Efficient Nearest Neighbor Search in Distributed Manner

Cheng, Ti-Chung (1155043421) Supervised by Prof. James Cheng
The Chinese University of Hong Kong, CSE Department

Agenda

- Introduction
- Background and Theory
- Overall Design and implementation
- From single machine to distributed computation
- Generalized framework
- Experiments and Results
- Discussion and Conclusion

1

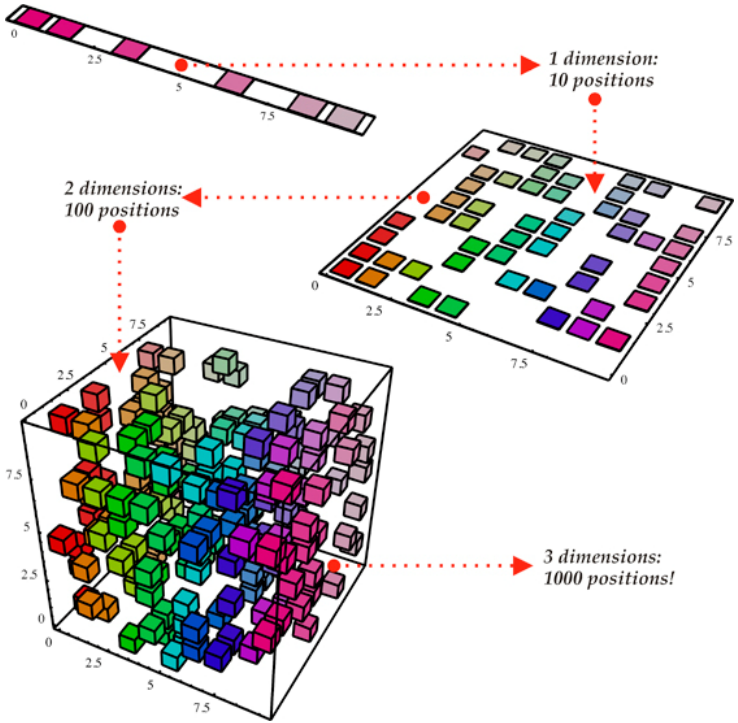
Introduction

This is where it all began

Introduction

- Last semester: explore the basic of Nearest Neighbor Search
- **Motivation: Solve real world nearest neighbor problems efficiently**
- Theory -> Application
- Compare real world solution
- Real world question: large dataset, high dimensional, useful
- Image retrieval

Curse of dimensionality



- Combinatorial explosion
- Exponential growth: $O(2^d)$
- Each added dimension doubles the need to apply all combination

Image reference: https://www.scinethpc.ca/wp-content/uploads/2012/02/bengio.jpg

Nearest Neighbor Search

Definition Nearest neighbor Search

Retrieving a set of k−most similar items as set $R \subset X$ when given a query q, where the superset X includes n points, $X = \{x_i \mid i = 1...n\}$ and $x_i \in R^d$

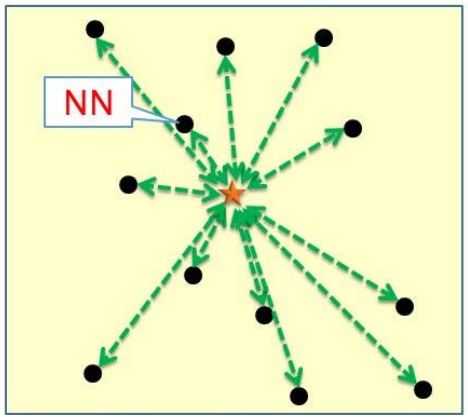
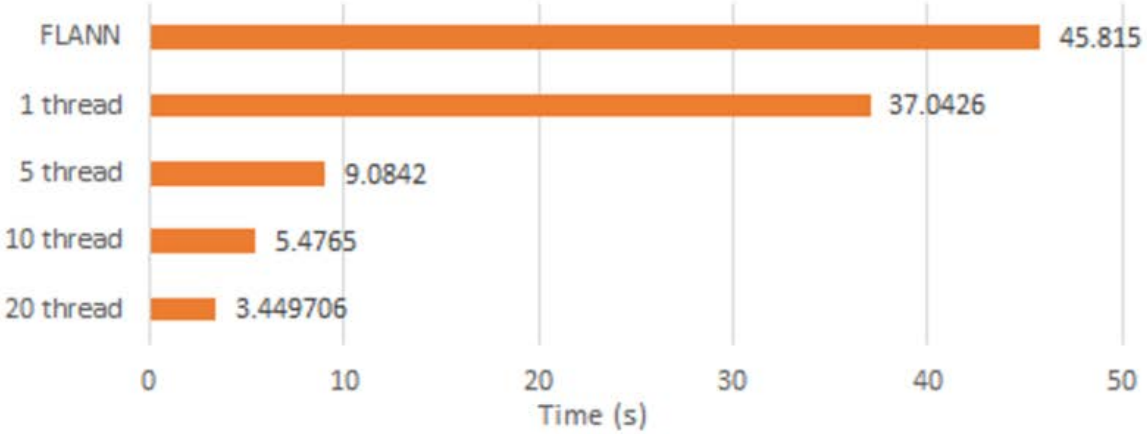


Image reference: http://images.slideplayer.com/21/6323701/slides/slide_6.jpg

Satisfying Results



- **Contributions:**
  - Working distributed Image retrieval using Distributed Nearest Neighbor Search Algorithm
  - Perform better than OpenCV FLANN library
  - Scalable compared to OpenCV FLANN library
  - Implement HLSH on general distributed LSH platform
  - Implement C2LSH on general distributed LSH platform

2

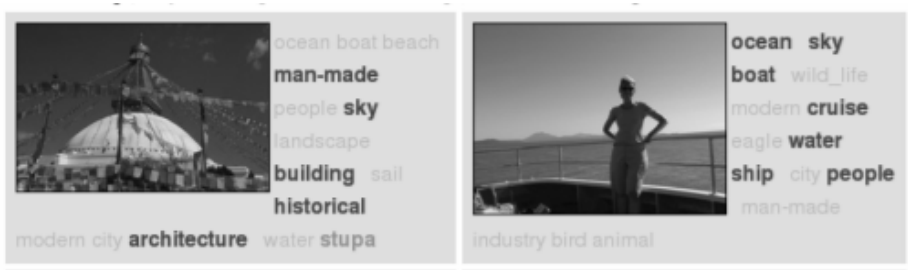
Background and Theory

Building upon others

Background

- **Content-based image retrieval (CBIR)**
- Focus on low-level image information
- ie. The color and shapes in images

- **Concept-based image retrieval**
- Focus on high-level image concepts
- ie. A boy riding a horse



Source: Image Retrieval: Ideas, Influences, and Trends of the New Age, RITENDRA DATTA, DHIRAJ JOSHI, JIA LI, and JAMES Z. WANG, The Pennsylvania State University https://goo.gl/ozdXc2

Nearest Neighbor Search is our focus

- Concept-based image retrieval
- Kd-tree?
  - Only low dimension : bad at high dimension pruning
  - NP – hard for comparing tree for exact search

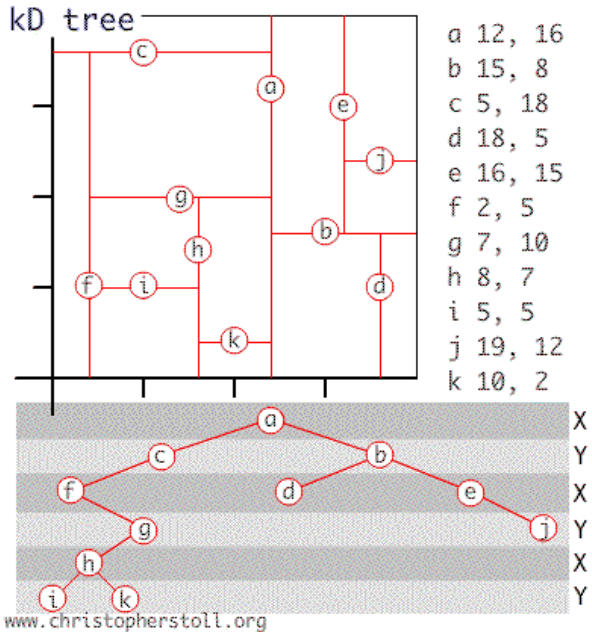
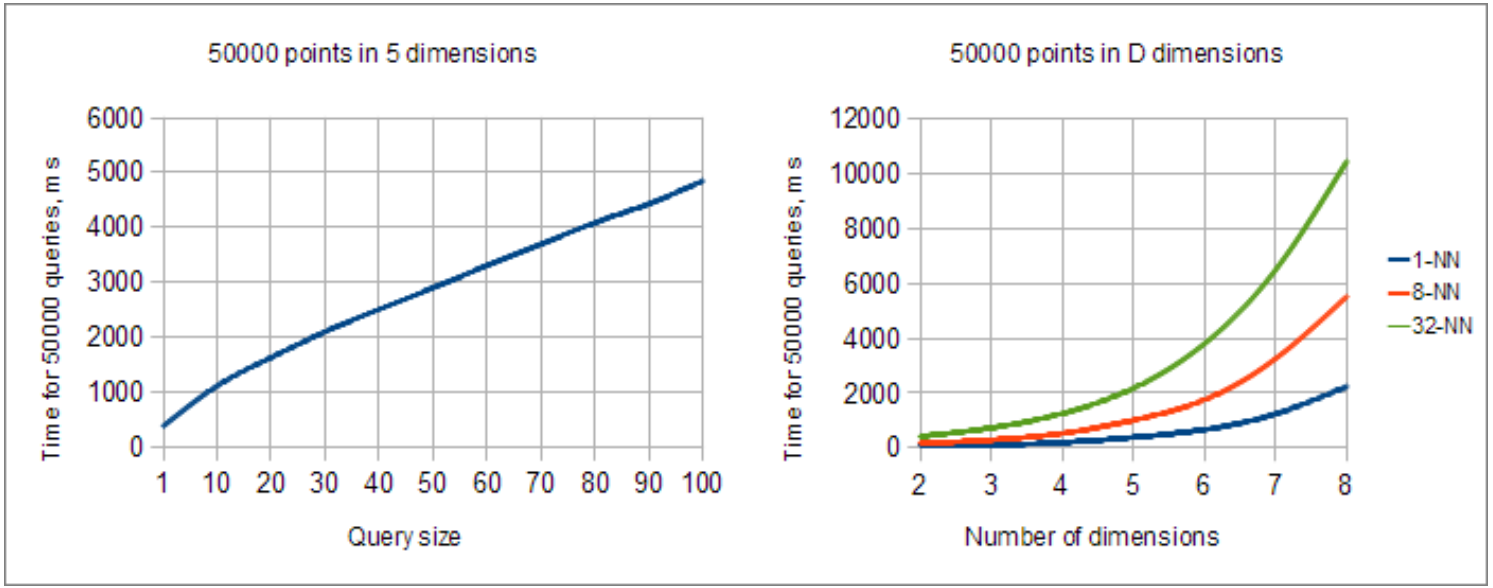


Image reference: Internet image

Nearest Neighbor Search is our focus



Experiment Reference: http://www.alglib.net/other/nearestneighbors.php

Nearest Neighbor Search is our focus

- Concept-based image retrieval
- Kd-tree?
  - Only low dimension : bad at high dimension pruning
  - NP – hard for comparing tree for exact search

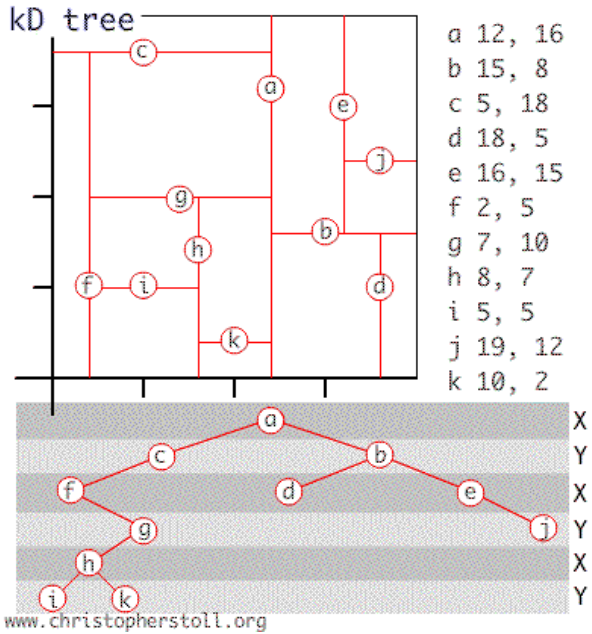


Image reference: file:///D:/Program%20Files/Downloads/kD-tree_02.gif

Nearest Neighbor Search is our focus

- Concept-based image retrieval
- Kd-tree?
  - Only low dimension : bad at high dimension pruning
  - NP – hard for comparing tree for exact search
- Approximate Nearest Neighbor Search

