



Evaluating a Class of Dimensionality Reduction Algorithms

Jelena Tesic
CS 290I Final project

Vision Research Laboratory
Department of Electrical and Computer Engineering



Outline

- ∩ **Introduction**
- ∩ **SVD-based algorithms**
- ∩ **Distance Mapping Algorithms**
- ∩ **Performance Evaluation**
- ∩ **Results**
- ∩ **Conclusion**
- ∩ **Further research**
- ∩ **Applications**

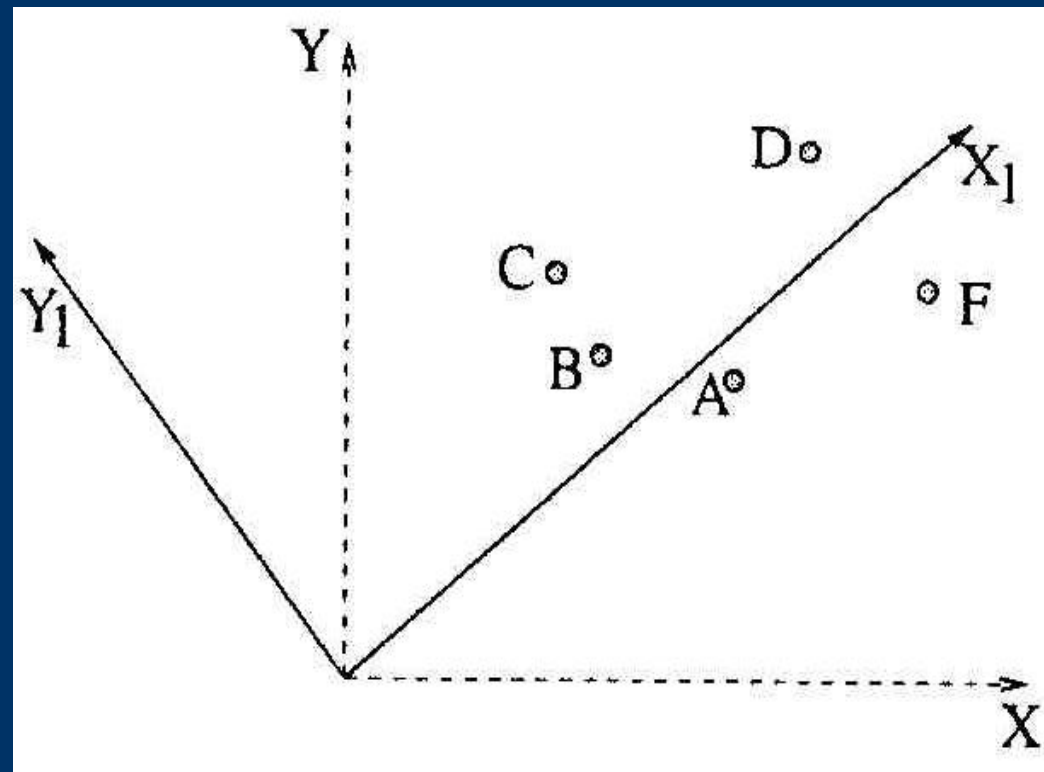
Introduction

- ⌚ Content-based retrieval of multimedia objects
- ⌚ Feature vector representation allows searching in high-dimensional space
- ⌚ Similarity searching in large dynamical databases should be fast and scalable
- ⌚ “Dimensionality curse” requires feature vector dimensionality reduction
- ⌚ Tradeoff between precision and scalability

Singular Value Decomposition

∞ Optimal way to map n -dimensional points to k -dimensional points, $k \leq n$

∞ Computationally expensive
 $O(Nn^2)$



Adaptive Eigenspace Computation

- ⌚ Orthogonal base of reduced space is changed every time when object is inserted/deleted
- ⌚ Expensive to recomputed SVD every time.
- ⌚ Incremental Eigenspace computation
 - compute incremental SVD only when accuracy drops below certain threshold level
 - good approximation
 - computational complexity is $O(Nn)$

Incremental SVD computation

⌚ $U \Sigma V^T$ - approximate low rank SVD of existing dataset A , $A = U \Sigma V^T + E$

