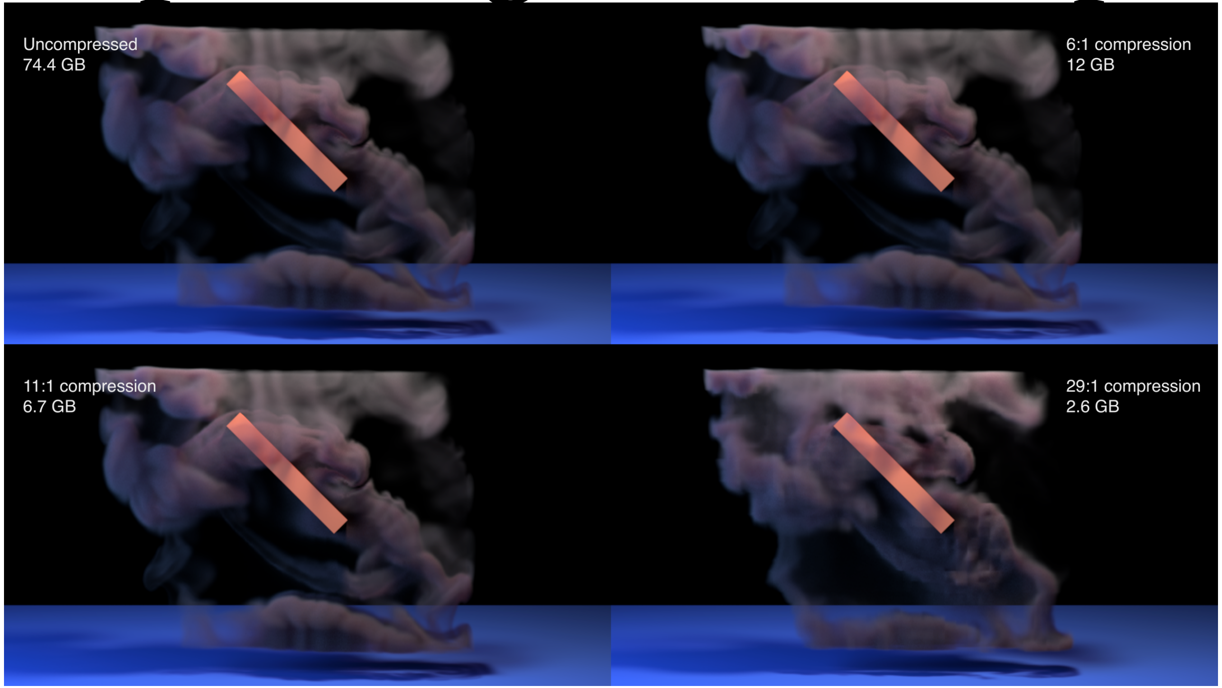
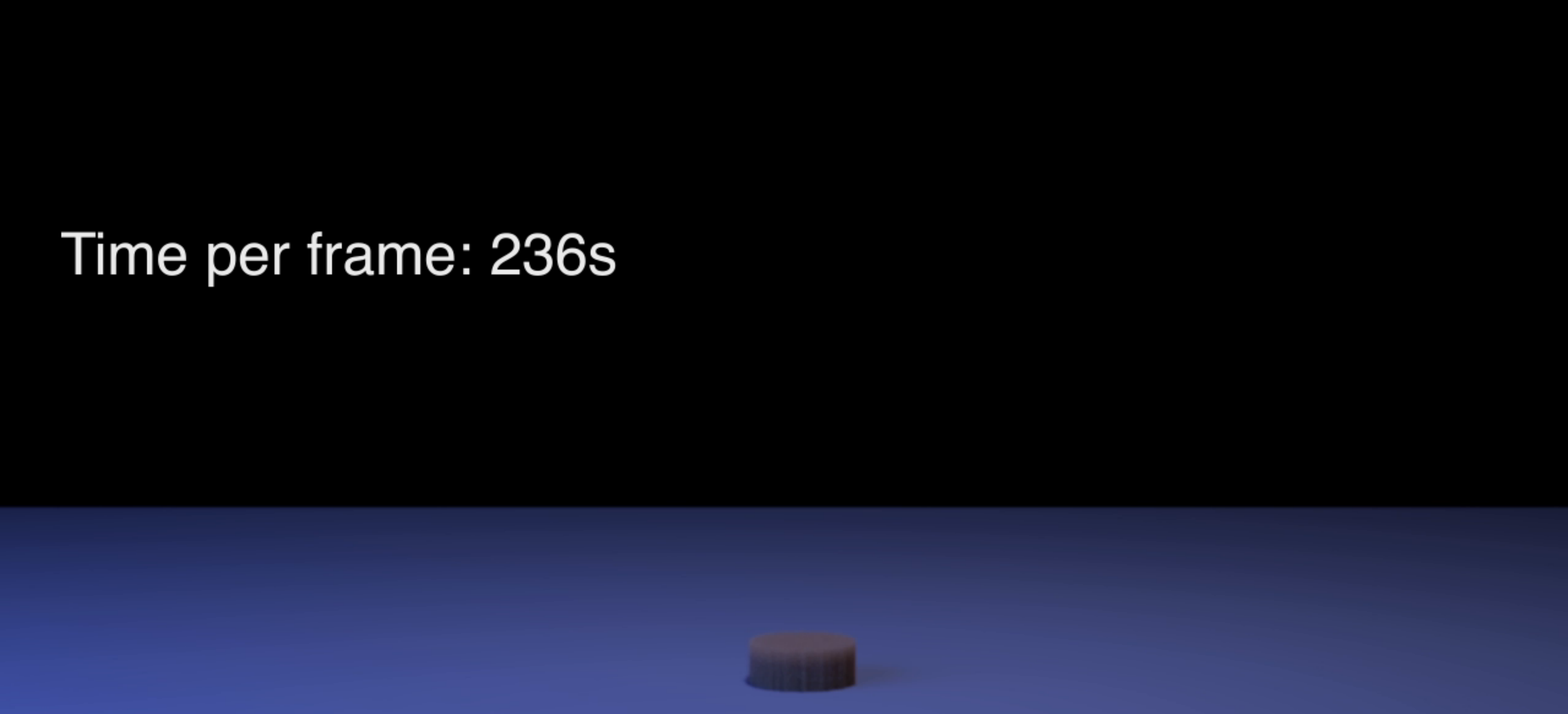
Compressing Fluid Subspaces

