



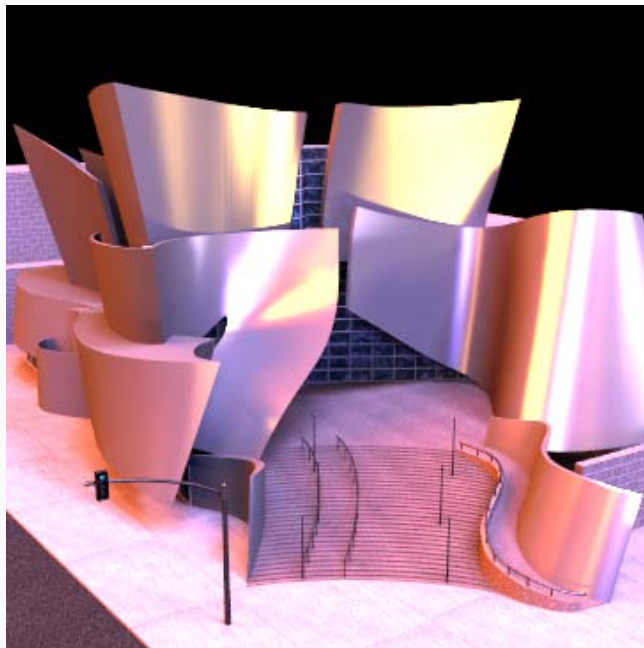
Improving Performance and Accuracy of Local PCA

V. Gassenbauer, J. Křivánek, K. Bouatouch,
C. Bouville, M. Ribardière

University of Rennes 1, France, and
Charles University, Czech Republic

Objective

Precomputed Radiance Transfer (PRT)



Bidirectional Texture Function (BTF) compression

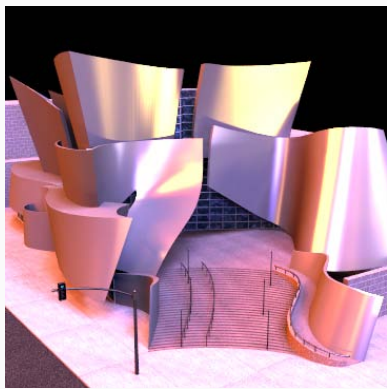


Image courtesy of Hongzhi Wu

- **Need to compress large data set**

Precomputed Radiance Transfer

- Relighting as a matrix-vector multiply



$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \vdots \\ P_4 \end{bmatrix}$$

$$=$$

$$\begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1M} \\ T_{21} & T_{22} & \cdots & T_{2M} \\ T_{31} & T_{32} & \cdots & T_{3M} \\ \vdots & \vdots & \ddots & \vdots \\ T_{N1} & T_{N2} & \cdots & T_{NM} \end{bmatrix}$$

$$\cdot \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_M \end{bmatrix}$$



- **Matrix $T(x, \omega_i)$ - Billions of elements**
 - Infeasible multiplication
 - Compression (wavelet, **Local PCA**)

pg2011

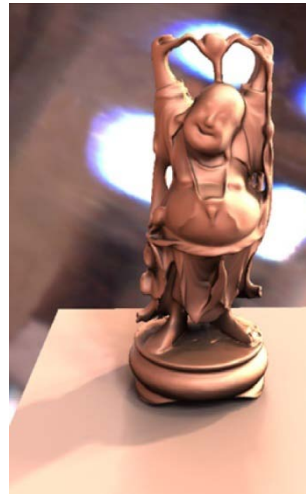
Kaohsiung, Taiwan

Related Work

- **Precomputation-based rendering**



[Sloan et al. 02,03]
Low-freq shadows,
inter-reflections



[Liu et al. 04],
[Huang et al. 10]
High-freq shadows,
inter-reflections



[Xu et al. 08]
Dynamic scene,
BRDF editing

- **Still use slow and inaccurate LPCA !**

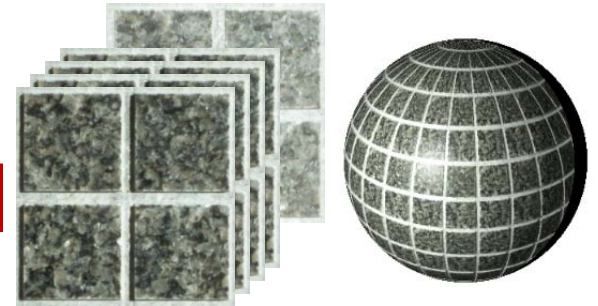
pg2011

Kaohsiung, Taiwan

Related Work

- **BTF compression**

- [Müller et al. 03], [Filip et al. 09]



- **Simpler LPCA: k -means problem**

- Better performance

[Phillips 02], [Elkan 03], [Hamerly10]

