

Multi-Vector Feature Space Based on Pseudo-Euclidean Space and Oblique Basis for Similarity Searches of Images

Yasuo Yamane

Tadashi Hoshiai

Hiroshi Tsuda

Language Processing Laboratory
Fujitsu Laboratories Ltd.

Kaoru Katayama

Manabu Ohta

Hiroshi Ishikawa

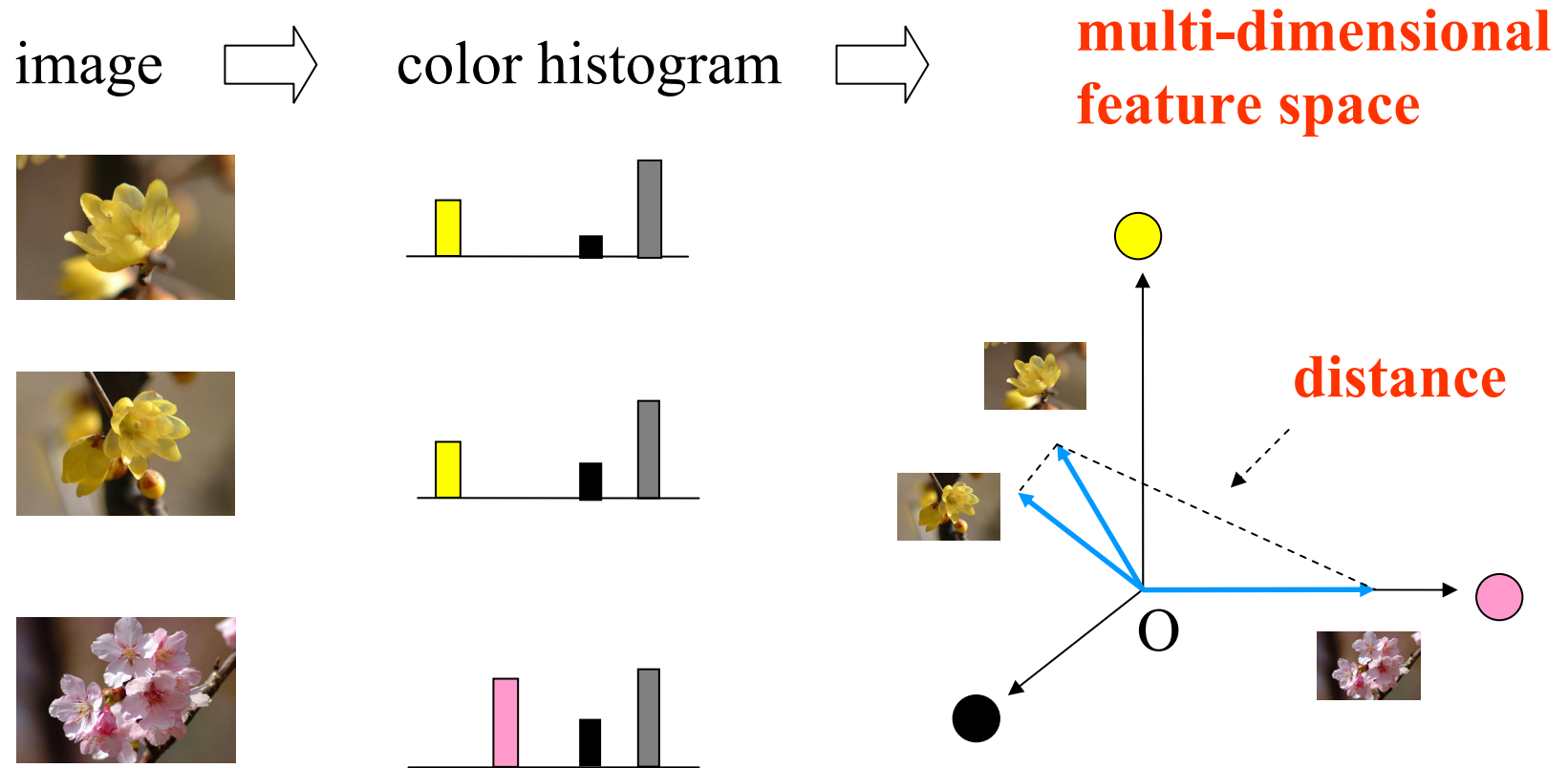
Graduate School of Engineering
Tokyo Metropolitan University

Contents

1. Problem
2. Multi-Vector Feature Space
3. Comparison with Earth Mover's Distance
4. Performance
5. Summary

1. Problem

Similarity Search of Images



Precision and performance: Important issues

Representative Distances and Our Proposal

Rubner (Stanford Univ.)

QBIC system (IBM)

Representative
distances

**Earth Mover's
Distance (EMD)**

**Quadratic-
Form Distance**

formalize

compare

solution

approximation

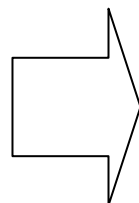
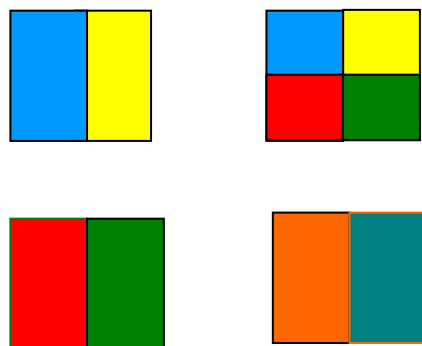
Our proposal

**Multi-Vector
Feature Space
(MVPO)**

D-distance

Problem of Quadratic-form Distance

Disappearance of Discriminability



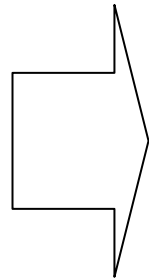
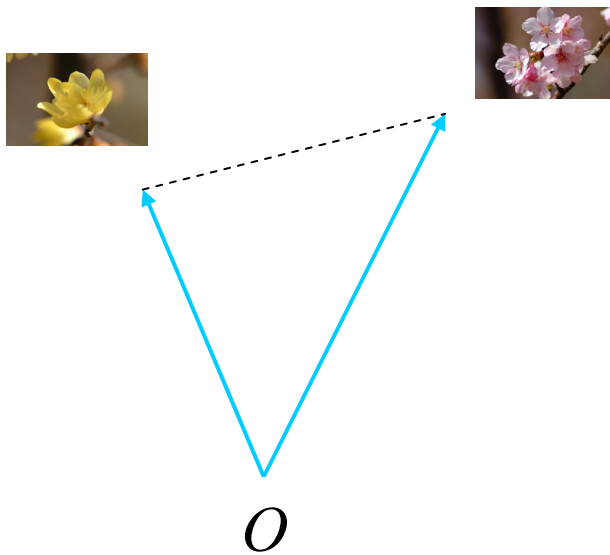
Distance = 0