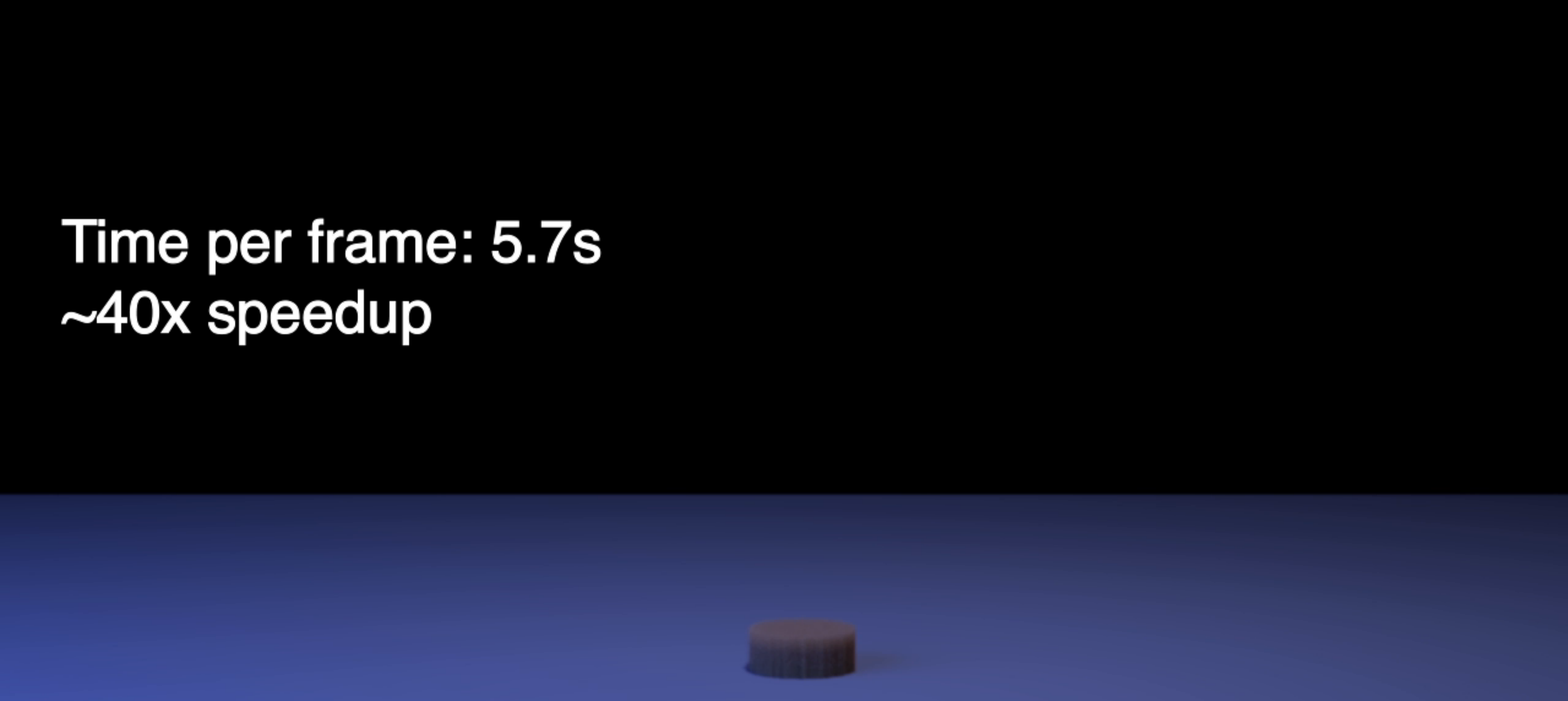


Aaron Demby Jones[1], Pradeep Sen[1], Theodore Kim[1,2]
[1]University of California, Santa Barbara
[2]Pixar Animation Studios

Full Space Simulation



Subspace Re-Simulation



The Catch

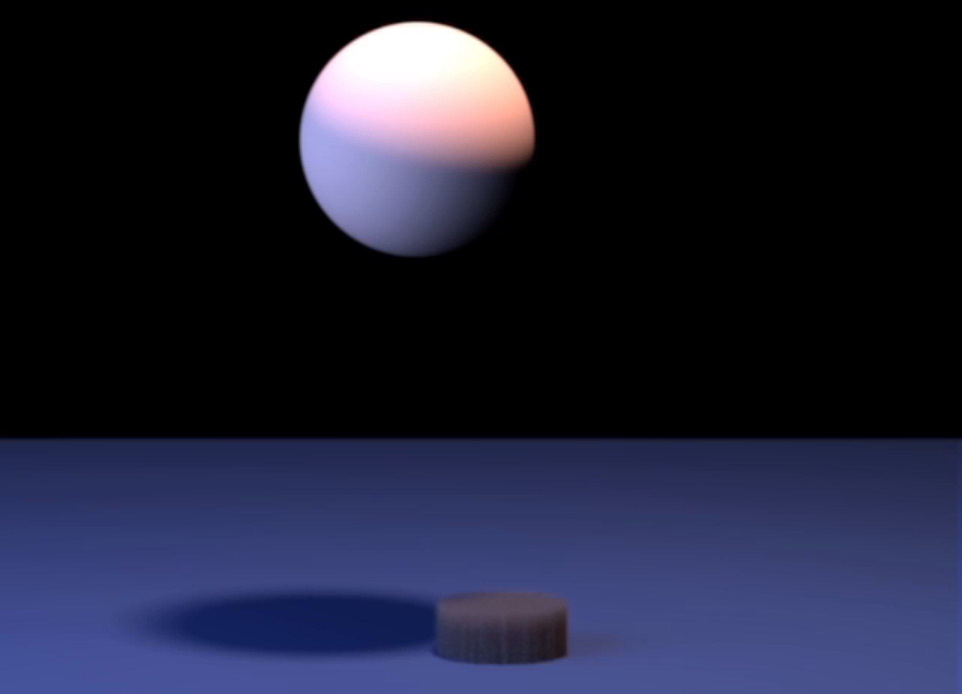
- Time-space tradeoff

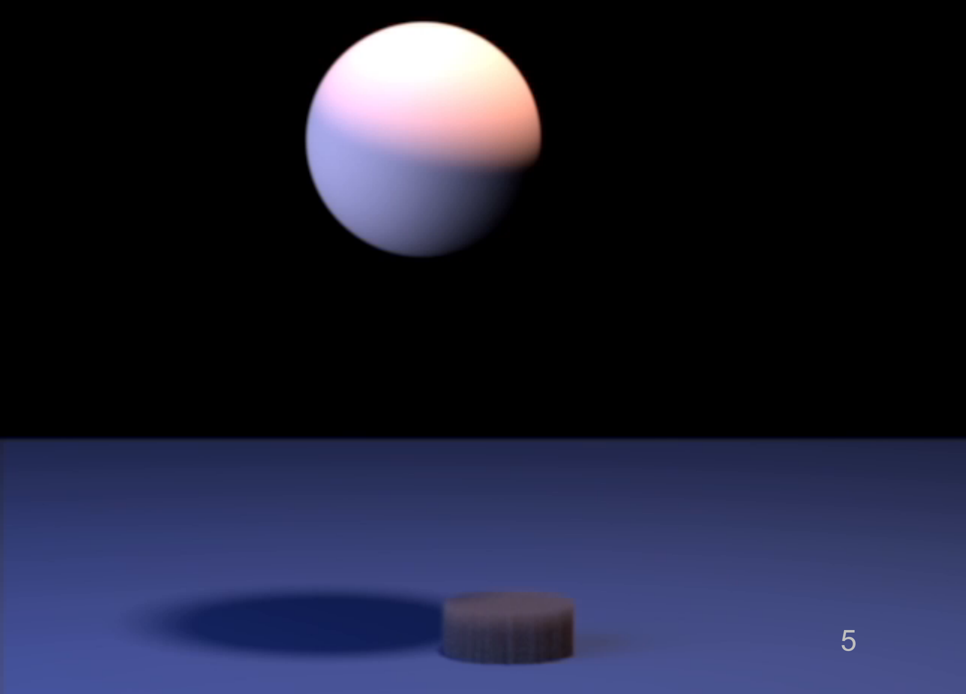
- Huge memory footprint
  - 70+ GB RAM for the previous example!

Our Contribution

Uncompressed
74.4 GB

14:1 compression
5.3 GB





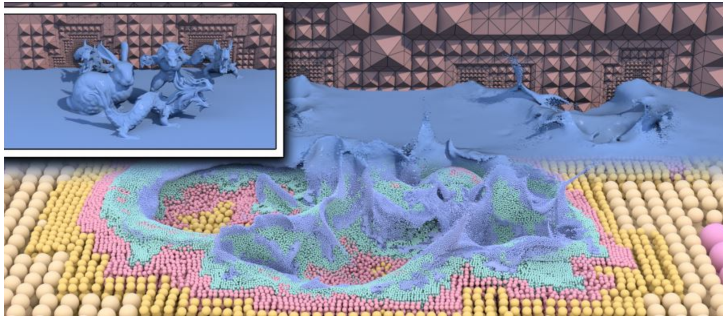
Outline

- Previous Work
- Subspace Preliminaries
- Compression
- Projection and Reconstruction
- Results
- Conclusion and Future Work

Outline

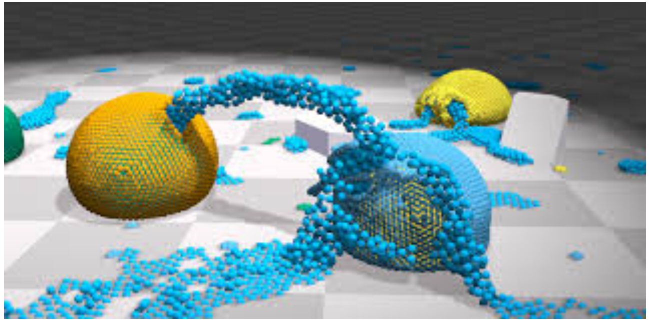
- Previous Work
- Subspace Preliminaries
- Compression
- Projection and Reconstruction
- Results
- Conclusion and Future Work

