Tutorial: Tensor Approximation in Visualization and Graphics

Graphics Applications

Renato Pajarola, Susanne K. Suter, and **Roland Ruiters**

University of Zurich UZH

VISUALIZATIONAND MULTIMEDIALAB

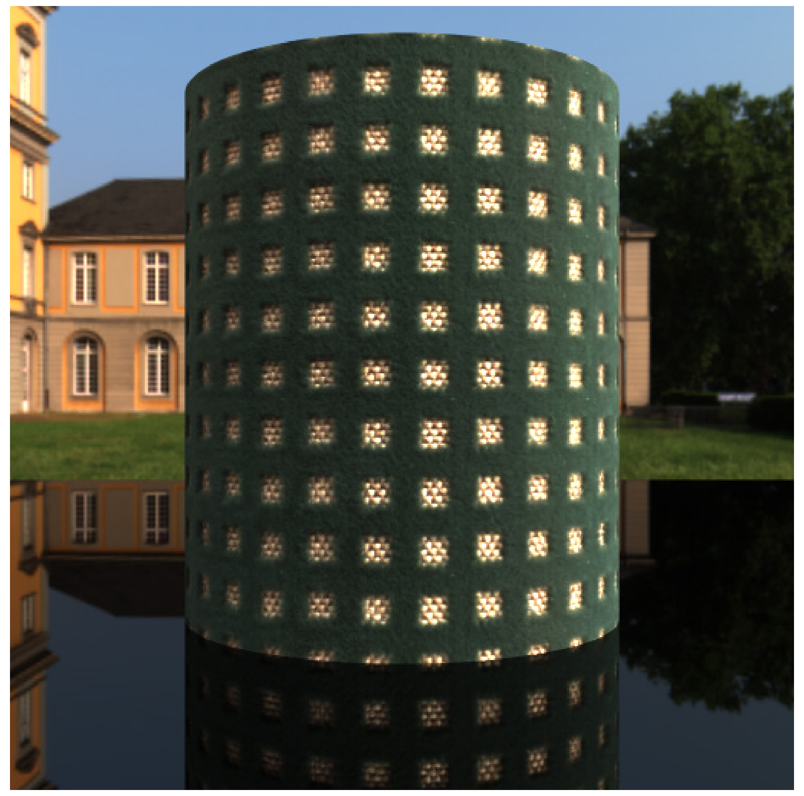
universitätbonn

Institute of Computer Science II
Computer Graphics

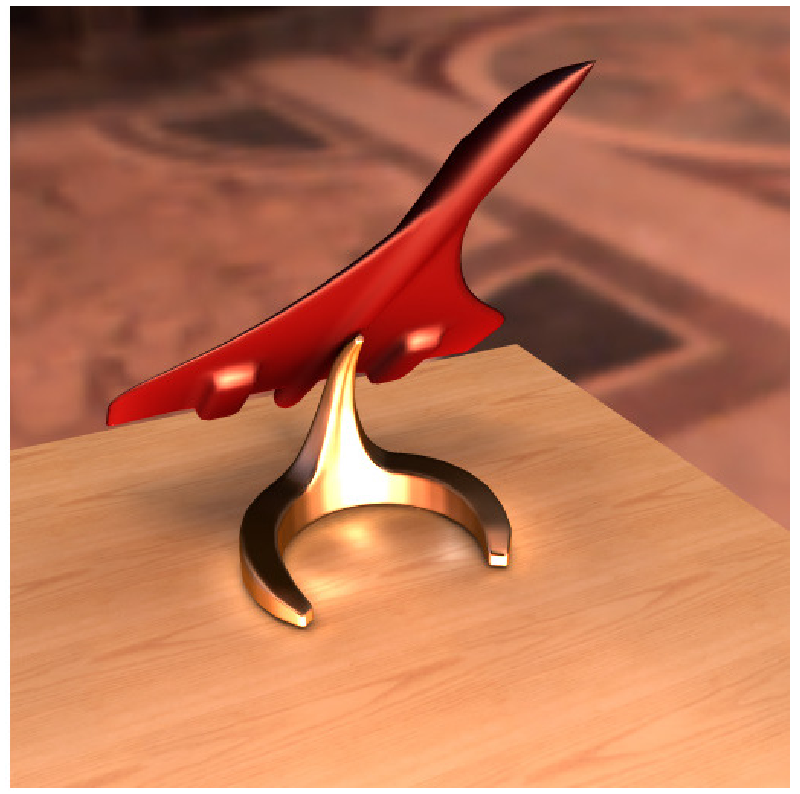
Multidimensional Datasets

Multidimensional datasets occur in many contexts in Computer Graphics

BRDFs

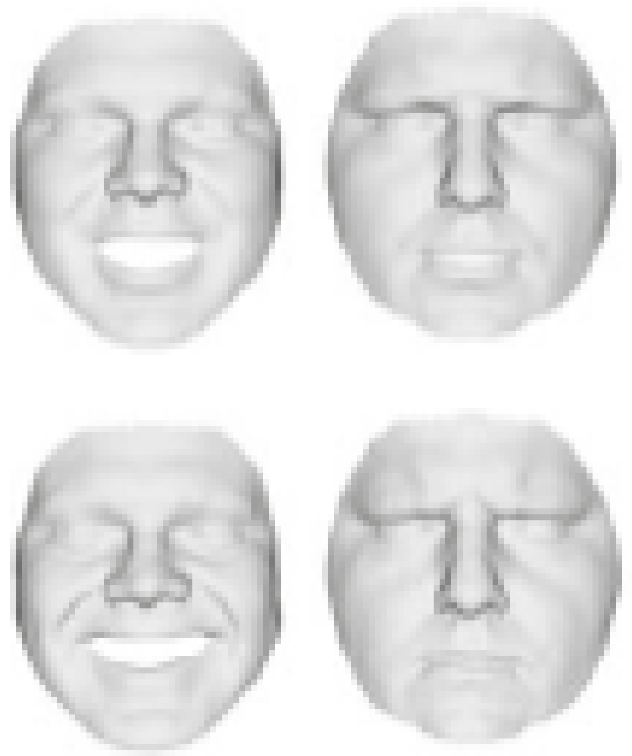


BTFs



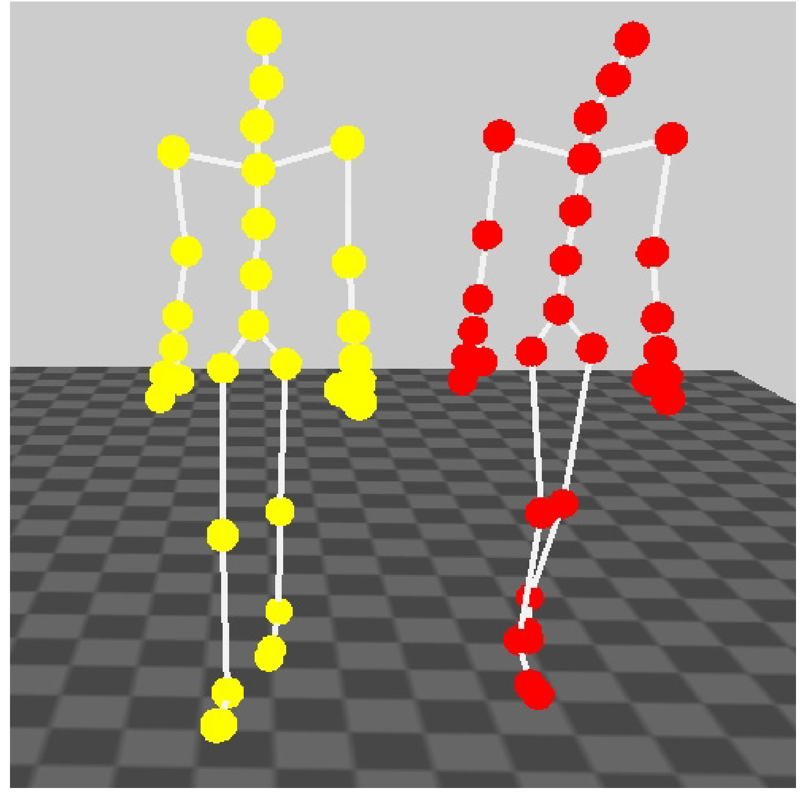
[Sun-2007]

Light Transport



[Vlasic-2005]

Image / Geometry Ensembles



[Krüger-2008]

Motion

[Wu-2008]

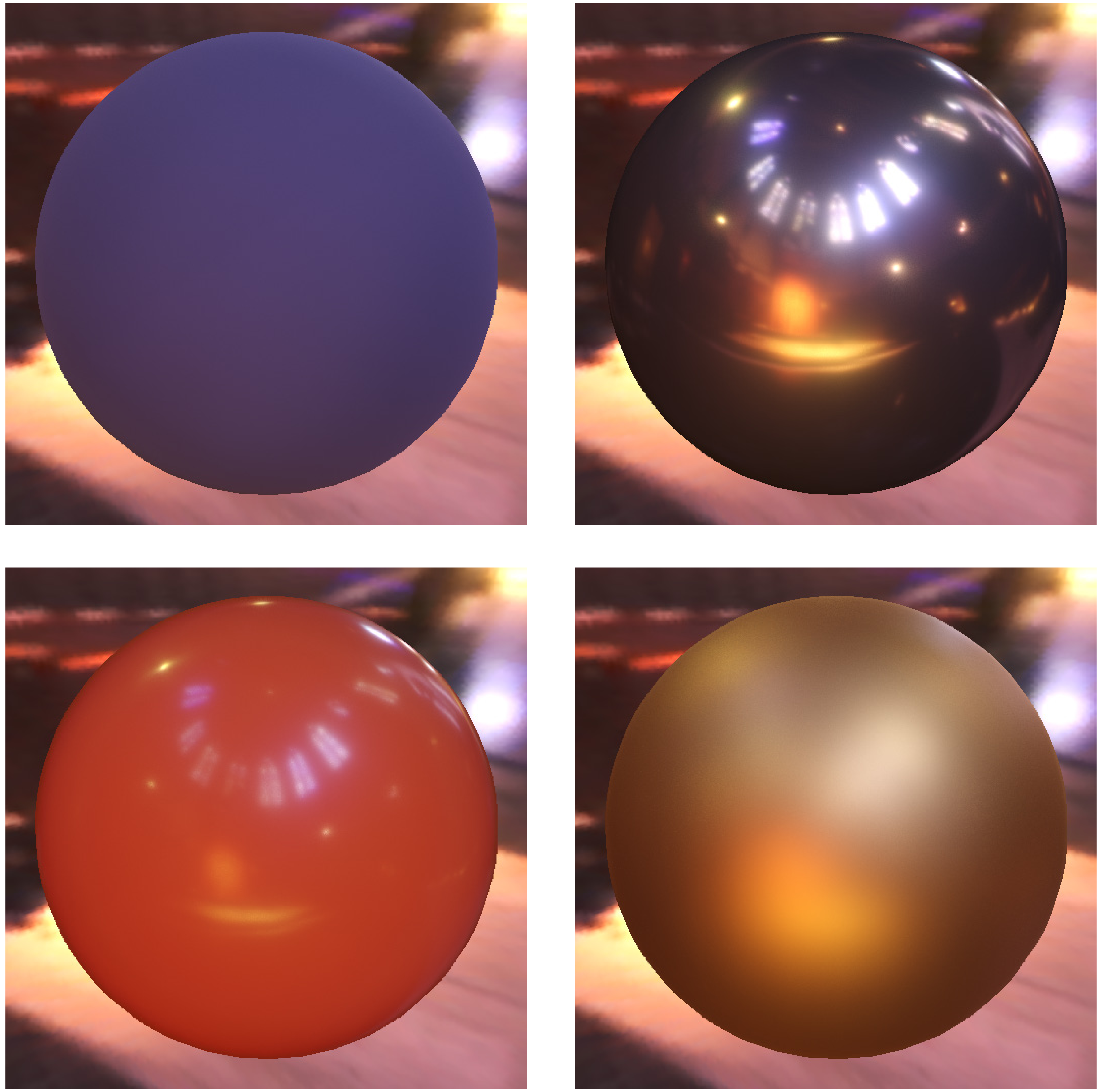
Dynamic Sequences

Bidirectional Reflectance Distribution Function (BRDF)

- 5-dimensional function
- Ratio between incoming irradiance and reflected radiance

$$\boldsymbol{\rho}(\boldsymbol{\varphi}_i, \boldsymbol{\theta}_i, \boldsymbol{\varphi}_o, \boldsymbol{\theta}_o, \boldsymbol{\lambda})$$

$\boldsymbol{\varphi}_i, \boldsymbol{\theta}_i$	Incoming light direction
$\boldsymbol{\varphi}_o, \boldsymbol{\theta}_o$	Outgoing light direction
$\boldsymbol{\lambda}$	Wavelength

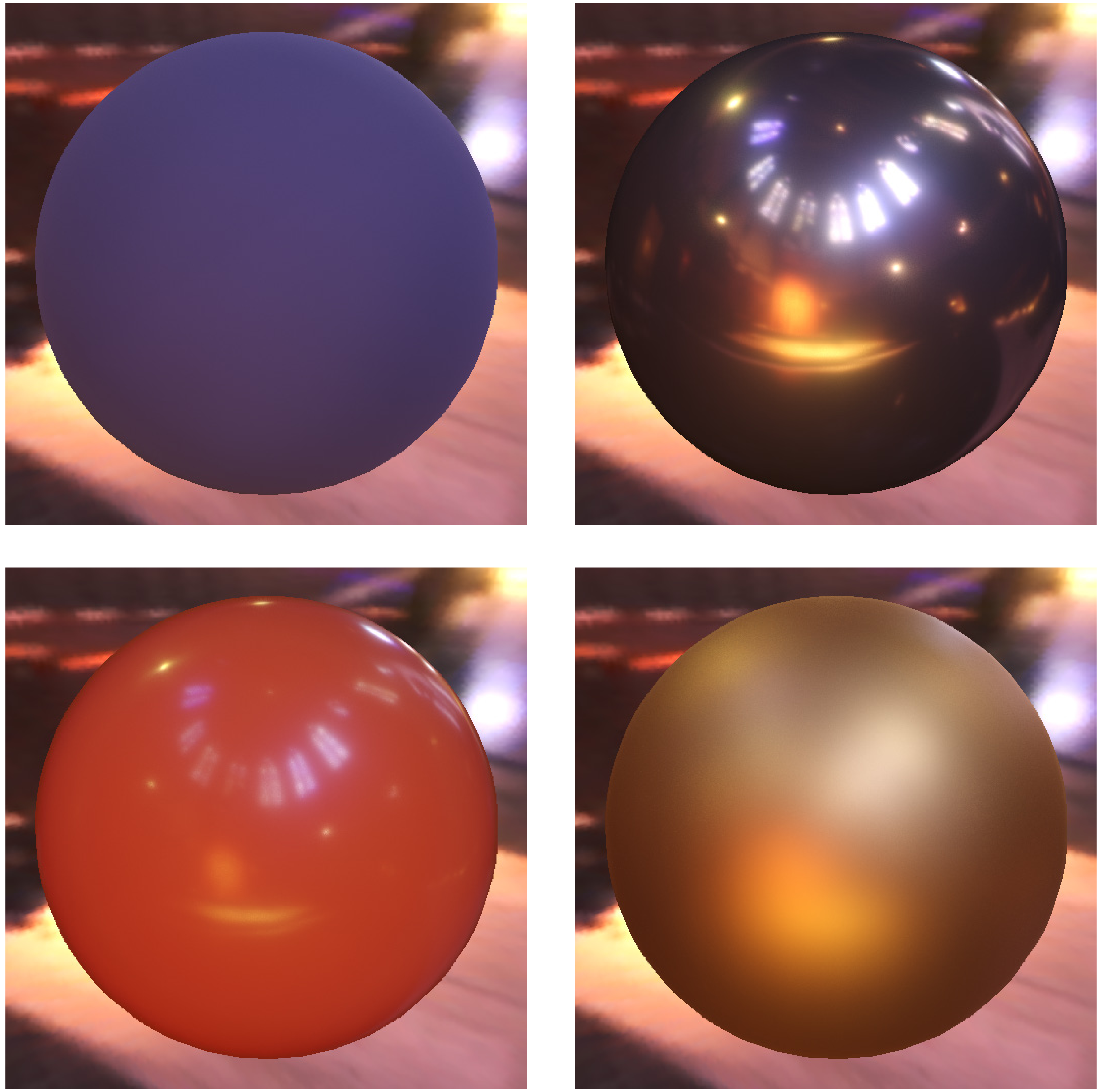


All used BRDF input samples are from the MERL BRDF Database [Matusik-2003]

Bidirectional Reflectance Distribution Function (BRDF)

- Focus mostly on isotropic BRDFs

[Sun-2007]	Tucker factorization, database of BRDFS, In-Out Parameterization
[Schwenk-2010]	CP, spectral BRDF, Half-Diff Parameterization
[Ruiters-2010]	CP, Weights to handle dynamic range, Half-Diff Parameterization
[Bilgili-2010]	Repeated Tucker, Log transform to handle dynamic range, Half-Diff Parameterization



Bidirectional Texture Functions & Spatially Varying BRDFs

- 7-dimensional functions
- Description of the spatially varying reflection behavior of a surface.

$$\boldsymbol{\rho(x, y, \varphi_i, \theta_i, \varphi_o, \theta_o, \lambda)}$$

x,y	Position on surface
φ_i, θ_i	Incoming light direction
φ_o, θ_o	Outgoing light direction
λ	Wavelength

[Schwartz-2011]

VISUALIZATIONAND MULTIMEDIALAB

universitätbonn

Bidirectional Texture Functions & Spatially Varying BRDFs

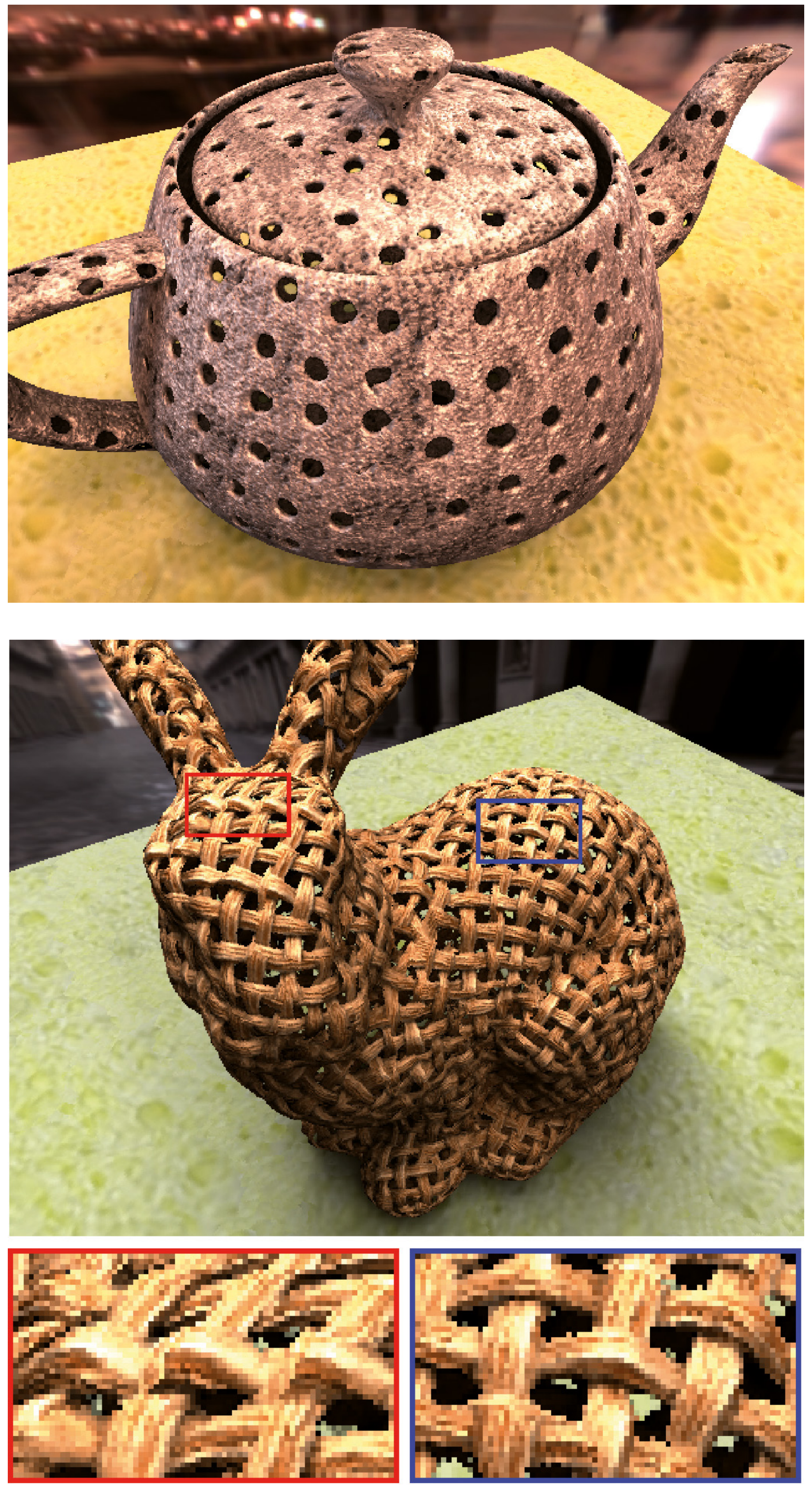
- Several approaches
  - Can be classified by decomposition type and tensor layout:

	Decomposition	Tensor Layout
[Furukawa-2002]	CANDECOMP/PARAFAC	View × Light × Position
[Vasilescu-2004]	Tucker	View × Light × Position
[Wang-2005]	Tucker	View × Light × X × Y
[Wu-2008]	Hierarchical Tucker	View × Light × X × Y
[Ruiters-2009]	Sparse Tensor Decomposition	View × (Color*Light) × Position
[Ruiters-2012]	CANDECOMP/PARAFAC	$\theta_h \times \theta_d \times \varphi_d \times$ Position × Color
[Tsai-2012]	K-CTA	View × Light × X × Y

[Schwartz-2011]

View-Dependent Occlusion Texture Functions

- Binary view-dependent opacity information [Tsai-2012]
  - Enables rendering of complex meso-structures with holes
- Results in a mode-3 tensor: View × X × Y
- Better to store signed distance function instead of binary texture



[Tsai-2012]

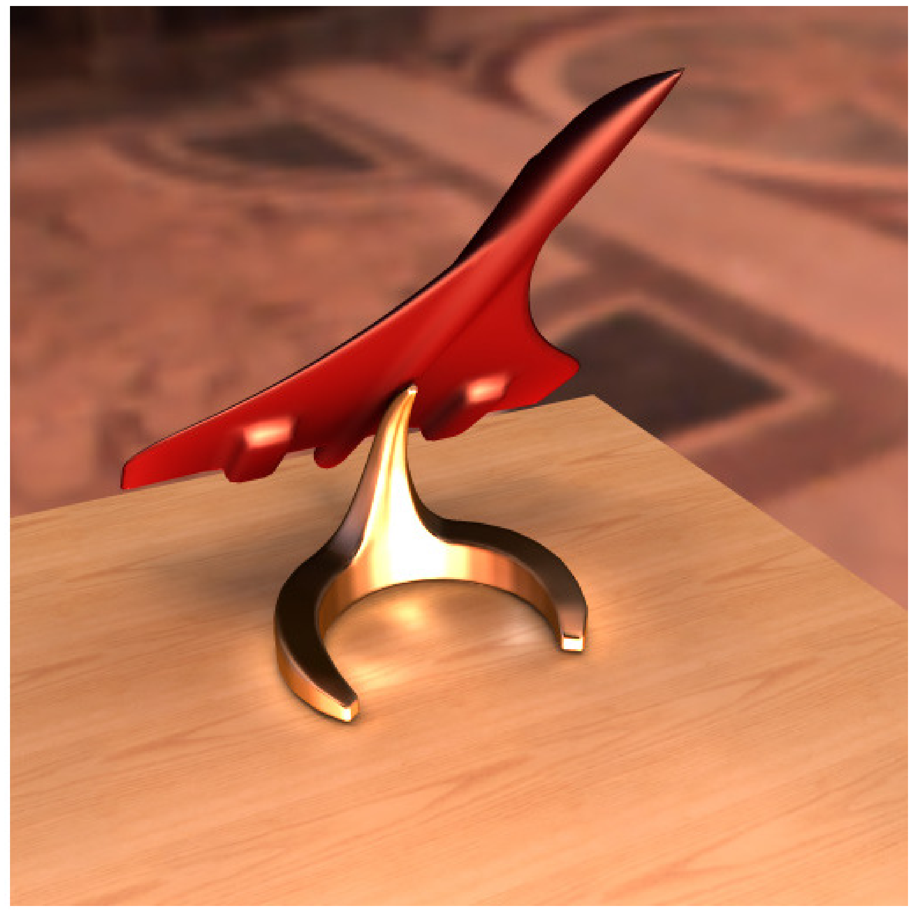
Precomputed / Captured Light Transport

- The **Reflectance Field** describes the light transport in a scene
- 11-dimensional function

$$R(x_i, y_i, z_i, \varphi_i, \theta_i; x_o, y_o, z_o, \varphi_o, \theta_o, \lambda)$$

  - For practical applications, simplifications to reduce the dimensionality of the function are necessary

[Tsai-2006]



[Sun-2007]

Precomputed / Captured Light Transport

[Garg-2006] Hierarchical Tensor Decomposition, Illumination and view point outside of the scene, Sparsity and symmetry of tensor utilized to improve measurement time, Mode-8 Tensor

[Tsai-2006] CTA, Representation of incoming and outgoing light using a linear basis, far field illumination, stored at vertices only, Mode-3 Tensor

[Sun-2007] CTA, Dynamic BRDFs introduce two additional modes per bounce for BRDF basis function and region: Mode-5 and Mode-7 Tensor for one and two bounces


[Tsai-2006]