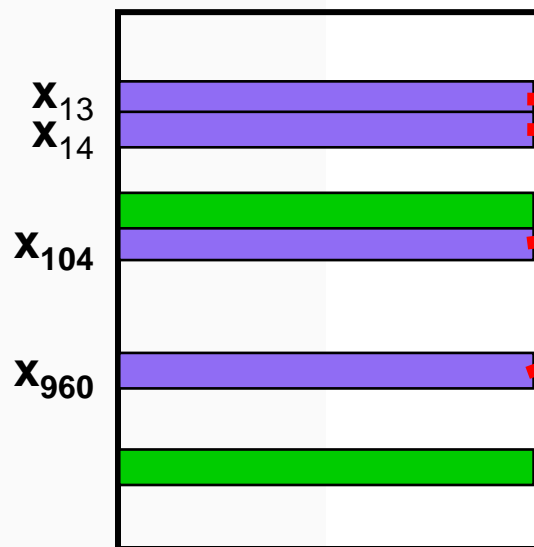
- Better accuracy

[Arthur et al. 07], [Kanugo et al. 02]

Compression Problem

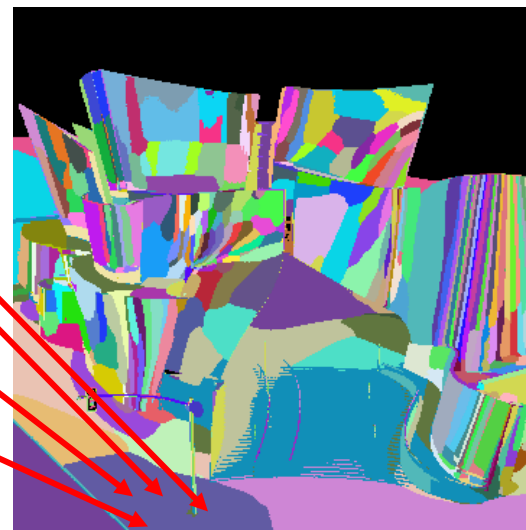
• Input

- High-dimensional points (rows of \mathbf{T})
- Number of clusters k



• Output

- Find k clusters – i.e. low-dimensional subspaces

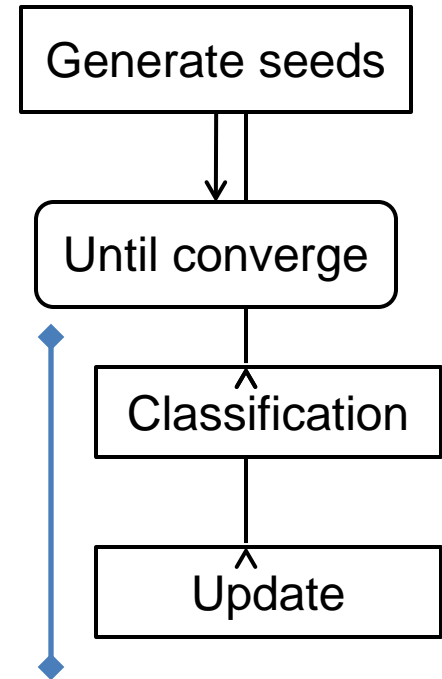


pg2011

Kaohsiung, Taiwan

LPCA – The Algorithm

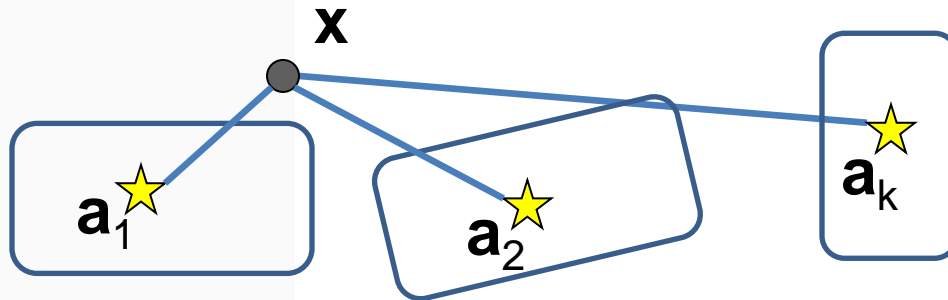
- **Guess k clusters (i.e. subs.)**
 - Initialize by randomly selected centers
- **Repeat until convergence**
 - Assign each point to the nearest cluster
 - Update PCA in the cluster



