Similarity Search for Adaptive Ellipsoid Queries Using Spatial Transformation

Yasushi Sakurai *(NTT Cyber Space Laboratories)*

Masatoshi Yoshikawa *(Nara Institute of Science and Technology)*

Ryoji Kataoka *(NTT Cyber Space Laboratories)*

Shunsuke Uemura *(Nara Institute of Science and Technology)*

Outline

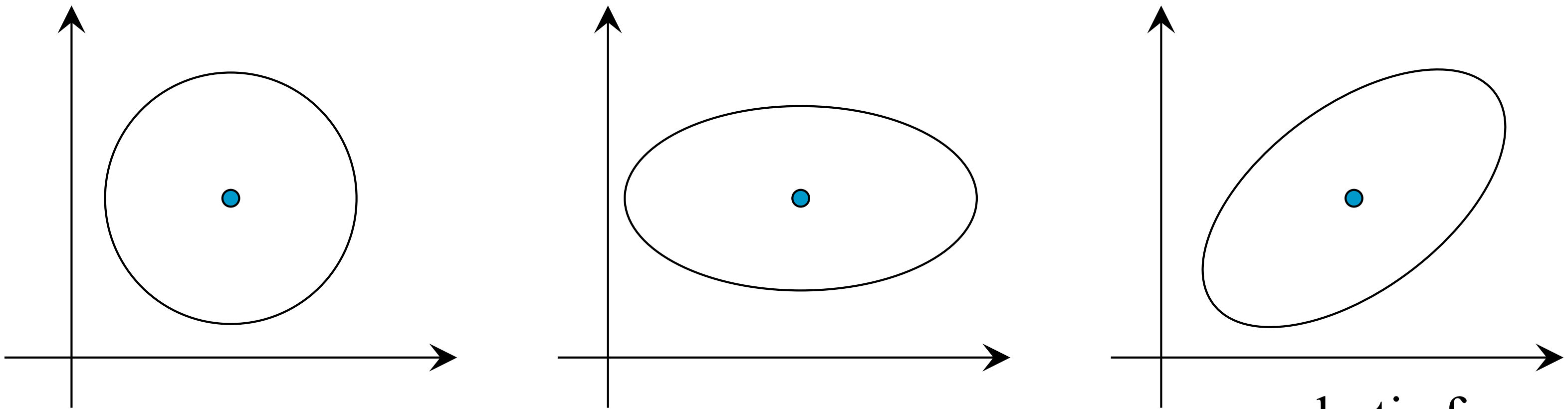
- Introduction
- STT (spatial transformation technique)
  - Definition of spatial transformation
  - Spatial transformation of rectangles
  - Search algorithm
- MSTT (multiple STT)
  - Index structure construction
  - Query processing
  - Dissimilarity of matrices
- Performance test
- Conclusion

Introduction

- Ellipsoid query
  - Search processing is performed by using quadratic form distance functions
  - Distance of p and q for a query matrix M :

$$d_M^2(p,q) = (p-q) \cdot M \cdot (p-q)^t$$

  - represents correlations between dimensions



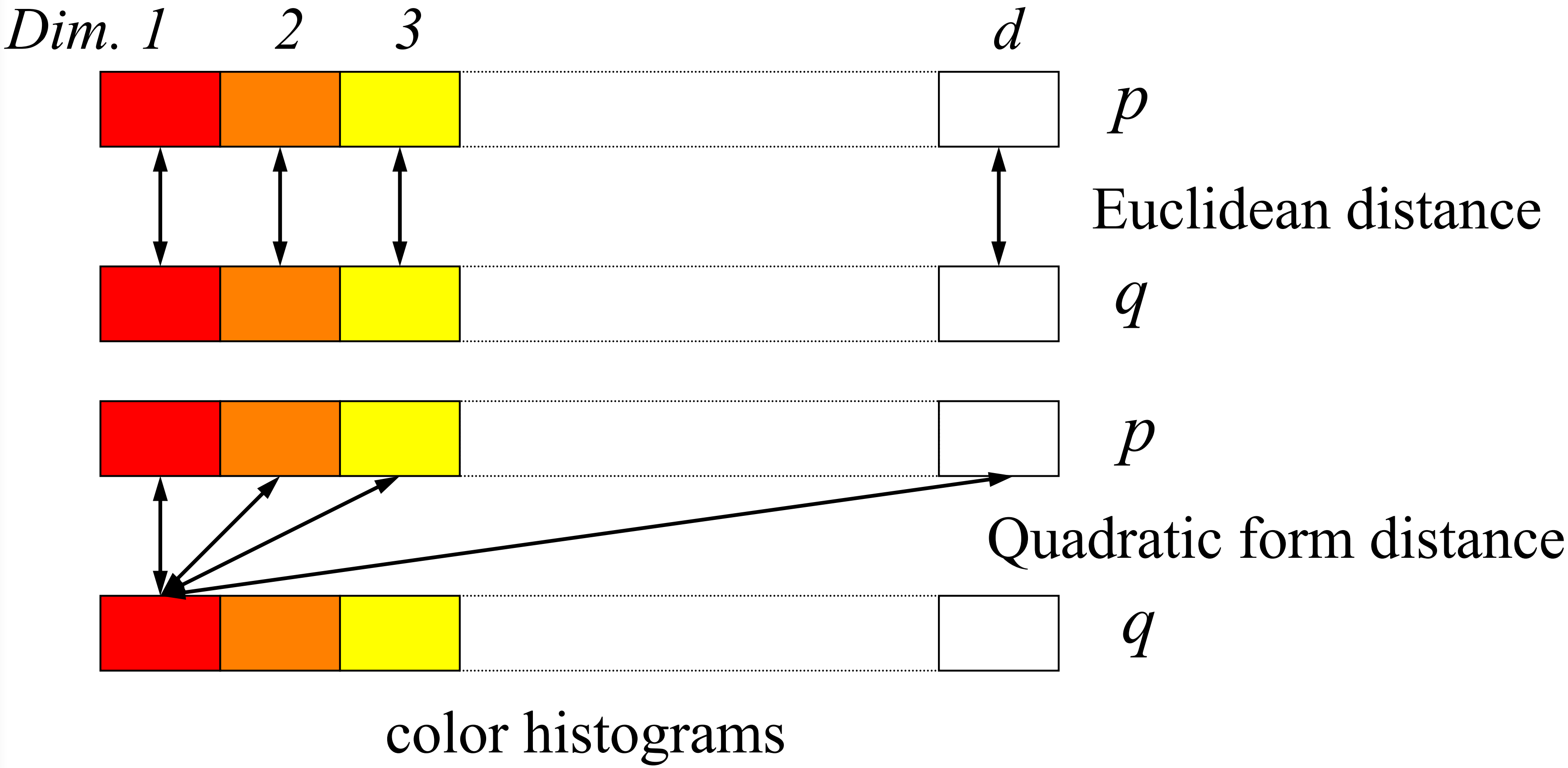
Euclidean
circles for isosurfaces

weighted Euclidean
iso-oriented ellipsoids

quadratic form
Ellipsoids
(Not necessarily aligned
to the coordinate axis)

Introduction

- An application of a quadratic form distance function
  - represent the similarity between colors *i* and *j*

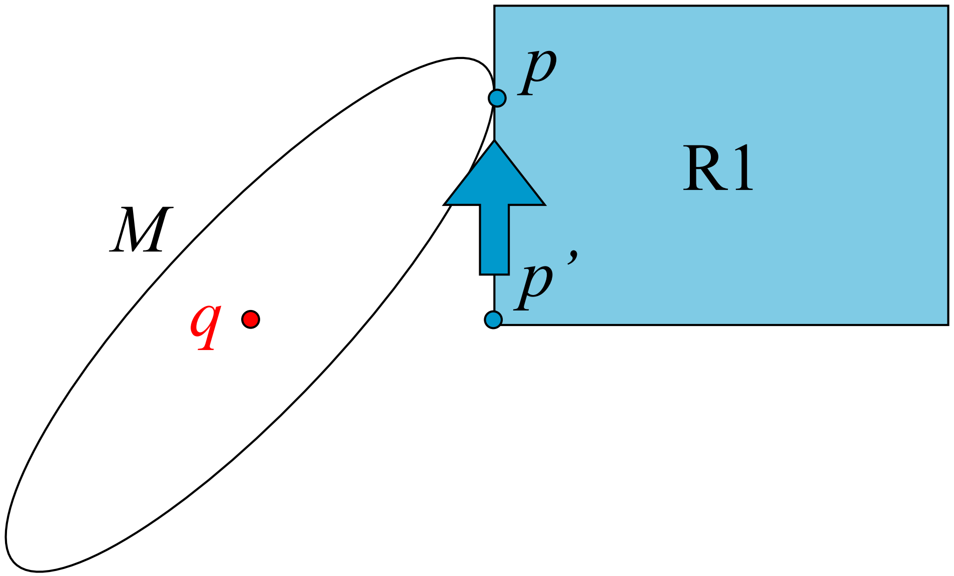


Introduction

- Spatial indices
  - e.g. R-tree family (R*-tree, X-tree, SR-tree, A-tree)
  - Based on the Euclidean distance function
    - ⇨ Cannot be applied to ellipsoid queries
- Efficient search methods for user-adaptive ellipsoid queries
  - Query matrix M is variable