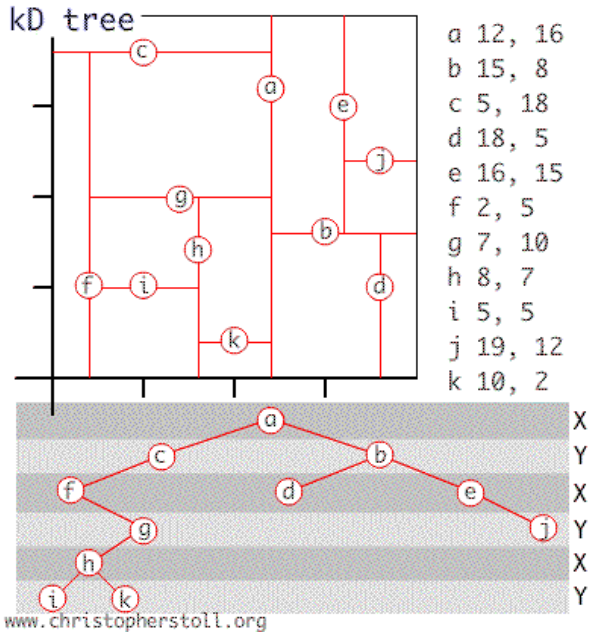


Image reference: file:///D:/Program%20Files/Downloads/kD-tree_02.gif

Locality Sensitive Hashing (LSH)

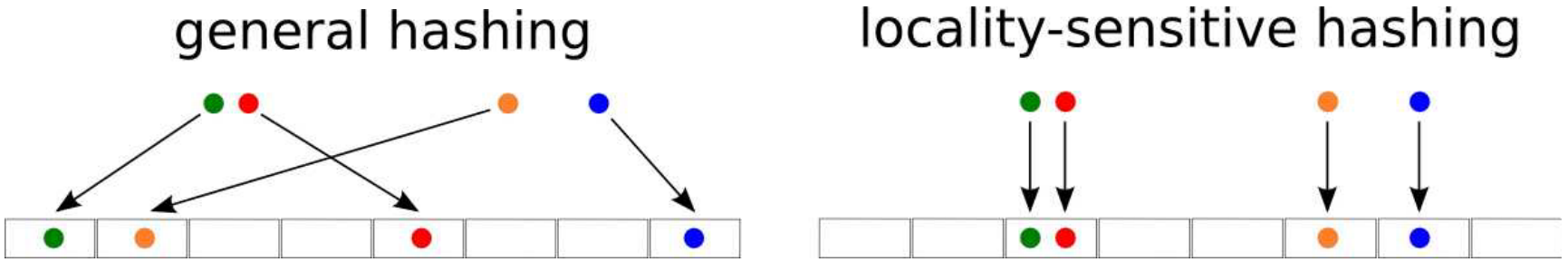
Definition 1 *Ball* $B_p(\vec{q}, r)$ *: In d-dimensional space* R^d*, under distance metric* l_p*,* B_p *is centered at point* $\vec{q}$ *with radius r such that :* $BallB_p(\vec{q}, r) = \{\vec{v} \in R^d \mid l_p(\vec{r}, \vec{q}) \leq r\}$

Only when the following conditions were met can this family function be called *(r, cr, p1, p2)*-sensitive:

Definition 2 (Locality-sensitive Hashing)

1. $if\ d(q,r) \leq r,\ then\ P_r[h(p) = h(q)] \geq p_1$
2. $if\ d(q,r) > cr,\ then\ P_r[h(p) = h(q)] \leq p_2$
3. $c \geq 1$ *and*
4. $p_1 > p_2$

Locality Sensitive Hashing (LSH)



$$sim(X,Y) = P\{h(x) = h(y)\}$$

Image reference:https://0110.be/files/attachments/416/general_vs_lsh.jpg

Husky

- Open Source Distributed Platform
- Object-oriented
- Load-balancing and fault tolerance
- Messaging paradigm
- Map-reduce friendly

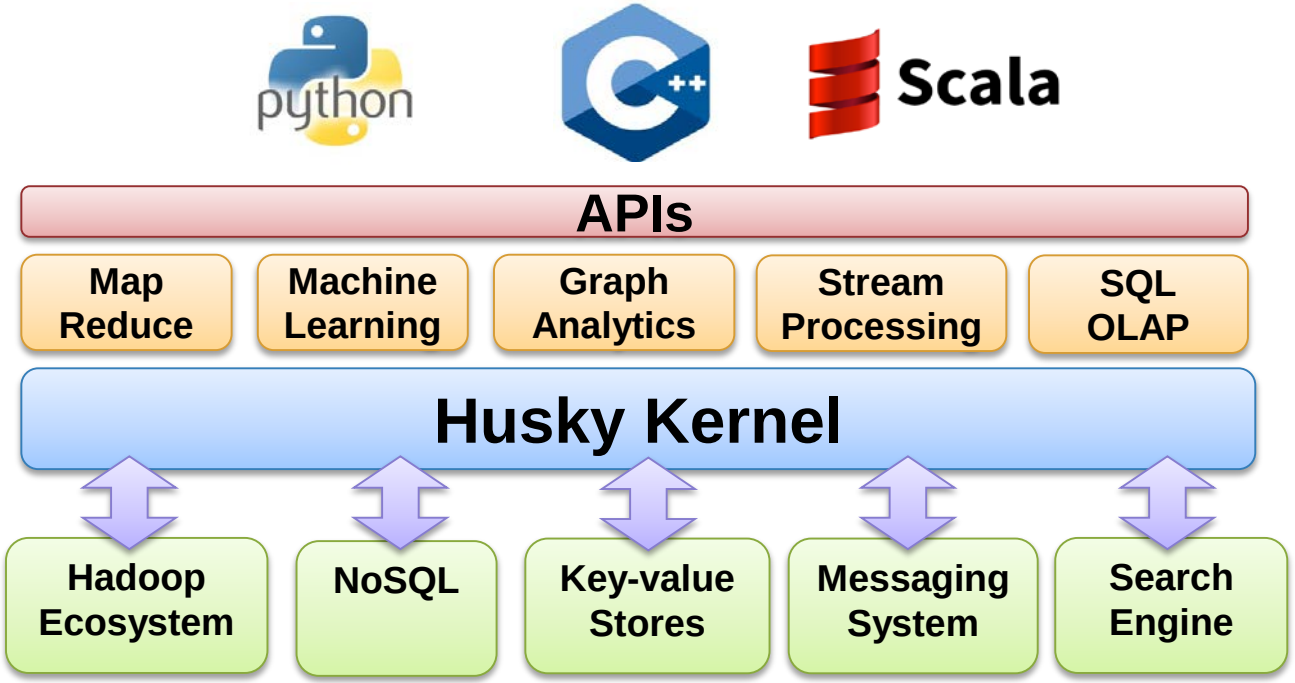


Image from CSCI5570 Lecture

3

Overall Design

This is how it works

Image Retrieval

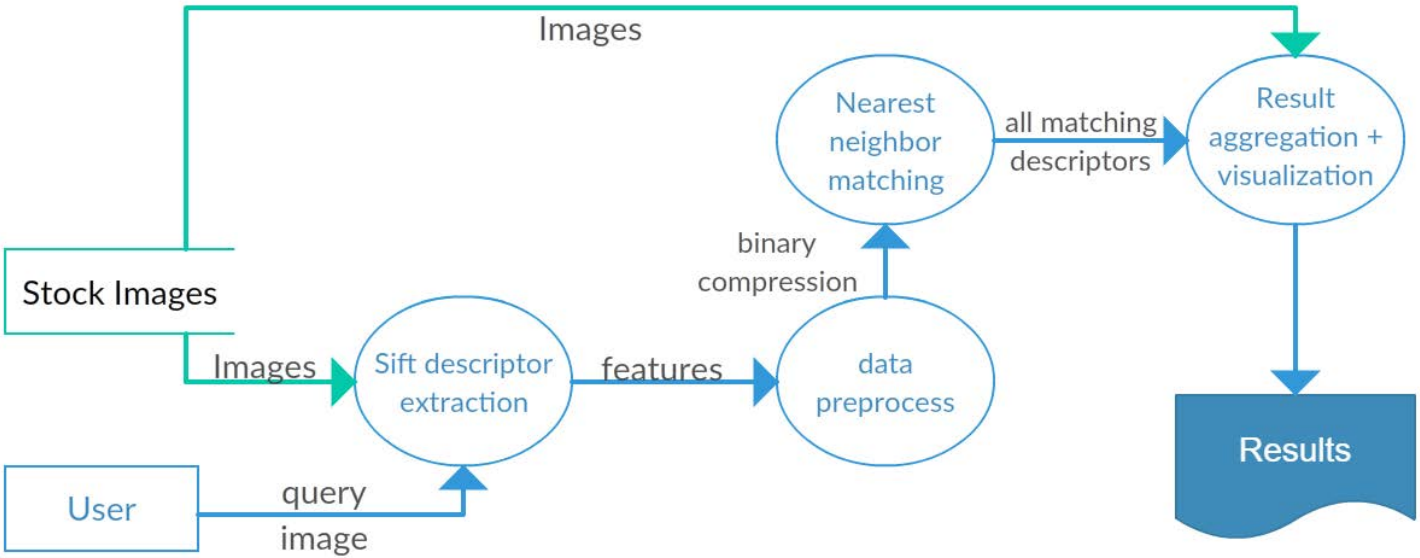


Image Retrieval

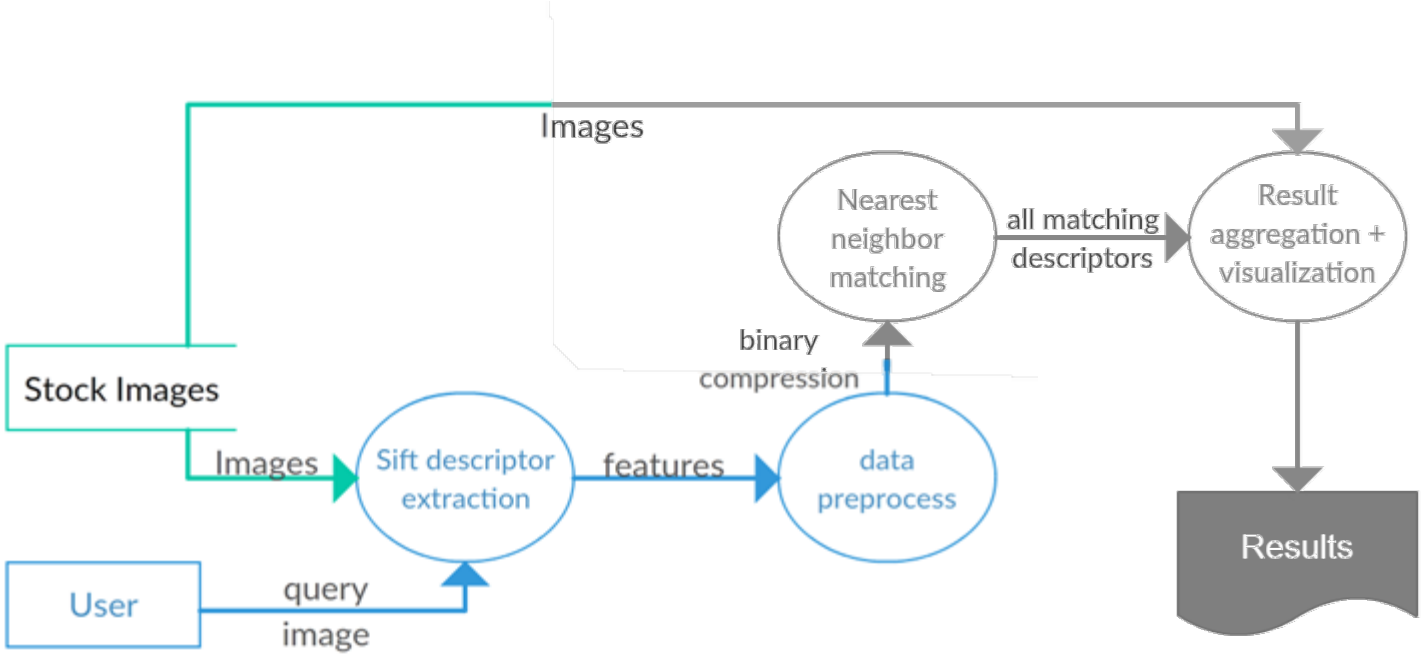
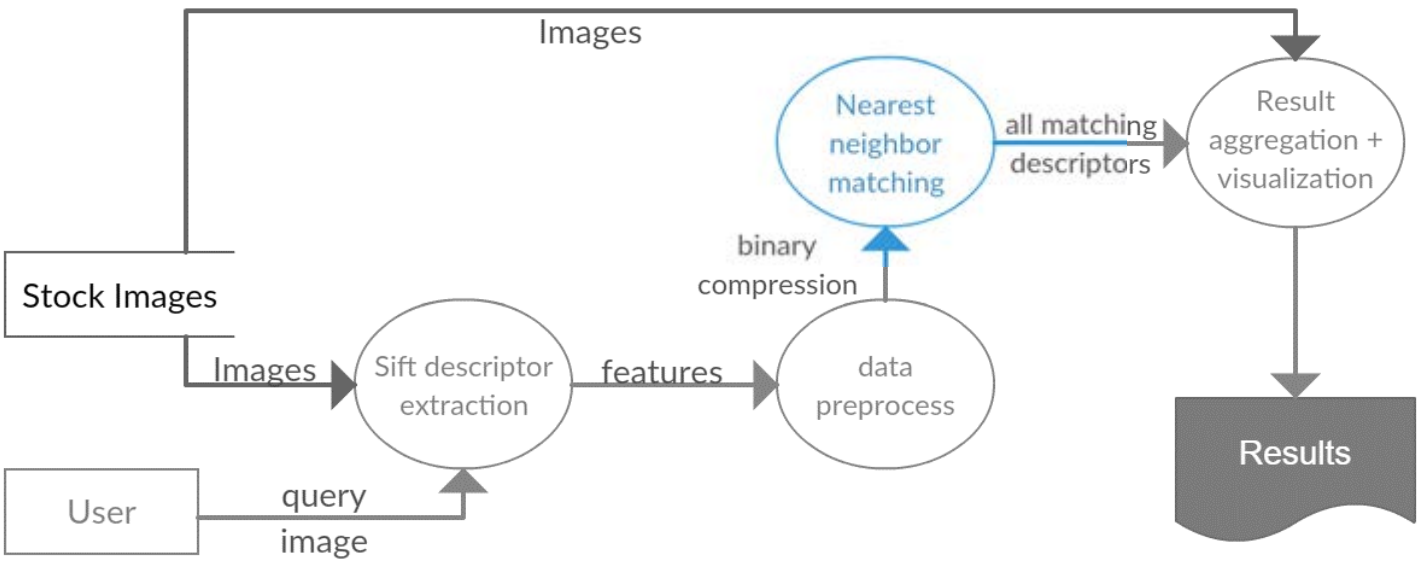