Spatial Coarsening



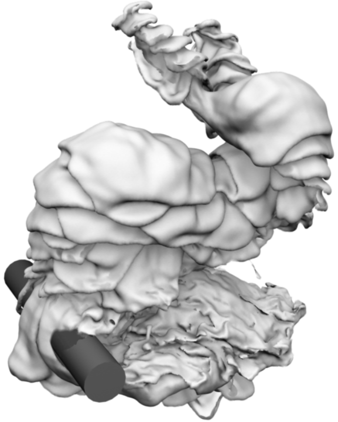
[Ando et al. 2013]

Position-Based Methods



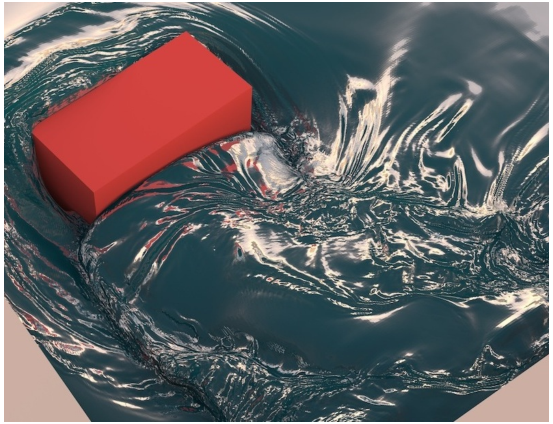
[Macklin et al. 2014]

Vortex Sheet



[Pfaff et al. 2012]

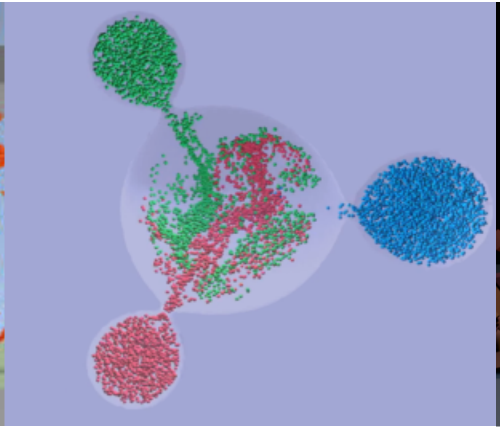
Turbulence Methods



[Thuerey et al. 2013]

Subspace Methods

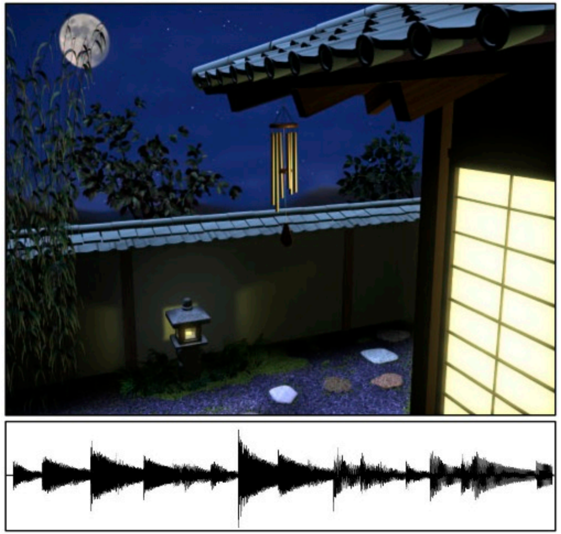
[Treuille et al. 2006]



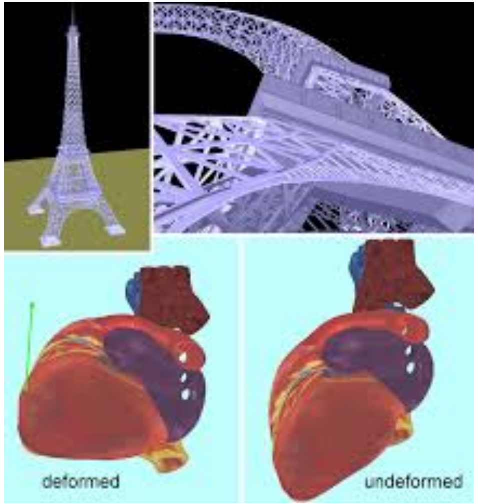
[Stanton et al. 2013]

[Kim and Delaney 2013]

Subspace Methods



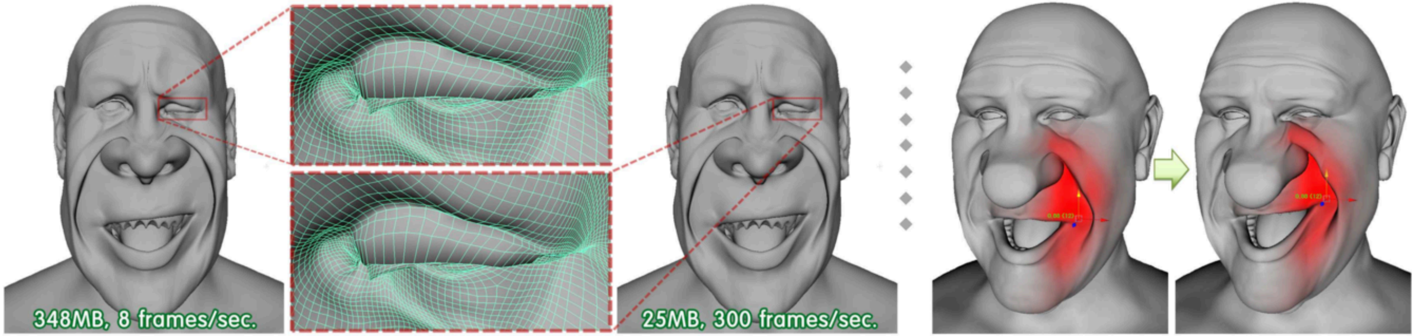
[O'Brien et al. 2002]



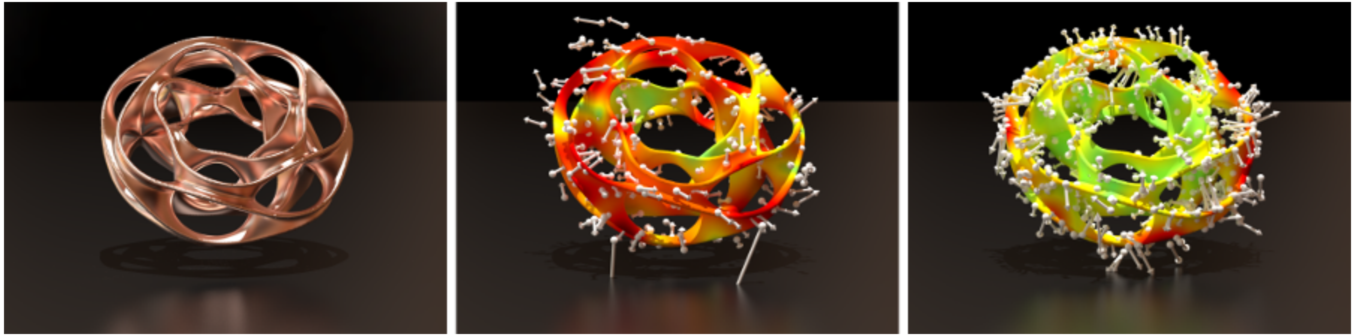
[Barbic and James 2005]

[Xu et al. 2014]

Compression Methods



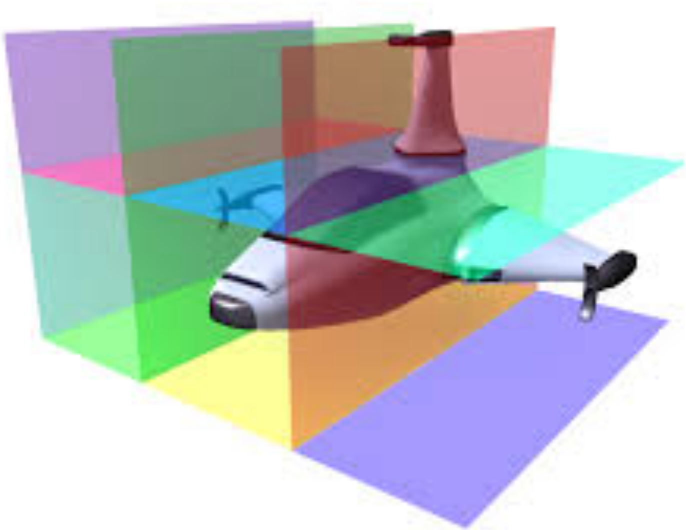
[Seo et al. 2011]



