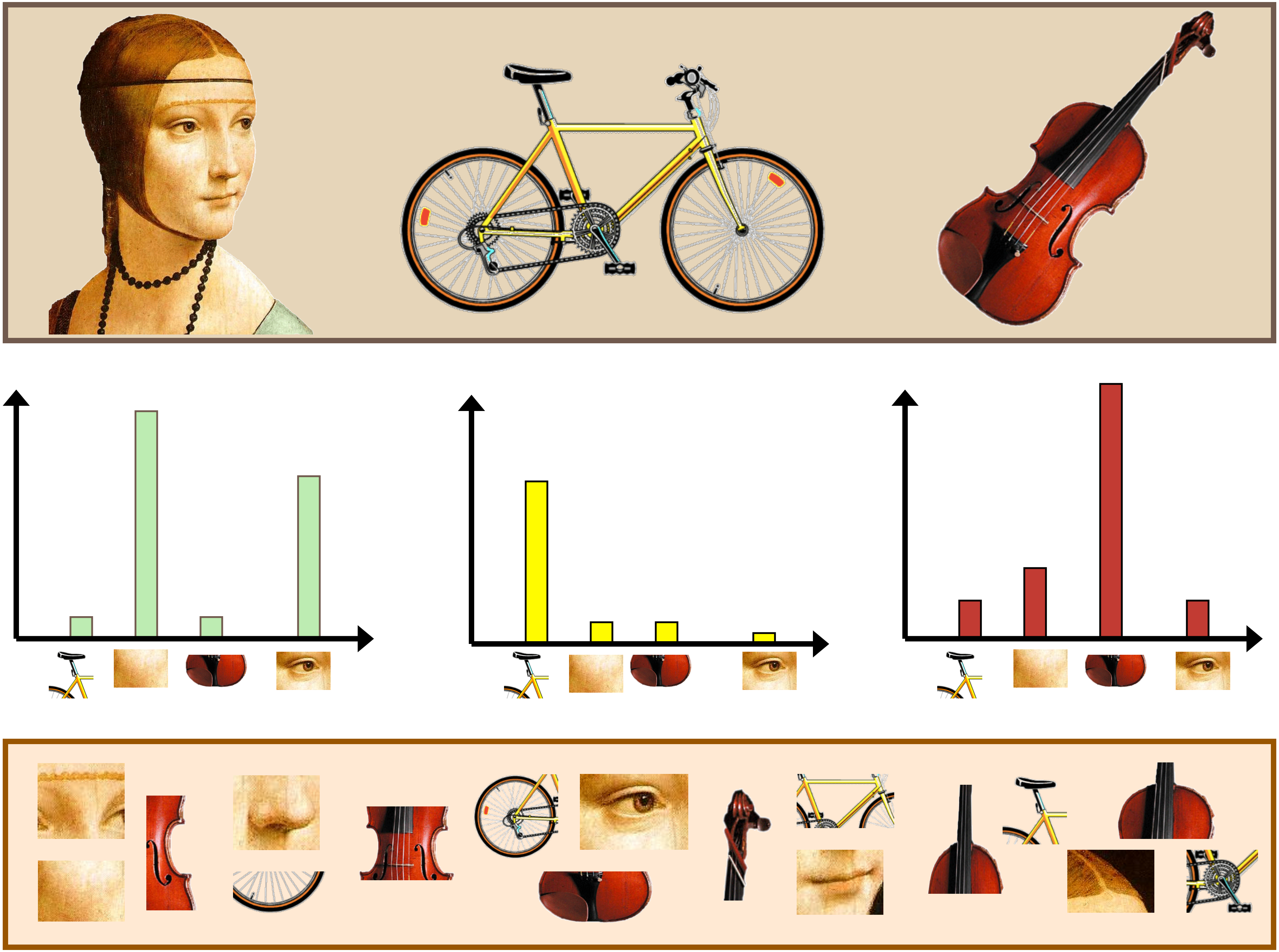
by Li Fei-Fei (UIUC)

Object

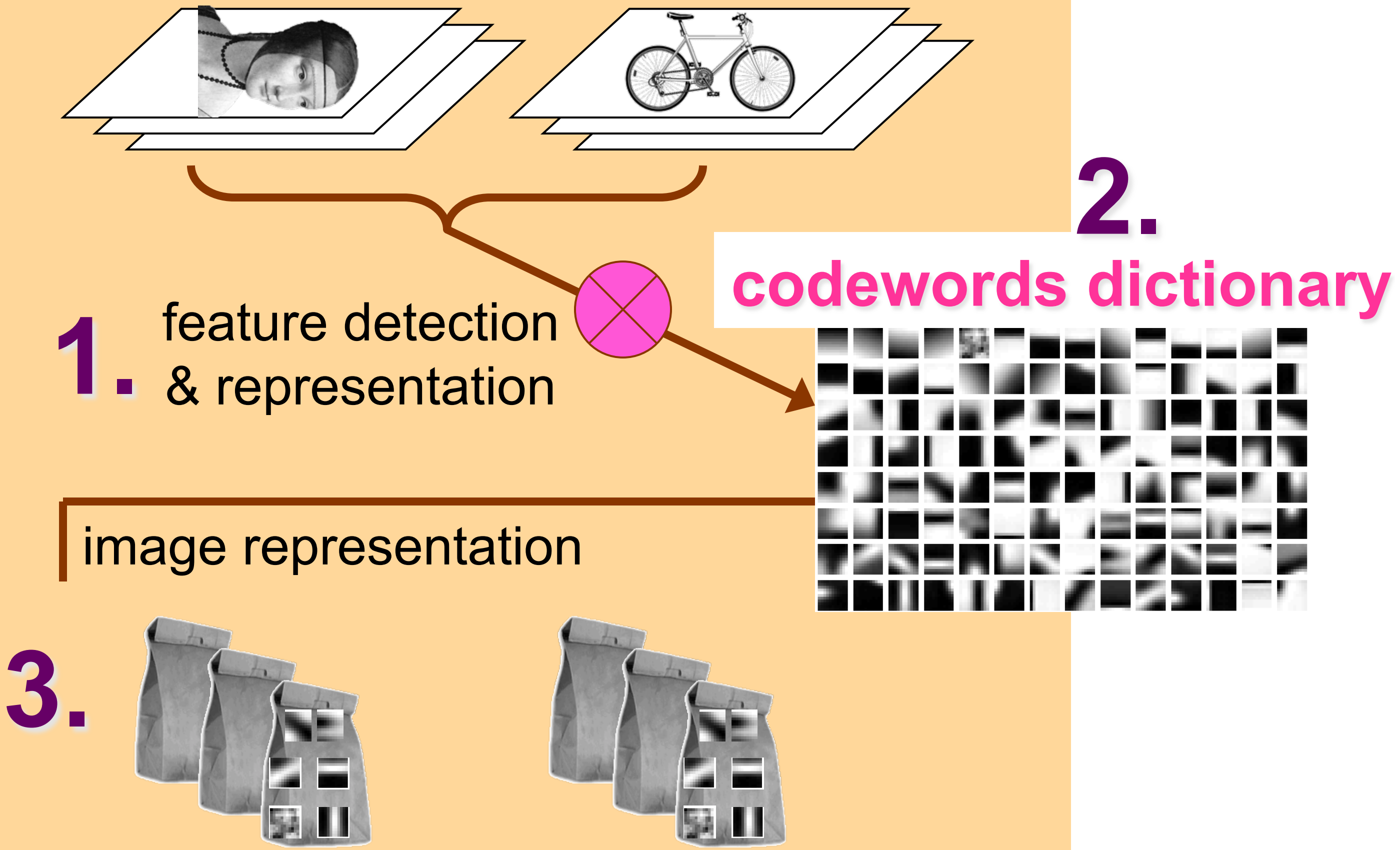
Bag of ‘words’

Analogy to documents

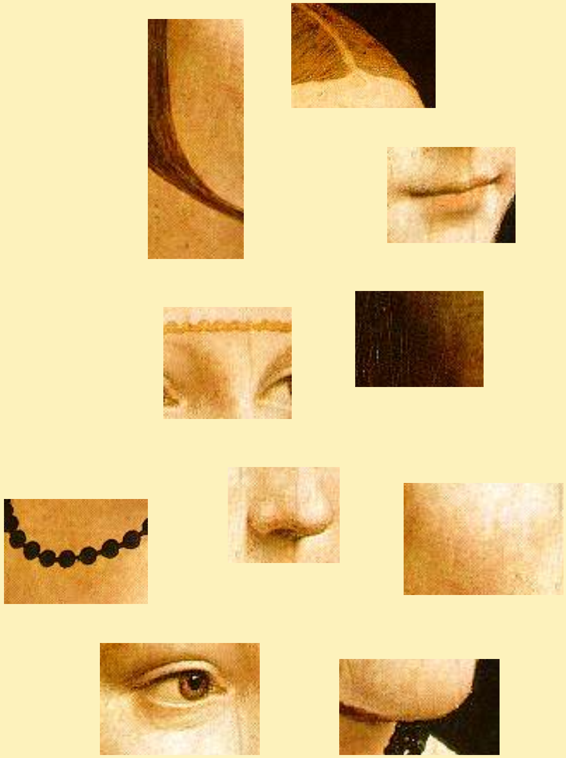




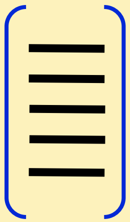
Representation



1.Feature detection and representation

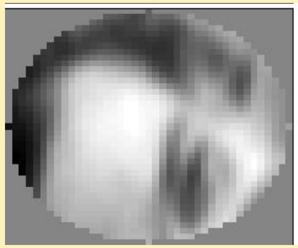


1.Feature detection and representation

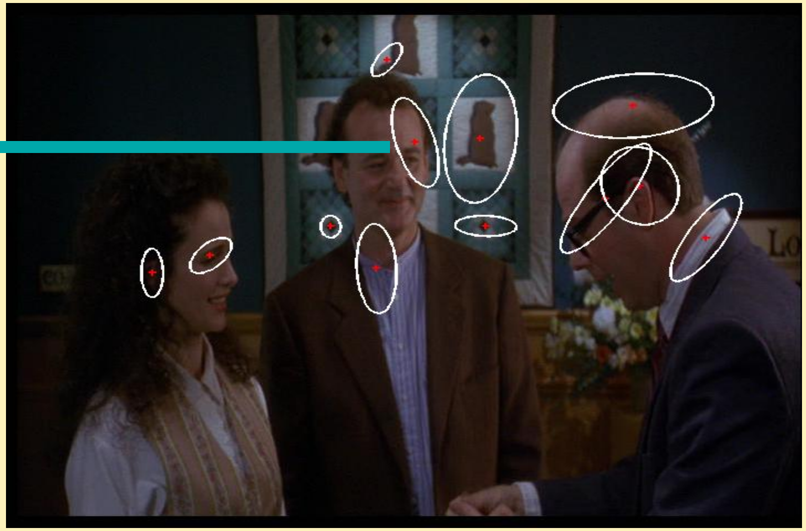


Compute SIFT descriptor

[Lowe'99]