Image to Data

Image Retrieval

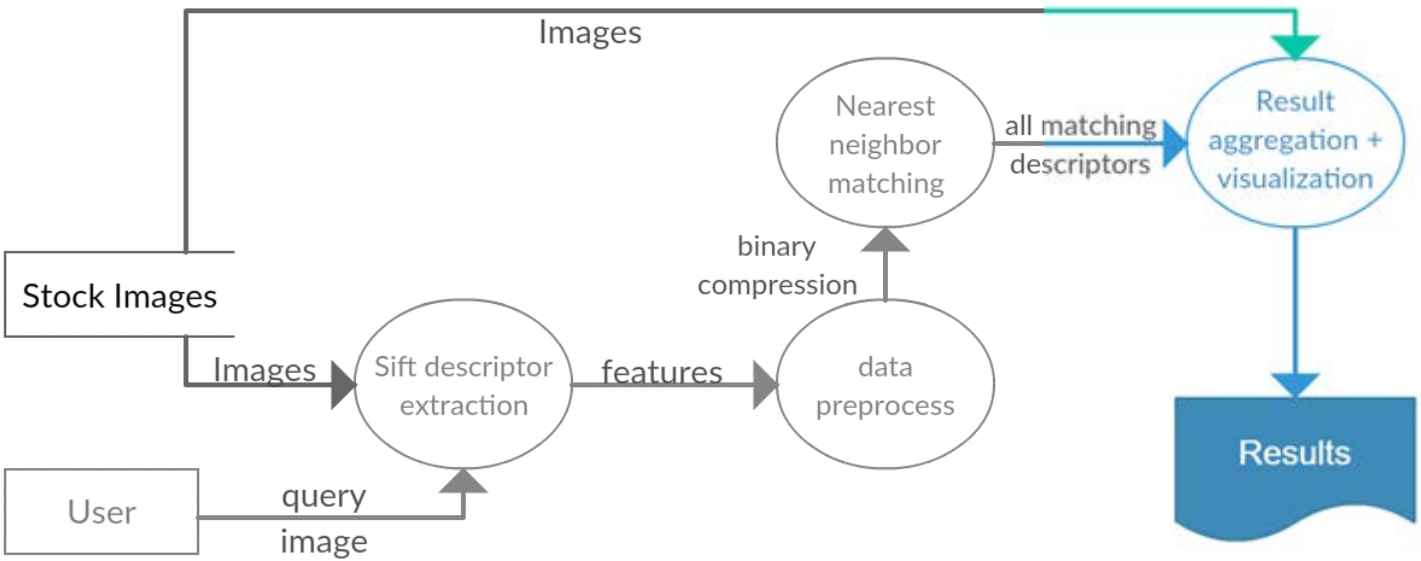


Nearest Neighbor Matching

I2D

NNS

Image Retrieval



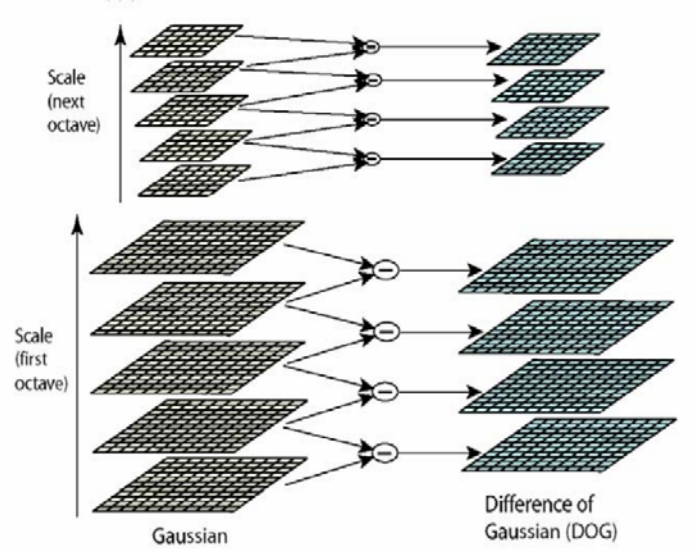
I2D

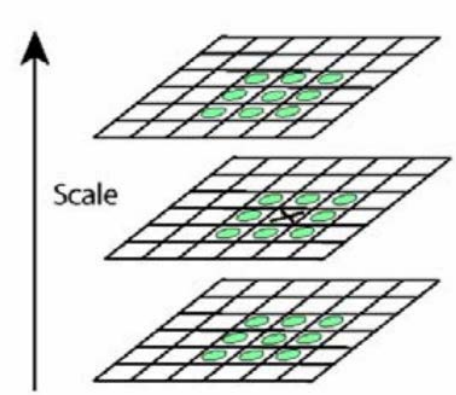
NNS

R2I

Results to Image

SIFT (Scale Invariant Feature Transform)





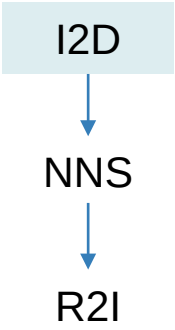
I2D

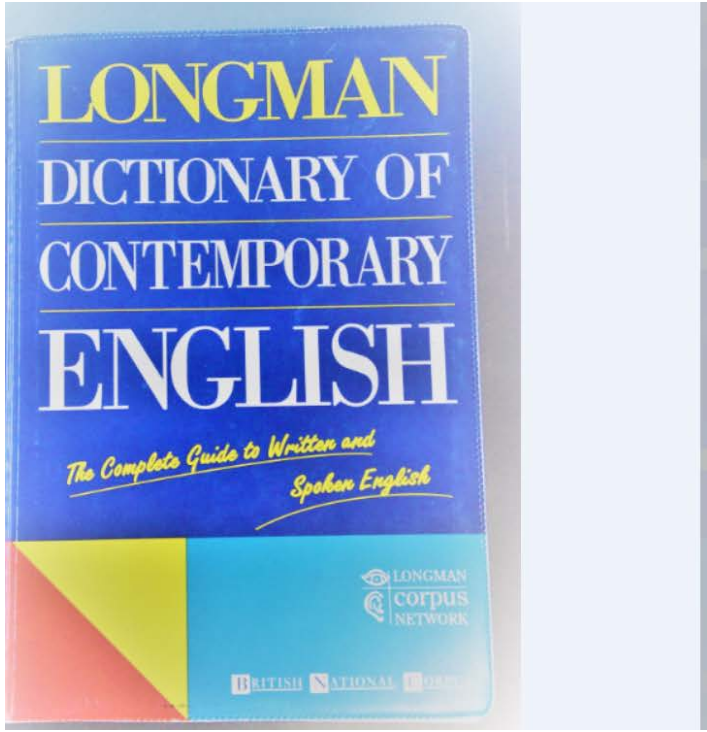
NNS

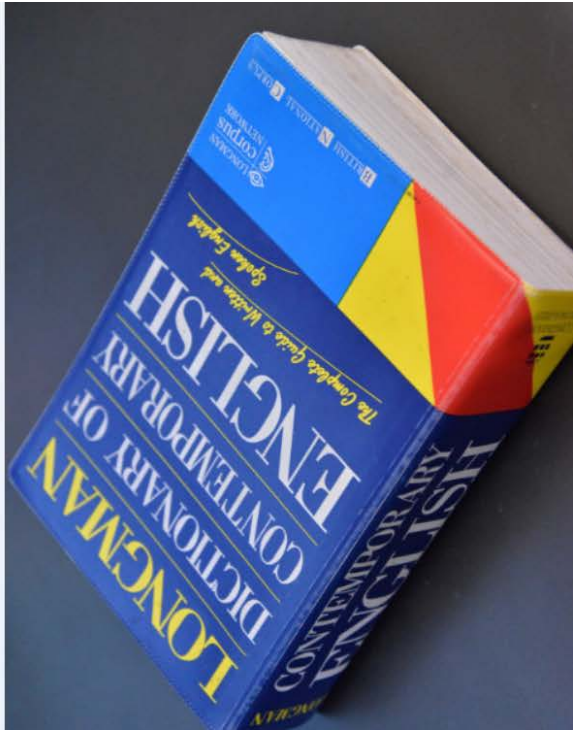
R2I

SIFT (Scale Invariant Feature Transform)

- Robust : perspective change
- Locality preserving
- Large Quantity




LONGMAN
DICTIONARY OF
CONTEMPORARY
ENGLISH
The Complete Guide to Written and Spoken English
LONGMAN
corpus
NETWORK


LONGMAN
DICTIONARY OF
CONTEMPORARY
ENGLISH

I2D

↓

NNS

↓

R2I

I2D

↓

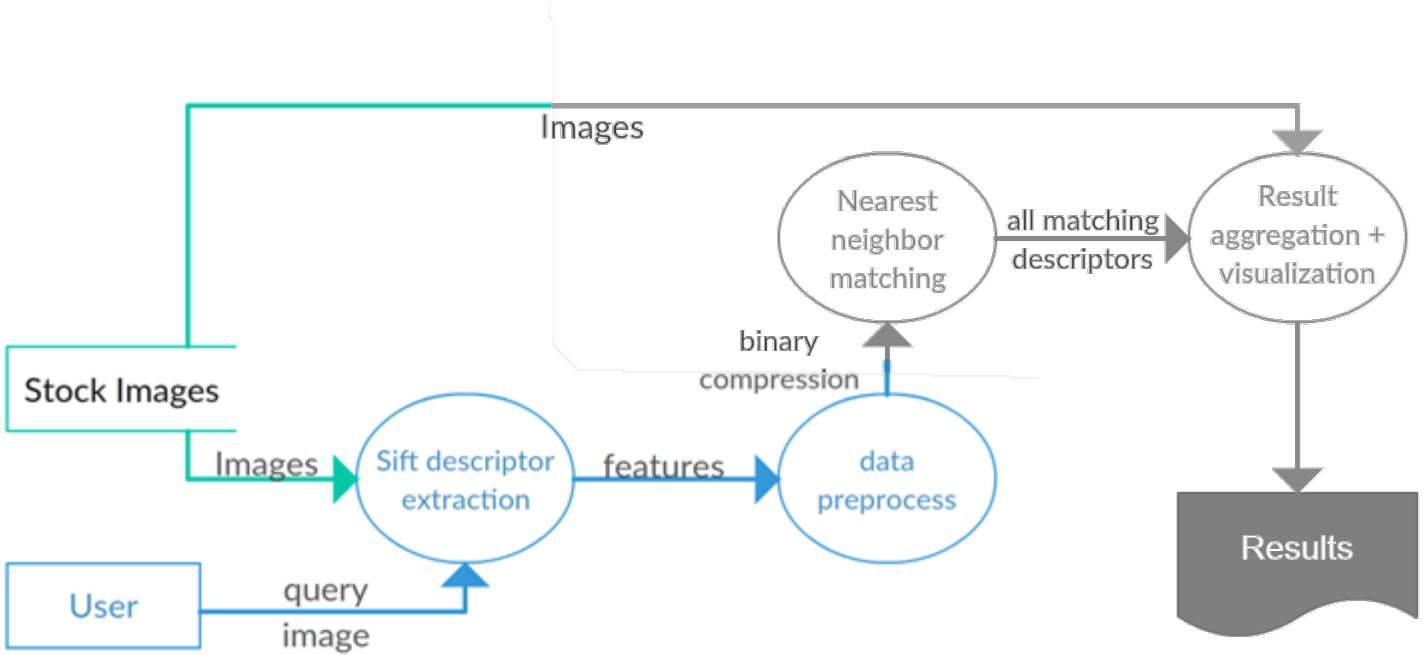
NNS

↓

R2I

Data Format: (ID + FileID) Descriptor*128

Image Retrieval



I2D

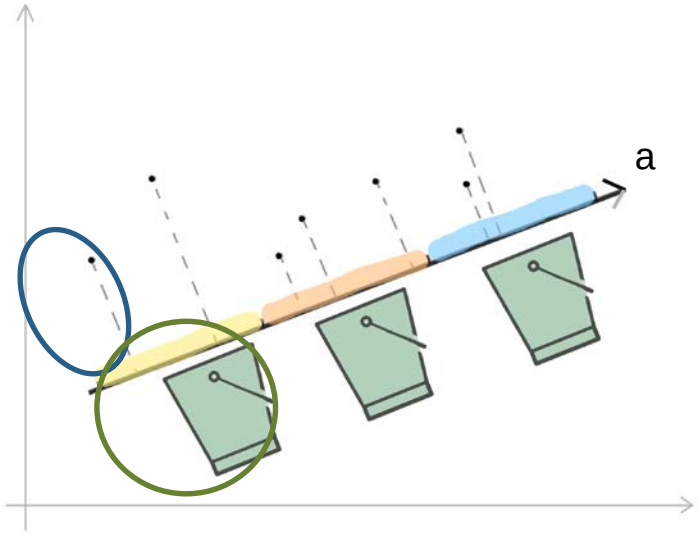
NNS

R2I

Image to Data

E2LSH (Exact Euclidean LSH)

$$h_{a,b}(v) = \left\lfloor \frac{a \cdot v + b}{w} \right\rfloor$$



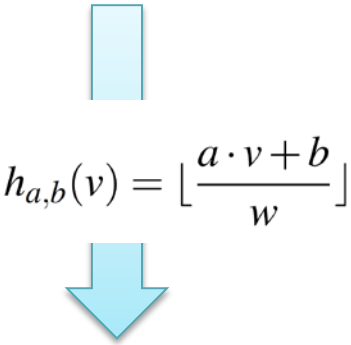
I2D

NNS

R2I

Data 1	Des. 1	Des. 2	Des. 3	Des. 4	Des. 5	Des. 6	Des. 7
Data 2	Des. 1	Des. 2	Des. 3	Des. 4	Des. 5	Des. 6	Des. 7
Data 3	Des. 1	Des. 2	Des. 3	Des. 4	Des. 5	Des. 6	Des. 7

