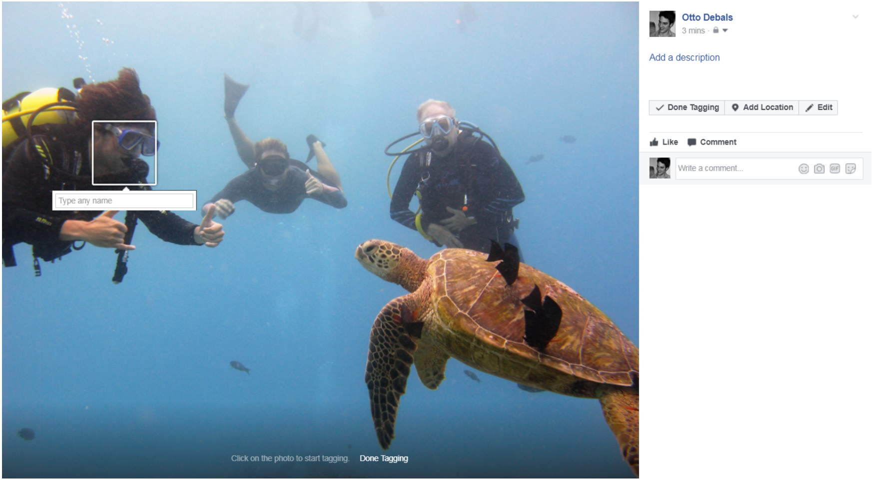
Numerical optimization algorithms for tensor-based face recognition

Otto Debals, Martijn Boussé, Nico Vervliet and Lieven De Lathauwer

How does Facebook suggest tags?



Face recognition . . .

different persons

Face recognition is a multidimensional problem

different persons

different illumination

Face recognition is a multidimensional problem

different persons

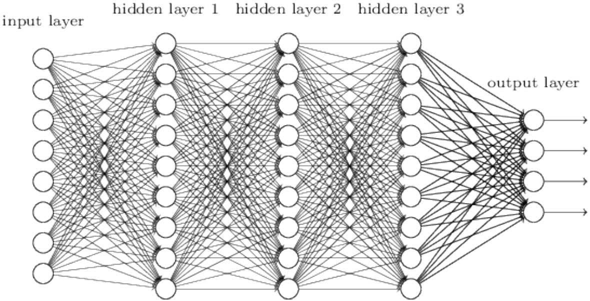
different illumination

and different pose, expression, ...

Face recognition is a multidimensional problem

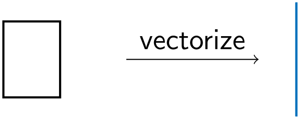
Linear algebra is limited to single-mode variations

Of course, nonlinear models such as neural nets can be used



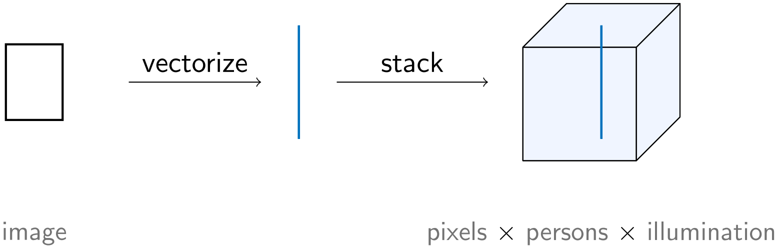
Tensors can explicitly accommodate for the multidimensional nature of facial images

Tensors can explicitly accommodate for the multidimensional nature of facial images



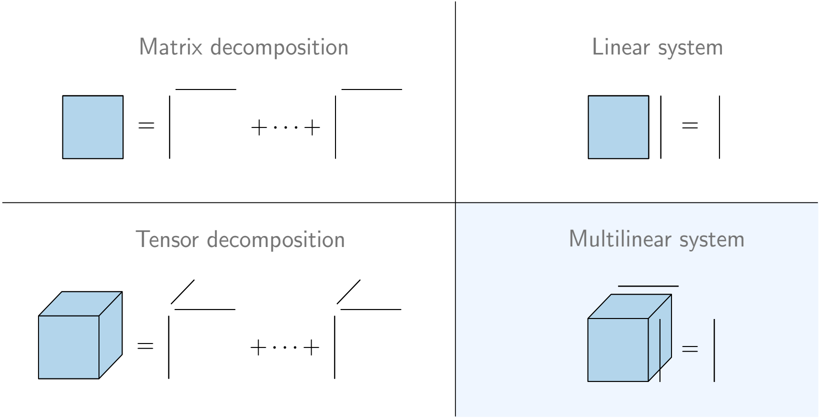
image

Tensors can explicitly accommodate for the multidimensional nature of facial images



