Analysis of Tensor Approximation for Compression-Domain Volume Visualization

Rafael Ballester-Ripoll, Susanne K. Suter, Renato Pajarola

rballester@ifi.uzh.ch

29th April 2016

Universität Zürich UZH

VISUALIZATION AND MULTIMEDIA LAB

Table of Contents

VISUALIZATIONAND
MULTIMEDIALAB

Section 1

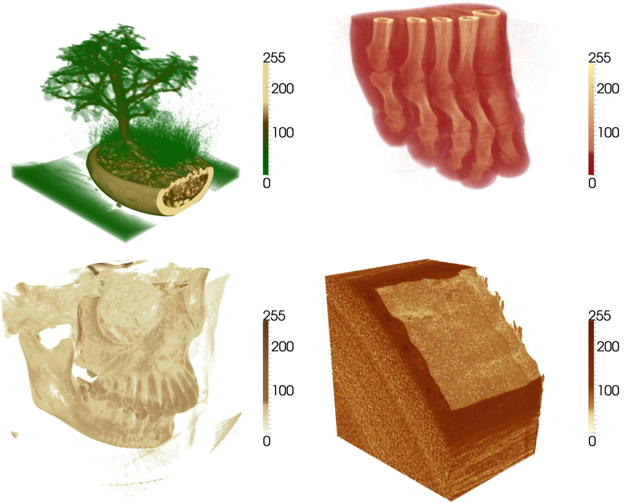
Background and Motivation

Context

- Large-scale interactive visualization: **complex** data over regular grids
  - Computer tomography, simulations, etc.
  - We tolerate (and encourage) approximations
- In volume rendering: data sets of size I^3, with I large (e.g. 2048).
  - Possible added dimension(s): features (RGB color, X-ray density), time, etc.
- Asymmetric pipeline:
  - Slow decomposition is acceptable (offline stage)
  - But **fast reconstruction** is critical (online stage)

VISUALIZATIONAND MULTIMEDIALAB

Example Volumes



VISUALIZATIONAND MULTIMEDIALAB

Tensor Approximation in Computer Graphics

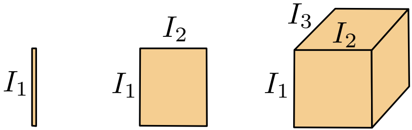
- **Texture synthesis** [VBP+05,CSS08,WXC+08]
- **Multiresolution rendering** [SIM+11,SMP13,BGG+14]
- **Micro-tomography compression** [BSP15, BP15]
- **Bidirectional texture functions** [WWS+05,WXC+08,RK09,TS12,Tsa15]
- **Bidirectional reflectance distribution functions** [RSK12]

Section 2

Introduction to Tensor Approximation

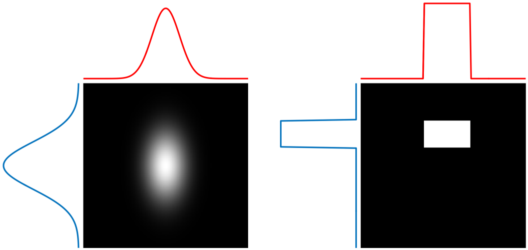
What is a Tensor?

- For us, a *multidimensional array*:
  - A vector is a 1D tensor
  - A matrix is a 2D tensor
  - Etc...



In a Nutshell

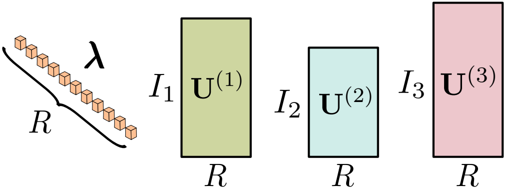
- Let us express a tensor as a **sum of *simpler* terms**
- Main ingredient: **separable** (rank-1) components
- Example in 2D (outer product $u \circ v$)