	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
Function 1	Sig. 1	Sig. 2	Sig. 3	Sig. 4	Sig. 5	Sig. 6	Sig. 7
Function 2	Sig. 1	Sig. 2	Sig. 3	Sig. 4	Sig. 5	Sig. 6	Sig. 7
Function 3	Sig. 1	Sig. 2	Sig. 3	Sig. 4	Sig. 5	Sig. 6	Sig. 7



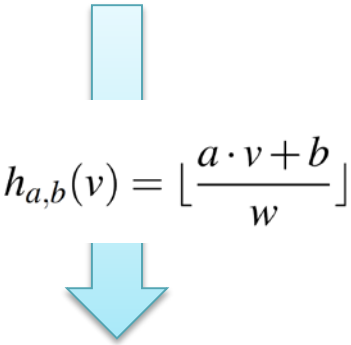
I2D

NNS

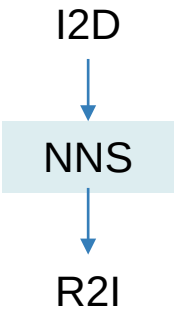
R2I

Data 1	Des. 1	Des. 2	Des. 3	Des. 4	Des. 5	Des. 6	Des. 7
Data 2	Des. 1	Des. 2	Des. 3	Des. 4	Des. 5	Des. 6	Des. 7
Data 3	Des. 1	Des. 2	Des. 3	Des. 4	Des. 5	Des. 6	Des. 7

	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
Function 1	Sig. 1	Sig. 2	Sig. 3	Sig. 4	Sig. 5	Sig. 6	Sig. 7
Function 2	Sig. 1	Sig. 2	Sig. 3	Sig. 4	Sig. 5	Sig. 6	Sig. 7
Function 3	Sig. 1	Sig. 2	Sig. 3	Sig. 4	Sig. 5	Sig. 6	Sig. 7



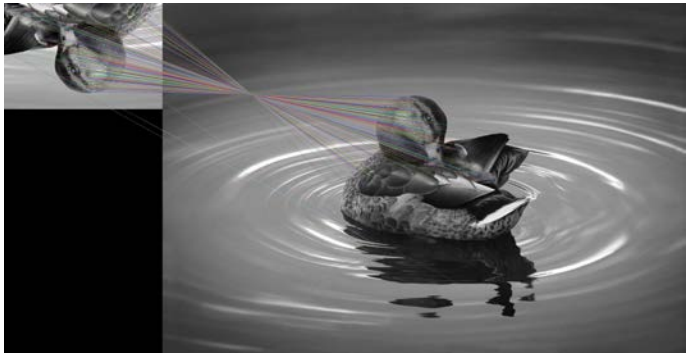
	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
Function 1	Sig. 1	Sig. 2	Sig. 3	Sig. 4	Sig. 5	Sig. 6	Sig. 7
Function 2	Sig. 1	Sig. 2	Sig. 3	Sig. 4	Sig. 5	Sig. 6	Sig. 7
Function 3	Sig. 1	Sig. 2	Sig. 3	Sig. 4	Sig. 5	Sig. 6	Sig. 7
Function 4	Sig. 1	Sig. 2	Sig. 3	Sig. 4	Sig. 5	Sig. 6	Sig. 7



And - Or operation : (1, 4) , (3, 7)

Matching Images

```
2     7568 1290602 4.472136
3     7660 1300002 3.316625
4     7762 1310302 6.403124
5     7940 1335302 6.403124
6     7966 1342302 4.358899
7     8041 1349502 4.582576
8     8095 1354902 4.795832
9     3176 2179303 47.053162
10    4924 2397503 43.046486
11    4918 2397503 49.203659
12    3101 1259909 34.044090
```



I2D

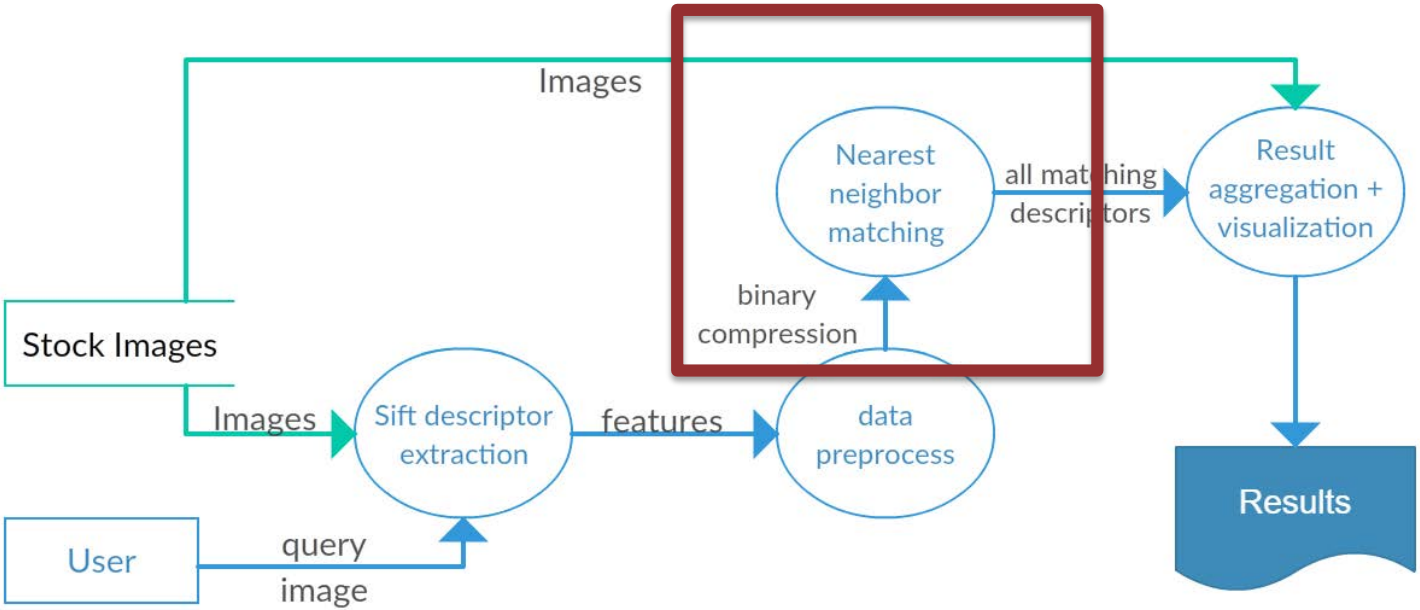
↓

NNS

↓

R2I

Image Retrieval



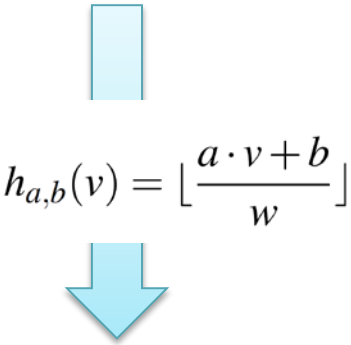
4

Distributing the design

Upgrade Avaliable

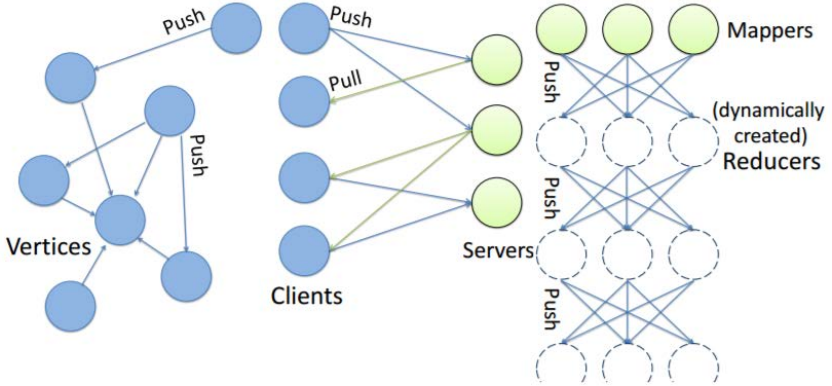
Data 1	Des. 1	Des. 2	Des. 3	Des. 4	Des. 5	Des. 6	Des. 7
Data 2	Des. 1	Des. 2	Des. 3	Des. 4	Des. 5	Des. 6	Des. 7
Data 3	Des. 1	Des. 2	Des. 3	Des. 4	Des. 5	Des. 6	Des. 7

	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
Function 1	Sig. 1	Sig. 2	Sig. 3	Sig. 4	Sig. 5	Sig. 6	Sig. 7
Function 2	Sig. 1	Sig. 2	Sig. 3	Sig. 4	Sig. 5	Sig. 6	Sig. 7
Function 3	Sig. 1	Sig. 2	Sig. 3	Sig. 4	Sig. 5	Sig. 6	Sig. 7

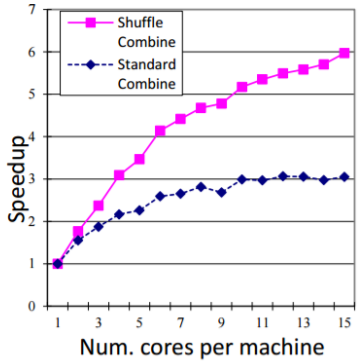


Husky

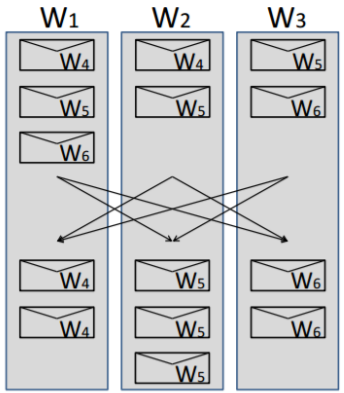
- The Husky Approach: three interaction pattern
- Objects and messaging
- Speedup shuffle combiner