Also, tensor tools are highly interpretable

Also, tensor tools are highly interpretable



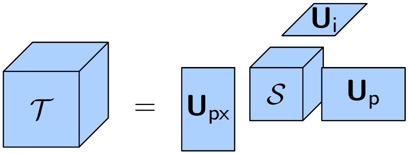
Tensor-based face recognition:

1. Construct bases using multilinear SVD (MLSVD)

Tensor-based face recognition:

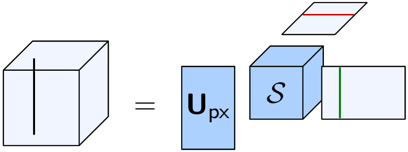
1. Construct bases using multilinear SVD (MLSVD)
2. Recognize a person using multilinear systems or Kronecker product equations (KPE)

We compute a multilinear SVD to approximately separate the influences of the different modes



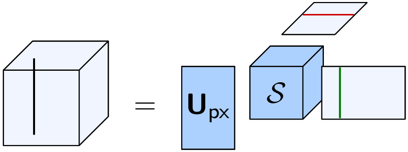
$$\mathcal{T} \approx \mathcal{S} \cdot_1 \mathbf{U}_{\mathrm{px}} \cdot_2 \mathbf{U}_{\mathrm{p}} \cdot_3 \mathbf{U}_{\mathrm{i}}$$

Every (vectorized) facial image can then be expressed as follows



$$\mathbf{d} = (\mathcal{S} \cdot_1 \mathbf{U}_{px}) \cdot_2 \mathbf{c}_p^T \cdot_3 \mathbf{c}_i^T$$

Every (vectorized) facial image can then be expressed as follows



$$\mathbf{d} = (\mathcal{S} \bullet_1 \mathbf{U}_{px}) \bullet_2 \mathbf{c}_p^T \bullet_3 \mathbf{c}_i^T$$

$$\updownarrow$$

$$\mathbf{d} = (\mathbf{U}_{px}\mathbf{S}_{(1)})(\mathbf{c}_i \otimes \mathbf{c}_p)$$

Classification of a new image reduces to a KPE

Classification of a new image reduces to a KPE

Tensorize (labeled) dataset into $\mathcal{T}$

Classification of a new image reduces to a KPE

Tensorize (labeled) dataset into $\mathcal{T}$

Compute MLSVD of $\mathcal{T} = \mathcal{S} \cdot_1 \mathbf{U}_{\mathsf{px}} \cdot_2 \mathbf{U}_{\mathsf{p}} \cdot_3 \mathbf{U}_{\mathsf{i}}$

Classification of a new image reduces to a KPE

Tensorize (labeled) dataset into $\mathcal{T}$

Compute MLSVD of $\mathcal{T} = \mathcal{S} \cdot_1 \mathbf{U}_{\mathsf{px}} \cdot_2 \mathbf{U}_{\mathsf{p}} \cdot_3 \mathbf{U}_{\mathsf{i}}$

Every image $\mathbf{d}$ admits $\mathbf{d} = (\mathbf{U}_{\mathsf{px}} \mathbf{S}_{(1)})(\mathbf{c}_{\mathsf{i}} \otimes \mathbf{c}_{\mathsf{p}})$

Classification of a new image reduces to a KPE

Tensorize (labeled) dataset into $\mathcal{T}$

Compute MLSVD of $\mathcal{T} = \mathcal{S} \cdot_1 \mathbf{U}_{\text{px}} \cdot_2 \mathbf{U}_{\text{p}} \cdot_3 \mathbf{U}_{\text{i}}$

Every image $\mathbf{d}$ admits $\mathbf{d} = (\mathbf{U}_{\text{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\text{i}} \otimes \mathbf{c}_{\text{p}})$

Hence, for a new image $\mathbf{d}^{(\text{new})}$ we solve

$$\mathbf{d}^{(\text{new})} = (\mathbf{U}_{\text{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\text{i}}^{(\text{new})} \otimes \mathbf{c}_{\text{p}}^{(\text{new})})$$

Classification of a new image reduces to a KPE

Tensorize (labeled) dataset into $\mathcal{T}$

Compute MLSVD of $\mathcal{T} = \mathcal{S} \cdot_1 \mathbf{U}_{\mathsf{px}} \cdot_2 \mathbf{U}_{\mathsf{p}} \cdot_3 \mathbf{U}_{\mathsf{i}}$

