Unsupervised Image Segmentation Using Comparative Reasoning and Random Walks

Anuva Kulkarni	Carnegie Mellon University
Filipe Condessa	Carnegie Mellon, IST-University of Lisbon
Jelena Kovacevic	Carnegie Mellon University

Outline

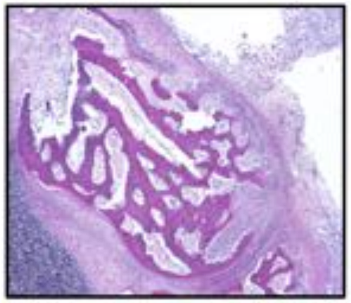
- Motivation
  - Training-free methods
  - Comparative Reasoning
  - Related work
- Approach
  - Winner Take All (WTA) Hash
  - Clustering based on Random Walks
- Some experimental results

Acknowledgements

- Example and test images taken from
  - Berkeley Segmentation Dataset (BSDS)
  - The Prague Texture Segmentation Data Generator and Benchmark

Motivation

- Goals:
  - Segment images where no. of classes unknown

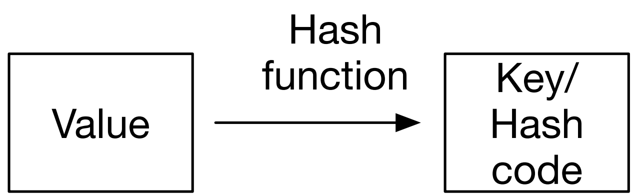


  - Eliminate training data (may not be available)
  - Fast pre-processing step for classification
- Segmentation is similarity search
- Comparative Reasoning is rank correlation using machine learning concept of "hashing"

Hashing

- Used to speed up the searching process

- A ‘hash function’ relates the data values to keys or ‘hash codes’



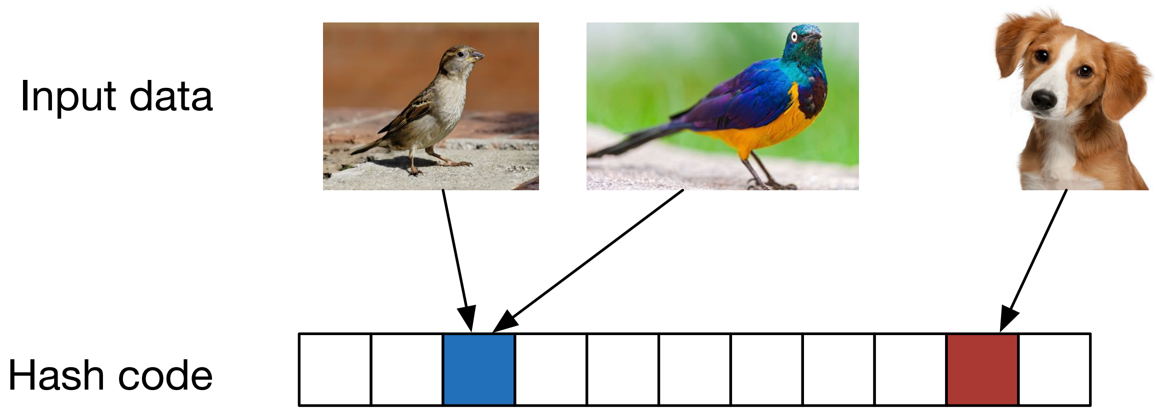
- Hash table is shortened representation of data

Hash table

Hash value	Data
001	Bird_type1
010	Bird_type2
011	Dog_type1
100	Fox_type1

Hashing

- Similar data points have the same (or close by) hash keys or “hash codes”

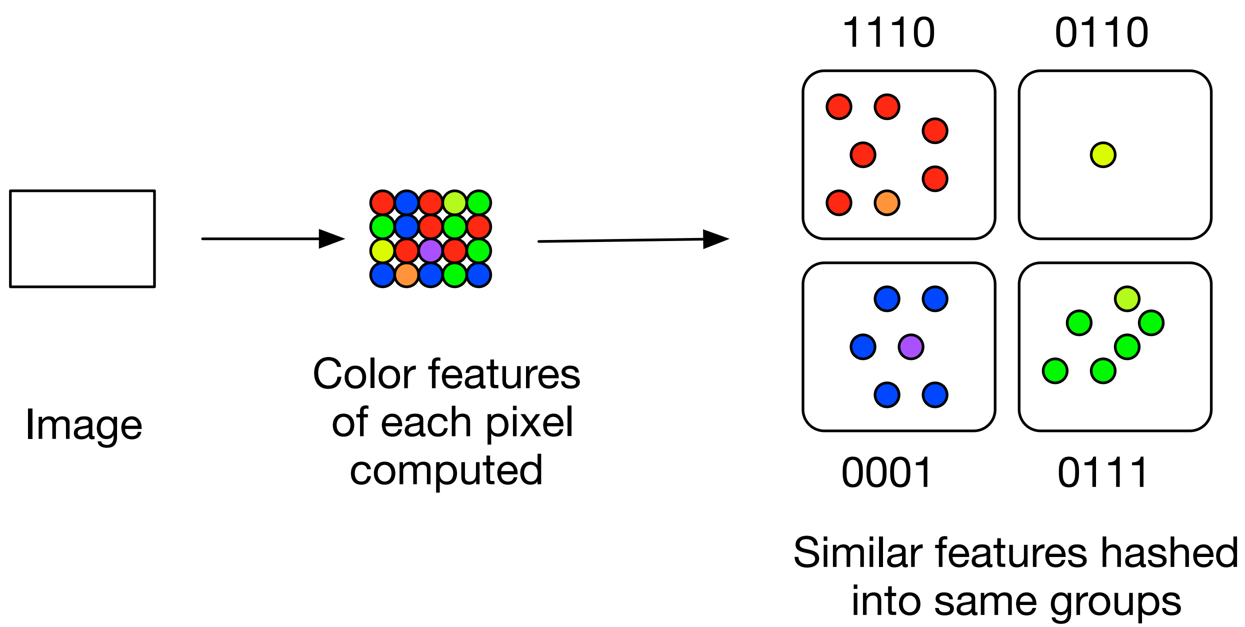


- Properties of hash functions
  – Always returns a number for an object
  – Two equal objects will always have the same number
  – Two unequal objects may not always have different numbers

Images from
https://upload.wikimedia.org/wikipedia/commons/3/32/House_sparrow04.jpg
Wikipedia
www.weknowyourdreams.com