(a) Pregel (b) Parameter Server (c) MapReduce Chain



(a) Effect of combiner designs

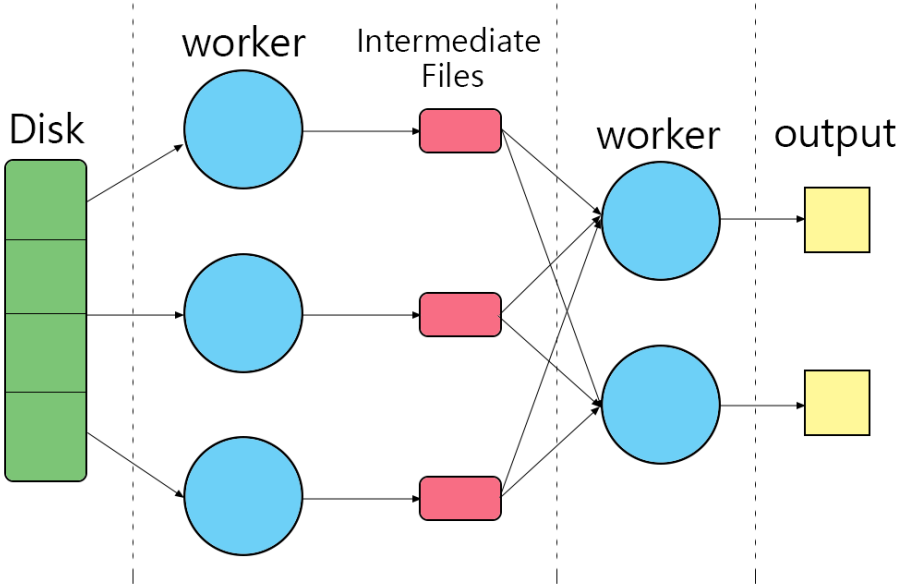


(b) Shuffle combiner

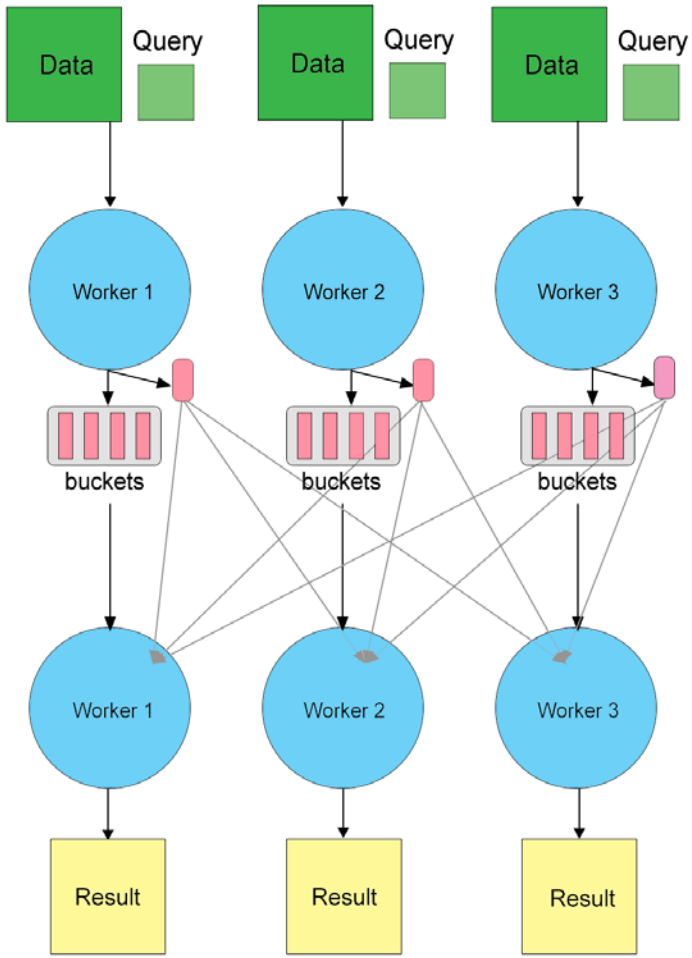
Reference: Husky: Towards a More Efficient and Expressive Distributed Computing Framework; Fan Yang, Jinfeng Li, James Cheng

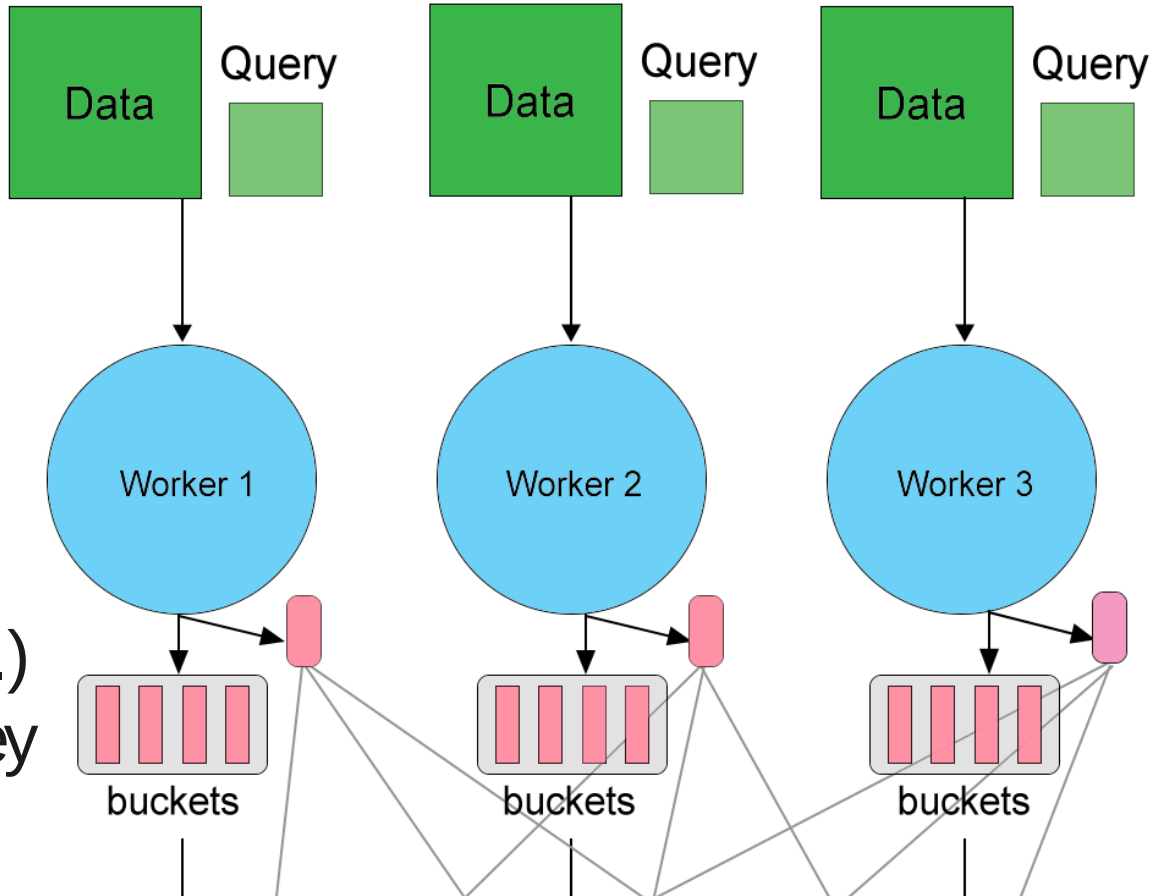
Distributing E2LSH

Map-Reduce



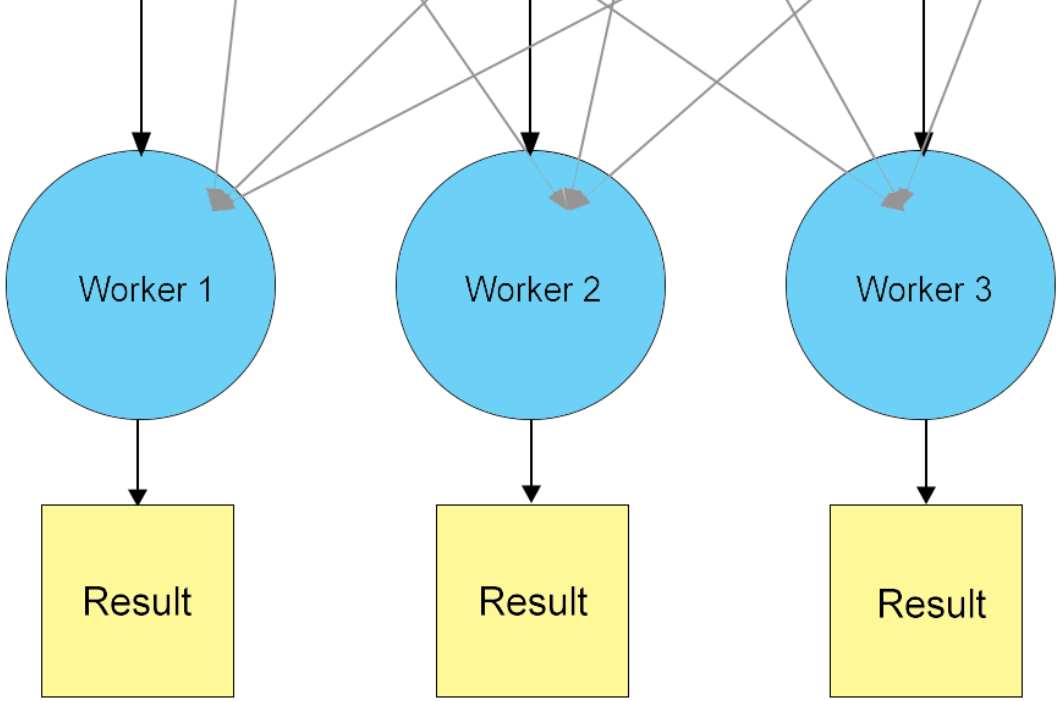
Distributed E2LSH





(Sig., Fn.)
Is the Key

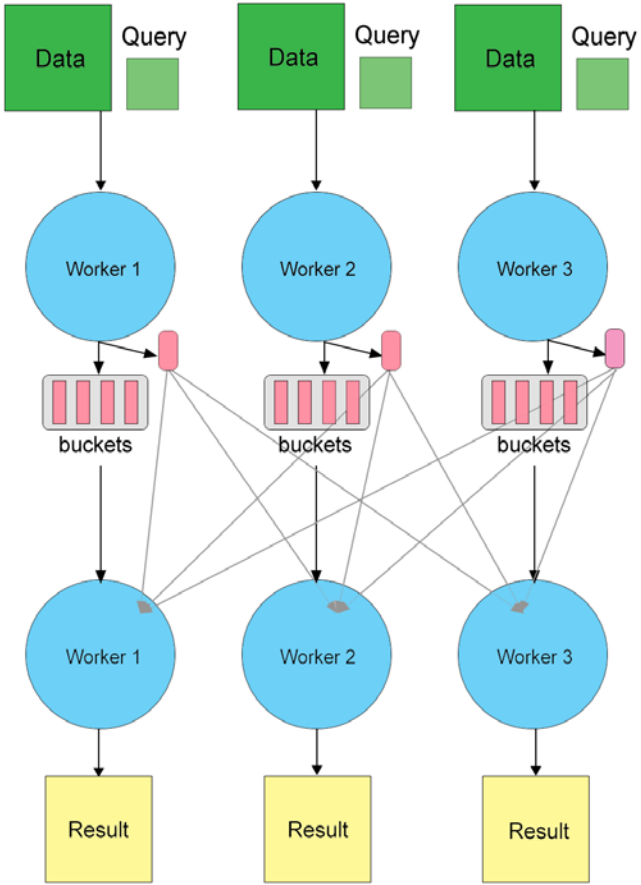
Function
Calculation



Aggregate the same key

Calculate Candidates

- We move only the hash value of Query
- Scalable



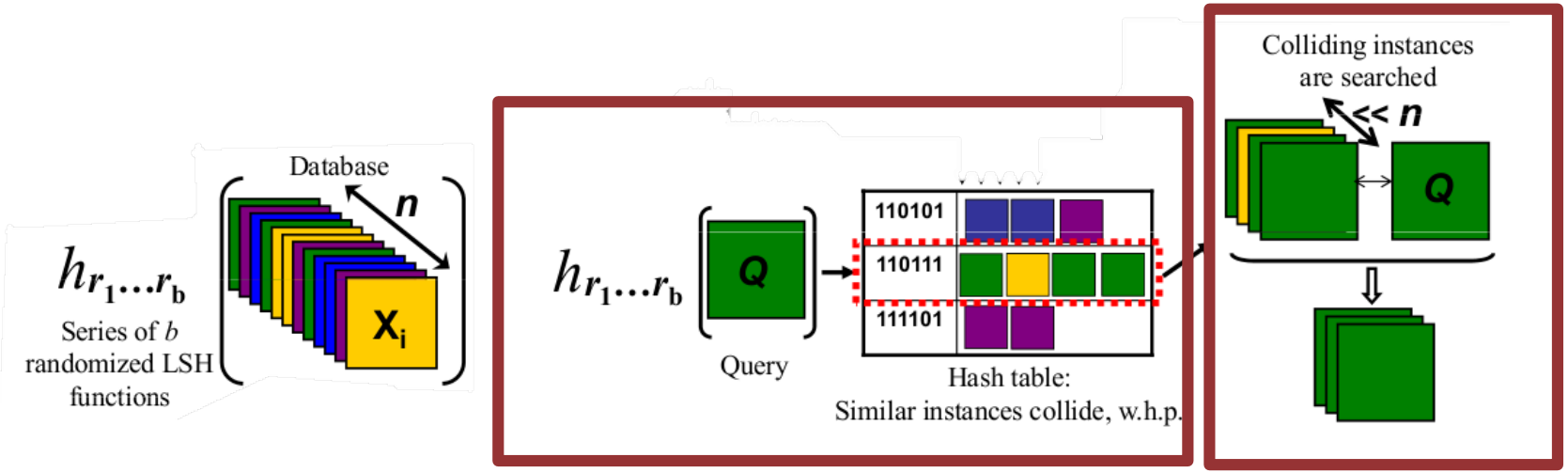
1. Initialization
2. Calculate hash values
3. Create the key/buckets
4. Calculate Candidate pairs