Related Work : Seidl and Kriegel, VLDB97

- Search method based on the steepest descent method
  - Works on spatial indices of R-tree family
  - Calculates the exact distance of a query point and an MBR in an index structure
  - …but requires high CPU cost which exceeds disk access cost



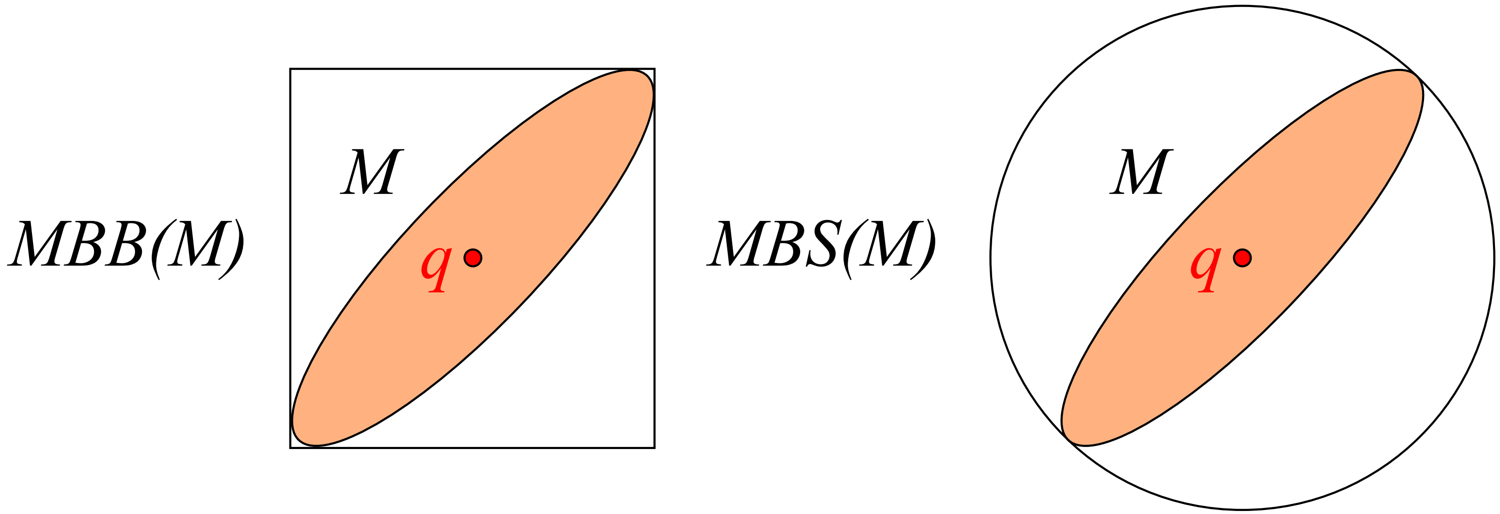
Moves p' toward p iteratively

CPU time $O(\omega d^2)$
ω…number of iterations
d…dimensionality

Related Work : Ankerst et al., VLDB98

- Technique that uses the MBB and MBS distance functions to reduce CPU time
  - MBB and MBS distance functions

$$d^2_{MBB(M)}(p,q) = \max_{i=1}^{d}\left((p_i - q_i)^2 / (M^{-1})_{ii}\right)$$
$$d^2_{MBS(M)}(p,q) = \lambda^2_{M_{\min}} \cdot (p-q)^2$$



Related Work : Ankerst et al., VLDB98

- Approximation technique by using the MBB and MBS distance functions
  - approximation distance : uses either MBB or MBS distance for better approximation quality
  - Calculates the exact distances only if data objects or MBRs cannot be filtered by their approximation distances
  - Saves CPU time by reducing the number of exact distance calculations
  - …but cannot reduce the number of exact distance calculations if its approximation quality is low

Our Contributions

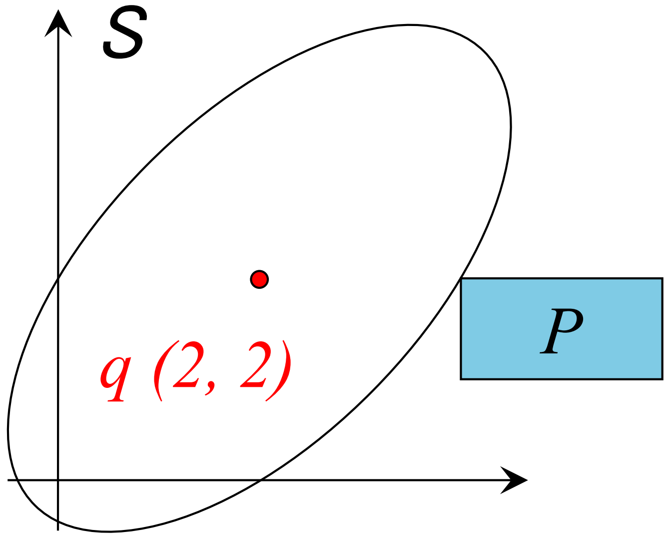
- STT (Spatial Transformation Technique)
  - Ellipsoid queries incur a high CPU cost
  - The efficiency depends on approximation quality
  - STT efficiently processes ellipsoid queries because of high approximation quality
- MSTT (Multiple Spatial Transformation Technique)
  - Does not use only the Euclidean distance function to make index structures
  - Ellipsoid queries give various distance functions
  - In MSTT, various index structures are created; the search algorithm utilizes a structure well suited to a query matrix

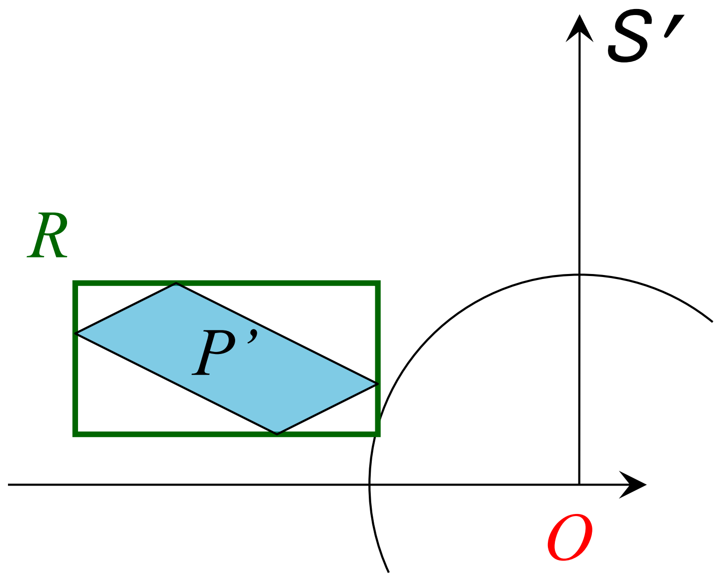
Outline

- Introduction
- STT (spatial transformation technique)
  - Definition of spatial transformation
  - Spatial transformation of rectangles
  - Search algorithm
- MSTT (multiple STT)
  - Index structure construction
  - Query processing
  - Dissimilarity of matrices
- Performance test
- Conclusion

Spatial Transformation Technique (STT)