Normalize patch



Detect patches

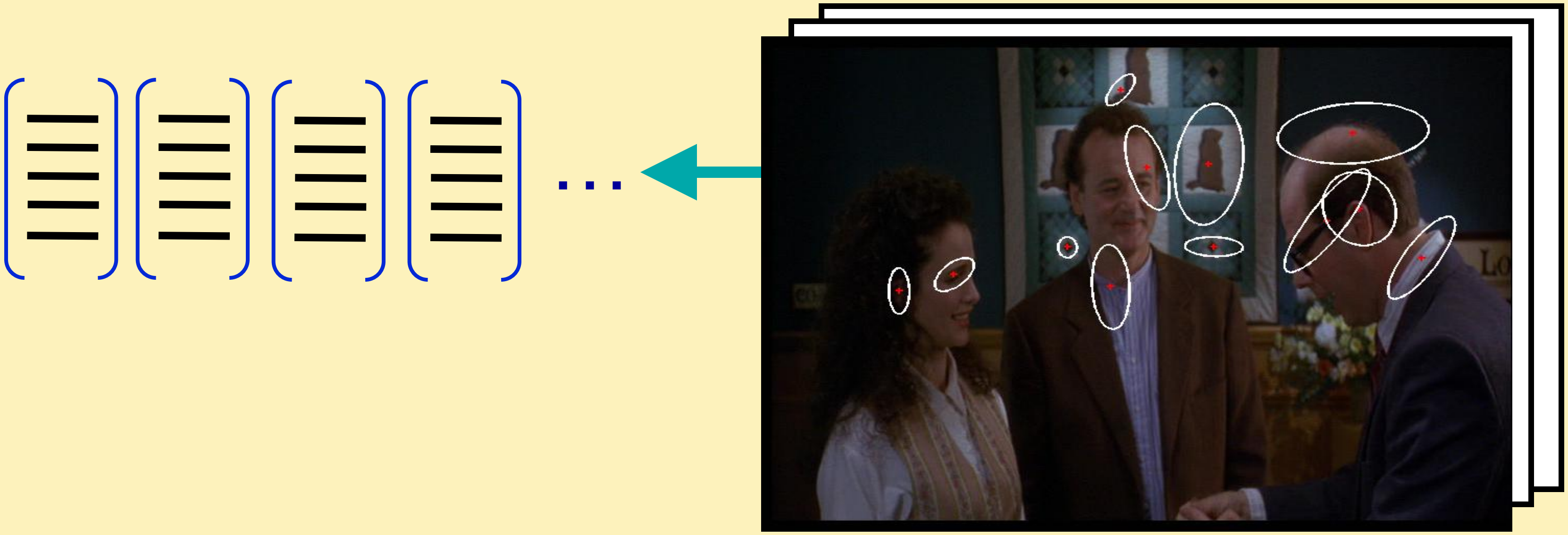
[Mikojaczyk and Schmid '02]

[Matas et al. '02]

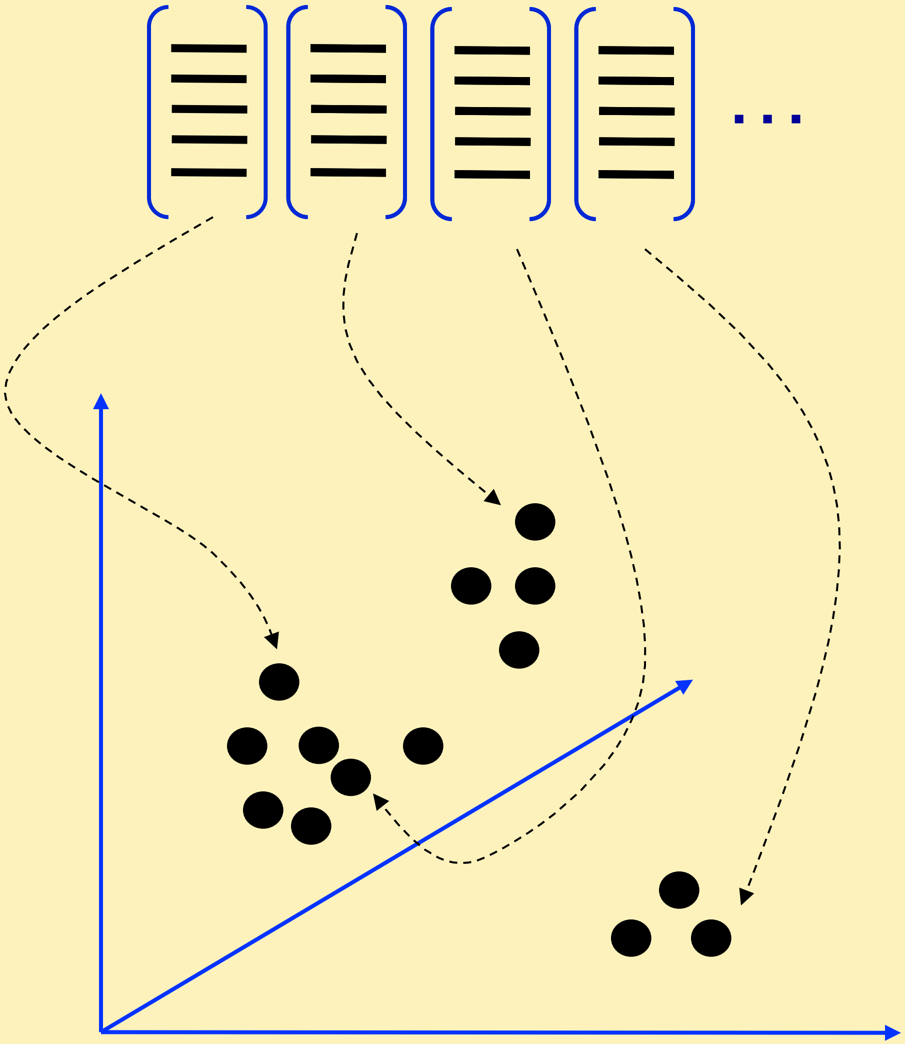
[Sivic et al. '03]

