TOSHIBA
Leading Innovation >>>

Feature Vector Compression based on Least Error Quantization

Tomokazu Kawahara and Osamu Yamaguchi

TOSHIBA Corporation
Corporate Research & Development Center

Biometrics Workshop (CVPR2016)
June 26, 2016

Outline

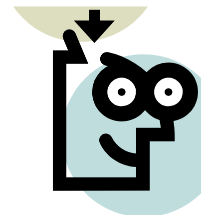
- **Motivation**
- **Our approach**
- **The proposed compression**
  - Concept and illustration of our approach
  - Solution of the global minimum
  - Data structure of a compressed vector
  - Distance calculation between compressed vectors
- **Experiments on face recognition**
  - Local Binary Pattern with L_1-distance
  - Deep learned feature with L_2-distance
- **Conclusion**

Outline

- **Motivation**
- **Our approach**
- **The proposed compression**
  - Concept and illustration of our approach
  - Solution of the global minimum
  - Data structure of a compressed vector
  - Distance calculation between compressed vectors
- **Experiments on face recognition**
  - Local Binary Pattern with L_1-distance
  - Deep learned feature with L_2-distance
- **Conclusion**

Motivation

- **The number of template for each person increases**
  - Multi-biometrics
  - Multi-template authentication



Face Fingerprint …

- **The smaller template size is, the better**
  - IC-Card
    - The memory size of IC-Card is limited.
  - Verification in video surveillance system
    - A large number of faces are identified.

Our approach

- **Reducing the memory size of the high performance features is important.**
- **New methods and new fascinating features are proposed one after another.**

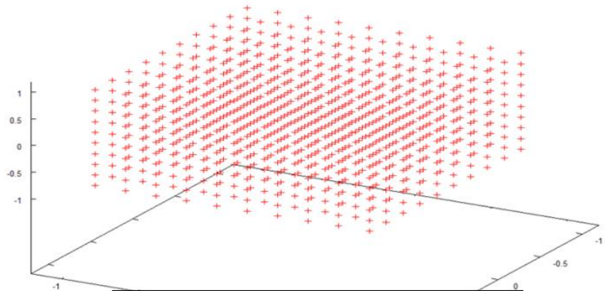
⬇

We propose method-independent feature vector compression based on least error quantization.

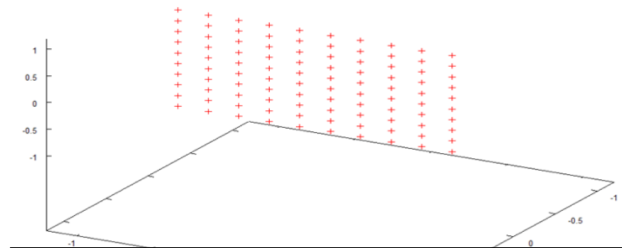
- Size : 1/5 - 1/10 and keep the recognition performance and calculation speed
- Applying our method to the state-of-the-art feature

Type of compressions

- **Statistical dimensionality reduction**
  - Ex. Subspace projection, …

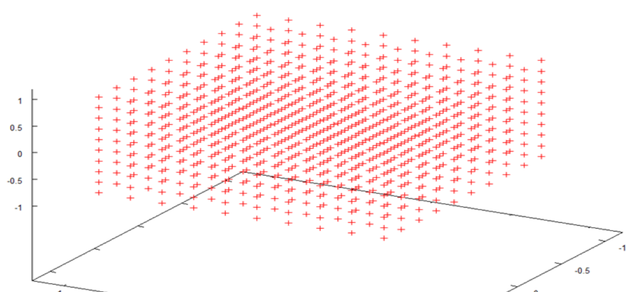


Original (3D-grid)

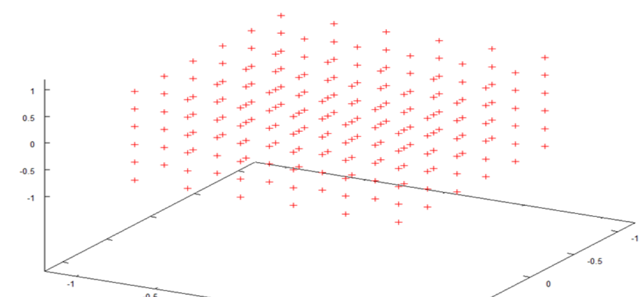


Subspace projection (2-dim)