Dissimilar Images

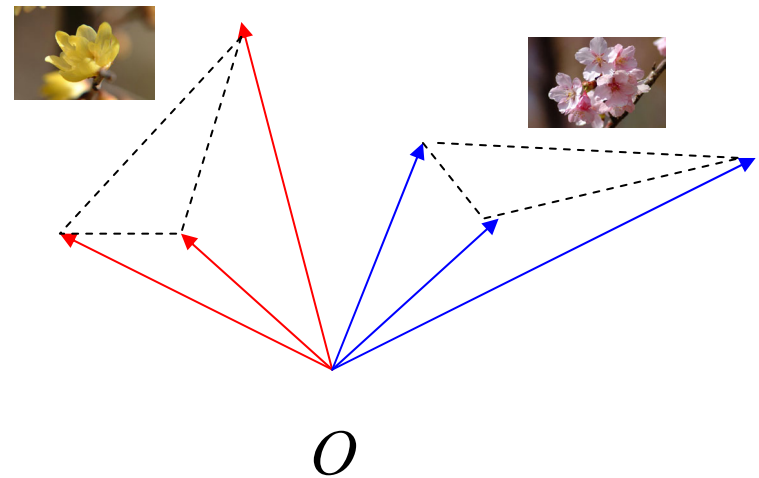
Judged to be the same

MVPO: From Point to Solid

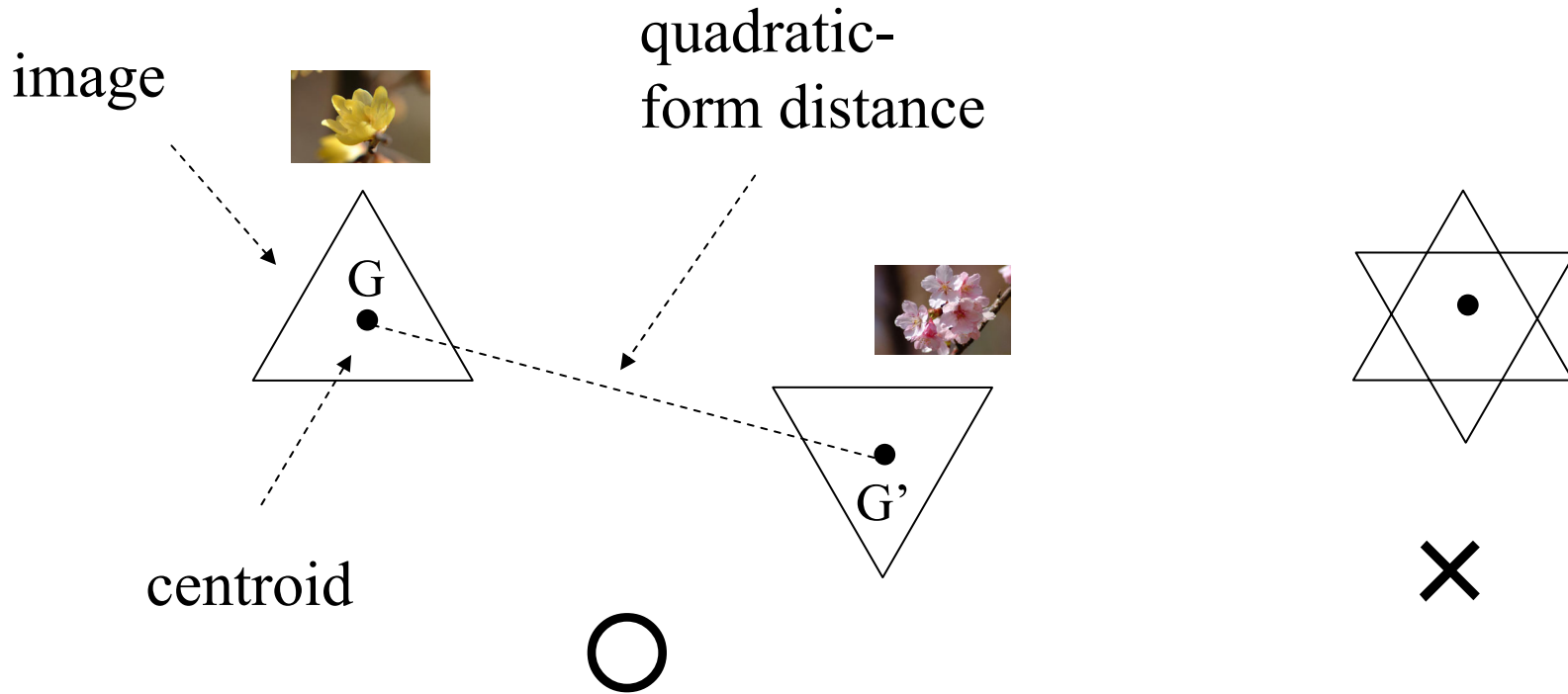
Distance between
Points (Vectors)



Distance between
Solids (Vector-Sets)



Why Solids ?

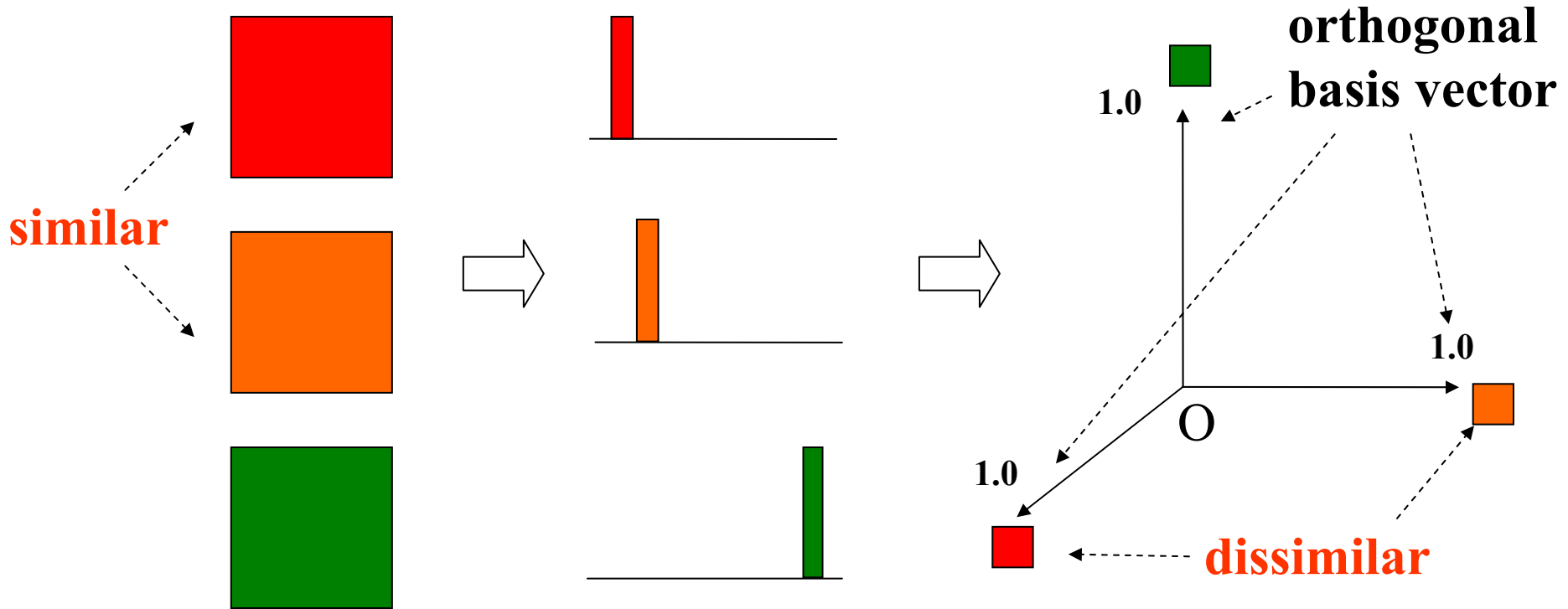


**Disappearance of
Discriminability**

Problem of Orthogonal Basis + Euclidean Distance

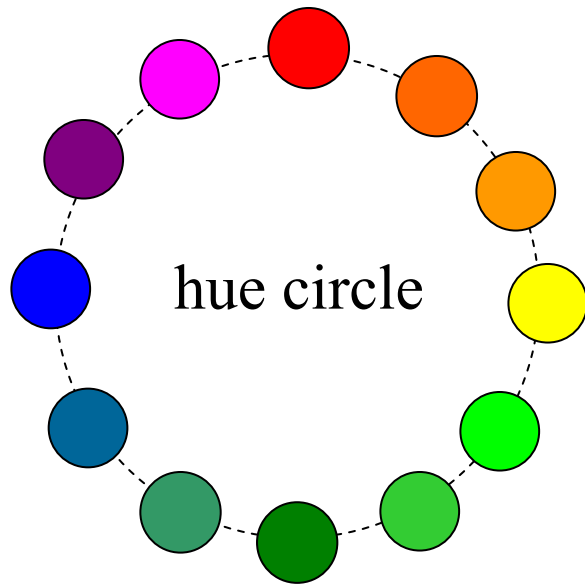
Similar images can be judged to be **dissimilar**

Images

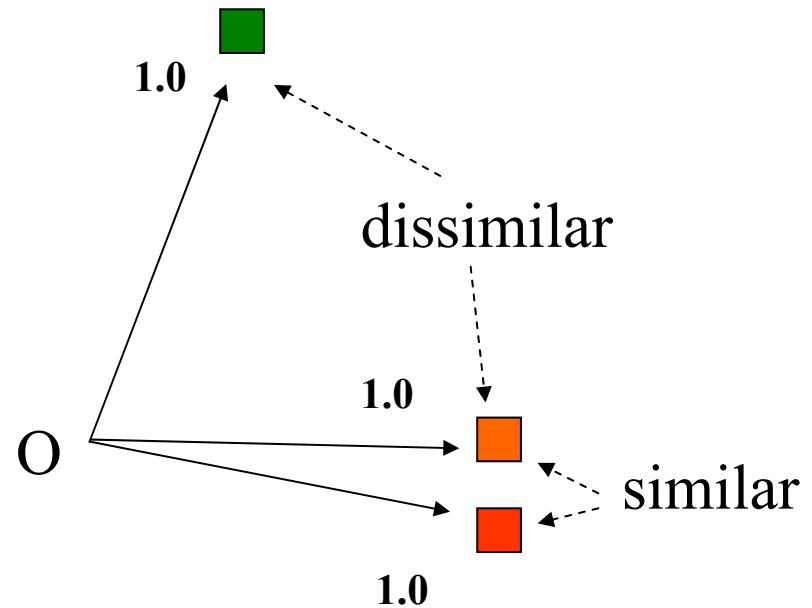
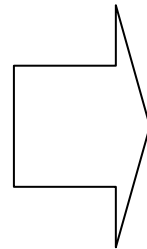


Distances are **equal** !