[Langlois et al. 2014]

Compression Methods

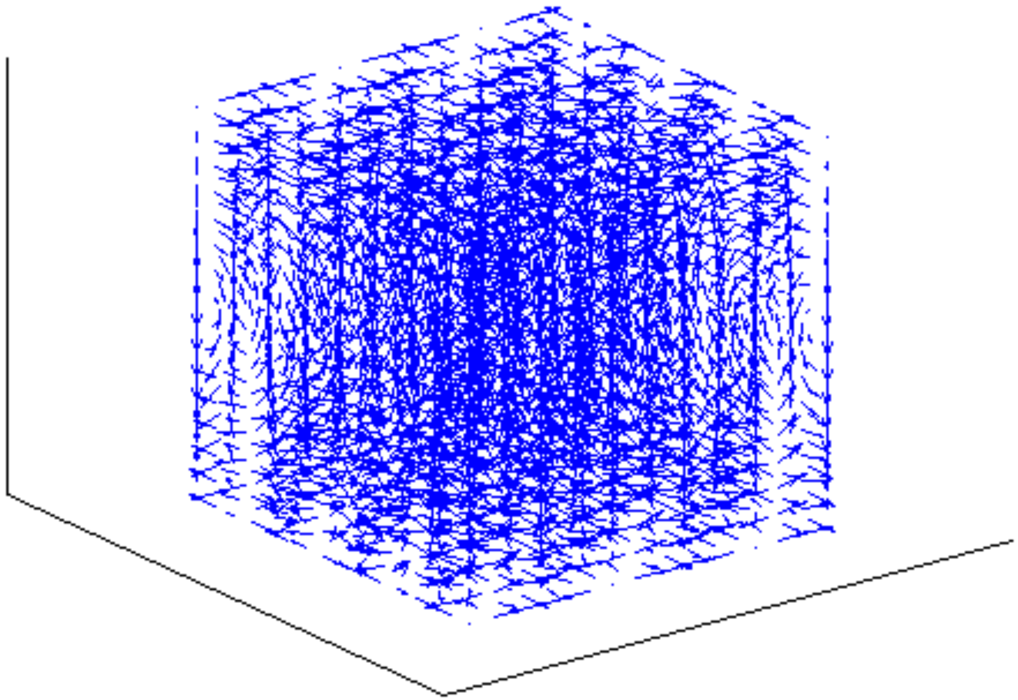
Domain Decomposition



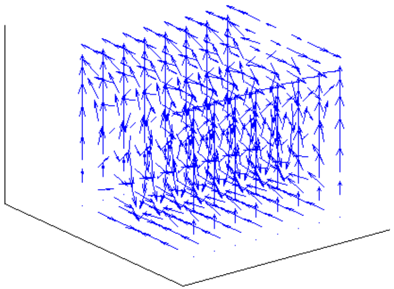
[Wicke et al 2009]

Outline

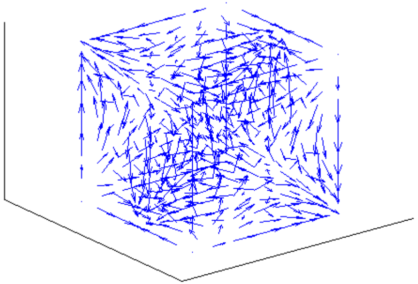
- Previous Work
- Subspace Preliminaries
- Compression
- Projection and Reconstruction
- Results
- Conclusion and Future Work



$$\mathbf{u} \in \mathbb{R}^{3N},\ N = xRes \times yRes \times zRes$$

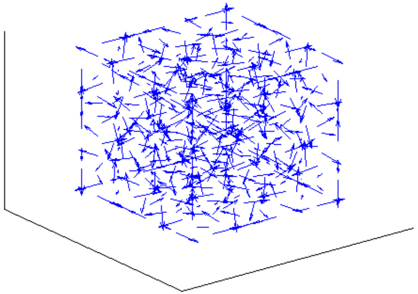


$\mathbf{u}_1$



$\mathbf{u}_2$

$\dots$



$\mathbf{u}_r$

$$\mathbf{u}_j \in \mathbb{R}^{3N},\ r \ll N$$

$$S = \mathrm{span}(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r) \Rightarrow S \simeq \mathbb{R}^r$$

$$\mathbf{U} = \begin{pmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_r \\ | & | & & | \end{pmatrix} \in \mathbb{R}^{3N \times r}$$

Huge matrix

$$\mathbf{U}^T : \mathbb{R}^{3N} \to S$$

Projection

$$\mathbf{U} : S \to \mathbb{R}^{3N}$$

Reconstruction

$$
\begin{tikzcd}
\mathbf{u}^{t} \in \mathbb{R}^{3N} \arrow[r, "\text{Add external forces}"] & {} \arrow[d, "\mathbf{U}^{T}"', "\text{Projection}"] & \mathbf{u}^{t+1} \in \mathbb{R}^{3N} \\
& \mathbf{q}^{t} \in S \arrow[r, "\text{Reduced-order integration}"] & \mathbf{q}^{t+1} \in S \arrow[u, "\mathbf{U}"', "\text{Reconstruction}"]
\end{tikzcd}
$$

Outline

- Previous Work
- Subspace Preliminaries
- Compression
- Projection and Reconstruction
- Results
- Conclusion and Future Work

Compression

$\mathbf{U} \in \mathbb{R}^{3N \times r}$ is the memory culprit

$N = 256^3,\ r = 50 \longrightarrow$ 18.75 GB

Compression

Should support:

- Projection
- Reconstruction
- Random access

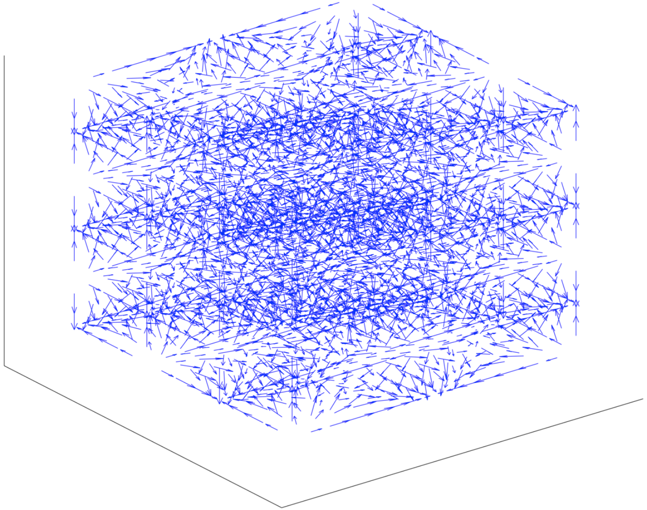
Compression

Transform compression

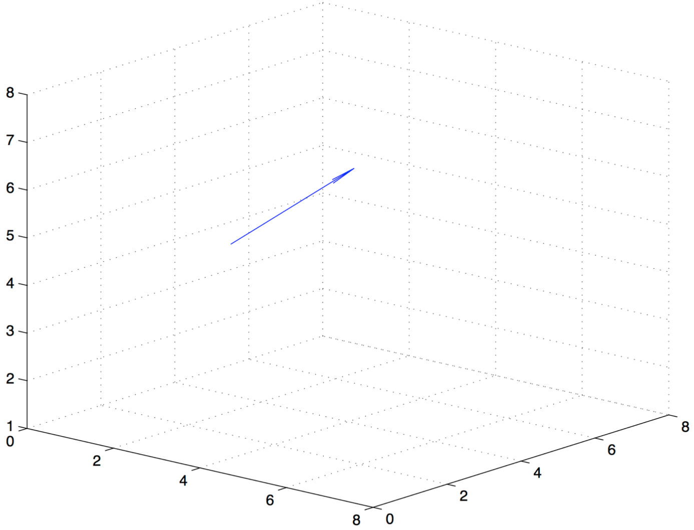
Dampen/Discard

JPEG 2D DCT basis

Compression



transform



$$\mathbf{u}_x(k_1,k_2,k_3)=\kappa_x \sin(k_1x)\cos(k_2y)\cos(k_3z)$$

$$\mathbf{u}_y(k_1,k_2,k_3)=\kappa_y \cos(k_1x)\sin(k_2y)\cos(k_3z)$$

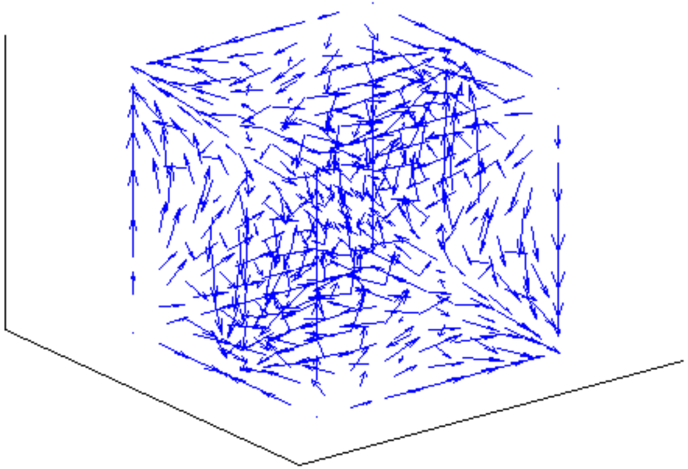
$$\mathbf{u}_z(k_1,k_2,k_3)=\kappa_z \cos(k_1x)\cos(k_2y)\sin(k_3z)$$

$$\hat{\mathbf{u}}=\kappa_{xyz}\delta(\xi-\mathbf{k})$$

[De Witt et al. 2012]

Compression

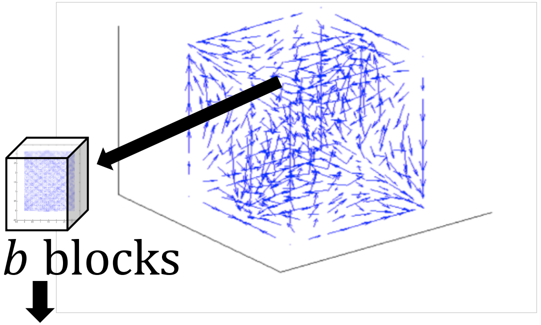
$$\mathbf{U} = \begin{pmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_r \\ | & | & & | \end{pmatrix}$$



$$\mathbf{u}_j = (\mathbf{u}_j^x, \mathbf{u}_j^y, \mathbf{u}_j^z)$$

Compression

Field Data



$b \times b \times b$ blocks

Tuned array selection ↓

3D DCT → Remapping → Damping → 3D Zigzag → Run-Length Encoding → Compressed Data

Compression

3D DCT ✓

Remapping

Floating point $\longrightarrow$ 32-bit integer

abs max $\longmapsto$ int max

Compression

Damping

JPEG damping array:

$$\mathbf{Q}_{\mathrm{2D}} = \begin{bmatrix} 16 & 11 & 10 & 16 & \cdots \\ 12 & 12 & 14 & 19 & \cdots \\ 14 & 13 & 16 & 24 & \cdots \\ 14 & 17 & 22 & 29 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \in \mathbb{Z}^{b\times b}$$

How to adapt to 3D?

$\mathbf{Q}_{\mathrm{2D}}(u,v) \propto u+v$, so how about

$\mathbf{Q}_{\mathrm{3D}}(u,v,w) = 1+u+v+w$? ✗

Compression

$\mathbf{Q}^{\gamma}_{3\mathrm{D}}(u,v,w)=(1+v+u+w)^{\gamma}, \gamma\in\mathbb{R}.$ How to choose γ?

Tuned array selection

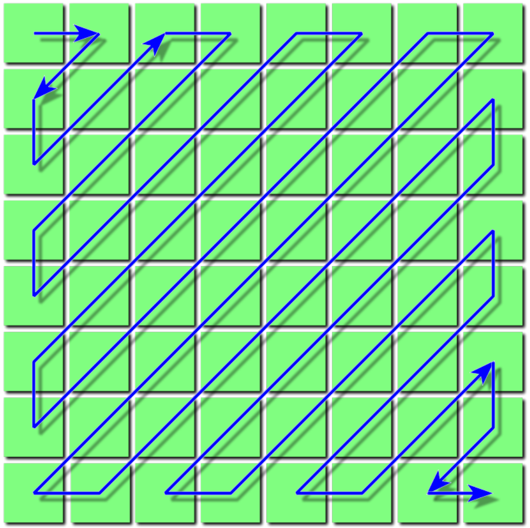
Energy-based search: Varying γ affects the retained energy

Specify quality parameter $q\in[0,1]$

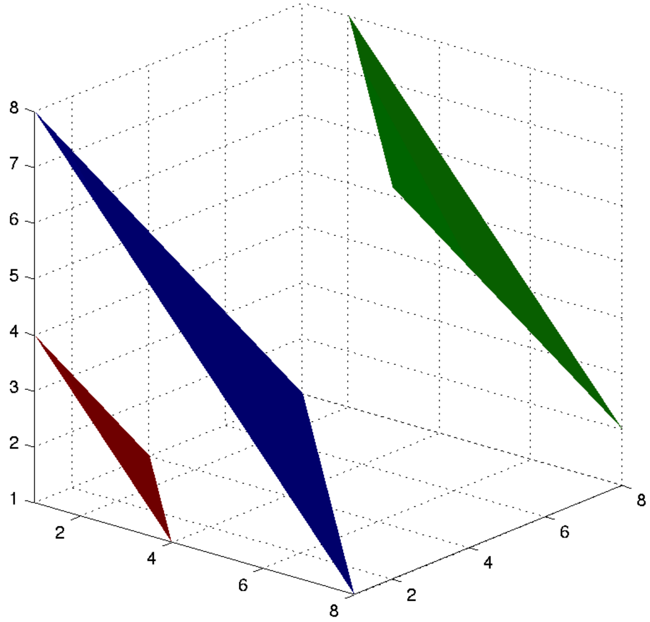
Bisection search for γ

Compression

3D Zigzag

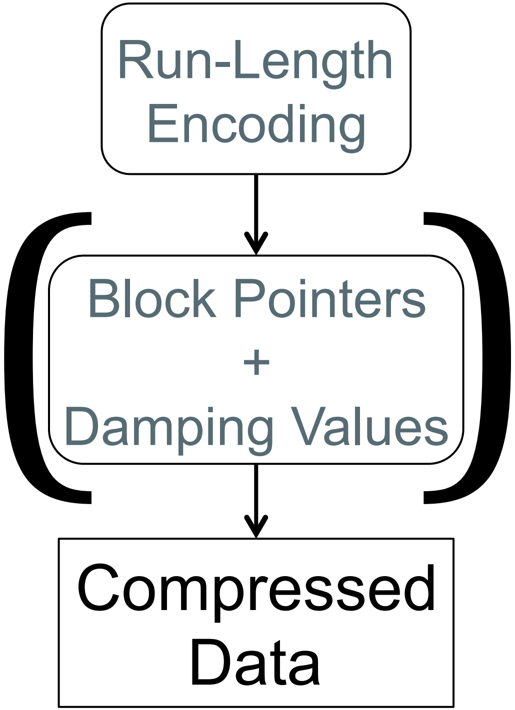


JPEG 2D Zigzag



Scan through in order of increasing index sum $c = 1 + u + v + w$

Compression



$$aaaabb \rightarrow a4b2$$

Random access

Needed for undamping

Projection/Reconstruction

$\mathbf{u} \in \mathbb{R}^{3N} \mapsto \mathbf{q} \in S$

$\mathbf{q} \in S \mapsto \mathbf{u} \in \mathbb{R}^{3N}$

$$\mathbf{U}^T = \begin{pmatrix} \text{---} & \mathbf{u}_1 & \text{---} \\ \text{---} & \mathbf{u}_2 & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{u}_r & \text{---} \end{pmatrix}$$