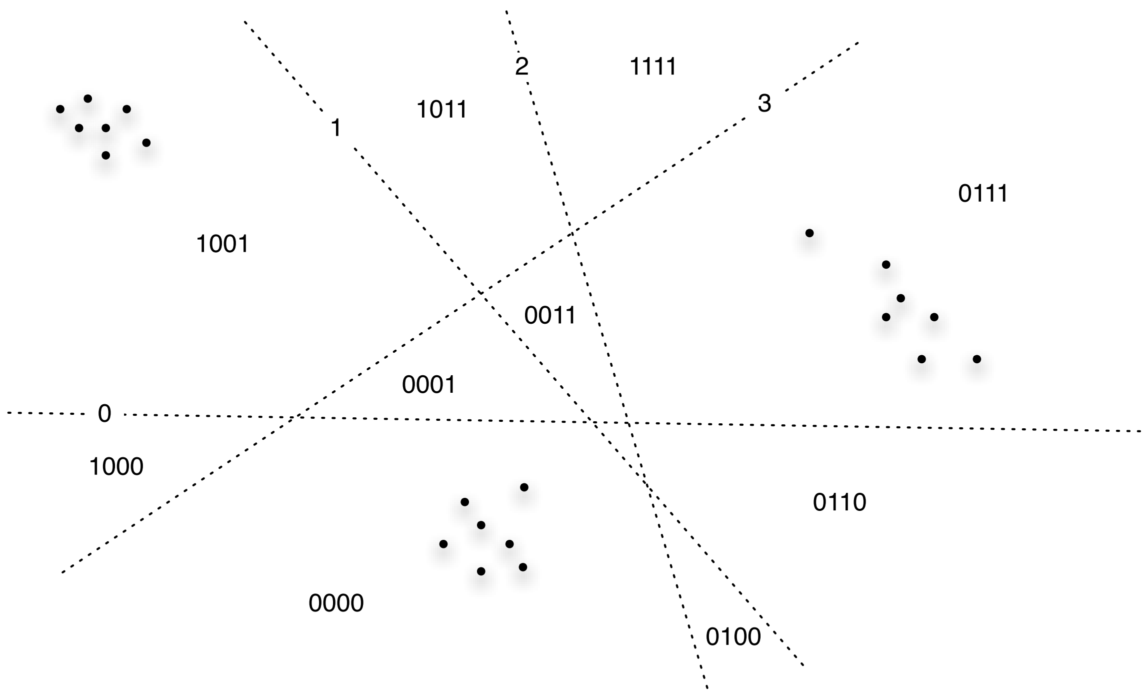
Hashing for Segmentation

- Each pixel is described by some feature vectors (*eg.* Color)
- Hashing is used to cluster them into groups



Segmentation and Randomized Hashing

- Random hashing i.e using a hash code to indicate the region in which a feature vector lies after splitting the space using a set of randomly chosen splitting planes



C. J. Taylor and A. Cowley, “Fast segmentation via randomized hashing.,” in BMVC, pp. 1–11, 2009.

Winner Take All (WTA) Hash

- A way to convert feature vectors into compact binary hash codes
- Absolute value of feature does not matter, only the ordering of values matters
- Rank correlation preserved
  - Stability
- Distance between hashes approximates rank correlation

J. Yagnik, D. Strelow, D. A. Ross, and R.s. Lin, “The power of comparative reasoning,” in *ICCV 2011,* pp. 2431–2438, IEEE, 2011.

Calculating WTA Hash

- Consider 3 feature vectors

  Step 1: Create random permutations

Permutation vector θ

3	1	5	2	6	4

	feature 1						feature 2						feature 3						
	13	4	2	11	5	3	12	5	3	10	4	2	1	90	44	5	15	6	
Step 1	2	13	5	4	3	11	3	12	4	5	2	10	44	1	15	90	6	5	Permute with θ

Calculating WTA Hash

- Step 2: Choose first K entries. Let K=3

Permutation vector θ

3	1	5	2	6	4

	feature 1						feature 2						feature 3						
	13	4	2	11	5	3	12	5	3	10	4	2	1	90	44	5	15	6	
Step 1	2	13	5	4	3	11	3	12	4	5	2	10	44	1	15	90	6	5	Permute with θ
Step 2	2	13	5	4	3	11	3	12	4	5	2	10	44	1	15	90	6	5	Choose first K entries

Calculating WTA Hash

- Step 3: Pick the index of the max. entry. This is the hash code 'h' of that feature vector

Permutation vector θ

3	1	5	2	6	4

	feature 1						feature 2						feature 3						
	13	4	2	11	5	3	12	5	3	10	4	2	1	90	44	5	15	6	
Step 1	2	13	5	4	3	11	3	12	4	5	2	10	44	1	15	90	6	5	Permute with θ
Step 2	2	13	5	4	3	11	3	12	4	5	2	10	44	1	15	90	6	5	Choose first K entries
Step 3	2	13	5	4	3	11	3	12	4	5	2	10	44	1	15	90	6	5	Hash code is index of top entry out of the K

h=2 h=2 h=1

Calculating WTA Hash

Notice that Feature 2 is just Feature 1 perturbed by one, but Feature 3 is very different

Permutation vector θ

3	1	5	2	6	4

	feature 1	feature 2	feature 3	
	13 4 2 11 5 3	12 5 3 10 4 2	1 90 44 5 15 6	
Step 1	2 13 5 4 3 11	3 12 4 5 2 10	44 1 15 90 6 5	Permute with θ
Step 2	2 13 5 4 3 11	3 12 4 5 2 10	44 1 15 90 6 5	Choose first K entries
Step 3	2 13 5 4 3 11	3 12 4 5 2 10	44 1 15 90 6 5	Hash code is index of top entry out of the K
	h=2	h=2	h=1	

Feature 1 and Feature 2 are similar

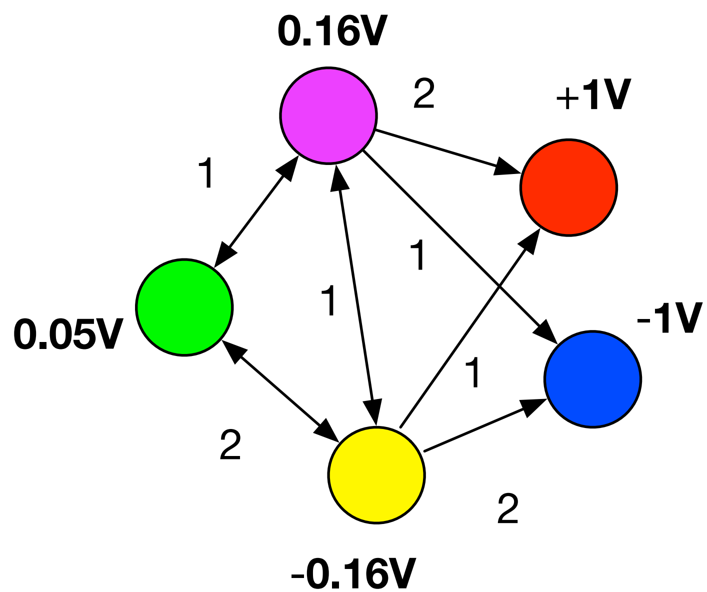
Random Walks

- Understanding proximity in graphs
- Useful in **propagation** in graphs – creates probability maps