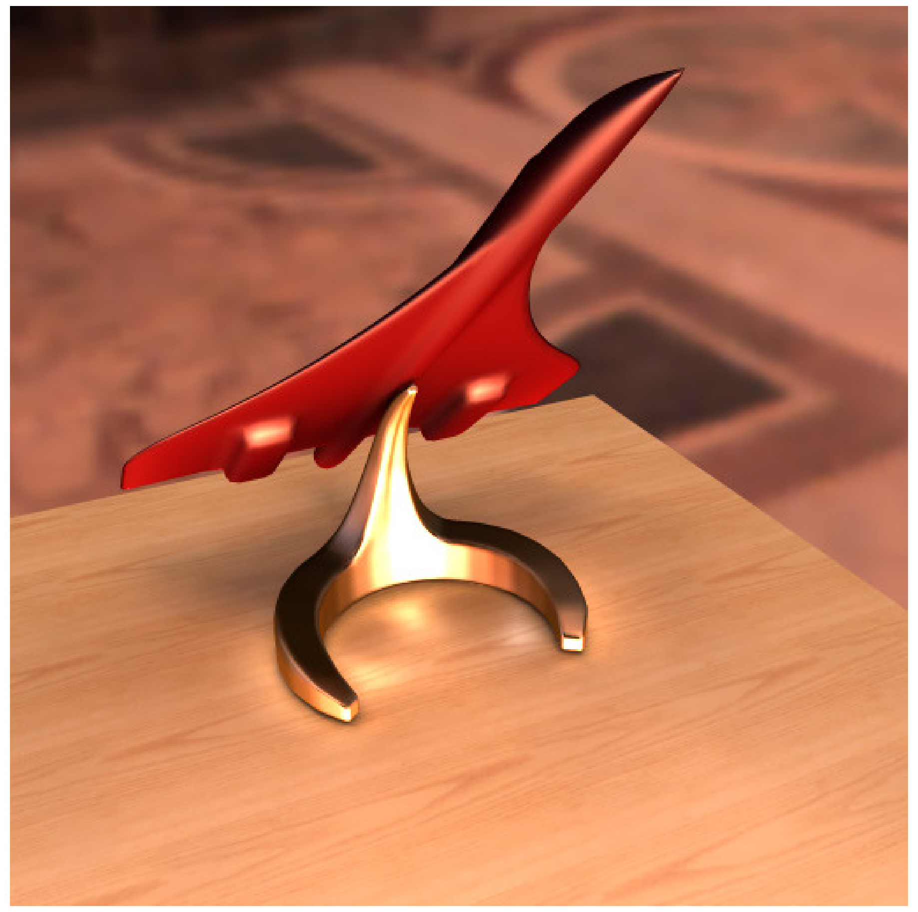
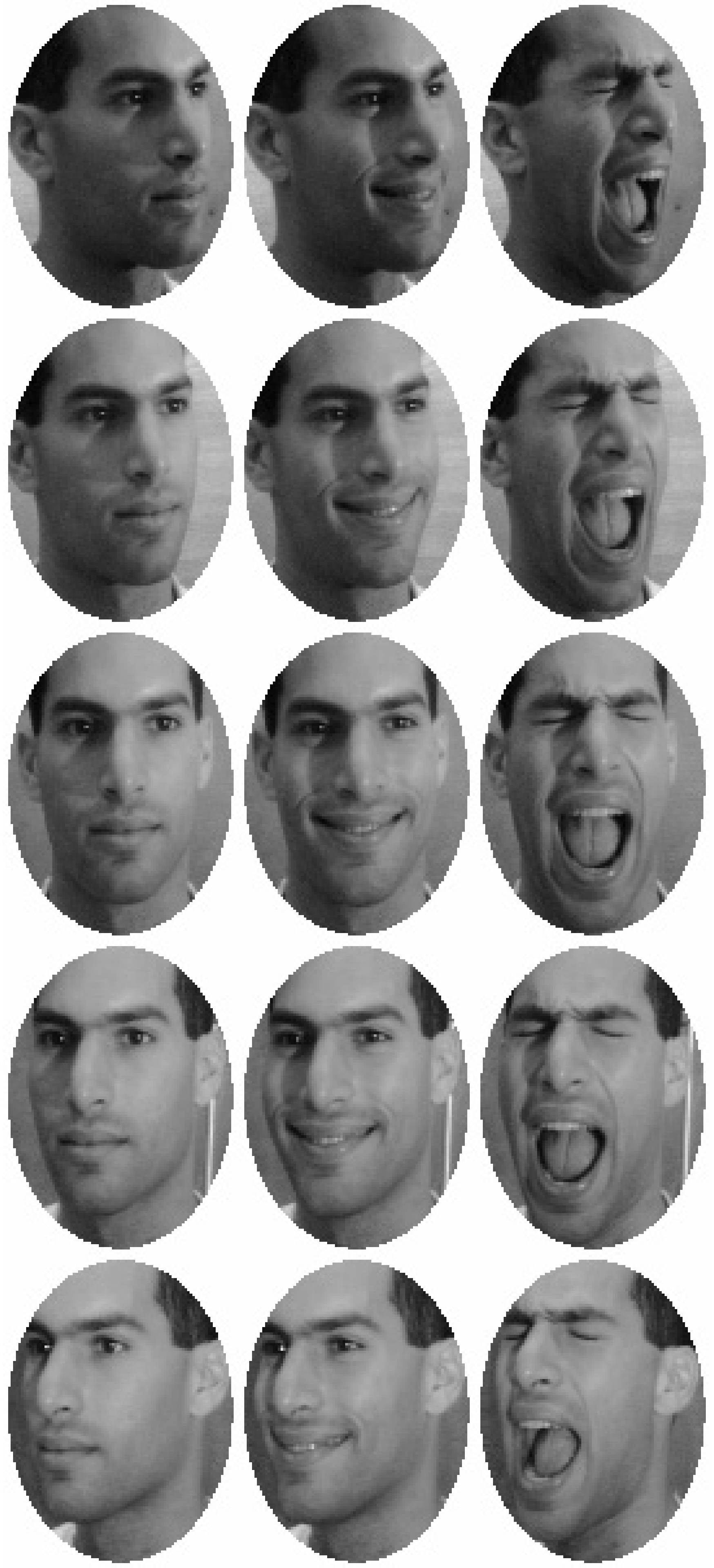
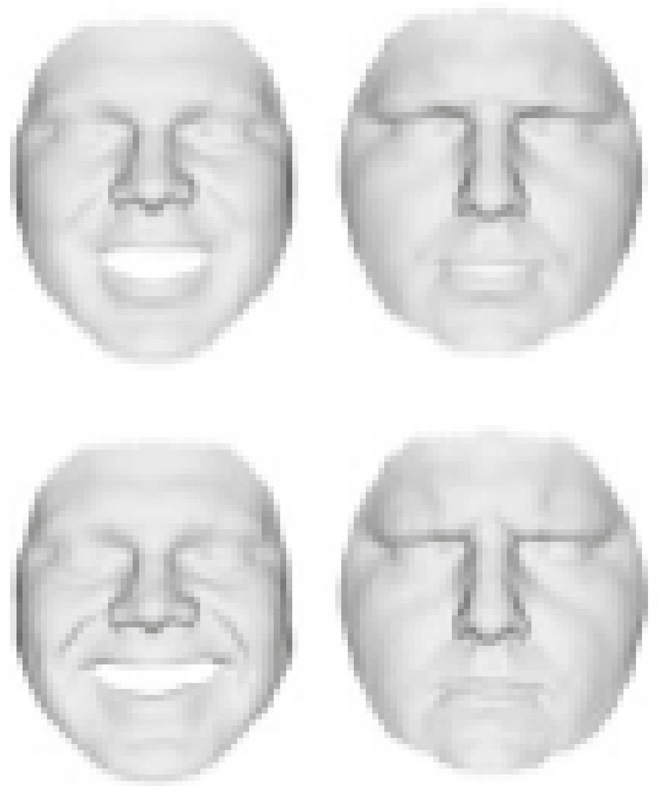

[Sun-2007]

Image / Geometry Ensembles

- In several applications one has to store a large collection of e.g.
  - Images (pixel colors)
    - [Vasilescu-2002a], [Vasilescu-2007], [Tu-2009]
  - Silhouettes (binary values)
    - [Peng-2008]
  - Geometry (vertex positions)
    - [Vlasic-2005],[Hasler-2010]
- in dependence on several parameters such as
  - Actor
  - Pose / Expression
  - Orientation
  - Illumination



[Vasilescu-2002]



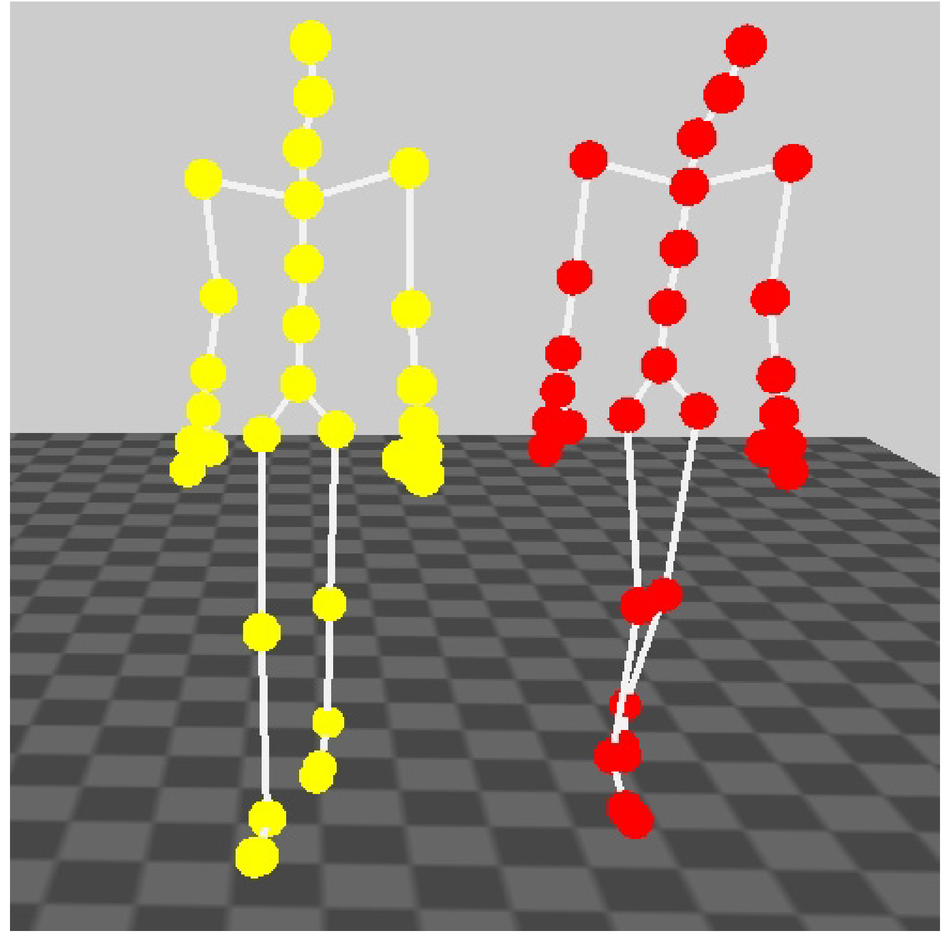
[Vlasic-2005]

[Peng-2008]

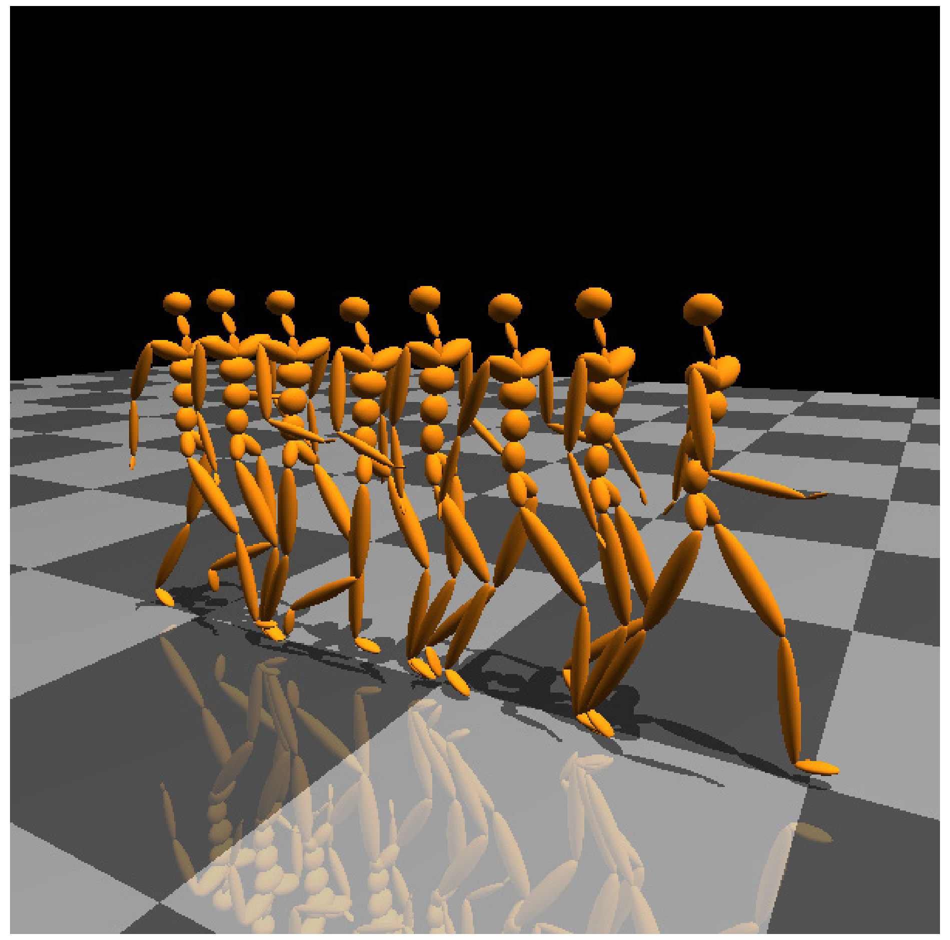
Motion

- Captured motion sequences consisting of
  - Center of gravity and joint angles
    - [Vasilescu-2002b], [Mukai-2007], [Krüger-2008], [Min-2010], [Liu-2011]
  - Positions of vertices or joints
    - [Perera-2007], [Wampler-2007]

- in dependence on parameters such as
  - Actor
  - Action
  - Style
  - Repetition number



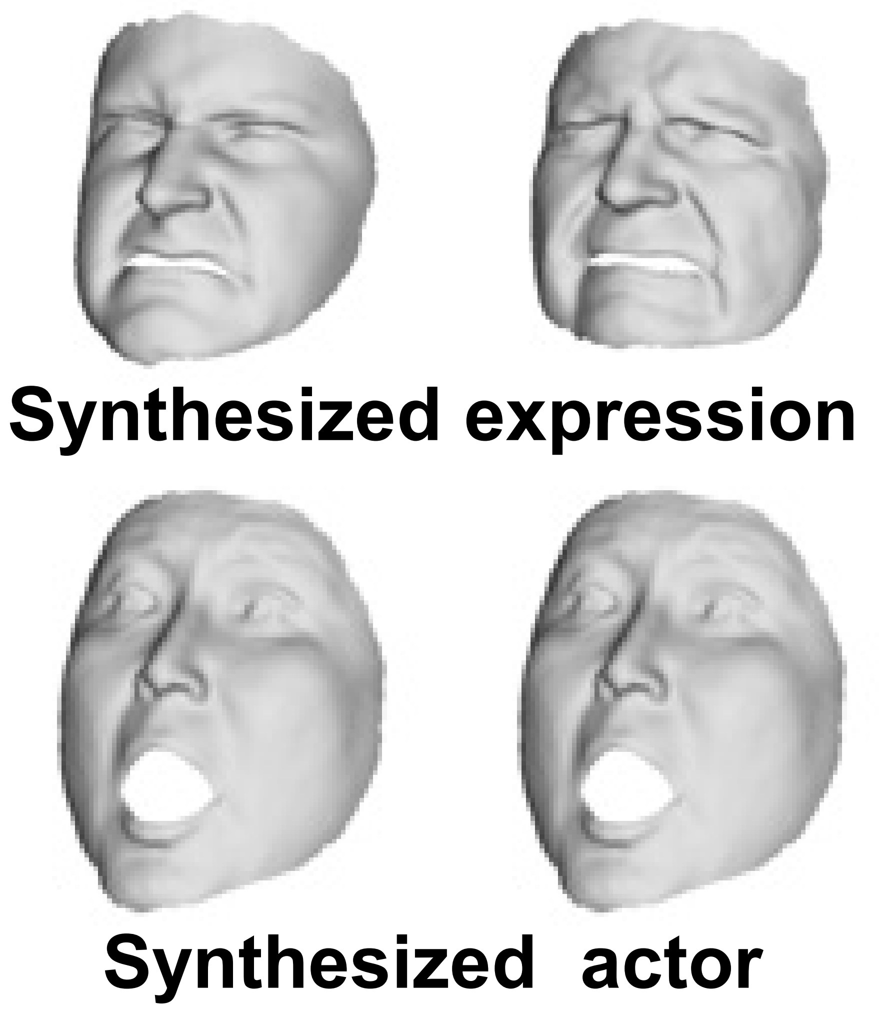
[Krüger-2008]

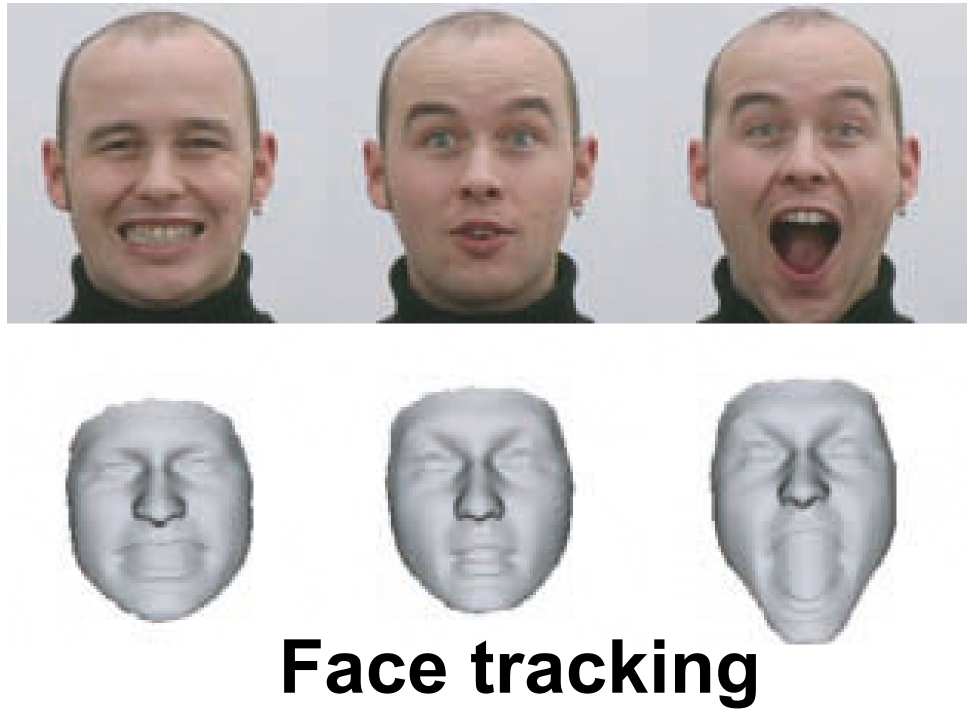


[Min-2010]

Applications

- A multi-linear model of such an ensemble has several possible applications:
  - Compression
  - Synthesis
    - Each row of the factor matrices U_i of a Tucker decomposition contains a set of weights describing the corresponding mode entry
      - By multiplying with a different set of weights a novel actor, motion, expression etc. can be synthesized
  - Imputation
    - How would an action look like, from an actor that was only filmed for different actions?
  - Recognition
    - To which actor and expression does this image correspond?

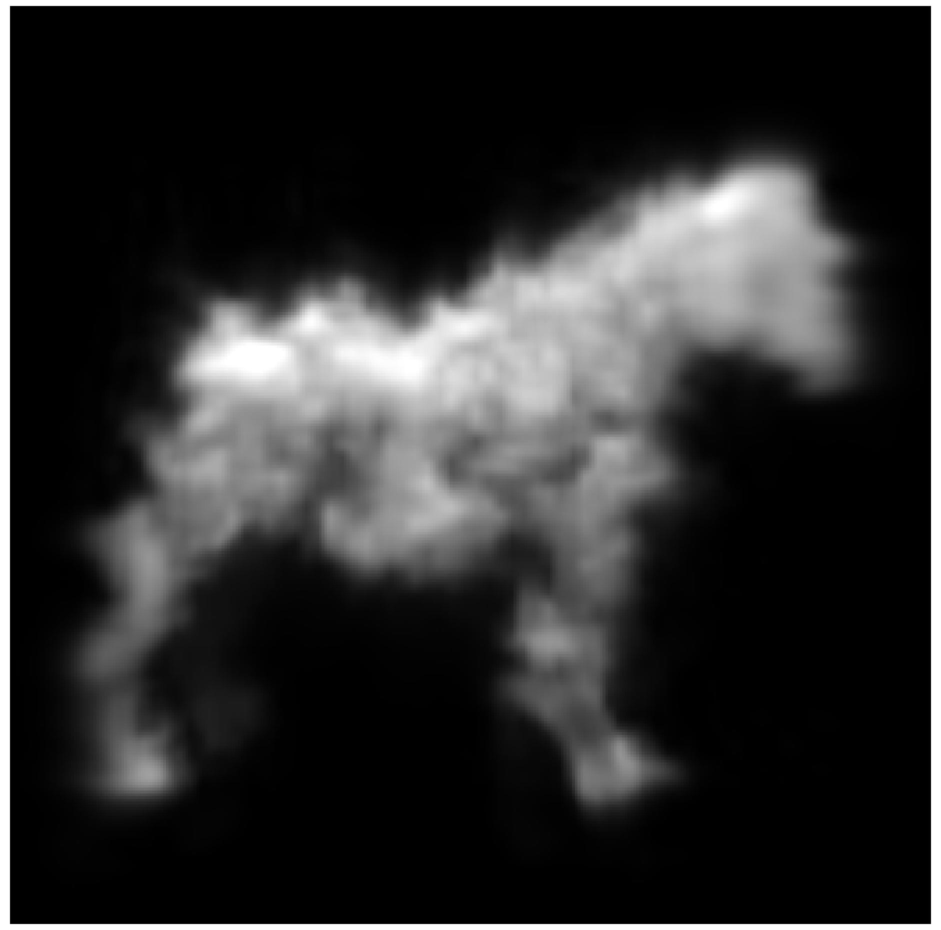




Examples from [Vlasic-2005]

Time Varying Sequences

- Adds an additional time dimension to datasets, such as
  - Textures
    - [Costantini-2008], [Wu-2008]
  - Reflectance
    - [Wang-2005]
  - Volumetric datasets
    - [Wang-2005], [Wu-2008]



[Wang-2005]

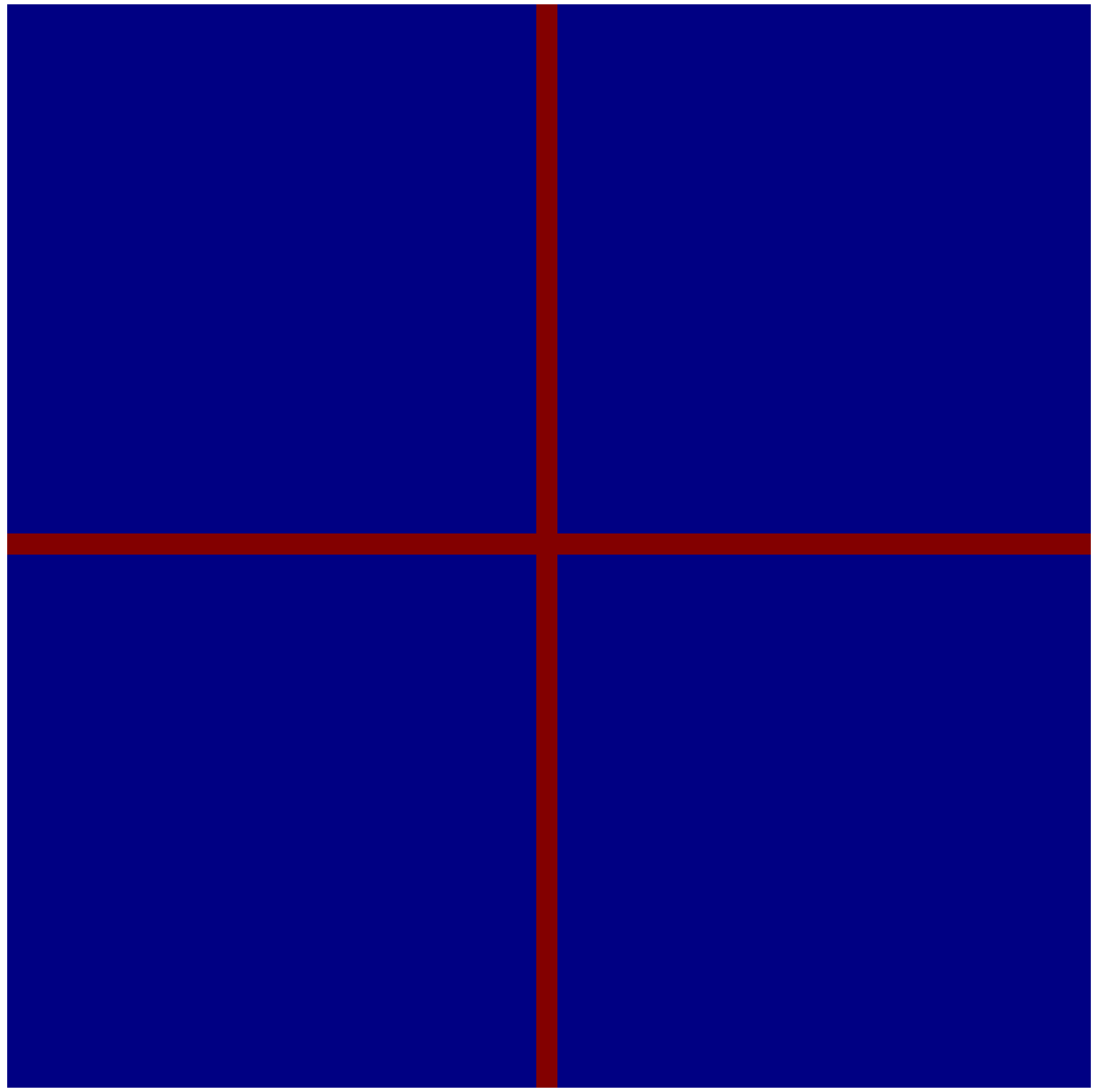
[Wu-2008]

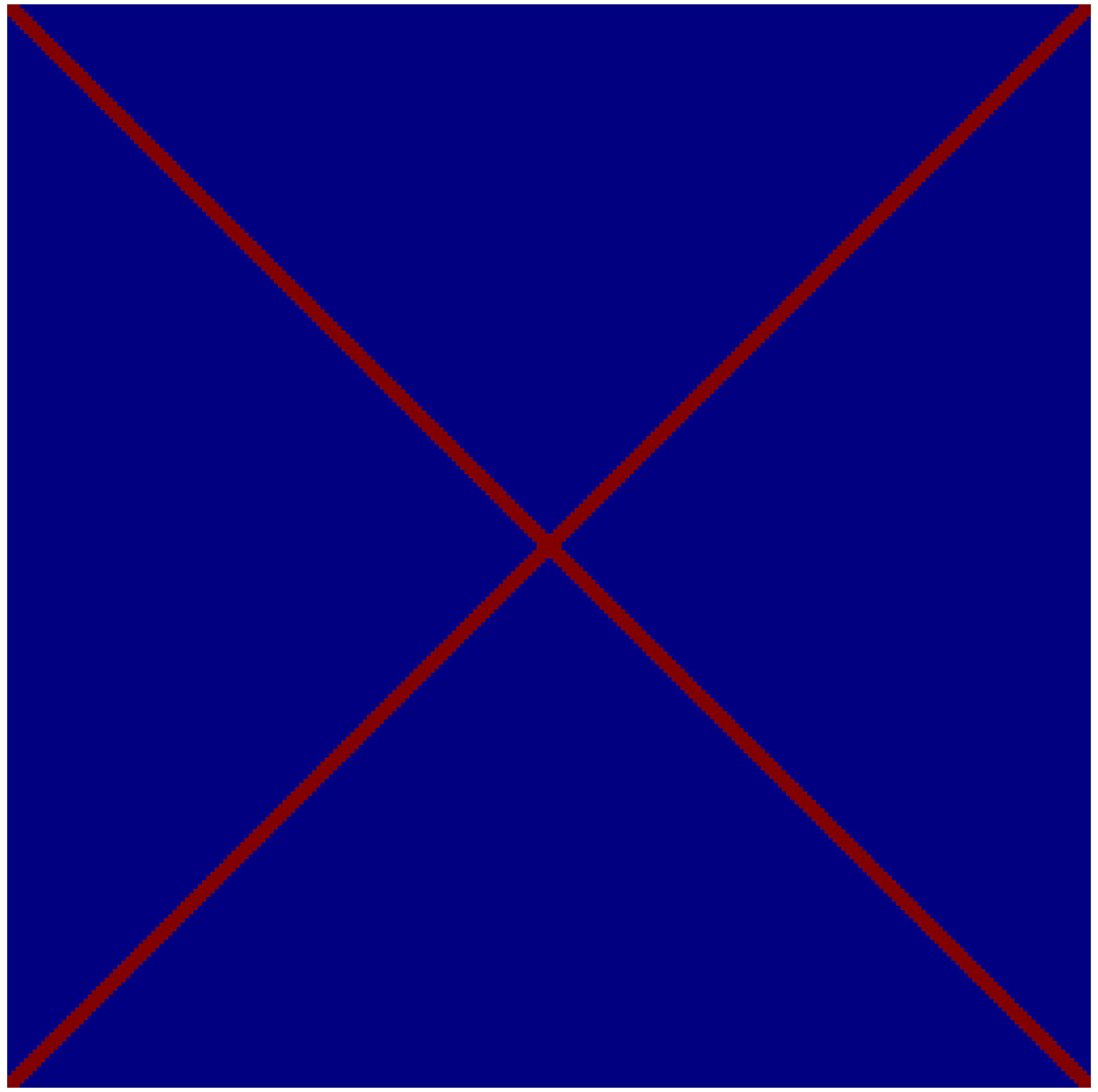
Tensor Approximation

- Several important questions have to be considered:
  - Which parameterization?
    - Is my input data registered correctly?
  - Which error measure?
  - Which decomposition?
  - Should every dimension be represented in an individual mode?