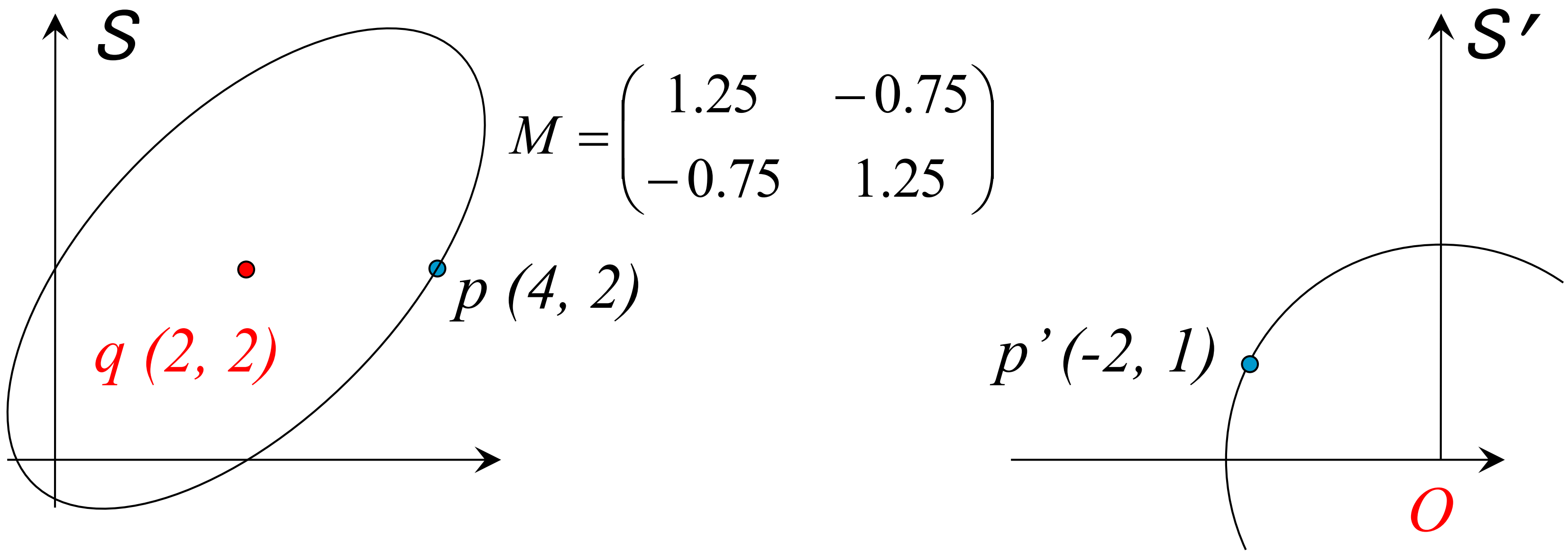
- High approximation quality
  - STT consumes less CPU time
- Spatial transformation
  - MBRs in a quadratic form distance space are transformed into rectangles in the Euclidean distance space





Spatial Transformation

- Definition of spatial transformation
  - p : a point in the quadratic form distance space S
  - p': a point in the Euclidean distance space S'
  - The distance between q and p in S is equal to the distance between p' and O in S'
  - Spatial transformation of p into p'



Spatial Transformation

- Definition of spatial transformation
  - $d_M^{\,2}(p, q)$: the distance of p and q in S

    $$d_M^2(p,q) = (p-q)\cdot M \cdot (p-q)^t$$

  - E_M: the eigenvector of M, Λ_M: the eigenvalues of M

    $$M = E_M \cdot \Lambda_M \cdot E_M^t$$

    $$d_M^2(p,q) = (p-q)\cdot E_M \cdot \Lambda_M \cdot E_M^t \cdot (p-q)^t$$

  - Spatial transformation of p into p'

    $$d_M^2(p,q) = p' \cdot p'^t = d^2(p',O)$$

    $$p' = (p-q)\cdot A_M$$

    $$A_M = E_M \cdot \Lambda_M^{1/2}$$

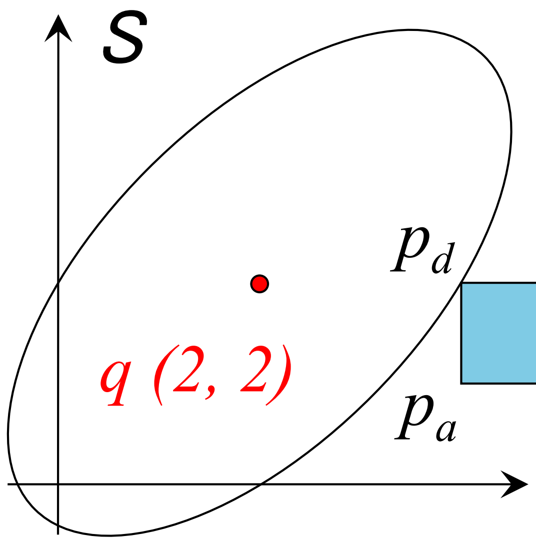
Approximation Rectangles

1. P in S is transformed into P' in S'

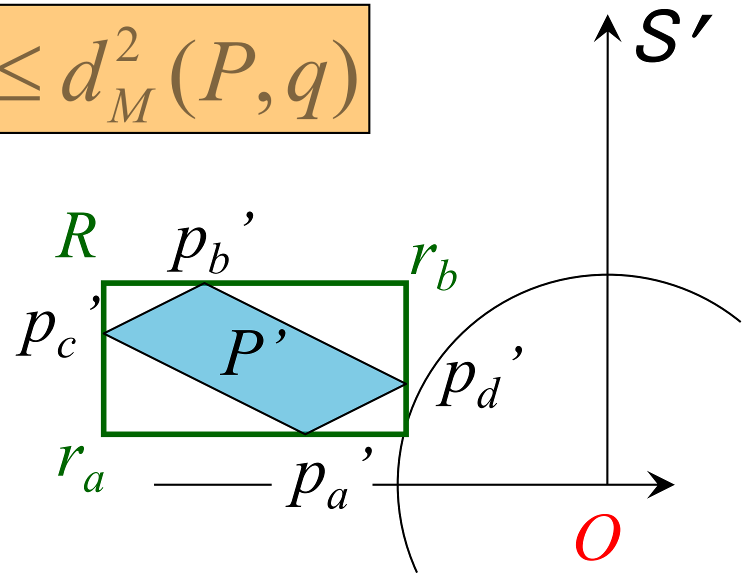
   The calculation of distance between the origin and polygons in high-dimensional spaces incurs a high CPU cost

2. P' is approximated by R
3. $d^2(R, O)$ is used instead of $d^2_M(P, q)$

} low CPU cost



$$d^2(R,O) \le d^2_M(P,q)$$



Approximation Rectangles

1. Calculates $p'_a = (p_a - q) \cdot A_M$

   p_a : lower endpoint of the major diagonal of P

2. Creates the two matrices from the components a_{ij} of A_M

$$\phi_{ij} = \begin{cases} a_{ij} & (a_{ij} < 0) \\ 0 & (otherwise), \end{cases} \quad \psi_{ij} = \begin{cases} a_{ij} & (a_{ij} > 0) \\ 0 & (otherwise) \end{cases}$$

3. Calculates the approximation rectangle R of P'

$$R = (r_a, r_b),$$

$$r_{a_j} = p'_{a_j} + \sum_{i=1}^{d} l_i \cdot \phi_{ij}, \quad r_{b_j} = p'_{a_j} + \sum_{i=1}^{d} l_i \cdot \psi_{ij}$$

   l_i : the edge length of P for the i-th dimension

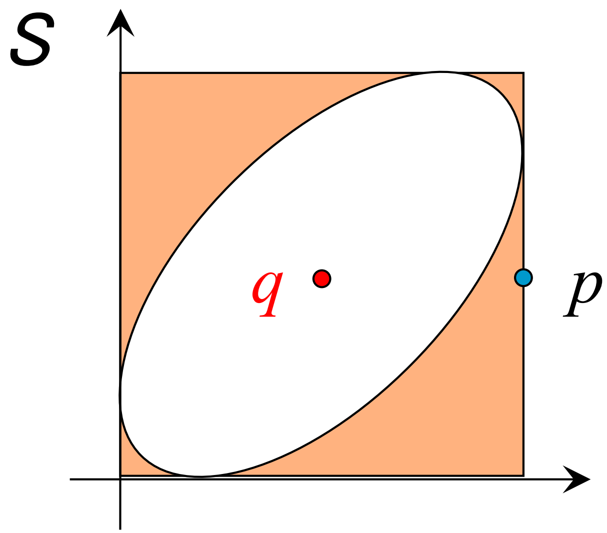
4. R can be used for search since R totally contains P', that is $d^2(R,O) \leq d_M^2(P,q)$

Search Algorithm

1. Calculates the transformation matrix of M
2. Searches for similarity objects by using an index

[Data nodes]

– Calculates $d_{MBB\text{-}MBS(M)}(p, q)$

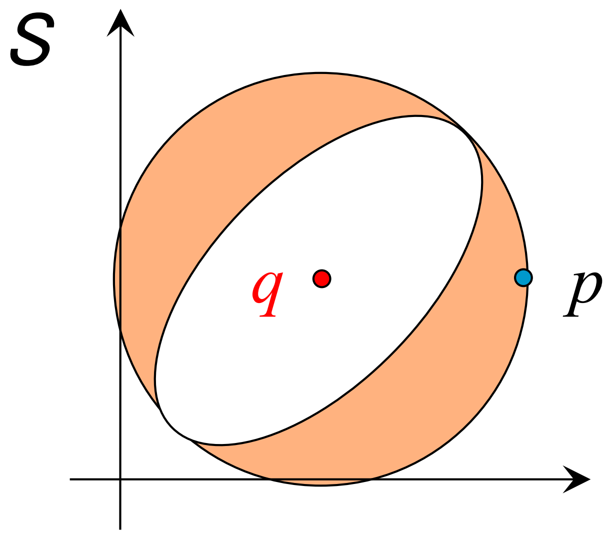


Search Algorithm

1. Calculates the transformation matrix of M
2. Searches for similarity objects by using an index

   [Data nodes]