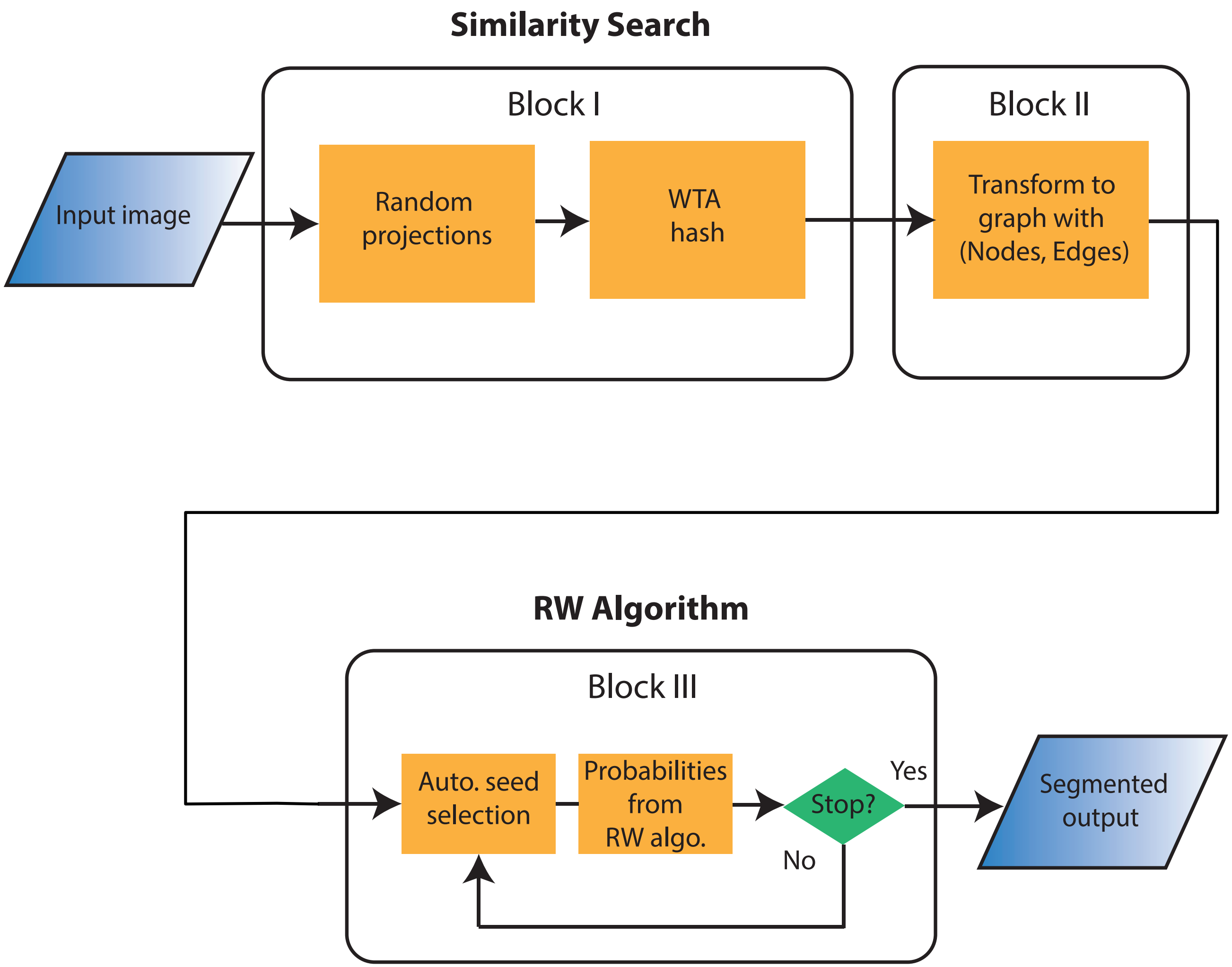
- Similar to electrical network with voltages and resistances

- It is supervised.