VISUALIZATIONAND MULTIMEDIALAB

CP Decomposition

- One **factor matrix** per dimension
  - Coefficients in a **diagonal** form

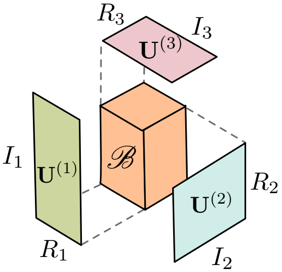


- Formula:

$$\mathscr{A} = \sum_{r=1}^{R} \lambda_r \cdot (u_r^{(1)} \circ_r^{(2)} \circ_r^{(3)})$$

Tucker Decomposition

- Generalization of CP
- We can enforce orthonormality
- Here, most space is **used by the core**



Tucker Decomposition

- Formula:

$$\mathscr{A} = \sum_{r_1=1, r_2=1, r_3=1}^{r_1=R_1, r_2=R_2, r_3=R_3} \mathscr{B}_{r_1 r_2 r_3} \cdot (u_{r_1}^{(1)} \circ u_{r_2}^{(2)} \circ u_{r_3}^{(3)})$$

- **Tensor-times-matrix** notation: $\mathscr{B} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)}$
- **Distance preservation**:

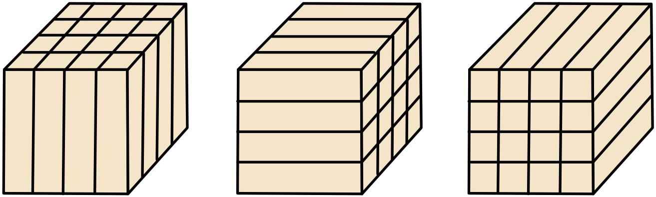
$$\left\{ \begin{array}{l} \mathscr{A}_1 \approx \mathscr{B}_1 \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \\ \mathscr{A}_2 \approx \mathscr{B}_2 \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \end{array} \right\} \Rightarrow \mathscr{B}_1 \approx \mathscr{B}_2$$

Decomposition Algorithms

- CP: challenging problem $\rightarrow$ algorithms only work well *in practice*
- Tucker: there are error bounds
  - **Algorithm gives intuition** $\rightarrow$ let us look at it

Fibers

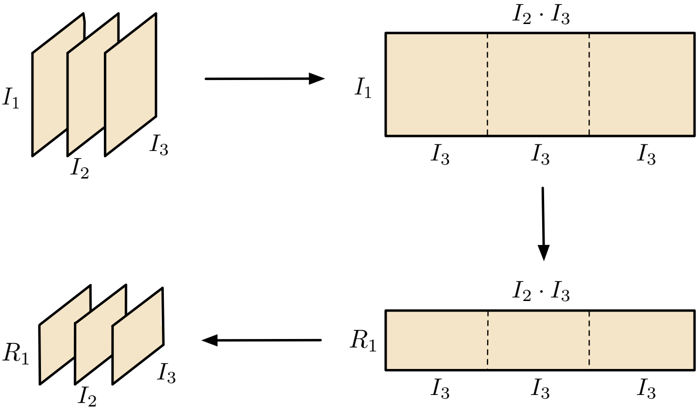
- In 3D: **columns**, **rows**, **tubes**



- Apply **principal component analysis** (PCA)

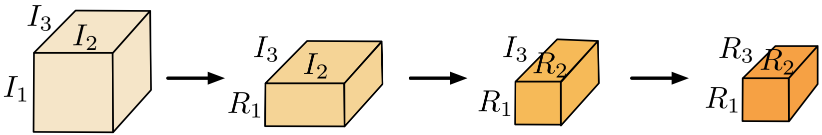
Unfoldings

- We do PCA **per fiber**



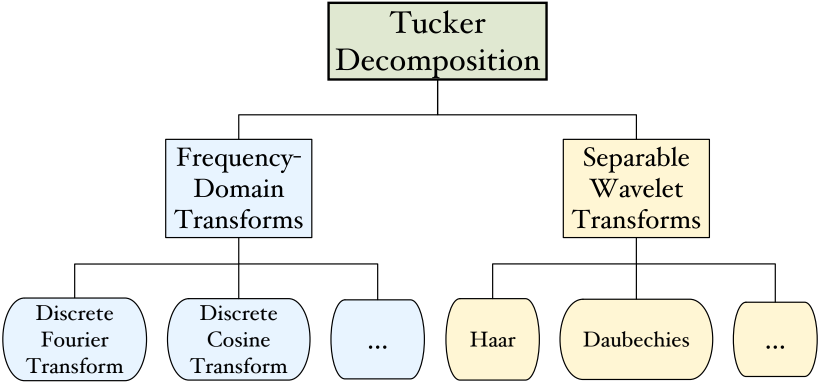
Multilinear Transforms

- **Orthogonal basis** of R vectors
- Fibers are compressed one dimension at a time