Parameterization

- Why is the parameterization of our function important?
- Lets consider two simple test cases (256x256 matrix with 0/1 values):



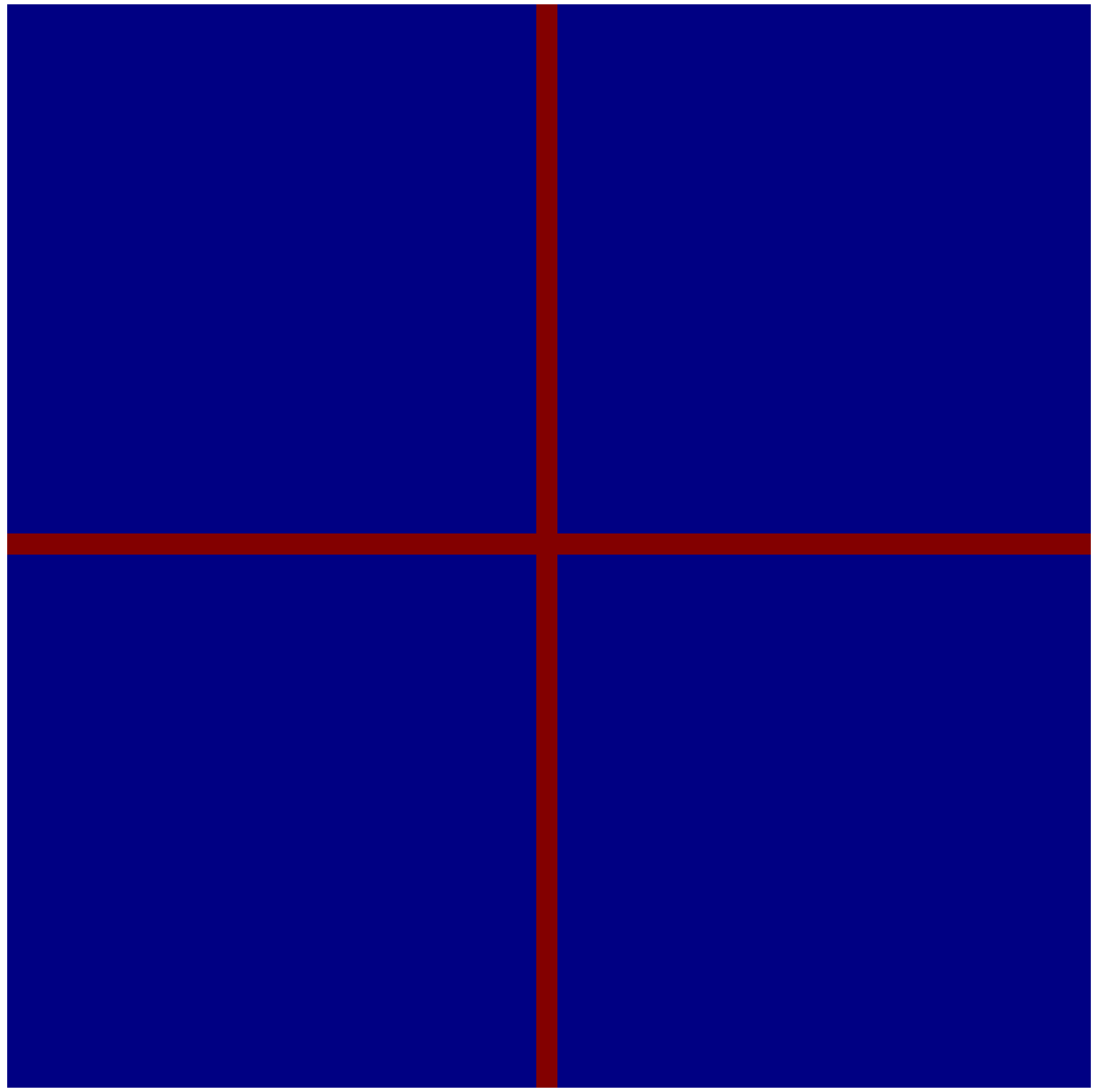


VISUALIZATIONAND MULTIMEDIALAB

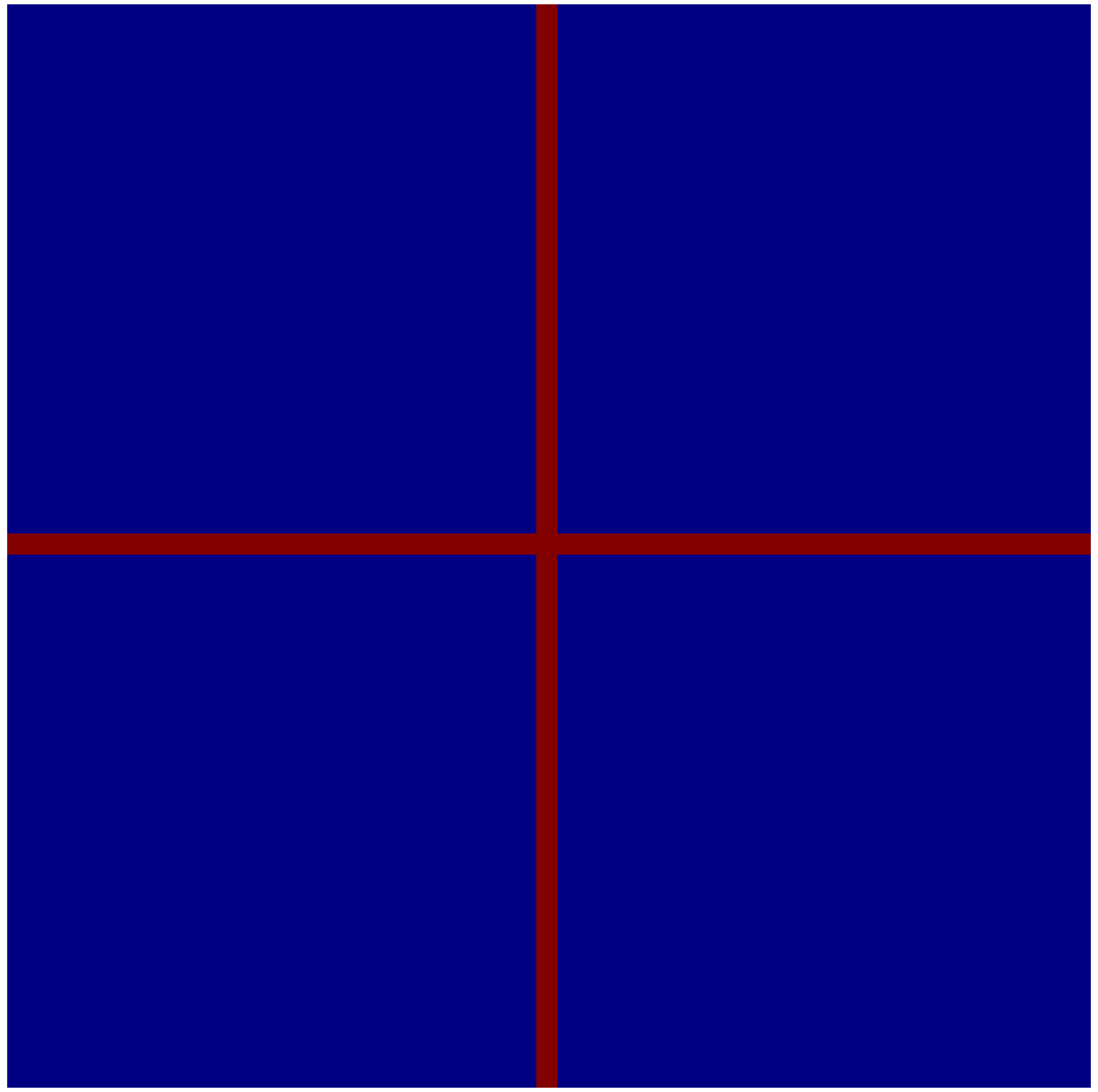
universitätbonn

Parameterization

- The first case can be approximated easily:



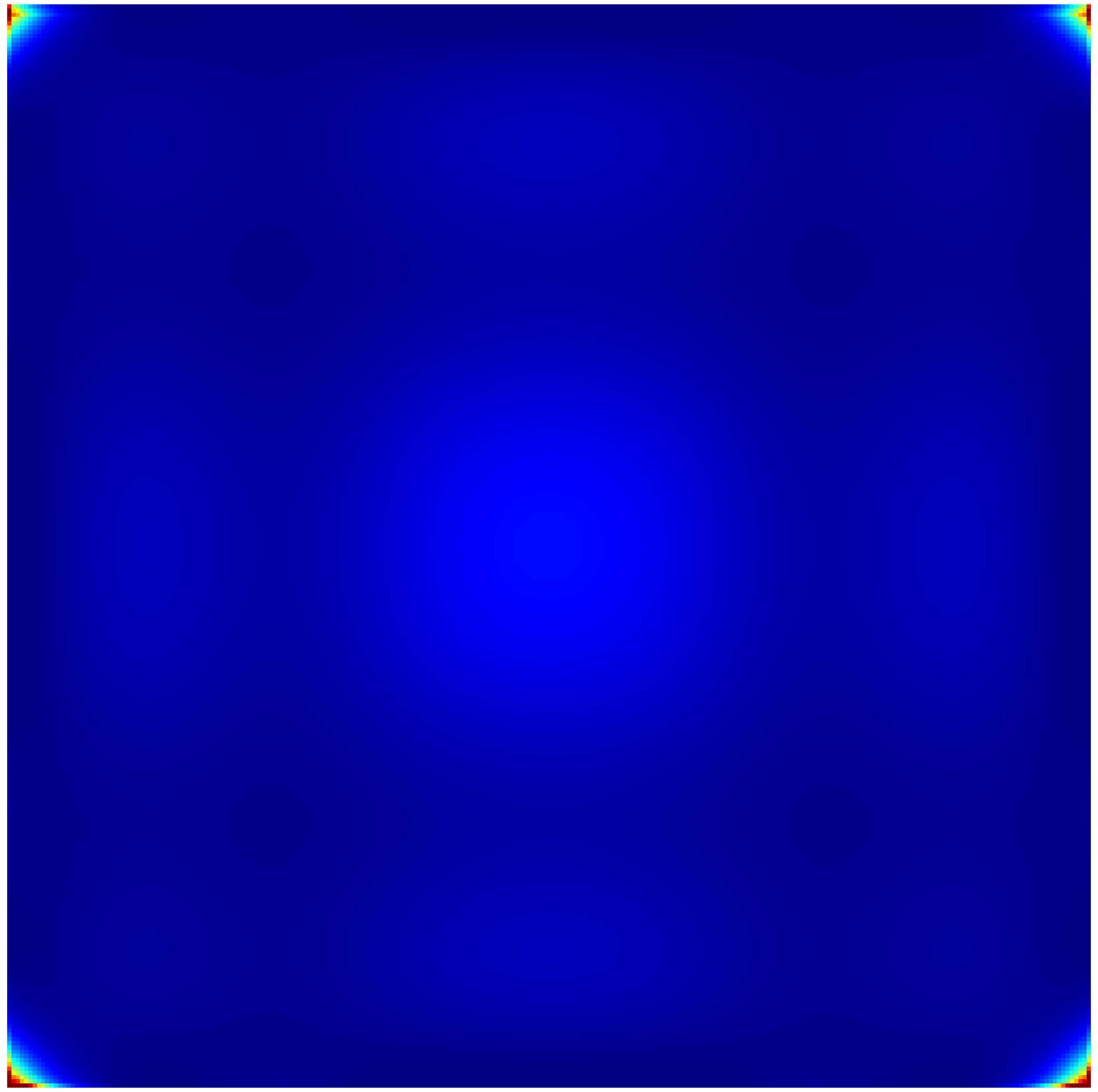
CP Decomposition
with 2 components



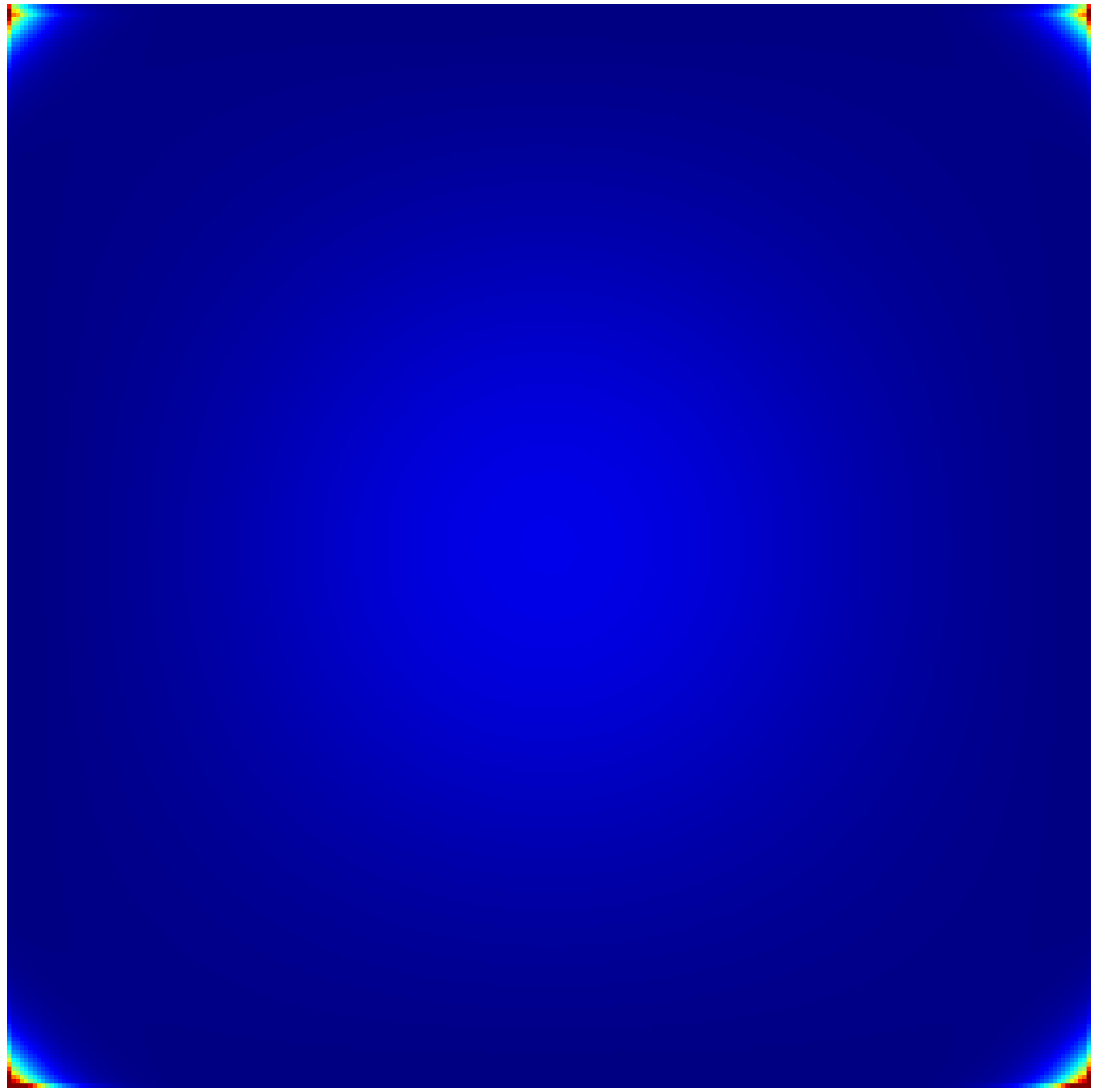
TUCKER Decomposition
with 2x2 core tensor

Parameterization

- But the second case is far more difficult:



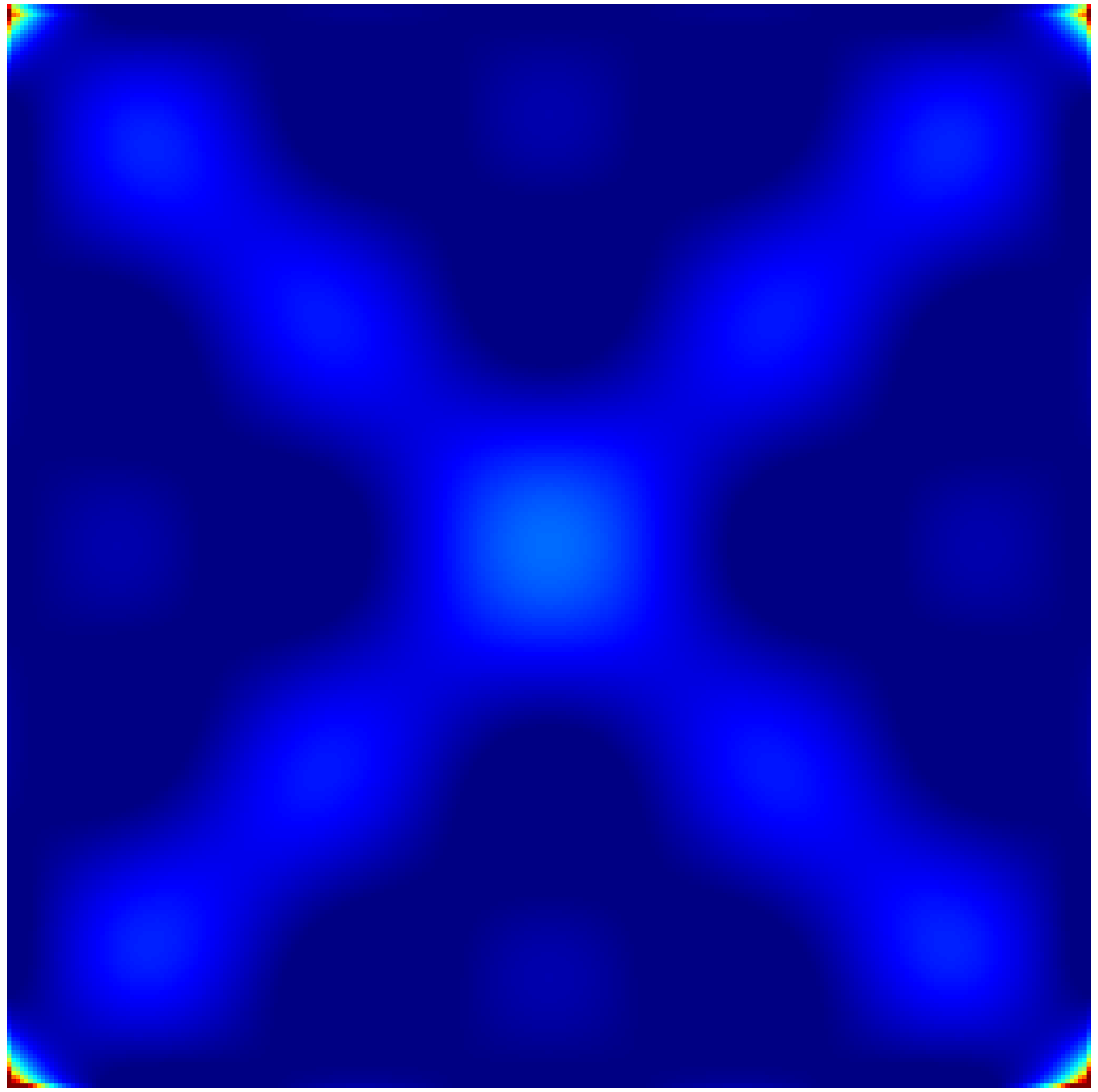
CP Decomposition
with 2 components



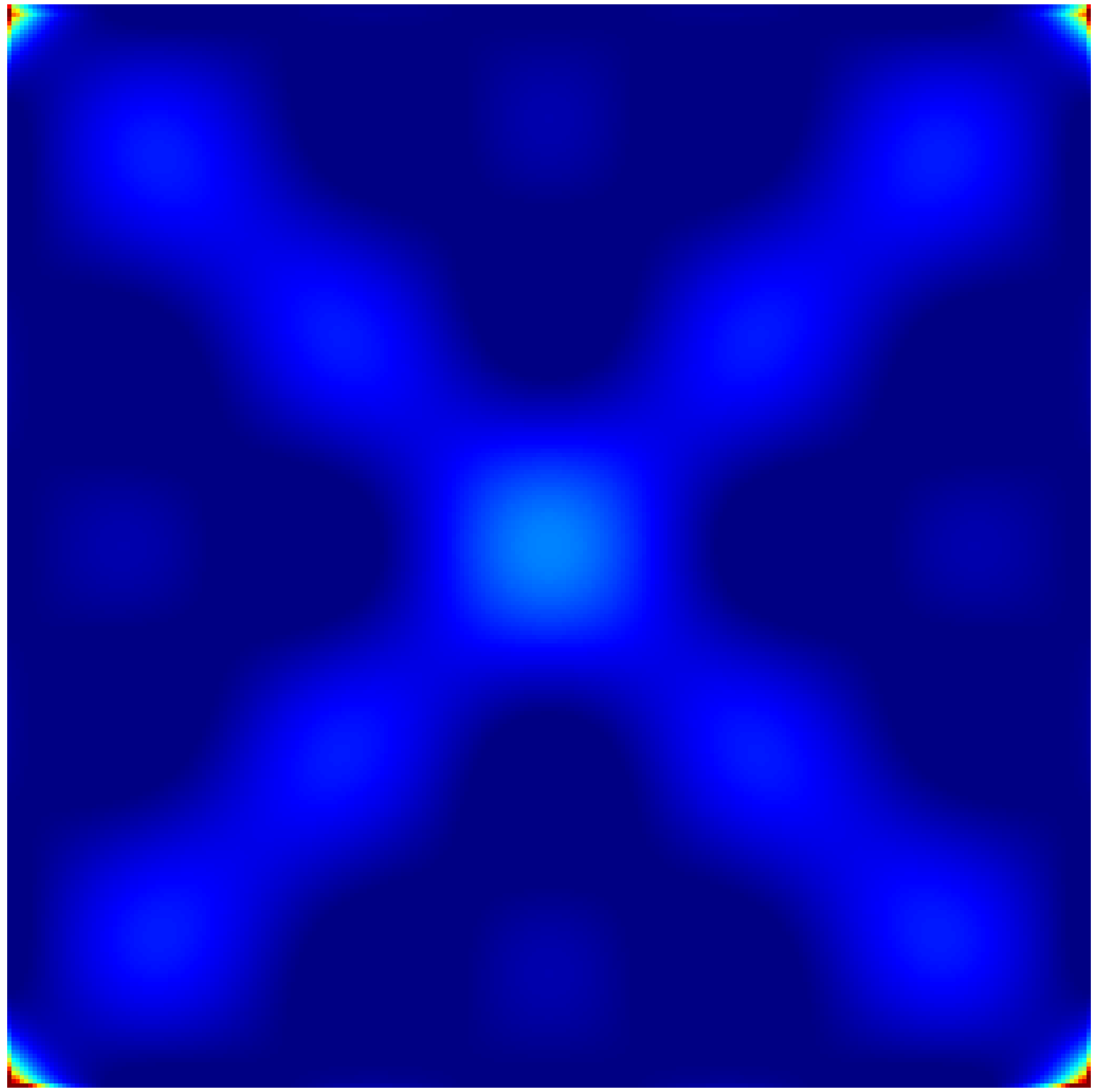
TUCKER Decomposition
with 2x2 core tensor