5

A Generalized Framework

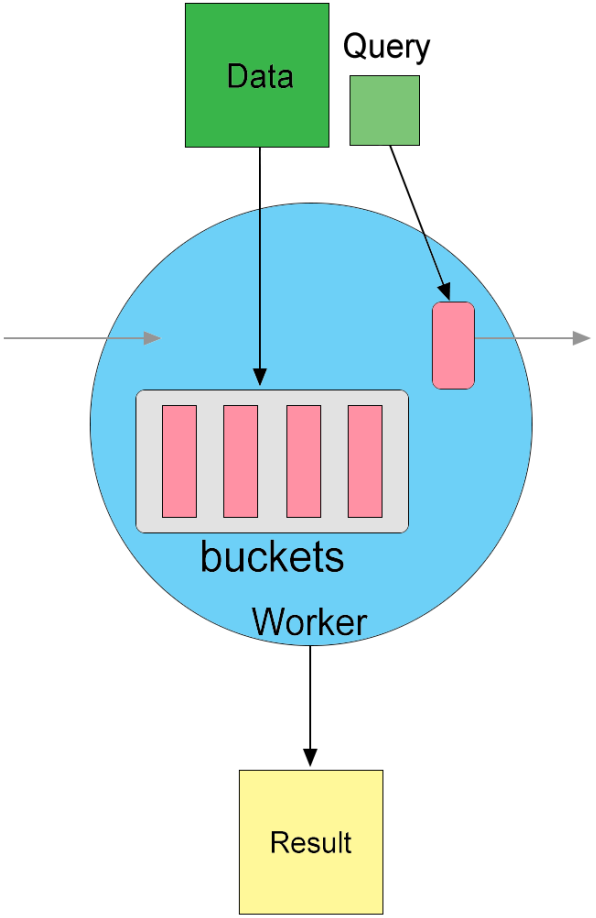
Now we can do more

Distributed E2LSH



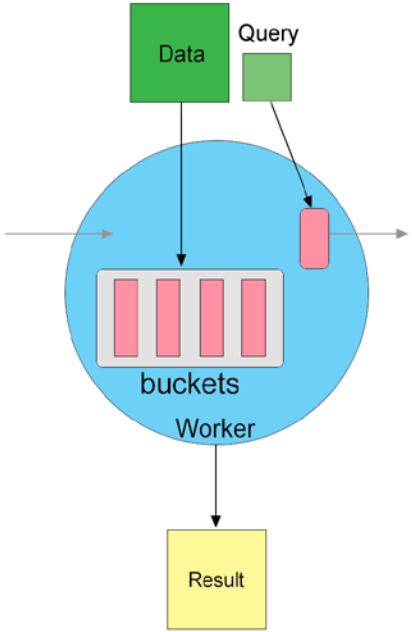
- Hash method
- Distance metric
- Search method
- Termination decision

Image reference: https://micvog.files.wordpress.com/2013/08/lsh1.png



1. Initialization
2. Create hash functions
3. Calculate hash values
4. Create the key/buckets
5. Calculate Candidate pairs
6. Decide termination

Hamming Distance Locality Sensitive Hashing (HLSH)

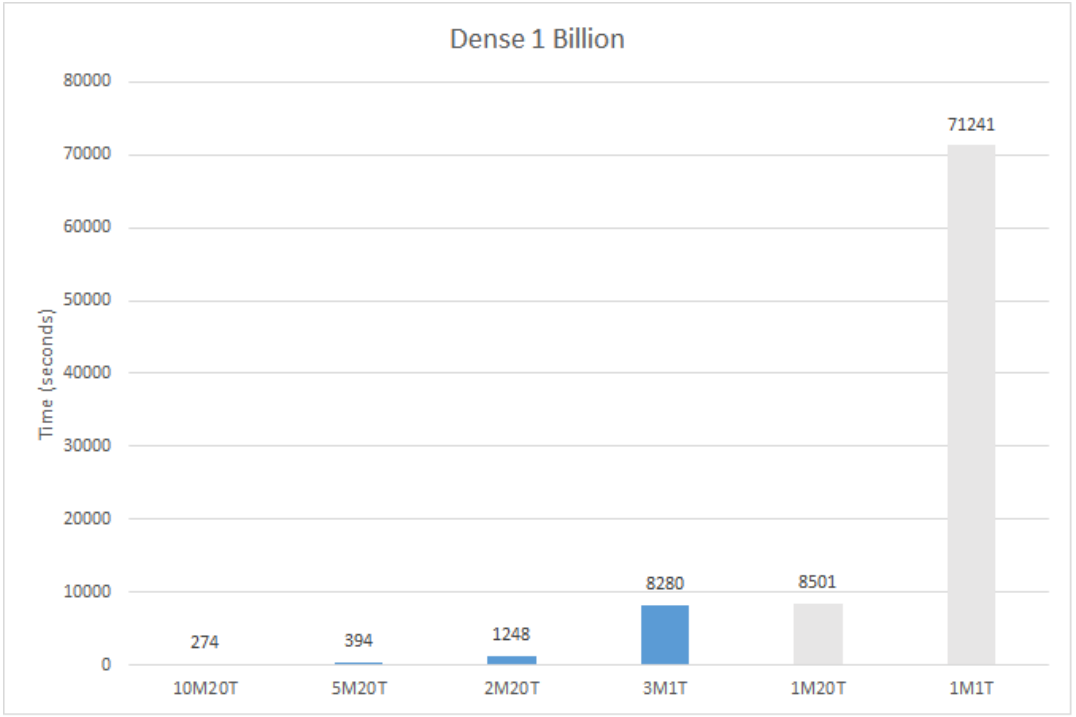


$$d(x,y) = \Sigma_{i=1}^{n} |x_i - y_i|$$

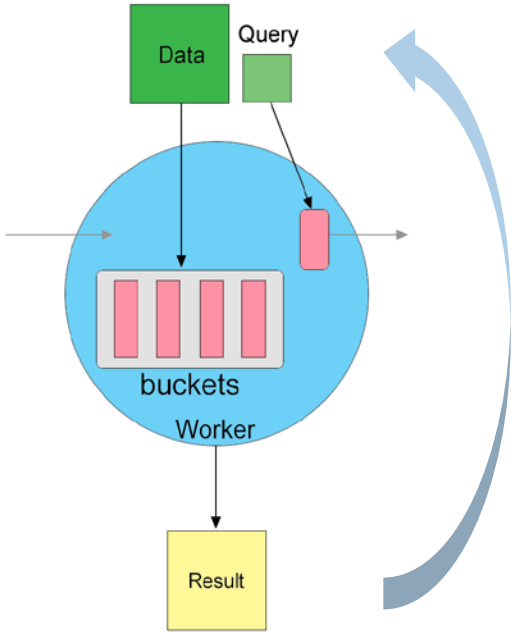
$$NN(x) = arg\min_{y \epsilon Y} (\Sigma_{i=1}^{n} |x_i - y_i|)$$

1. Initialization
2. **Create hash functions**
3. Calculate hash values
4. Create the key/buckets
5. Calculate Candidate pairs
6. Decide termination


Efficient Nearest Neighbor
Search in Distributed Manner


Dense 1 Billion
Time (seconds)
80000
70000
60000
50000
40000
30000
20000
10000
0
274
394
1248
8280
8501
71241
10M20T
5M20T
2M20T
3M1T
1M20T
1M1T

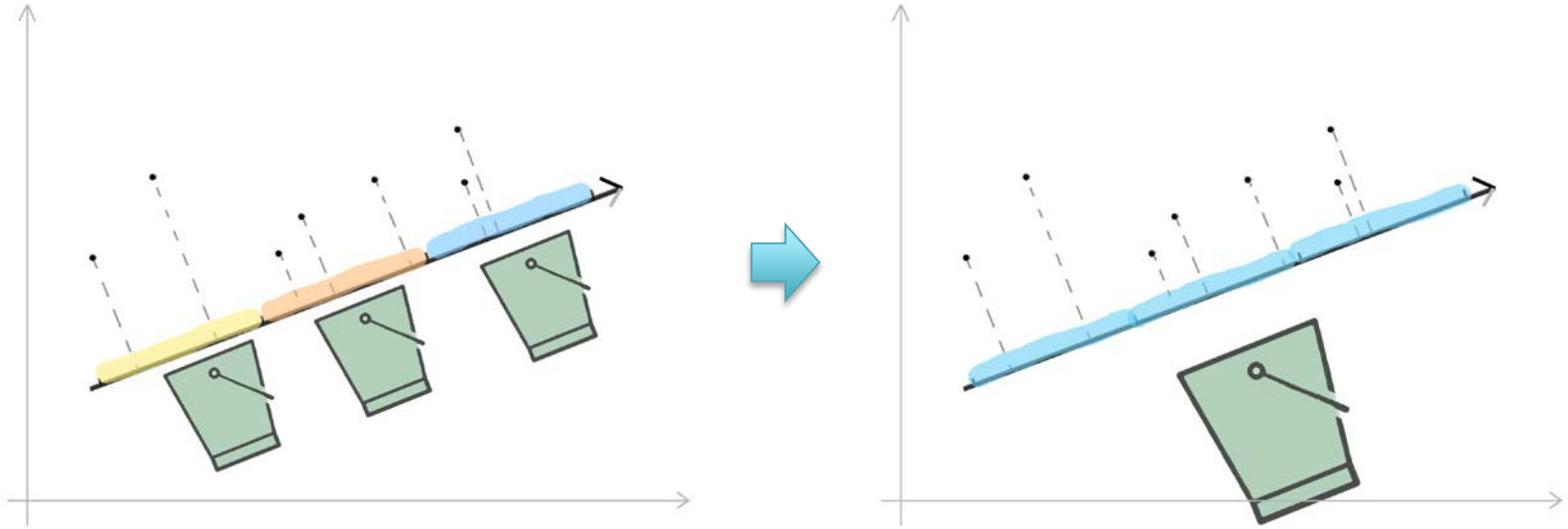
Collision Counting Locality Sensitive Hashing (C2LSH)



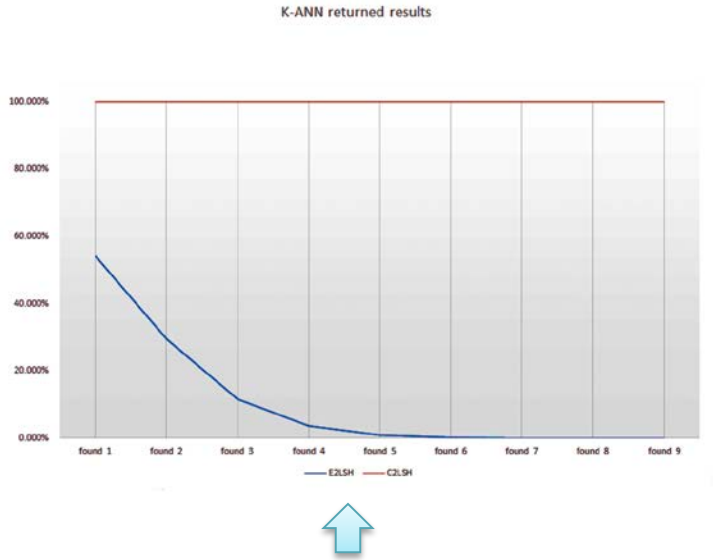
$$h_{a,b}(v) = \lfloor \frac{a \cdot v + b}{w} \rfloor$$

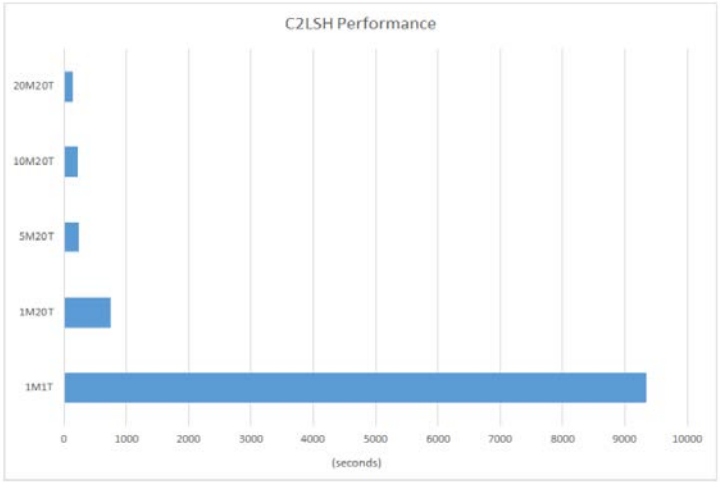
$$H^R(o) = \lfloor \frac{h(o)}{R} \rfloor$$

1. Initialization
2. **Create hash functions**
3. Calculate hash values
4. Create the key/buckets
5. Calculate Candidate pairs
6. **Decide termination**



C2LSH


K-ANN returned results
100.000%
80.000%
60.000%
40.000%
20.000%
0.000%
found 1
found 2
found 3
found 4
found 5
found 6
found 7
found 8
found 9
E2LSH
C2LSH


C2LSH Performance
20M20T
10M20T
5M20T
1M20T
1M1T
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
(seconds)

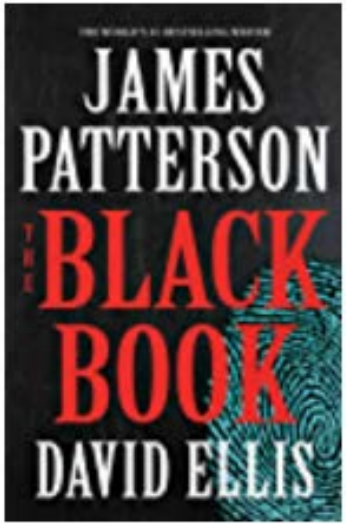
6

Experiments and Results

It’s pretty fast, and scalable!