⌚ Let $A' = [A : x]$ $A' = [U \Sigma V^T : x] + [E : 0]$

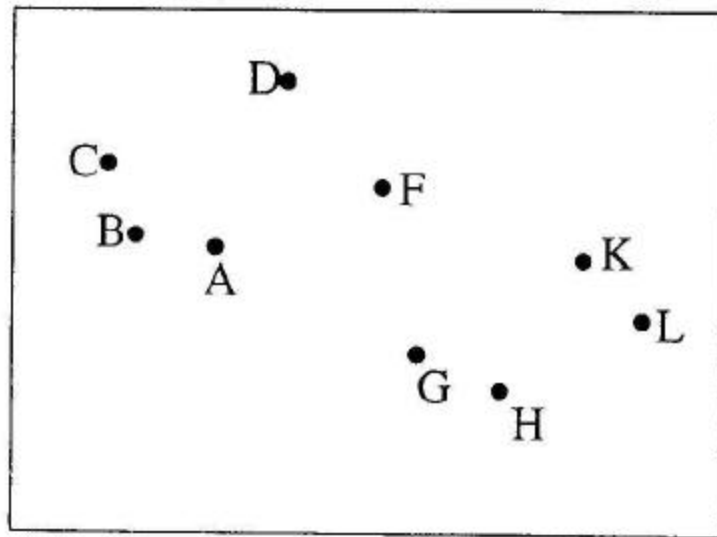
⌚ Compute $[U \Sigma V^T : x] \approx U' \Sigma' V'^T$

⌚ Approximate SVD computation:

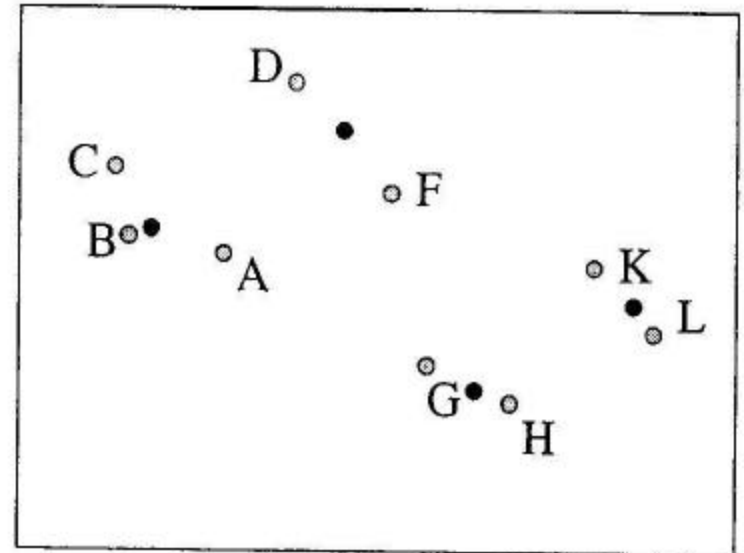
$$a = \frac{x^T - U(U^T x)}{\|x^T - U(U^T x)\|} \quad \begin{pmatrix} \Sigma & U^T x \\ 0 & a^T x \end{pmatrix} = \Omega \Lambda \Phi^T$$

$$U' = (Ua)\Omega \quad V' = \begin{pmatrix} V & 0 \\ 0 & 1 \end{pmatrix} \Phi \quad \Sigma' = \Lambda$$

Approximate SVD



(a) All-Data-SVD



(b) Approximate-SVD

Aggregated Eigenspace Computation

- ⌚ SVD computation is performed on aggregated dataset obtained from an index structure
- ⌚ Recomputation of SVD only when accuracy drops below certain threshold level