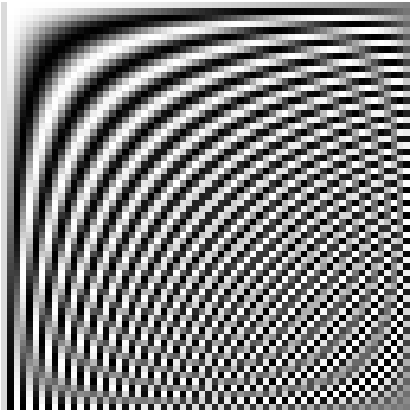
- This basis is precisely the matrices we want

In Context...

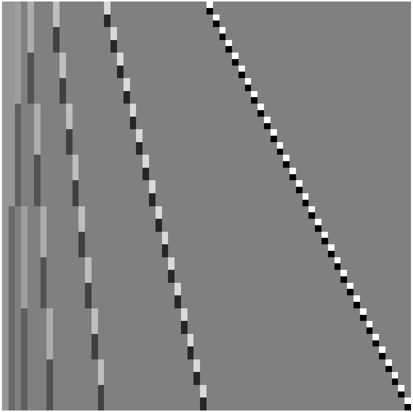


VISUALIZATIONAND MULTIMEDIALAB

Discrete Cosine Transform

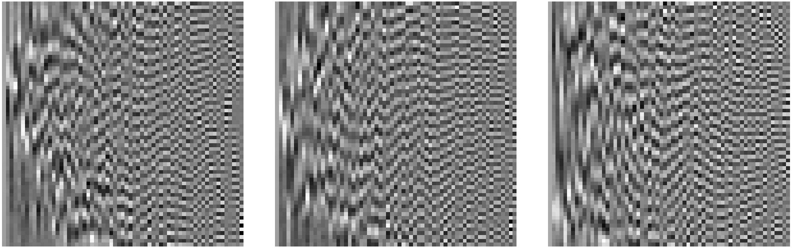


Haar Wavelets



Tucker Decomposition

- We leave it free $\rightarrow$ find **optimal** matrices

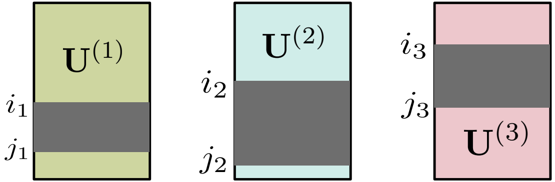


Advantages of Tensor Approximation

- Optimal bases $\rightarrow$ **competitive compression** rates
  - Good for out-of-core solutions
  - Often, the compressed data fits **entirely in the GPU**
- For *many dimensions*, virtually **the only way to go**
- We can operate on the factor matrices:
  - Translation
  - Stretching
  - Projection
  - Convolution
  - Frequency-domain transforms
- Example: DCT on the factors + Reconstruction = Reconstruction + DCT on the result

Spatial Selectivity

- To reconstruct the subregion $[i_1, j_1] \times [i_2, j_2] \times [i_3, j_3]$:

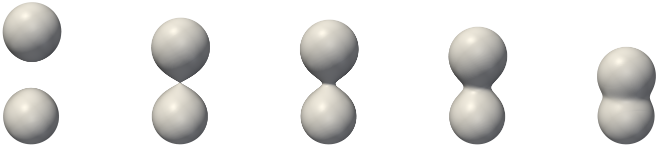


Section 3

Tensor-Based Compression

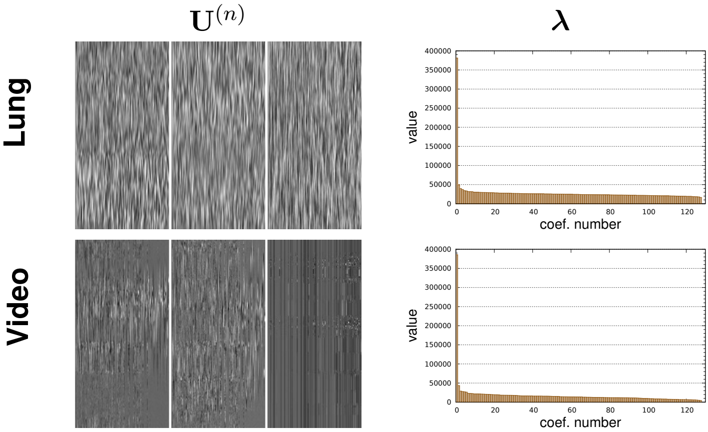
Smooth Feature Compression

- At high compression rates, tensor approximation is good at **preserving visual features**
- One way to see it: **isosurfaces**
  - For example, spheres are isosurfaces of multivariate Gaussians (rank-1)



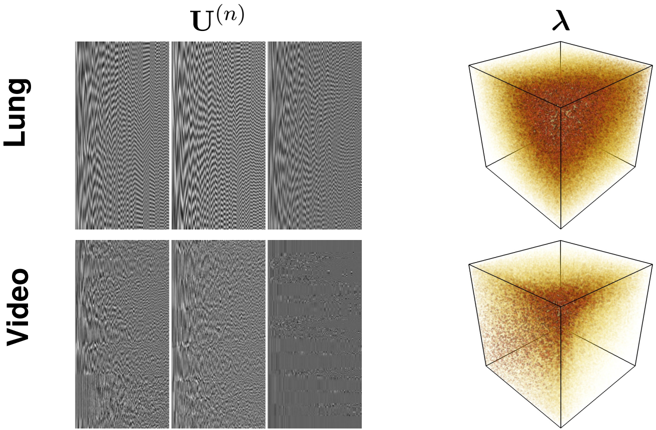
Metaballs: isosurfaces of a rank-1 function

Example: CP



$\mathbf{U}^{(n)}$
$\boldsymbol{\lambda}$
Lung
Video
value
coef. number
400000
350000
300000
250000
200000
150000
100000
50000
0
0
20
40
60
80
100
120

Example: Tucker

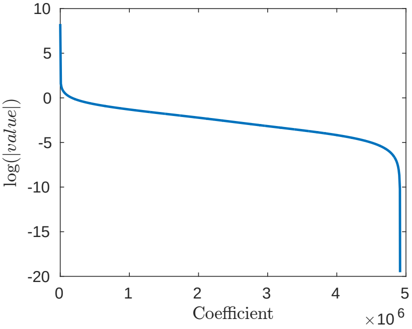


Volume Compression

- Quantization [SMP13]
- Thresholding [BP15]
- Truncation [SMP13,BP15,BSP15]

Quantization

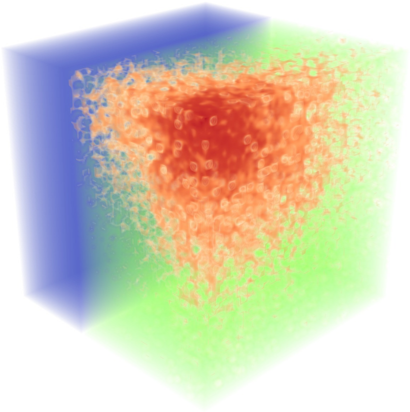
- Coefficients are roughly **logarithmic**:



- **Logarithmic quantization** works best
  - E.g. 8 bits + 1 sign bit

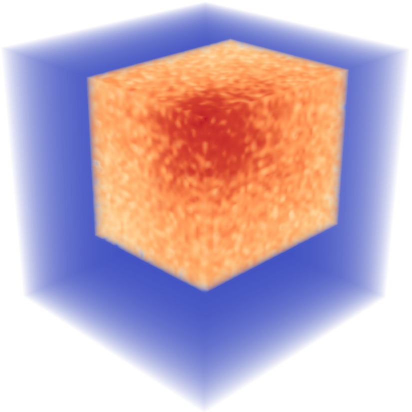
Tucker Thresholding

- Make small elements 0
- Run-length + Entropy encoding



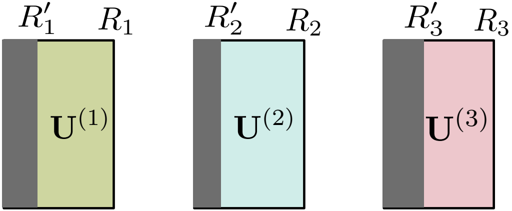
Tucker Truncation

- During visualization, **reduce** ranks as needed
- Very fast to apply



Tucker Truncation

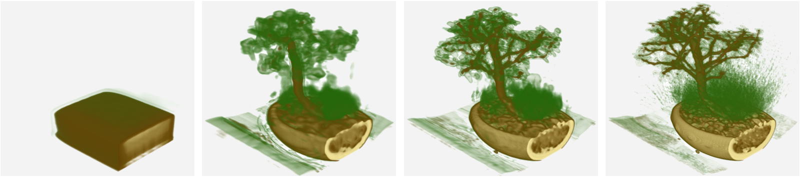
- Rank selection for interactive level-of-detail [SMP13]: Tucker core from $\mathbb{R}^{R_1 \times R_2 \times R_3}$ to $\mathbb{R}^{R'_1 \times R'_2 \times R'_3}$



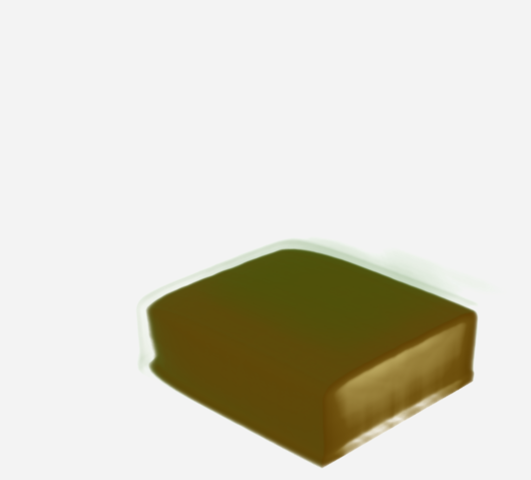
VISUALIZATIONAND MULTIMEDIALAB

Tucker Truncation

- **Different ranks select different features**
  - Example: bonsai (256^3), from 1 to 256 Tucker ranks