- **Quantization**
  - Ex. Uniform quantization



Original (3D-grid)



Uniform quantization

Compression by quantization

- **Quantization**
  - Uniform quantization
  - Non-uniform quantization

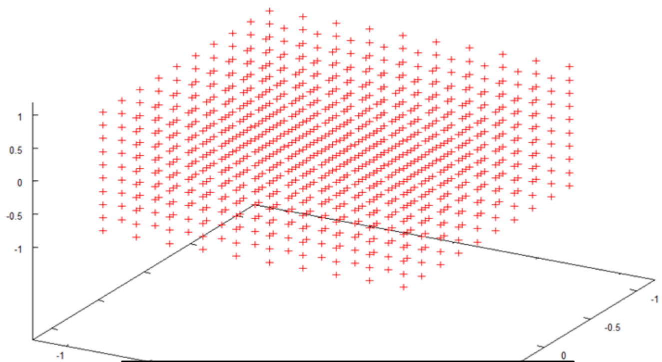
    Otsu's quantization (Image Processing like binarization)

    Extended Otsu's quantization : the proposed method

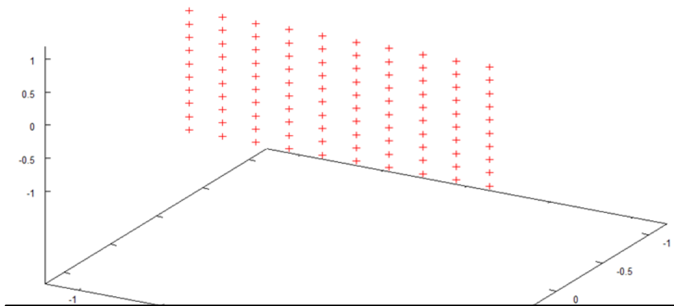
	Training data	Compression Error	Metric for recognition
Dimensionality reduction	Need	Small	Any
Uniform quantization	Not Need	Large	Any
Otsu's quantization	Not Need	Very small (L_2-error)	Optimized L_2-distance
The proposed method	Not Need	Very small (L_p-error) $(0 < p \leq \infty)$	Optimized L_p-distance $(0 < p \leq \infty)$

The proposed method can be optimized for L_p-distance, and we propose data structure and distance calculation for recognition.

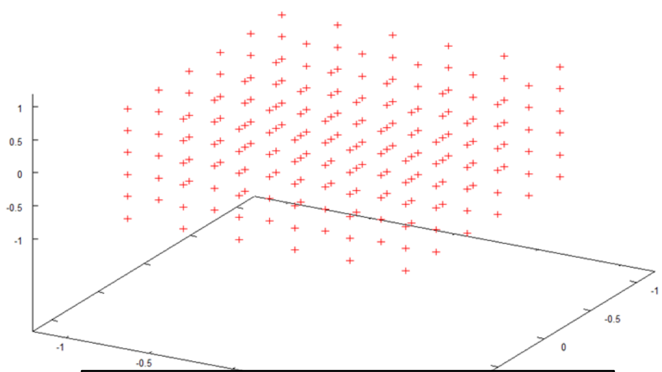
Our compression method



Original (3D-grid)

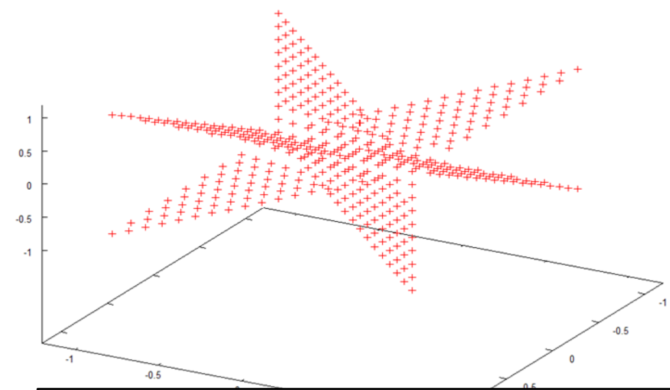


Subspace projection (2-dim)



Uniform quantization

The proposed method



Non-uniform quantization
with least error

Outline

- Motivation
- Our approach
- **The proposed compression**
  - Concept and illustration of our approach
  - Solution of the global minimum
  - Data structure of a compressed vector
  - Distance calculation between compressed vectors
- Experiments on face recognition
  - Local Binary Pattern with L_1-distance
  - Deep learned feature with L_2-distance
- Conclusion