$$\mathbf{U} = \begin{pmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_r \\ | & | & & | \end{pmatrix}$$

$$\mathbf{q} = \mathbf{U}^T\mathbf{u} = \sum_{j=1}^{r} \mathbf{u}_j \cdot \mathbf{u}$$

$$\mathbf{u} = \mathbf{U}\mathbf{q} = \sum_{j=1}^{r} q_j \mathbf{u}_j$$

Projection/Reconstruction

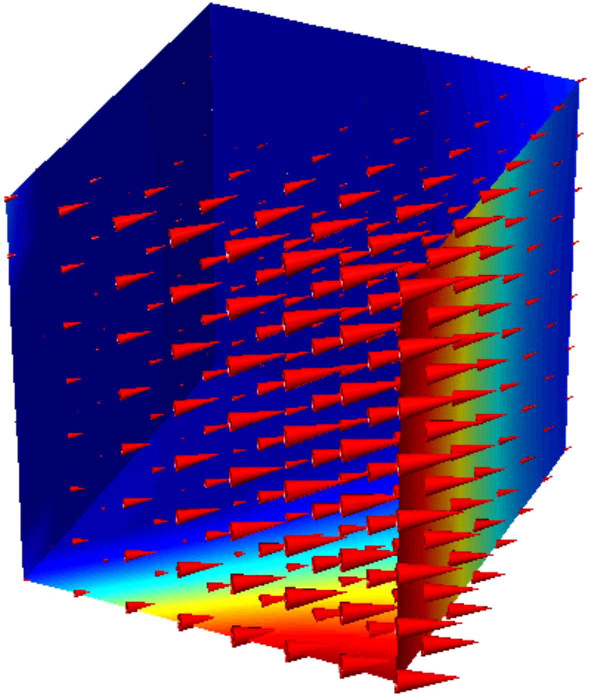
Naive: Decompress $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r$ separately

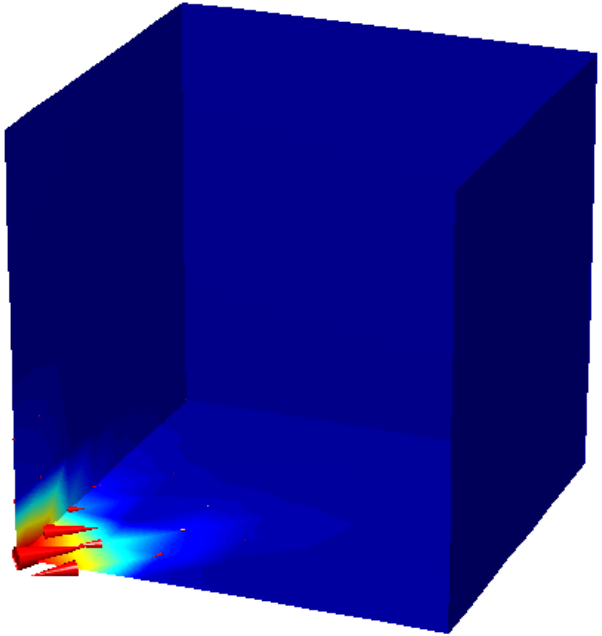
Block size $B = b^3$

$\frac{3N}{B} r$ IDCTs at every timestep

$\mathcal{O}(3Nr)$ for the inner products/vector sums

Too slow! ☹





An 8 x 8 x 8 velocity field block, $\mathbf{u}$

Sparse frequency-domain representation, $\hat{\mathbf{u}}$

Projection/Reconstruction

- DCT properties:
  - *Unitary*: $\left\langle \mathbf{x}, \mathbf{y} \right\rangle = \left\langle \hat{\mathbf{x}}, \hat{\mathbf{y}} \right\rangle$
  - *Linear*: $\widehat{\sum_j c_j \mathbf{x}_j} = \sum_j c_j \hat{\mathbf{x}}_j$

Projection/Reconstruction

$$\mathbf{q} = \mathbf{U}^T\mathbf{u} = \sum_{j=1}^{3N} \mathbf{u}_j \cdot \mathbf{u} \overset{\text{unitary}}{=} \sum_{j=1}^{3N} \hat{\mathbf{u}}_j \cdot \hat{\mathbf{u}}$$

$$\mathbf{u} = \mathbf{U}\mathbf{q} = \sum_{j=1}^{r} q_j \mathbf{u}_j \underset{\text{linear}}{=} \mathrm{IDCT}\left(\sum_{j=1}^{r} q_j \hat{\mathbf{u}}_j\right)$$

Projection/Reconstruction

	Frequency Domain	Naive
DCTs/IDCTs	$\frac{3N}{B}$	$\frac{3N}{B}r$
Inner products	$\mathcal{O}(S)$	$\mathcal{O}(3Nr)$

$\approx 10\times$ speedup ☺

Outline

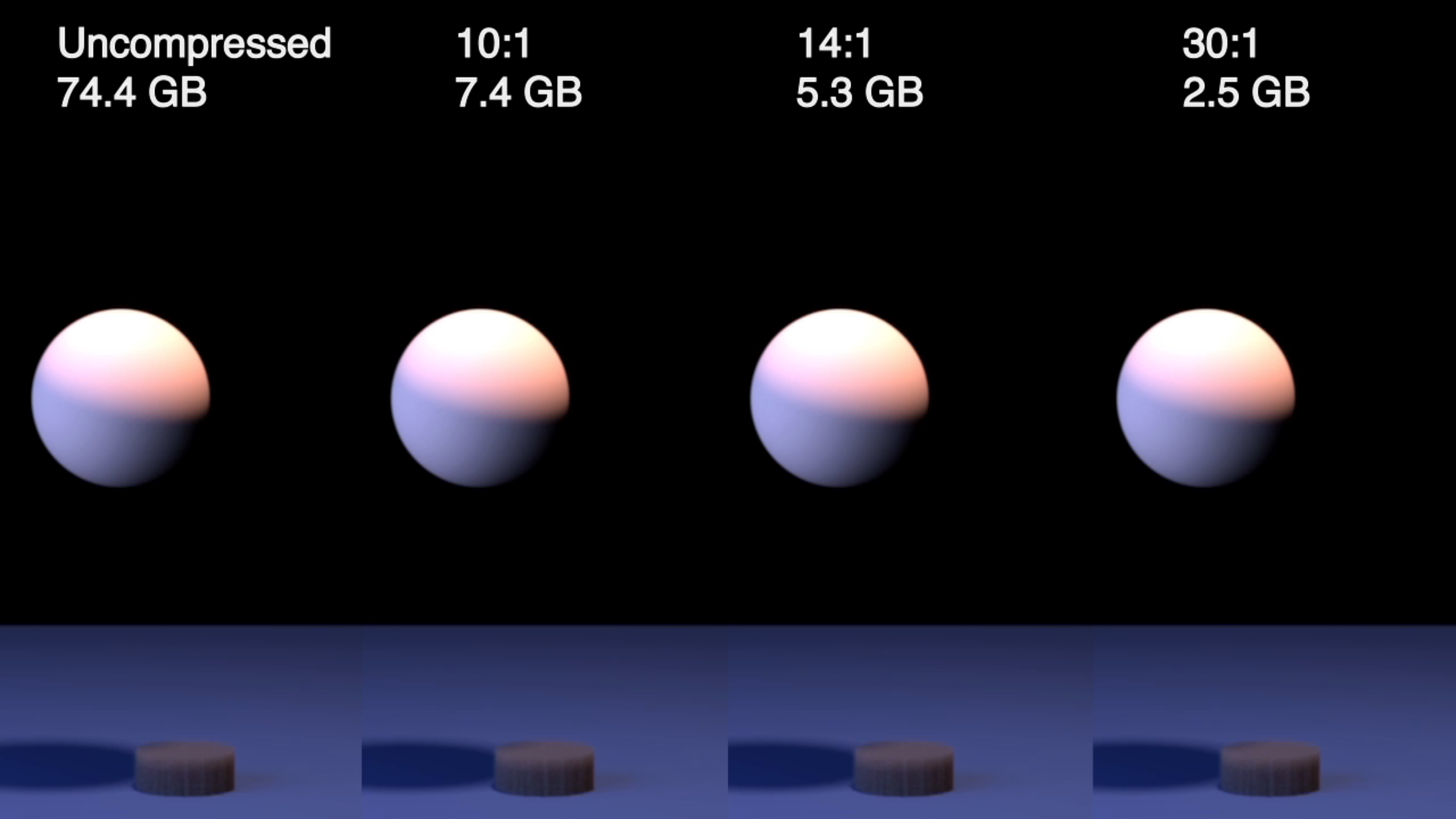
- Previous Work
- Subspace Preliminaries
- Compression
- Projection and Reconstruction
- Results
- Conclusion and Future Work

