HMLFC: Hierarchical Motion-Compensated Light Field Compression for Interactive Rendering

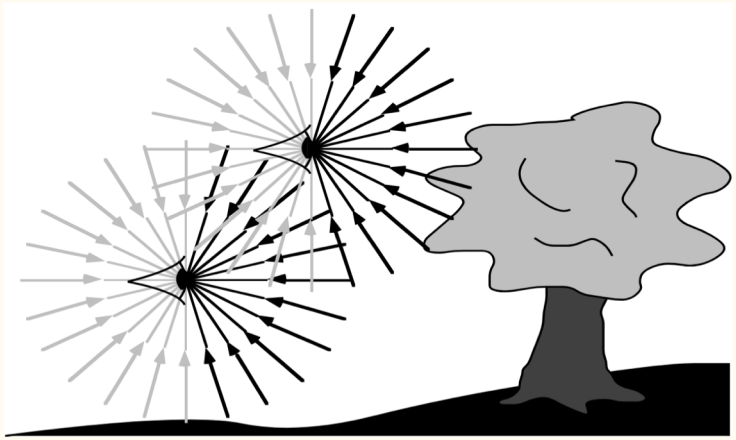
Srihari Pratapa[1] & Dinesh Manocha[2]

University of North Carolina at Chapel Hill[1]
University of Maryland at College Park[2]

High Performance Graphics, July 2019

Light Fields (LF)

- Capture the scene by storing all the light rays
- Render the scene using the light rays
- Photo-realistic rendering of the scene



[Adelson et al. 1991]

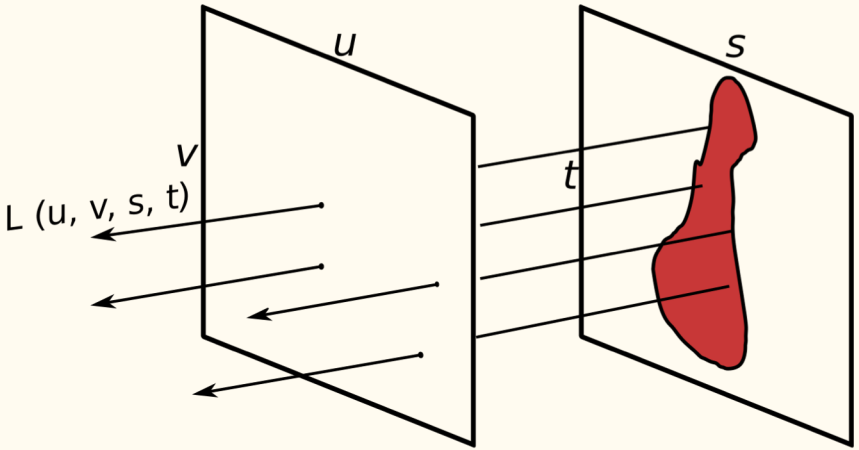
Light Fields (LF)

- Highly complex scenes
- Light rays are sampled using camera images
- Render new views by filtering sampled light rays

[Joan Charmant 2015]

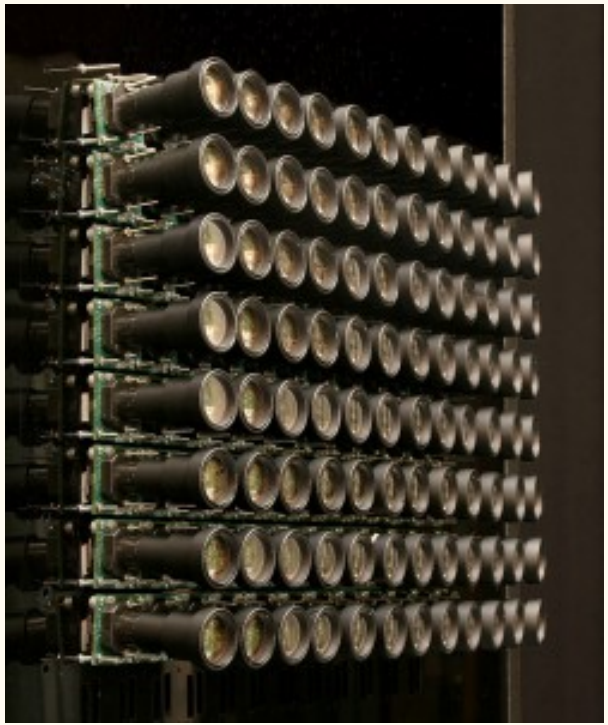
Light Fields: Representation

- Several ways to represent the idea are proposed [McMillan et al. 1995] [Shum et al. 2004]
- Intensity of light rays stay constant
- 4D representations: $\mathrm{L} = P(u, v, s, t)$ [Gortler et al. 1996, Levoy et al. 1996]
  - Two planes: (u, v) & (s, t)