Motivation

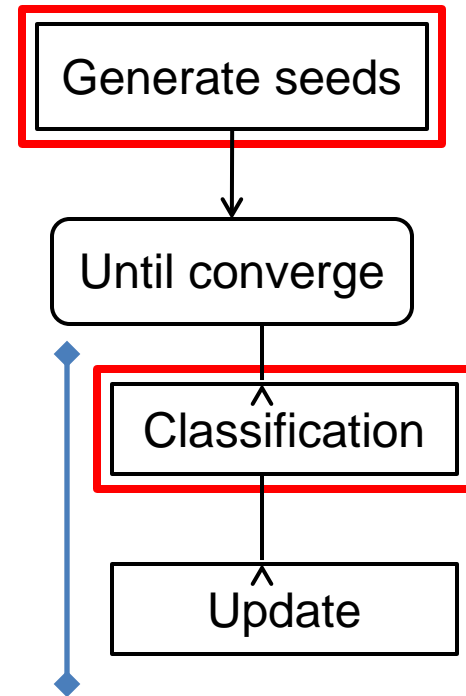
- **Inefficient**

- For each point compute distance to all clusters a_i



- **Inaccurate**

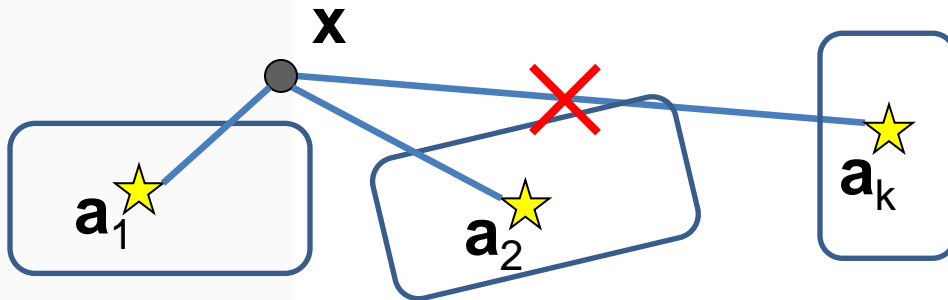
- Prone to get stuck in a local optimum



Our Contribution

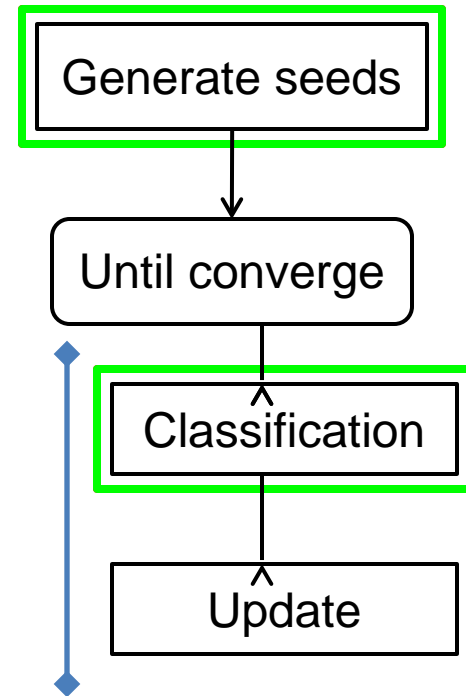
- ~~Inefficient~~

- ~~Sort each point compute distance to all clusters a_i~~



- ~~Inaccurate~~

- ~~Prono to get stuck in a local optimum~~

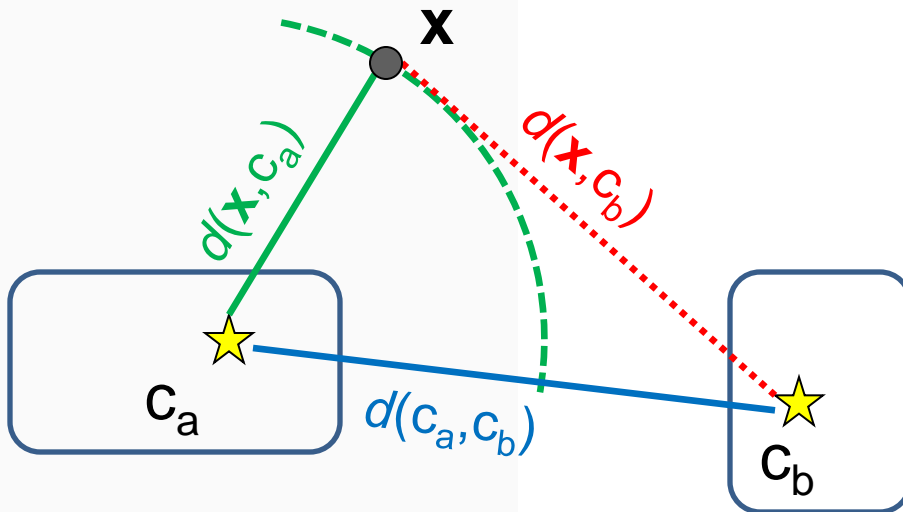


Outline

- **Improving Performance**
 - SortCluster-LPCA (SC-LPCA)
- **Improving Accuracy**
 - SortMeans++
- **Results**

Accelerated k -means

- Δ -inequality [Phillips 02]