Parameterization

- But the second case is far more difficult:



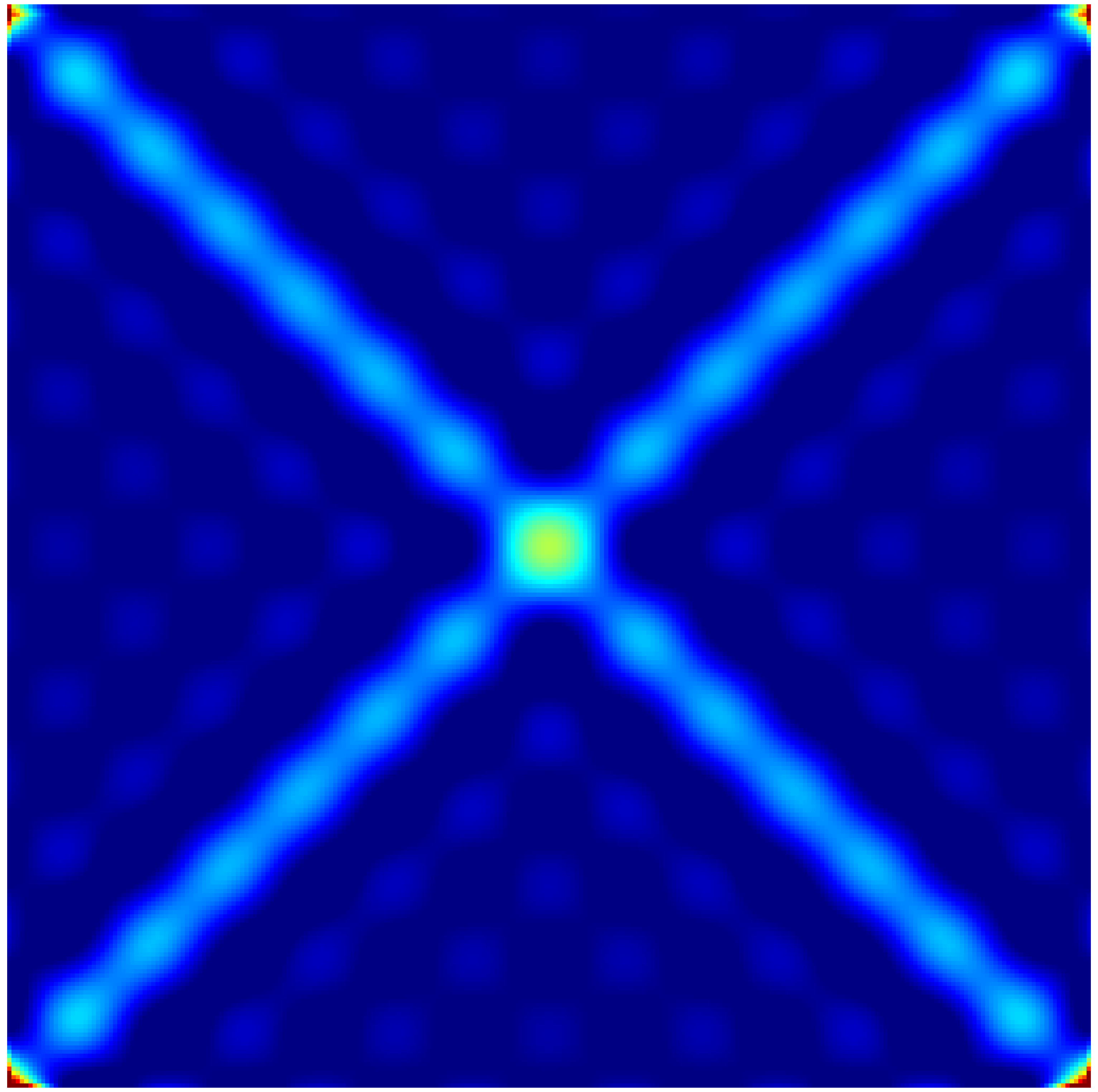
CP Decomposition
with 4 components



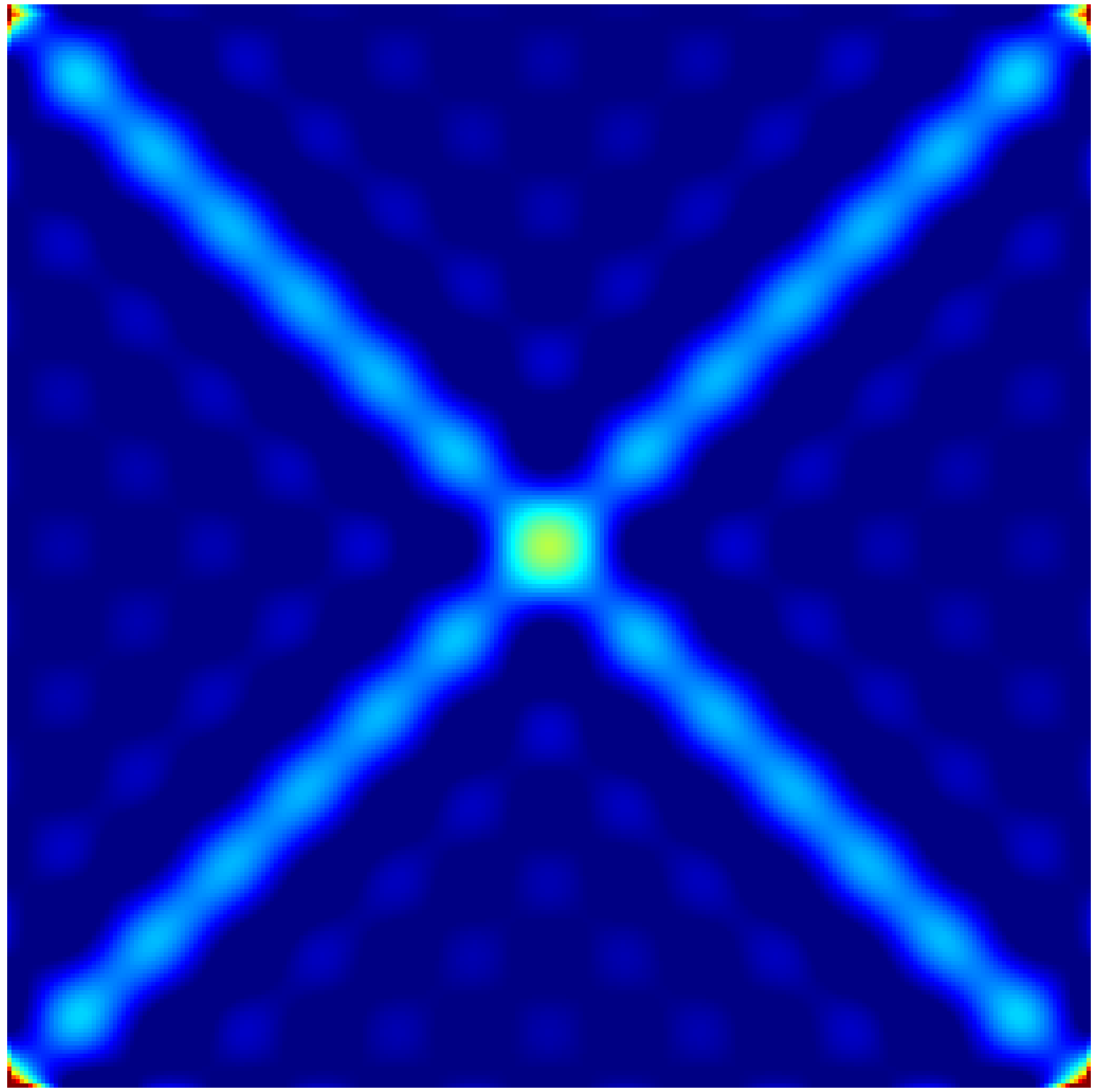
TUCKER Decomposition
with 4x4 core tensor

Parameterization

- But the second case is far more difficult:



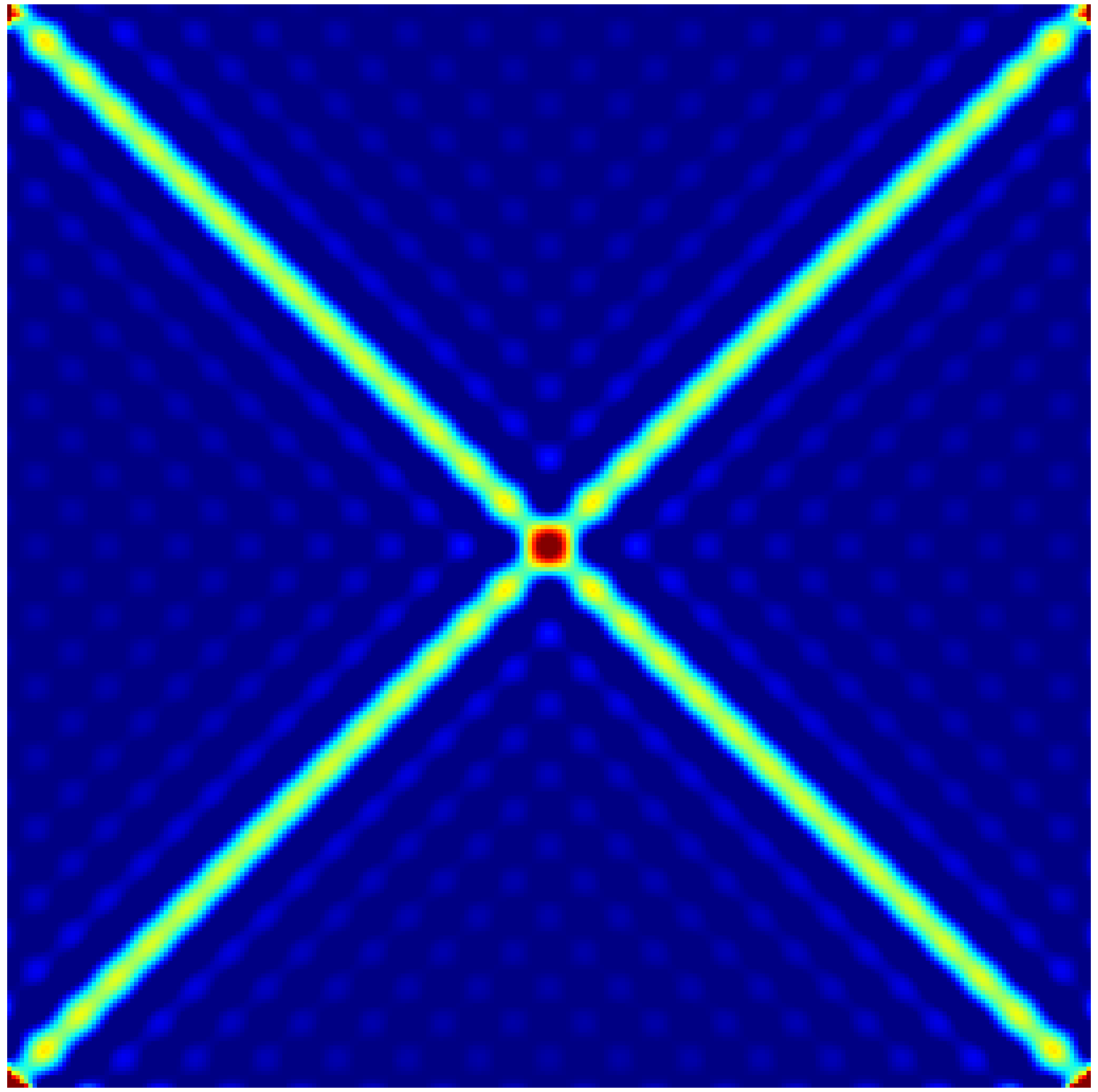
CP Decomposition
with 8 components



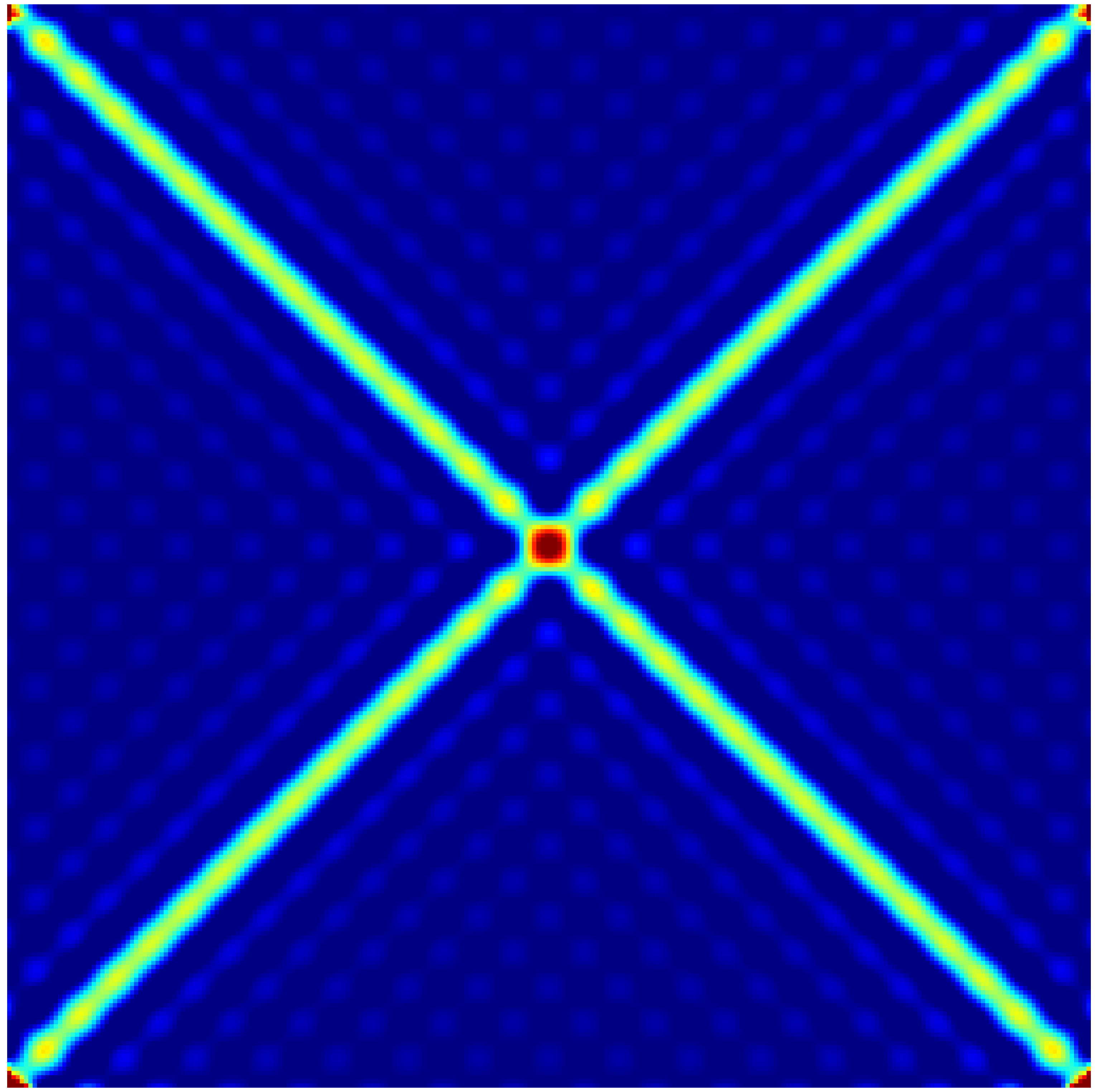
TUCKER Decomposition
with 8x8 core tensor

Parameterization

- But the second case is far more difficult:



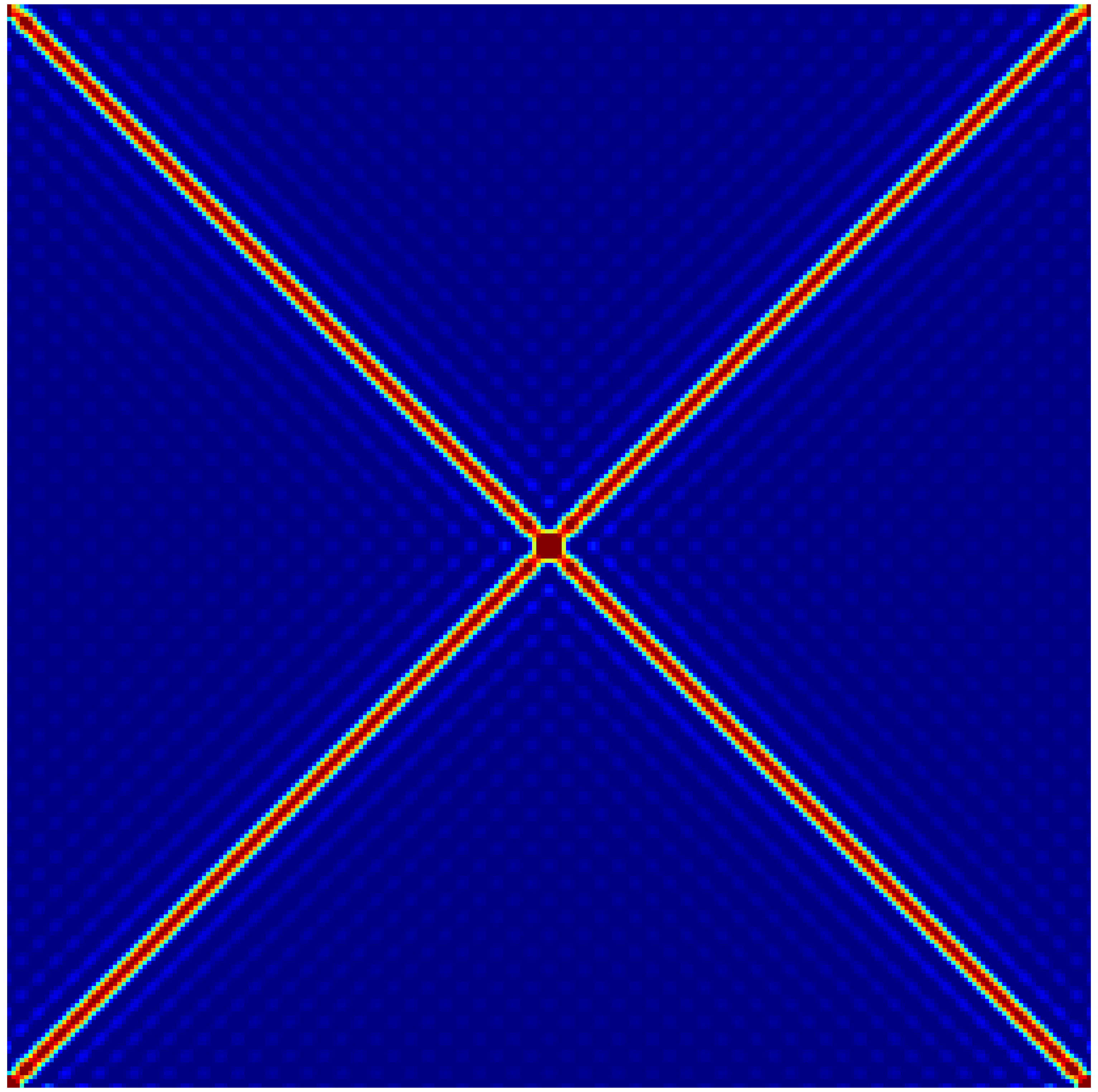
CP Decomposition
with 16 components



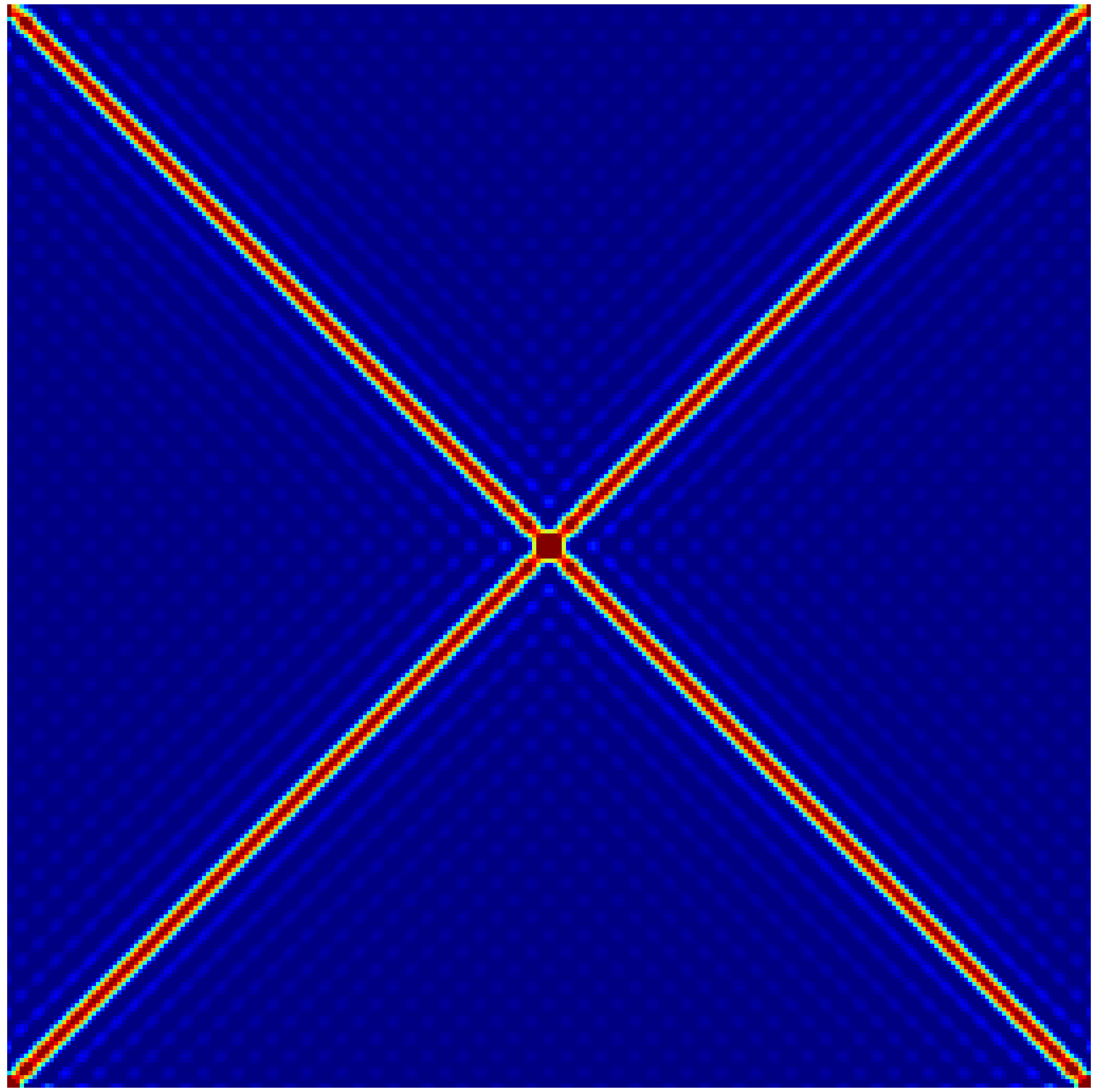
TUCKER Decomposition
with 16x16 core tensor

Parameterization

- But the second case is far more difficult:



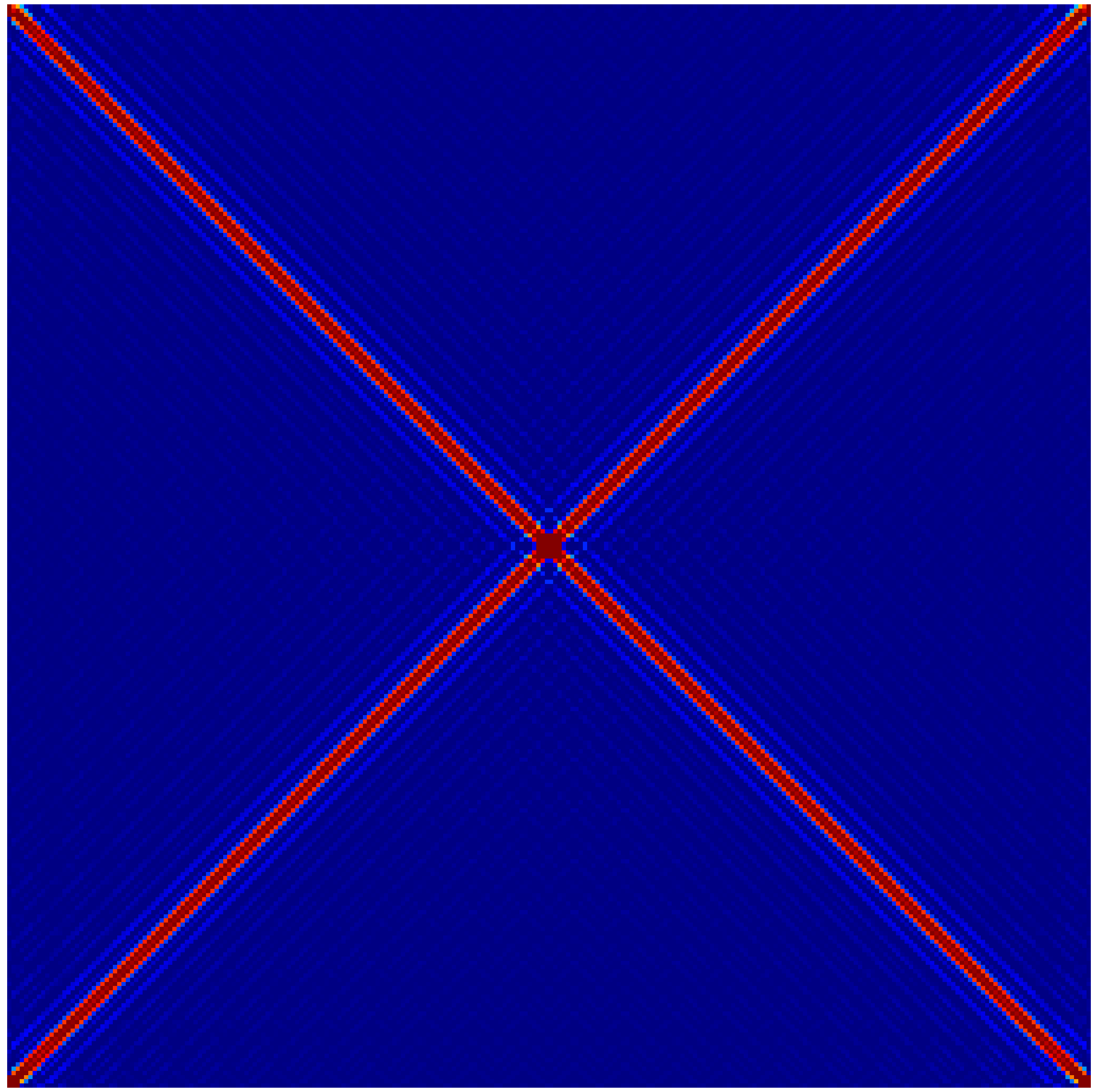
CP Decomposition
with 32 components



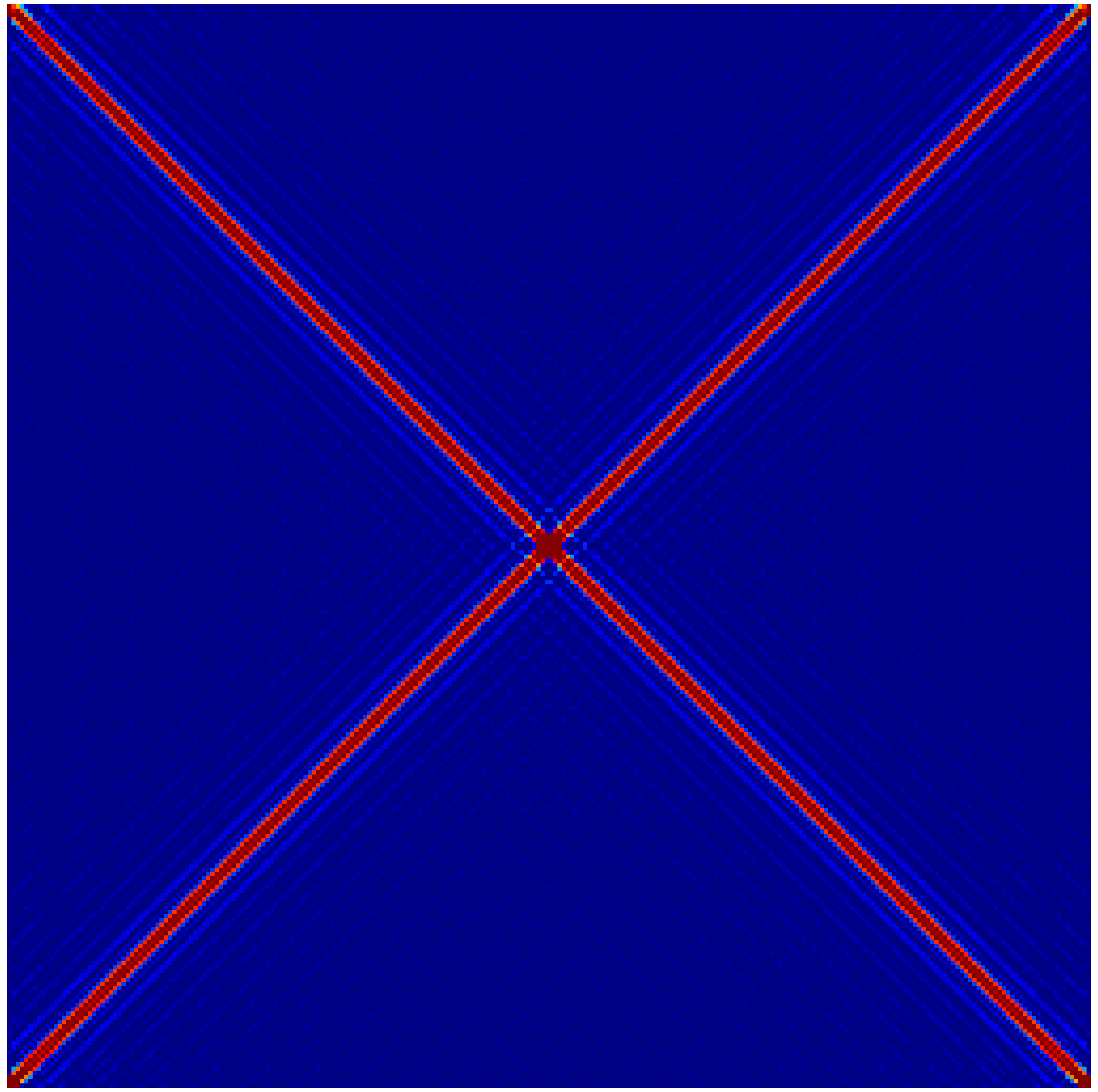
TUCKER Decomposition
with 32x32 core tensor

Parameterization

- But the second case is far more difficult:



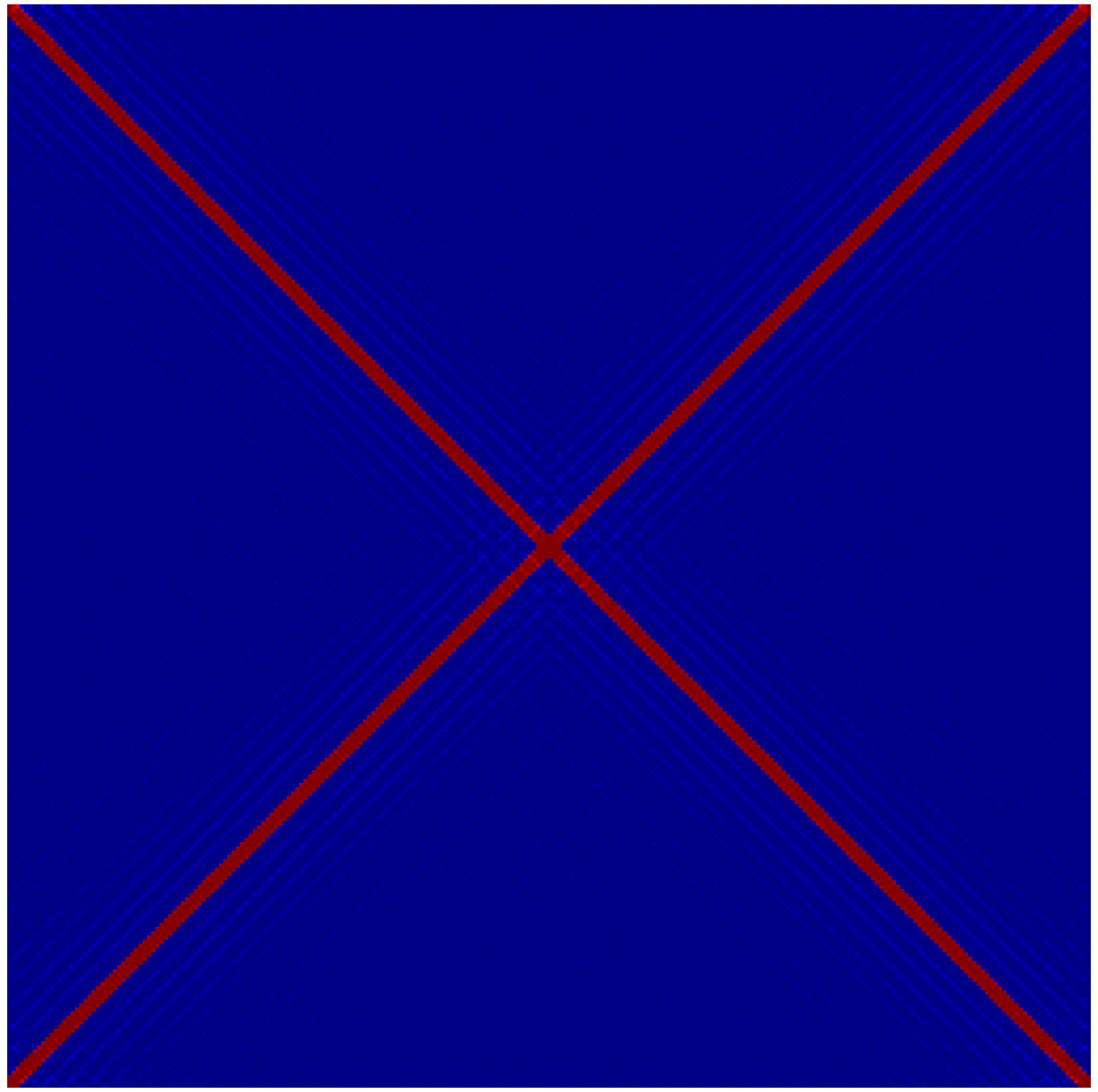
CP Decomposition
with 64 components



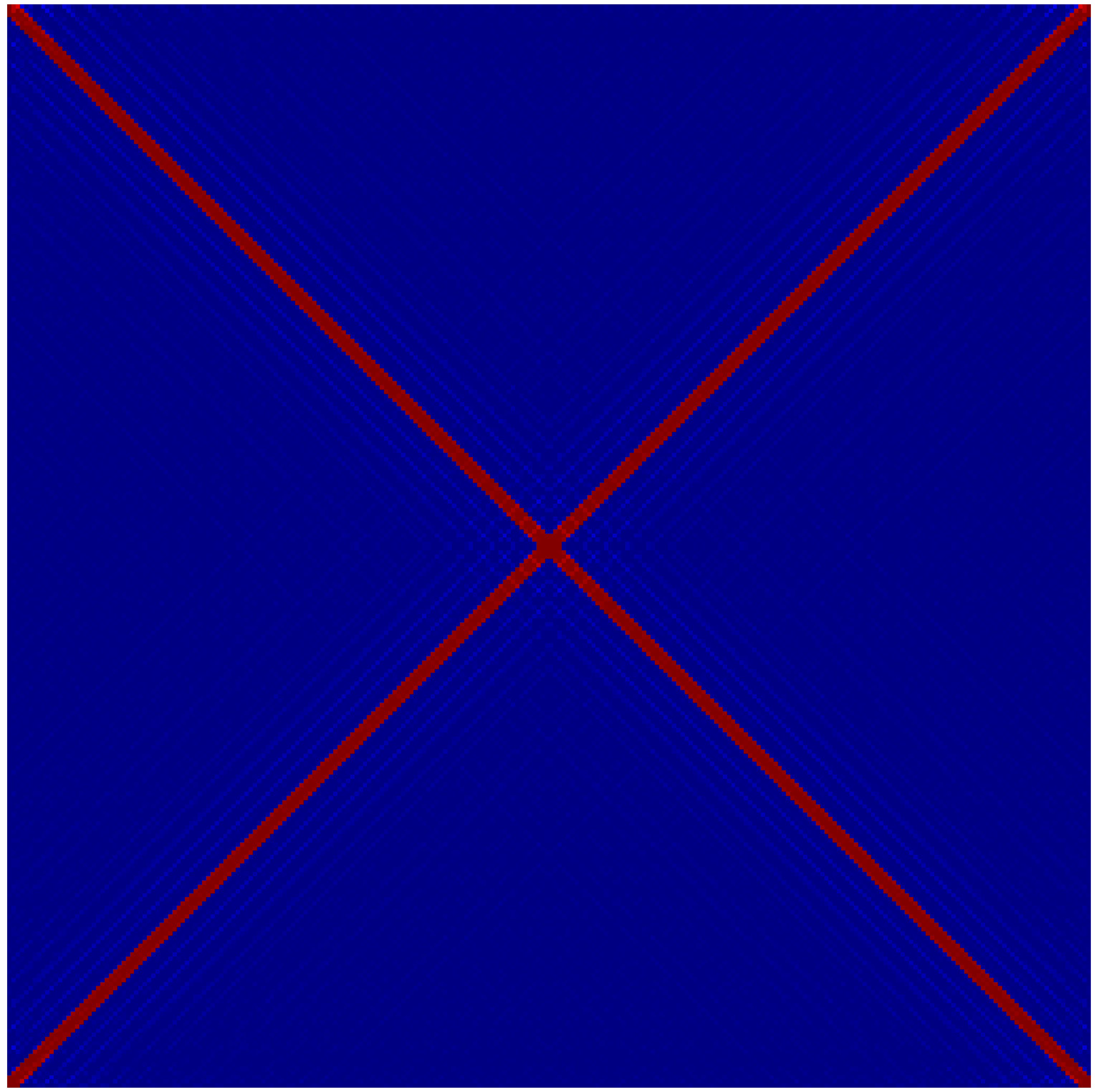
TUCKER Decomposition
with 64x64 core tensor

Parameterization

- But the second case is far more difficult:



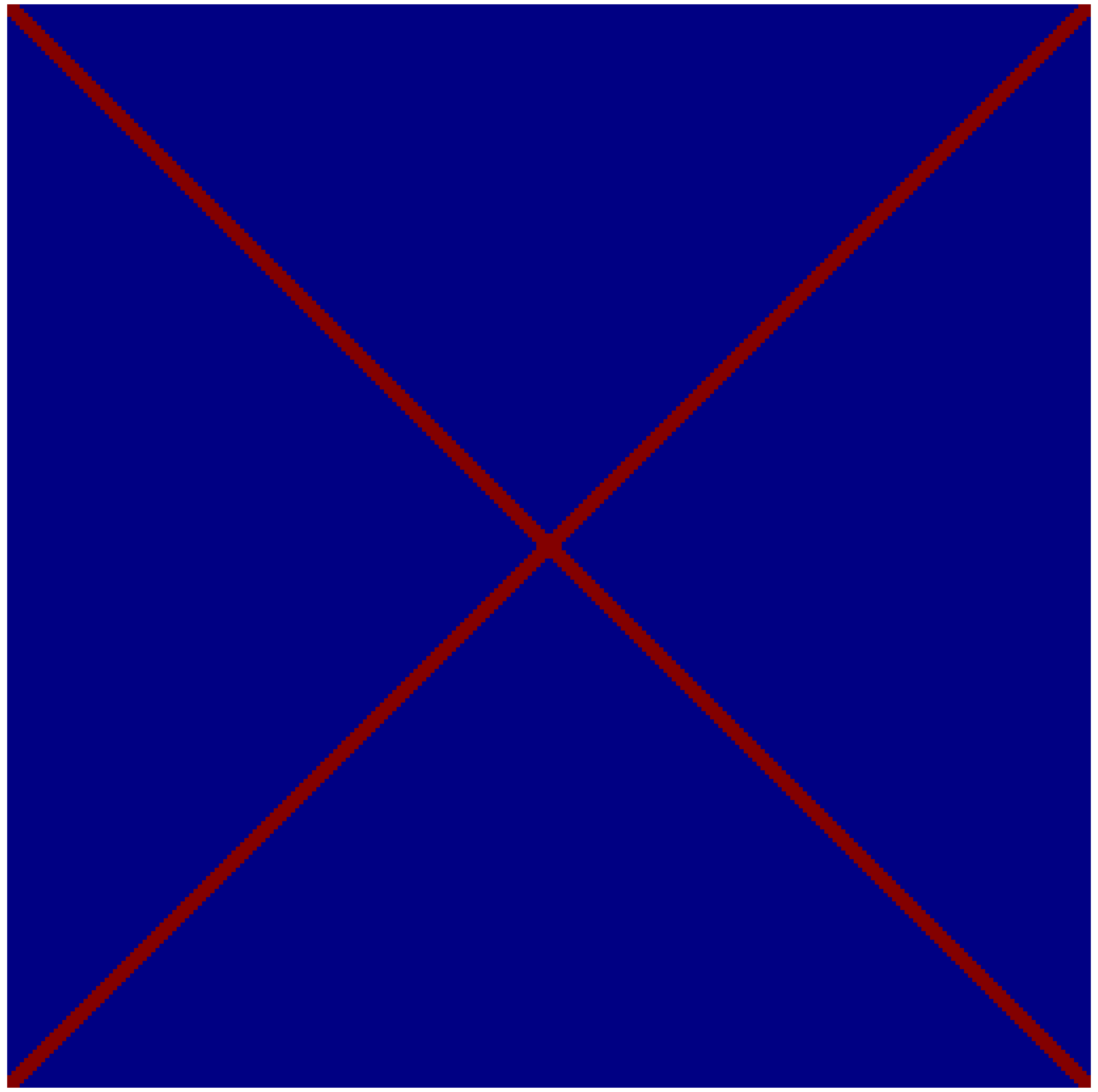
CP Decomposition
with 100 components



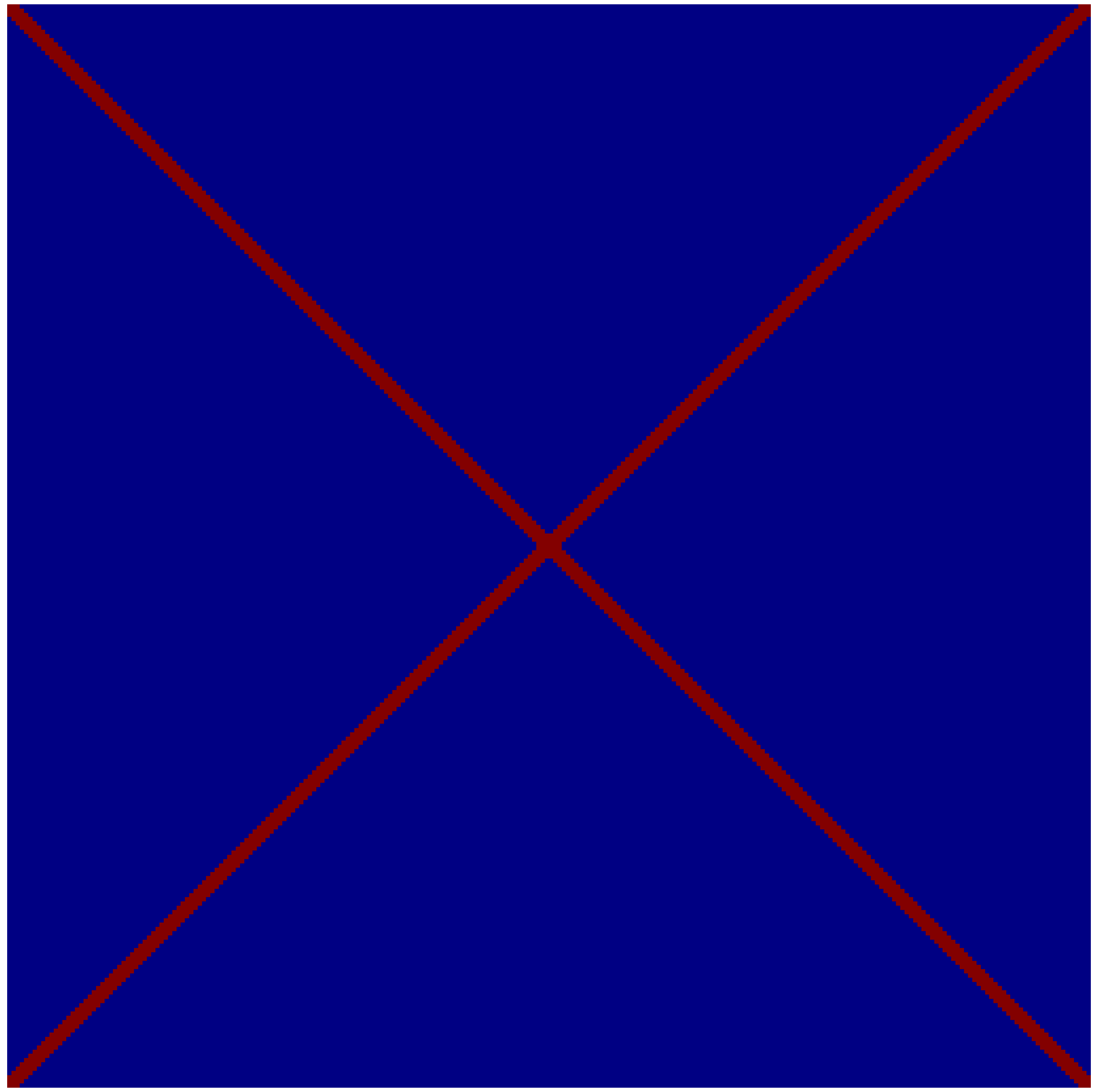
TUCKER Decomposition
with 100x100 core tensor

Parameterization

- But the second case is far more difficult:



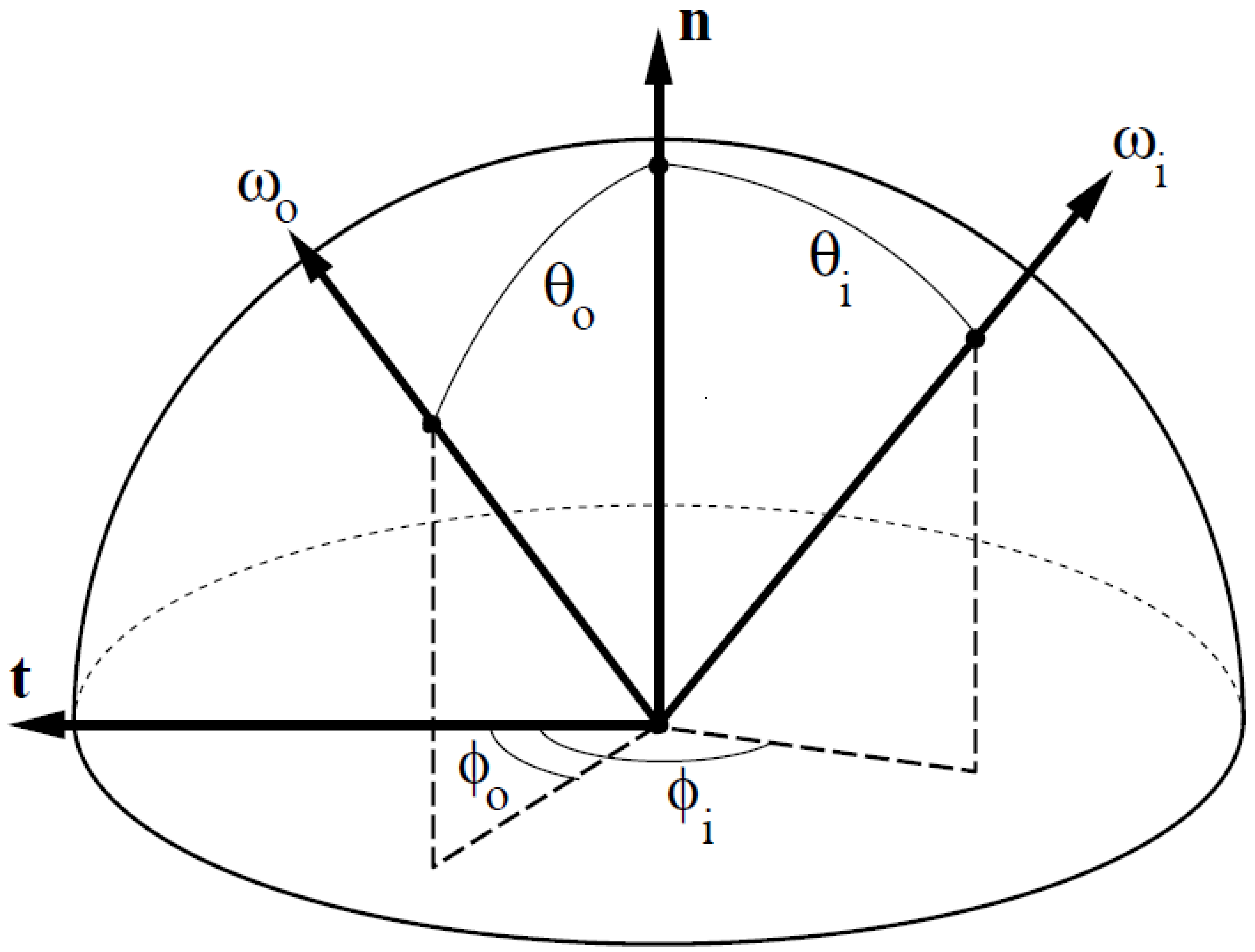
CP Decomposition
with 128 components



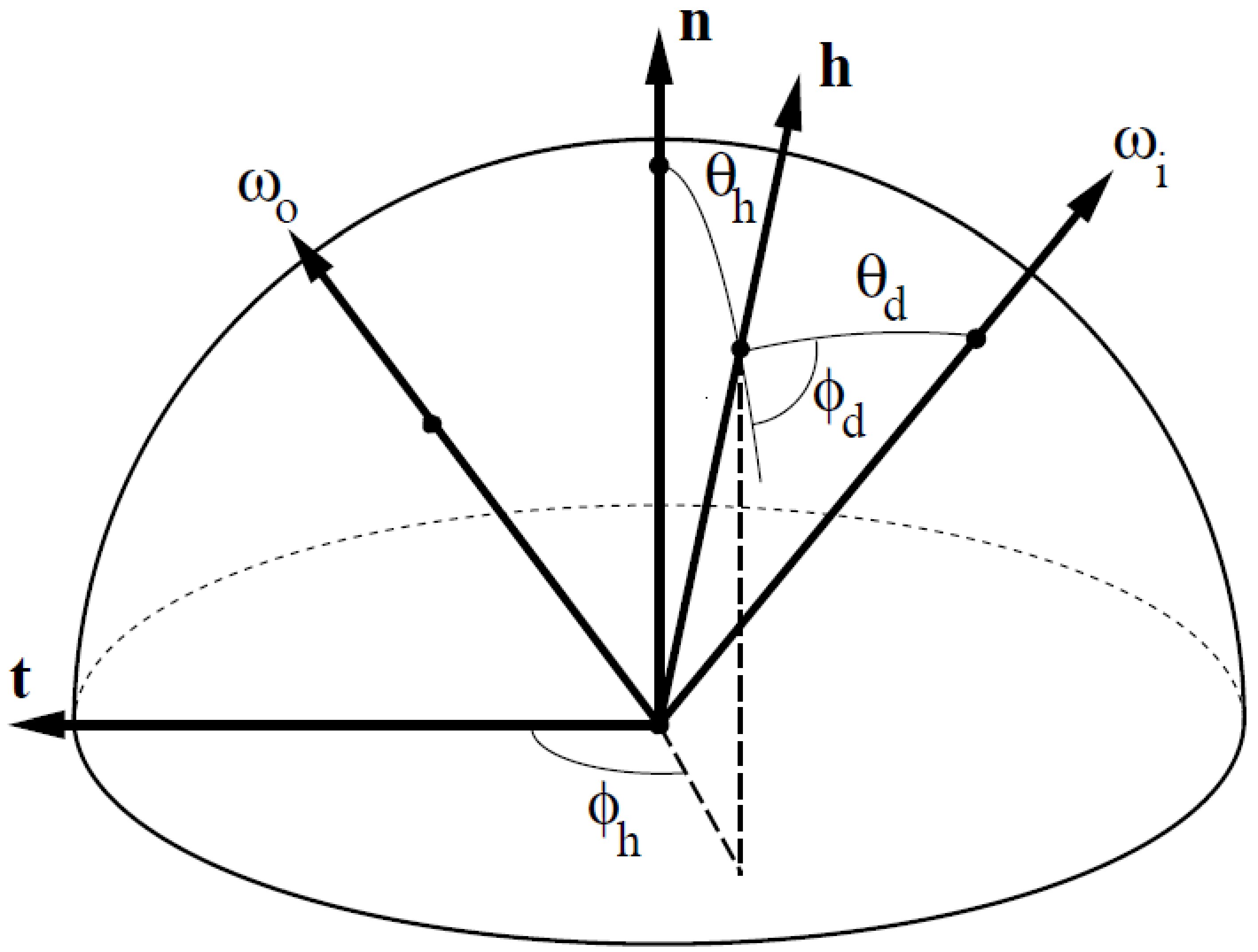
TUCKER Decomposition
with 128x128 core tensor