Slide credit: Josef Sivic

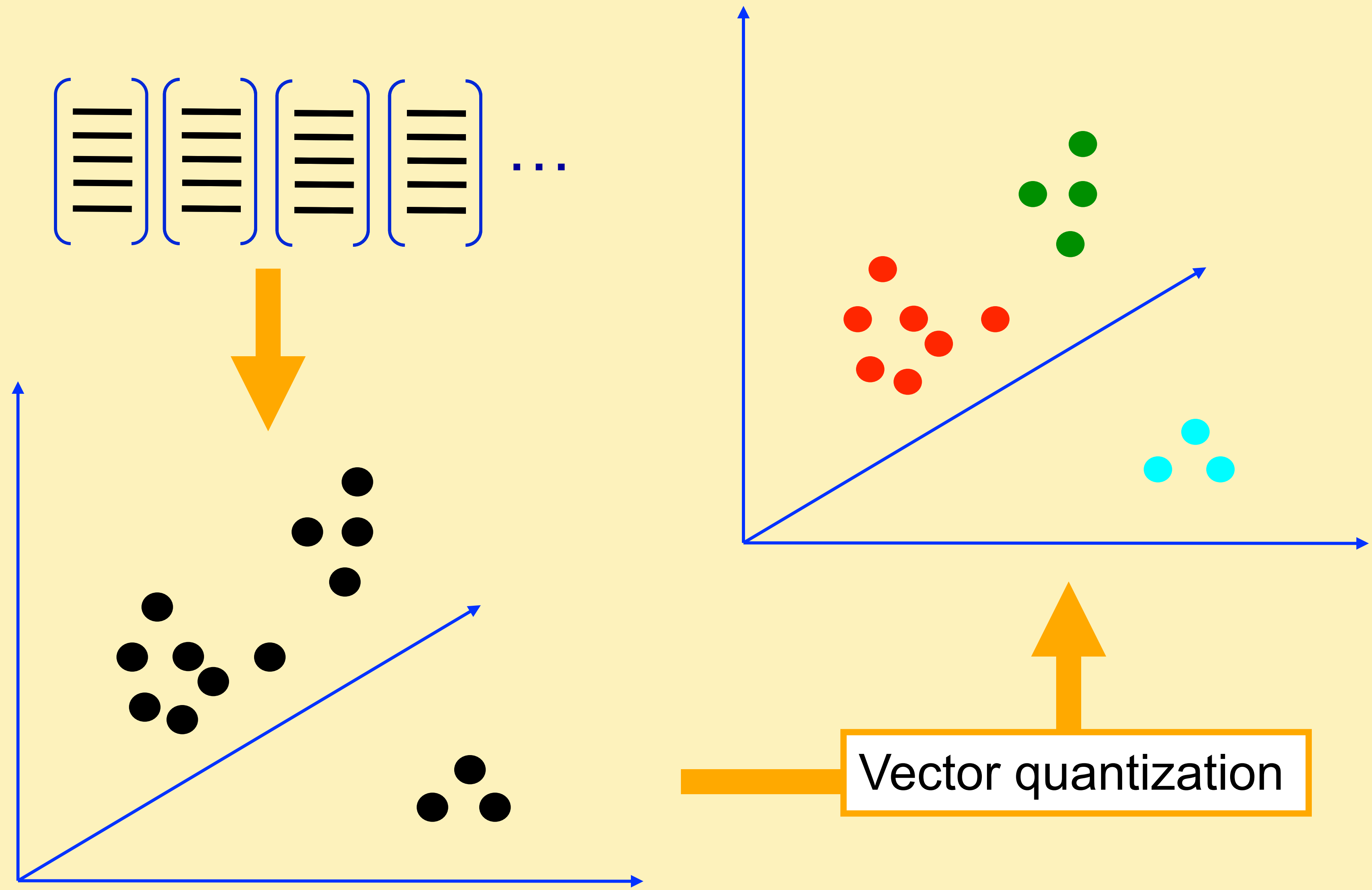
1.Feature detection and representation



2. Codewords dictionary formation

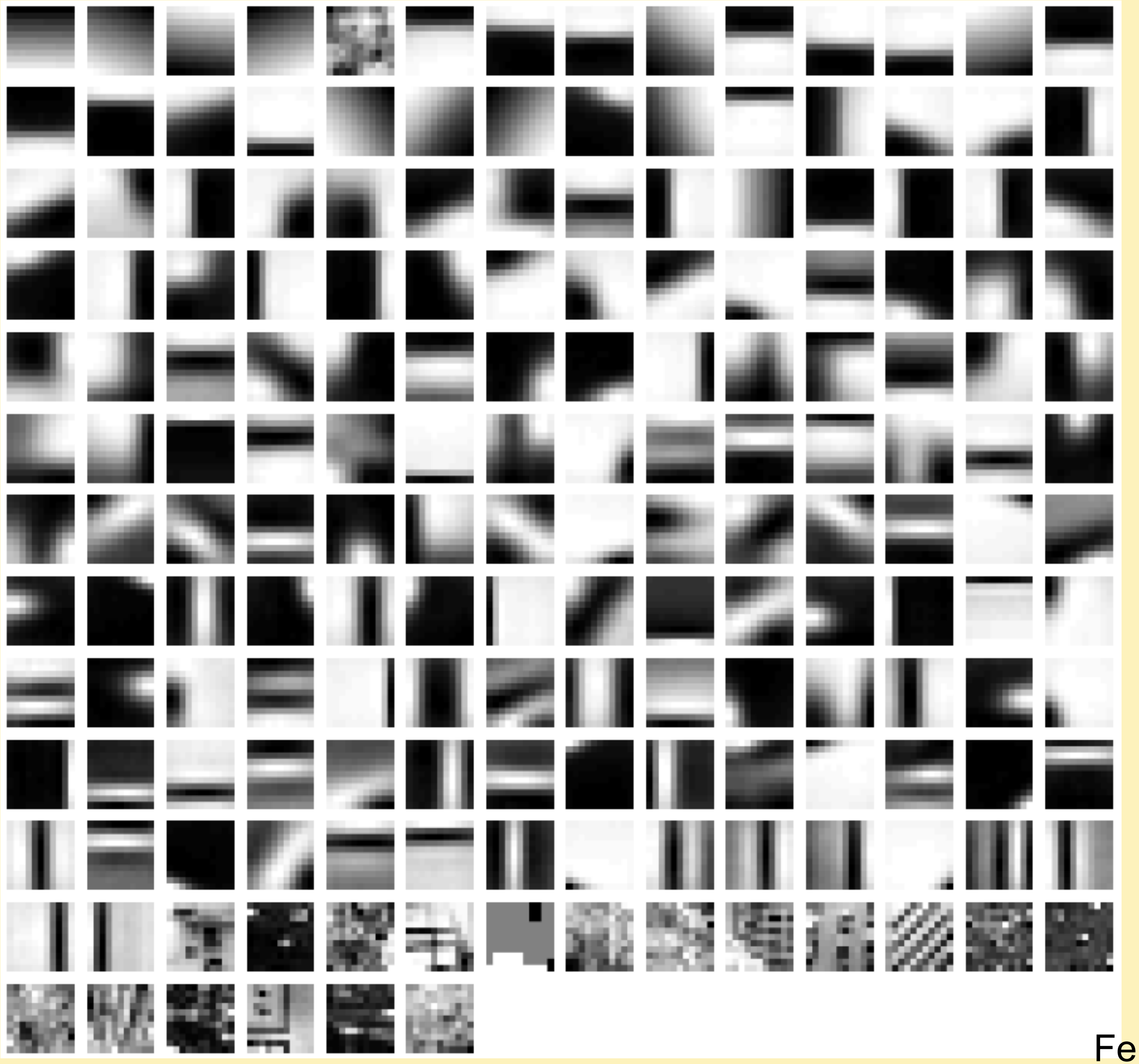


2. Codewords dictionary formation



Slide credit: Josef Sivic

2. Codewords dictionary formation



Fei-Fei et al. 2005

What Is a Good Clustering?

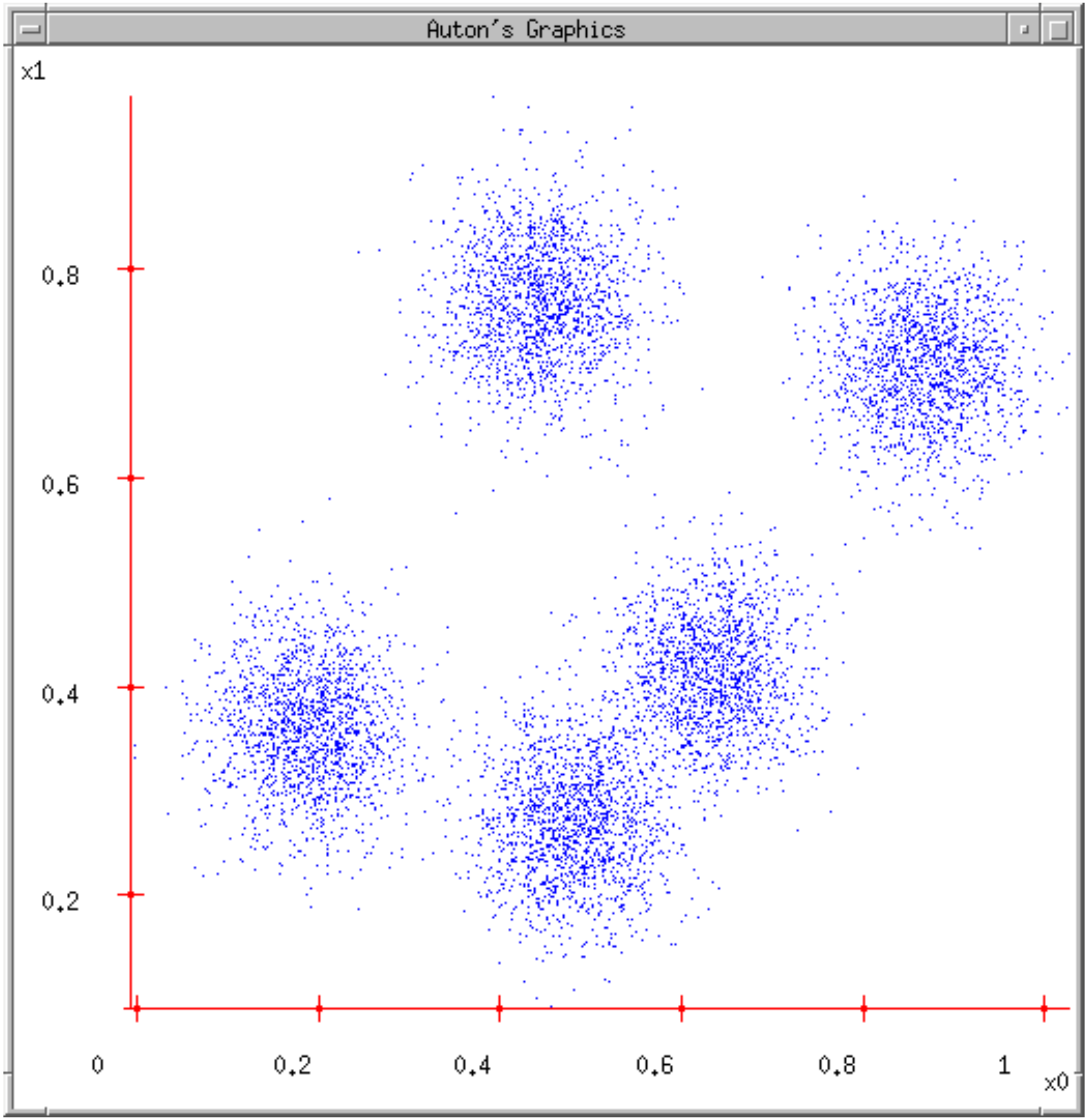
- A good clustering method will produce clusters with
  - High <u>intra-class</u> similarity
  - Low <u>inter-class</u> similarity
- Precise definition of clustering quality is difficult
  - Application-dependent
  - Ultimately subjective

K-Means Clustering

- Given *k*, the *k-means* algorithm consists of four steps:
  - Select initial centroids at random.
  - Assign each object to the cluster with the nearest centroid.
  - Compute each centroid as the mean of the objects assigned to it.
  - Repeat previous 2 steps until no change.

