An Eigenspace Update Algorithm for Image Analysis

B. S. Manjunath†, S. Chandrasekaran†,
and Y. F. Wang‡

‡ Department of Computer Science
† Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106
manj@ece.ucsb.edu, shiv@ece.ucsb.edu,
yfwang@cs.ucsb.edu

Outline

- Introduction
- Mathematical Framework
- Experimental Results
- Conclusion

Introduction

- Eigen space representations
  - An *object*-centered basis expansion
  - A compact representation of the input image ensemble
  - Image space or feature (e.g., wavelet) space
  - Applications:
    * face recognition (Sirovich & Kirby, 1982, Turk & Pentland, 1991)
    * multi-spectrum image and video coding (Murakami & Kumar, 1982, Murase & Lindenbaum, 1995)
    * pose estimation (Murase & Nayar, 1993)
  - Deterministic: SVD
  - Statistic: Karhunen-Loeve transform (KLT)

Accomplishments

- A comparative study of SVD/KLT techniques
- Introducing an SVD update algorithm (Gu & Eisenstat, 1994)
  - Numerical stability
  - Computational efficiency
  - Adaptive refinement

Mathematical Framework

A_i: row-scanned column image ($m \times 1$ vector)
B_i: $[A_1 A_2 \cdots A_i]$ (image ensemble matrix)
k: ϵ-rank of B_i (number of singular values of $B_i > \epsilon$)

We have:

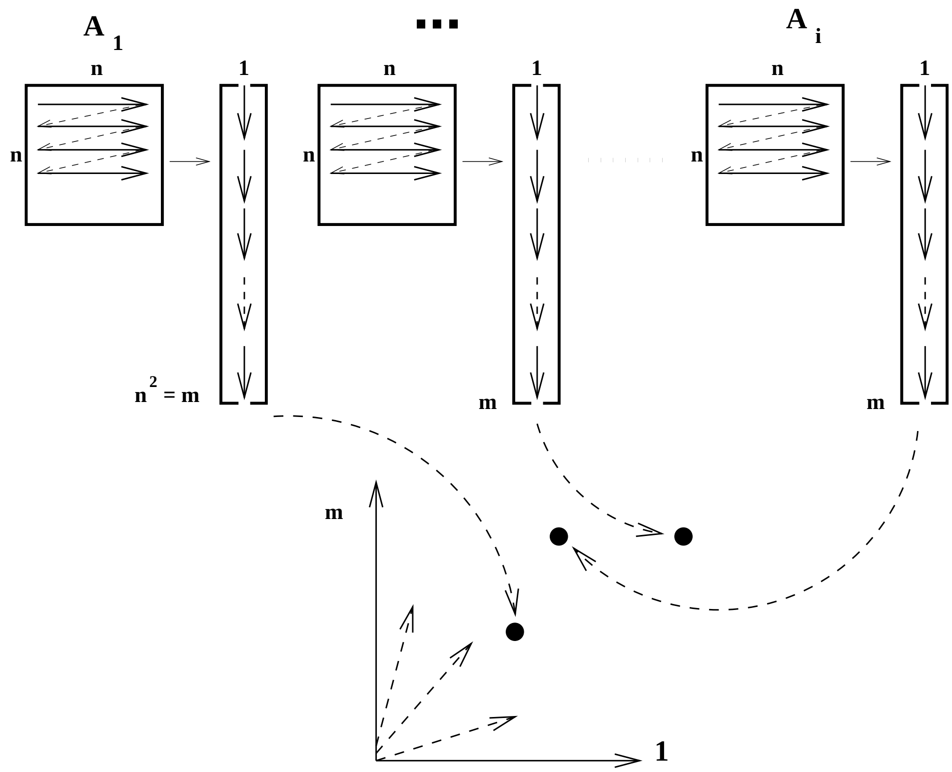
$$B_i \approx [U_i \;\; *] \begin{bmatrix} \Sigma_i & 0 \\ 0 & [\epsilon] \end{bmatrix} \begin{bmatrix} V_i^T \\ * \end{bmatrix}$$