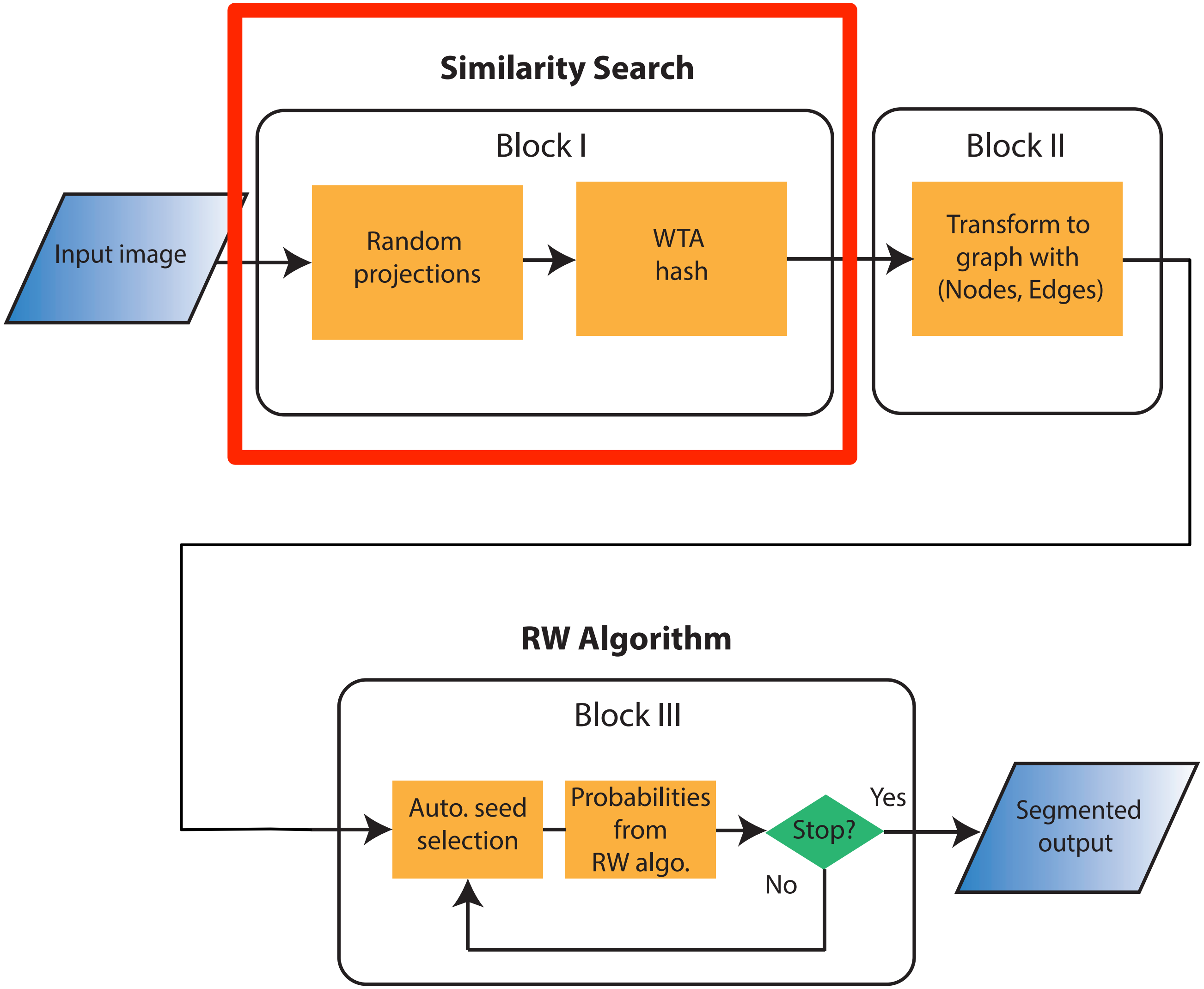
User must specify seeds



Our Approach

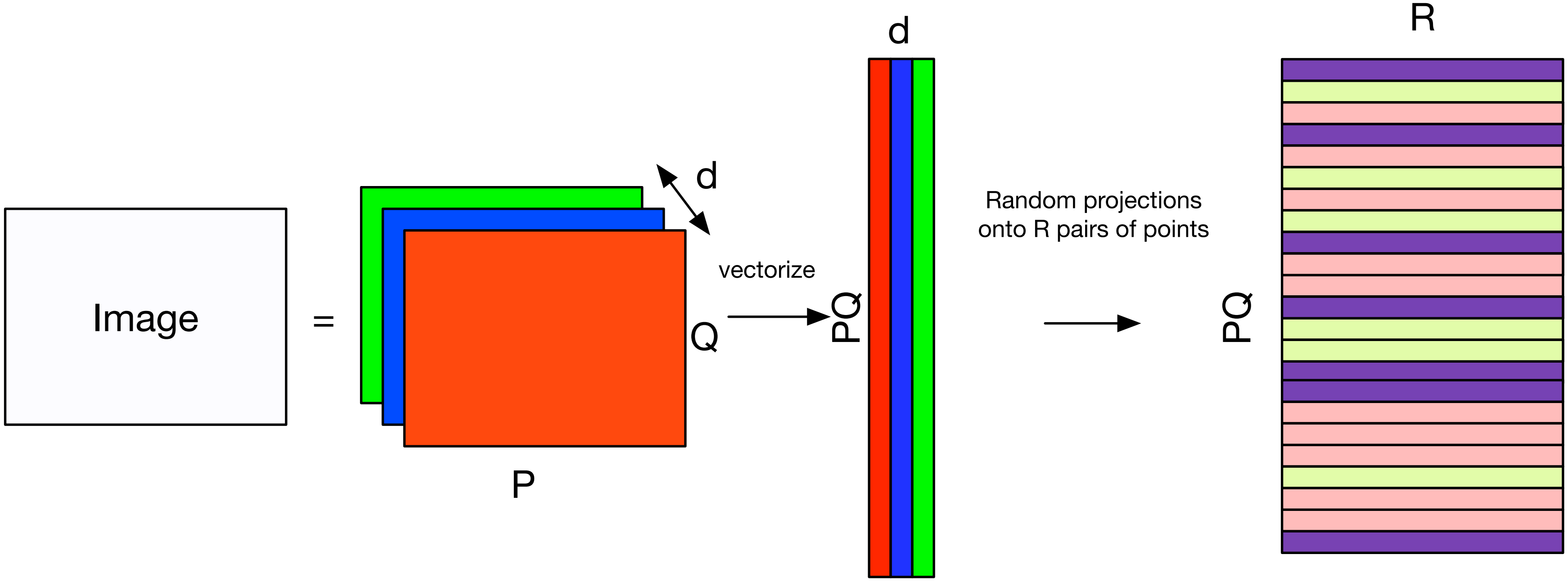


Block I: Similarity Search



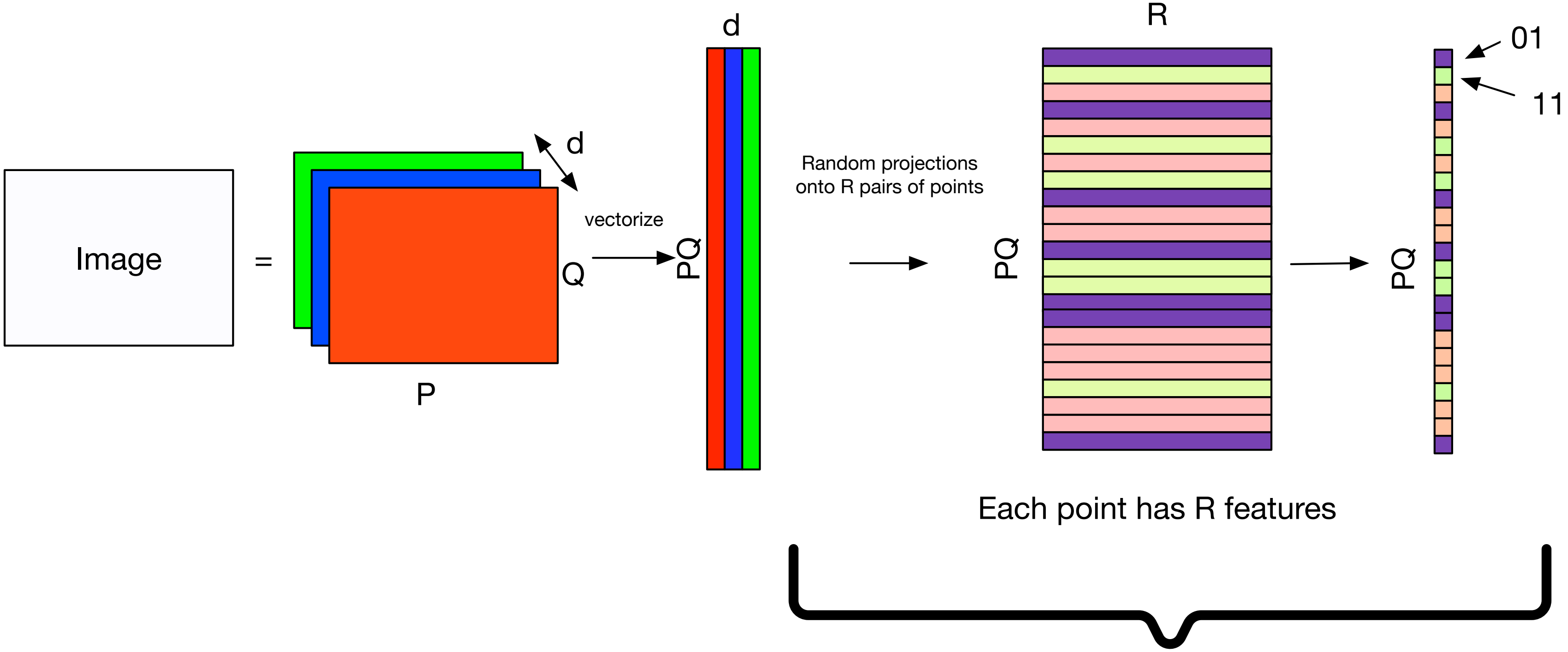
WTA hash