Half-Diff Parameterization

- Parameterization of BRDF via incoming and outgoing direction not well suited
  - Better alternative via a halfway and a difference vector has been proposed in [Rusinkiewicz-1998]



In/Out Parameterization



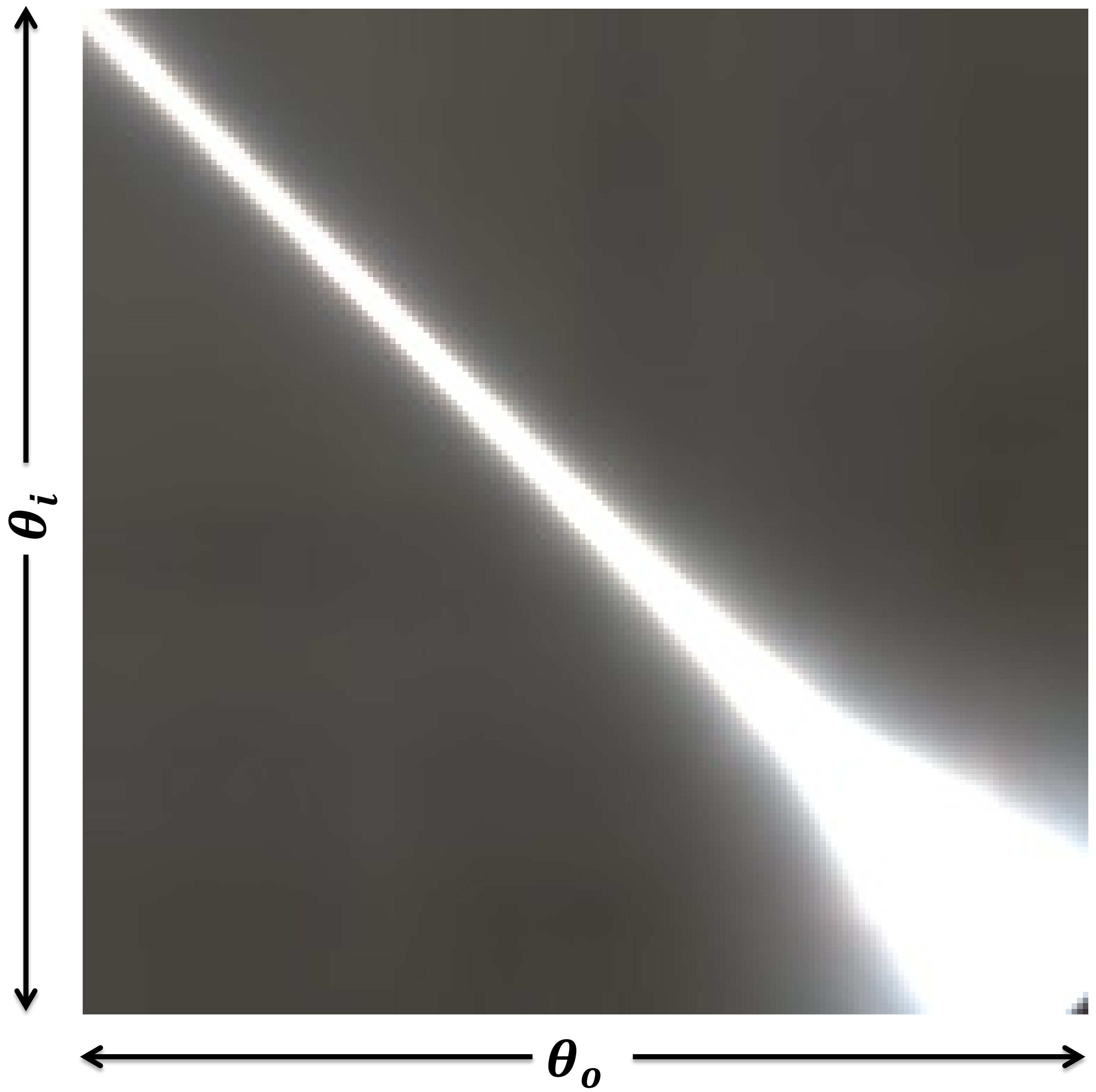
Half/Diff Parameterization

Image from [Rusinkiewicz-1998]

Half-Diff Parameterization

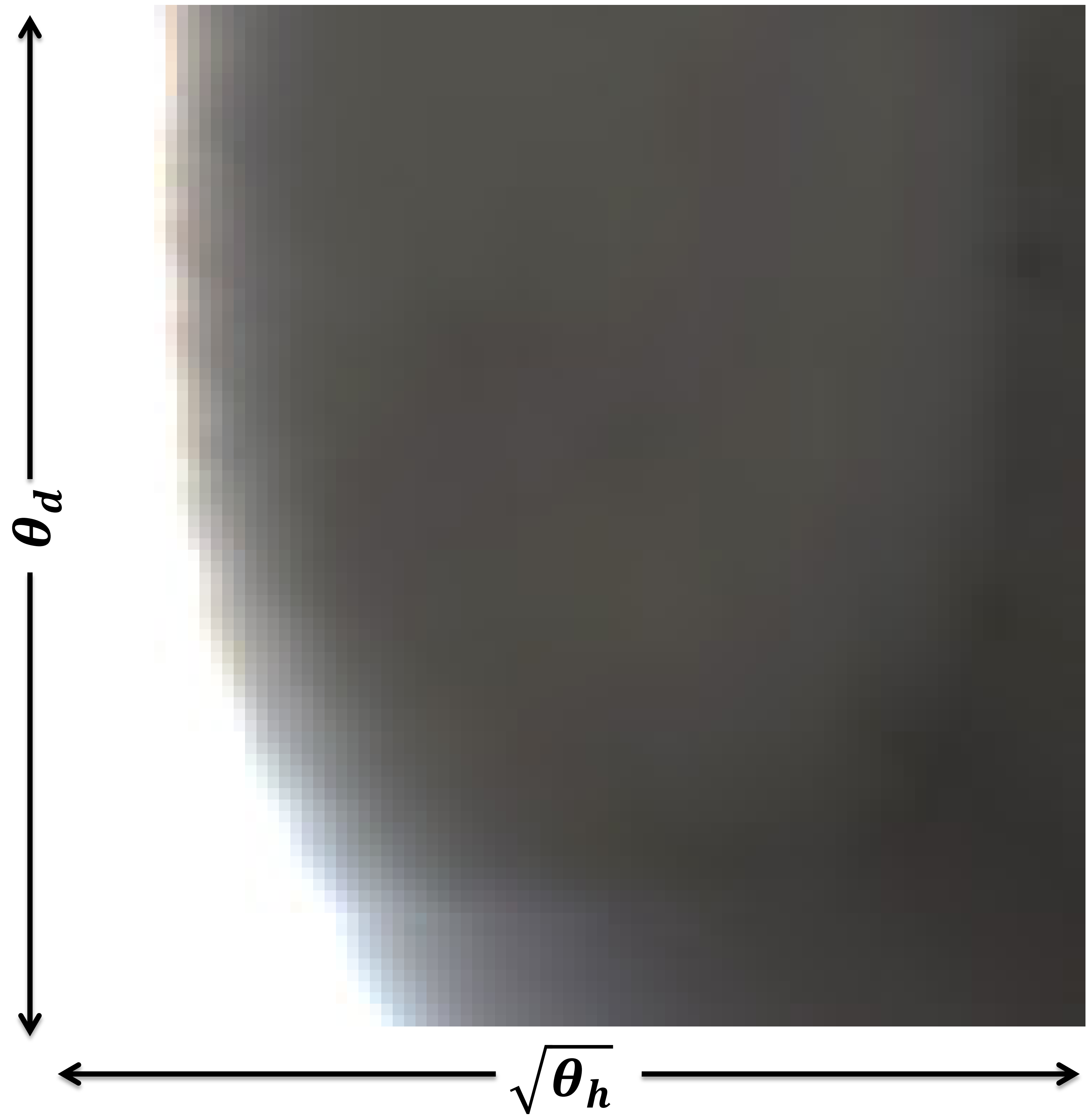
- Comparison of two slices through the Mode-3 tensor of an isotropic BRDF

In/Out Parameterization



$\varphi_0 = 180°$

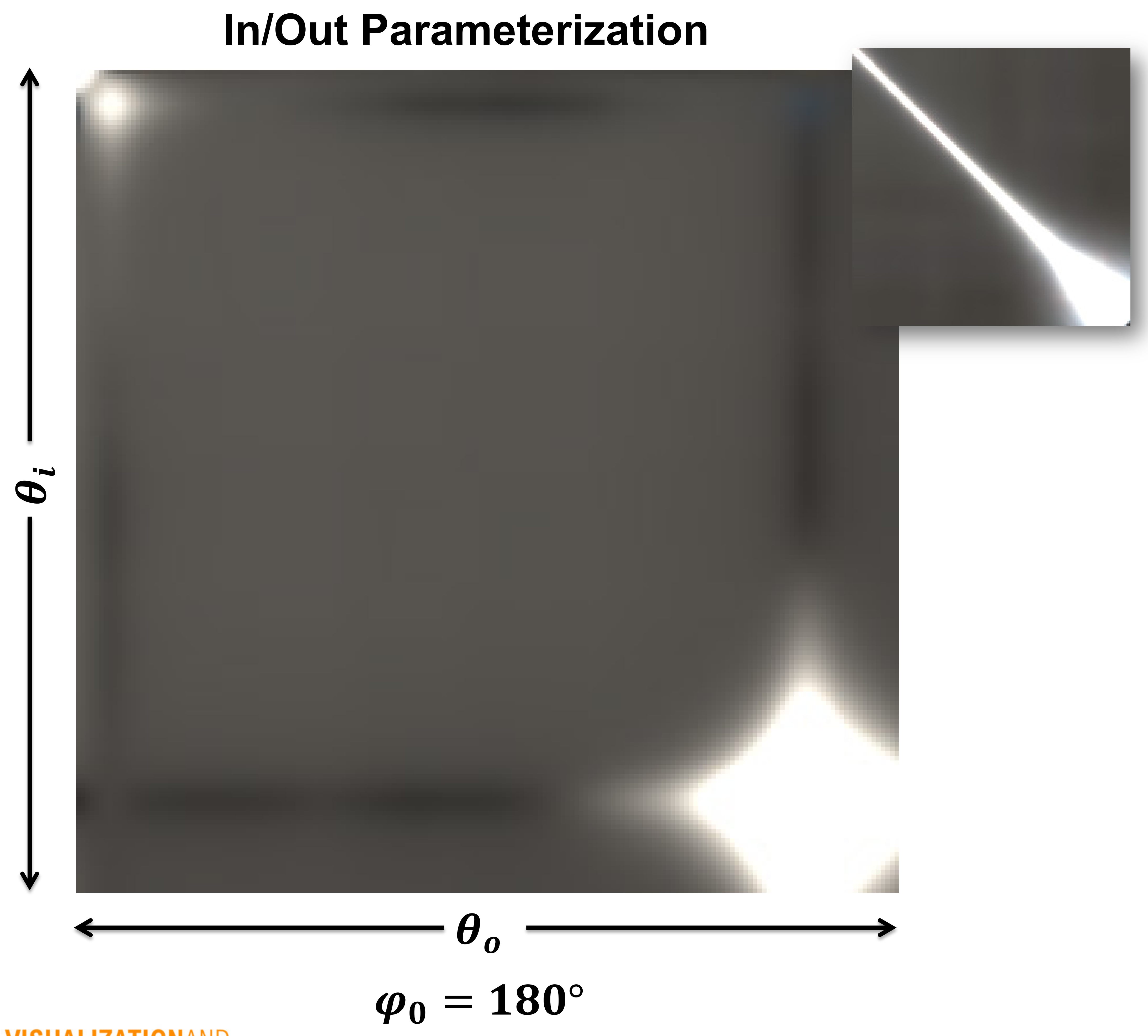
Half/Diff Parameterization

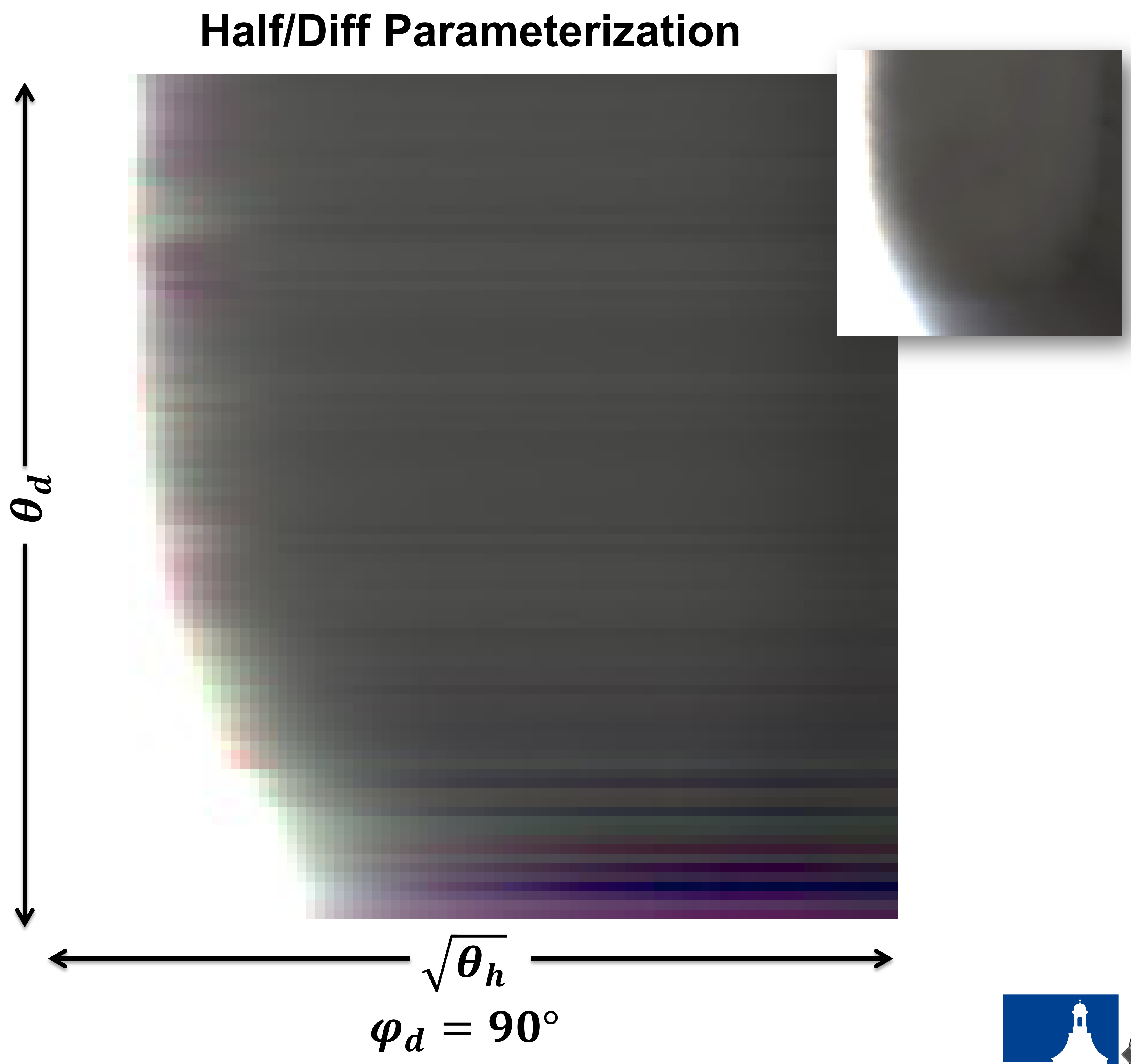


$\varphi_d = 90°$

Half-Diff Parameterization

- CP approximation of the tensor with 6 components





Parameterization

- The difference is also clearly visible in renderings:

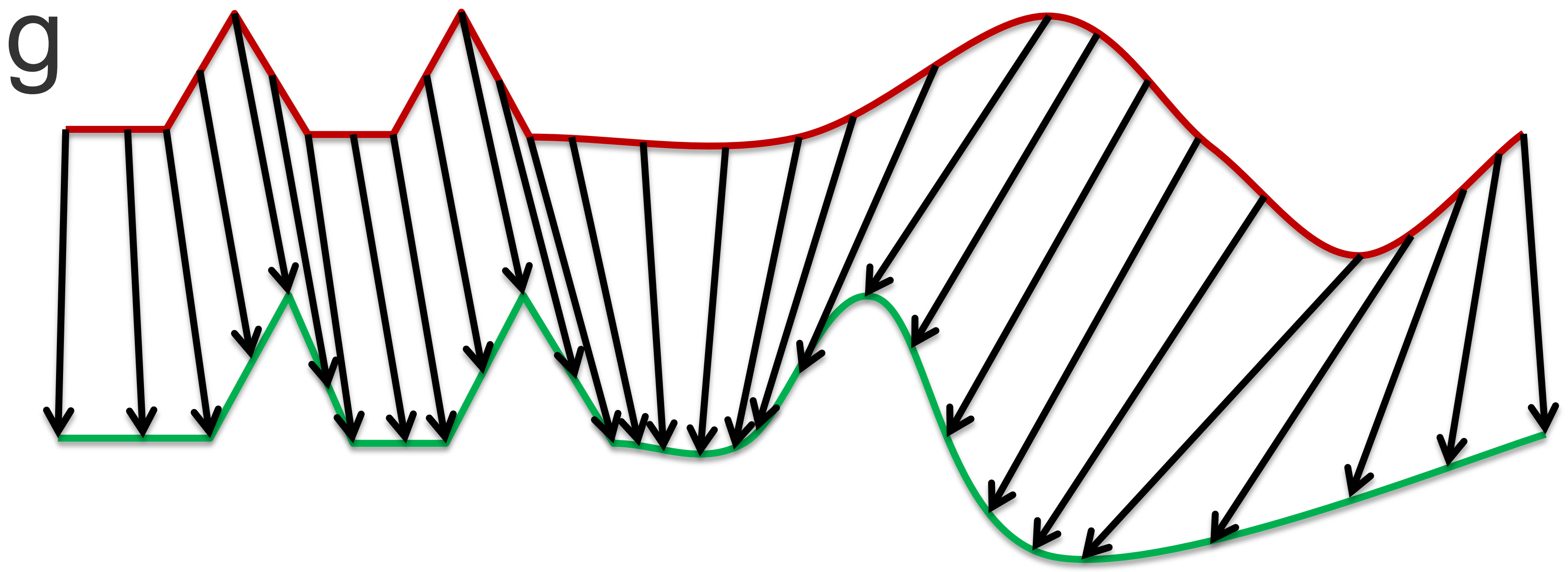
Uncompressed BRDF

In/Out Parameterization

Half/Diff Parameterization

Registration

- Correlations can only be exploited, if corresponding features are aligned with each other
  - The input data has to be registered correctly!
- Depending on the data-type different types of registration can be employed, e.g.

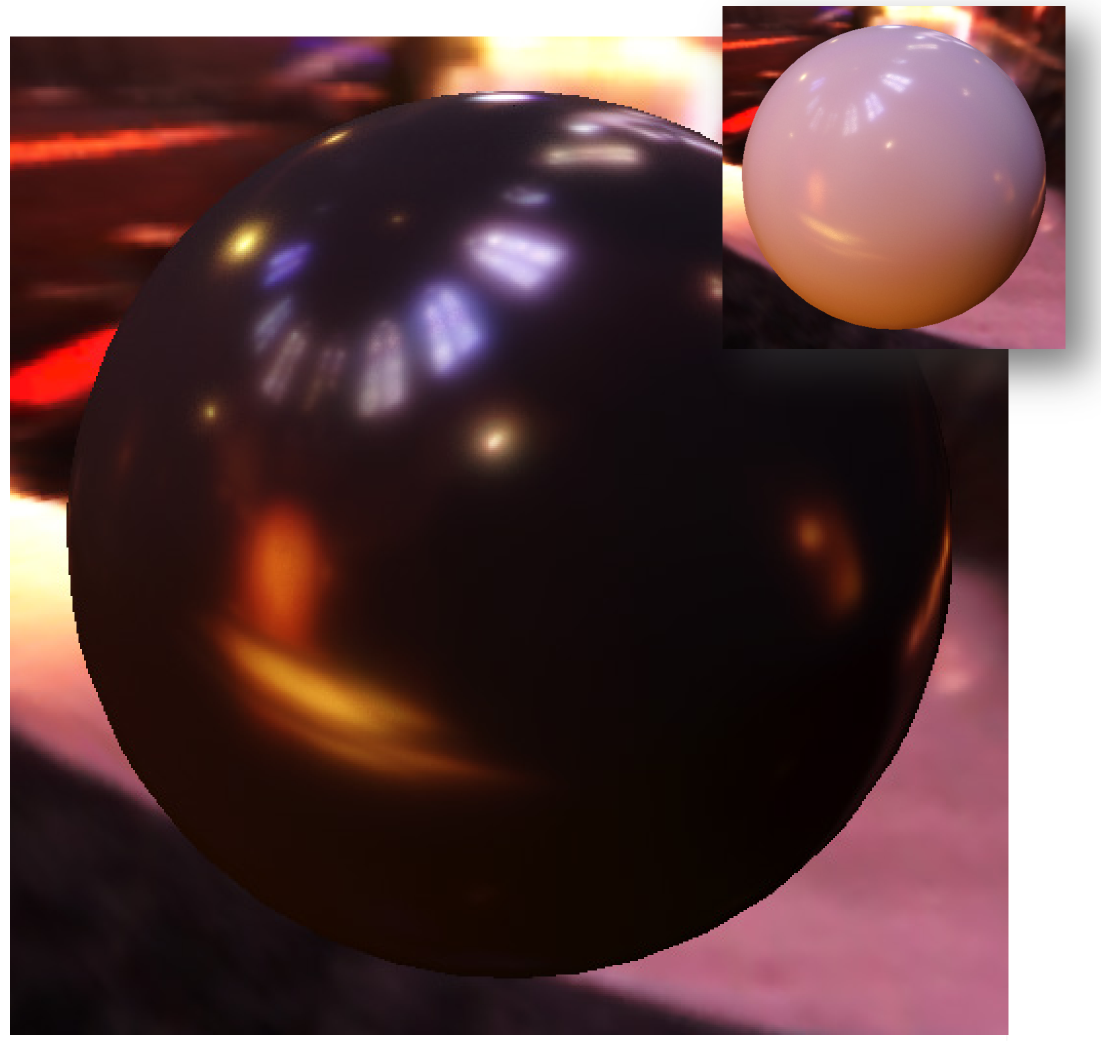


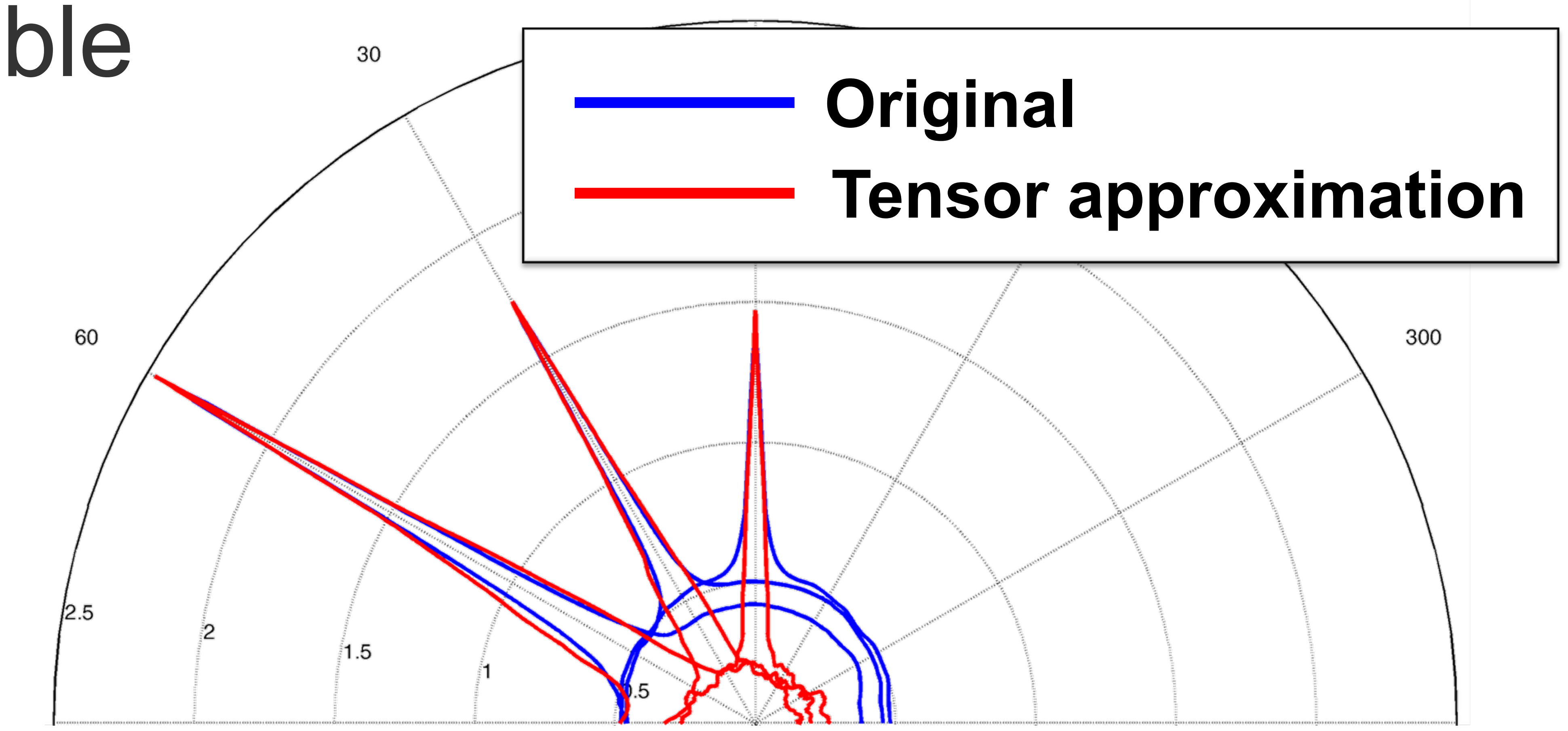
Registration of two functions via Dynamic Time Warping

Geometry	Rigid alignment, Non-Rigid alignment, Reparameterization of the surface
Motion Data	Dynamic Time warping
Images, Volumetric Data	Rigid registration, Warping
BTFs	Alignment of local coordinate systems, Good choice of reference plane, Parallax correction via reference geometry

Error Measure

- Some datasets have a very high dynamic range
  - Example: BRDFs can exhibit a dynamic range of 10,000:1
- Errors in parts with small values can still be perceptually relevant
  - Example: diffuse component of a BRDF
- In these cases the ℓ^2 error measure is not suitable





Fourth root was applied to the plot!