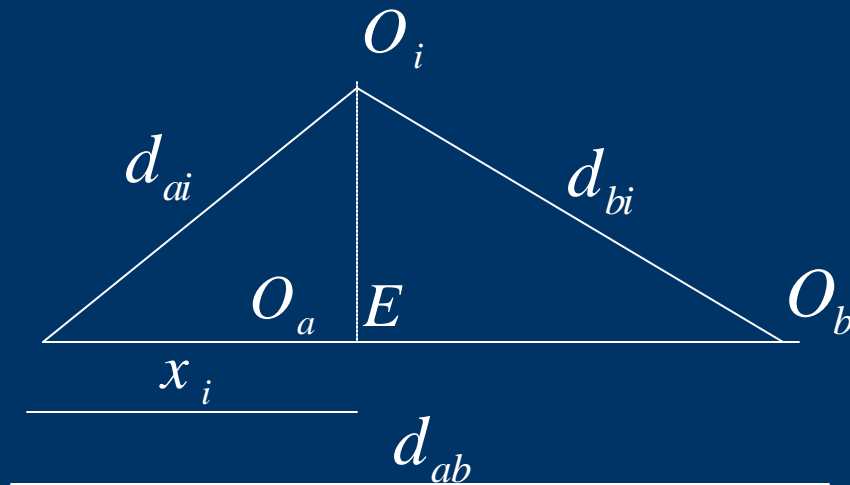


- ⌚ Linear dependence is reduced

Distance Mapping Algorithms

FastMap

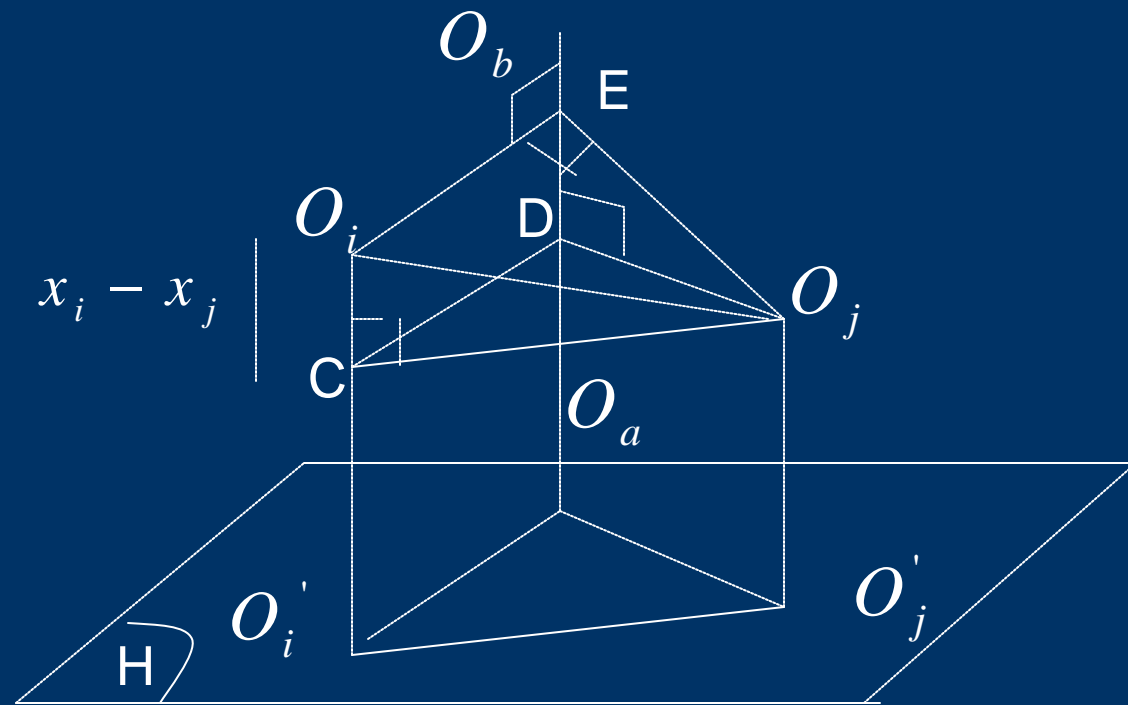
- Map each object O_i into a k -dimensional point $P_i = (x_{i1}, x_{i2}, \dots, x_{ik})$ so that the distances are approximately preserved
- mapping based on cosine law