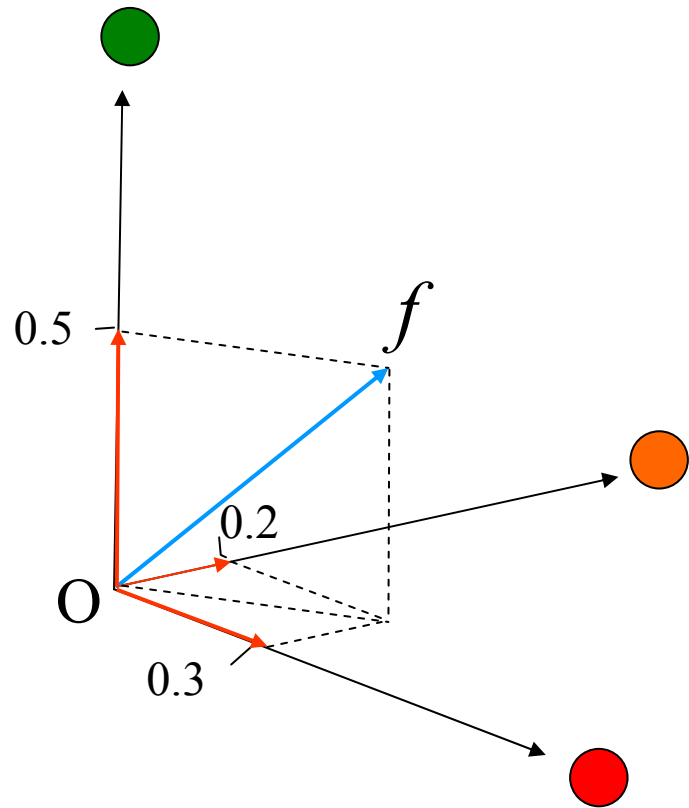
Solution based on Oblique Basis



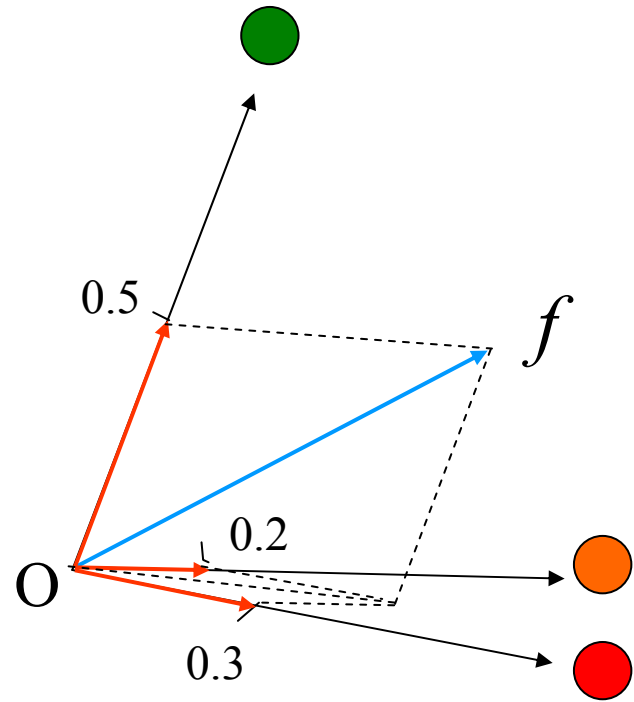
Reflection
of Similarity



Orthogonal and Oblique Representation

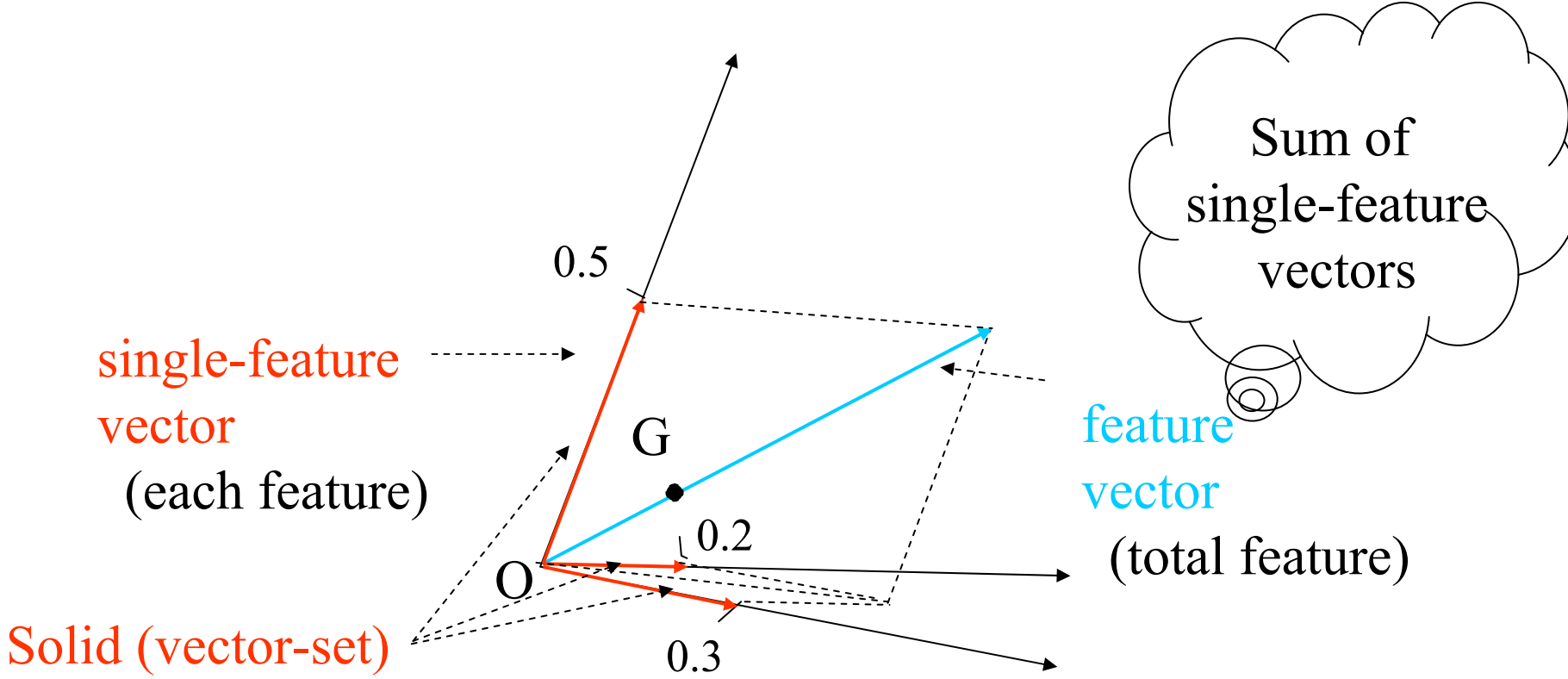


Orthogonal



Oblique

Relation between Solid and Feature Vector