Concept of our approach

- **Solve the minimum clustering error problem and apply the solution to quantization**

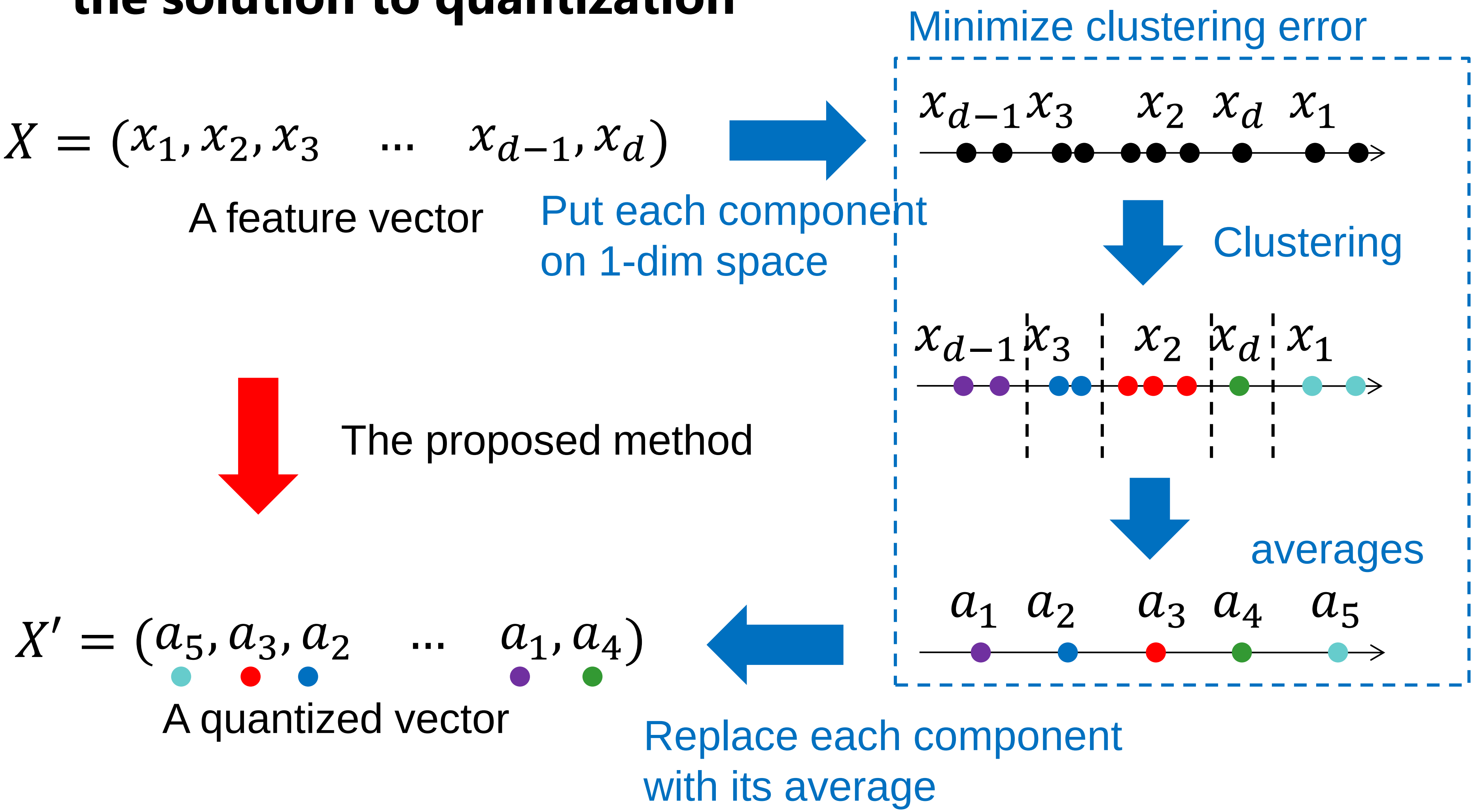
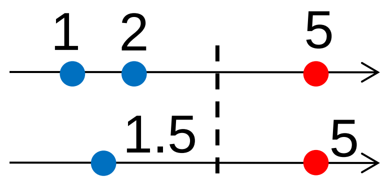