   – Calculates $d_{MBB\text{-}MBS(M)}(p, q)$



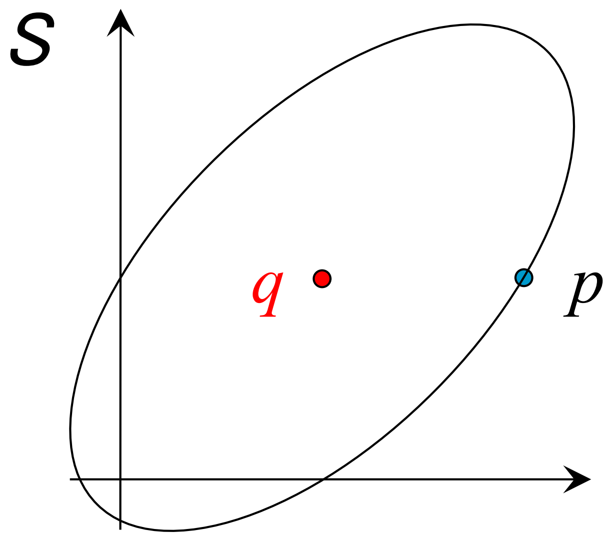
Search Algorithm

1. Calculates the transformation matrix of M
2. Searches for similarity objects by using an index

[Data nodes]

– Calculates $d_{MBB\text{-}MBS(M)}(p, q)$

– Calculates $d_M(P, q)$ if $d_{MBB\text{-}MBS(M)}(p, q) \leq d_{(M)}(k\text{-}NN, q)$

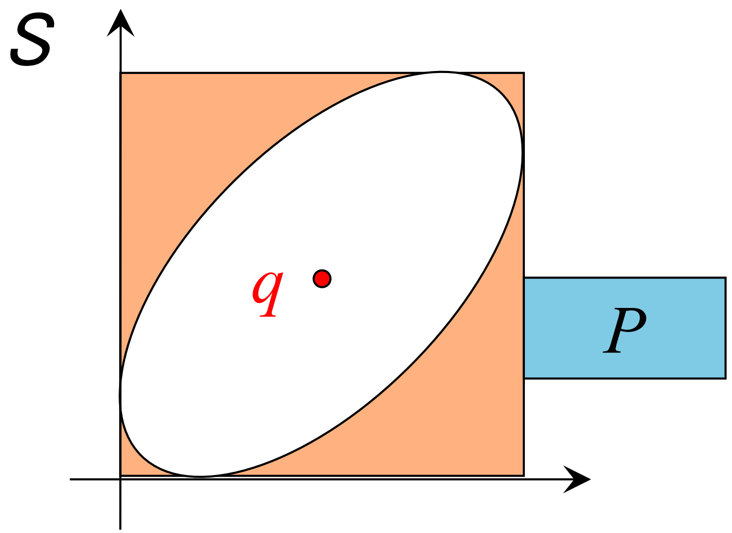


Search Algorithm

1. Calculates the transformation matrix of M
2. Searches for similarity objects by using an index

[Directory nodes]

– Calculates $d_{MBB\text{-}MBS(M)}(P, q)$

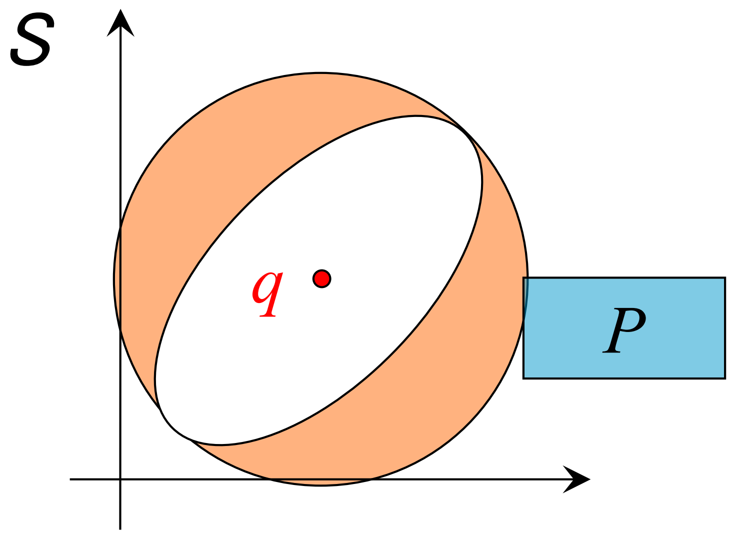


Search Algorithm

1. Calculates the transformation matrix of M
2. Searches for similarity objects by using an index

[Directory nodes]

– Calculates $d_{MBB\text{-}MBS(M)}(P, q)$



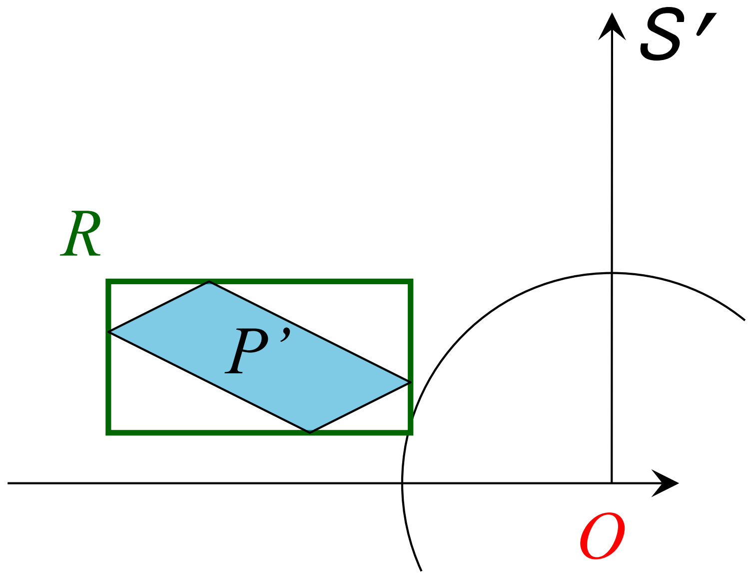
Search Algorithm

1. Calculates the transformation matrix of M
2. Searches for similarity objects by using an index

   [Directory nodes]

   – Calculates $d_{MBB\text{-}MBS(M)}(P, q)$

   – Calculates $d(R, O)$ if $d_{MBB\text{-}MBS(M)}(P, q) \leq d_{(M)}(k\text{-}NN, q)$



Search Algorithm

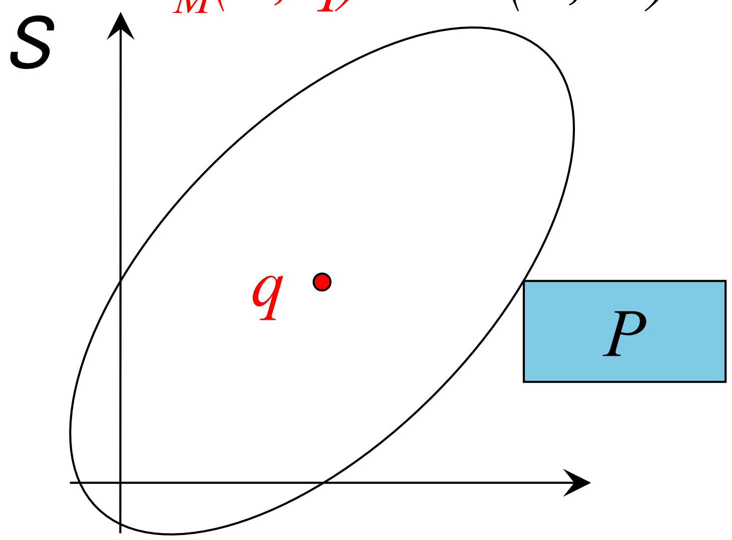
1. Calculates the transformation matrix of M
2. Searches for similarity objects by using an index

[Directory nodes]

– Calculates $d_{MBB\text{-}MBS(M)}(P, q)$

– Calculates $d(R, O)$ if $d_{MBB\text{-}MBS(M)}(P, q) \leq d_{(M)}(k\text{-}NN, q)$