$$x_i = \frac{d_{ai}^2 + d_{ab}^2 - d_{bi}^2}{2d_{ab}}$$

$$d_{ij} = D(O_i, O_j)$$

Projection on a Hyper plane

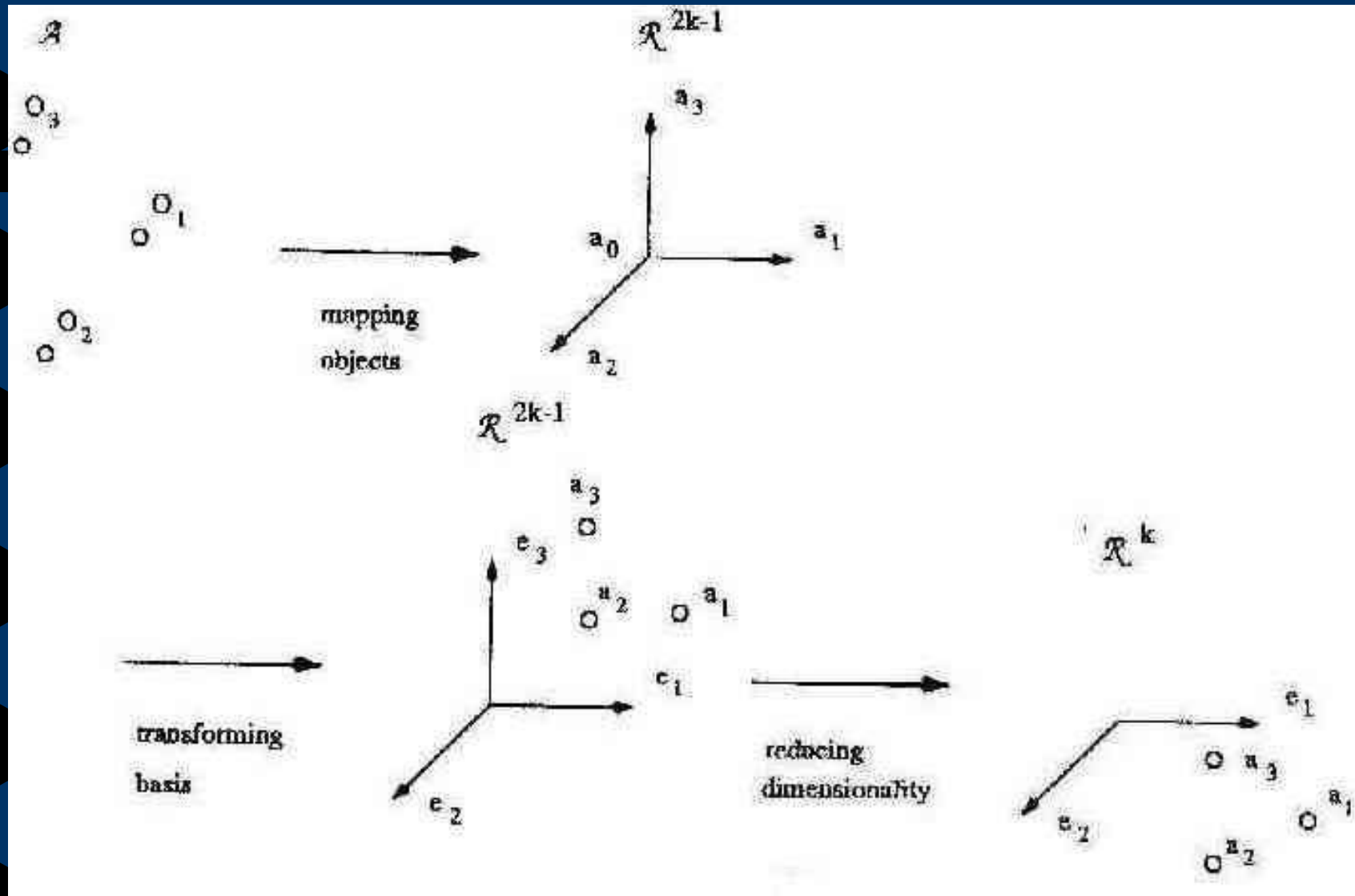


$$(D'(O_i - O_j))^2 = (D(O_i - O_j))^2 - (x_i - x_j)^2$$

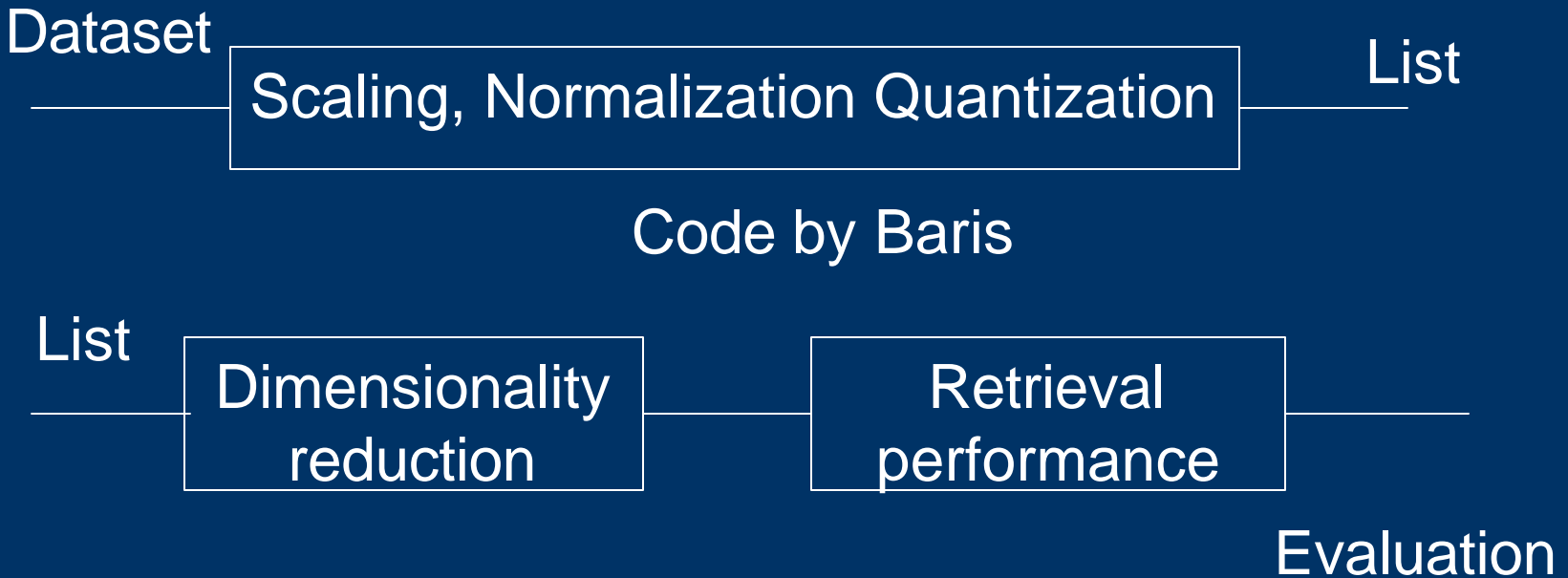
FastMap Algorithm

- ⌚ Choose pivot objects and record their ID's
- ⌚ Project every other object on line (O_a, O_b) and compute x_i
- ⌚ Consider the projection of the objects on a hyper plane perpendicular to the line (O_a, O_b) and repeat the algorithm
- ⌚ Complexity is $O(Nk)$
- ⌚ Pivot objects are recorded so that new or query object can be mapped to k -d space - $O(1)$