Dynamic Range Reduction

- Reduce dynamic range by applying transformation to the data prior to tensor decomposition
  - E.g. $\mathbf{log}(x)$ was used for BRDFs in [Bilgili-2011]
    - Other functions like **roots** or **sigmoid functions** could also be used
  - Has to be inverted after decompression
  - Decomposition is no longer linear
    - Can be a problem in applications, where a linear decomposition is needed
      - For example, in [Sun-2007], the Tucker Decomposition is used to create a linear basis for BRDFs

Fourth root was applied to the plots!

Relative Error via Per-Element Weights

- Employ a different error metric during the optimization
  - Only ℓ^2 errors can be minimized efficiently via ALS
  - **Per-element** weights w can be included into the approximation
    - Can be used to minimize relative errors:

(x original value, $\tilde{x}$ approximation)

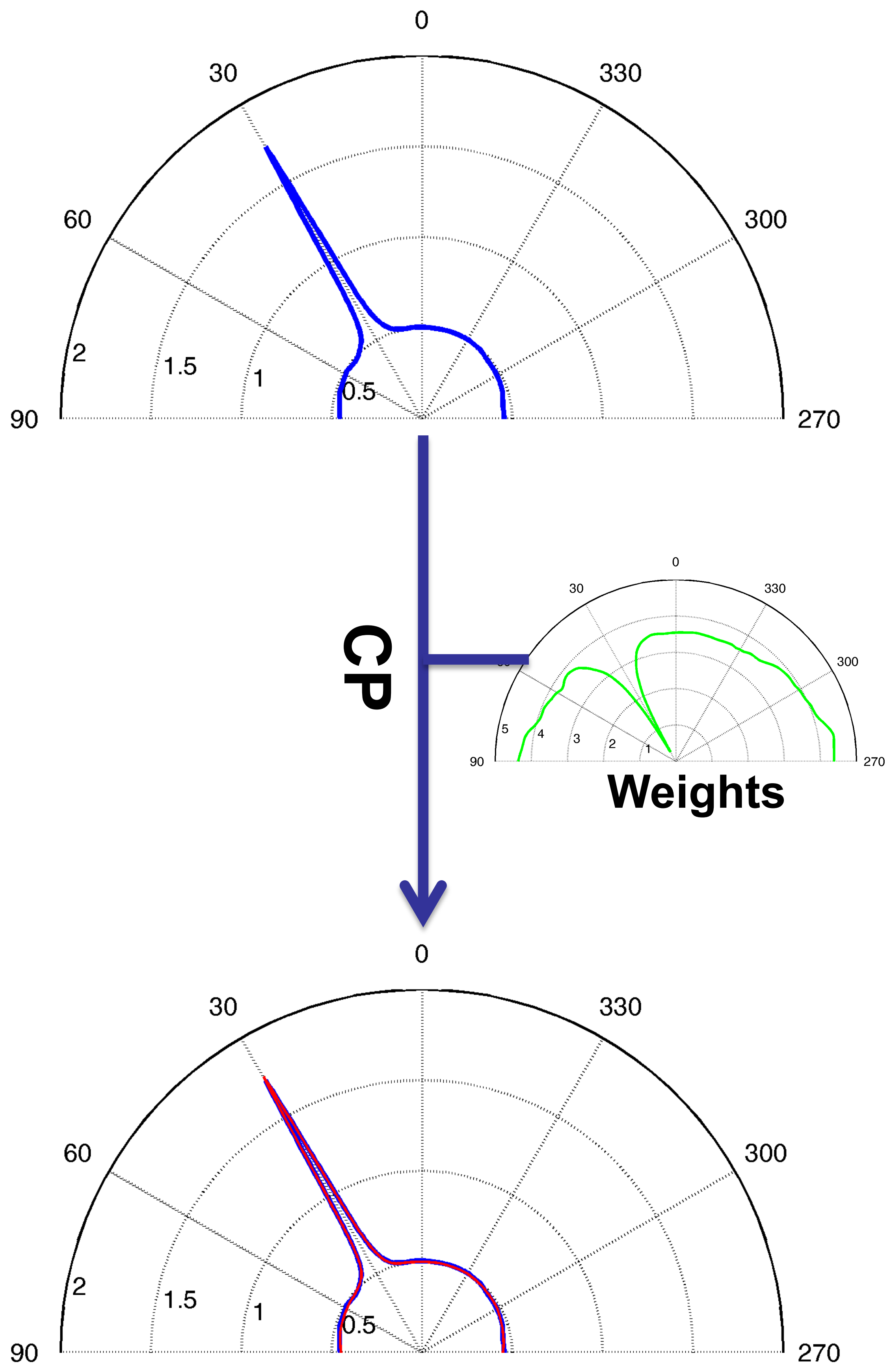
Squared error relative to original value

$$\frac{|x-\tilde{x}|^2}{|x|} = w|x - \tilde{x}|^2 \quad \text{with } w = \frac{1}{|x|}$$

Square of the relative error

$$\frac{|x-\tilde{x}|^2}{|x|^2} = w|x - \tilde{x}|^2 \quad \text{with } w = \frac{1}{|x|^2}$$

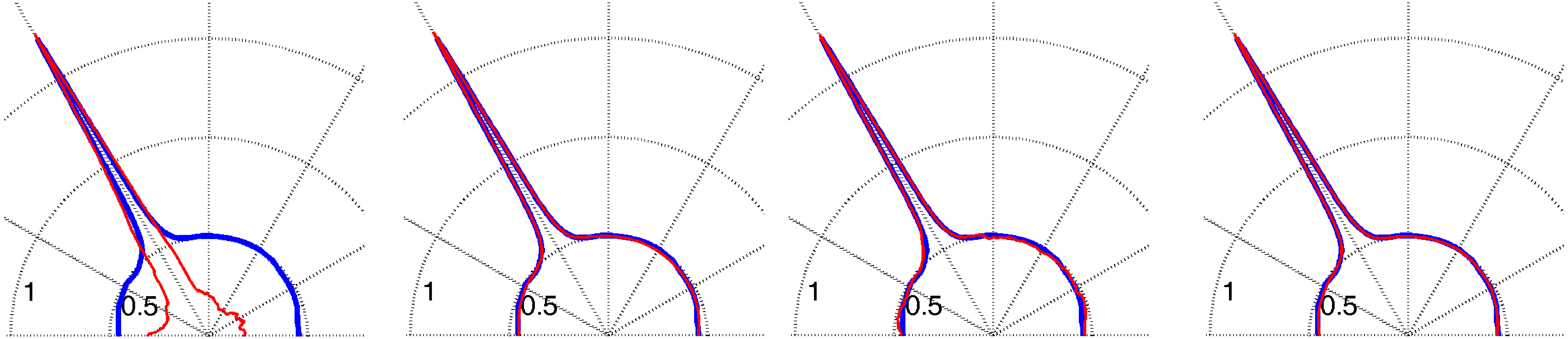
  - Decomposition remains linear and no inversion is necessary after decompression
  - Additional weights can be used to compensate for the irregular sampling, cosine θ_i fall-off, reliability of the input data etc.



Fourth root was applied to the plots!

Error Measure (comparison)

Original	ℓ^2 Error	Log error	Squared Error relative to original value	Square of the relative error
	$\|x - \tilde{x}\|^2$	$\|\log(x) - \log(\tilde{x})\|^2$	$\frac{\|x - \tilde{x}\|^2}{\|x\|}$	$\frac{\|x - \tilde{x}\|^2}{\|x\|^2}$

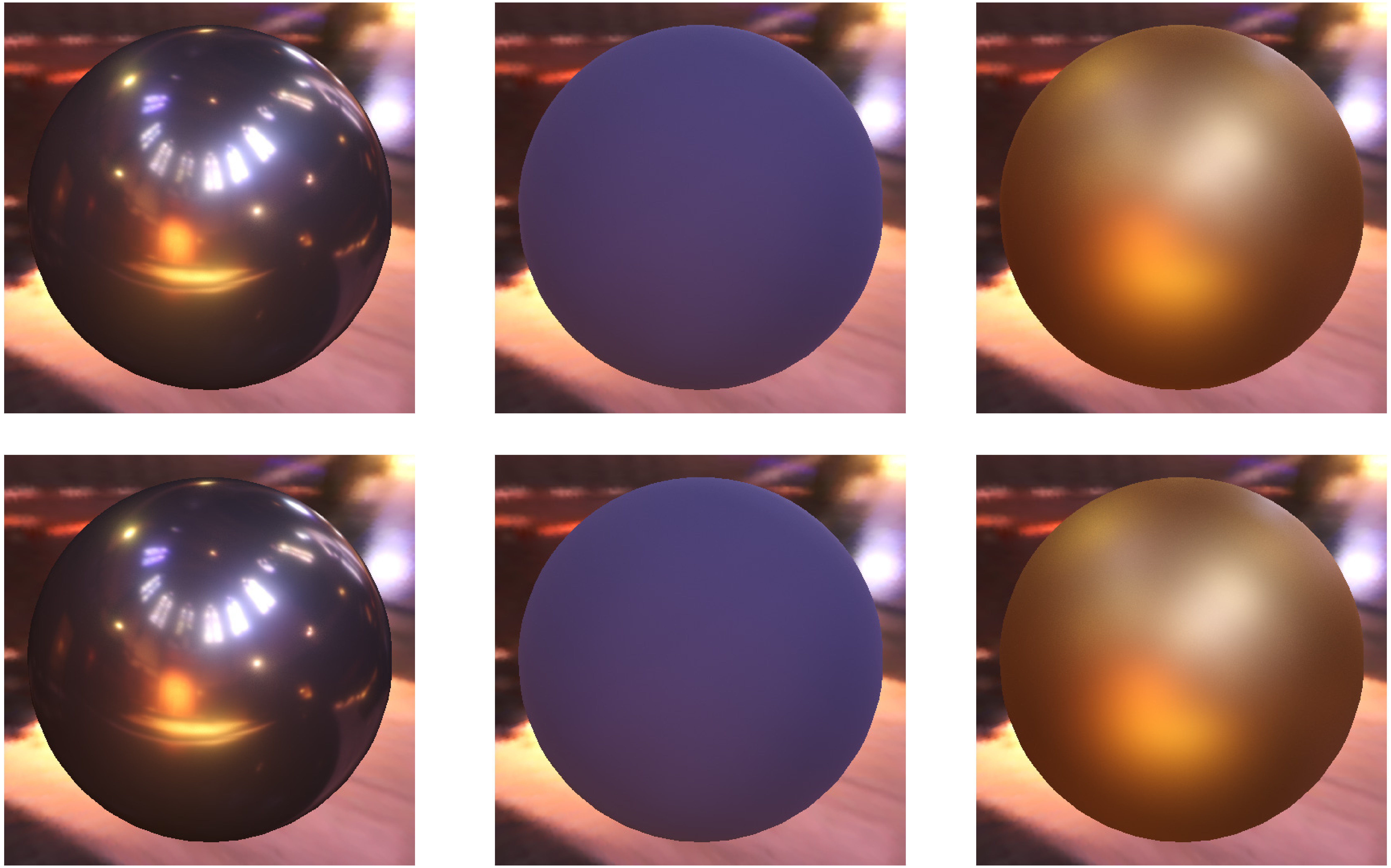


Fourth root was applied to the plots!

BRDF Compression Results

Uncompressed

Compressed



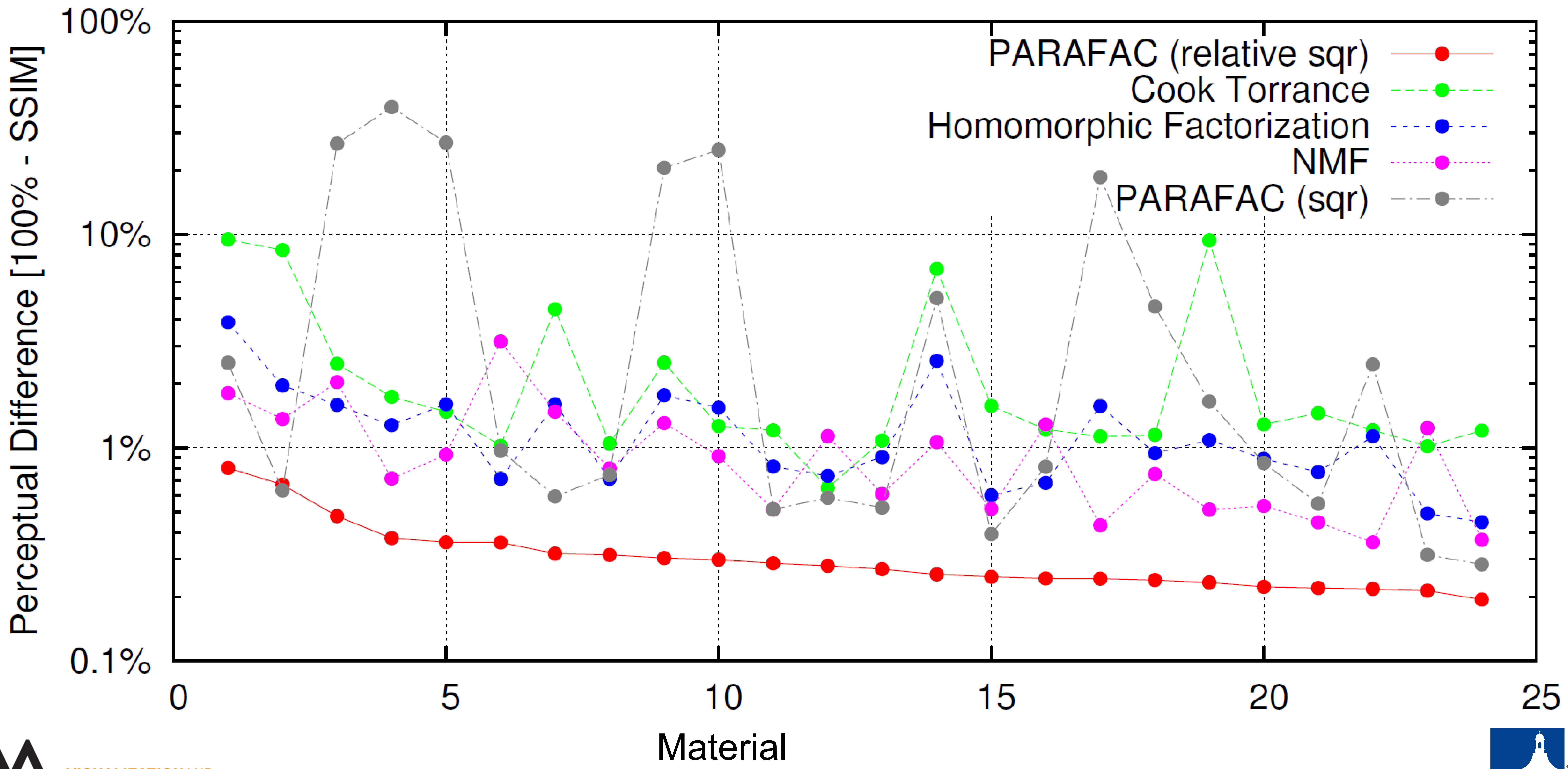
CP Compression

Components:	**8**
Original:	**33 MB**
Compressed:	**23 KB**
Ratio:	≈**1500:1**
E. Measure:	$\frac{\lvert x - \tilde{x}\rvert^2}{\lvert x\rvert}$

Additional weights to compensate for irregular sampling and for $\cos\theta_i$ and $\cos\theta_o$

Results from [Ruiters-2010]

BRDF Compression Results



Results from [Ruiters-2010]

Which Decomposition to use?

Tucker Decomposition

- Potentially better compression ratios
  - Only when the core tensor is small and not too sparse
    - Size of core tensor increases as the product of the reduced ranks
  - Flexibility: user can choose the rank for each mode individually

- Random access very expensive for large core-tensors
  - Summation over all entries of the core tensor necessary:

$$\mathcal{T}_{i_1,\ldots,i_n} = (\boldsymbol{\mathcal{C}} \times_1 \boldsymbol{U}^{(1)} \times_2 \cdots \times_n \boldsymbol{U}^{(n)})_{i_1,\ldots,i_n} = \sum_{j_1} U^{(1)}_{i_1 j_1} \sum_{j_2} U^{(2)}_{i_2 j_2} \cdots \sum_{j_n} U^{(n)}_{i_n j_n} \mathcal{C}_{j_1,\ldots,j_n}$$

23% of storage for core tensor ($6 \times 6 \times 6 \times 3$)

99% of storage for core tensor ($28 \times 28 \times 128 \times 128$)

Which Decomposition to use?

CANDECOMP/PARAFAC Decomposition

- Sparse core tensor: diagonal structure
- More columns in the factor matrices needed
- Random access usually less expensive:

$$\mathcal{T}_{i_1,\ldots,i_n} = \left(\sum_{j=1}^{C} \sigma_j \circ \boldsymbol{v}_j^{(1)} \circ \cdots \circ \boldsymbol{v}_j^{(n)}\right)_{i_1,\ldots,i_n} = \sum_{j=1}^{C} \sigma_j v_{i_1,j}^{(1)} \cdots v_{i_n,j}^{(n)}$$

Which Decomposition to Use?

Alternatives

- Hierarchical Tensor Approximation
  - Possibly faster decompression
  - More compact compression for data with multi-resolution decomposition

- Clustered Tensor Approximation / Sparse Tensor Decomposition
  - Reduction of decompression cost via clustering
  - More compact when the underlying data can be clustered well
  - See: next part

How many modes to use?

- Tensor decompositions can be considered as factorization of a high dimensional function into a sum of products of one-dimensional functions:

PARAFAC $$f(x_1, \dots, x_n) = \sum_{i=1}^{C} f_i^1(x_1) f_i^2(x_2) \cdots f_i^n(x_n)$$

Tucker $$f(x_1, \dots, x_n) = \sum_{i_1=1}^{C_1} \cdots \sum_{i_n=1}^{C_n} \mathcal{C}_{i_1,\dots,i_n} f_{i_1}^1(x_1) f_{i_2}^2(x_2) \cdots f_{i_n}^n(x_n)$$

- One can instead factorize into higher-dimensional functions, e.g.

$$f(x_1, x_2, x_3, x_4, x_5) = \sum_{i=1}^{C} f_i^1(x_1, x_2) f_i^2(x_3)\, f_i^3(x_4, x_5)$$

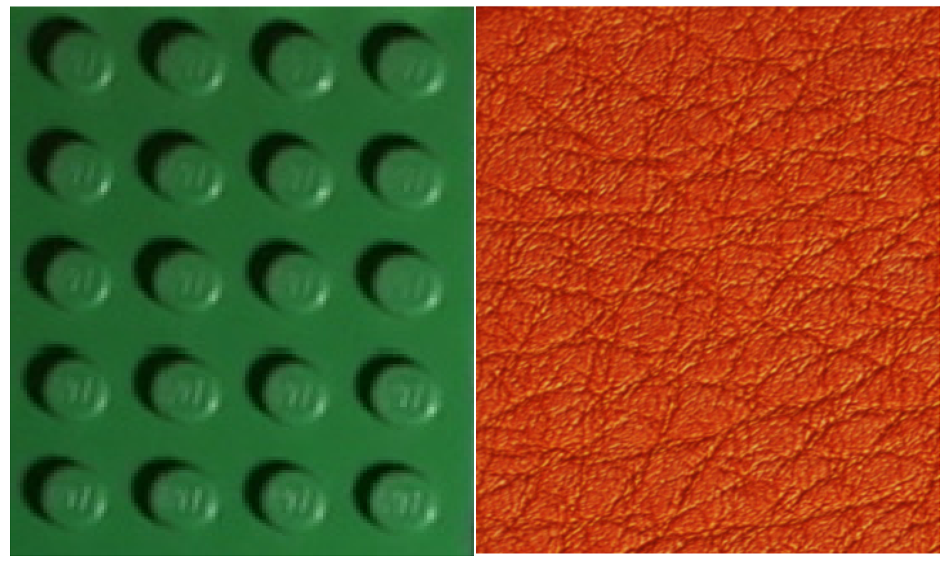
- This is done by *"unfolding"* several dimensions into one mode of the tensor

VISUALIZATIONAND MULTIMEDIALAB

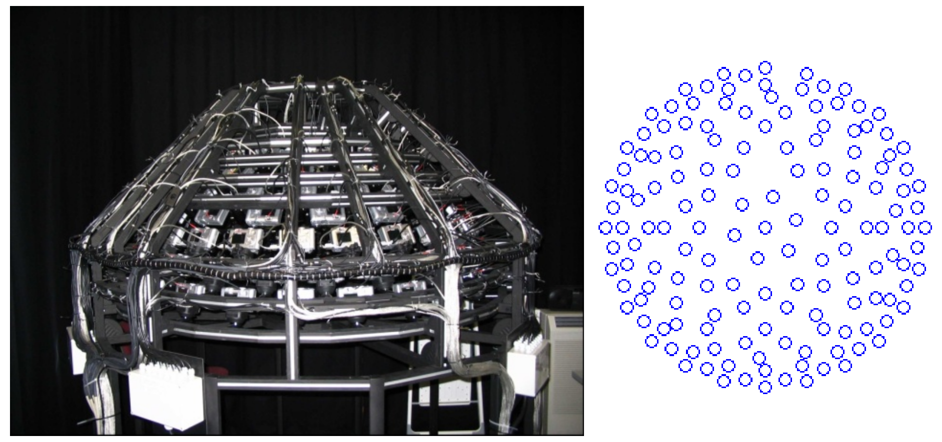
universitätbonn

How many modes to use?

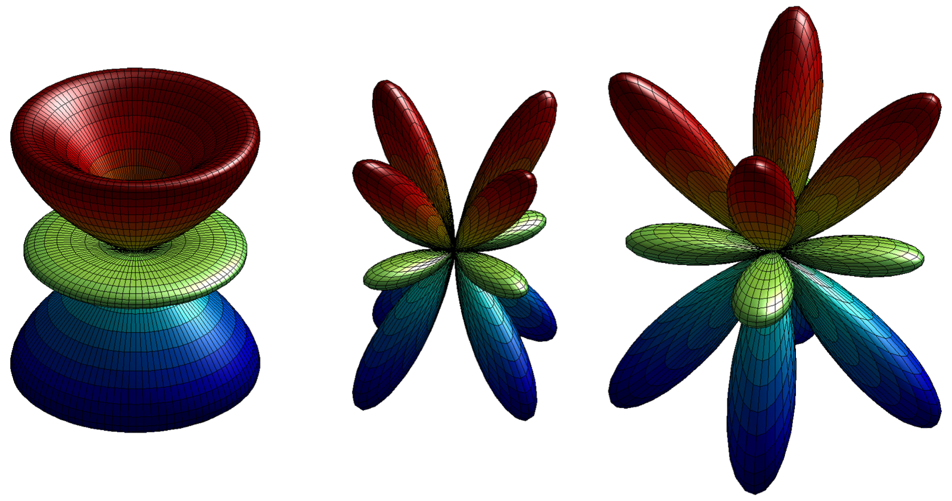
- Sometimes it is not advisable to represent all *"natural"* dimensions of the input dataset as modes
  - The dimensions exhibit a high complexity, which cannot be factorized well
    - No significant gain in compression ratio
    - A large number of components would be needed to encode the complexity
      - Slow decompression
      - [Wang-2005] and [Tsai-2012] decompress spatial compression prior to rendering
        - Does not help with limitation of the GPU / main memory
        - For sequential decompression on the CPU other techniques, e.g. wavelets [Schwartz-2011], could be used instead



  - An irregular sampling pattern is present
    - Often the case with BTF measurements
    - It would be necessary to resample the input data

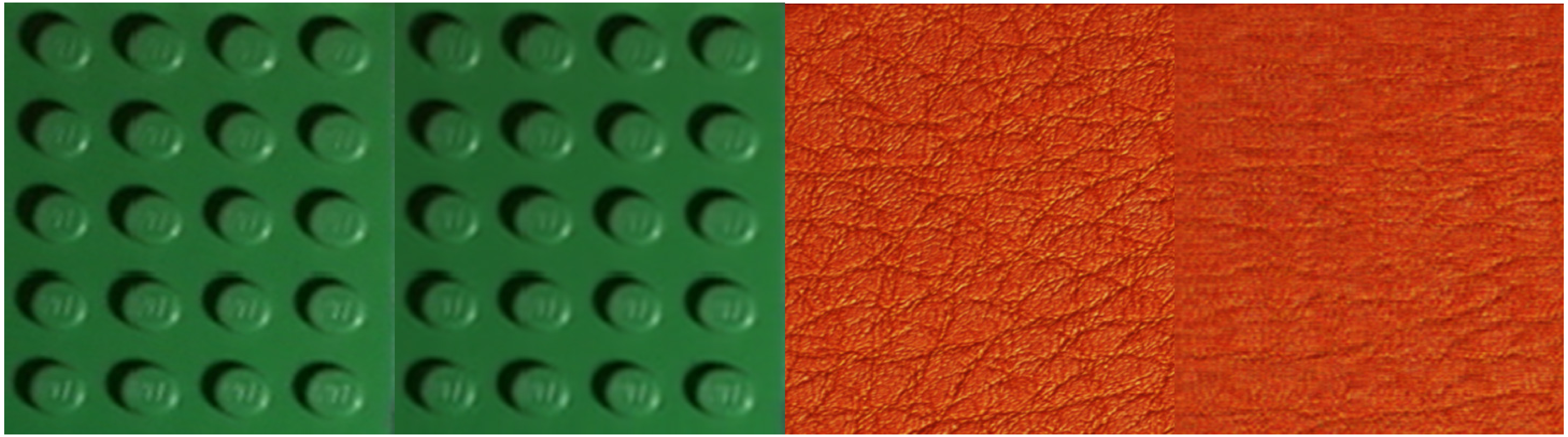


  - The function has to be represented in a specific linear basis in these modes
    - E.g. spherical harmonics, radial basis functions, wavelets, a basis from a PCA…
      - For example for PRT computations [Tsai-2006, Sun-2007]



How many modes to use?