– Calculates $d_M(P, q)$ if $d(R, O) \leq d_{(M)}(k\text{-}NN, q)$



Outline

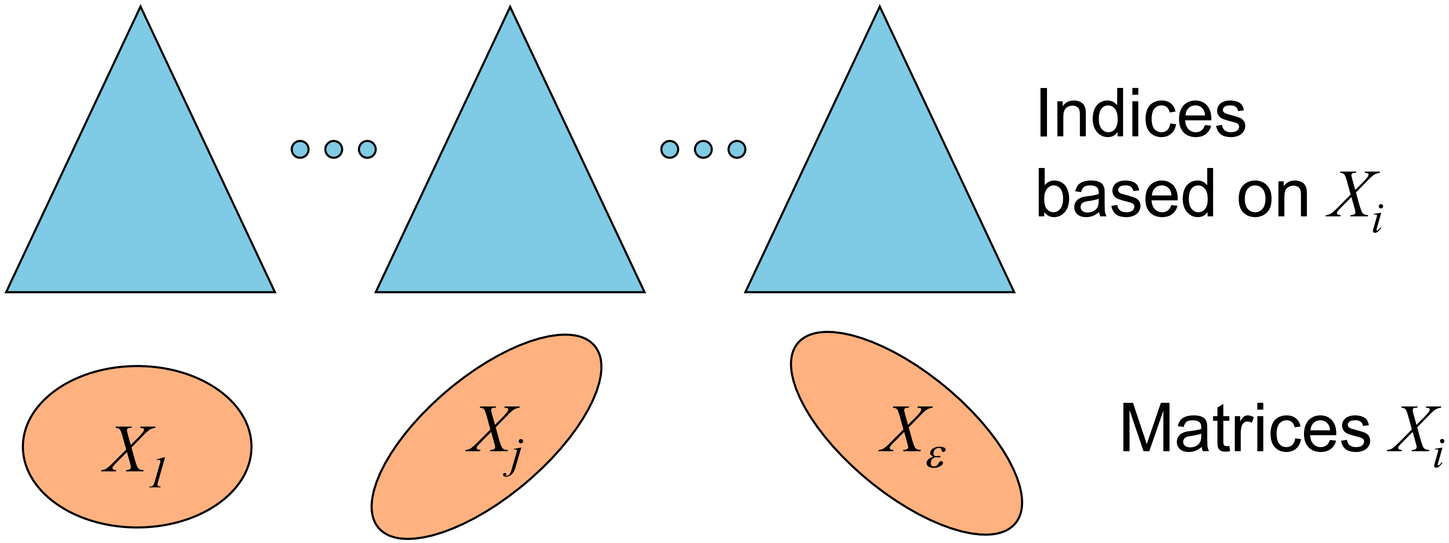
- Introduction
- STT (spatial transformation technique)
  - Definition of spatial transformation
  - Spatial transformation of rectangles
  - Search algorithm
- MSTT (multiple STT)
  - Index structure construction
  - Query processing
  - Dissimilarity of matrices
- Performance test
- Conclusion

Multiple Spatial Transformation Technique (MSTT)

- Node access problem
  - If a query matrix is NOT similar to the unit matrix, it causes a large number of node accesses
  - Index structures are constructed by the Euclidean distance function
- Constructs various index structures by using quadratic form distance functions
  - Chooses a structure that gives sufficient search performance in query processing
  - Reduces both CPU time and number of page accesses for ellipsoid queries

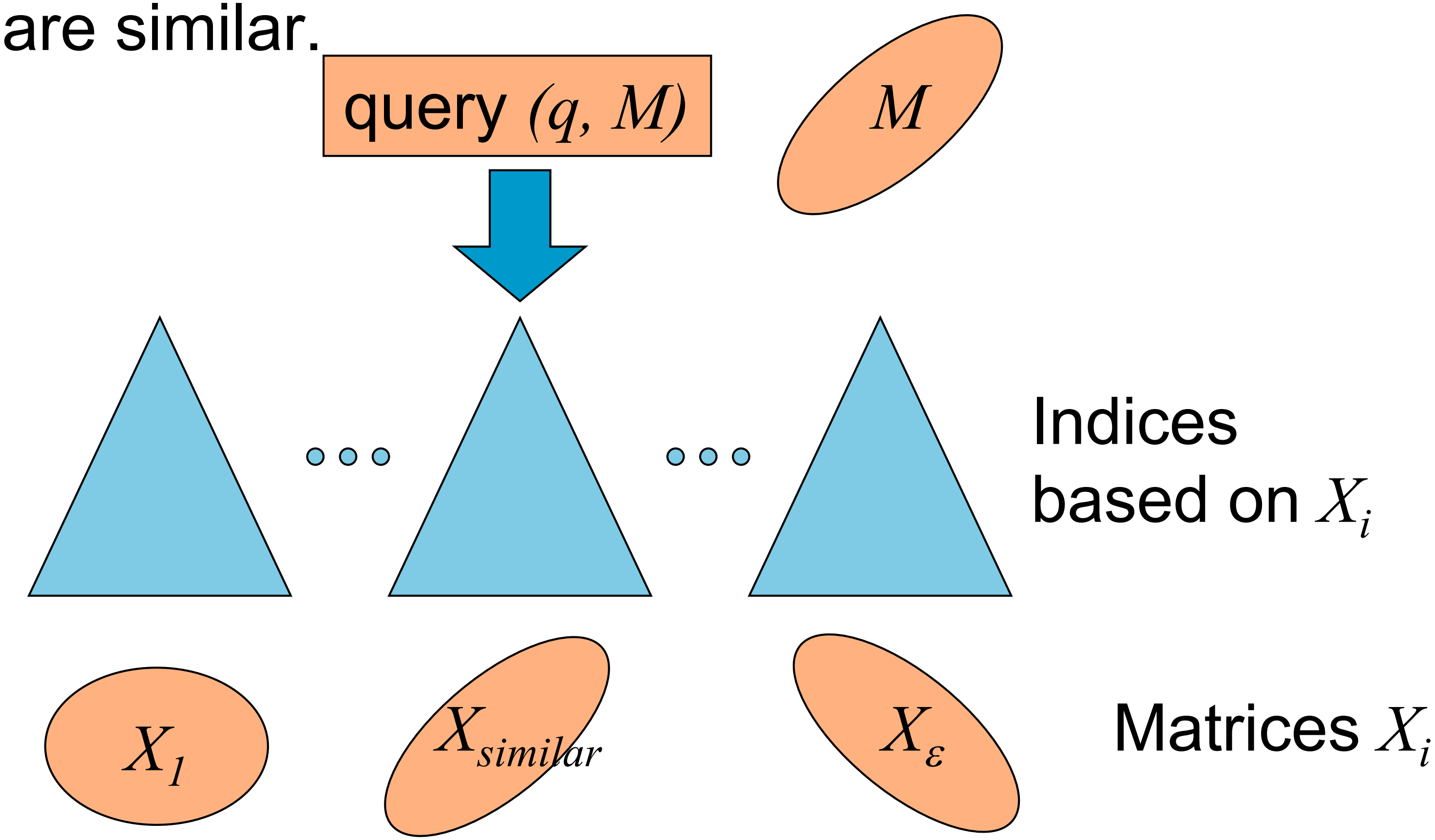
Basic Idea

- Similarity of matrices
  - High search performance can be expected when the query matrix and the matrix of selected index are similar.