If $d(c_a, c_b) \geq 2d(x, c_a)$

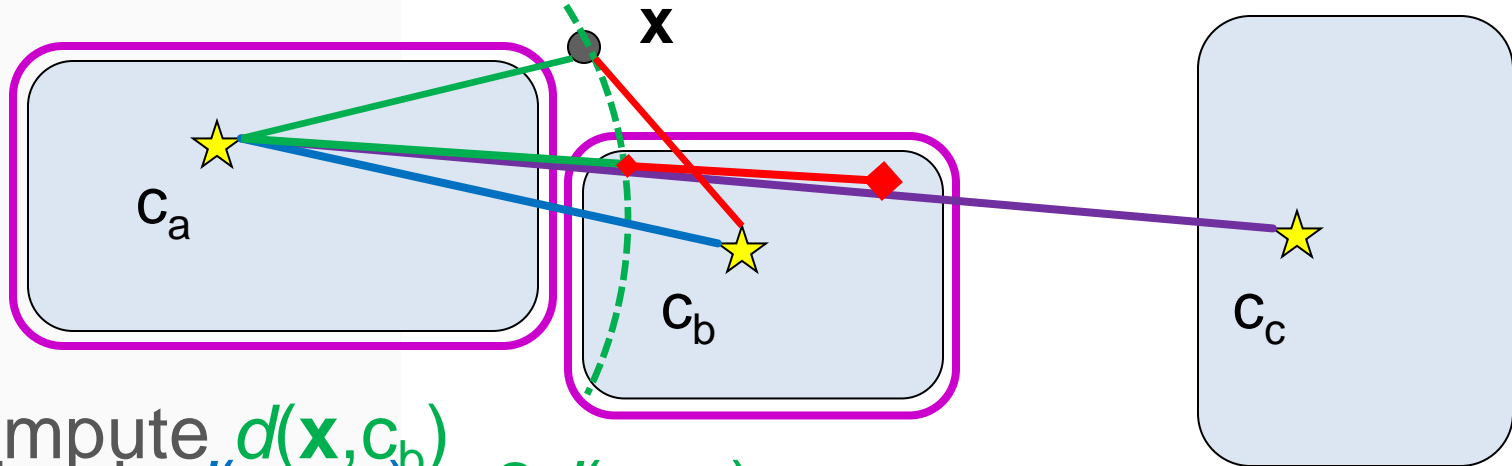
→ $d(x, c_b) \geq d(x, c_a)$

→ $d(x, c_b)$ not necessary
to compute !

How does it work ?

We know: potentially nearest cluster to \mathbf{x}

We know: Distances of other cluster w.r.t. C_a



Compute $d(\mathbf{x}, C_b)$
 Check $d(C_a, C_b) \geq 2d(\mathbf{x}, C_a)$
 Compute $d(\mathbf{x}, C_a)$
 C_b becomes a new nearest cluster
 Check $d(C_a, C_c) \geq d(\mathbf{x}, C_a) + d(\mathbf{x}, C_b)$