or

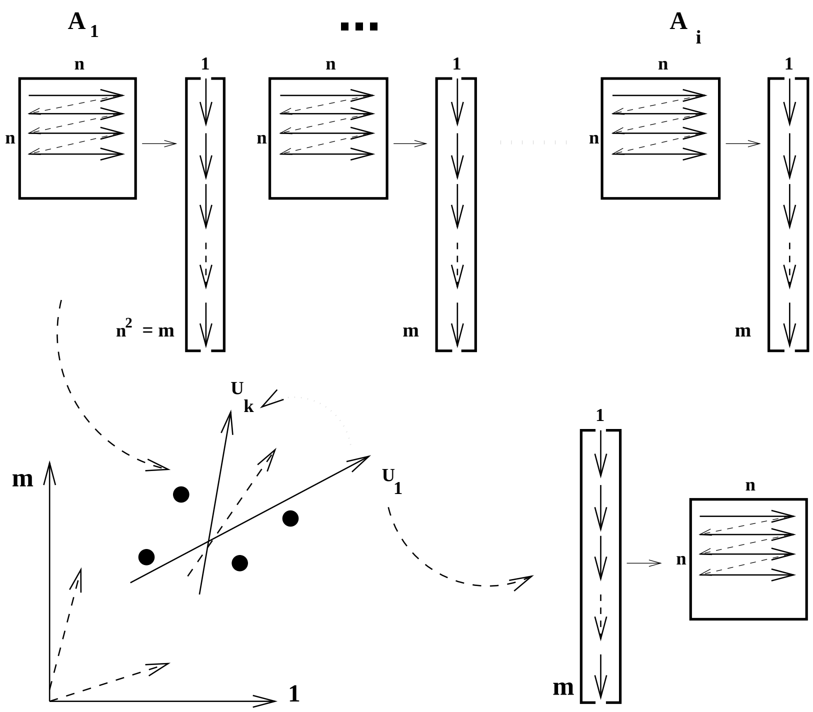
$$||B_i - U_i \Sigma_i V_i^T|| \leq \epsilon$$

where

U_i: $m \times k$ matrix
Σ_i: $k \times k$ matrix
V_i: $i \times k$ matrix



The ε-rank is usually much lower than i



Find a *compact* set of basis vectors (images) to represent the image ensemble

$$A_j = U_i \Sigma_i V_{i_j}^T$$

V_{i_j}: the j-row of V_i

Computational Requirements

- Compute $\{U_i, \Sigma_i, V_i\}$ efficiently
- Alternatives:
  - Standard SVD ($O(mi^2)$)
  - $B_i^T B_i$ ($O(mi^2)$, numerical stability)
  - Adaptive $B_i^T B_i$ ($O(mki)$, numerical stability)
  - Adaptive B_i ($O(mki)$)

Comparison of Algorithms

Authors	Method	Update
Murakami and Kumar (1982)	$B^T B$	yes
Kirby and Sirkovich (1990)	BB^T	no
Turk and Pentland (1991)	$B^T B$	no
Murase and Lindenbaum (1995)	iterative (BB^T)	no
This paper	GES (1994)	yes

Summary of

Adaptive SVD Algorithm

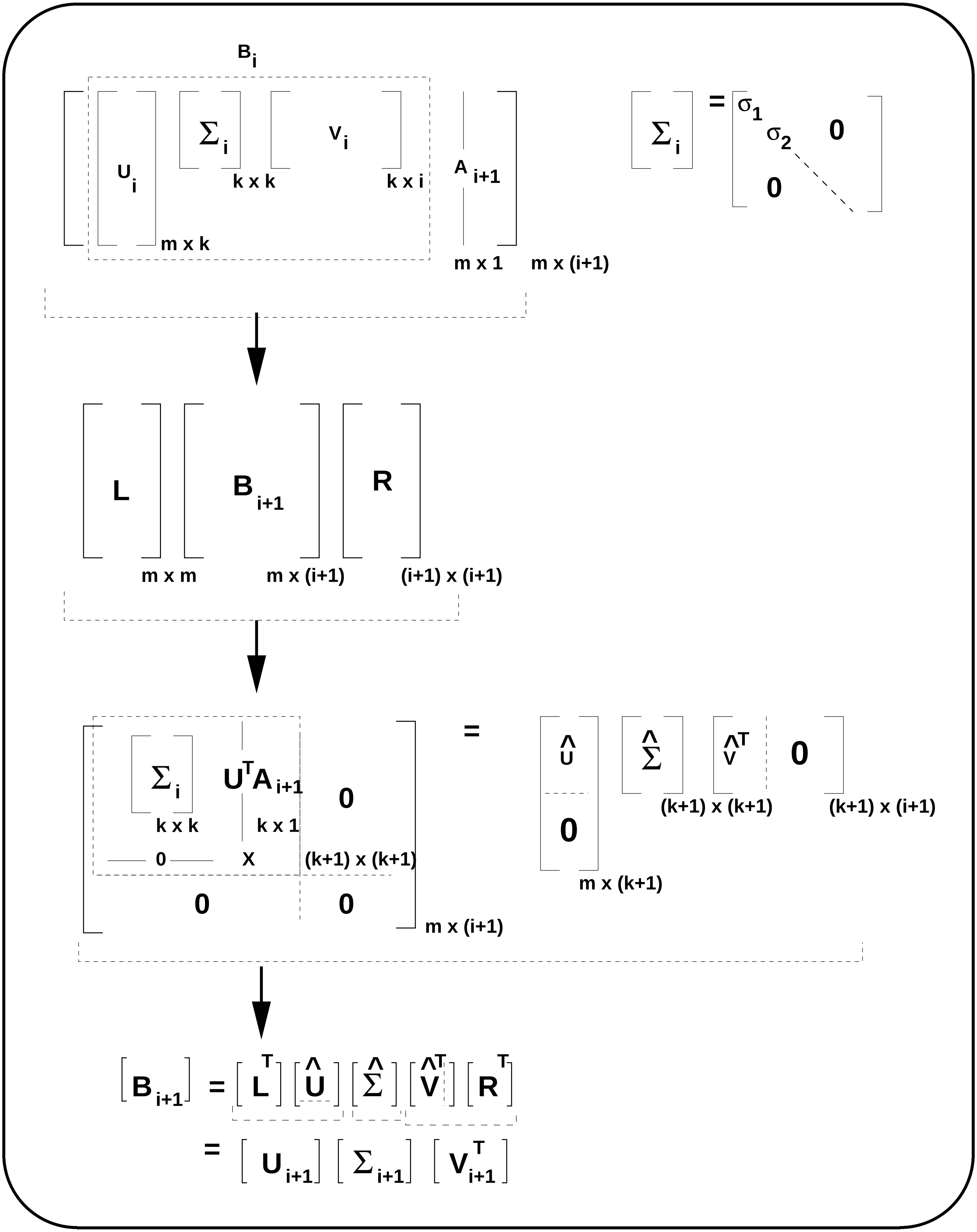
Given, at step i:

$$[B_i]_{m \times i} = [U_i]_{m \times k} \ \ [\Sigma_i]_{k \times k} \ \ [V_i^T]_{k \times i}$$

Compute *incrementally*, at step $i+1$:

$$[B_i A_{i+1}]_{m \times (i+1)} =$$

$$[U_{i+1}]_{m \times (k+1)} [\Sigma_{i+1}]_{(k+1) \times (k+1)} [V_{i+1}^T]_{(k+1) \times (i+1)}$$


B i
U i
m x k
Σ i
k x k
V i
k x i
A i+1
m x 1
m x (i+1)
Σ i
= σ 1 σ 2 0
0
L
B i+1
R
m x m
m x (i+1)
(i+1) x (i+1)
Σ i
k x k
U T A i+1
k x 1
0
0
X
(k+1) x (k+1)
0
0
m x (i+1)
=
^ U
0
m x (k+1)
^ Σ
(k+1) x (k+1)
^ V T
0
(k+1) x (i+1)
B i+1 = L T ^ U ^ Σ ^ V T R T
= U i+1 Σ i+1 V T i+1

Major Steps in Adaptive SVD

- $O(mk)$

$$\hat{A}_{i+1} = \frac{(I - U_i U_i^T) A_{i+1}}{||(I - U_i U_i^T) A_{i+1}||}$$

- $O(m)$

$$X = \hat{A}_{i+1}^T A_{i+1}$$

- SVD of

$$\begin{bmatrix} \Sigma_i & U_i^T A_{i+1} \\ 0 & X \end{bmatrix}$$

  - Brute force $O(12k^3)$
  - Clever use of broken arrowhead matrix structure $O(6k^3)$
  - GES $O(100k^2)$

Major Steps in Adaptive SVD (cont.)

- $O(mk)$ (with GES)

$$U_{i+1} = L^T \hat{U} = [U_i \;\; \hat{A}_{i+1}]\, \hat{U}$$

-

$$\Sigma_{i+1} = \hat{\Sigma}$$

- $O(mk)$ (with GES)

$$V_{i+1} = R\hat{V} = \begin{bmatrix} V_i & 0 \\ 0 & 1 \end{bmatrix} \hat{V}$$

Adaptive Eigenspace

Update Algorithm

$U = A_1/||A_1||, \;\; V = 1, \;\; \Sigma = ||A_1||$

For $i = 2$ to N

$\quad \left[U\Sigma V^T \;\; A_i\right] = U'\Sigma'V'^T$

$\quad$ Find k such that $\sigma'_k > \delta \geq \sigma'_{k+1}$

$\quad$ Let U be the first k columns of U'

$\quad$ Let V be the first k columns of V'

$\quad$ Let Σ be the leading $k \times k$ principal submatrix of Σ

End

δ: approximation accuracy

ϵ/N (number of images)

Experimental Results

- Demonstrate visual learning to select meaningful views
- Procedures:
  - Place test objects on a rotation stage
  - One picture every 10^o of rotation
  - Iteratively:
    - Estimate reconstruction error using current basis set
    - If error is large, update the basis set with the current image

Conclusions

- Adaptive update improves efficiency
- SVD improves numerical stability
- Possible applications:
  - Active exploration
  - Pose alignment
  - Browsing of large image databases