MetricMap Algorithm



Algorithm Flow Chart



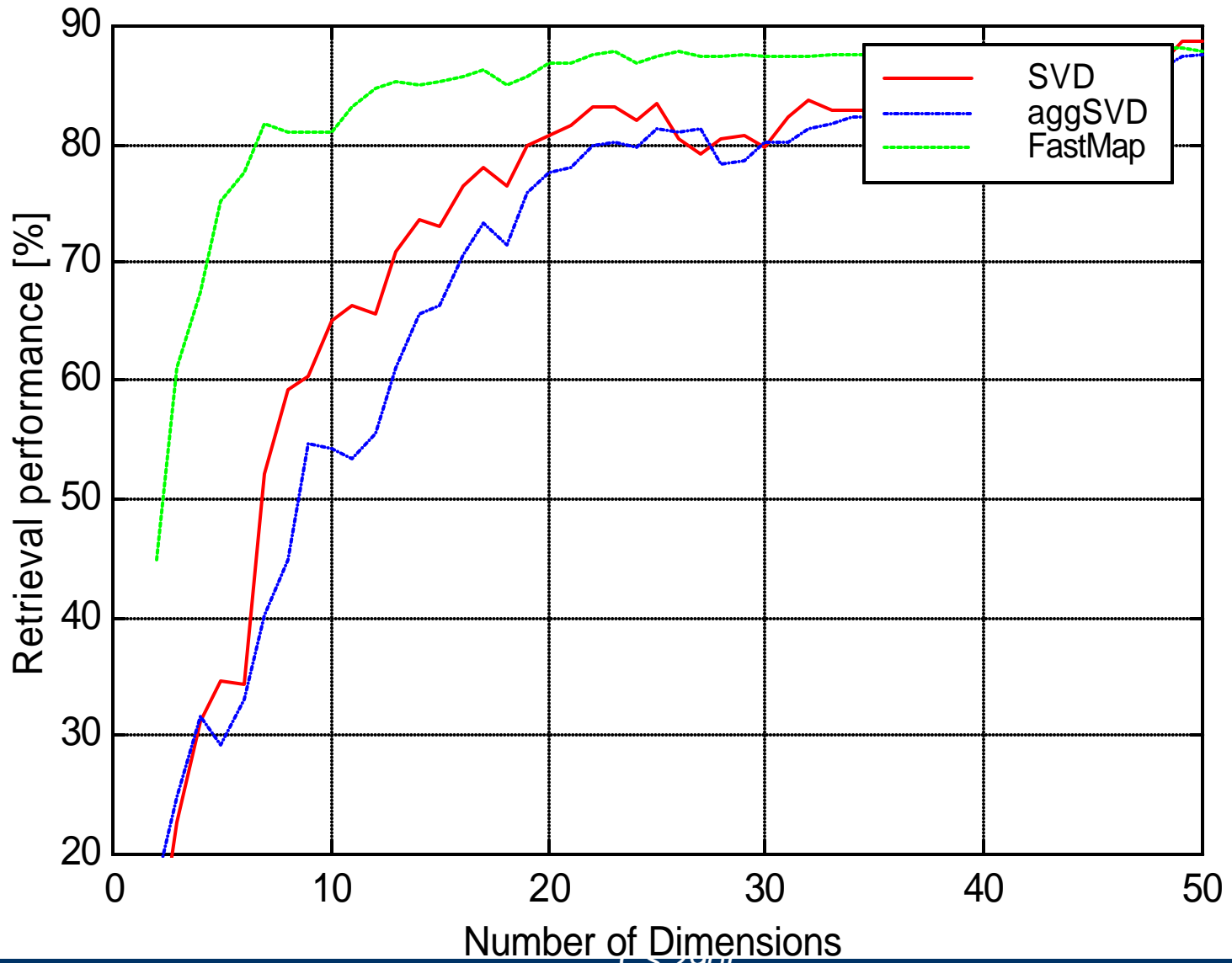
Performance Evaluation

- ⌚ If reduction algorithm have a good retrieval performance we can always incorporate an indexing structure to achieve fast query response.
- ⌚ 50-dimensional texture feature vectors
 - Brodatz album set of 1856 images that belong to 116 different classes and ICU set of 832 images
 - Collection of aerial photographs extracted from 40 large air photos. Non-overlapping 64x64 tiles form the dataset. No classes