centroid G of solid = feature vector / dimension

Quadratic-form Distance

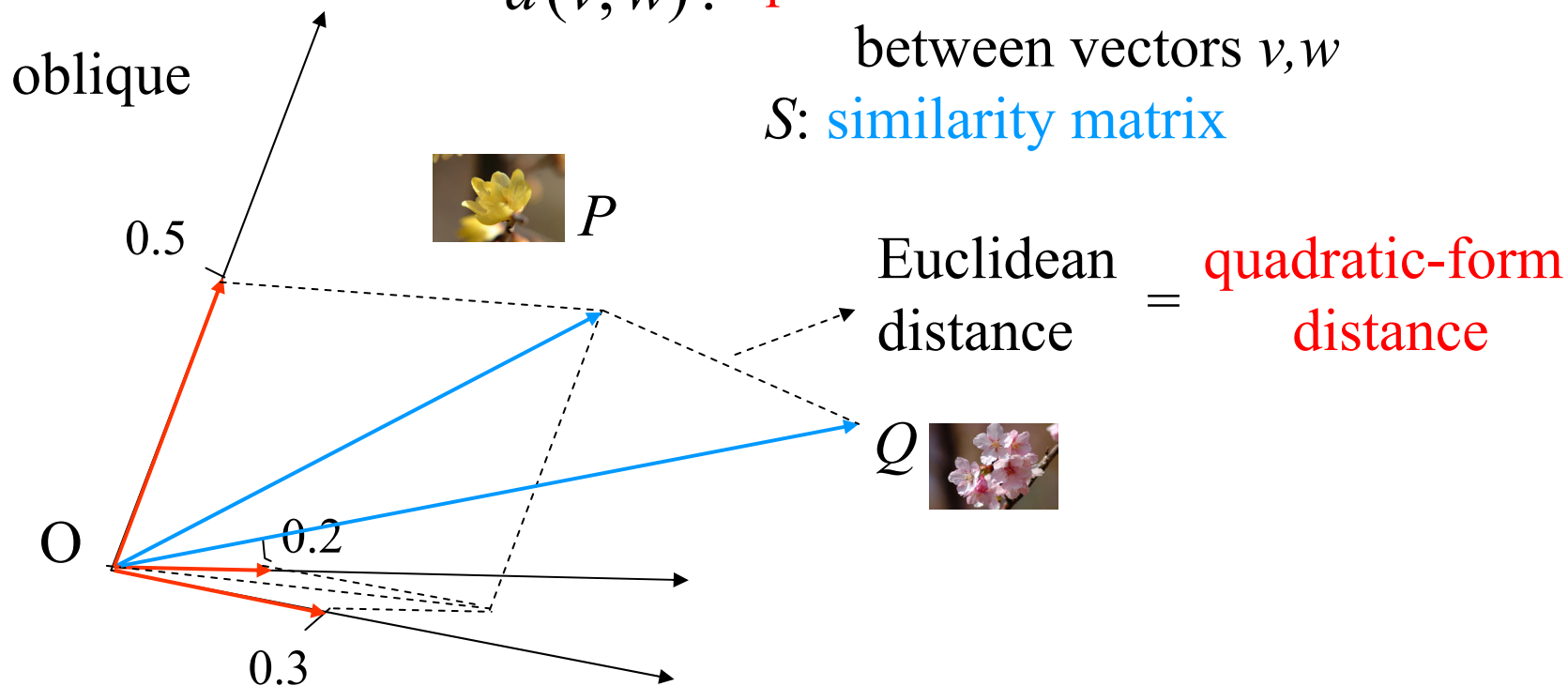
Used in IBM's QBIC (Query By Image Content) system

$$d(v, w)^2 = {}^t(v - w)S(v - w)$$

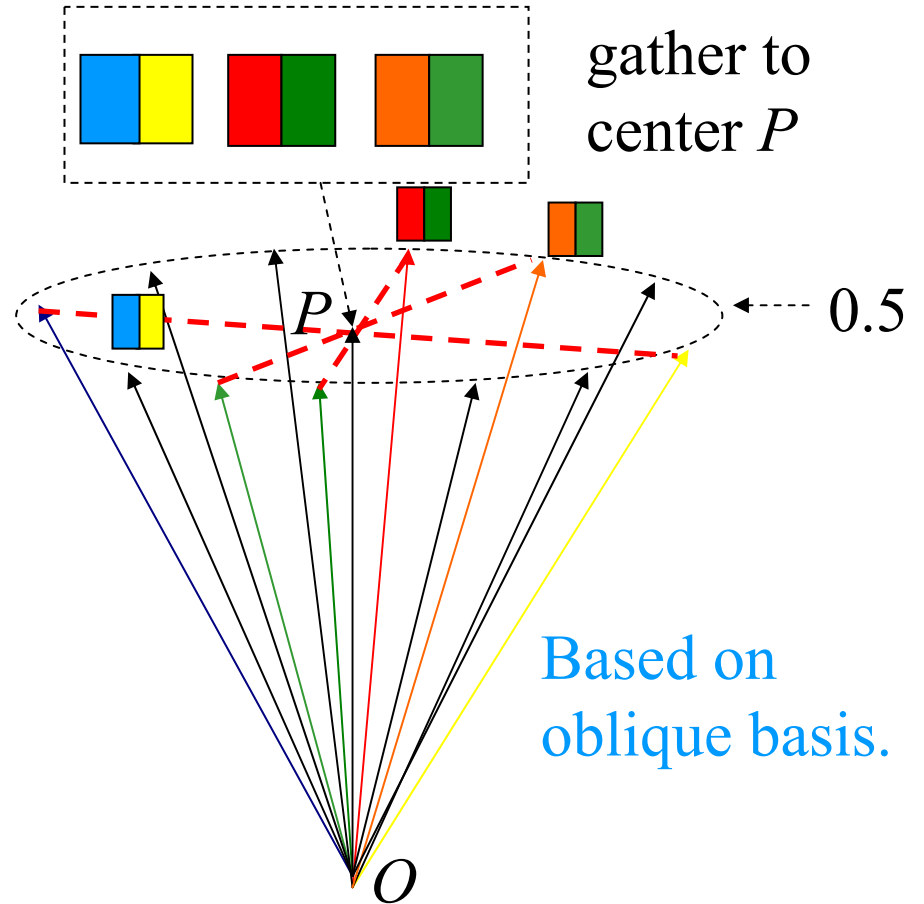
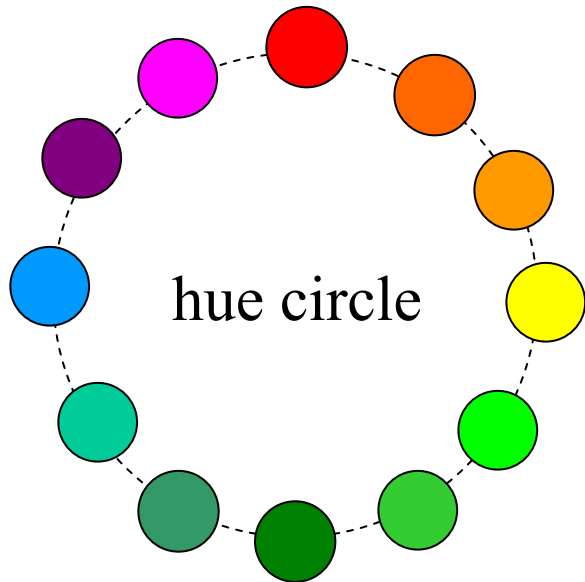
$d(v, w)$: **quadratic-form distance**