Illustration of our approach

- **Quantize 3-dimensional vectors with 2 Averages**

$X = (1 \quad 2 \quad 5)$

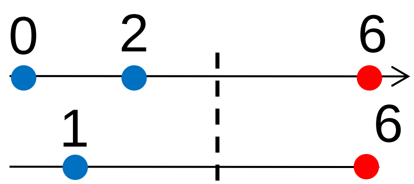


Averages

$X' = (1.5 \quad 1.5 \quad 5)$

Projection to 2-dim subspaces (a, a, b)

$Y = (6 \quad 2 \quad 0)$

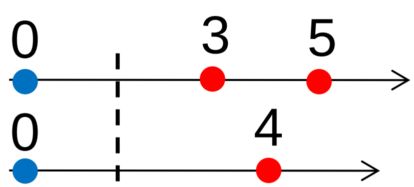


Averages

$Y' = (6 \quad 1 \quad 1)$

Projection to 2-dim subspaces (b, a, a)

$Z = (5 \quad 0 \quad 3)$

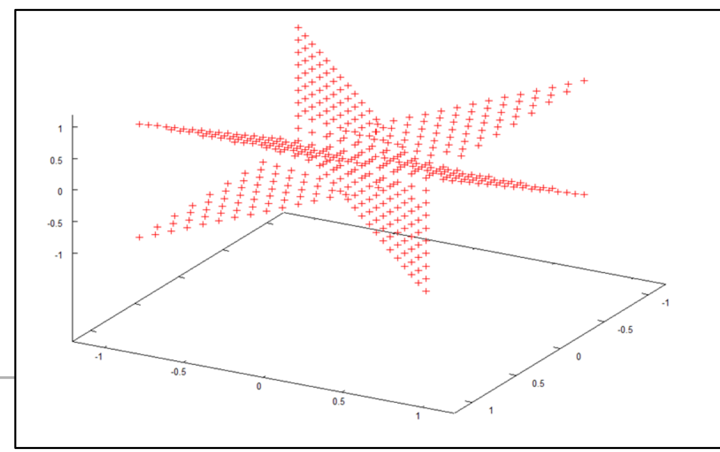


Averages

$Z' = (4 \quad 0 \quad 4)$

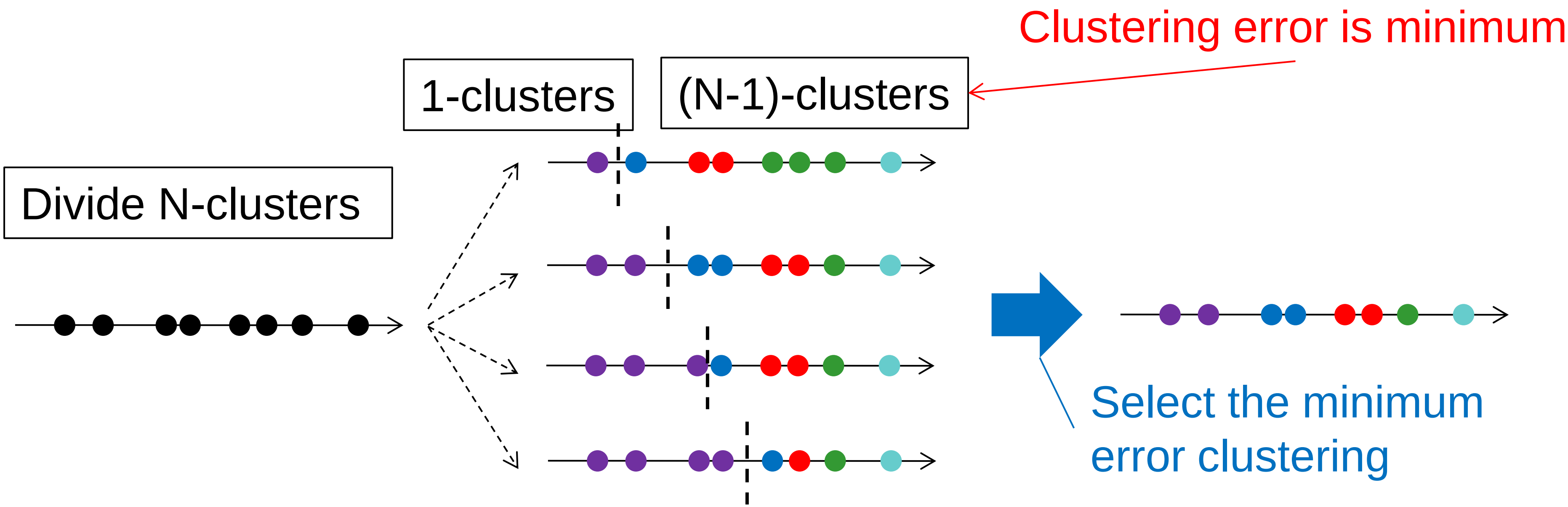
Projection to 2-dim subspaces (a, b, a)