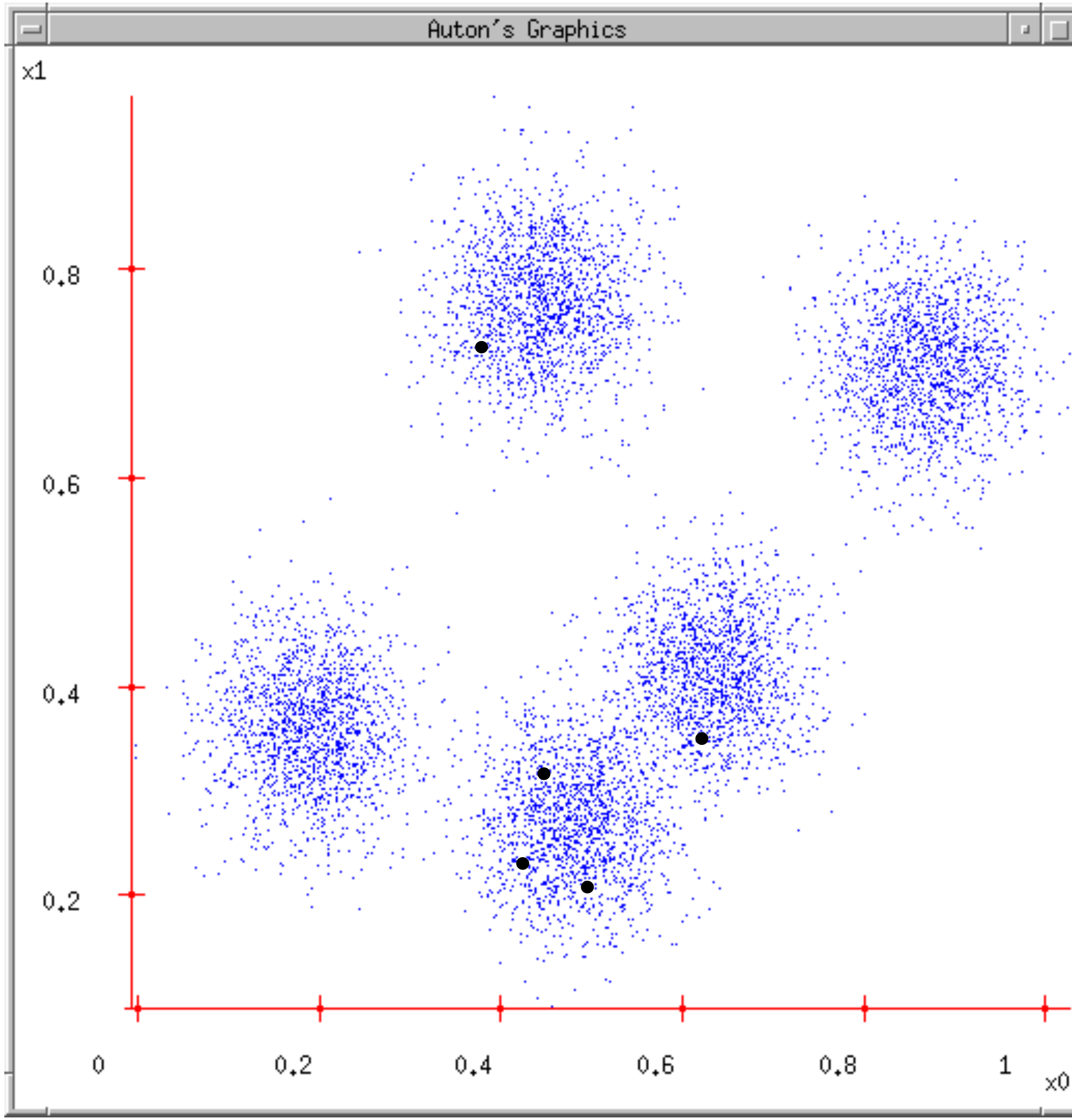
K-means

1. Ask user how many clusters they’d like. *(e.g. k=5)*



K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations

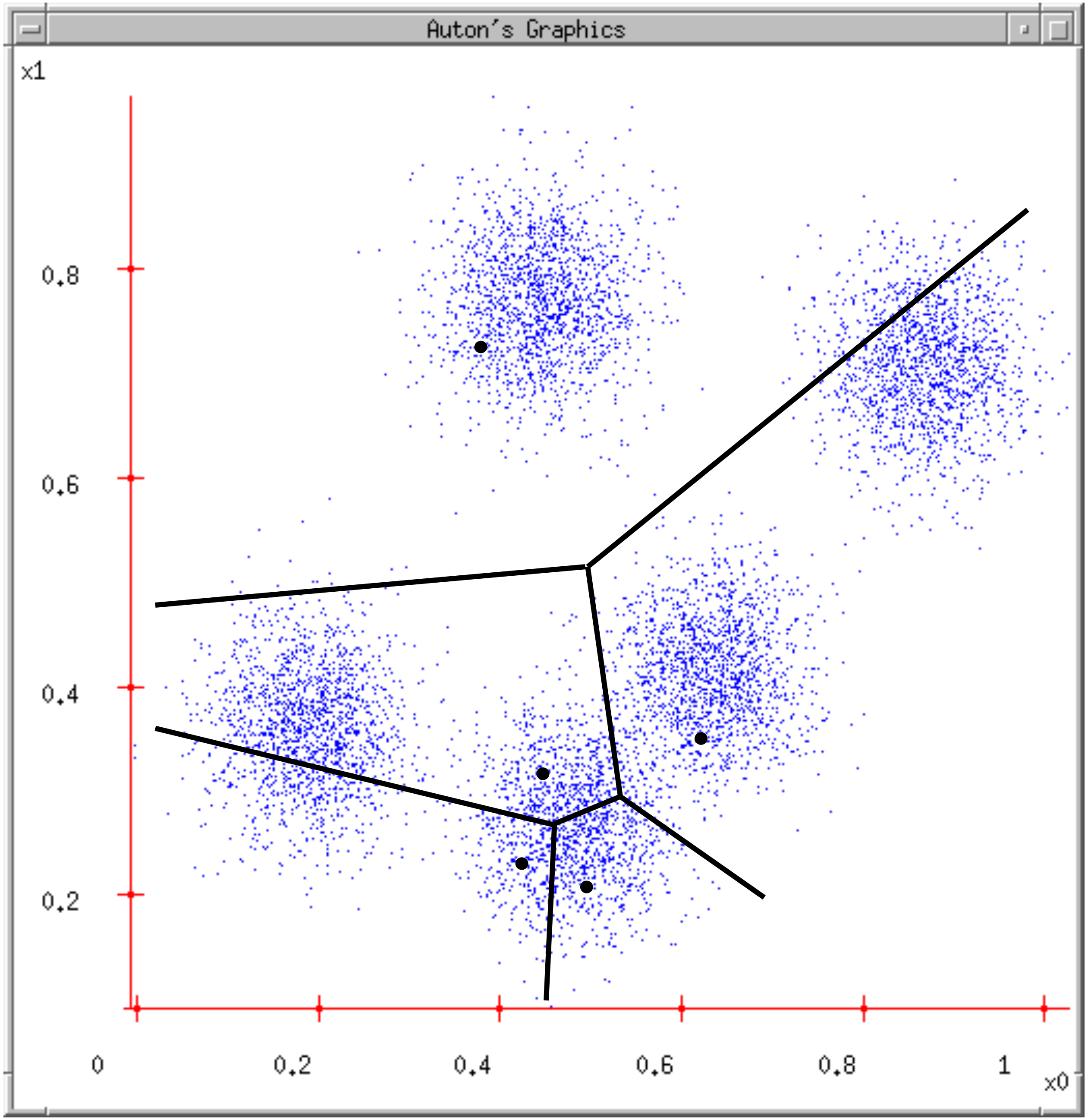


K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*

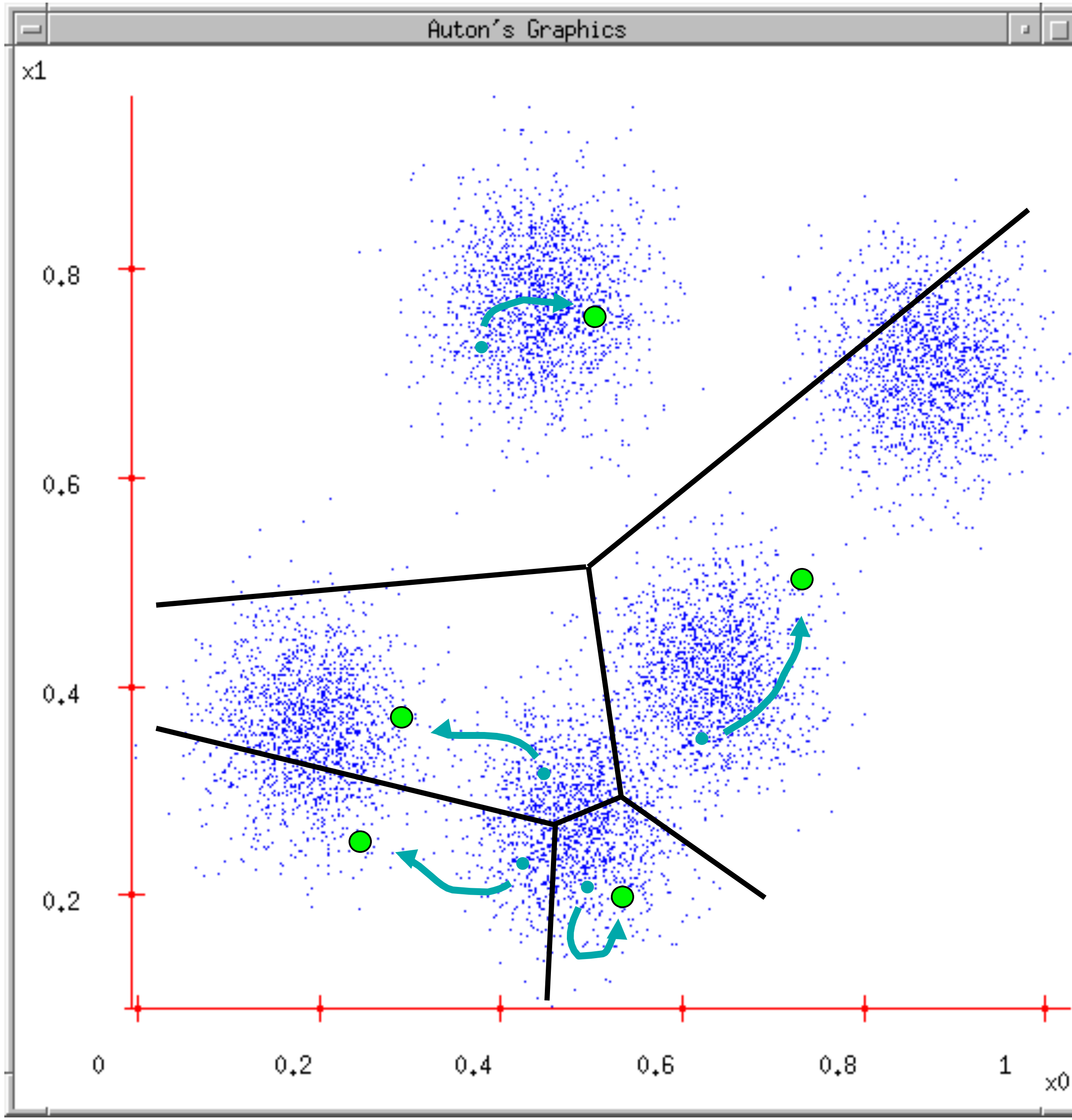
2. Randomly guess k cluster Center locations

3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



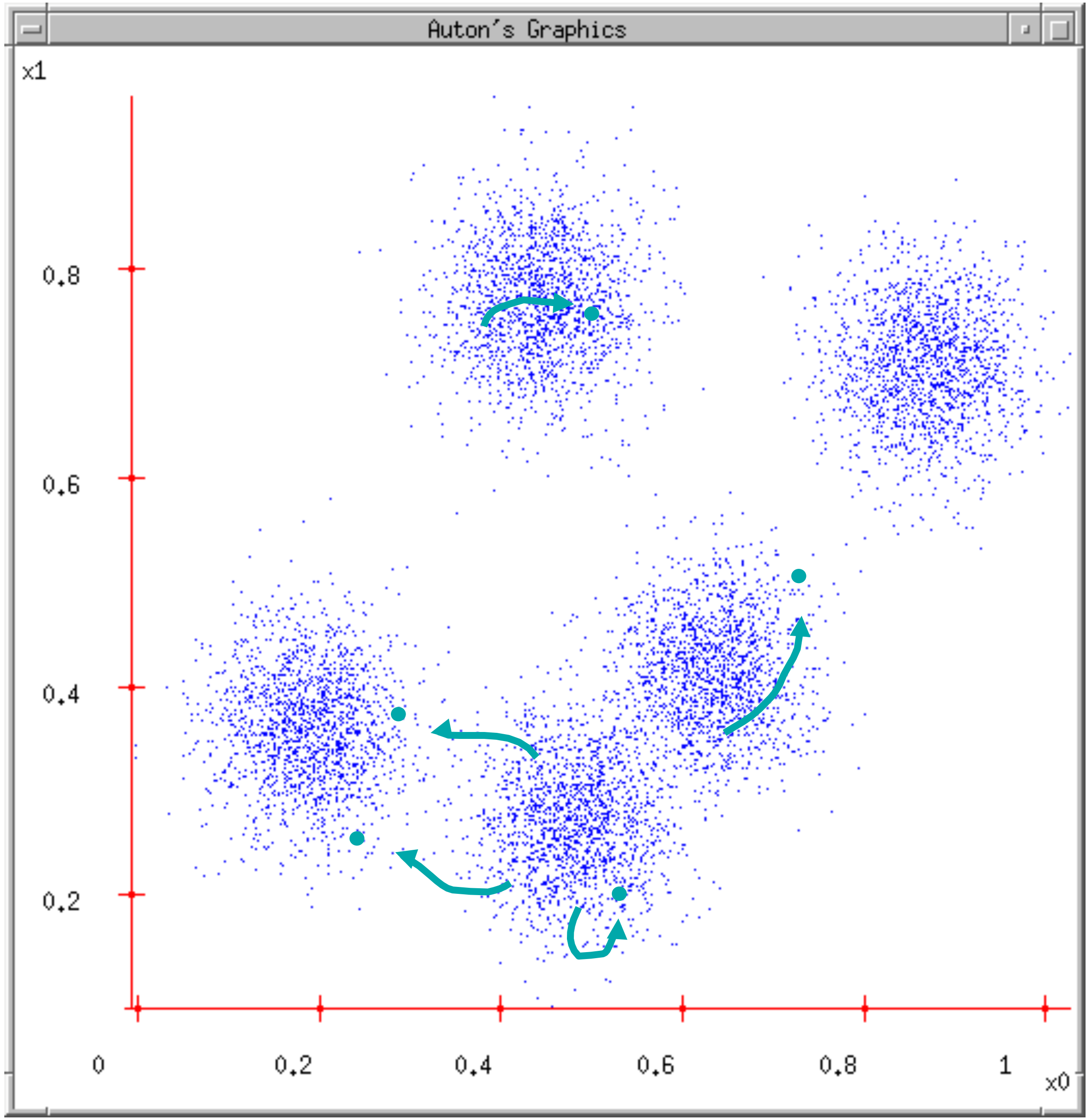
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