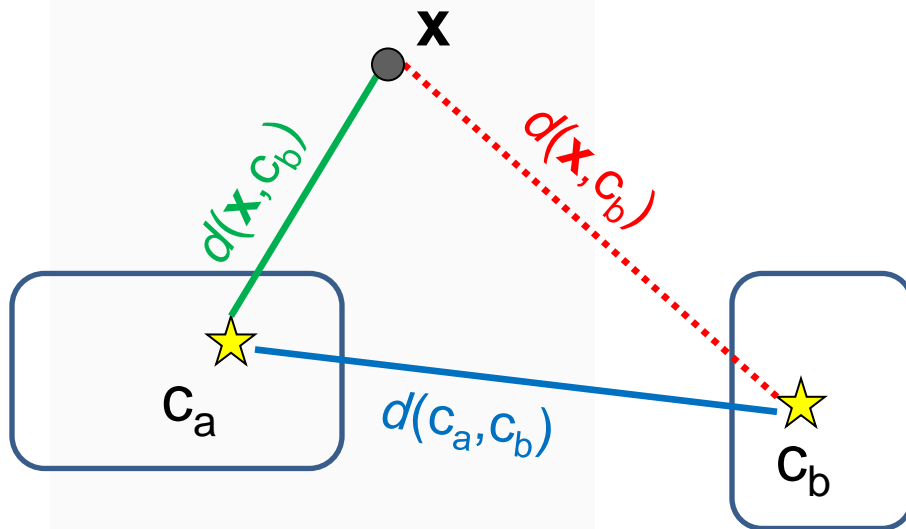
pg2011

Kaohsiung, Taiwan

Our Contribution: From k -means to

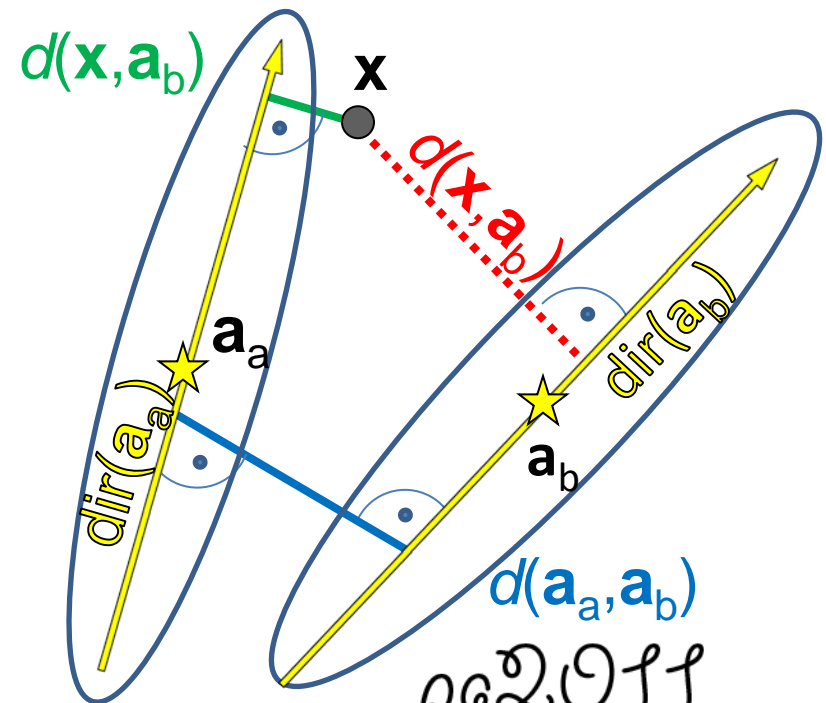
k -means

Piecewise
reconstruction



LPCA

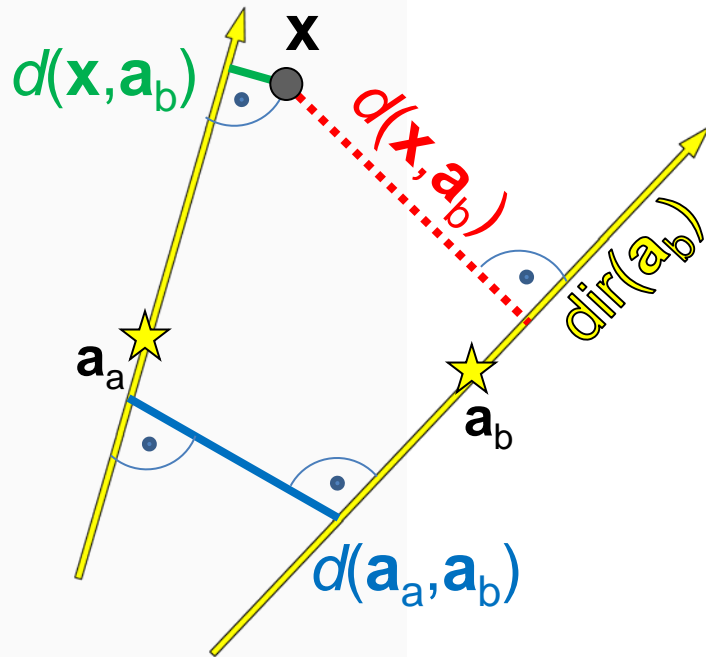
Piecewise reconstruction in
low-dimensional subspaces



▶ Δ -inequality for LPCA

- Distance between subspaces

- $d(a_a, a_b) = \inf\{ \|p - q\| ; p \text{ in } a_a, q \text{ in } a_b \}$



- Δ -inequality for subspaces

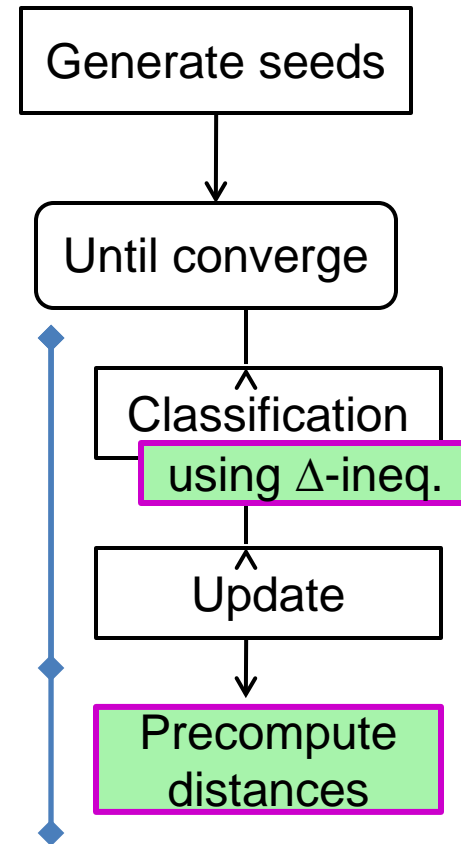
If $d(a_a, a_b) \geq 2d(x, a_b)$

→ $d(x, a_a) \geq d(x, a_b)$

→ $d(x, a_b)$ not necessary to compute !

Our SortCluster-LPCA

- **When assigning x**
 - Start from a_j
 - Proceed a_i in increasing order of distances w.r.t. a_j
 - Check Δ -inequality
- **For all a_i precompute**
 - Distances to each others
 - Ordering

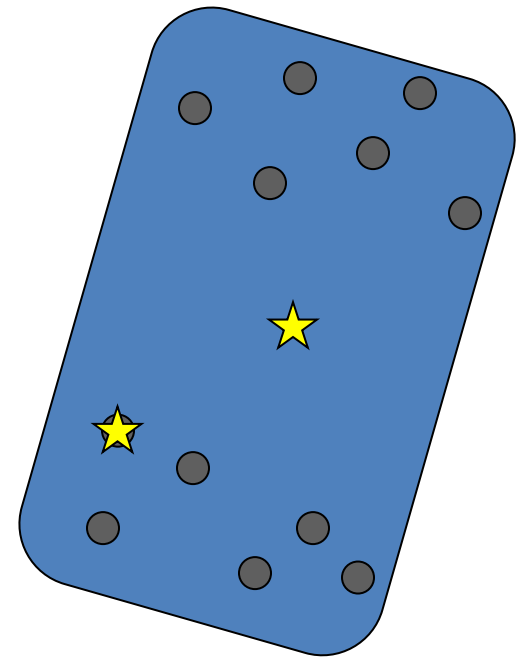
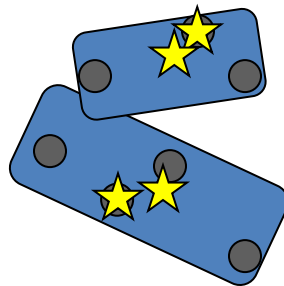


