Hand Shape Recognition based on Kernel Orthogonal Mutual Subspace Method

Yasuhiro Ohkawa and Kazuhiro Fukui (University of Tsukuba)

Our Goal

Developments of a method of recognizing hand shape with high performance.

Problems and Solutions

Prob. 1: Changes of an appearance of hand : due to view points, illumination conditions, individual characteristics

⇩

We introduce **Kernel Orthogonal Mutual Subspace Method** (KOMSM [1]) to deal with the nonlinearity of distribution of multiple viewpoints images.

Prob. 2: A lot of computation of KOMSM: depending on the number of learning patterns.

⇩

We reduce the learning patterns by using k-means.

Prob.3: Difficulty of segmentation of a hand from an image:

⇩

We achieve recognition without segmentation by introducing shift-invariant feature: Higher-order Local Auto-Correlation (HLAC) feature[2].

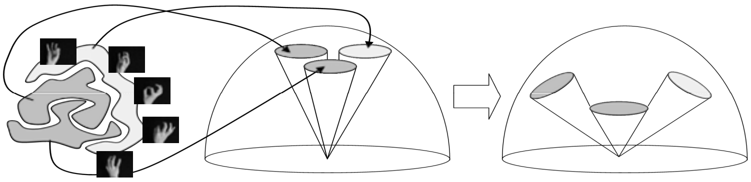
[1] K. Fukui, O. Yamaguchi: "The Kernel Orthogonal Mutual Subspace Method and its Application to 3D Object Recognition," Proc. Asian Conference on Computer Vision 2007, pp.467-476, 2007.

[2] N. Otsu and T. Kurita: "A new scheme for practical flexible and intelligent vision systems." In IAPR Workshop on Computer Vision, 1988.

Algorithm of KOMSM

STEP I:

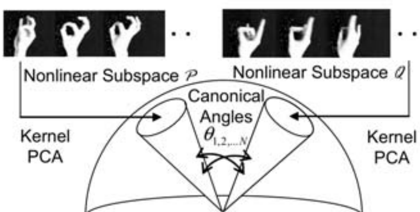
Pattern distributions of each class are represented by nonlinear subspace, which is generated by Kernel PCA.



Kernel PCA　　Orthogonalization

STEP II:

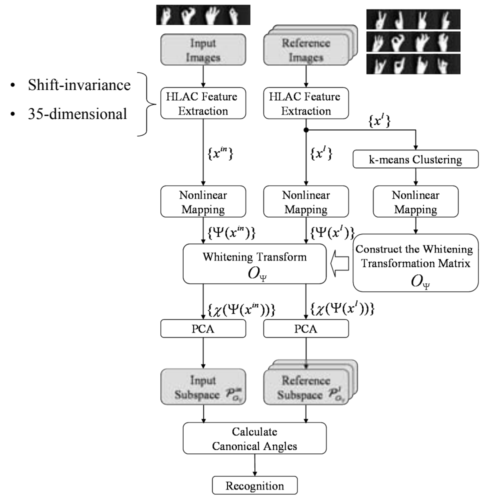
Nonlinear subspaces are orthogonalized to each other in high dimensional feature space.



STEP III:

The similarity between input subspace and reference subspaces is calculated from **canonical angles** θ. The set of input patterns is classified as the class with the highest similarity.

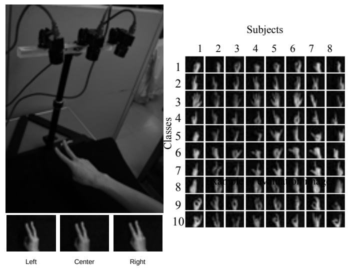
The flow of Proposed method



Evaluation Experiments and Considerations

【 Experimental condition 】

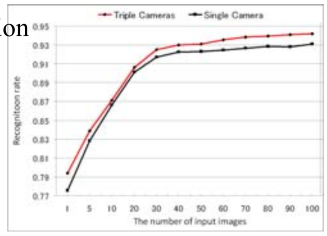
- 10 kinds of hand shapes
- Three cameras system



【 Experiment I : Validity of multiple view images 】

- 8 subjects
- Leave one-out cross validation

Example of input image set



- The performance was more improved as the number of input patterns increased and the number of cameras.
- This result indicates that various view images worked well for achieving high performance recognition.

【 Experiment II: Reduction of computing time by k-means】

- 3.0 GHz CPU

Num. of clusters	Recog.rate[%]	Recog.time[ms]
10	85.7	0.26
20	94.2	0.46
40	98.1	0.96
60	99.2	2.54
80	99.5	8.81
100	99.5	12.85
120	99.5	18.23
200	99.5	45.84

- The computing time was reduced largely while keeping high performance.
- Reducing the computing time by 95 percent when using 60 clusters.