Every image $\mathbf{d}$ admits $\mathbf{d} = (\mathbf{U}_{\mathsf{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\mathsf{i}} \otimes \mathbf{c}_{\mathsf{p}})$

Hence, for a new image $\mathbf{d}^{(\mathsf{new})}$ we solve

$$\mathbf{d}^{(\mathsf{new})} = (\mathbf{U}_{\mathsf{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\mathsf{i}}^{(\mathsf{new})} \otimes \mathbf{c}_{\mathsf{p}}^{(\mathsf{new})})$$

Classification of a new image reduces to a KPE

Tensorize (labeled) dataset into $\mathcal{T}$

Compute MLSVD of $\mathcal{T} = \mathcal{S} \cdot_1 \mathbf{U}_{\mathrm{px}} \cdot_2 \mathbf{U}_{\mathrm{p}} \cdot_3 \mathbf{U}_{\mathrm{i}}$

Every image $\mathbf{d}$ admits $\mathbf{d} = (\mathbf{U}_{\mathrm{px}} \mathbf{S}_{(1)})(\mathbf{c}_{\mathrm{i}} \otimes \mathbf{c}_{\mathrm{p}})$

Hence, for a new image $\mathbf{d}^{(\mathrm{new})}$ we solve

$$\mathbf{d}^{(\mathrm{new})} = (\mathbf{U}_{\mathrm{px}} \mathbf{S}_{(1)})(\mathbf{c}_{\mathrm{i}}^{(\mathrm{new})} \otimes \mathbf{c}_{\mathrm{p}}^{(\mathrm{new})})$$

and we compare the estimate of $\mathbf{c}_{\mathrm{p}}^{(\mathrm{new})}$ with the rows of $\mathbf{U}_{\mathrm{p}}$ to find the closest match and use the corresponding label.

What about multiple images of the same person?

$$\mathbf{d} = (\mathbf{U}_{\mathrm{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\mathrm{i}} \otimes \mathbf{c}_{\mathrm{p}})$$

What about multiple images of the same person?

$$\mathbf{d} = (\mathbf{U}_{\mathrm{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\mathrm{i}} \otimes \mathbf{c}_{\mathrm{p}})$$

$$\Downarrow$$

$$\mathbf{d}^{(\mathrm{q})} = (\mathbf{U}_{\mathrm{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\mathrm{i}} \otimes \mathbf{c}_{\mathrm{p}}^{(\mathrm{q})})$$

What about multiple images of the same person?

$$\mathbf{d} = (\mathbf{U}_{\text{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\text{i}} \otimes \mathbf{c}_{\text{p}})$$

$$\Downarrow$$

$$\mathbf{d}^{(q)} = (\mathbf{U}_{\text{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\text{i}} \otimes \mathbf{c}_{\text{p}}^{(q)})$$

The latter, which is a coupled KPE, can be written as

$$\mathbf{D} = (\mathbf{U}_{\text{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\text{i}} \otimes \mathbf{C}_{\text{p}})$$

Illustration on a real-life dataset

We use the extended Yale B dataset

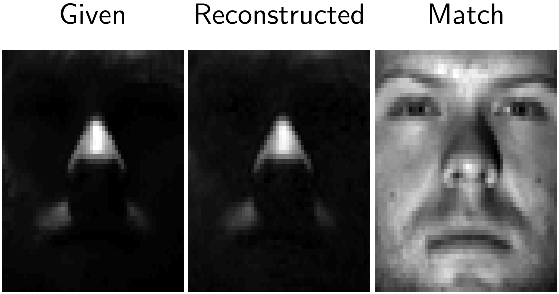
51×58 image
37 persons
64 illuminations

We use the extended Yale B dataset

51×58 image
37 persons
64 illuminations

Of course, we use Tensorlab for all computations in MATLAB

We correctly classify a *known* person even though the image is almost completely dark



The KPE-based approach has a very competitive performance

Table: Performance (%) and recognition time (s) in comparison to conventional techniques such as 'EigenFaces' [Turk and Pentland, 1991] as well as another tensor-based approach 'TensorFaces' [Vasilescu and Terzopoulos, 2002]

	'Eigenfaces'	'Tensorfaces'	KPE
Accuracy	93.3	93.5	**95.7**
Precision	90.6	94.4	**96.6**
Recall	88.4	90.9	**95.8**
Time of PCA/MLSVD	**2.97**	3.29	3.29
Time of recognition	**0.004**	0.135	0.097

Higher performance can be achieved using multiple images under different illuminations

	'Eigenfaces'			cKPE-based method		
# illuminations	1	2	3	1	2	3
Accuracy	92.7	93.3	96.3	**95.8**	**97.1**	**97.3**
Precision	89.8	91.2	97.9	**97.0**	**99.3**	**99.9**
Recall	87.7	87.8	97.5	**96.2**	**99.2**	**99.9**

What about results for someone who is not in the database?