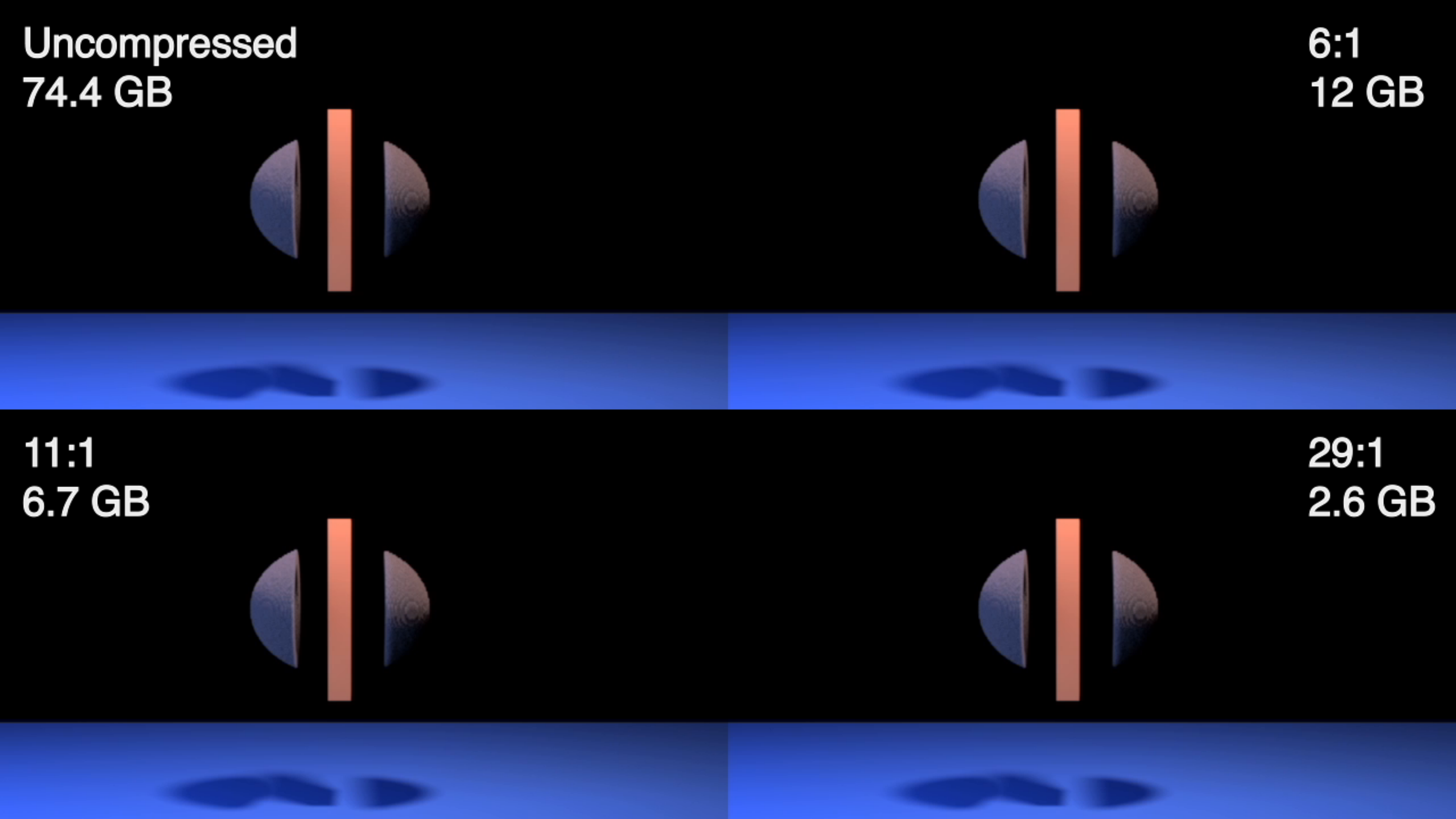
Uncompressed
74.4 GB

6:1
12 GB

11:1
6.7 GB

22:1
3.4 GB


Uncompressed
74.4 GB
10:1
7.4 GB
14:1
5.3 GB
30:1
2.5 GB


Uncompressed
74.4 GB
6:1
12 GB
11:1
6.7 GB
29:1
2.6 GB


Uncompressed
74.4 GB

11:1 compression
6.7 GB

Timings

Plume	Uncompressed	6 : 1	11 : 1	22 : 1
Time per frame Preprocess	4.5s N/A	19.9s 02h 07m 55s	16.1s 02h 07m 55s	12.8s 02h 15m 59s

Sphere	Uncompressed	10 : 1	14 : 1	30 : 1
Time per frame Preprocess	5.4s N/A	17.1s 02h 01m 38s	15.1s 02h 20m 14s	12.2s 02h 27m 59s

Fan	Uncompressed	6 : 1	11 : 1	29 : 1
Time per frame Preprocess	5.7s N/A	23.2s 01h 08m 25s	18.3s 02h 18m 45s	14.8s 02h 11m 51s

Outline

- Previous Work
- Subspace Preliminaries
- Compression
- Projection and Reconstruction
- Results
- Conclusion and Future Work

Contributions

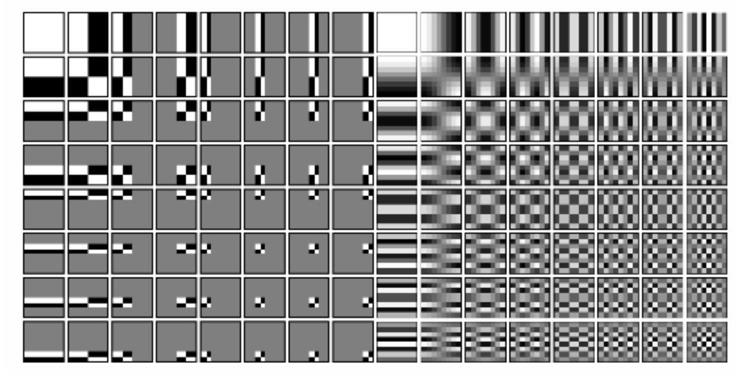
- Compression algorithm for subspace matrices
  - Order of magnitude compression
  - No perceivable artifacts
  - Generality of subspace preserved
- Sparse frequency-domain subspace projection
  - Efficient decompression

Limitations

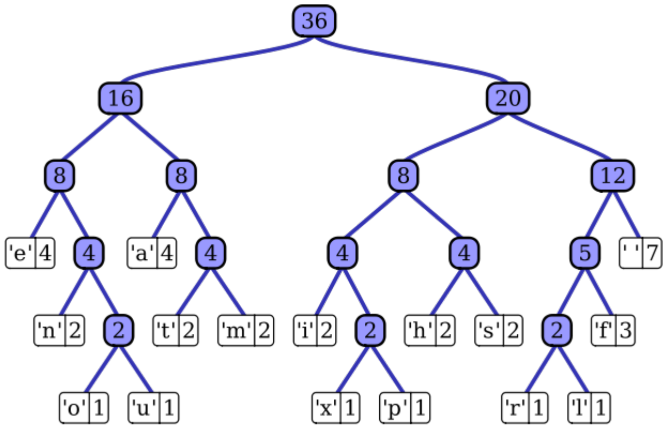
- Speed
- Scene complexity
- Limited to regular grids

Future Work

- More general transform basis?