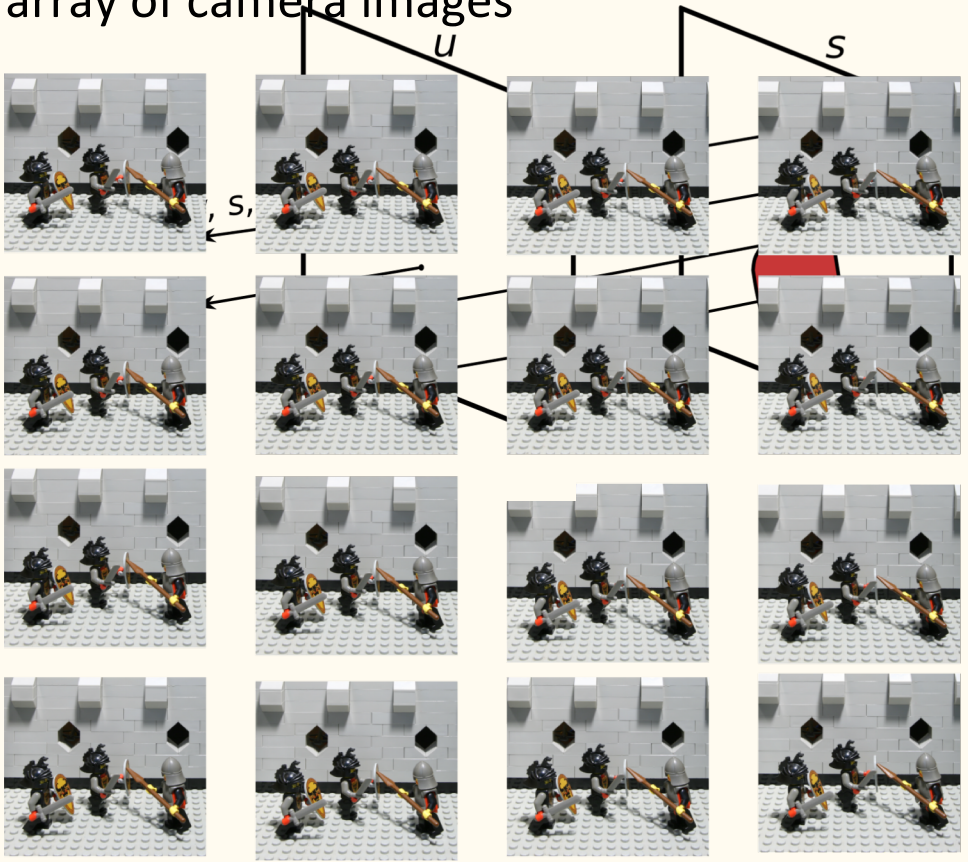


Light Fields: Capturing

- Light rays are captured using large array of camera images

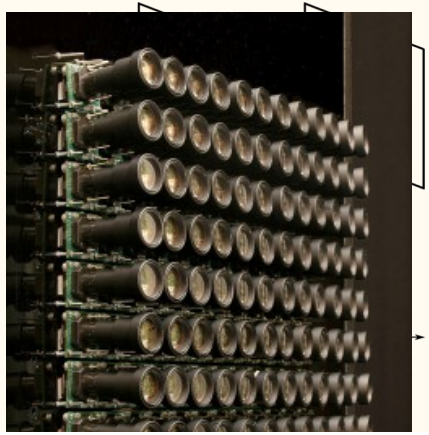


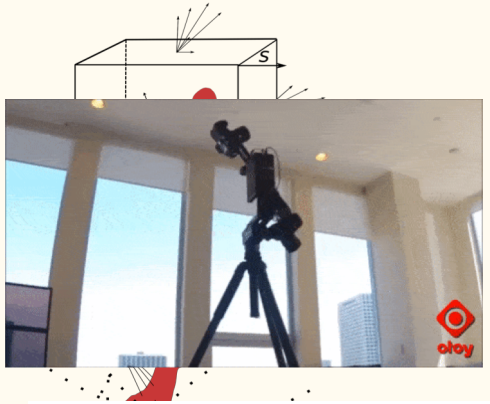
Stanford Camera Rig

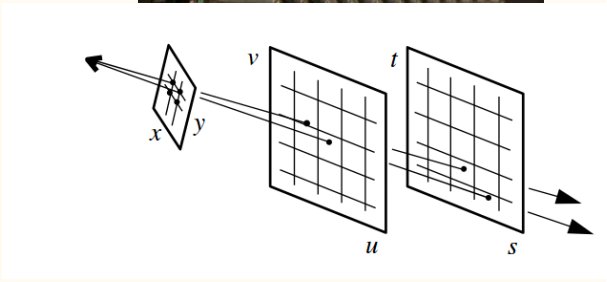


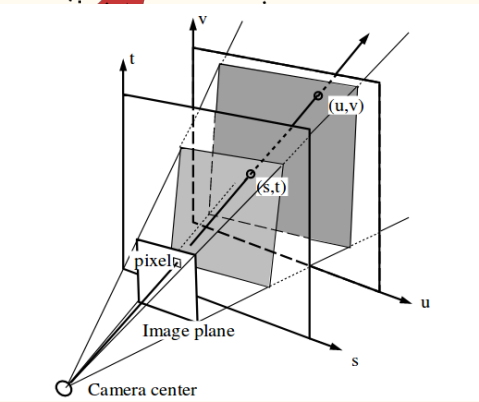
Challenges

- Parameterizing the scene
- Capturing the scene using images
- Representations of the images
- Compression of the captured images
- Rendering from the sampled images







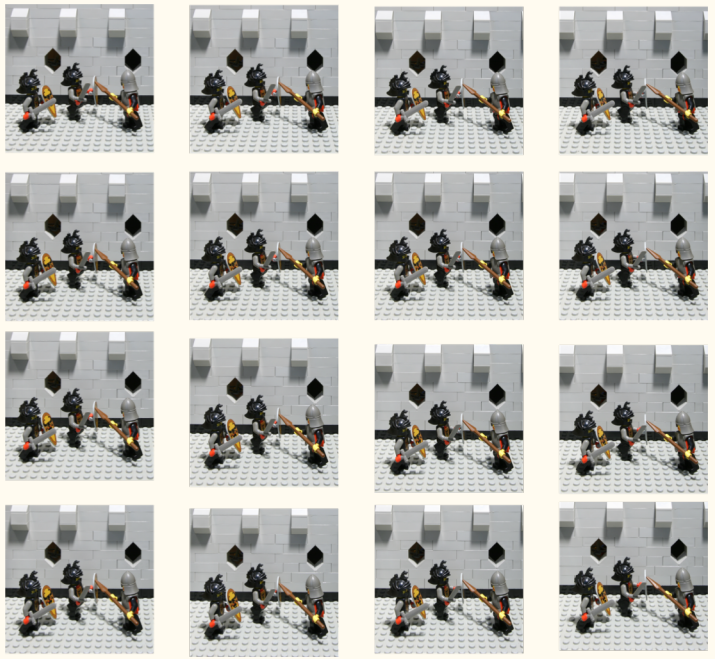


Challenges

- Parameterizing the scene
- Capturing the scene using images
- Representations of the images
- **Compression of the captured data**
- Rendering the scene back

Data – Rendering Quality

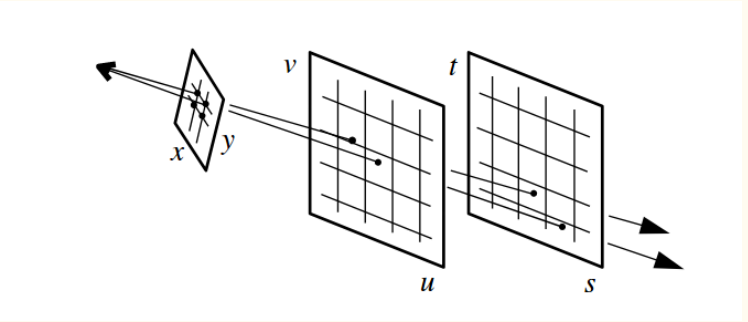
- High image sampling rate (1K – 10K images) [chai et al. 2000]
- High resolution of image samples (1K, 2K)
- Data sizes range from 300MB- 100 GB [Levoy et al. 2000 , Lin et al. 2000]
- Major bottle neck in terms of storage, transmission, and real-time rendering