- Image Dimensions: *P* x *Q* x *d*
- Project onto R randomly chosen hyperplanes
  - Each point in image has R feature vectors



WTA hash

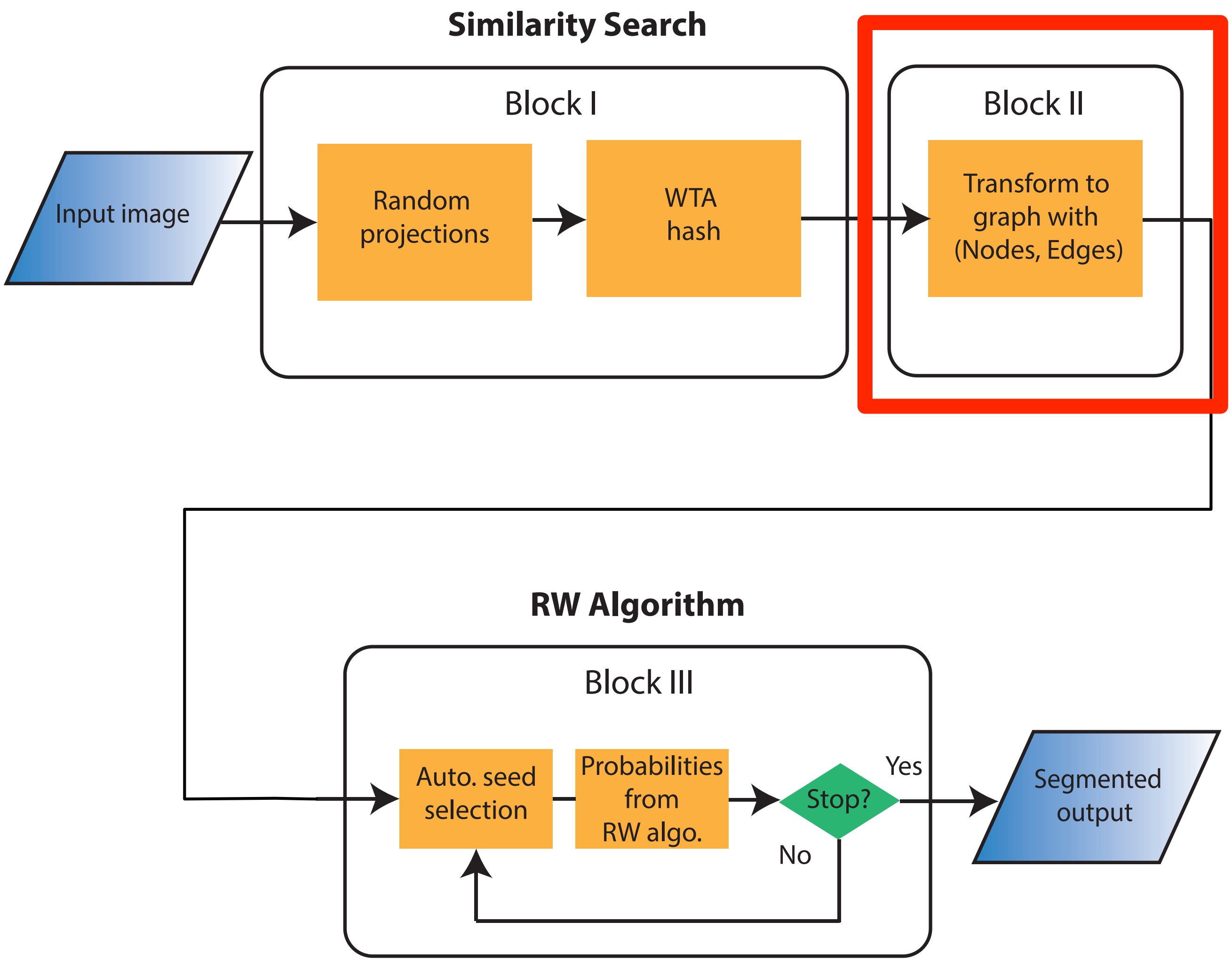
- Run WTA hash *N* times.