K-means

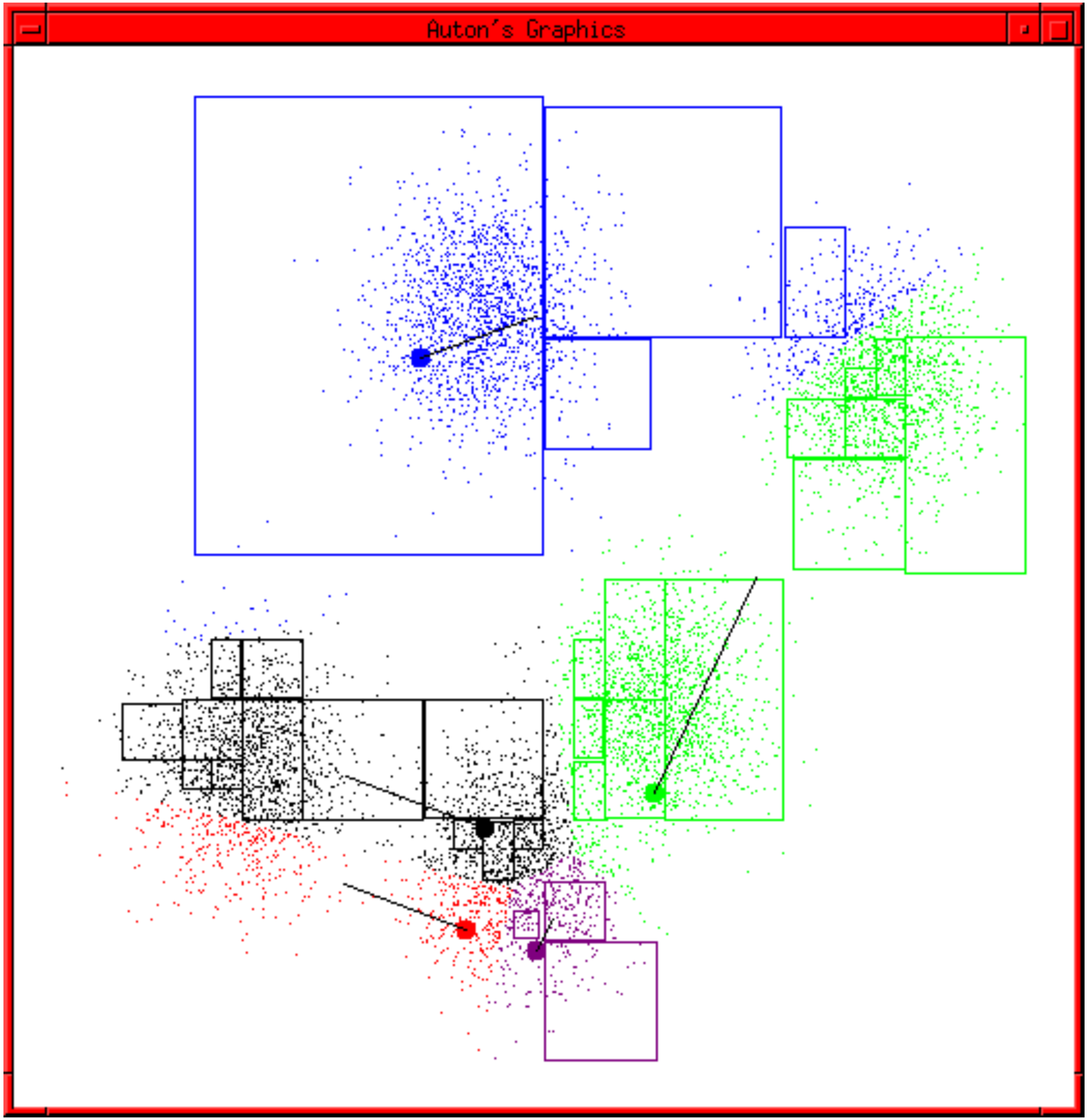
1. Ask user how many clusters they' d like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it' s closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!



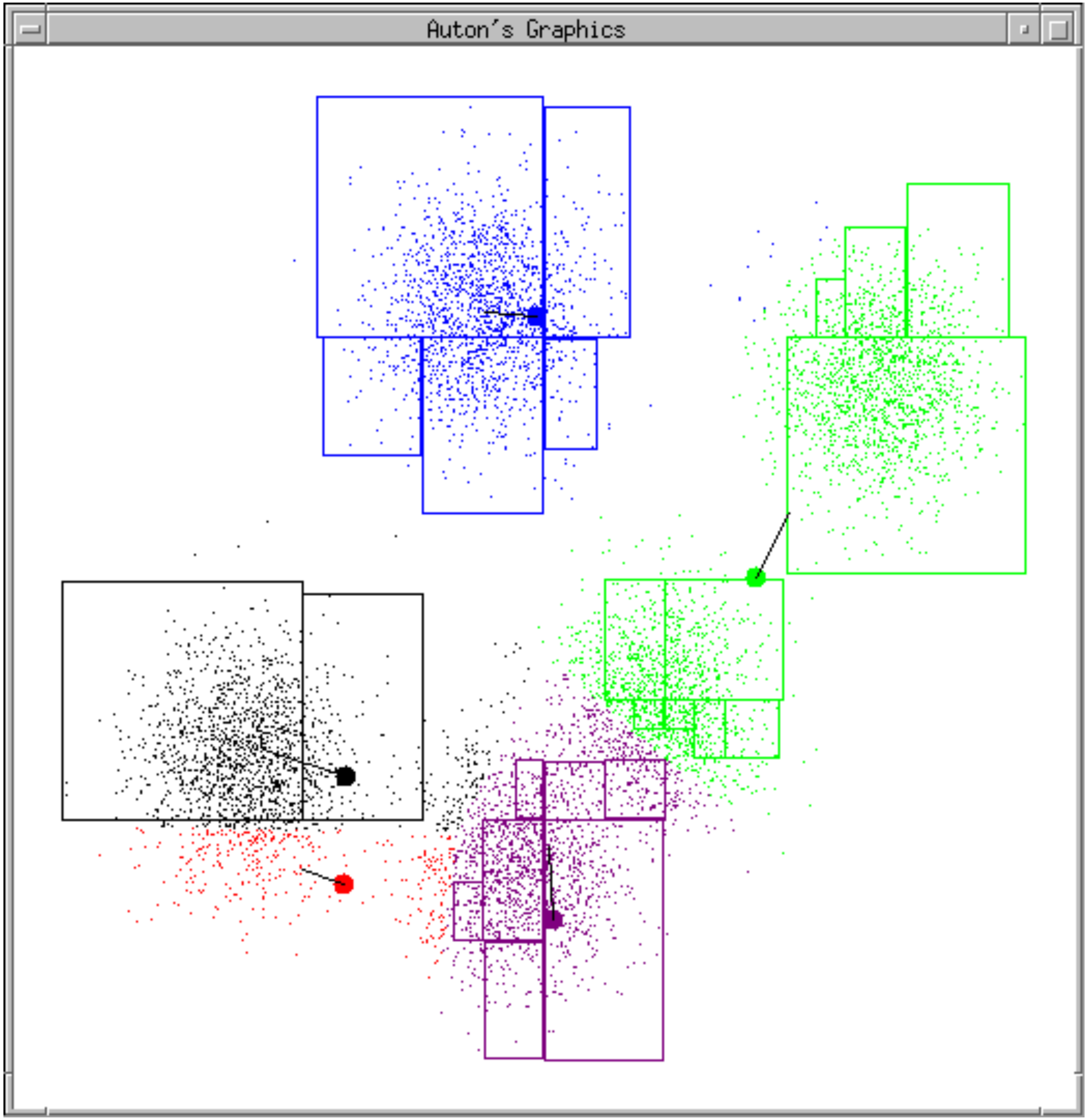
K-means Start

Example generated by Dan Pelleg's super-duper fast K-means system:

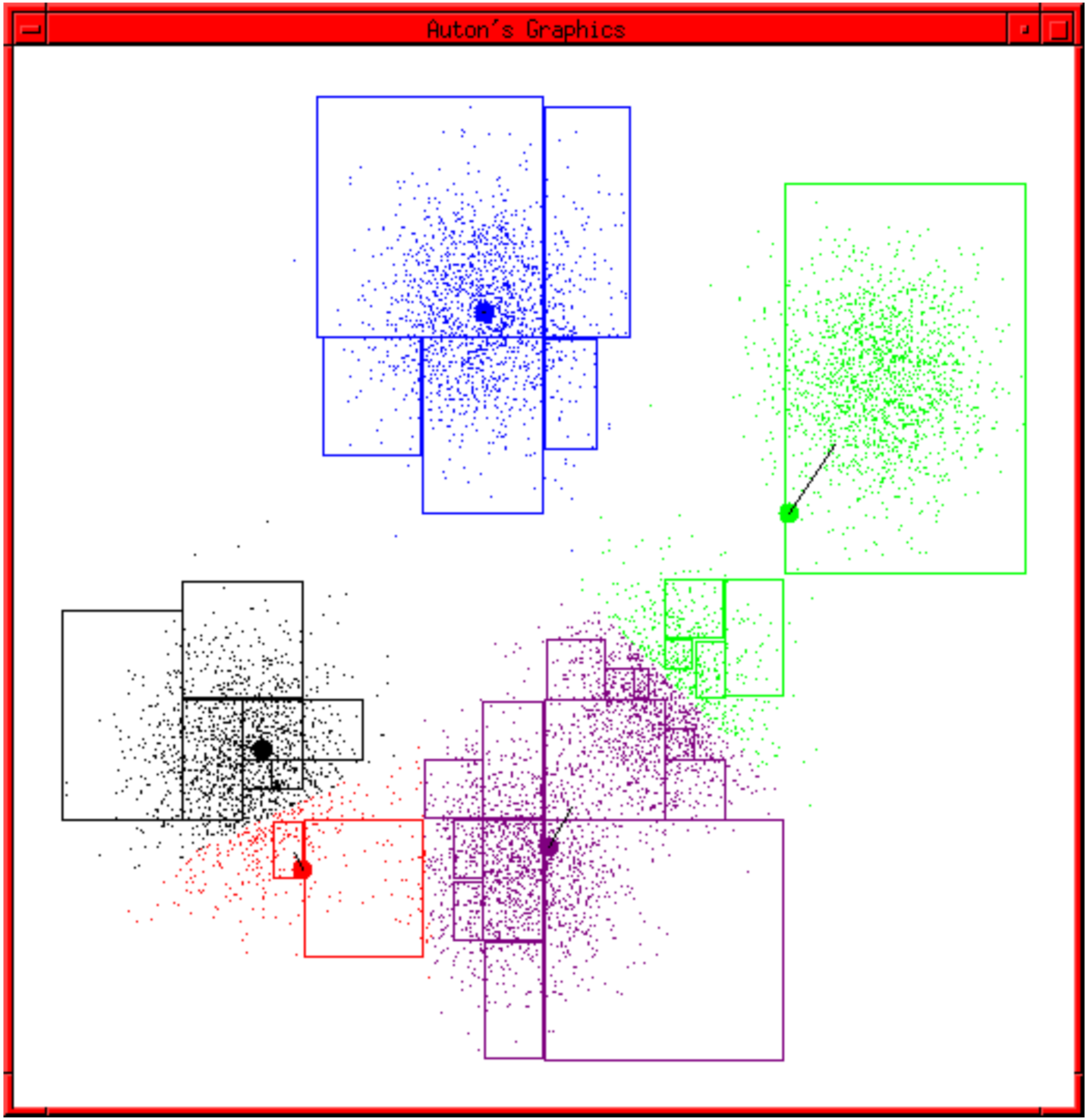
> *Dan Pelleg and Andrew Moore. Accelerating Exact k-means Algorithms with Geometric Reasoning. Proc. Conference on Knowledge Discovery in Databases 1999, (KDD99) (available on* www.autonlab.org/pap.html*)*