Data – Real-time Rendering

- Parallel processing is required for real-time rendering
- New views are generated using filtering from the captured images
- Fast parallel access to the pixel values
- Image samples should be present in video memory

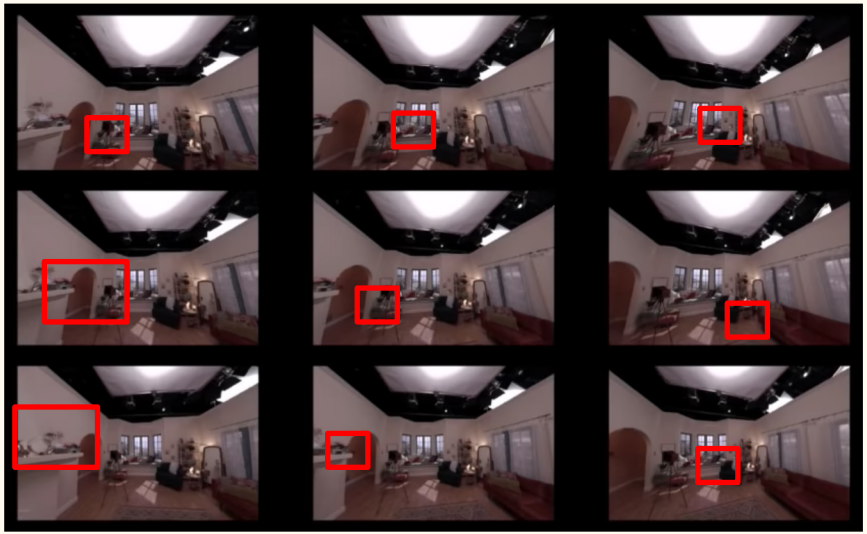
Ray Tracing



Levoy et al. 96

Rendering

- Rendering a new view requires only small set of pixels from image samples

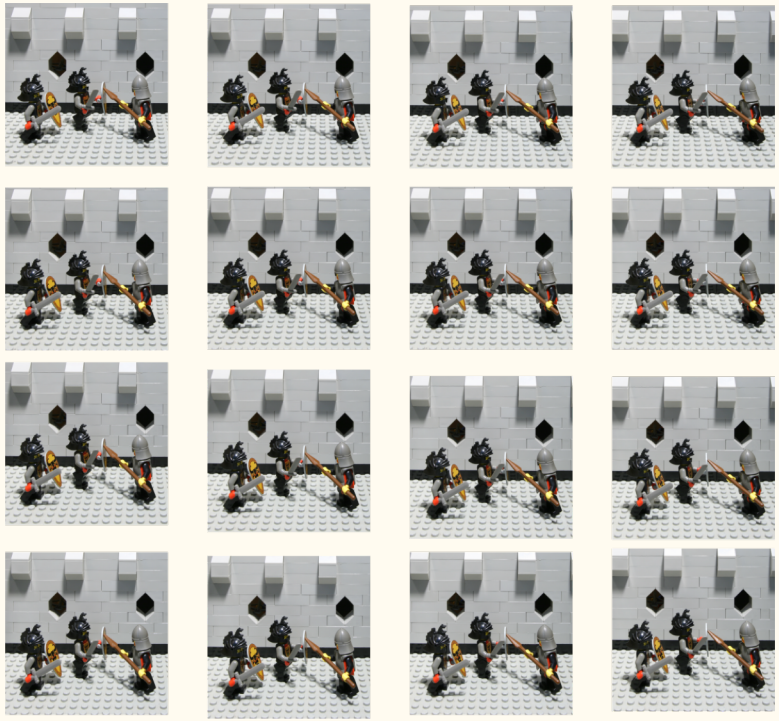


Captured Images

Rendered views

Compression: Requirements

- Storing uncompressed data is expensive
- **Random Access** – Decode only the required set of pixels
- Fast **low-latency decoding** capability
- Video memory & memory bandwidth are scarce on mobile devices

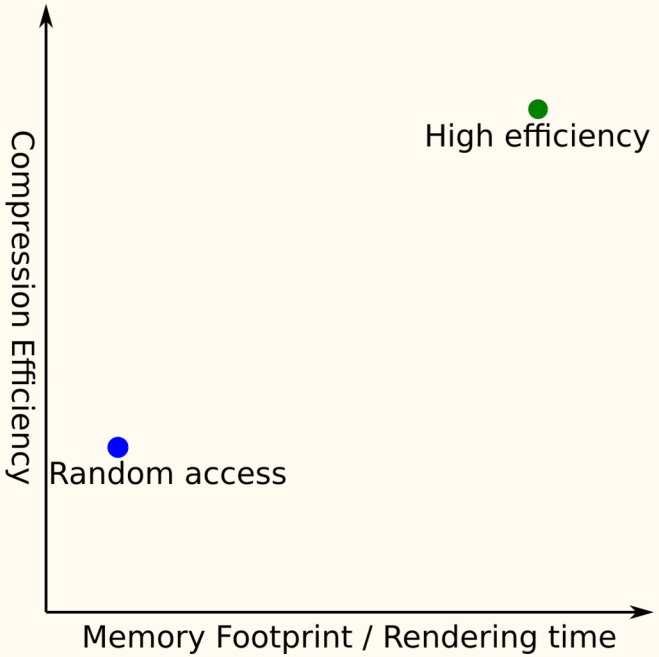


Organization

- Introduction
- **Background**
- Our Approach – HMLFC
- HMLFC Implementation
- Results
- Limitations & Future Work