Experiments

- Stock images + Query Images
- Experiment 1: simple correctness test, small dataset
- Experiment 2: Robustness test, medium dataset
- Experiment 3: Scalability test, large dataset




000.jpg


001.jpg


002.jpg

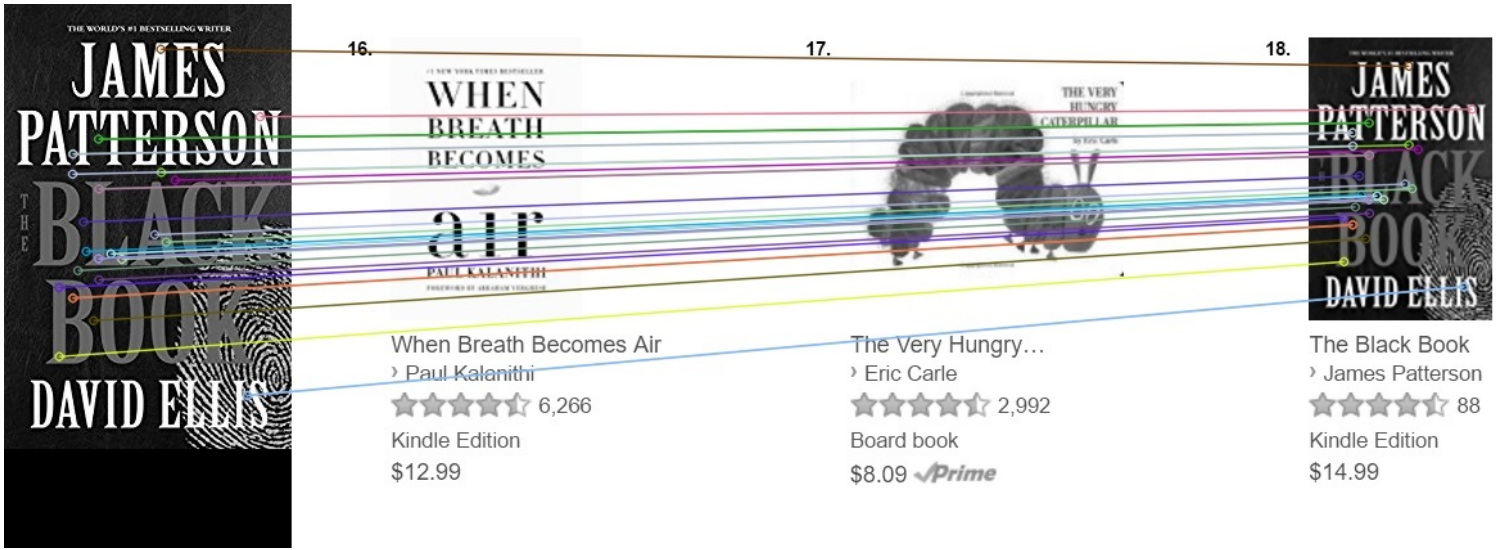

003.jpg


004.jpg


005.jpg

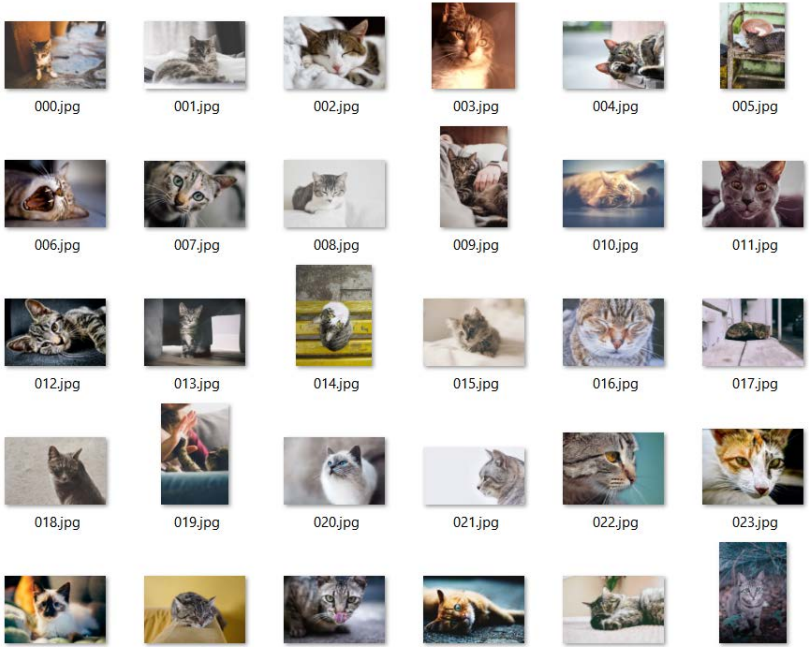

006.jpg

18940 SIFT points, 9.4MB, 0.49 seconds

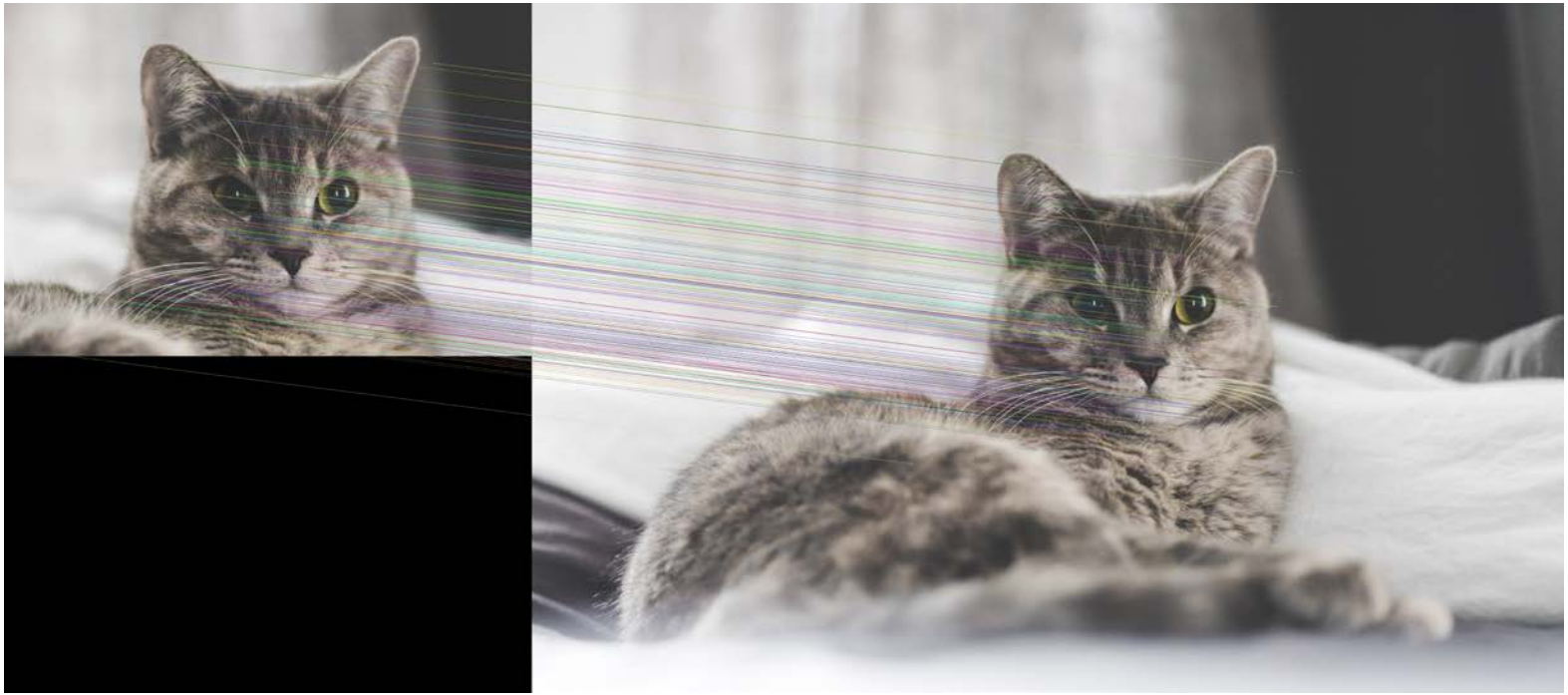

16.
When Breath Becomes Air
› Paul Kalanithi
6,266
Kindle Edition
$12.99
17.
The Very Hungry…
› Eric Carle
2,992
Board book
$8.09 Prime
18.
The Black Book
› James Patterson
88
Kindle Edition
$14.99

Experiment – Robustness

- 30 high quality images
- Around 440 MB
- Cats: very similar item test



Experiment 2



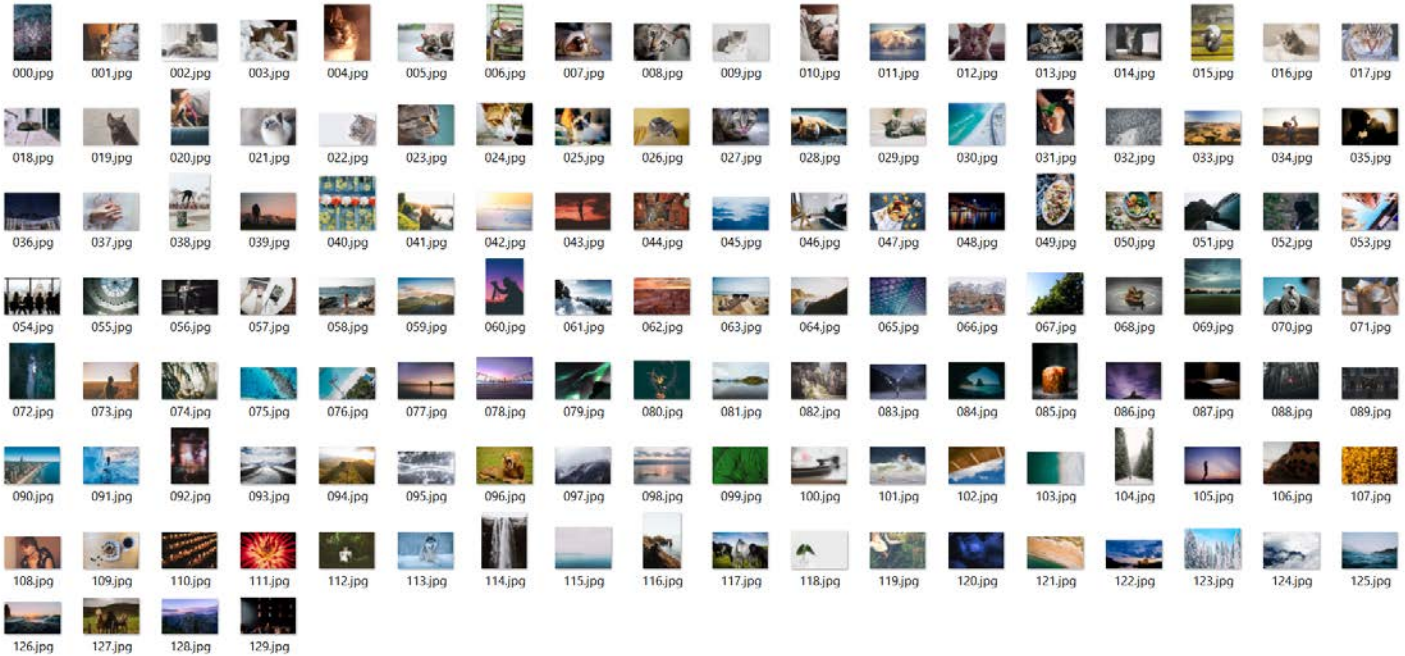
Experiment: OpenCV FLANN

FLANN (Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration)

2009 Marius Muja, David G. Lowe. 2000 citations

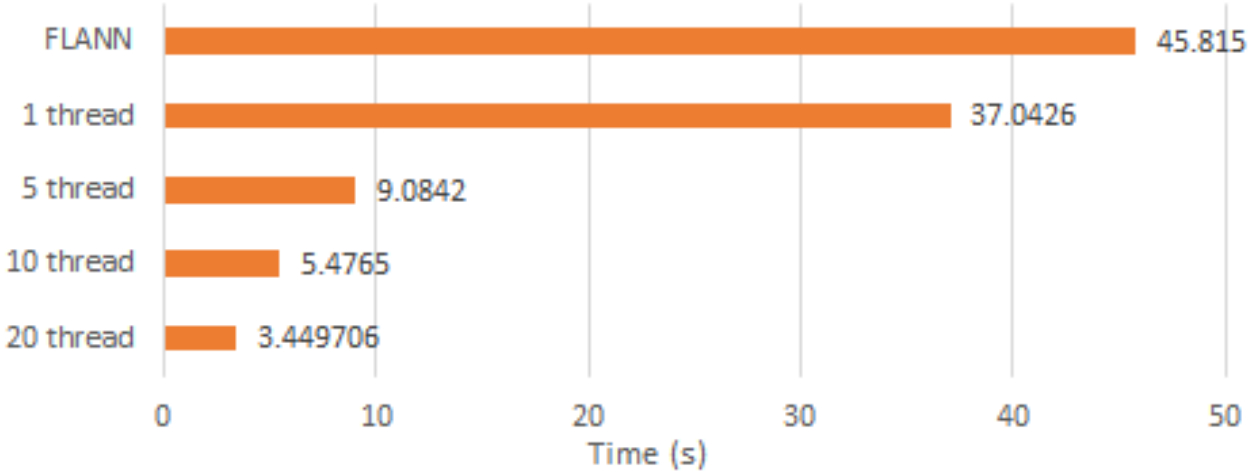
- Compare NNS time
- Prove scalability
- Prove performance