Outline

- **Improving Performance**
 - SortCluster-LPCA (SC-LPCA)
- **Improving Accuracy**
 - SortMeans++
- **Results**

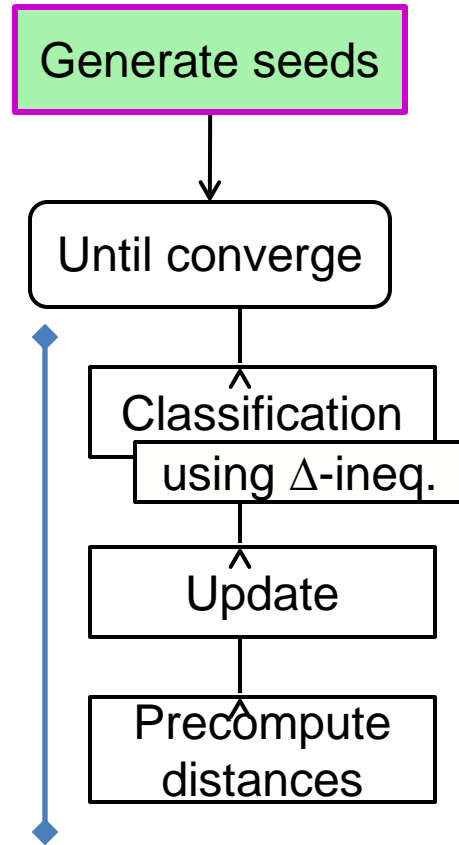
LPCA Accuracy

- **LPCA prone to stuck in local optimum**
- **Observation**
 - Error comes from poor selections of cluster centers



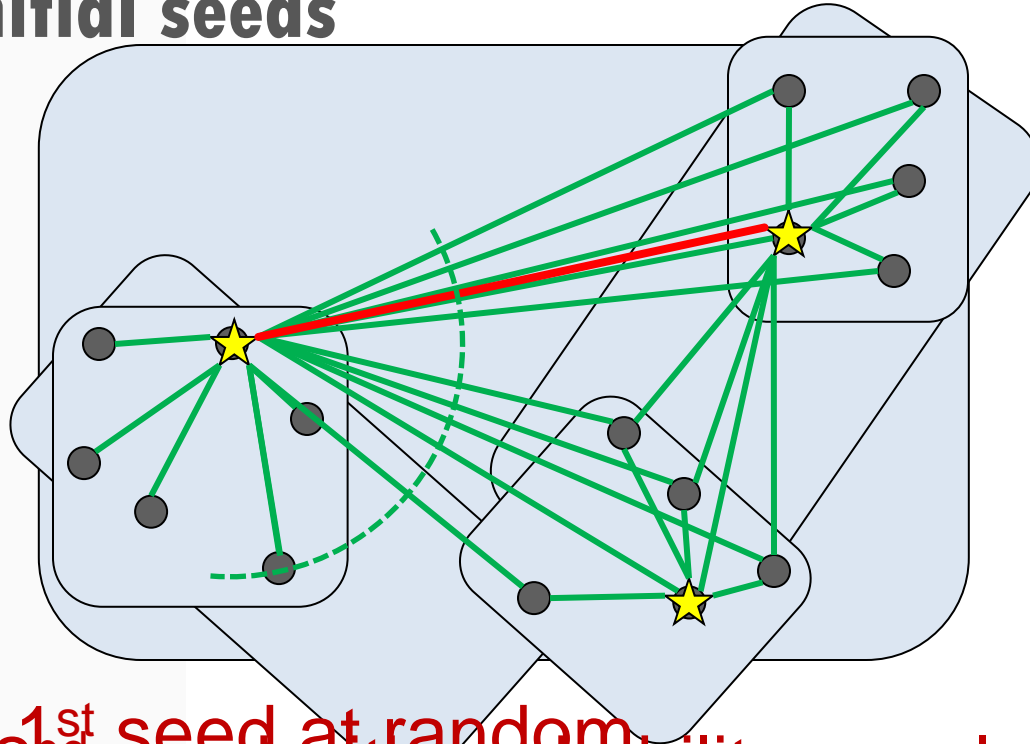
Generation of Seeds

- **Some heuristic**
 - Farthest first [Hochbaum et al. 85]
 - Sum based [Hašan et al. 06]
 - *k*-means++ [Arthur and Vassil. 07]
 - ...
- ***Our approach: SortMeans++***
 - Based on *k*-means++
 - Faster



Our SortMeans++

- **Select initial seeds**



Select 1st seed at random
 Take 2nd seed with probability equal distances
 Recluster using A inequality
 Take 3rd seed with probability equal distances
 Recluster ...