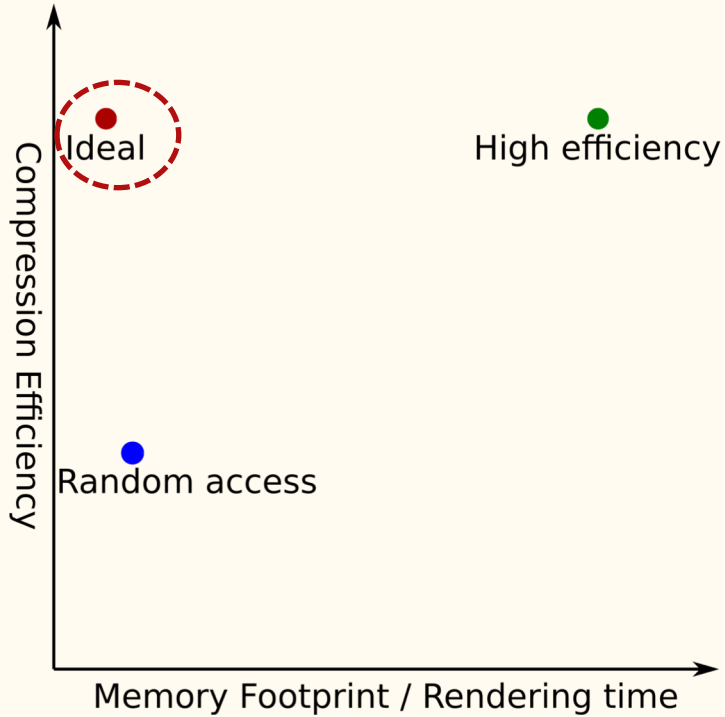
Background: LF Compression

- High efficiency compression schemes
  - Modified MPEG and JPEG schemes
  - Girod et al. [2003] Magnor et al. [2000] Chen et al. [2018] Liu et al. [2016] Perra et al. [2016] : ~ 100 – 500X
- Schemes that enable random access
  - Vector quantization [Levoy et al. 1996] : ~ 40X
  - Wavelet based hierarchy [Peter et al. 2001] : ~ 20-40X
  - MRF with Just-in-time [Zhang & Li 2000] : ~ 80X
  - RLFC [Pratapa & Manocha 2019] : ~ 20 – 200X



LF Compression: Ideal Requirements

- High efficiency compression schemes
- Random Access
- Ideal compression scheme
  - High compression efficiency
  - Low memory footprint

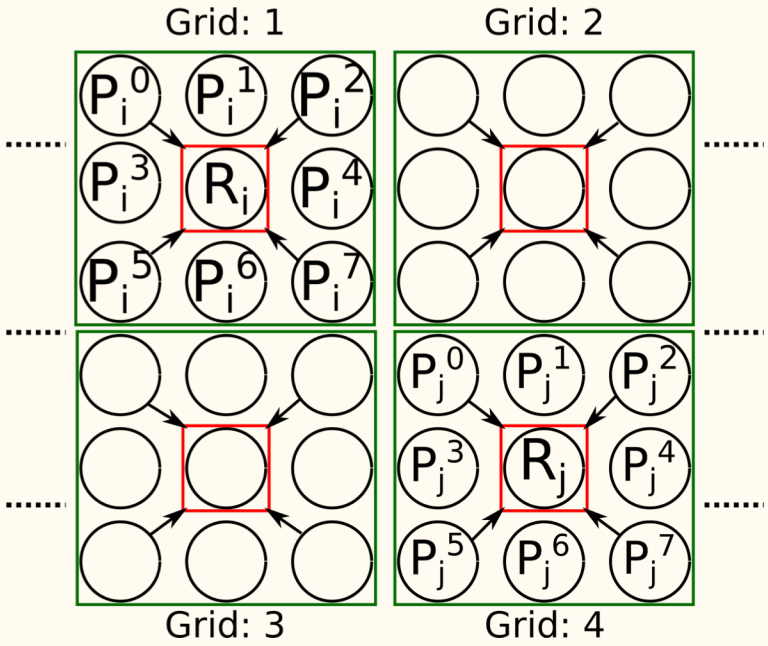


LF Compression: Motion vs. Hierarchy

- Motion compensation compression schemes
  - Girod et al. [2003]; Magnor et al. [2000]; Chen et al. [2018]; Liu et al. [2016]; Perra et al. [2016] : ~ 100 – 500X
  - MRF with Just-in-time [Zhang & Li 2000]: ~80X

- Hierarchical compression schemes
  - Wavelet based hierarchy [Peter et al. 2001] : ~ 20 – 40X
  - RLFC [Pratapa & Manocha 2019] : ~ 20 – 200X
  - Hierarchical coding of light fields [Magnor & Girod 1999] : ~ 40 – 100X

Overview: Motion Compensation



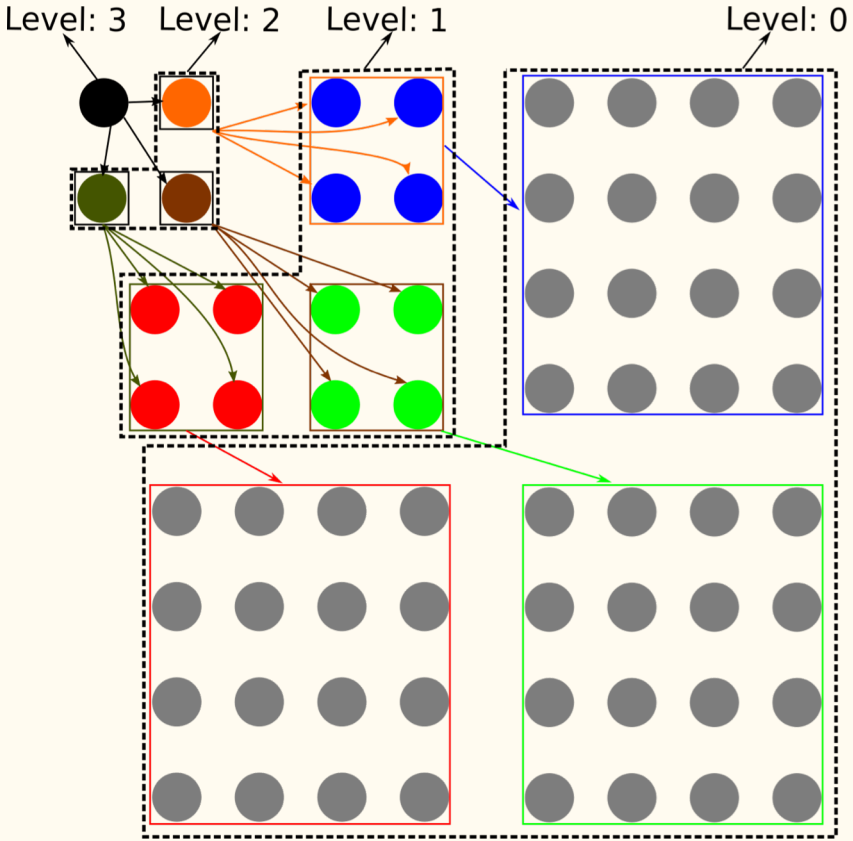
: Light Field Image

R_i : Reference Images

$\{P_i^0, P_i^1, \ldots, P_i^7\}$: Predictive Images

- Redundancy in grids:
  - R_i <--> $\{P_i^0, P_i^1, \ldots, P_i^7\}$
- Redundancy between grids?
- Redundancy between (R_i <--> R_j)
- Redundancy between (R_j <--> P_i^k)