K=3
Hence possible values
of hash codes
are 00, 01, 11

Repeat this N times to get PQ x N matrix of hash codes

Block II: Create Graph



Create Graph

- Run WTA hash N times → each point has N hash codes
- Image transformed into lattice
- Calculate edge weight between nodes *i* and *j*

$$\omega_{i,j} = \exp(-\beta \nu_{i,j})$$

where:

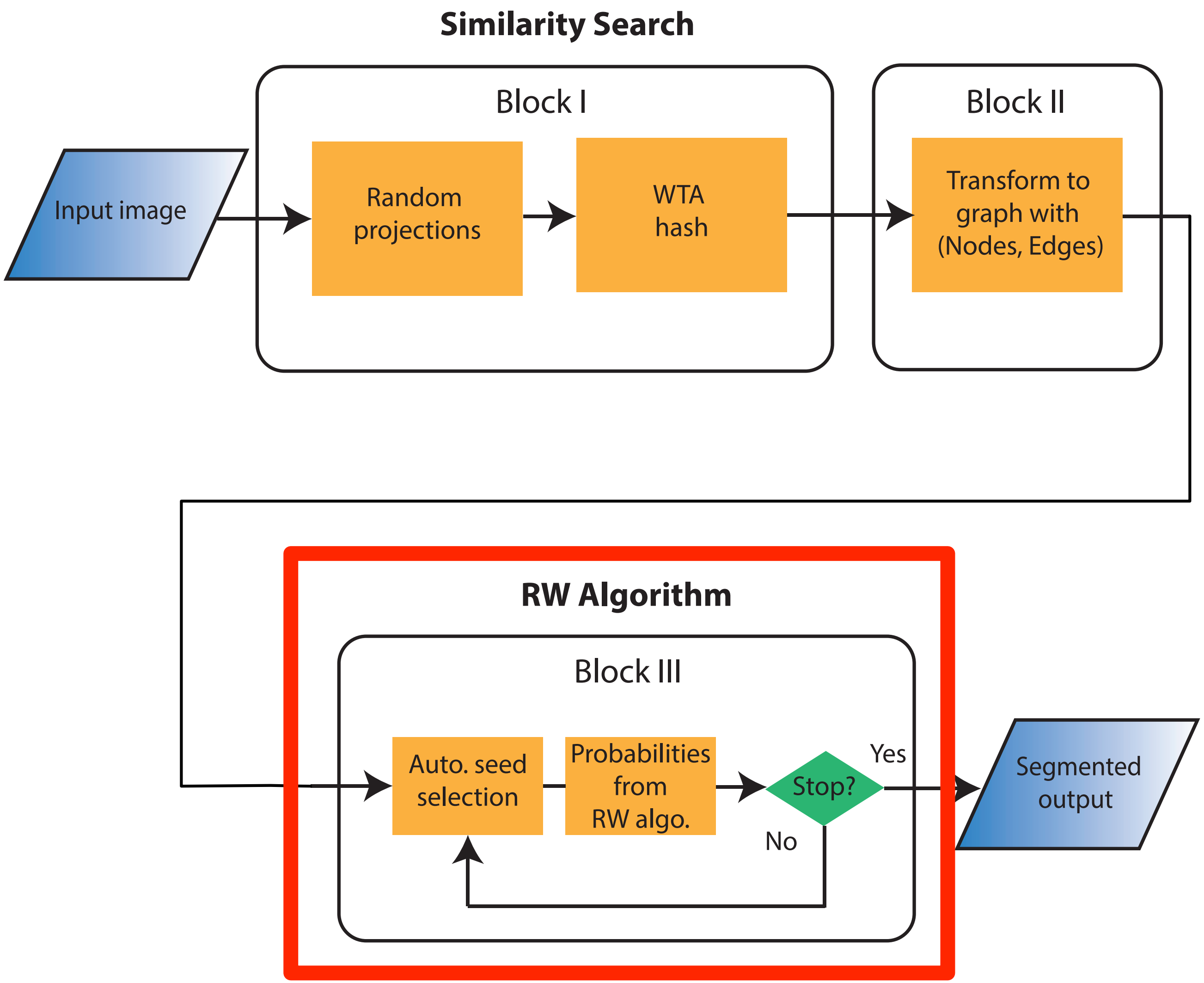
$$\nu_{i,j} = \frac{d_H(i,j)}{\gamma}$$

$d_H(i,j) =$ Avg. Hamm. distance over all N hash codes of i and j

$\gamma =$ Scaling factor

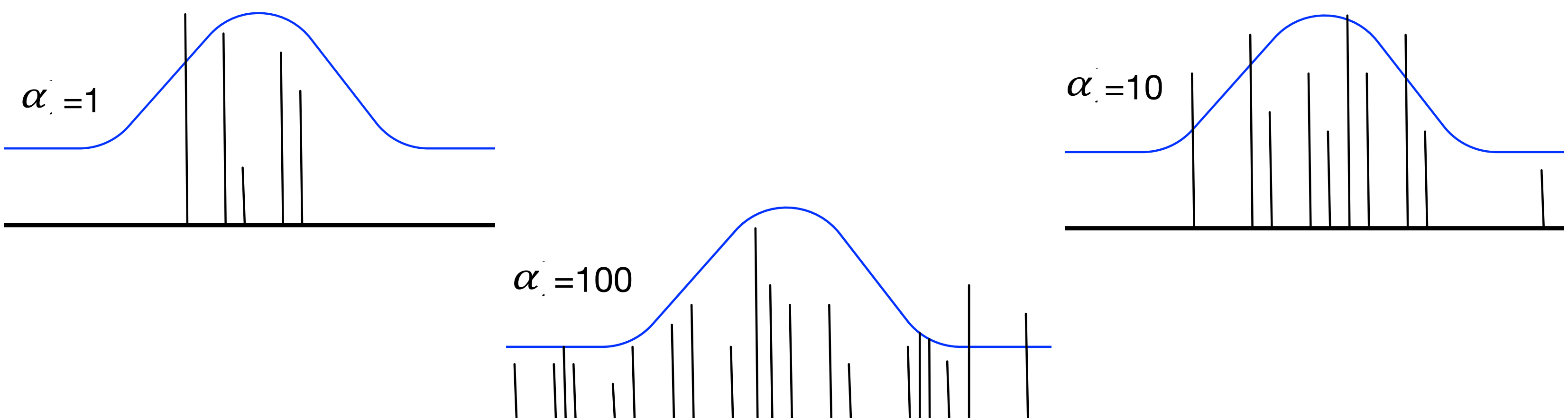
$\beta =$ Weight parameter for the RW algorithm

Block III: RW Algorithm



Seed Selection

- Needs initial seeds to be defined
- Unsupervised draws using Dirichlet processes
- DP(G_0,α)
  - G_o is base distribution
  - α is discovery parameter
- Larger α leads to discovery of more classes



Seed Selection

- Probability that a new seed belongs to a new class is proportional to α
- Probability for the i^{th} sample with class label y_i
  - Result by Blackwell and MacQueen, 1973

$$p(y_i = c|\mathbf{y}_{-i}, \alpha) = \frac{n_c^{-i} + \frac{\alpha}{C_{tot}}}{n - 1 + \alpha}$$

where:

$C_{tot} = \text{Total number of classes}$

$y_i = \text{Class label } c, c \in \{1, 2 \ldots C_{tot}\}$

$\mathbf{y}_{-i} = \{y_j | j \neq i\}$

$n_c^{-i} = \text{number of samples in } c\text{th class excluding } i\text{th sample}$

Seed Selection

- Unsupervised, hence C_{tot} is infinite. Hence,

$$\lim_{C_{tot}\to\infty} p(y_i = c|\mathbf{y}_{-i}, \alpha) = \frac{n_c^{-i}}{n-1+\alpha} \quad \forall c, n_c^{-i} > 0$$

Class is non-empty

- "Clustering effect" or "rich gets richer"

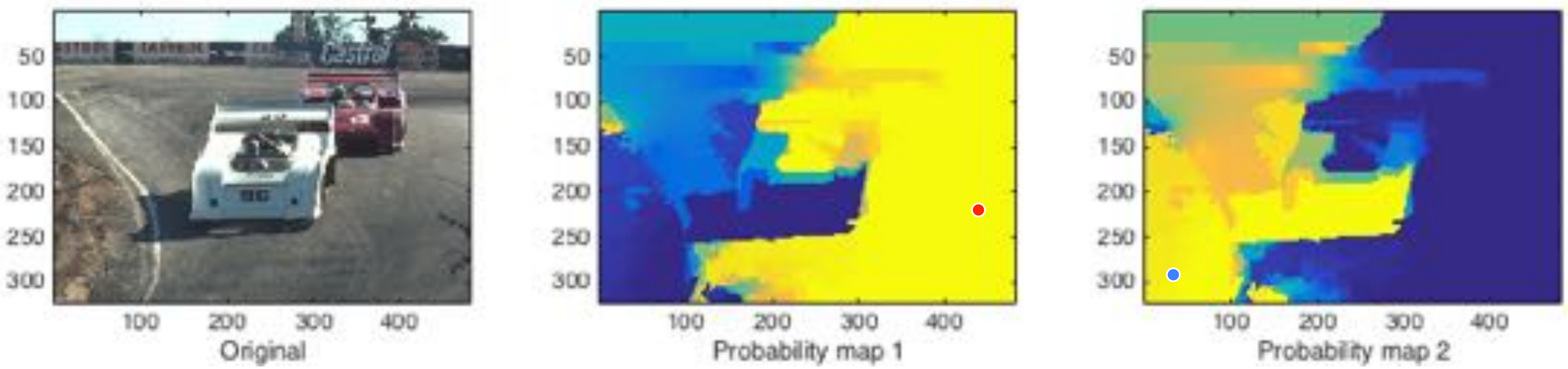
- Probability that a new class is discovered:

$$\lim_{C_{tot}\to\infty} p(y_i \neq y_j \text{ for all } j < i|\mathbf{y}_{-\mathbf{i}}, \alpha) = \frac{\alpha}{n-1+\alpha} \quad \forall c, n_c^{-i} = 0$$

Class is empty or new

Random Walks

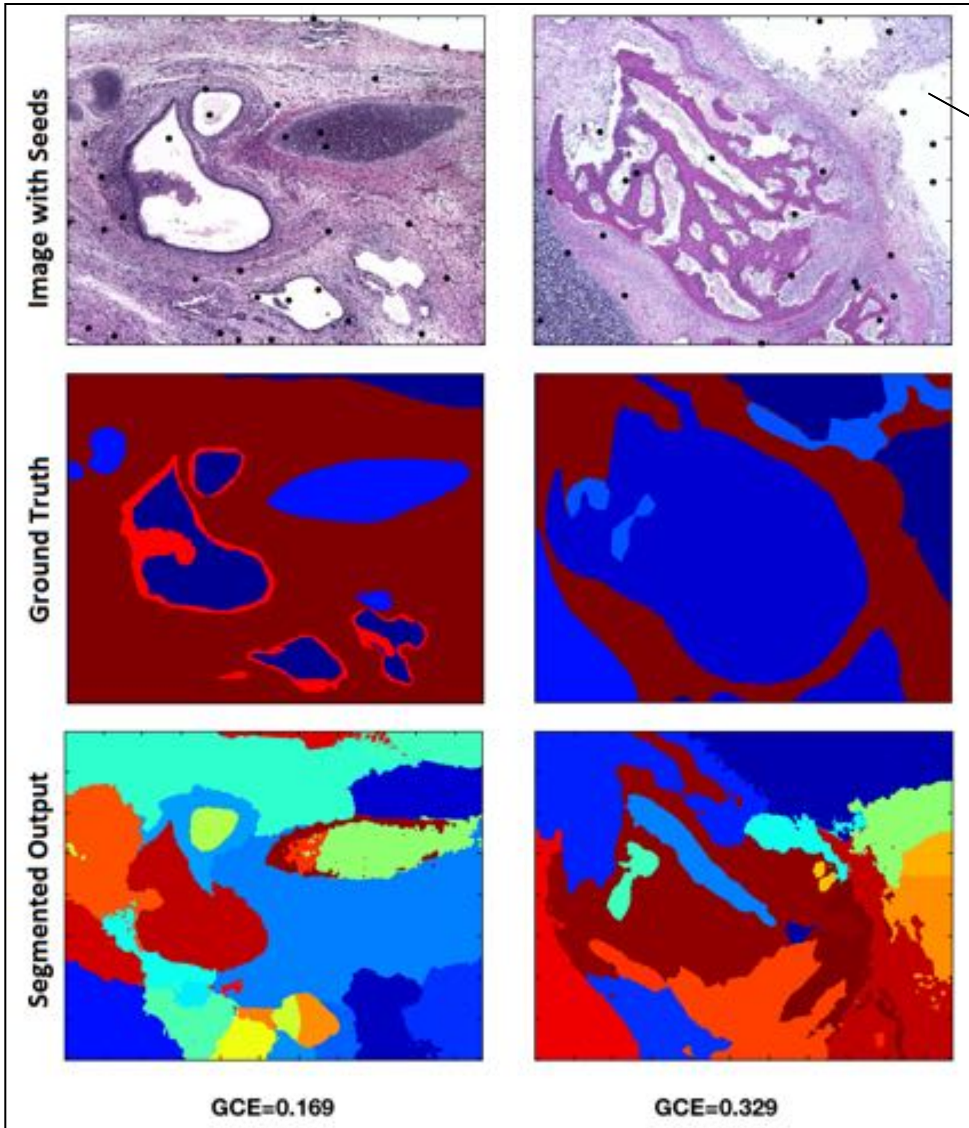
- Use the RW algorithm to generate probability maps in each iteration