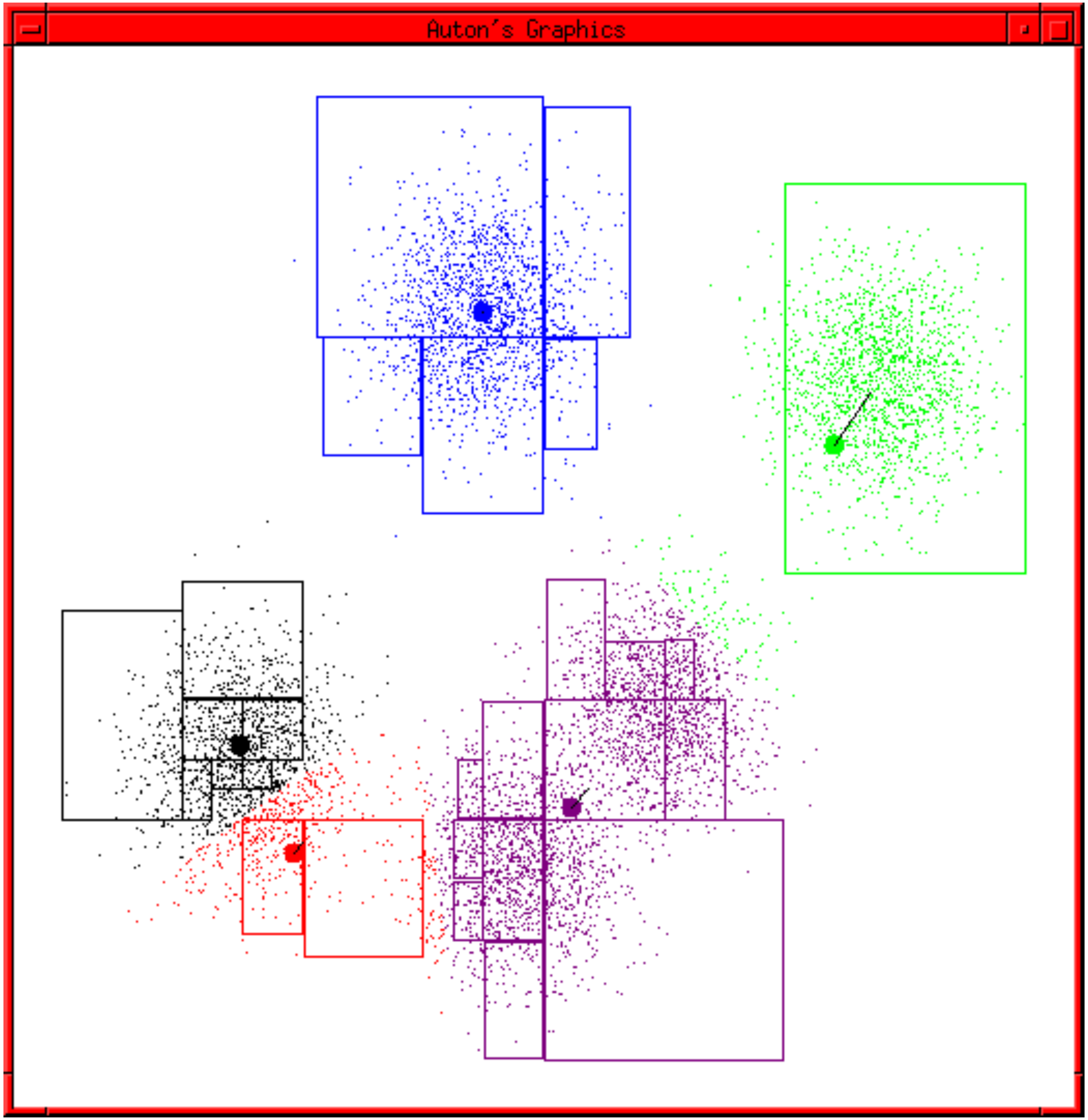
K-means continues …



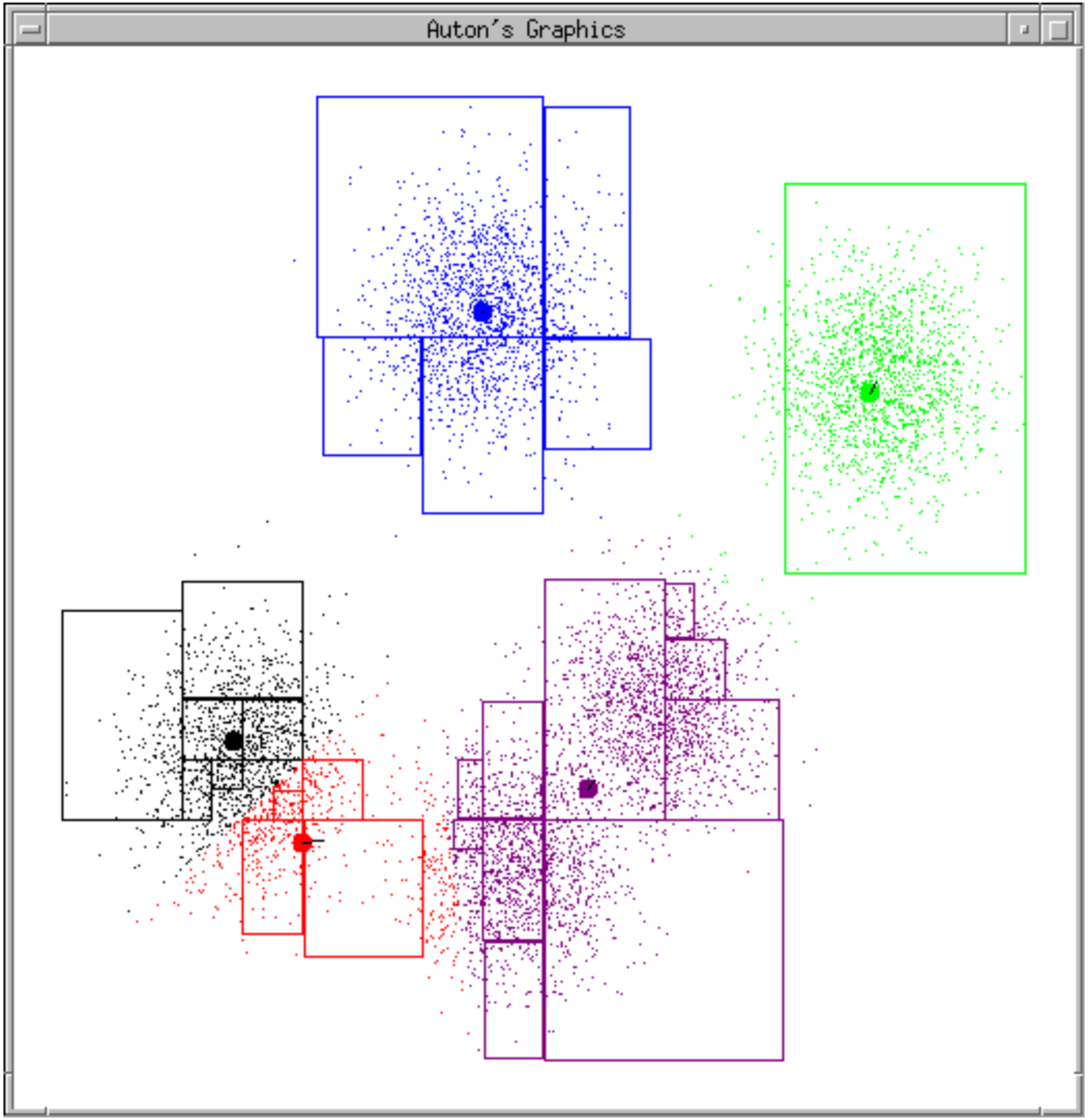
K-means continues …



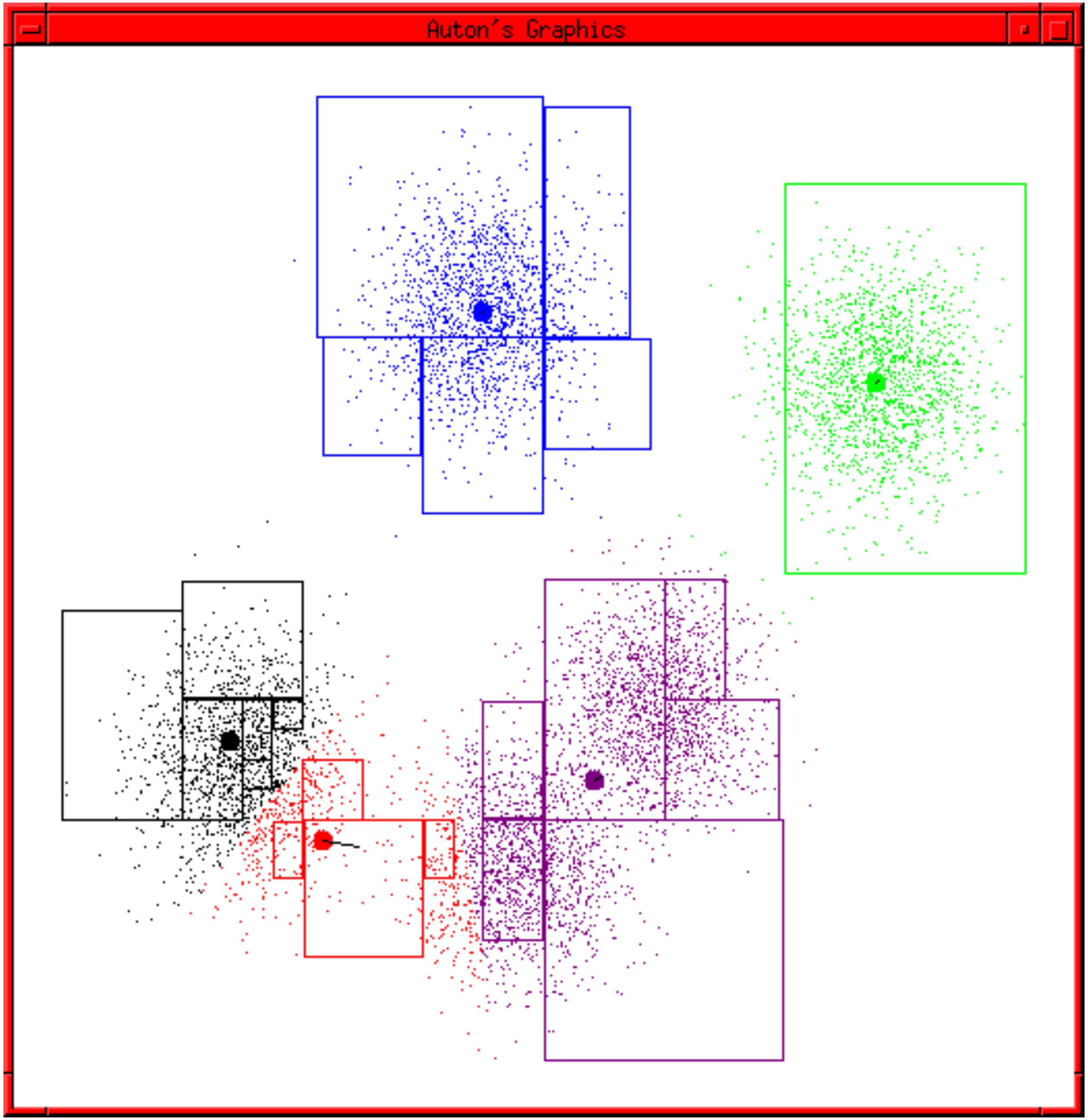
K-means continues …



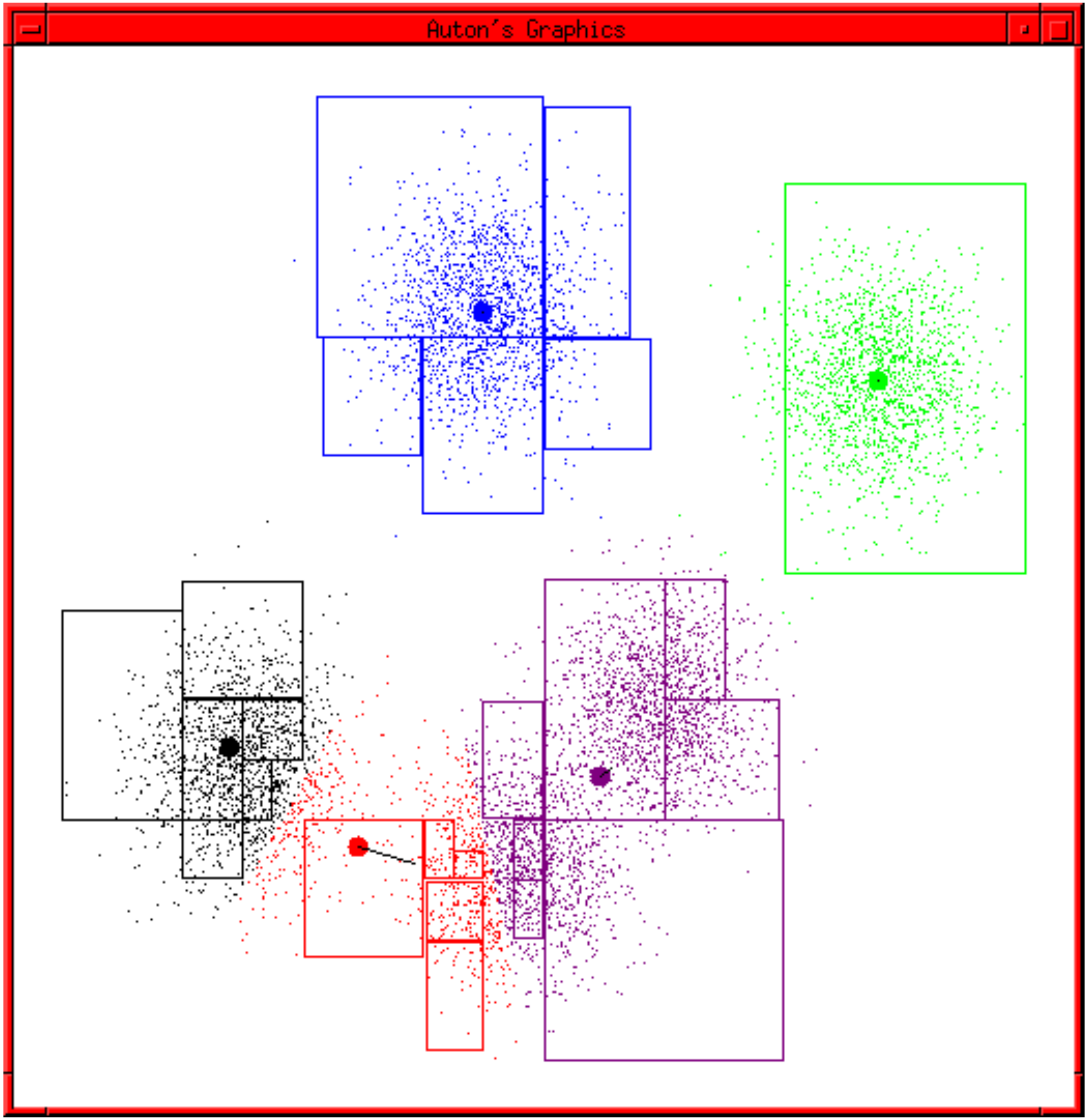
K-means continues …



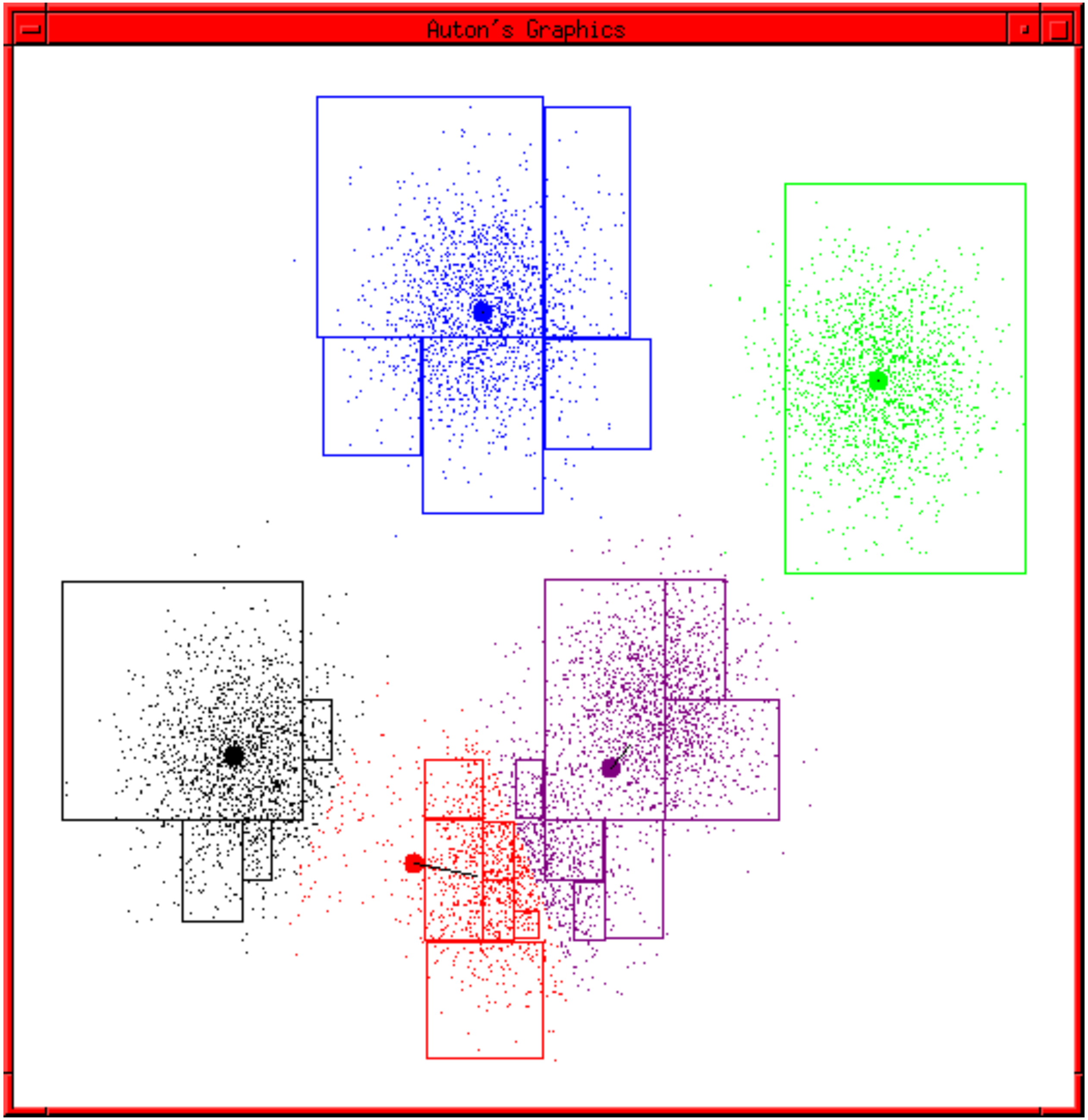
K-means continues …



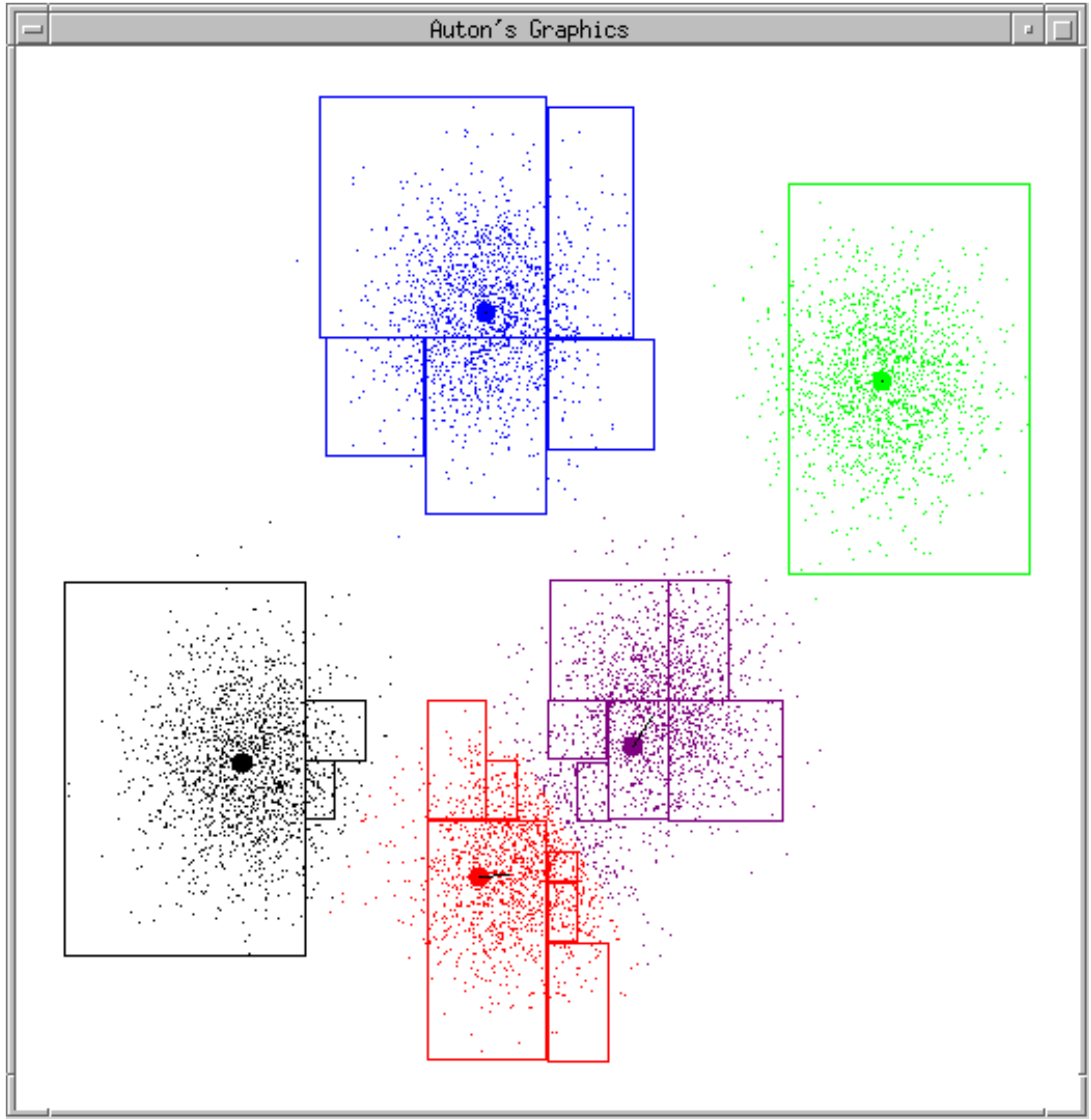
K-means continues …