Tucker Truncation



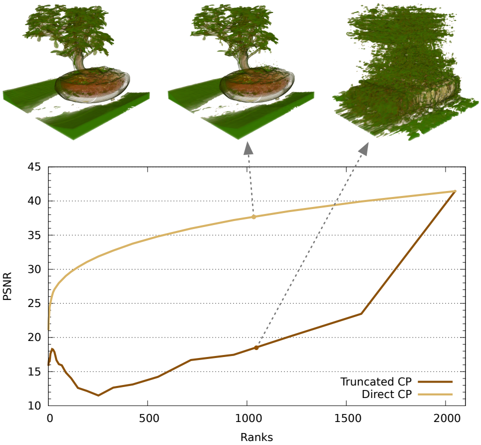
Tucker Truncation

Tucker Truncation

Tucker Truncation

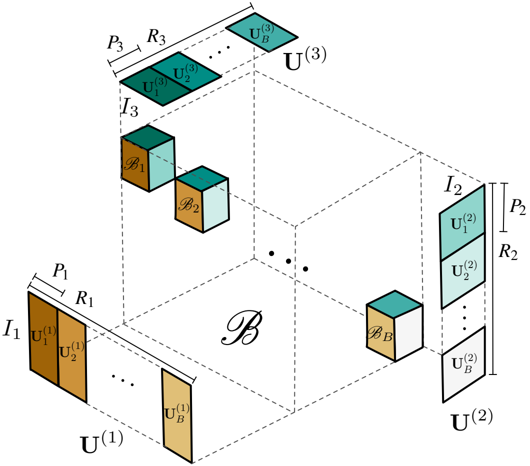
CP Rank Truncation

- Truncation problems

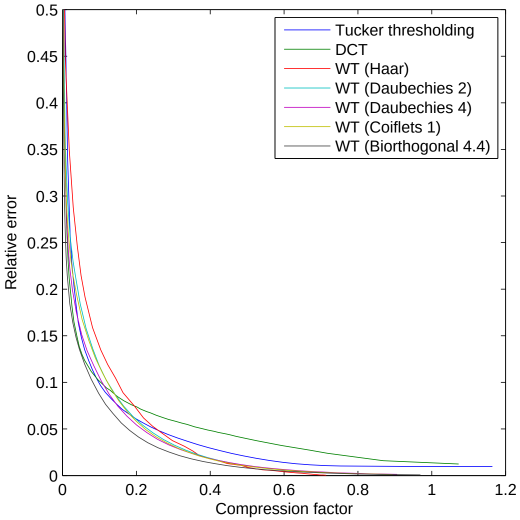


One Solution

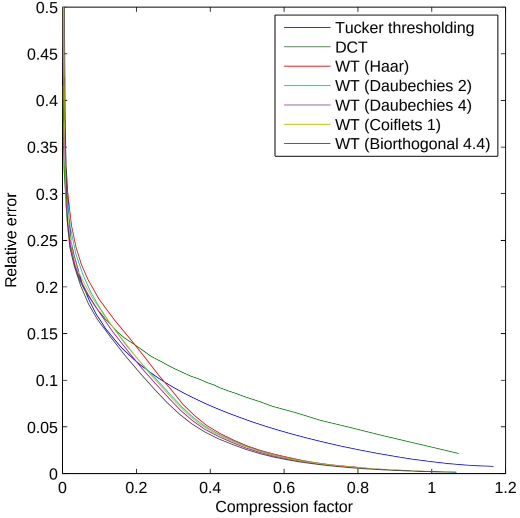
- Use incremental compression



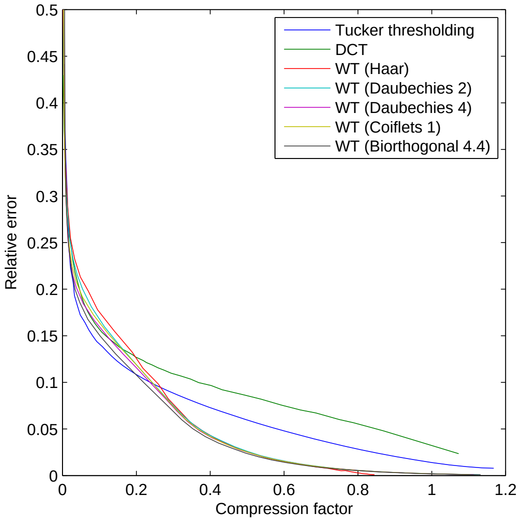
Tucker vs. Wavelets (Bonsai)



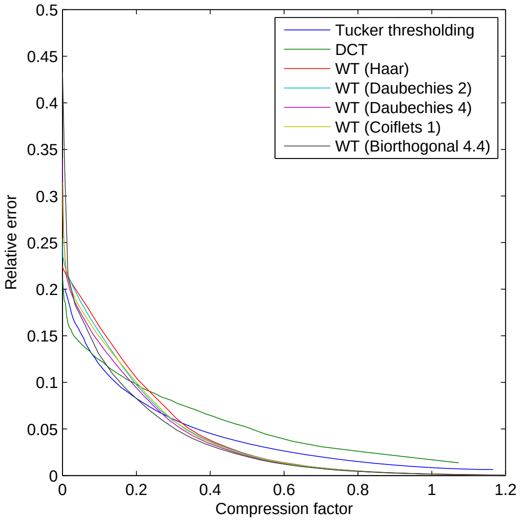
Tucker vs. Wavelets (Foot)



Tucker vs. Wavelets (Skull)