Overview: Hierarchical Schemes



- The entire LFI are transformed using image processing and manipulations
- Captures redundancies across all the LFI
- Redundancy captured is limited

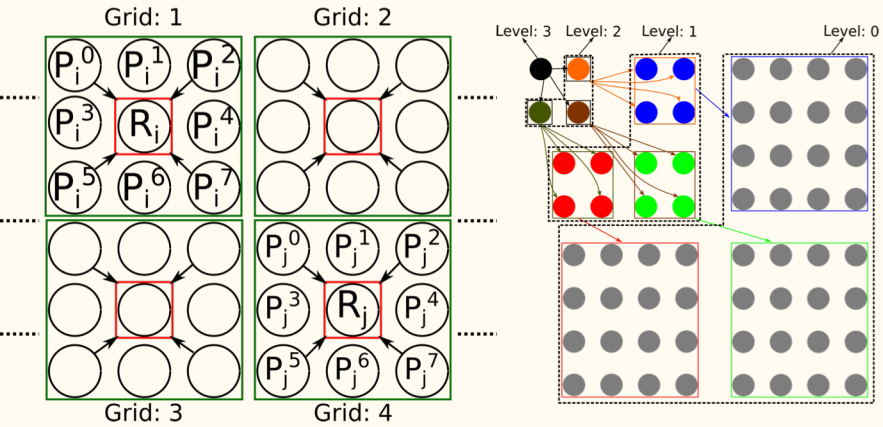
LF Compression: Categories

Motion compensation	Hierarchical compression
• Exhaustive search for redundancy using motion vectors	• Image transformations and manipulation to separate redundancies
• Efficiently exploit the **local coherence** among LFI	• Exploit the **global coherence** among LFI
• Select reference images from LFI & predict the rest using motion vectors	• Build a hierarchy of new parent & child images using the transformations

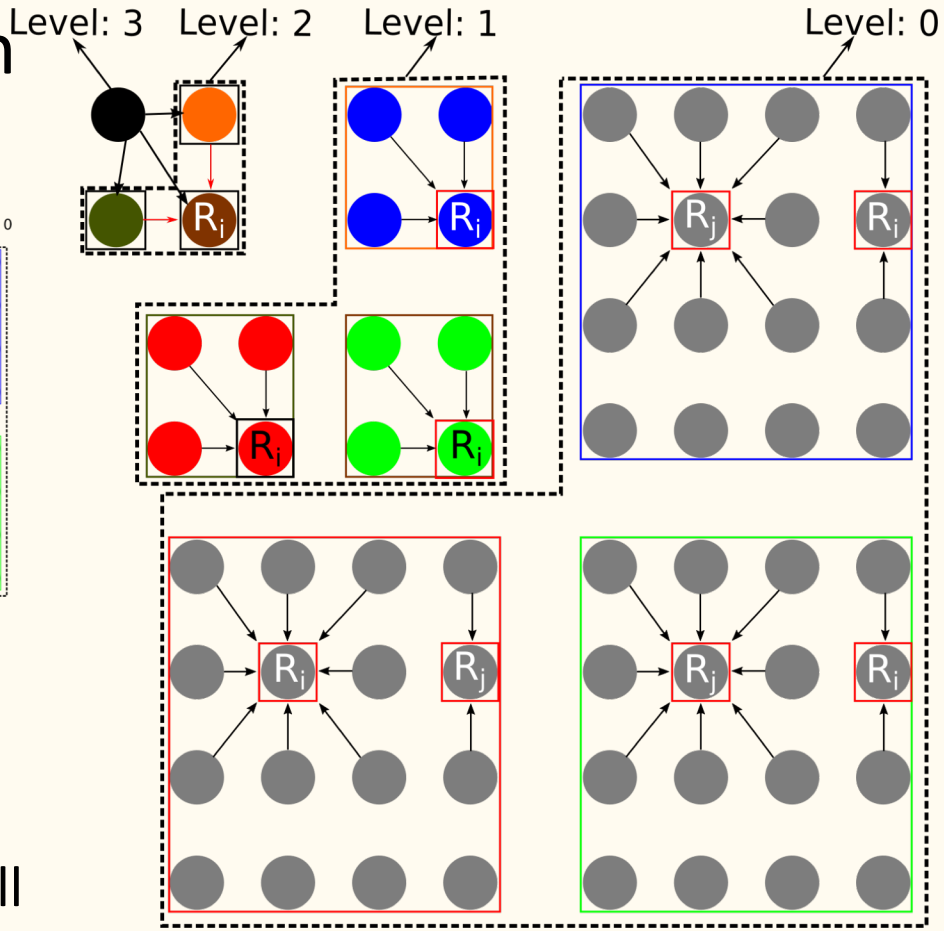
Organization

- Introduction
- Background
- **Our Approach – HMLFC**
- HMLFC Implementation
- Results
- Limitations & Future Work

Key Idea: Hybrid Approach



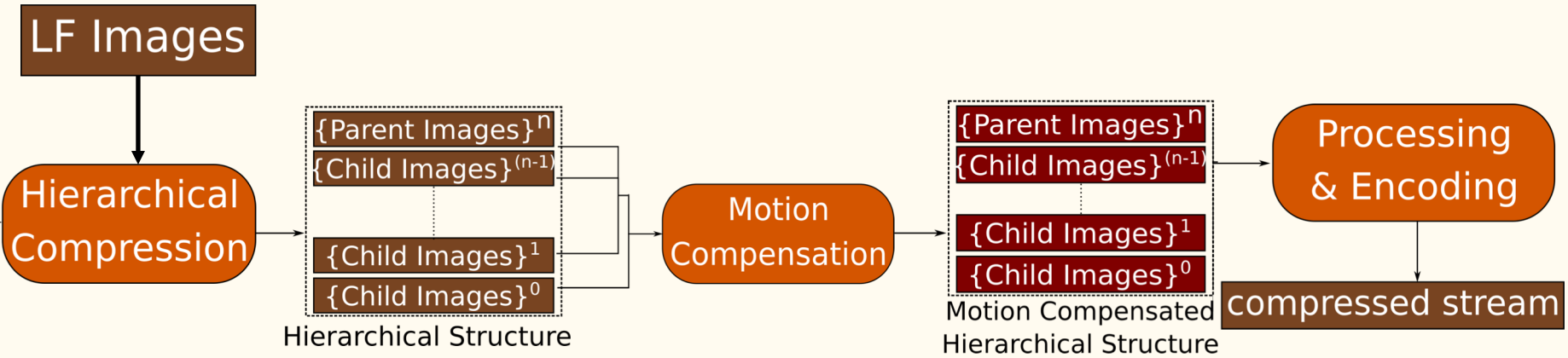
R_i : Reference Images
$\{P_i^0, P_i^1, \ldots, P_i^7\}$: Predictive Images