The proposed method projects vectors to multiple subspaces



Solution the minimum clustering error problem

- **Solve the problem by dynamic programing algorithm**



N-clustering problem is solved by (N-1)-clustering problems

$$\sum_{i=1}^{N}\sum_{k\in C_i}|x_k - a_i|^p \qquad L_p\text{-clustering error } (0 < p \leq \infty)$$

The proposed compression

- **Compress a feature vector (level = N)**

Minimize clustering error
for each vector

Average table A_X

Index	1	2	…	N
Average	$\boldsymbol{a}_1$	$\boldsymbol{a}_2$	…	$\boldsymbol{a}_N$

Replace each component with its average

$$X = (x_1 \quad \dots \quad x_d) \longrightarrow X' = (a_{i_1} \quad \dots \quad a_{i_d})$$

A feature vector

A compressed vector

- **Data structure of a compressed vector**

$A_X = \{a_1 \dots a_N\}$ Average table

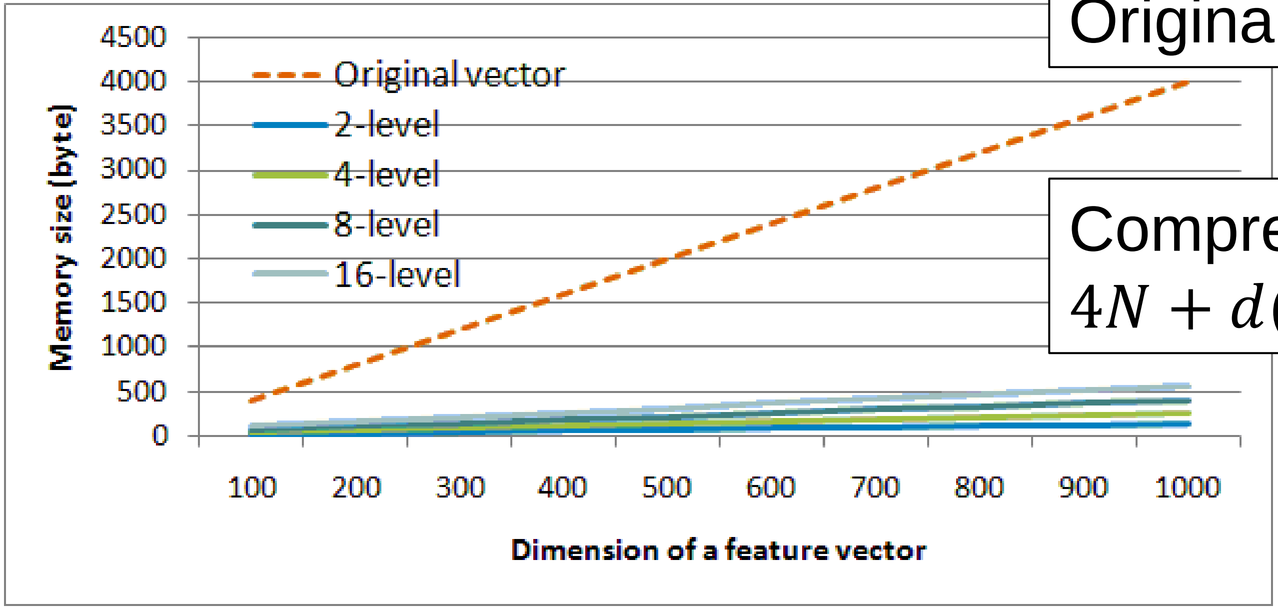
$I_X = (i_1 \dots i_d)$ Index vector

Memory size of a compressed vector

- **Data size of each component**

	Range	Size (byte)	Number	Total (byte)
Average table $A_X = \{a_1 \dots a_N\}$	$-\infty < a < \infty$	4 (float)	N	$4N$
Index vector $I_X = (i_1 \dots i_d)$	$1 \leq i \leq N$	$(\log_2 N)/8$ (binary)	d	$\mathrm{d}(\log_2 N)/8$