between vectors v, w

S : **similarity matrix**



Cause of Problem



Solid : Precise

Composing feature vector: Cause

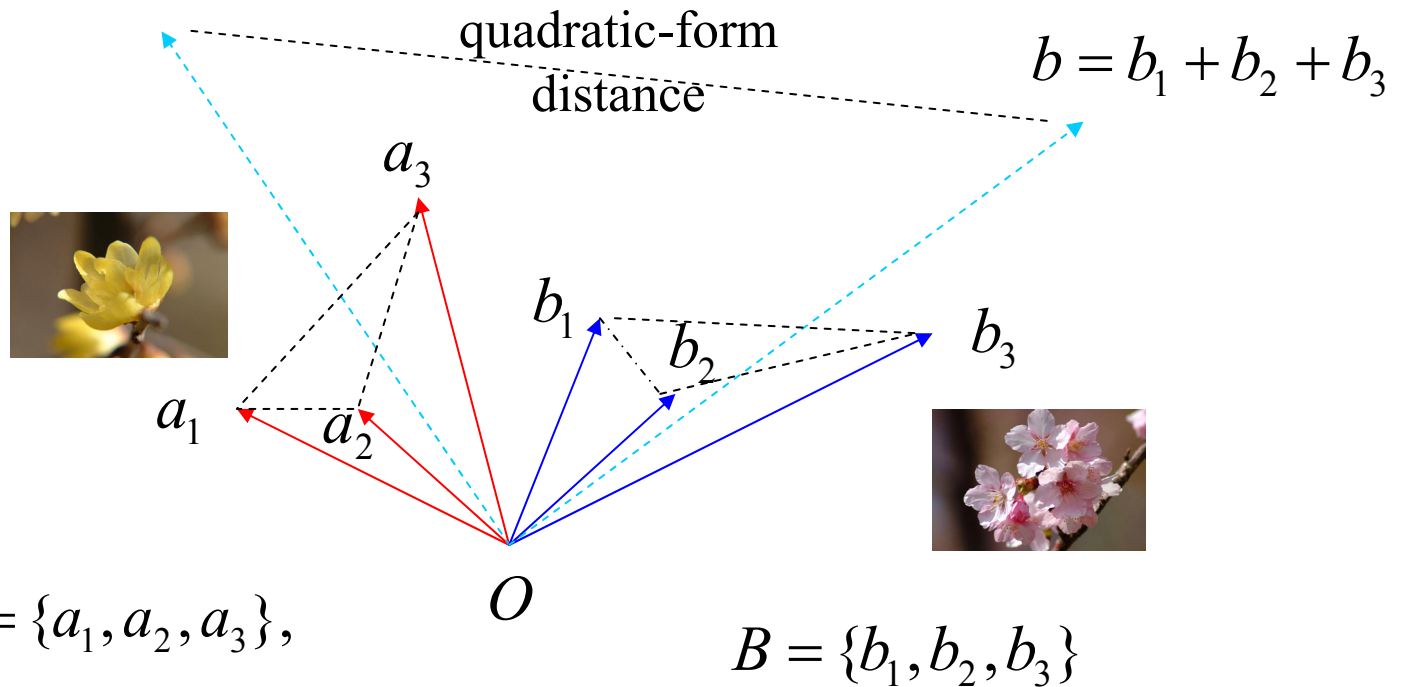
2. Multi-Vector Feature Space

MVPO (Multi-Vector Feature Space
Based on
Pseudo-Euclidean Space and
Oblique Basis)

Example of MVPO

$$a = a_1 + a_2 + a_3$$

$$b = b_1 + b_2 + b_3$$



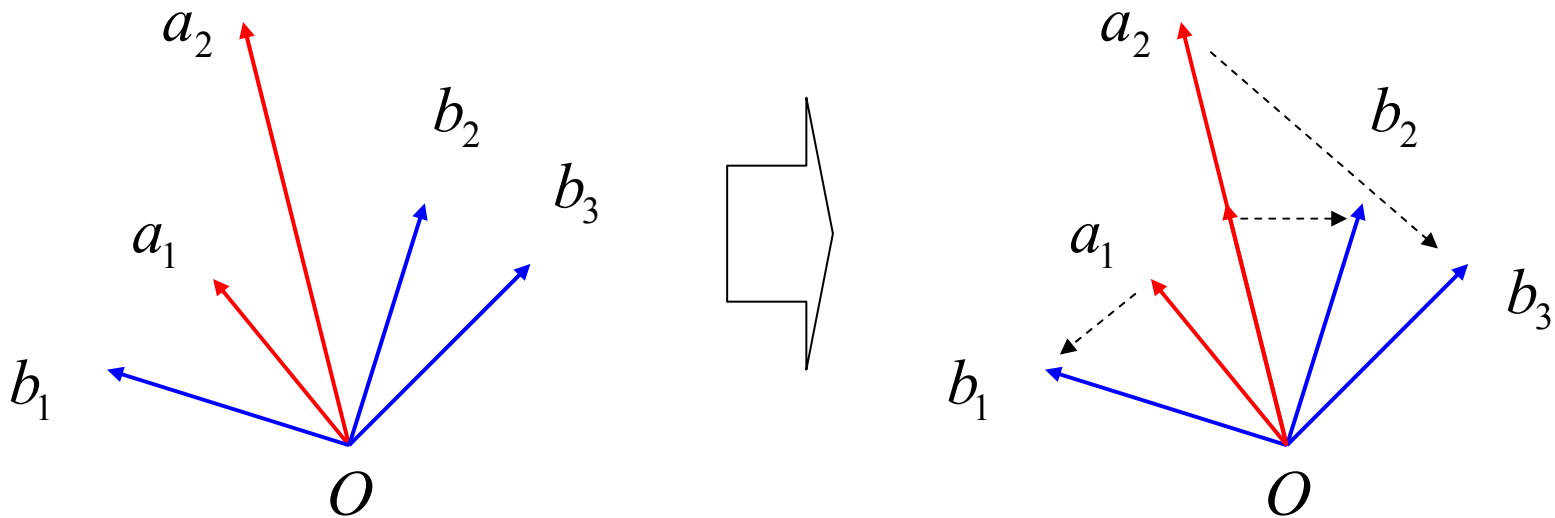
How should we define **the distance between vector-sets** ?

D-distance

transformation cost of a solid

Break solid A , and **construct** solid B

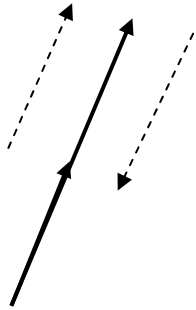
$$A = \{a_1, a_2\}, \quad B = \{b_1, b_2, b_3\}$$