- Apply motion compensation at all levels

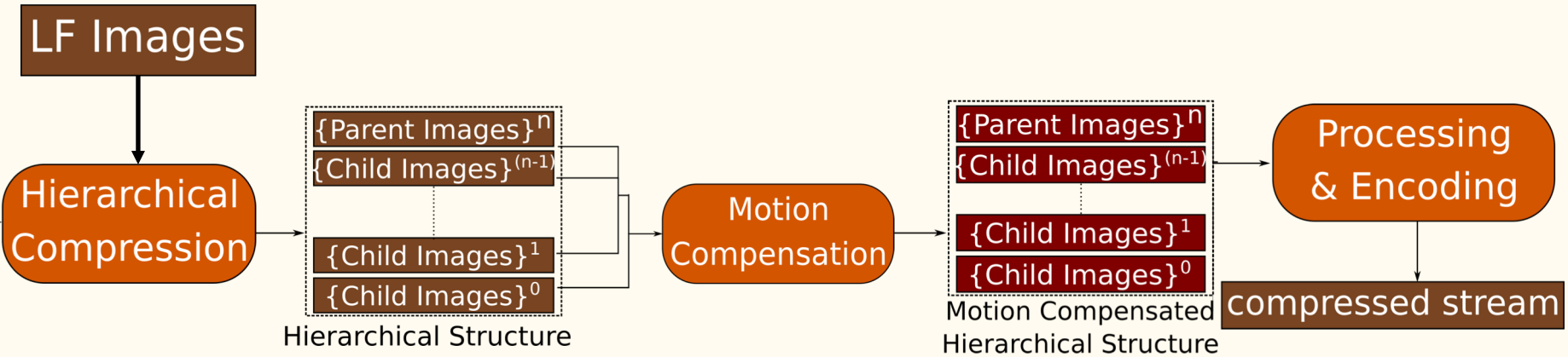
HMLFC : Hierarchical Motion-compensated Light Field Compression

- Motion compensation schemes and hierarchical schemes capture different kind of redundancies
- Optimize the compression by merging different redundancies in both approaches



HMLFC: Main Challenges

- The structure of the hierarchy should remain the same
- New motion compensation method for transformed images
- Over-heads should be minimal



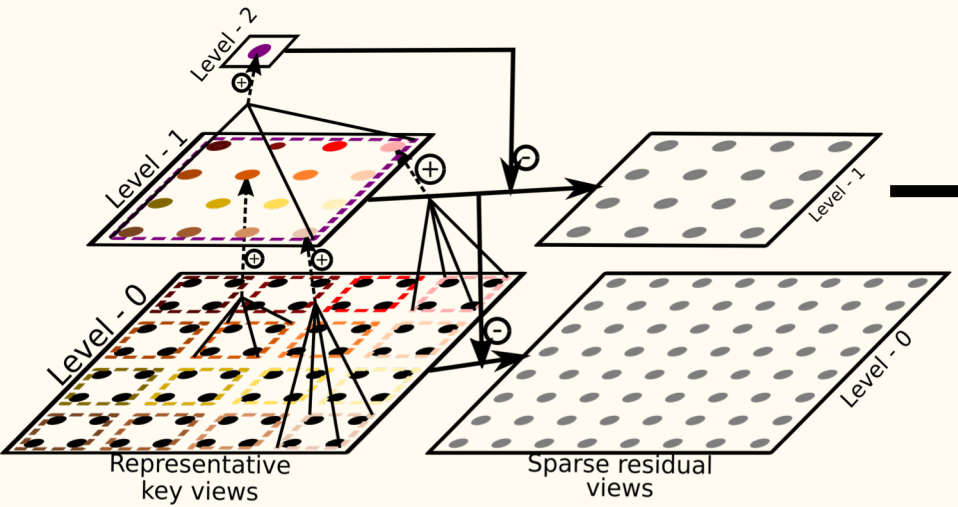
Organization

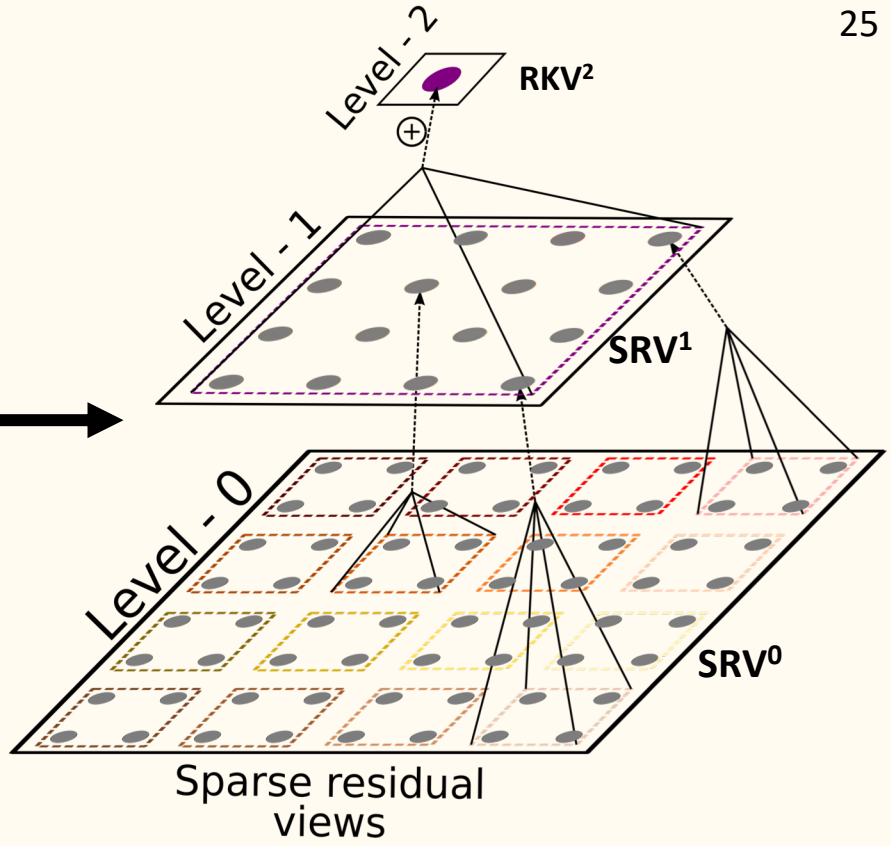
- Introduction
- Background
- Our Approach – HMLFC
- **HMLFC Implementation**
- Results
- Limitations & Future Work