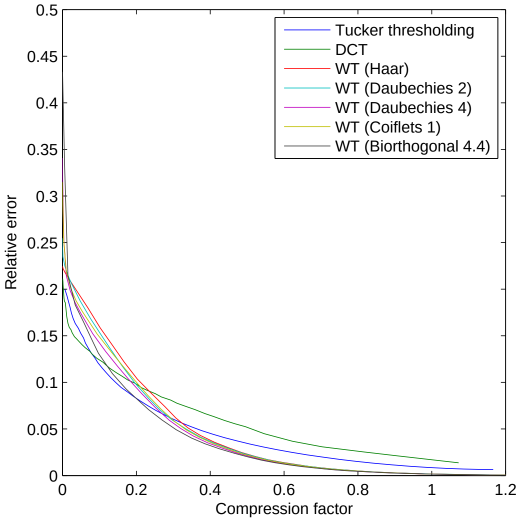


Tucker vs. Wavelets (Wood)



VISUALIZATIONAND MULTIMEDIALAB

Tucker vs. Wavelets (Wood)

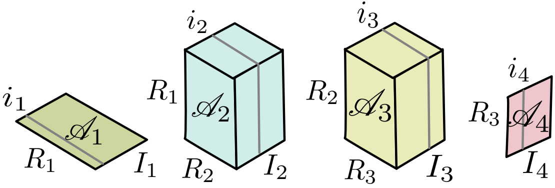


Software and Methods

- C++: **vmmlib**
- MATLAB: Tensor Toolbox, Tensorlab
- Decomposition:
  - Up to 2048^3 is fine
  - After that, there are **incremental algorithms**
- Reconstruction:
  - Must be **fast**. We have also a CUDA implementation

Future Work

- Tensor Train (TT): more recent model [Ose11]



- Even better suited for **many dimensions**
- Fast **random-access**

VISUALIZATIONAND MULTIMEDIALAB

Conclusions

- Tensor approximation **generalizes**:
  - **Frequency-based** transforms
  - Separable **wavelets**
- Good **compression quality**
- Good at **selecting features**
- Designed to overcome the **curse of dimensionality**
  - The more dimensions, the better the advantage

Thank you!

[VBP+05] D. Vlasic, M. Brand, H. Pfister, J. Popović: Face transfer with multilinear models.
[WWS+05] H. Wang, Q. Wu, L. Shi, Y. Yu, N. Ahuja: Out-of-core tensor approximation of multi-dimensional matrices of visual data.
[CSS08] R. Costantini, L. Sbaiz, S. Süsstrunk: Higher order SVD analysis for dynamic texture synthesis.
[WXC+08] Q. Wu, T. Xia, C. Chen, H.-Y. Lin, H. Wang, Y. Yu: Hierarchical tensor approximation of multidimensional visual data.
[RK09] R. Ruiters, R. Klein: BTF compression via sparse ensor decomposition.
[SIM+11] S. K. Suter, J. A. Iglesias Guitián, F. Marton, M. Agus, A. Elsener, C. Zollikofer, M. Gopi, E. Gobbetti, R. Pajarola: Interactive multiscale tensor reconstruction for multiresolution volume visualization.
[Ose11] I. Oseledets: Tensor-train decomposition.
[TS12] Y.-T. Tsai, Z.-C. Shih: K-clustered tensor approximation: a sparse multilinear model for real-time rendering.
[RSK12] R. Ruiters, C. Schwartz, R. Klein: Data driven surface reflectance from sparse and irregular samples.
[SMP13] S.K. Suter, M. Makhynia, R. Pajarola: TAMRESH: Tensor approximation multiresolution hierarchy for interactive volume visualization.
[BGG+14] M. Balsa Rodríguez, E. Gobbetti, J. A. Iglesias Guitián, M. Makhinya, F. Marton, R. Pajarola, S. K. Suter: State-of-the-art in compressed GPU-based direct volume rendering.
[Tsa15] Y.-T. Tsai: Multiway K-clustered tensor approximation: toward high-performance photorealistic data-driven rendering.
[BP15] R. Ballester-Ripoll, R. Pajarola: Lossy volume compression using Tucker truncation and thresholding.