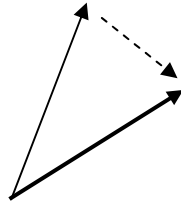
Cost of Vector Operation

Cost = displacement of vector's terminal point

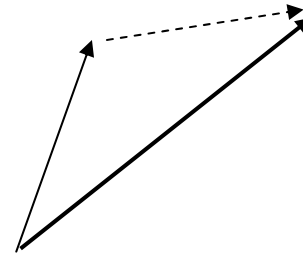
changing length



rotation

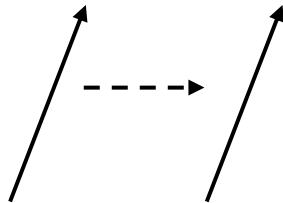


changing length
+ rotation

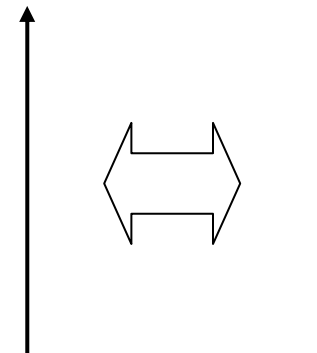


Cost = 0

parallel movement

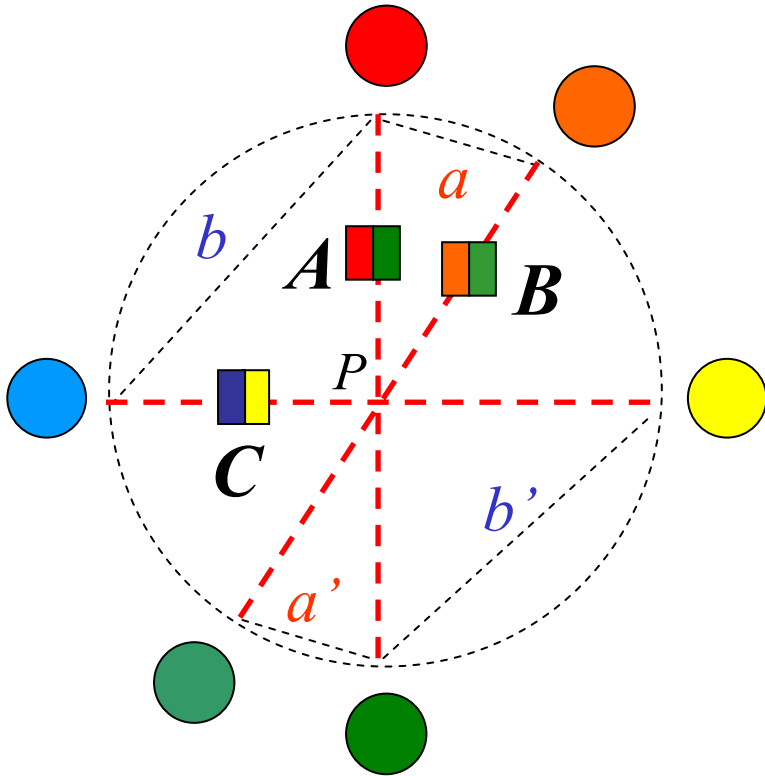


partition/composition

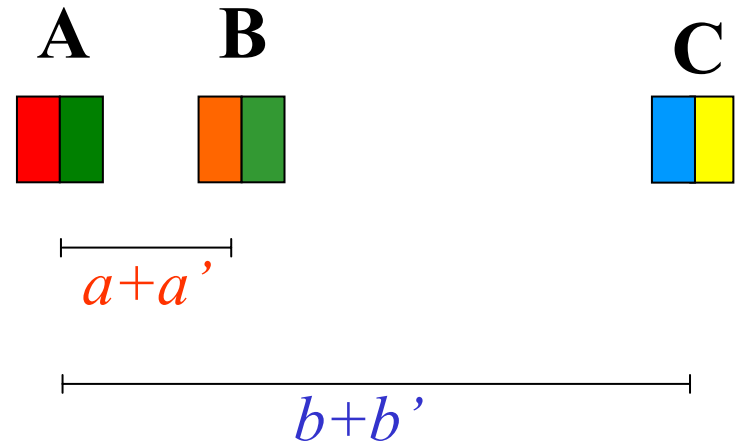


Solved?

Overhead view
of the cone-shaped diagram

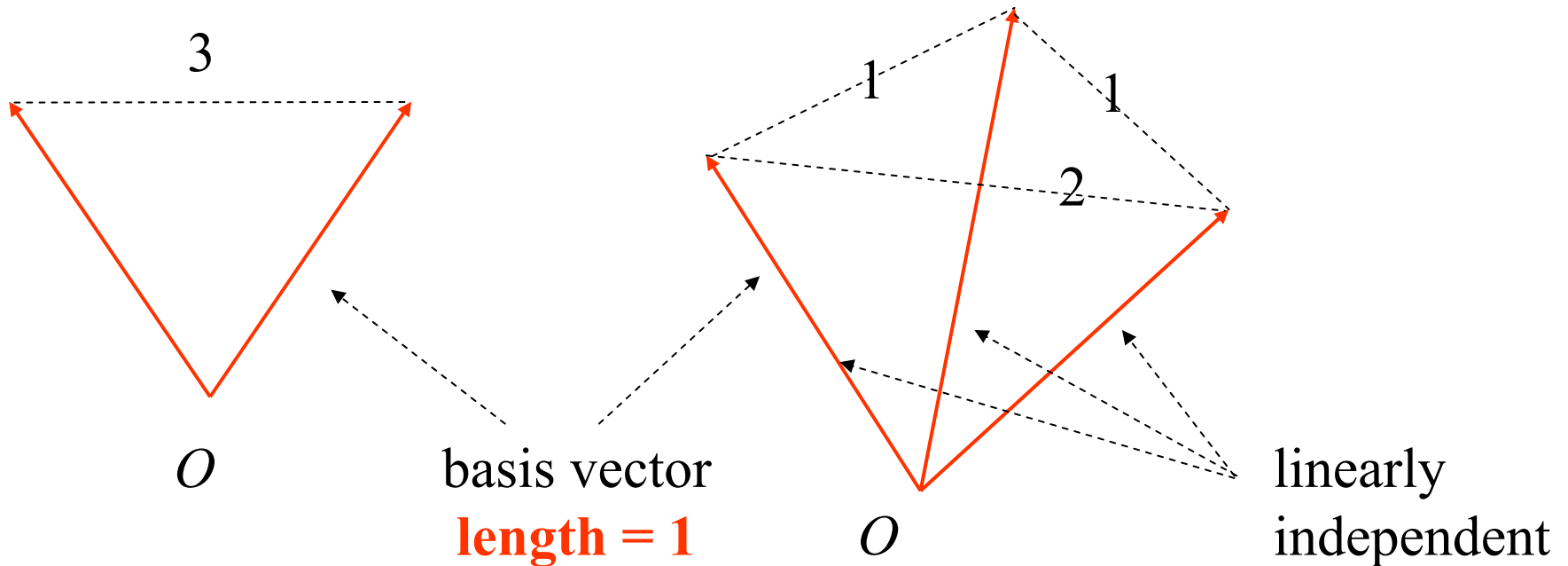


D-distances
between vector-sets



Why Pseudo-Euclidean Space?

Any similarity + linear independence



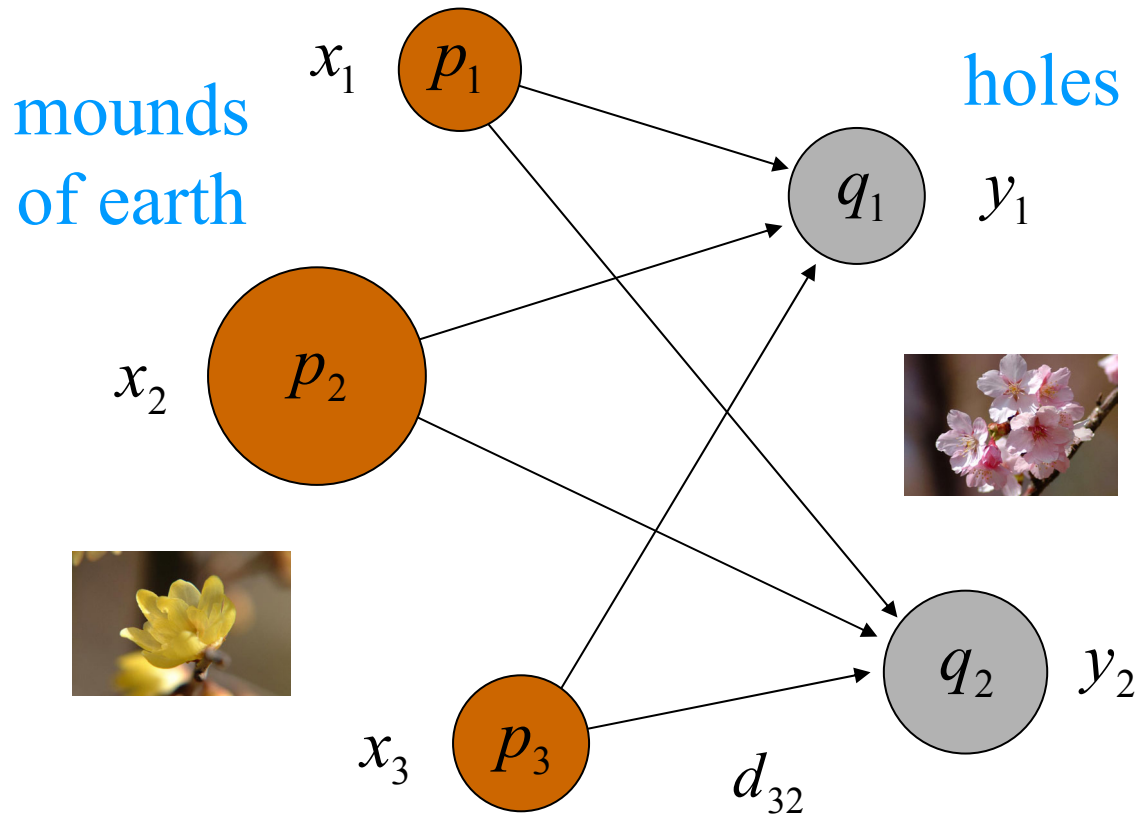
3. Comparison with Earth Mover's Distance (EMD)

EMD

- 97~98 Yossi Rubner (Stanford University)
- **Distance between sets**, like D-distance
- Experimentally, **more precise than other distances** such as quadratic-form distance [Rubner01]
- **Theoretical justification** has not been shown [Levina01]