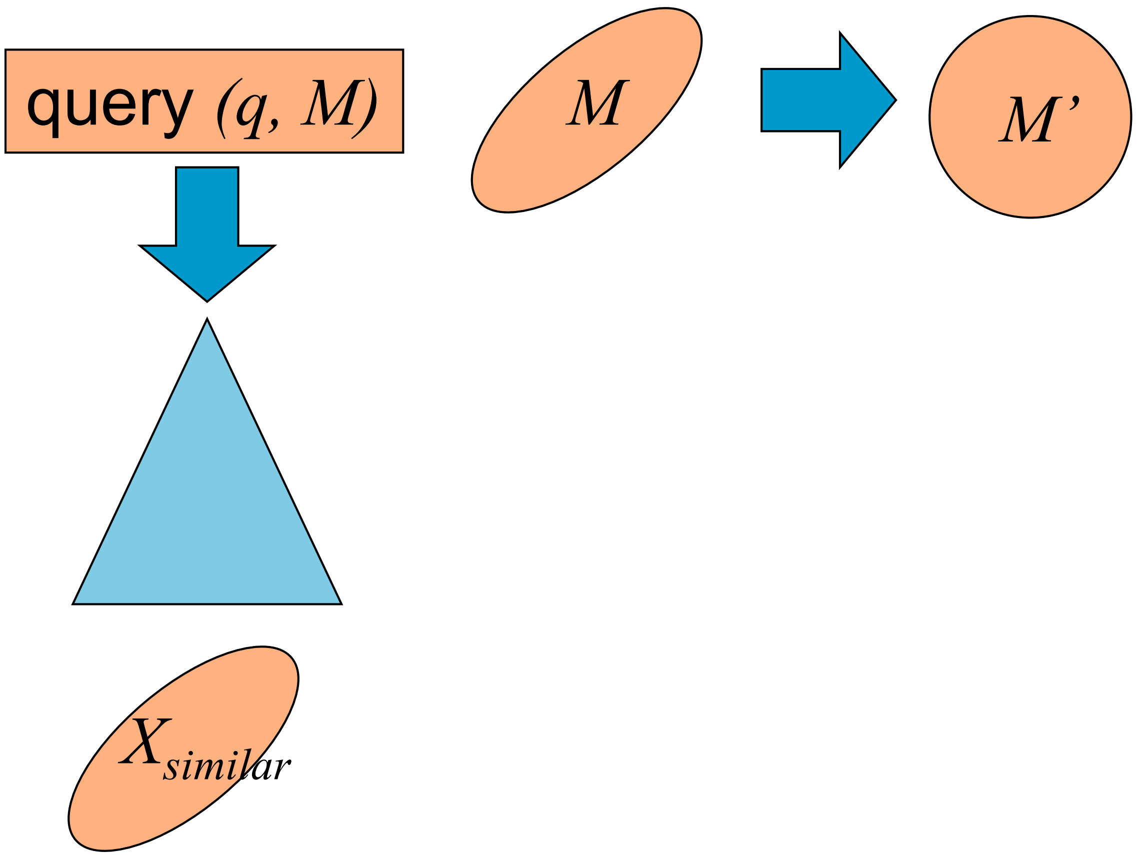
Basic Idea

- Similarity of matrices
  - High search performance can be expected when the query matrix and the matrix of selected index are similar.



Basic Idea

- Similarity of matrices
  - High search performance can be expected when the query matrix and the matrix of selected index are similar.



Indexing and Retrieval Mechanism

- Index structure construction
  - C : the matrix for constructing the index I_C
  - Transformation matrix $A_C = E_C \cdot \Lambda_C^{1/2}$
  - All data points in a data set are transformed

    $p' = p \cdot A_C$
  - I_C is constructed using transformed data points

Indexing and Retrieval Mechanism

- Query processing
  1. Calculates the transformed query point $q' = q \cdot A_C$
  2. Calculates the query matrix
     $M' = A_C^{-1} \cdot M \cdot (A_C^{-1})^t$
  3. Performs search processing by using I_C, M', q'
- The query of M can be processed by using I_C

$$
\begin{aligned}
d_M^2(p,q) &= (p-q)\cdot M \cdot (p-q)^t \\
&= (p'-q')\cdot A_C^{-1} \cdot M \cdot (A_C^{-1})^t \cdot (p'-q')^t \\
&= (p'-q')\cdot M' \cdot (p'-q')^t
\end{aligned}
$$

Similarity of Matrices

- Flatness of a query matrix
  - The variance σ^2_M of the eigenvalues of M is called the flatness of M:

$$\sigma_M^2 = \sum_{i=1}^{d} (\lambda_{M_i} - \bar{\lambda}_M)^2, \quad \bar{\lambda}_M = \sum_{j=0}^{d} \lambda_{M_j} / d$$

λ_{M_i} : the i-th dimensional eigenvalue

$\bar{\lambda}_M$: the average of the eigenvalues of M

  - The flatness of the unit matrix is 0 (search of the Euclidean space).

Similarity of Matrices

- Dissimilarity of M and I_C
  - MSTT employs $\sigma^2_{M'}$ as the measure of the dissimilarity between M and I_C
  - $\sigma^2_{M'}$: the flatness of M'

    $$M' = A_C^{-1} \cdot M \cdot (A_C^{-1})^t$$
  - The effectiveness of I_c relative to M improves as $\sigma^2_{M'}$ decreases

Outline

- Introduction
- STT (spatial transformation technique)
  - Definition of spatial transformation
  - Spatial transformation of rectangles
  - Search algorithm
- MSTT (multiple STT)
  - Index structure construction
  - Query processing
  - Dissimilarity of matrices
- Performance test
- Conclusion

Performance Test

- Data sets: real data set (rgb histogram of images)
- Data size: 100,000
- Dimensionality: 8 and 27
- Page size: 8 KB
- 20-nearest neighbor queries
- Evaluation is based on the average for 100 query points
- Index structure : A-tree (Sakurai et al., VLDB2000)
- CPU: SUN UltraSPARC-II 450MHz

Performance Test

- Query matrices for experiments
  - [HSE+95] : the components of M

$$m_{ij} = \exp(-\alpha(d_w(c_i, c_j)/d_{\max})^2)$$

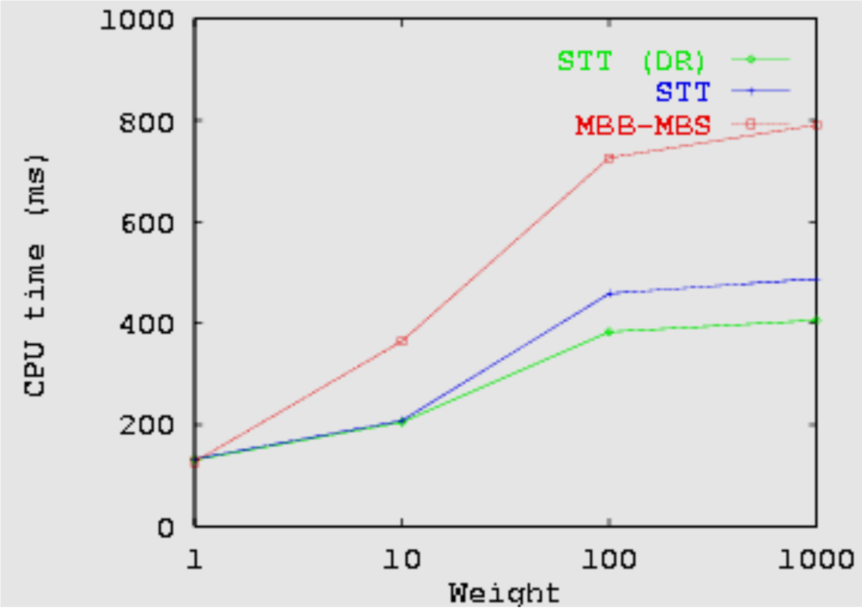
α : positive constant,

$d_w(c_i, c_j)$: the weighted Euclidean distance between the color c_i and c_j,

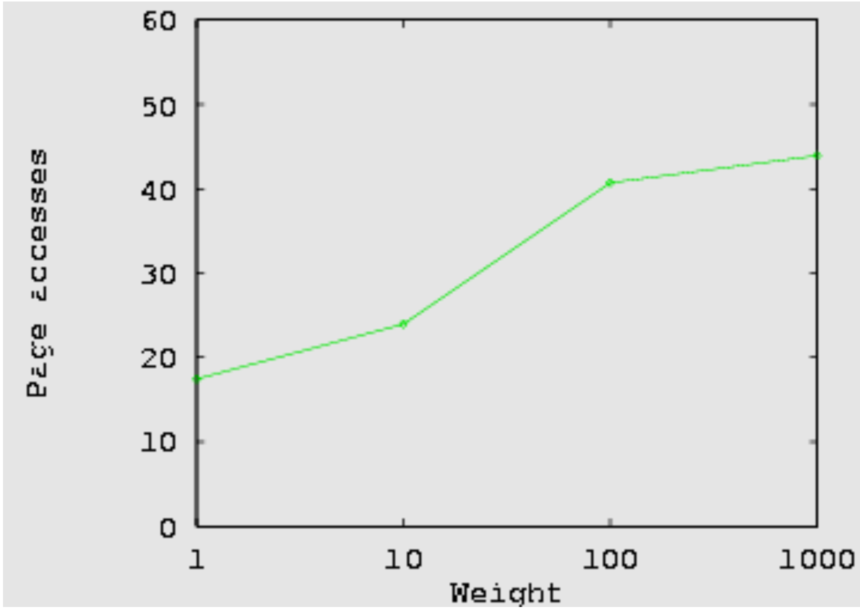
$w=(w_r, w_g, w_b)$: the weightings of the red, green and blue components in RGB color space

  - $\alpha=10, w_g=w_b=1$
  - w_r was varied from 1 to 1,000
  - The flatness of M increases as w_r becomes large