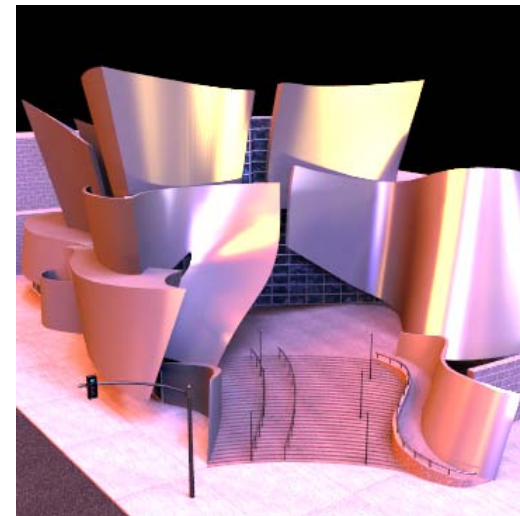
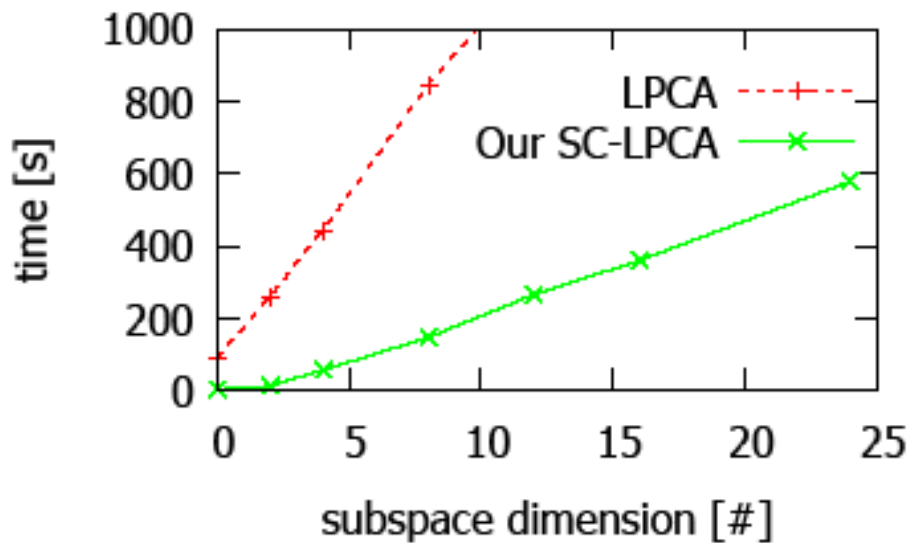
pg2011

Outline

- **Improving Performance**
 - SortCluster-LPCA (SC-LPCA)
- **Improving Accuracy**
 - SortMeans++
- **Results**

Results

- Evaluate method with different parameters
 - Scalability with the subspace dimension

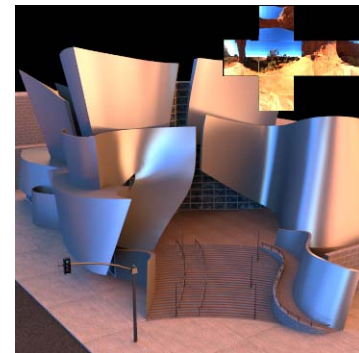
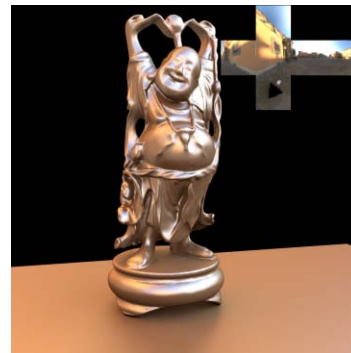
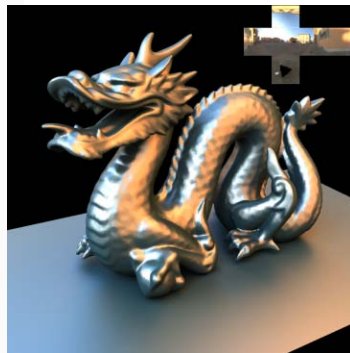
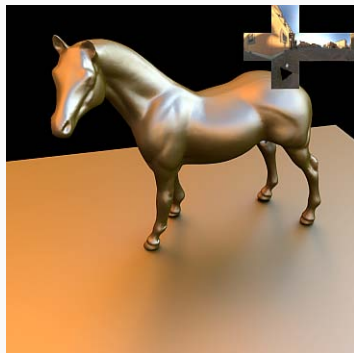