- **Data size of vector**



Original: $4d$ byte

Compressed: $4N + d(\log_2 N)/8$ byte

Proposed distance calculation

- **Calculate L_p-distance without decoding**

1. Calculate all average pairs ${A_{ij}}^p = |a_i - b_j|^p$ and generate table

	1	2	…	N
1	${A_{11}}^p$			${A_{1N}}^p$
2				
…				
N	${A_{N1}}^p$			${A_{NN}}^p$

$$X' = \begin{cases} A_X = \{a_1 \dots a_N\} \\ I_X = (i_1 \dots i_d) \end{cases}$$

$$Y' = \begin{cases} B_Y = \{b_1 \dots b_N\} \\ J_X = (j_1 \dots j_d) \end{cases}$$

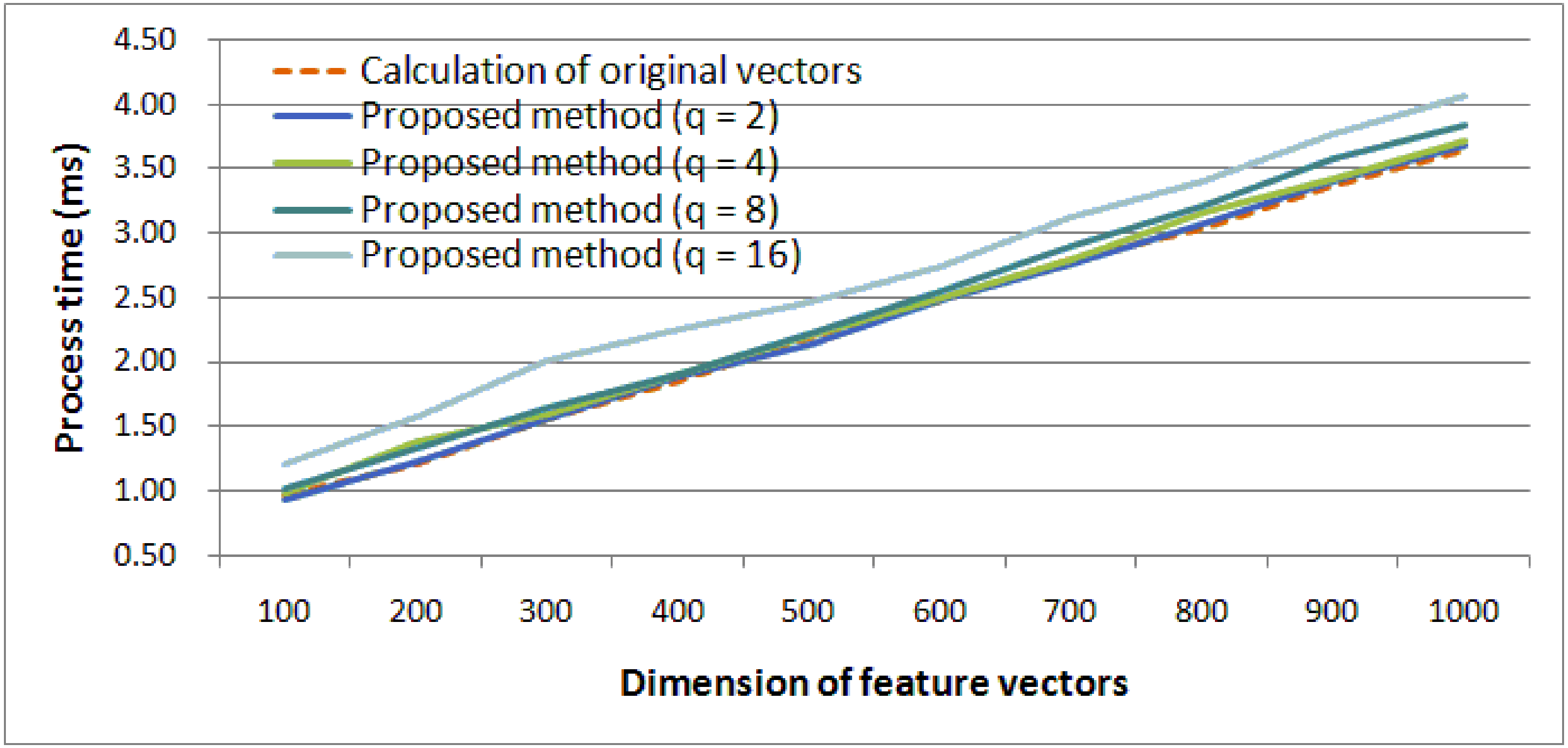
select

2. Select and add each value

$$\boldsymbol{L_p}\text{-distance} \quad \|X' - Y'\|_p{}^p = \sum_{k=1}^{d} |a_{i_k} - b_{j_k}|^p = \sum_{k=1}^{d} \boxed{{A_{i_k j_k}}^p}$$