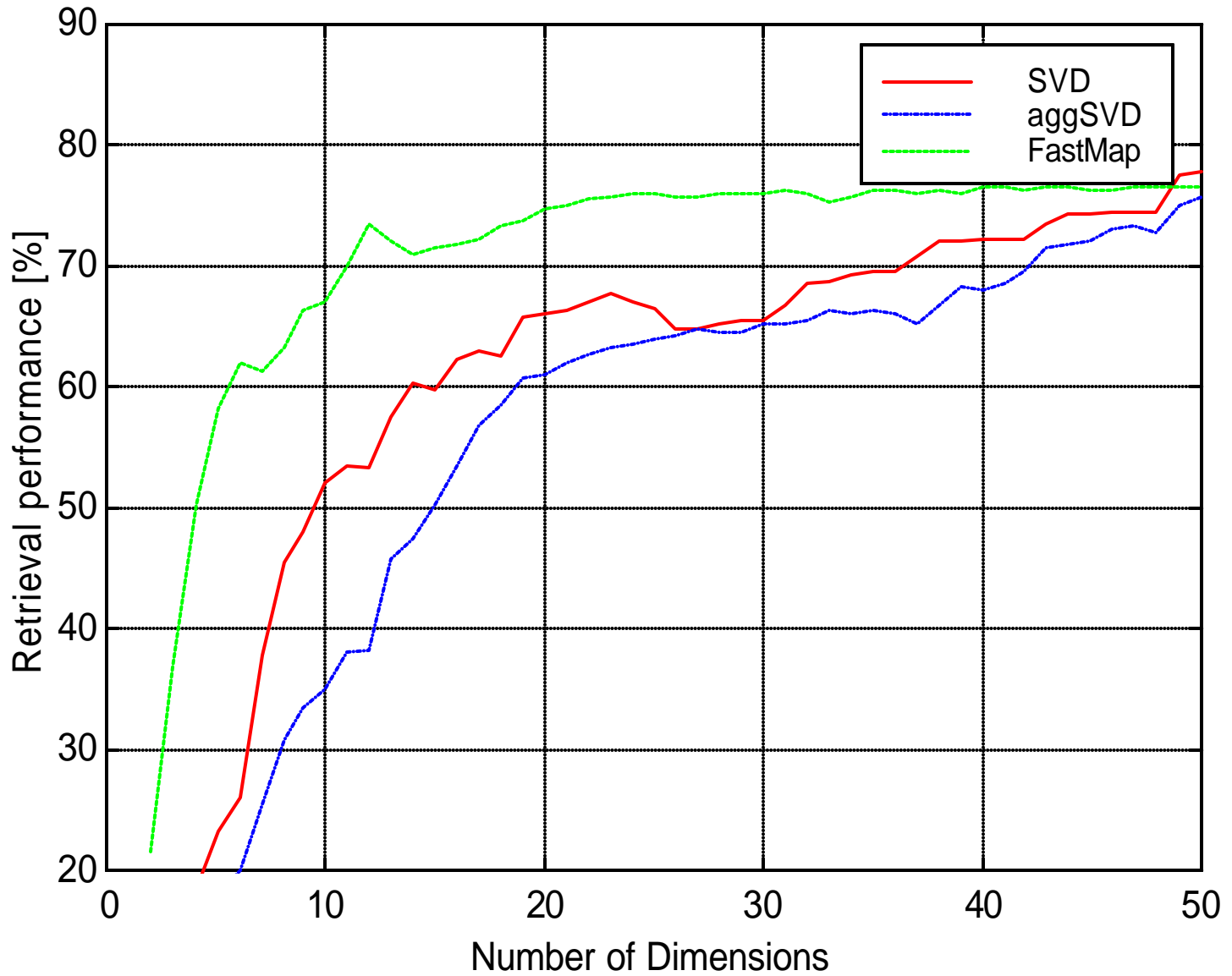
Implemented Methods

- ⌚ SVD - reference plot over dimensions
- ⌚ FastMap - average performance over couple of runs for every dimension
- ⌚ incSVD - repeat for every k
- ⌚ aggSVD - performance evaluated based on the centroids of existing classes
- ⌚ MetricMap - inter object distances

Normalized ICU texture data



Normalized Brodatz texture data



Open issues -first step-

- ⌚ Precision - comparison to FastMap
- ⌚ Scaling factor and Distance measure influence
- ⌚ How to classify aerial photo tiles
 - subjective evaluation or build an index structure
- ⌚ More results
 - <http://maya.ece.ucsb.edu/~jelena/CS290I.html>
 - quantized data

Present/Future work

- ∞ Air photos
- ∞ Use of color feature vectors dataset
- ∞ Index structure
- ∞ Computational complexity of algorithms
- ∞ Improvements
 - distance measure
 - target space for feature vectors
 - index node aggregated data space
 - classification

Conclusion

- ∞ Scalable and precise algorithm - fast indexing
- ∞ Content-based similarity search
- ∞ Finding underlying patterns and relations among images in large databases
- ∞ Visualization tool
- ∞ Classification and clustering tool

References

- ⌚ <http://maya.ece.ucsb.edu/~jelena/links.html>

Acknowledgments

- ⌚ Prof. Ambuj Singh
- ⌚ Prof. Manjunath
- ⌚ Baris Sumengen, Shawn Newsam
- ⌚ authors of referenced papers