What about results for someone who is not in the database?

- To 'register' a person $\mathbf{d}^{(!)}$ not in the database we solve

$$\mathbf{d}^{(!)} = (\mathbf{U}_{\mathsf{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\mathsf{i}}^{(!)} \otimes \mathbf{c}_{\mathsf{p}}^{(!)})$$

and we add $\mathbf{c}_{\mathsf{p}}^{(!)}$ to the 'database' $\mathbf{U}_{\mathsf{p}}$

What about results for someone who is not in the database?

- To 'register' a person $\mathbf{d}^{(!)}$ not in the database we solve

$$\mathbf{d}^{(!)} = (\mathbf{U}_{\mathsf{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\mathsf{i}}^{(!)} \otimes \mathbf{c}_{\mathsf{p}}^{(!)})$$

  and we add $\mathbf{c}_{\mathsf{p}}^{(!)}$ to the 'database' $\mathbf{U}_{\mathsf{p}}$

- New image $\mathbf{d}^{(?)}$ under different illumination, we solve

$$\mathbf{d}^{(?)} = (\mathbf{U}_{\mathsf{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\mathsf{i}}^{(?)} \otimes \mathbf{c}_{\mathsf{p}}^{(?)})$$

  and compare $\mathbf{c}_{\mathsf{p}}^{(?)}$ to $\mathbf{U}_{\mathsf{p}}$

The MLSVD model generalizes quite well for a *new* person using only one image with a neutral illumination

Given Reconstructed

We correctly classify the *new* person using a different illumination

Given Reconstructed Best match Second match

Again, using images with multiple illuminations can improve the results

Person	One illumination	Two illuminations	Three illuminations
16	51.8	56.4	**59.3**
25	64.3	72.7	**75.9**
28	58.9	63.6	**70.4**
All	61.8	66.2	**68.1**

Kronecker Product Equations are a powerful tool for many problems, including face recognition

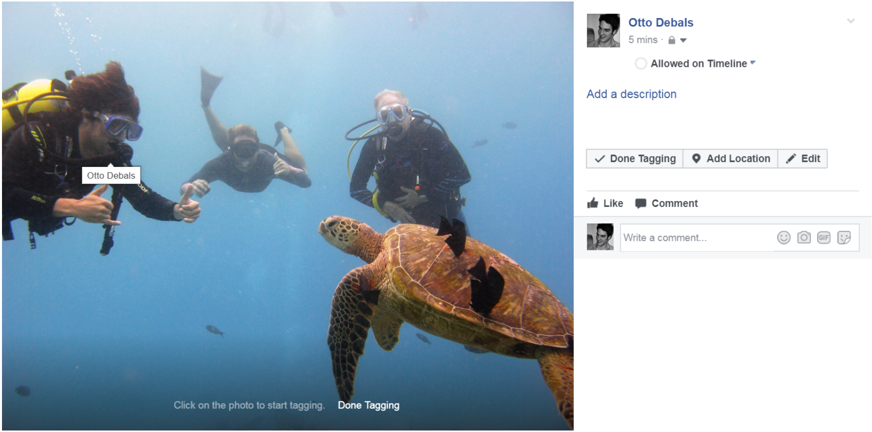
The generic problem of finding $\mathbf{b}$ and $\mathbf{c}$ in

$$\mathbf{d} = \mathbf{A}(\mathbf{b} \otimes \mathbf{c})$$

has many other applications [Boussé et al., 2017]:

- Classification of ECG signals
- Blind separation of telecommunication signals
- Weighted tensor decompositions
- Tensor decomposition updating
- ...

Kronecker Product Equations are a powerful tool for many problems, including face recognition



Numerical optimization algorithms for tensor-based face recognition

Otto Debals, Martijn Boussé, Nico Vervliet and Lieven De Lathauwer

KU LEUVEN

STADIUS
Center for Dynamical Systems, Signal Processing and Data Analytics

erc
European Research Council
Established by the European Commission