Definition of EMD

Minimum cost of transportation from mounds to holes



p_i : earth position

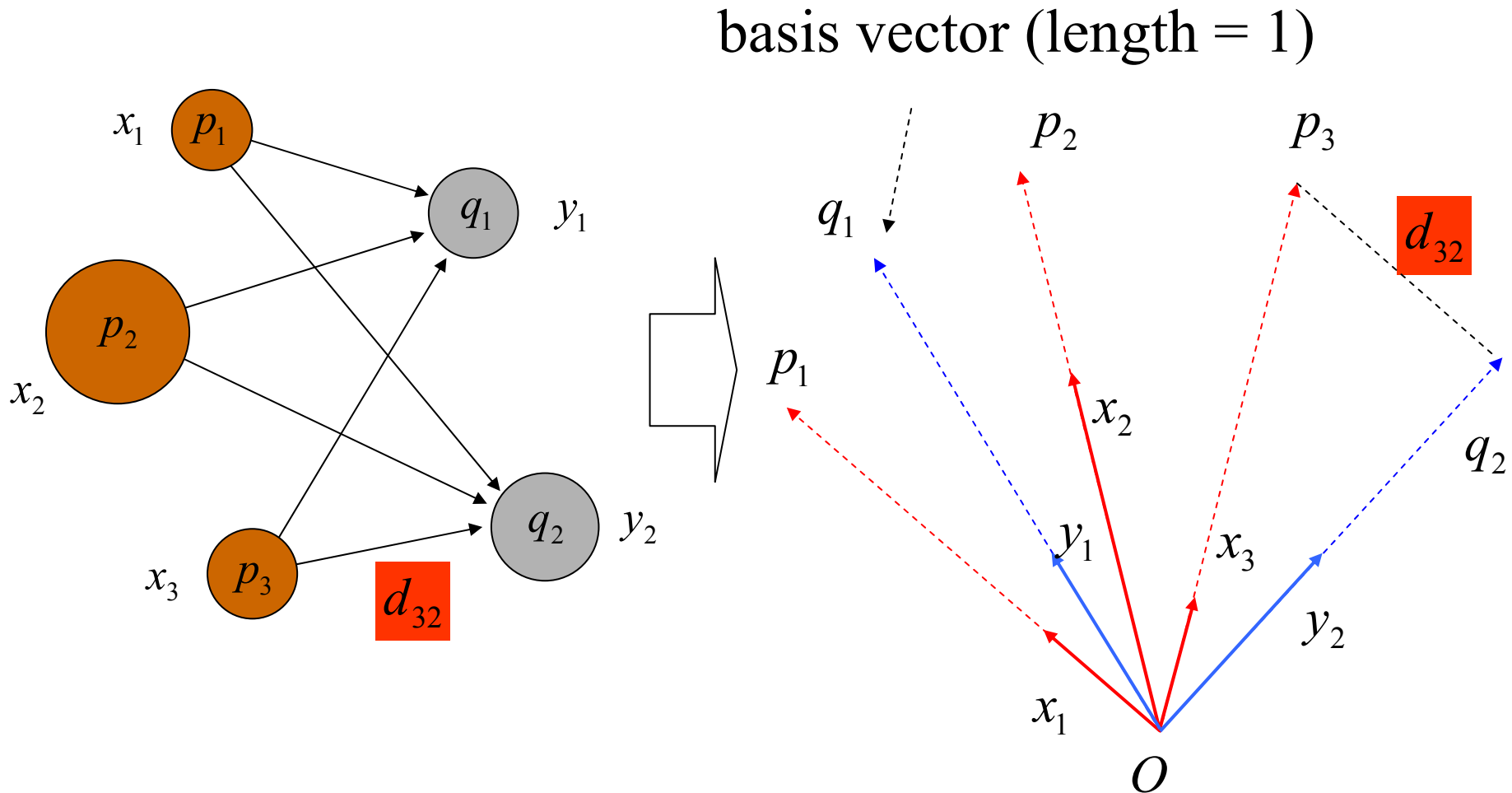
q_j : hole position

x_i : earth quantity

y_j : hole capacity

d_{ij} : distance between
 p_i and q_j

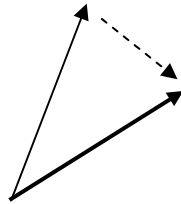
Formalization based on MVPO



Vector Operation of EMD

Cost = displacement of terminal point

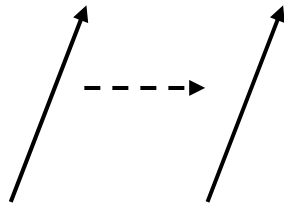
rotation



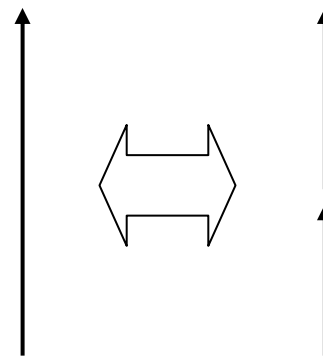
**Changing length
is not allowed**

Cost = 0

parallel movement



partition/composition



Comparison between D-distance and EMD

	D-distance	EMD
matching	total	partial
changing length	allowed	not allowed
similarity matrix	restricted	not restricted
computation	nonlinear	linear
	programming	programming

In case total masses of 2 images are equal,

$$\mathbf{D-distance} \leq \mathbf{EMD}$$

4. Performance

Practicality of D-distance

issue	Practicality
Exact Computation	× or Δ
Approximation	○
Multi-dimensional Indexing	× ?
Parallel Processing	Δ

○: affirmative Δ : neutral ×: negative

Exact Computation

Quadratic-form distance < EMD < D-distance
Linear programming Nonlinear programming

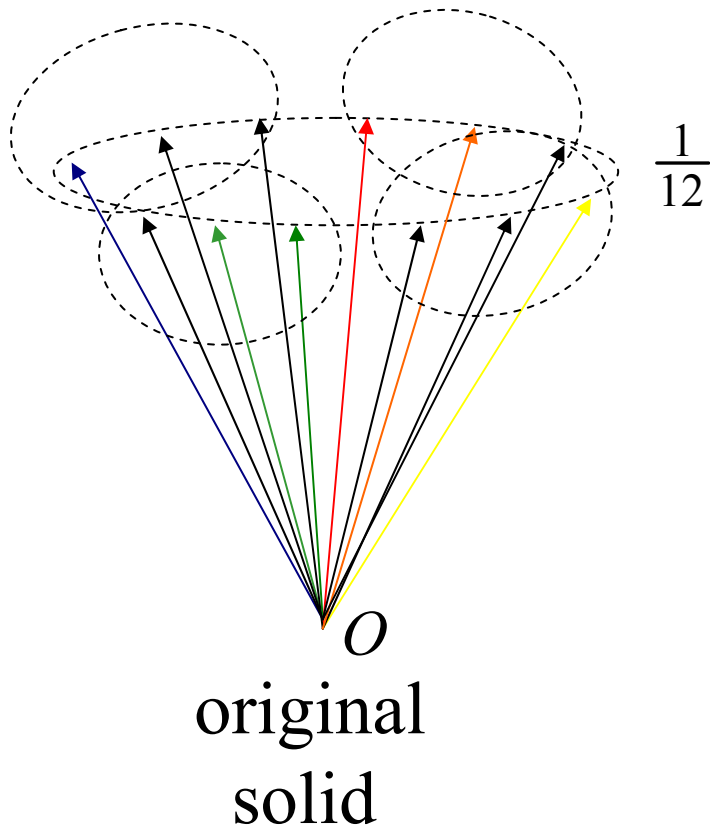
Approximation

Two approaches:

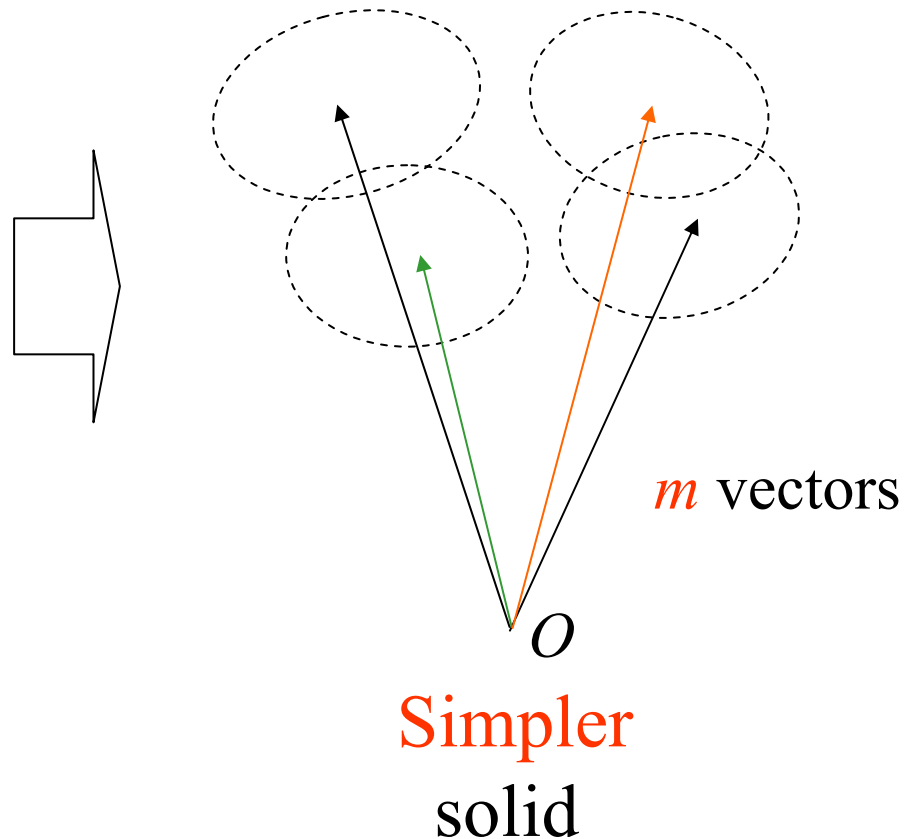
Solids are (1) **changed**
(2) **not changed**

Approximation of Solids

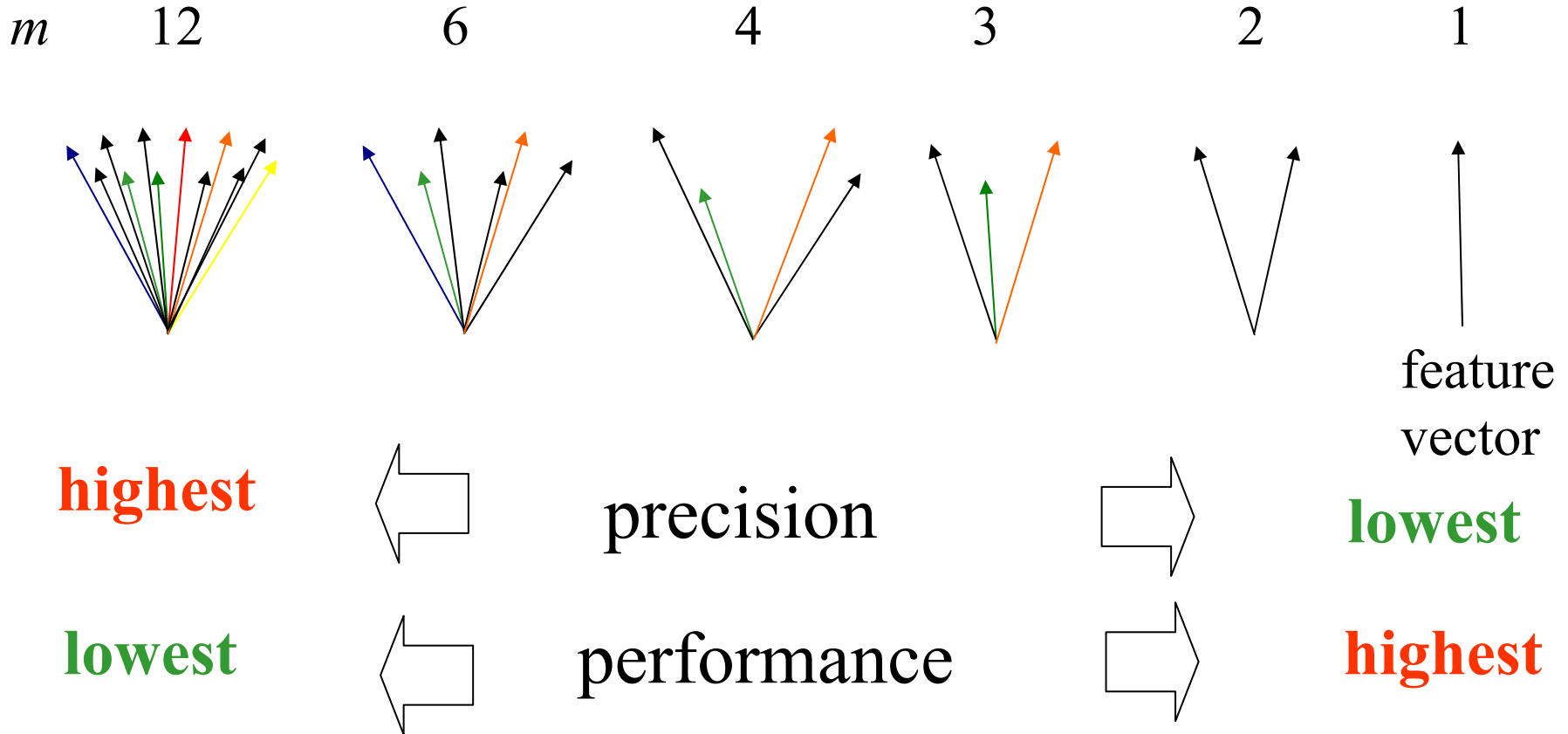
Partition into m vector-sets



Approximate each vector-set
by its centroid



Adjustability



Practical on a large scale, approaching full throughput

Multi-dimensional Indexing

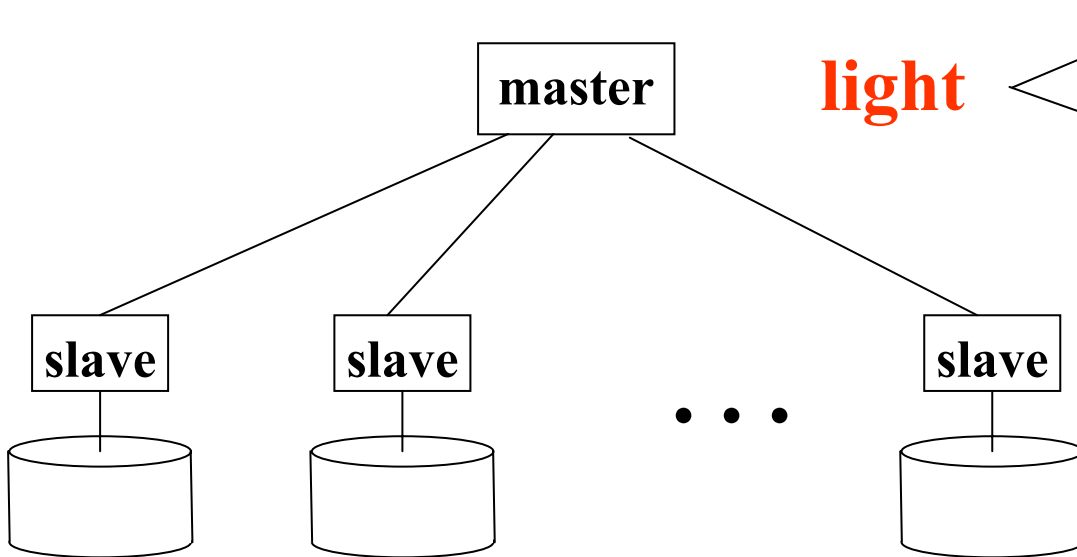
Two major difficulties:

- (1) Pseudo-Euclidean Space
- (2) Solid

In addition,

Curse of Dimensionality

Parallel Processing



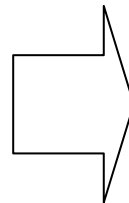
light

- 1) Distribute **query image**
- 3) Compute **top k** ranking

- 2) Compute **D-distance** between **query image** and each **stored image** **exactly** or **approximately**

Images are **partitioned equally**

Data independence: **high**
Master processing: **light**



High degree of parallelism

5. Summary

Conclusion

Proposal of MVPO and D-distance

- **Geometrical views** to quadratic-form distance and EMD
- **Theoretical explanation** of discriminability disappearance, and why EMD outperforms quadratic-form distance.
- Comparison with EMD
changing vector length: essential difference

Discussion of Performance Issues of MVPO

Approximation of solids: practical

even on a large scale

adjustable between precision and performance
approaching **full throughput**

Challenges for the Future

- **Experimental verification** of D-distance
(by real images)
- Application of MVPO to **documents**
Discriminability disappears ?
- Finding efficient **algorithms**
to exactly or approximately compute **D-distance**