Performance of STT



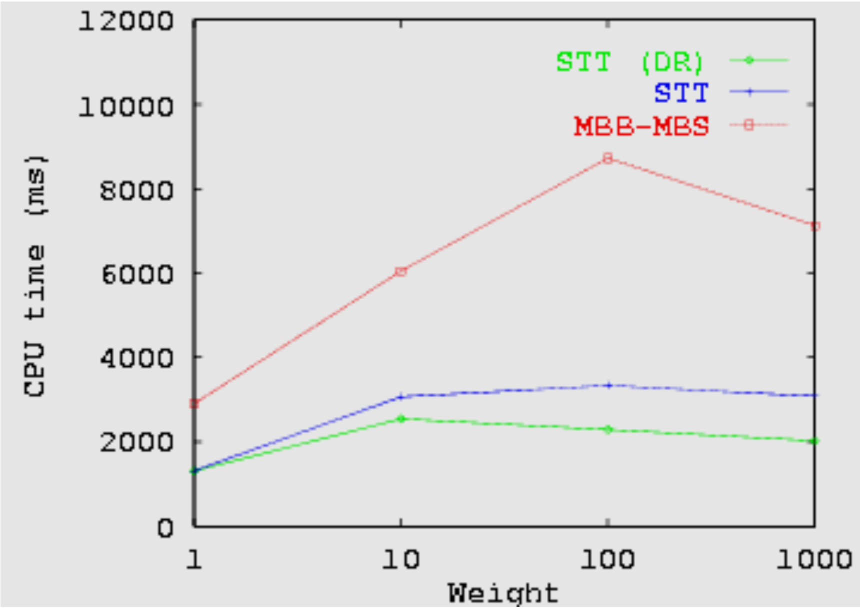
CPU time (d = 8)



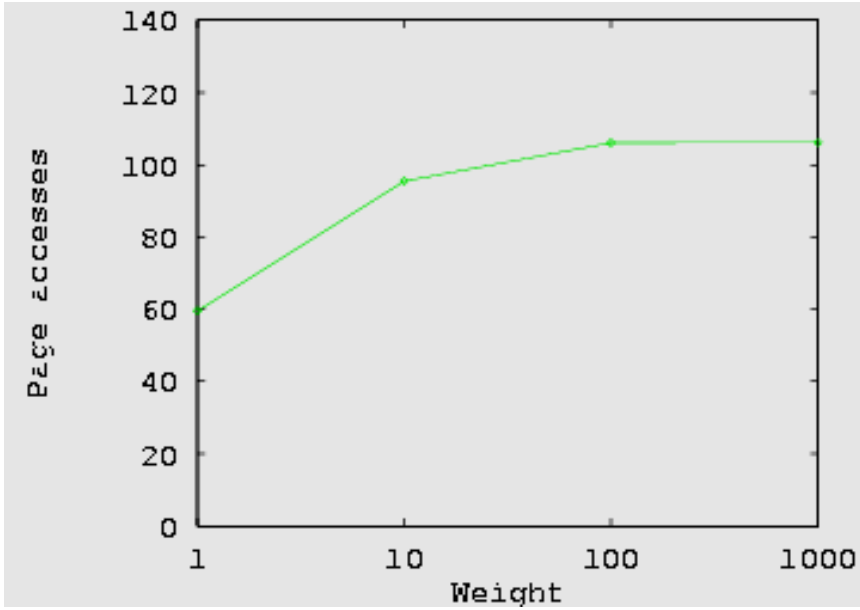
Number of page accesses (d = 8)

- Comparison of STT and MBB-MBS (8D)
  - Both methods require the same number of page accesses since they utilize exact distance functions
  - Low CPU cost : STT increases approximation quality, and reduces the number of exact calculations
  - The effectiveness of STT increases with matrix flatness

Performance of STT



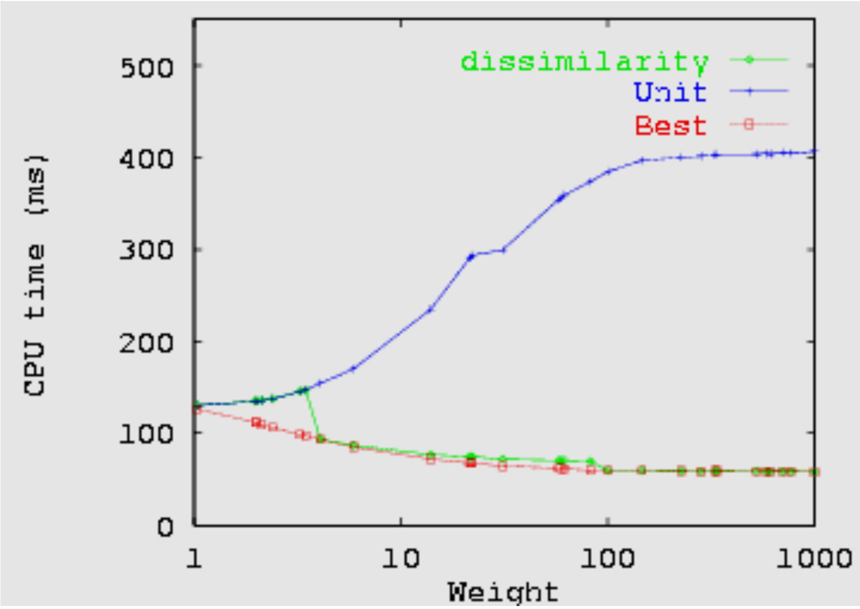
CPU time (d = 27)



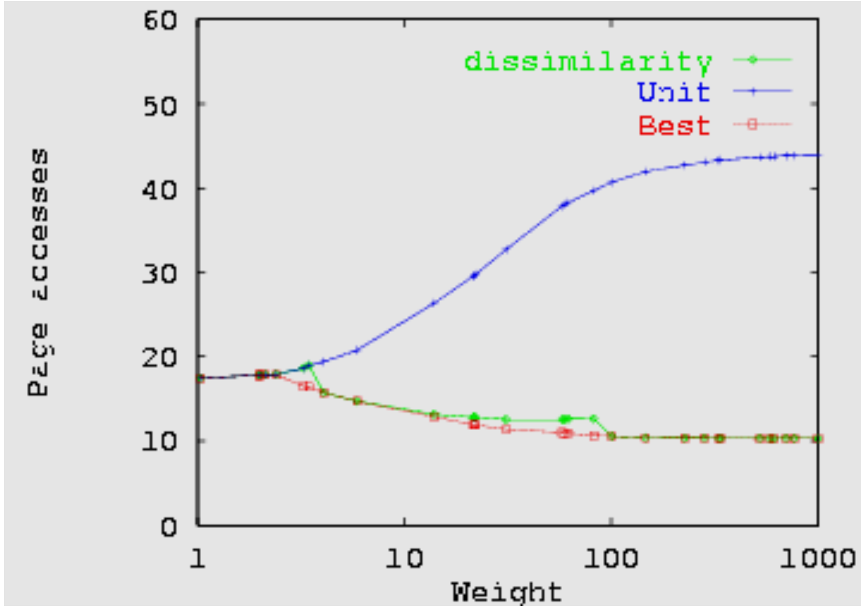
Number of page accesses (d = 27)

- Comparison of STT and MBB-MBS (27D)
  - The effectiveness of STT increases as either dimensionality or matrix flatness grows
  - STT achieves a 74% reduction in CPU cost for high dimensionality and matrix flatness

Performance of MSTT



CPU time (d = 8)



Number of page accesses (d = 8)

- Three structures
  - structure constructed by the unit matrix (Unit)
  - structure constructed by the matrix w_r=10
  - structure constructed by the matrix w_r=1000
- Performance of MSTT
  - Dissimilarity : the cost of search using a structure chosen by the dissimilarity function
  - Dissimilarity is not optimal, but provides good performance

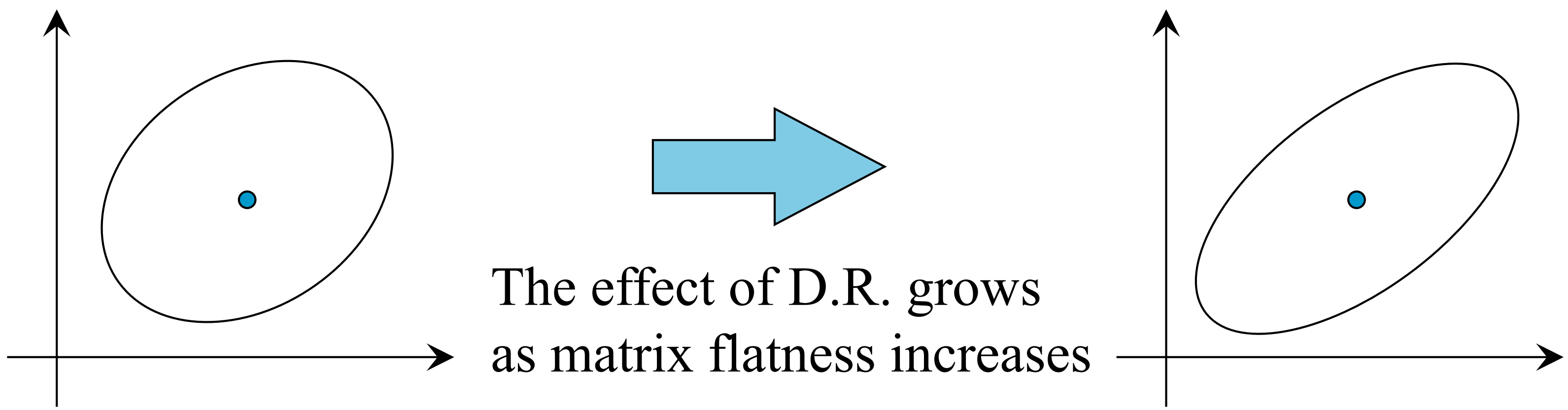
Conclusions

- Search methods for user-adaptive ellipsoid queries
- STT (Spatial Transformation Technique)
  - Spatial transformation ： MBRs in the quadratic form distance space are transformed into rectangles in the Euclidean distance space
  - STT performs ellipsoid queries efficiently even when dimensionality or matrix flatness is high
- MSTT (Multiple Spatial Transformation Technique)
  - MSTT creates various index structures; the search algorithm utilizes a structure well suited to a query matrix
  - MSTT reduces both CPU time and the number of page accesses