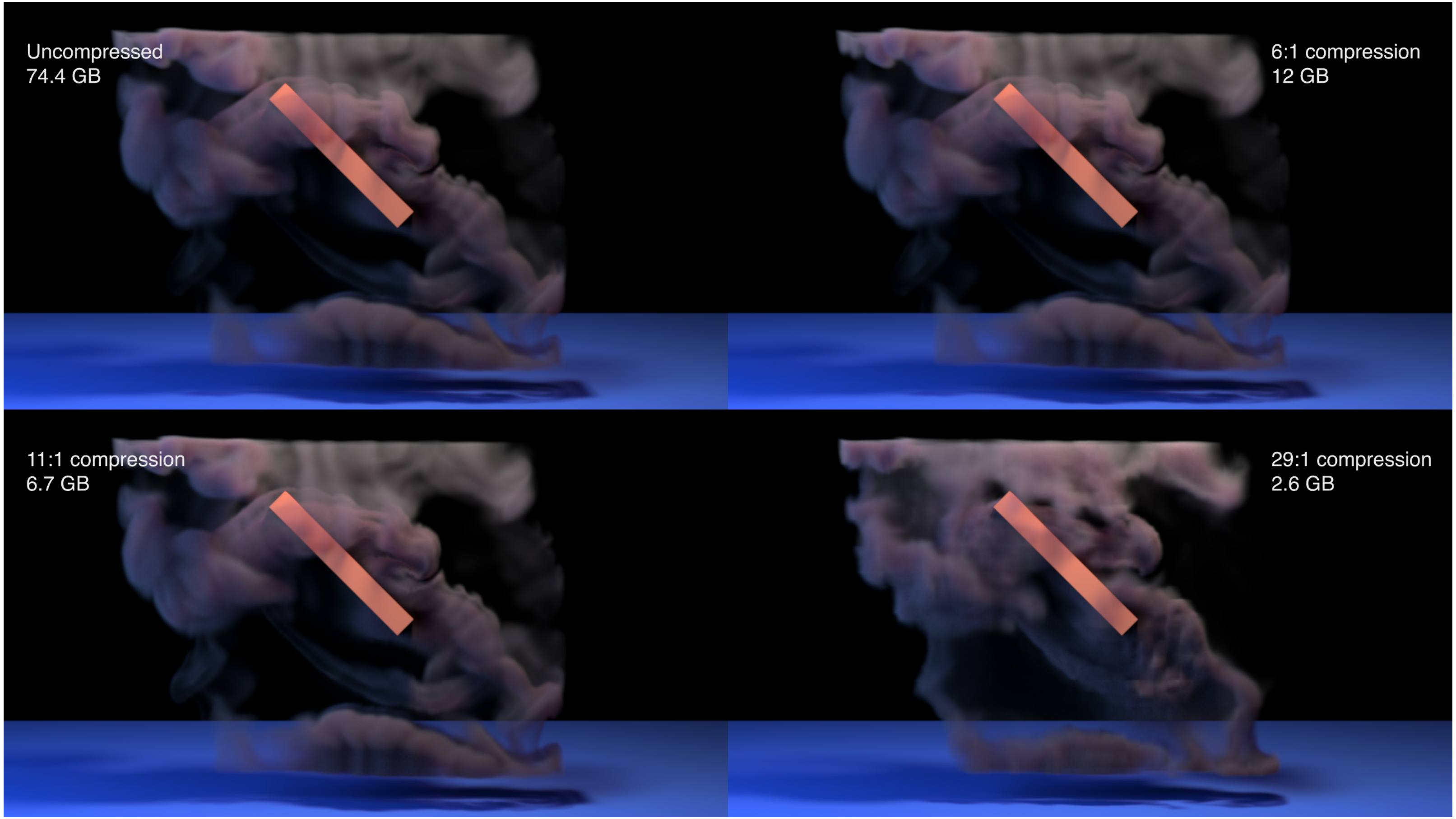
- Lossless compression?



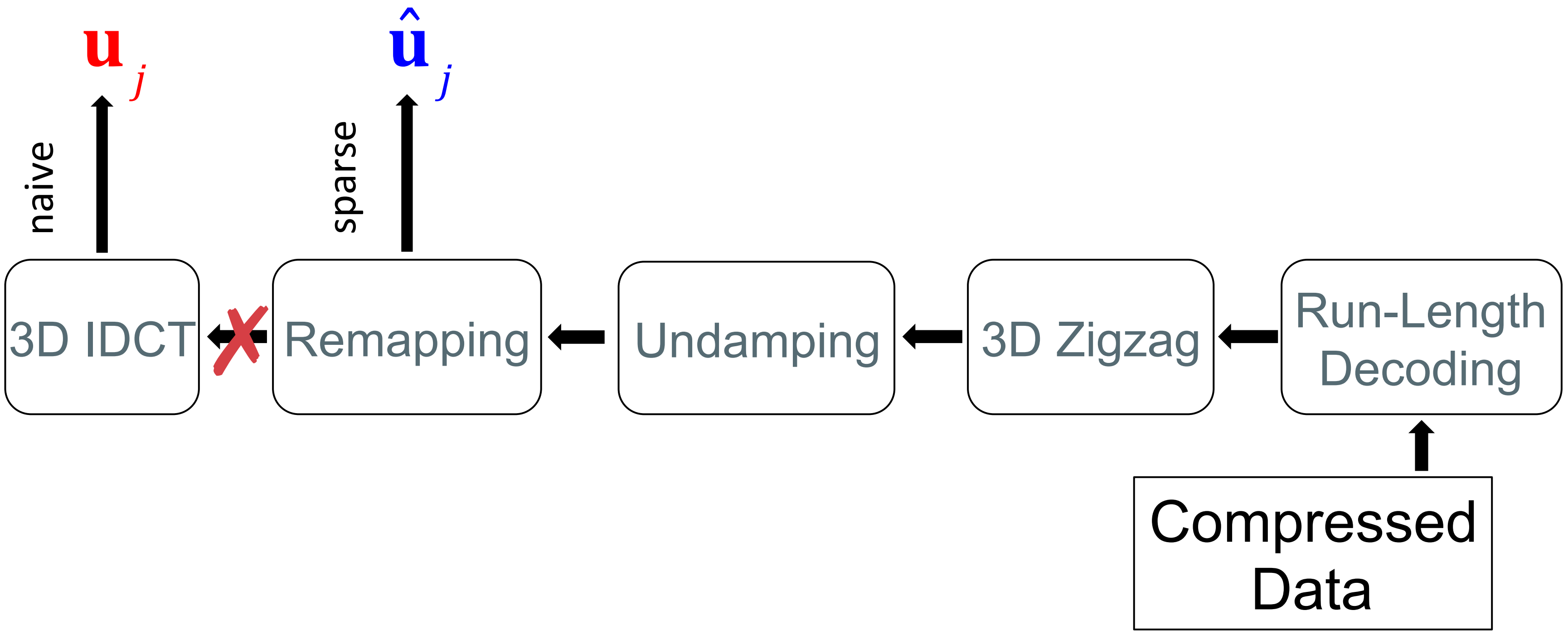
Acknowledgments

- NSF CAREER award IIS-1253948
- NSF grant IIS-1321168
- UCSB Center for Scientific Computing
  - NSF MRSEC (DMR-1121053)
  - Hewlett Packard

Thank you!


Uncompressed
74.4 GB
6:1 compression
12 GB
11:1 compression
6.7 GB
29:1 compression
2.6 GB

Projection/Reconstruction



Compression

$\mathbf{Q}_{3\mathrm{D}}^{\gamma}(u,v,w)=(1+v+u+w)^{\gamma}$, $\gamma\in\mathbb{R}$. How to choose γ?

$$
\begin{tikzcd}
B\in\mathbb{Z}^{b\times b\times b} \arrow[r, "\text{damping}"] \arrow[d, "\text{energy}"] & C=\mathrm{round}\left(\dfrac{B}{\mathbf{Q}_{3\mathrm{D}}^{\gamma}}\right) \arrow[r, "\text{“undamping”}"] & \tilde{B}=\mathbf{Q}_{3\mathrm{D}}^{\gamma}\cdot C \arrow[d, "\text{energy}"] \\
\|B\|^2=\sum_{u,v,w=1}^{b}\left|B_{u,v,w}\right|^2 & \|\tilde{B}\|\le\|B\| & \|\tilde{B}\|^2=\sum_{u,v,w=1}^{b}\left|\tilde{B}_{u,v,w}\right|^2
\end{tikzcd}
$$