Process time of the proposed calculation

- Compressed vector (Level is 2, 4, 8, 16) and original vector
- Intel Core 2 Extreme PC with 3GHz CPU



The proposed calculation is as fast as calculation of original vectors

Outline

- Motivation
- Our approach
- The proposed compression
  - Concept and illustration of our approach
  - Solution of the global minimum
  - Data structure of a compressed vector
  - Distance calculation between compressed vectors
- **Experiments on face recognition**
  - Local Binary Pattern with L_1-distance
  - Deep learned feature with L_2-distance
- **Conclusion**

Experiment 1 (Evaluation of recognition with L_1-distance)

- **Feature: Local Binary Pattern**
  - Crop 96 x 96 pixels after 3D-normalization ([16])
  - Divided a LBP pattern face to 8 $\times$ 8 rectangular regions
    - Dimension of a feature vector: 16,384 = 256 x 8 x 8
- **Metric**
  - L_1-distance
- **Dataset: FERET (Face database)**

	Number of images	Explanation of images
fa	1,196	Frontal images (gallery)
fb	1,195	Alternative facial expression
fc	194	Different lighting condition
dup I	722	Taken later time
dup II	234	Taken at lease a year after

Experiment 1 (Evaluation of L_1 -distances)

- **LBP with L1 distance**
- **Compression level = 16**

Our method

Conventional method

Compression	Distance	fb	fc	dup I	dup II	Mean	Size
L_1	L_1	95.0	51.0	**63.0**	**44.0**	**63.3**	8KB
L_2	L_1	**95.1**	**51.6**	52.6	42.7	62.9	8KB
L_∞	L_1	51.4	37.1	49.0	36.8	49.3	8KB
No-compression	L_1	95.3	50.0	62.6	42.7	62.7	64KB

The LBP with *L*1-compression is better than LBP with L2 (conventional method)

Experiment 2 (Evaluation of deep learned feature)

- **Feature: Deep learned feature**
  - VGG-Face CNN descriptor (one of state-of-the-arts)
  - We use “fc6” feature
  - L2-distance
- **Dataset**
  - FERET
    - fa, fb, fc, dup I, dup II
  - Label the Face in the Wild (LFW)
    - “Deep funneled” images [11]

Experiment 2 (Evaluation of deep features)

- **FERET**

Feature	Level	Compression	Distance	fb	fc	dup I	dup II	Size (Kbyte)
Compressed VGG-Face fc6	2	L_2	L_2	99.9	100	97.4	94.9	0.5
	4	L_2	L_2	99.9	100	97.9	96.6	1.0
	8	L_2	L_2	99.9	100	98.2	97.0	1.6
	16	L_2	L_2	99.9	100	98.2	97.0	2.1
VGG-Face fc6 [23]	No-Compression		L_2	99.9	100	98.2	97.0	16.4
SLBFLE (R=4) [18]	No-Compression			99.9	100	95.2	92.7	

- **LFW (deep funneled)**

Feature	Level	Compression	Distance	Accuracy	Size(Kbyte)
Compressed VGG-Face fc6	2	L_2	L_2	93.08 ± 3.7	0.5
	4	L_2	L_2	96.00 ± 3.5	1.0
	8	L_2	L_2	96.52 ± 3.1	1.6
	16	L_2	L_2	96.62 ± 3.1	2.1
VGG-Face fc6 [23]	No-Compression		L_2	96.62 ± 3.1	16.4

Almost same the accuracy of non-compression

Conclusion

- **We propose a feature vector compression using least error quantization calculated with arbitrary metrics**
  - We extend Otsu's quantization to handle arbitrary metrics
- **We also propose data structure and distance calculation for recognition.**
- **The proposed method can be applied to several features and significantly reduces the memory size without degrading the recognition performance.**
  - In particular, applying our method to the state-of-the-art feature, we are able to obtain the high performance face feature whose size is very small.

TOSHIBA

Leading Innovation >>>