HMLFC Implementation on RLFC

- RLFC [Pratapa & Manocha 2019] as the hierarchical scheme
- **RKV:** Representative Key Views
  - Redundancies across the LF
- **SRV:** Sparse Residual Views
  - Specific details of the LF
- Encoding the **RKV** & **SRV**

RLFC Overview





- Light Field Images

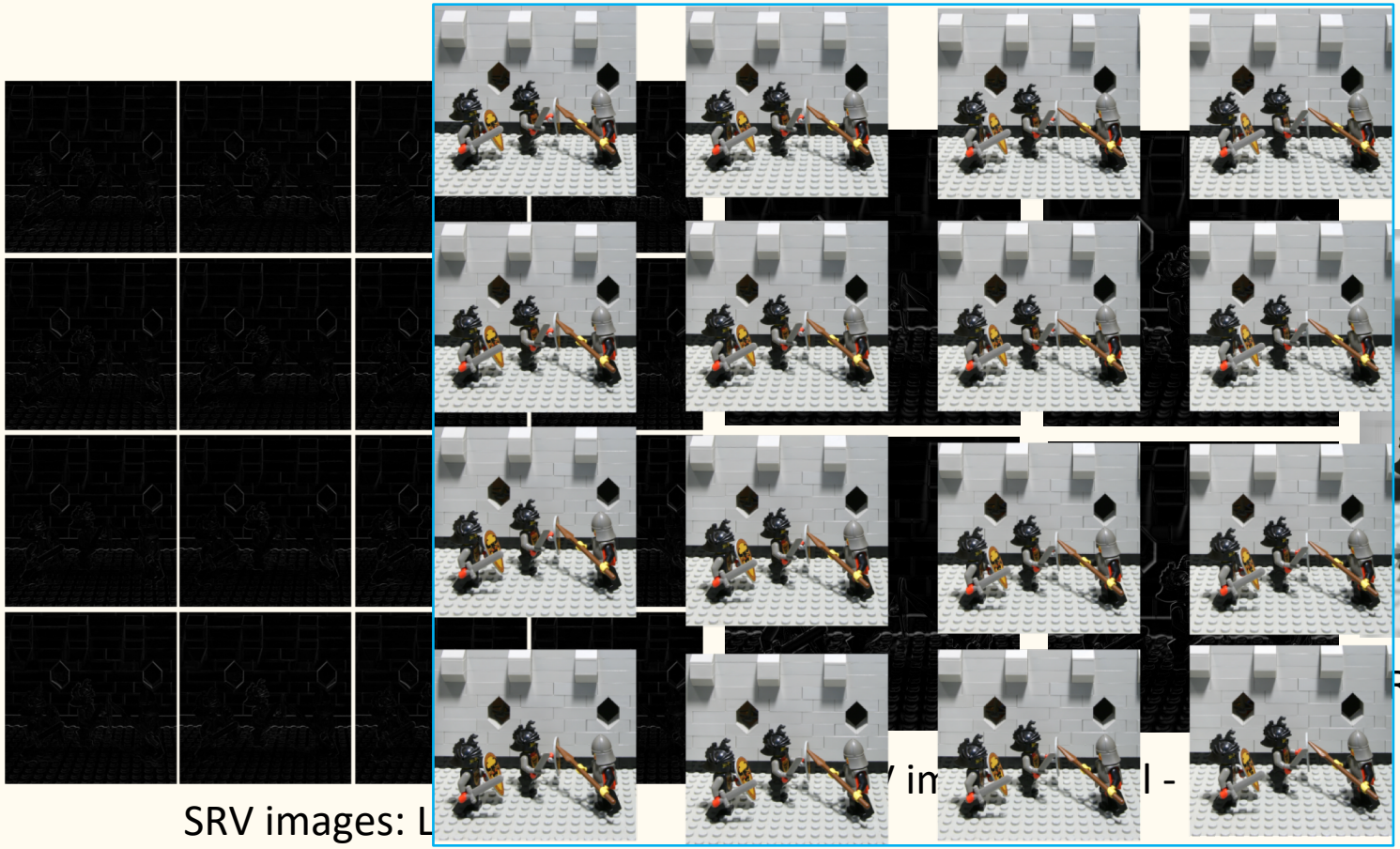
$$RKV_j^l = \sum_{I \in C_j^{(l-1)}} w_{jI}^{(l-1)} \times I.$$