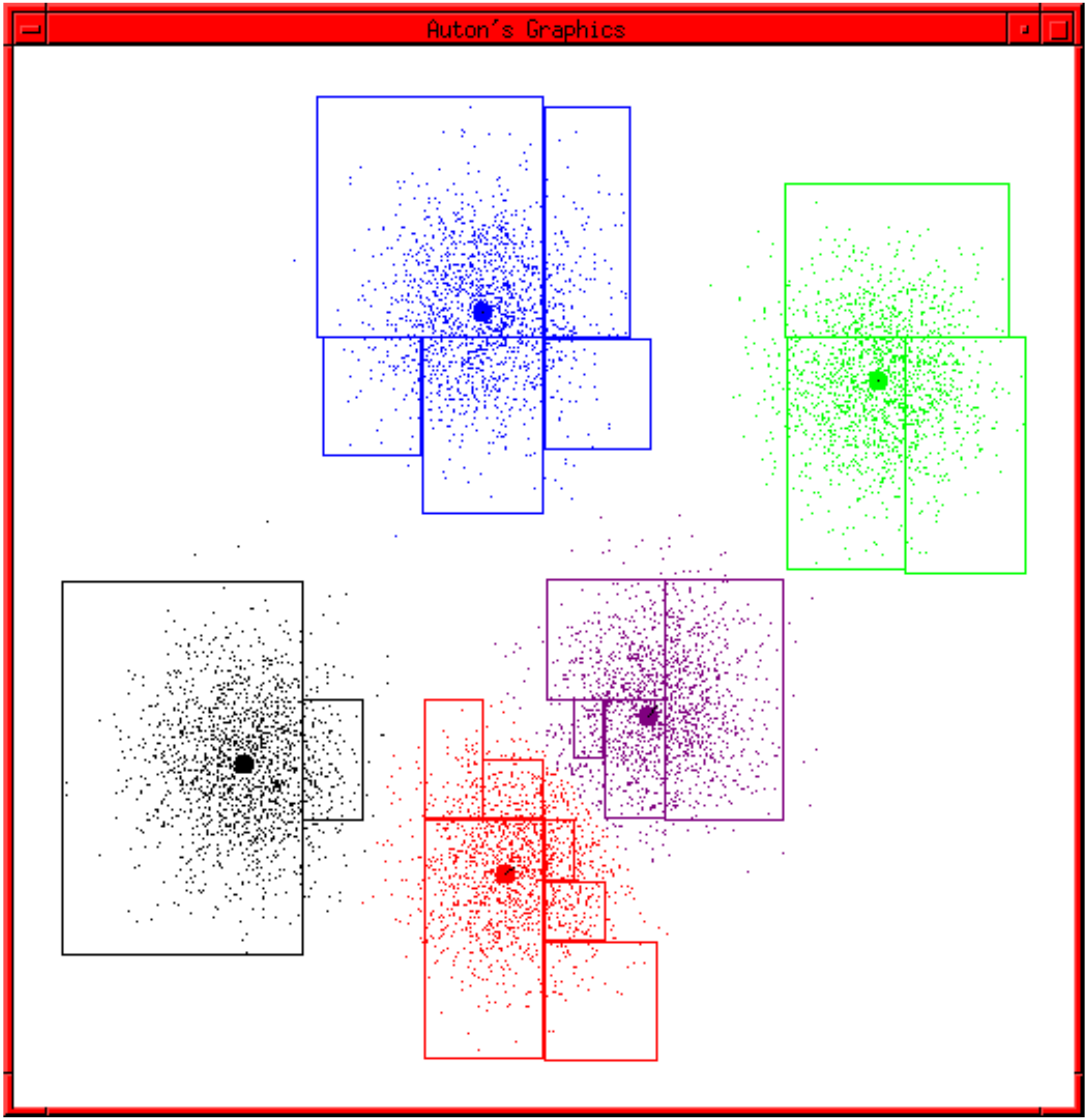
K-means continues …



K-means continues …



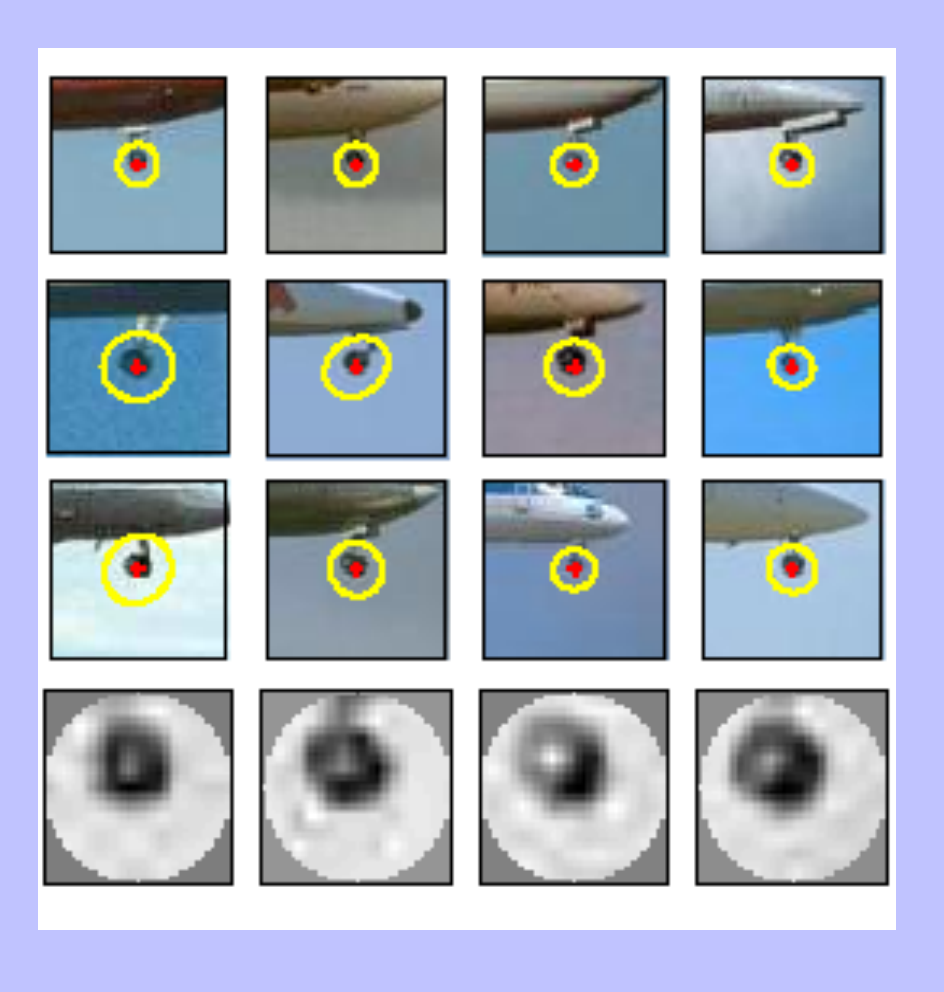
K-means terminates

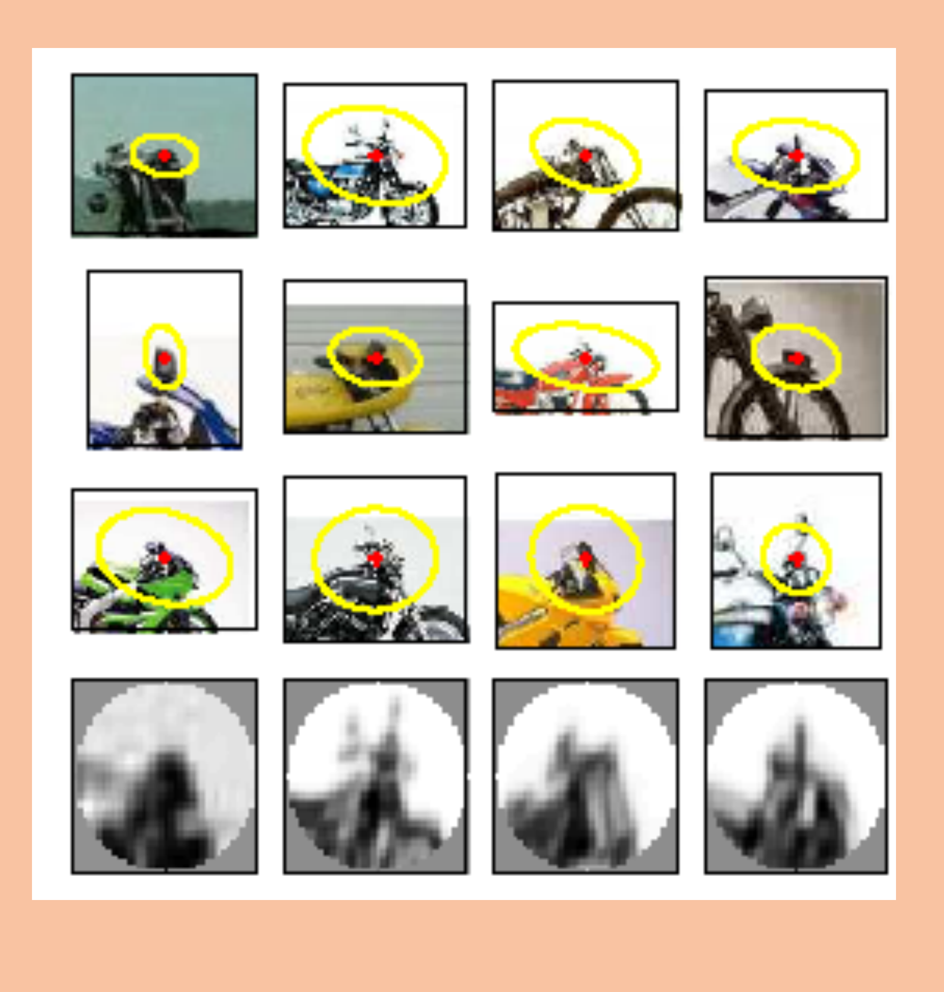


K-means Questions

- What is it trying to optimize?
- Are we sure it will terminate?
- Are we sure it will find an optimal clustering?
- How should we start it?
- How could we automatically choose the number of centers?