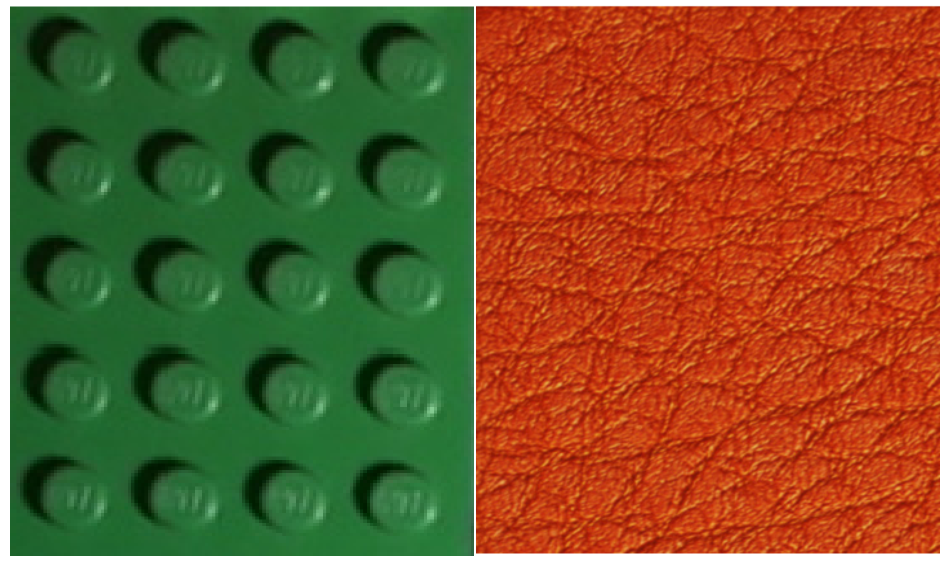
- Sometimes it is not advisable to represent all *"natural"* dimensions of the input dataset as

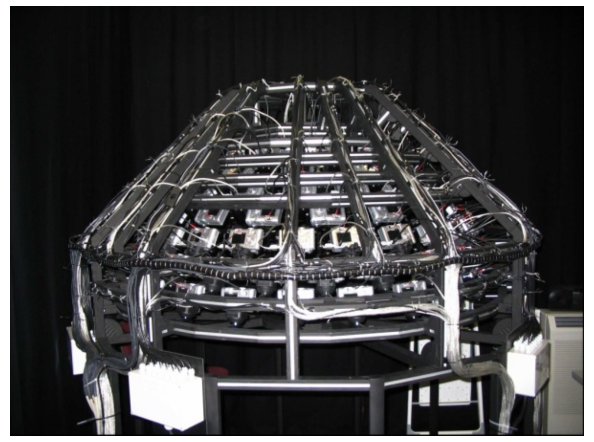


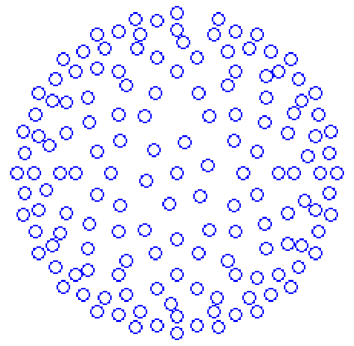
Original **16 Components** **Original** **16 Components**

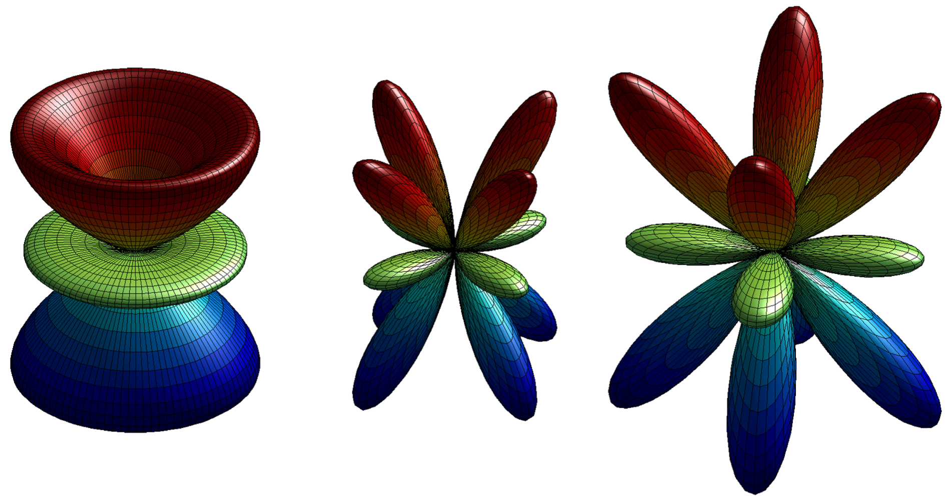
- The Lego Blocks are an example for a BTF used in [Wang-2005]
  - The factorization of the spatial mode has considerable advantages
- More complex leather sample
  - A much larger number of components would be needed for a good reconstruction

Note: only one image was factorized for this example







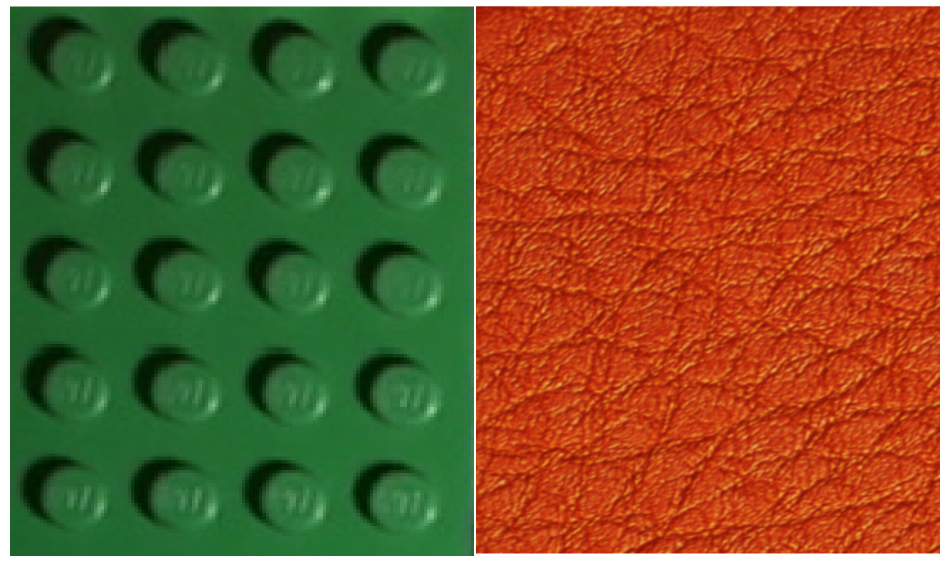


VISUALIZATIONAND MULTIMEDIALAB

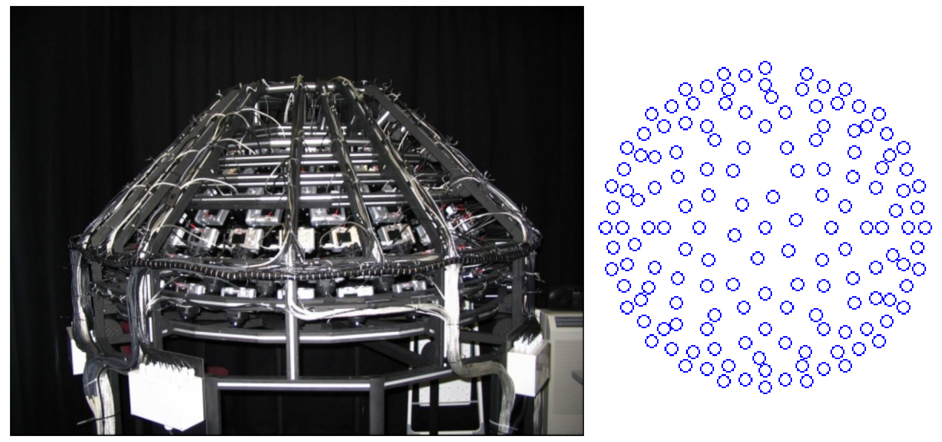
universitätbonn

How many modes to use?

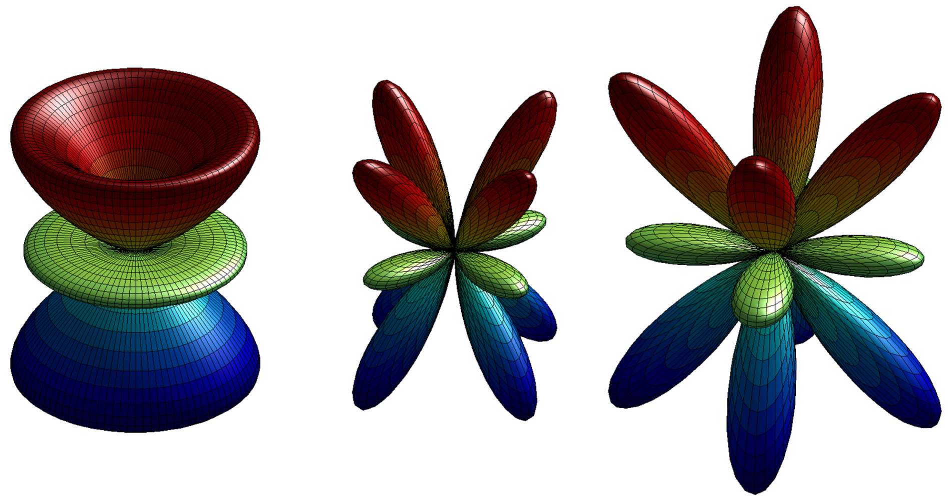
- Sometimes it is not advisable to represent all *"natural"* dimensions of the input dataset as modes
  - The dimensions exhibit a high complexity, which cannot be factorized well
    - No significant gain in compression ratio
    - A large number of components would be needed to encode the complexity
      - Slow decompression
      - [Wang-2005] and [Tsai-2012] decompress spatial compression prior to rendering
        - Does not help with limitation of the GPU / main memory
        - For sequential decompression on the CPU other techniques, e.g. wavelets [Schwartz-2011], could be used instead



  - An irregular sampling pattern is present
    - Often the case with BTF measurements
    - It would be necessary to resample the input data



  - The function has to be represented in a specific linear basis in these modes
    - E.g. spherical harmonics, radial basis functions, wavelets, a basis from a PCA…
      - For example for PRT computations [Tsai-2006, Sun-2007]



Compression results on BTFs

Uncompressed

PCA, 100 Components
RMSE 0.008
SSIM 97.06%

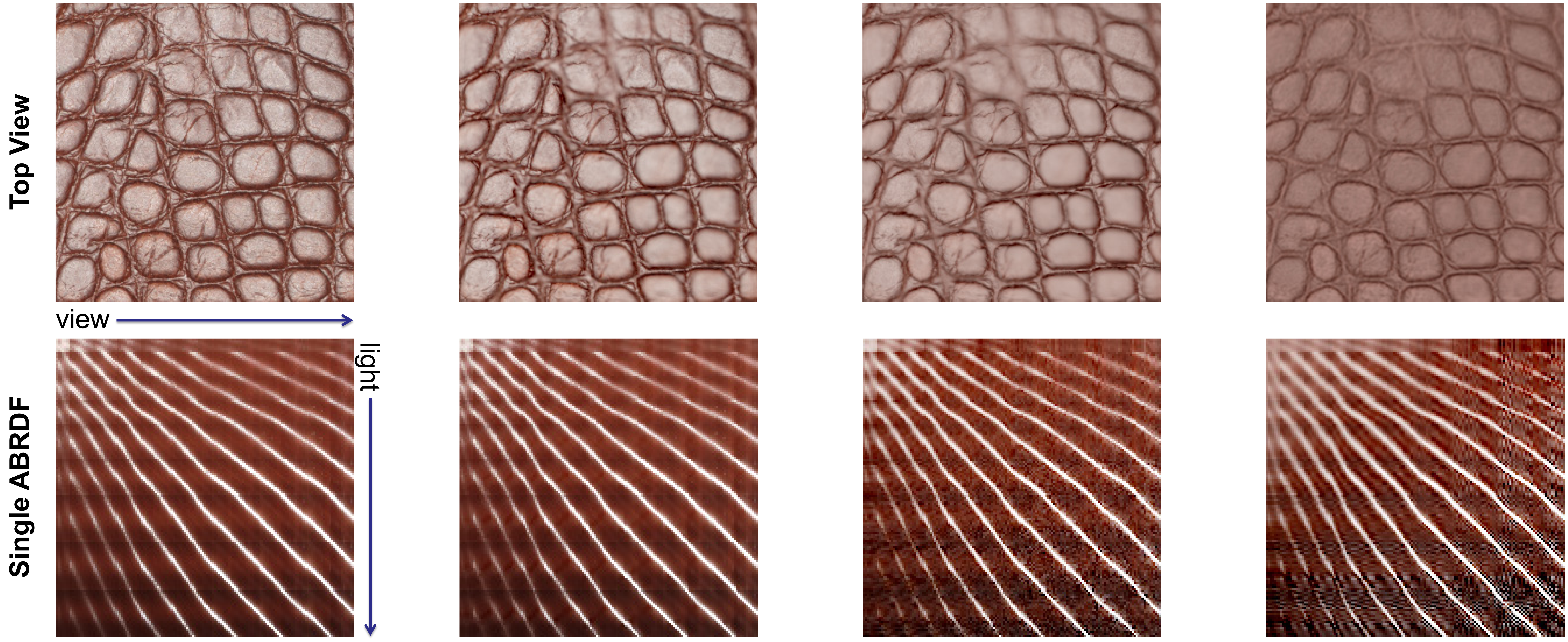
CP, 200 Components
RMSE 0.013
SSIM 96.15%

TUCKER, $28 \times 28 \times 128 \times 128$ core
RMSE 0.022
SSIM 95.49%

All datasets were compressed to about 25 MB. Input: $3 \times 151 \times 151 \times 256 \times 256 \approx 8.8$ GB

Compression results on BTFs

Parameterization via view/light: $(\varphi_i, \theta_i, \varphi_o, \theta_o)$



Uncompressed

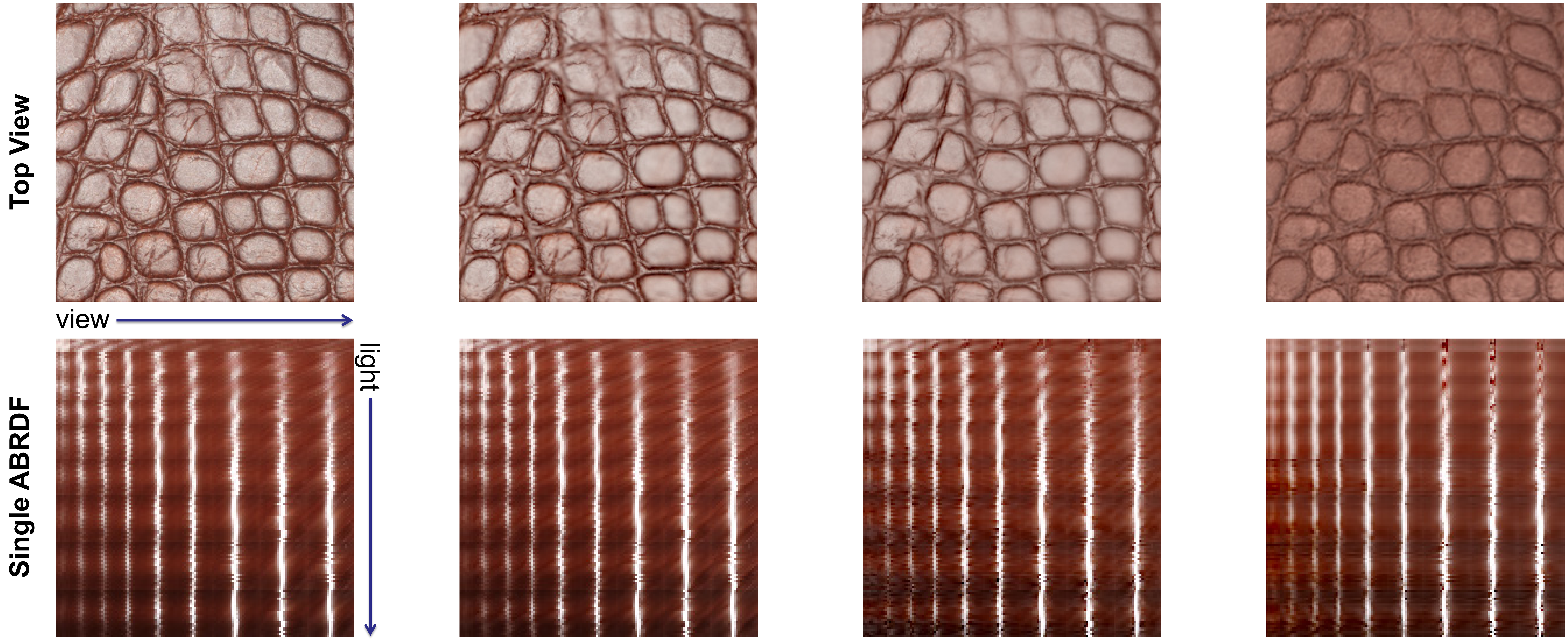
PCA, 100 Components
RMSE 0.008

CP, 200 Components
RMSE 0.013

TUCKER, $28 \times 28 \times 128 \times 128$ core
RMSE 0.022

Compression results on BTFs

Reordered (without resampling): $(\varphi_i, \theta_i, \varphi_o, \theta_o) \rightarrow (\varphi_i, \theta_i, \varphi_o - \varphi_i, \theta_o)$



Uncompressed

PCA, 100 Components
RMSE 0.008

CP, 200 Components
RMSE 0.010

TUCKER, $28 \times 28 \times 128 \times 128$ core
RMSE 0.013

Summary

- Quality of the results depends strongly on your data and problem
  - It is worth considering your parameterization, tensor layout, error metric and decompression requirements

- BRDFs
  - Good results when all these aspects are taken into account
- BTFs
  - Results often not better than PCA based compression
  - More research on parameterization might be interesting
    - More complex than for BRDFs
      - Some effects like parallax or cosine fall-off, depend on light or view direction
      - Highlights better parameterized via halfway vector
      - Normal directions vary spatially
    - Combining several parameterizations [Suykens-2003] might give better results, but was not yet used tensor compression

References

Bilgili-2010	BILGILI A., ÖZTÜRK A., KURT M.: A general BRDF representation based on tensor decomposition. *Computer Graphics Forum* 30, 8 (2011), pp. 2427–2439.
Costantini-2008	COSTANTINI R., SBAIZ L., SUSSTRUNK S.: Higher order svd analysis for dynamic texture synthesis. In *IEEE Transactions on Image Processing* 17, 1 (2008), pp. 42 –52.
Furukawa-2002	FURUKAWA R., KAWASAKI H., IKEUCHI K., SAKAUCHI M.: Appearance based object modeling using texture database: acquisition, compression and rendering. In *Eurographics workshop on Rendering* (2002), pp. 257–266.
Garg-2006	GARG G., TALVALA E.-V., LEVOY M., LENSCH H. P.: Symmetric photography: exploiting data-sparseness in reflectance fields. In *Eurographics conference on Rendering Techniques* (2006), pp. 251–262.
Hasler-2010	GARG G., TALVALA E.-V., LEVOY M., LENSCH H. P.: Symmetric photography: exploiting data-sparseness in reflectance fields. In *Eurographics conference on Rendering Techniques* (2006), pp. 251–262.
Krüger-2008	KRÜGER B., TAUTGES J., MÜLLER M., WEBER A.: Multi-mode tensor representation of motion data. In *Journal of Virtual Reality and Broadcasting* 5, 5 (July 2008).
Liu-2011	LIU G., XU M., PAN Z., RHALIBI A. E.: Human motion generation with multifactor models. In *Journal of Visualization and Computer Animation* 22, 4 (2011), pp. 351–359.
Matusik-2003	MATUSIK W., PFISTER H., BRAND M., MCMILLAN L.: A data-driven reflectance model. In *ACM Transactions on Graphics* 22, 3 (2003), pp. 759–769.
Min-2010	MIN J., LIU H., CHAI J.: Synthesis and editing of personalized stylistic human motion. In *SIGGRAPH symposium on Interactive 3D Graphics and Games* (2010), pp. 39–46.
Mukai-2007	MUKAI T., KURIYAMA S.: Multilinear motion synthesis with level-of-detail controls. In *Pacific Conference on Computer Graphics and Applications* (2007), pp. 9 –17.
Peng-2008	PENG B., QIAN G.: Binocular dance pose recognition and body orientation estimation via multilinear analysis. In *Conference on Computer Vision and Pattern Recognition Workshops* (2008), pp. 1 –8.
Perera-2007	PERERA M., SHIRATORI T., KUDOH S., NAKAZAWA A., IKEUCHI K.: Multilinear analysis for task recognition and person identification. In *International Conference on Intelligent Robots and Systems* (2007), pp. 1409 –1415.
Ruiters-2009	RUITERS R., KLEIN R.: BTF compression via sparse tensor decomposition. In *Computer Graphics Forum* 28, 4 (July 2009), 1181–1188.
Ruiters-2010	RUITERS R., KLEIN R.: A compact and editable representation for measured BRDFs. *Tech. Rep. CG-2010-1*, University of Bonn, (2010).
Ruiters-2012	RUITERS R., SCHWARTZ C., KLEIN R.: Data driven surface reflectance from sparse and irregular samples. In *Computer Graphics Forum* 31, 2 (May 2012), 315–324.
Rusinkiewicz-1998	RUSINKIEWICZ S.: A new change of variables for efficient BRDF representation. In *Eurographics Workshop on Rendering Techniques* (1998), pp. 11–22

Schwartz-2011	SCHWARTZ C., WEINMANN M., RUITERS R., KLEIN R.: Integrated high-quality acquisition of geometry and appearance for cultural heritage. In *Symposium on Virtual Reality, Archeology and Cultural Heritage* (2011), pp. 25–32.
Schwartz-2011b	SCHWARTZ C., RUITERS R., WEINMANN M., KLEIN R.: WebGL-based Streaming and Presentation Framework for Bidirectional Texture Functions. In *Symposium on Virtual Reality, Archeology and Cultural Heritage* (2011), pp. 113-120.
Schwenk-2010	SCHWENK K., KUIJPER A., BOCKHOLT U.: Modeling wavelength-dependent BRDFs as factored tensors for real-time spectral rendering. In *International Conference on Computer Graphics Theory and Applications* (2010), pp. 165–172.
Sun-2007	SUN X., ZHOU K., CHEN Y., LIN S., SHI J., GUO B.: Interactive relighting with dynamic BRDFs. In *ACM Transactions on Graphics* 26, 3 (2007), 27.
Suykens-2003	SUYKENS F., BERGE K. V., LAGAE A., DUTRÉ P.: Interactive Rendering with Bidirectional Texture Functions In *Computer Graphics Forum* 22,3 (2003) pp. 463-472 (2003)
Tsai-2006	TSAI Y.-T., SHIH Z.-C.: All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. In *ACM Transactions on Graphics* 25, 3 (2006), pp. 967–976.
Tsai-2012	TSAI Y.-T., SHIH Z.-C.: K-clustered tensor approximation: A sparse multilinear model for real-time rendering. In *ACM Transactions on Graphics* 31, 3 (2012), 19.
Tu-2009	TU J., FU Y., HUANG T.: Locating nose-tips and estimating head poses in images by tensorposes. *IEEE Transactions on Circuits and Systems for Video Technology* 19, 1 (2009), pp. 90 –102.
Vlasic-2005	VLASIC D., BRAND M., PFISTER H., POPOVIĆ J.: Face transfer with multilinear models. In *ACM Transactions on Graphics* 24, 3 (2005), pp. 426-433
Vasilescu-2002a	VASILESCU M. A. O., TERZOPOULOS D.: Multilinear analysis of image ensembles: Tensorfaces. In *Proceedings of the 7th European Conference on Computer Vision-Part* I (2002), pp. 447–460
Vasilescu-2002b	VASILESCU M.: Human motion signatures: analysis, synthesis, recognition. In *International Conference on Pattern Recognition* (2002), vol. 3, pp. 456 –460
Vasilescu-2004	VASILESCU M. A. O., TERZOPOULOS D.: TensorTextures: Multilinear image-based rendering. *ACM Transactions on Graphics* 23, 3 (2004), pp. 336–342.
Vasilescu-2007	VASILESCU M., TERZOPOULOS D.: Multilinear projection for appearance-based recognition in the tensor framework. In *International Conference on Computer Vision* (2007), pp. 1–8.
Wampler-2007	WAMPLER K., SASAKI D., ZHANG L., POPOVIĆ Z.: Dynamic, expressive speech animation from a single mesh. In *SIGGRAPH/Eurographics symposium on Computer animation* (2007), pp. 53–62.
Wang-2005	WANG H., WU Q., SHI L., YU Y., AHUJA N.: Out of- core tensor approximation of multi-dimensional matrices of visual data. In *ACM Transactions on Graphics* 24, 3 (2005), pp. 527–535.
Wu-2008	WU Q., XIA T., CHEN C., LIN H.-Y. S., WANG H., YU Y.: Hierarchical tensor approximation of multidimensional visual data. In *IEEE Transactions on Visualization and Computer Graphics* 14, 1 (2008), pp. 186–199