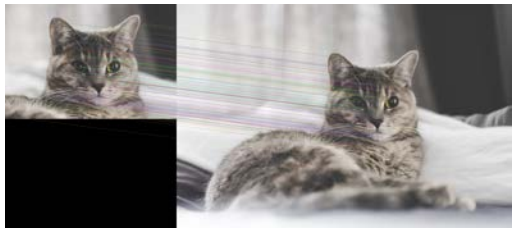
Experiment 3 - 1



- 130 high quality images
- 4 Million Data vs. 8500 query
- Medium dataset
- Single machine Multi Thread

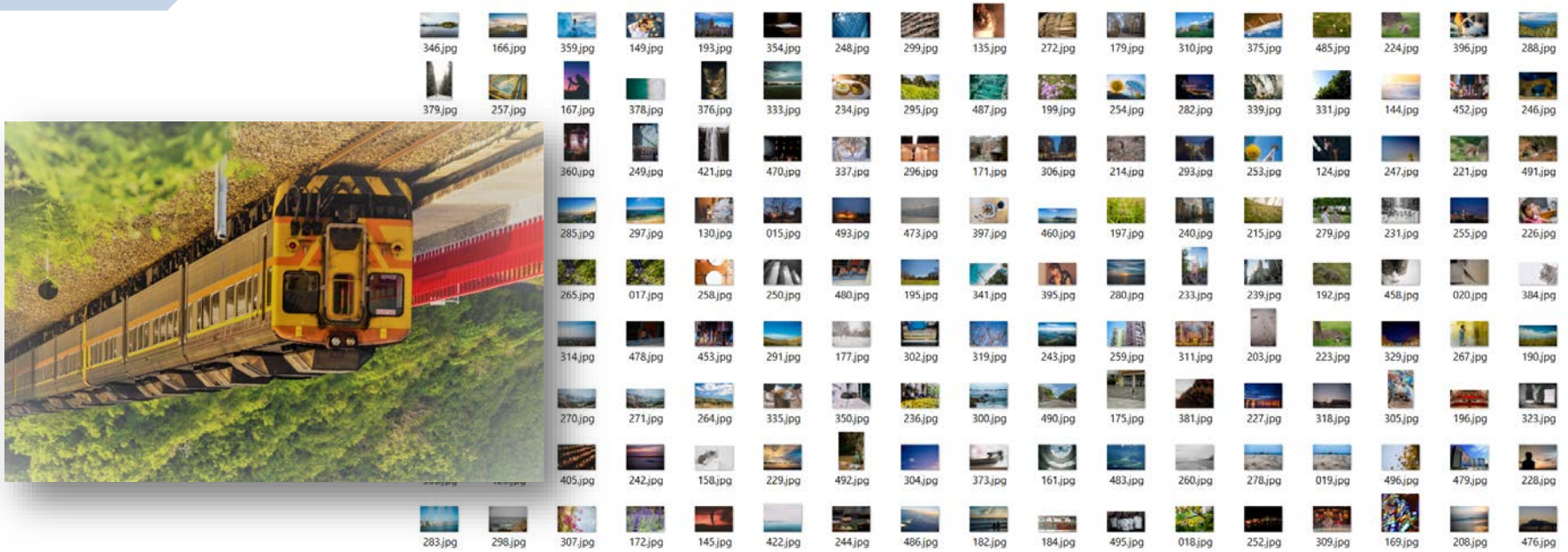
Experiment 3 - 1





- 130 high quality images
- 4 Million Data vs. 8500 query
- Medium dataset
- Single machine Multi Thread

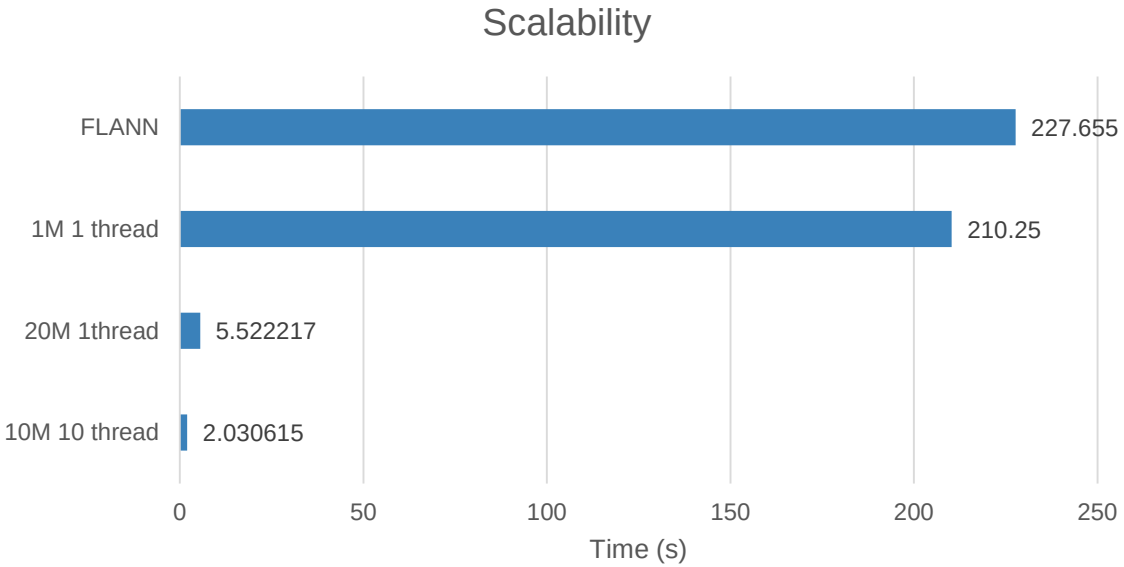
Experiment 3 - 2



- 500 high quality images
- 13.5 Million Data
- Huge dataset

Experiment 3 - 2

Experiment 3 - 2



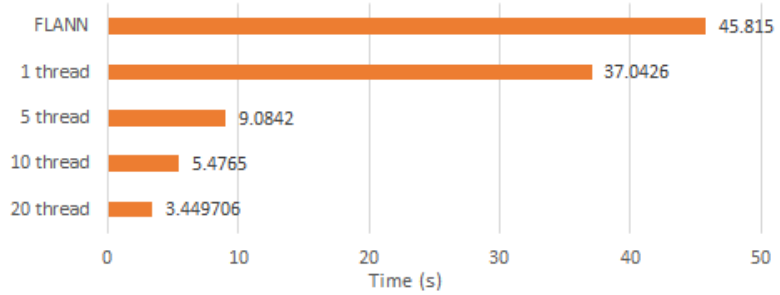
- 1/100 time possible
- Distributed setup = Scalability

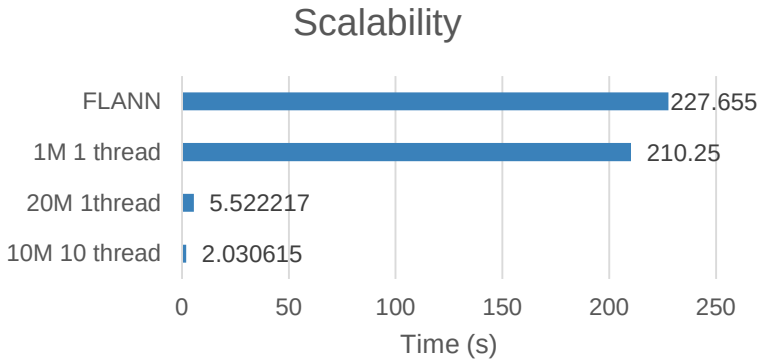
7

Discussion and Conclusion

Not the end, but the start

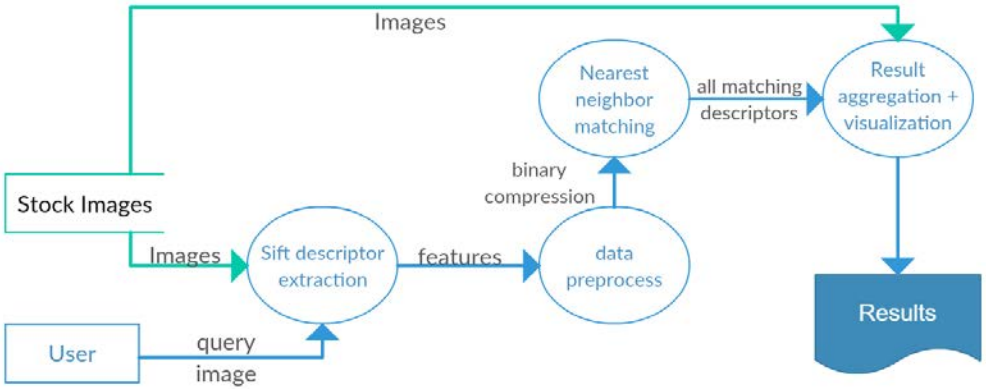
Discussion

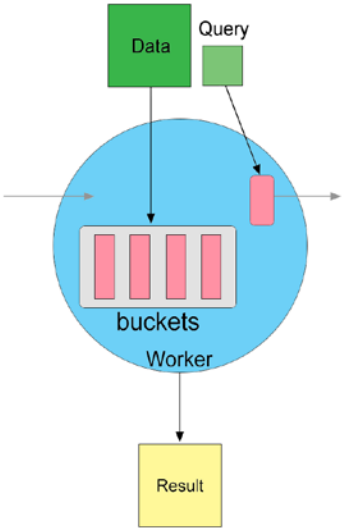




- It is pretty robust
- Very efficient

Conclusion


Images
Nearest neighbor matching
all matching descriptors
Result aggregation + visualization
binary compression
Stock Images
Images
Sift descriptor extraction
features
data preprocess
Results
User
query image


Data
Query
buckets
Worker
Result

Conclusion

- Working distributed Image retrieval using Distributed Nearest Neighbor Search Algorithm
- Performance: better than OpenCV FLANN library
- Scalability: better than OpenCV FLANN library
- Implement HLSH, C2LSH on general distributed LSH platform
- Open-sourcing in near future

Efficient Nearest Neighbor Search in Distributed Manner

Cheng, Ti-Chung (1155043421) Supervised by Prof. James Cheng
The Chinese University of Hong Kong, CSE Department

References
Acknowledgements
Appendix

Some thoughts and supplumentary

References

[1] Jingdong Wang, Heng Tao Shen, Jingkuan Song, and Jianqiu Ji. Hashing for similarity search: A survey. Aug 13, 2014.

[2] Mohammad Reza Abbasifard, Bijan Ghahremani, and Hassan Naderi. A survey on nearest neighbor search methods. *International Journal of Computer Applications*, 95(25), Jan 1, 2014.

[3] Jon Bentley. Multidimensional binary search trees used for associative searching, Sep 1, 1975.

[4] Alexandr Andoni and Piotr Indyk. E2lsh 0.1 user manual.

[5] Wang Weihong and Wang Song. A scalable content-based image retrieval scheme using locality-sensitive hashing. volume 1, pages 151–154, 2009.

[6] Hsin-Liang Chen and Edie M Rasmussen. Intellectual access to images. *Library Trends*, 48(2):291, Oct 1, 1999.

[7] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov 2004.

[8] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. page 604613, New York, NY, USA, 1998. ACM.

[9] Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Semi-supervised hashing for scalable image retrieval. pages 3424–3431, 2010.

[10] Fan Yang, Jinfeng Li, and James Cheng. Husky: Towards a more efficient and expressive distributed computing framework. *Proc. VLDB Endow.*, 9(5):420431, January 2016.

[11] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. SCG '04, pages 253–262. ACM, Jun 8, 2004.

[12] Jeffrey Dean and Sanjay Ghemawat. Mapreduce, Jan 1, 2008.

[13] Ti-Chung Cheng. Efficient nearest neighbor search in distributed manner. Final year project midterm report., dec 2016.

[14] Junhao Gan, Jianlin Feng, Qiong Fang, and Wilfred Ng. Locality-sensitive hashing scheme based on dynamic collision counting. SIGMOD '12, pages 541–552. ACM, May 20, 2012.

[15] Ti-Chung Cheng. Implementation and analysis of collision counting lsh on the husky framework. Distributed C2LSH on husky for CUHK 2016 summer research project, aug 2016.

[16] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application VISSAPP'09)*, pages 331–340. INSTICC Press, 2009.

[17] Eliezer S. Silva and Eduardo Valle. K-medoids lsh: a new locality sensitive hashing in general metric space, 2013.

Acknowledgments

This Final Year Project has been extremely fruitful. Standing at the crossroad between distributed computing, machine learning, computer vision and data science has been challenging but fun. Without prior knowledge in the four fields, I am excited to learn along the way while getting my hands dirty, implemented different ideas and examined the hypothesis.

With that being said, I am very grateful to have the opportunity to soar with the eagles, test the theoretical and apply the practical. I am extremely thankful to have my adviser, Professor James Cheng, who gave me the opportunity and time to learn and grow. I must also thank my tutor Mr. Jinfeng Li for his knowledge, patience and guidance.

Ti Chung Cheng

April 2017