- More than 5x speed-up

Overall Performance

- Did several iterations of (SC)LPCA up to 24 basis

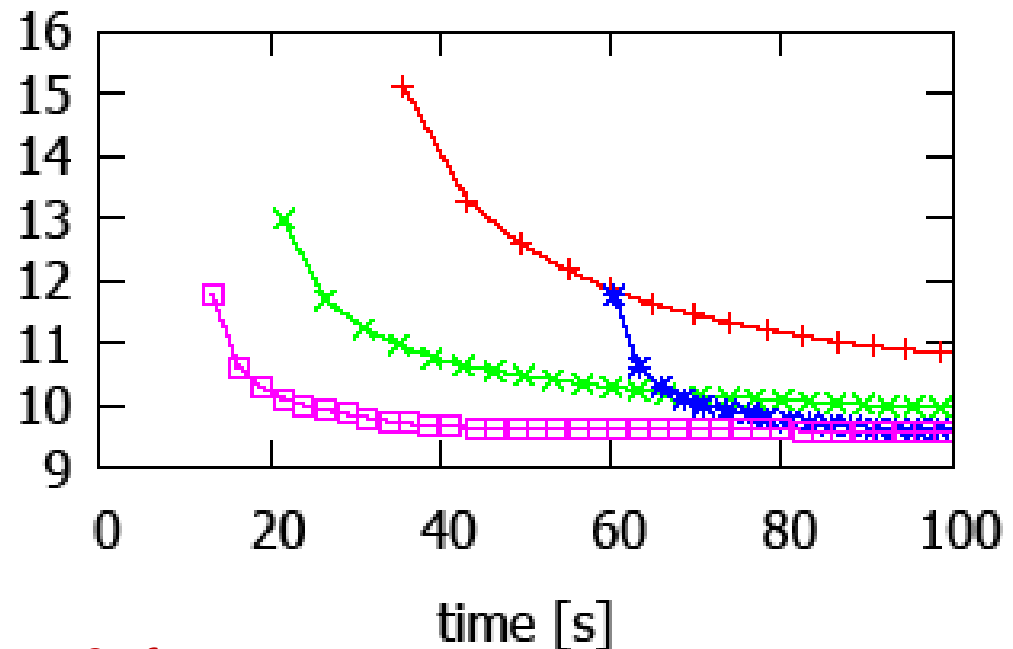
Model	Vertices [#]	LPCA	SC-LPCA	Speed Up	PRT
Horse	67.6k	3h 48m	11m	20.3x	22.5 s
Dragon	57.5k	3h 1m	28m	6.4x	25.5 s
Buddha	85.2k	4h 38m	50m	5.6x	31.6 s
Disney	106.3k	5h 50m	1h 8m	5.1x	46.5 s



Improving Accuracy

- **Latency and accuracy of seeding algs:**

random —+—
 sums-based —x—
 kmeans++ —*—
 SortMeans++ —□—

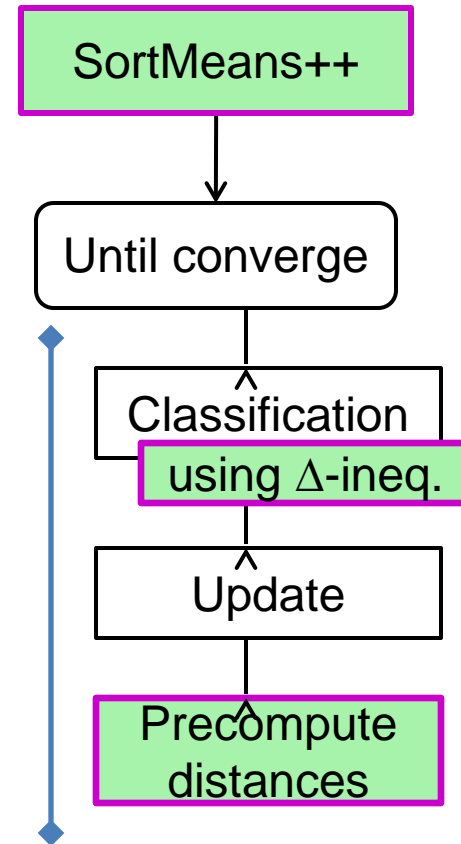


- **Our SortMeans++**

- Generally lowest error & fast
- Improve performance of SC-LPCA !

Conclusion

- **SC-LPCA (accelerated LPCA)**
 - Avoid unnecessary distance comp.
 - Speed-up of 5 to 20 on our PRT data
 - Without changing output !
- **Improve accuracy (SortMeans++)**
 - More accurate data approximation
- **Future work**
 - Test on other CG data
 - GPU acceleration

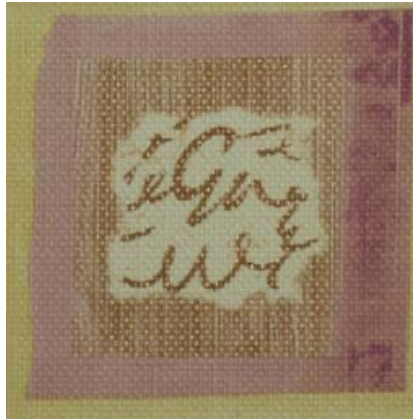
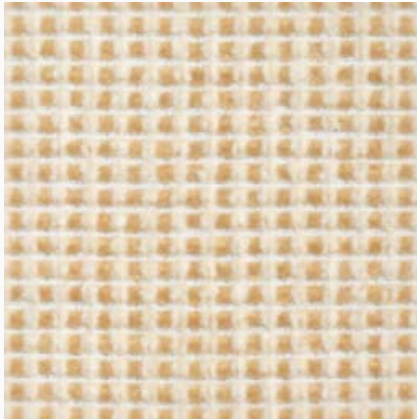


pg2011

Kaohsiung, Taiwan

BTF Compression

- Try several data sets



- **SC-LPCA speed-up of about only 1.5x**
 - Reason: Small number of subspaces

The End

- **Acknowledgement**
 - **European Community**
 - **Marie Curie Fellowship PEOF-GA-2008-221716**
 - **Ministry of Education, Czech Republic**
 - **Research program LC-06008.**
 - **Anonymous reviewers**

Thank You for your attention