

# Nearest Neighbour Searching in High Dimensional Metric Space

David Marshall

March 21, 2006

## Abstract

Given one item, finding its closest match within a database of other such items is a task performed in numerous domains. Image matching, data mining, and electroencephalogram data analysis are a few varied examples. The extension of the concept of Euclidean distance in 2D and 3D space to higher dimensional space provides an effective comparison of items in these sorts of domains.

Particular regard will be given to the performance of nearest neighbour searching in a large database of SIFT descriptors. (Scale Invariant Feature Transform) SIFT descriptors are useful in support of many image matching tasks.

There are a number of algorithms, each with their own issues of storage size and search performance. The literature review (COMP6720) will aim to describe the significant algorithms and their performance attributes. The review should also identify opportunities for the enhancement (or enhanced implementation) of existing algorithms for the purposes of computer vision. Such further work would be the subject of the (COMP6702) follow-on project.

## 1 Objectives

The nearest neighbour approach to finding similarities between images is a significant tool in the field of computer vision. Its effective application can contribute to many worthy uses of computer vision, such as:

- Safety (identifying drowning in a pool)
- Security (facial recognition)
- Health (medical imaging and diagnostics)

This literature review will focus on the application of nearest neighbour searching to the field of computer vision.

The first objective of this review is to discover what are the most significant algorithms used for this purpose. Through understanding and describing these algorithms, it is expected that some opportunities for improvement will be identified. These improvements might be in the form of a new algorithm, a modified algorithm, or a more efficient implementation of an algorithm.

The ultimate objective of this review will be to take one or more of these opportunities and develop a plan for exploiting them through further research in next semesters project: COMP6702.

## **2 Initial Reading List**

### **2.1 Background**

[1, 5, 7, 2]

### **2.2 Data Structures**

[3, 18, 17]

### **2.3 High Dimensions/Large Databases**

[9, 8, 10, 11, 14, 4]

### **2.4 Other**

[12, 6, 13, 15, 16]

## **3 Timetable**

The simple timeline for this project is expected to be as follows:

- 9 March: Introduce the project
- 10 March: Submit project topic and plan
- 21 March: Re-submit project topic and plan
- 24 March: Review 5 articles.
- 31 March: Review 5 articles.
- 7 April: Review 5 articles.
- 14 April: Review 5 articles.
- 21 April: Review 5 articles and submit draft report to supervisor
- 28 April: Review 5 articles and redraft.
- 5 May: Review 5 articles and redraft.
- 12 May: Review 5 articles and redraft.
- 19 May: Review 5 articles and redraft.
- 26 May: Redraft
- 2 June: Submit final report
- 5-11 June: Publicly present report

## 4 Background

Nearest neighbour searching is the problem of being given a point in space and finding the nearest other point from a set of points. Simple two-dimensional examples include:

- finding the nearest airport to a plane, for an emergency landing
- finding the nearest pizzeria to a customers home, for a pizza delivery

## 5 Nearest Neighbour in Computer Vision

A common sub-task of many more complex computer vision problems is that of matching an interesting portion or patch from an image to its closest match in a set of other patches. The means of identifying and extracting these patches is not the subject of this review. Comparing them to find the closest match is.

Means of comparison such as averaging or summing the colour value of all pixels in the patch are not very useful for determining similarity in much detail. What is required is a means of comparing all of the pixels to their corresponding pixels in a potential match at the same time. It so happens that if each pixel in the patch is considered as a dimension of space, the patch can be plotted as if it were a point in that space.

In **metric space**, there is a valid concept of distance between points. If we treat two image patches as points in space, we can determine the distance between them. The lesser the distance between them, the more similar they are in appearance. Since it is easy to plot points in two dimensional space on a flat screen or piece of paper, Figure 1 is a good visual example. Three image patches of two pixels each are compared. Representing white/light-grey/dark-grey/black as 0/1/2/3/4 we plot the patches as points. By calculating the distances  $\overline{AB}$  and  $\overline{AC}$ , we can determine that B is closer to A than C is.