Image patch examples of codewords



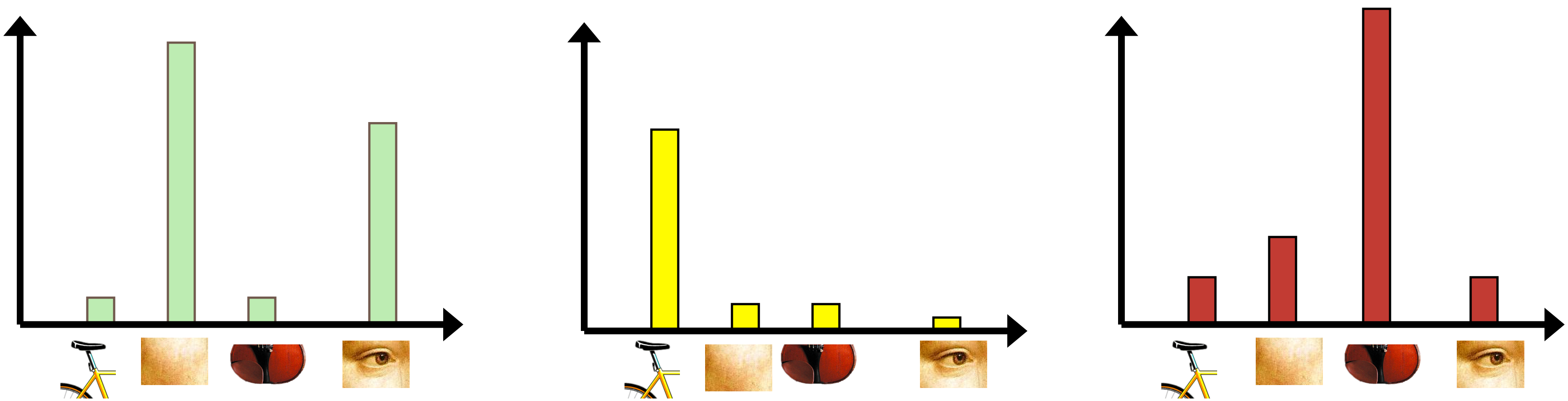


Sivic et al. 2005

3. Image representation

Image classification

- Given the bag-of-features representations of images from different classes, how do we learn a model for distinguishing them?



Discriminative methods

- Learn a decision rule (classifier) assigning bag-of-features representations of images to different classes