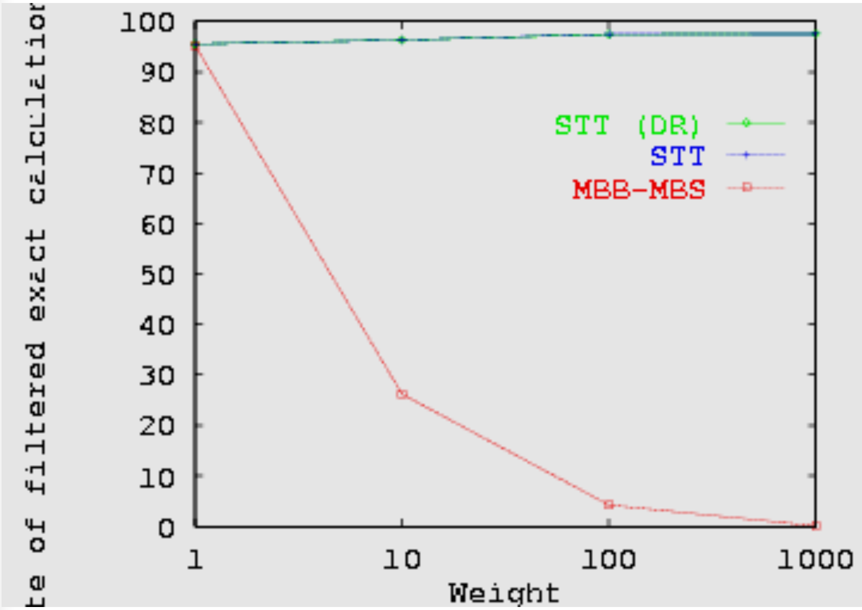
Dimensionality Reduction

- Eigenvalues of a query matrix
  - Dimensions corresponding to small eigenvalues contribute less to approximation quality
  - These dimensions are eliminated to save on CPU costs
  - Calculation time for the spatial transformation of rectangles is reduced to $n/d \ (n \geq d)$

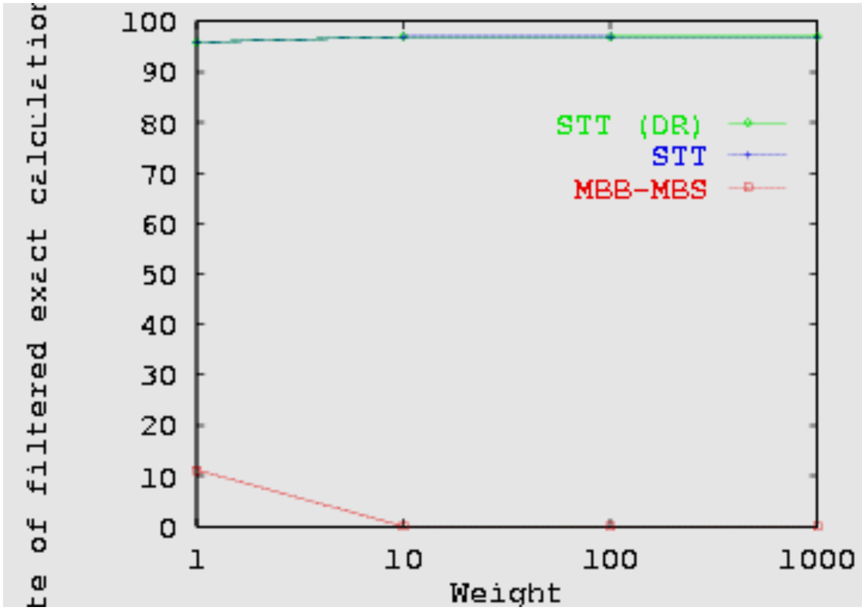
  n : the number of dimensions used



Performance of STT (2)



d = 8

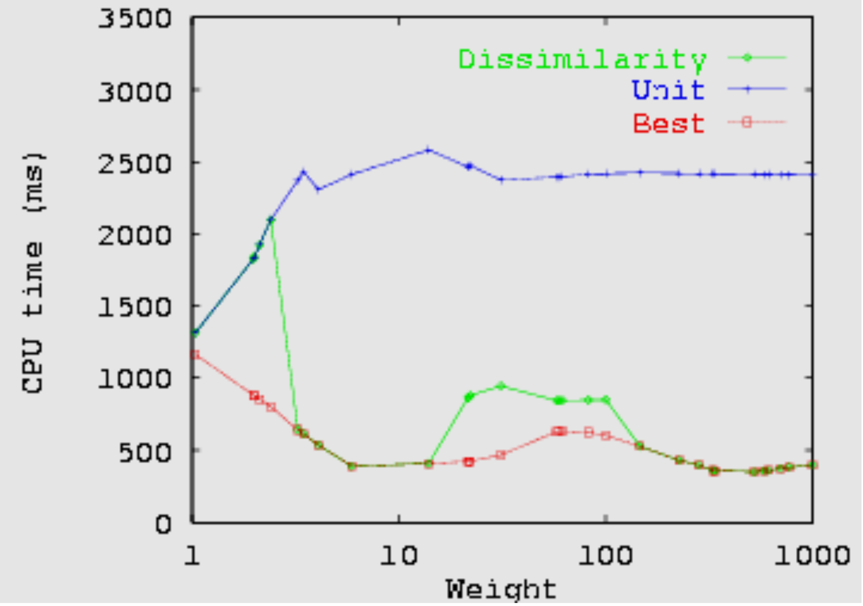


d = 27

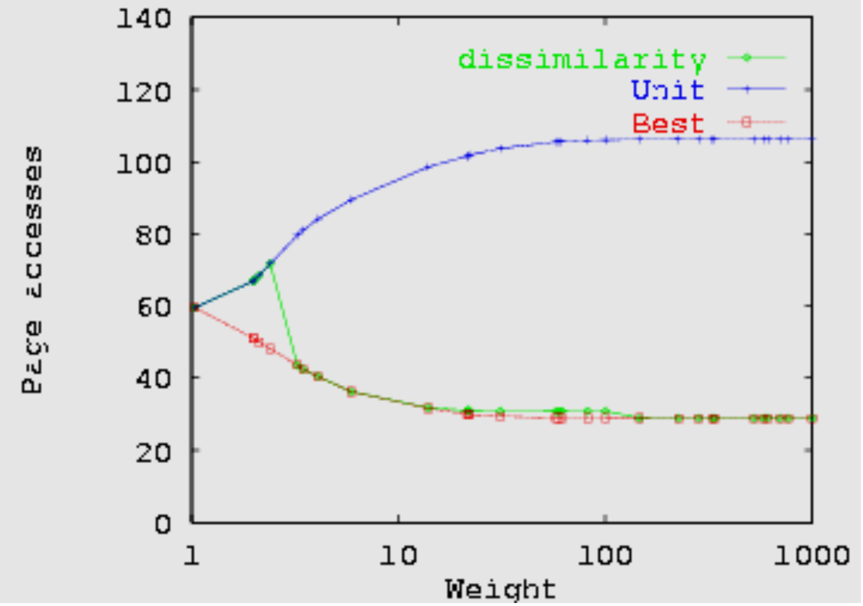
Rate of filtered exact calculations

- Percentage of filtered exact distance calculations
  - The efficiency of MBB-MBS decreases as matrix flatness grows
  - STT effectively filters exact distance calculations for all queries

Performance of MSTT



CPU time (d = 27)



Number of page accesses (d = 27)

- Low search cost
  - Compared with the structure by the Euclidean distance function, MSTT reduces both CPU time and the number of page accesses
  - MSTT constructs various structures
  - Dissimilarity function chooses structures well suited to the query matrix.