## 6 Motivation for High-Dimensional Space

A two-pixel patch, although useful as a visual demonstration, is fairly trivial. Typically, interesting patches of images are much larger. As the number of dimensions increase, however, the concept and means of computing distance between points remains valid. High-dimensional space can, however, have a negative impact on the performance of nearest neighbour searching algorithms that were designed with simpler 2D or 3D space in mind.

As an example of what will be reviewed, a cursory look at some of the existing literature has revealed that the degraded performance caused by increasing dimensions can sometimes be overcome by knowing something about the distribution of points. One dimension might provide greater discrimination than another, for instance.

## 7 Time/Space Complexity

A brute force approach to finding a nearest neighbour would take  $O(n)$  time to evaluate each point in the set. Since there might be millions of points, and

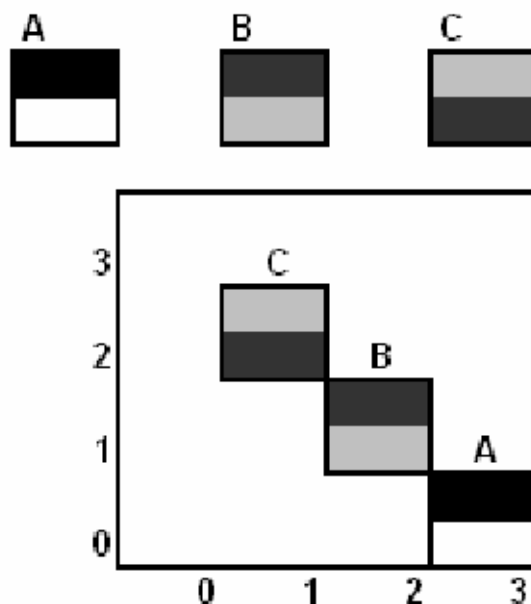


Figure 1: Simple Images Plotted as Points

retrieval might be expensive, time can become a serious issue.

Consider also that a 121-pixel patch in 24bit colour occupies 363 bytes. 1 million patches would take 363MB. And 11 million patches would take 4GB. At this number, available memory could become a serious issue as well.

## References

- [1] *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*, chapter 1. MIT Press, 2006.
- [2] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.
- [3] Christian Böhm, Stefan Berchtold, and Daniel A. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Comput. Surv.*, 33(3):322–373, 2001.
- [4] Sergey Brin. Near neighbor search in large metric spaces. In *Proc. 21st Inter. Conf. on Very Large Data Bases*, pages 574–584, 1995.
- [5] Kenneth Clarkson. Nearest neighbor queries in metric spaces. In *Proc. 39th ACM Symp. Theory Comp.*, pages 609–617, 1997.
- [6] Duncan, Goodrich, and Kobourov. Balanced aspect ratio trees: Combining the advantages of k-d trees and octrees. In *SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)*, 1999.

- [7] Jerome H. Freidman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3(3):209–226, 1977.
- [8] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *The VLDB Journal*, pages 518–529, 1999.
- [9] P. Indyk. Nearest neighbors in high-dimensional spaces. In Jacob E. Goodman and Joseph O’Rourke, editors, *Handbook of Discrete and Computational Geometry, chapter 39*. CRC Press, 2004. 2nd edition.
- [10] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. of 30th STOC*, pages 604–613, 1998.
- [11] Jon M. Kleinberg. Two algorithms for nearest-neighbor search in high dimensions. pages 599–608, 1997.
- [12] Shyjan Mahamud and Martial Hebert. The optimal distance measure for object detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2003.
- [13] Ketan Mulmuley. Randomized multidimensional search trees: Further results in dynamic sampling (extended abstract). In *Proc. 32nd Symp. on Found. Comput. Sci.*, pages 216–227, 1991.
- [14] Sameer A. Nene and Shree K. Nayar. A simple algorithm for nearest neighbor search in high dimensions. *IEEETPAMI: IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 1997.
- [15] S. M. Omohundro. Five balltree construction algorithms. Technical report, International Computer Science Institute, Berkeley, CA, 1989.
- [16] Hanan Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, Ma, 1989.
- [17] Jeffrey K. Uhlmann. Satisfying general proximity/similarity queries with metric trees. *Information Processing Letters*, 40(4):175–179, 1991.
- [18] Peter Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proc. 4th ACM Symp. on Discrete Algorithms*, pages 311–321, 1993.