- Entropy calculated with probability maps
- Entropy-based stopping criteria
  - Cluster purity ↑, Avg. image entropy ↓

Experimental Results

Histology images

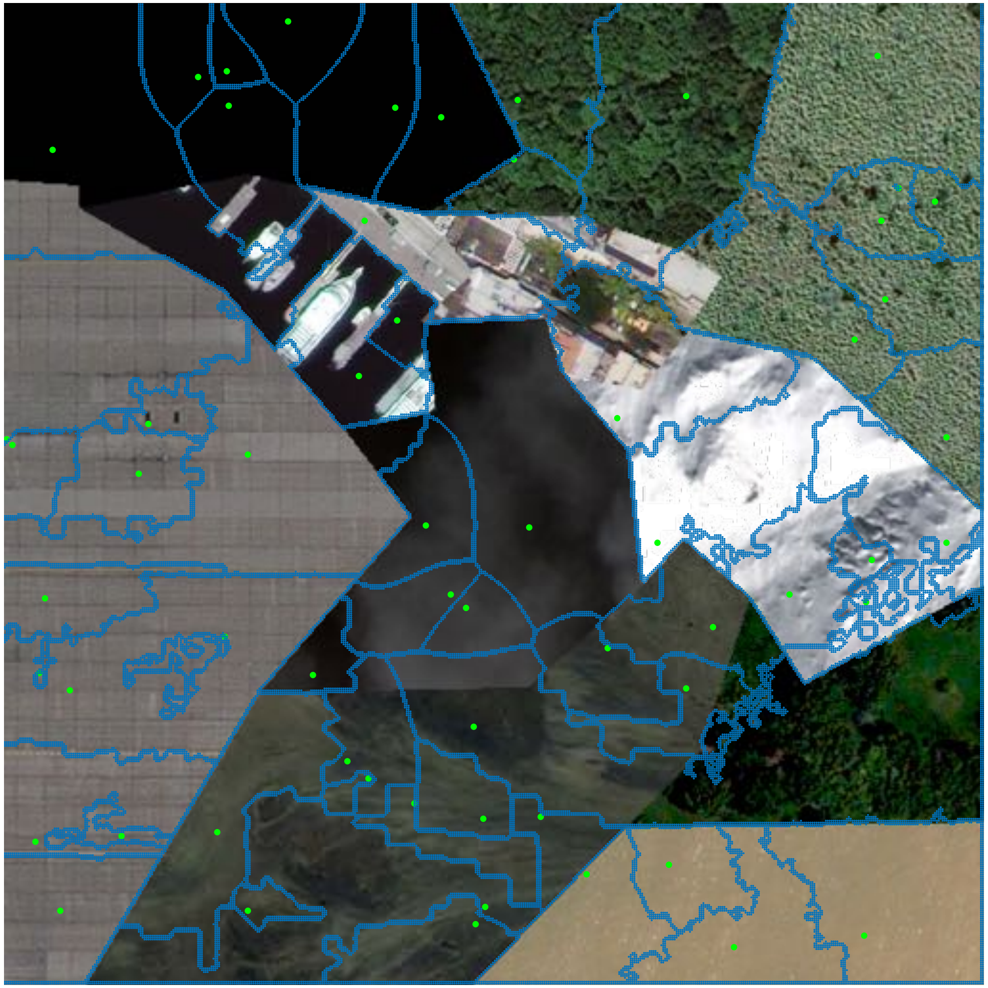


Automatically
Picked seeds

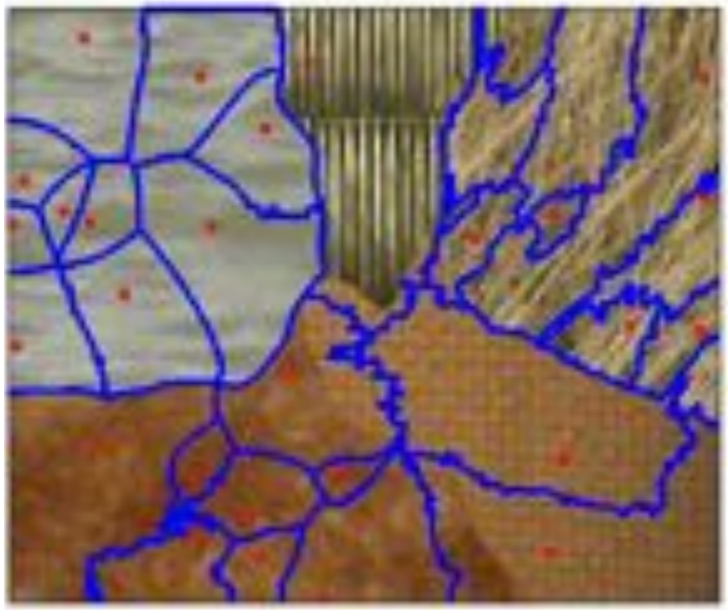
Berkeley segmentation subset
Avg. GCE of dataset = 0.186

Experimental Results

TexGeo
Avg GCE of dataset = 0.134



TexBTF
Avg. GCE of dataset = 0.061



Experimental Results

- Comparison measure: Global Consistency Error (GCE)*
  - Lower GCE indicates lower error

No. of features	GCE Score			
	BSDSubset	TexBTF	TexColor	TexGeo
10	0.179	0.063	0.159	0.102
20	0.180	0.065	0.159	0.129
40	0.186	0.061	0.156	0.134

*C. Fowlkes, D. Martin, and J. Malik, "Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches," vol. 2, pp. II–54, IEEE, 2003.

Experimental Results

- Comparison measure: Global Consistency Error (GCE)
  - Lower GCE indicates lower error

No. of features	GCE Score			
	BSDSubset	TexBTF	TexColor	TexGeo
10	0.179	0.063	0.159	0.102
20	0.180	0.065	0.159	0.129
40	0.186	0.061	0.156	0.134

- Comparison with other methods**:
  - Performed on BSDS Subset

Method	Human	RAD	Seed	Learned Affinity	Mean Shift	Normalized cuts
GCE	**0.080**	0.205	0.209	0.214	0.260	0.336

**E. Vazquez, J. Van De Weijer, and R. Baldrich, "Image segmentation in the presence of shadows and highlights," pp. 1–14, Springer, 2008.

Conclusions

- Comparative reasoning and Winner Take All hash enables **fast similarity search**
- Our method performs **unsupervised segmentation using context** (Random Walks-based clustering)
- There is no need to predefine the **number of classes**
- This can be used as a **pre-processing step** for classification of hyperspectral images, biomedical images etc.

Thank you