RKV: Representative Key Views

$$(SRV_i)^{(l-1)} = (RKV_i)^{(l-1)} - (RKV_p)^l$$

SRV: Sparse Residual Views

RLFC Overview

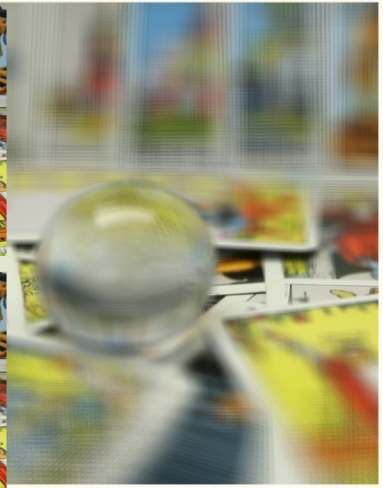


SRV images: L

RKV image: Level - 2

RLFC Overview

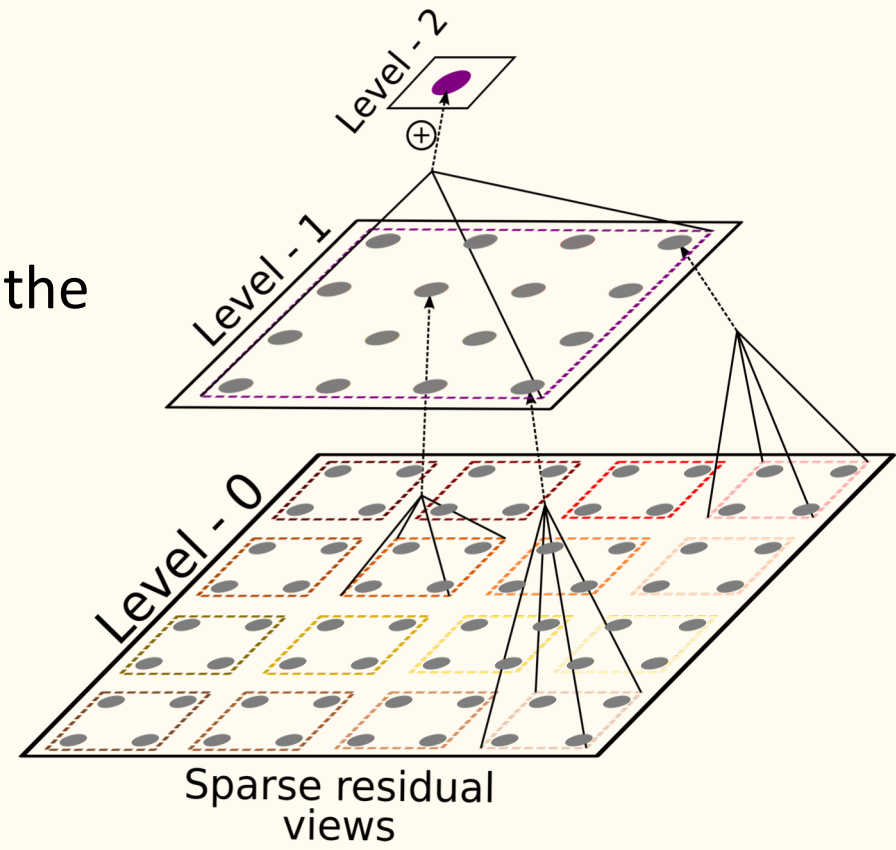
SRV images: Leve



RRV image: Level - 2

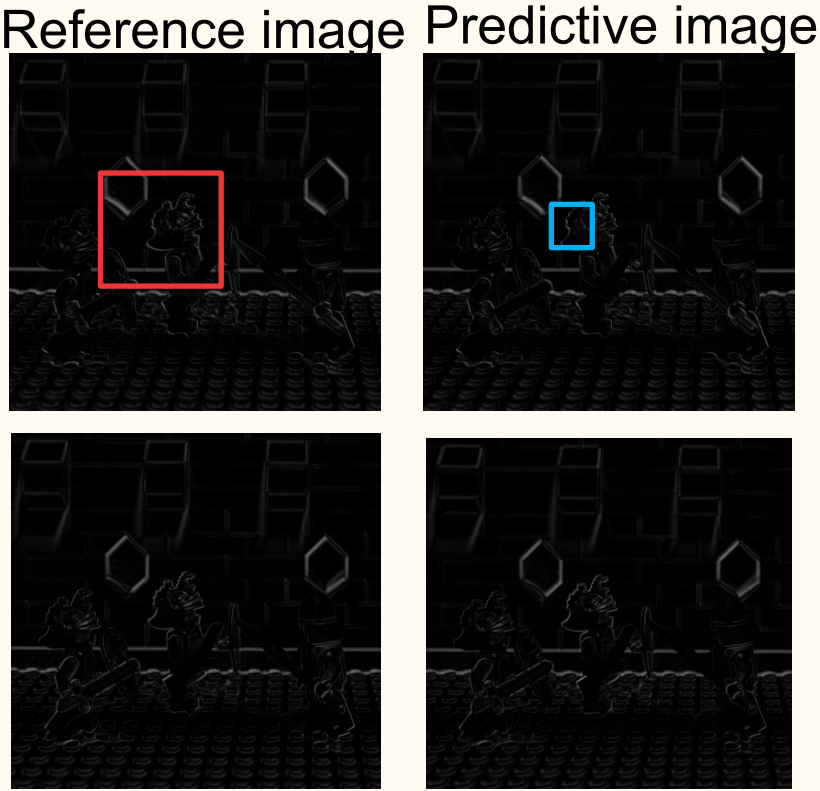
Hybrid approach for RLFC

- Compute the RLFC hierarchy
- Apply motion compensation to all the level of hierarchy



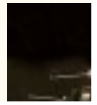
HMLFC: Extending RLFC with Motion Vectors

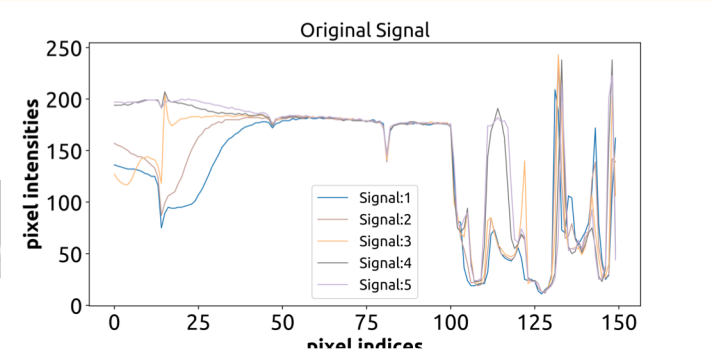
- Lot of redundancy between SRV images
- Block based motion compensation
- Small blocks in **predictive image** are motion compensated
- Search in a given window in a **reference image**
- Find the best matching block

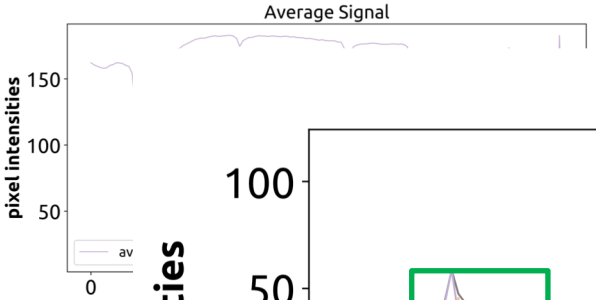


Phase-shifted Motion Prediction

- Series of shifted 2D – signals







- $\Delta = Pred - \mathrm{Ref}$
- $(m_x,\ m_y)$

- $\Delta = Pred + \mathrm{Ref}$
- $(m_x,\ m_y)$

HMLFC: SRV Images

- The data in the motion compensated SRV images is significantly less than original SRV images
- Reference frame is marked in red

Original SRV images

Motion compensated SRV images

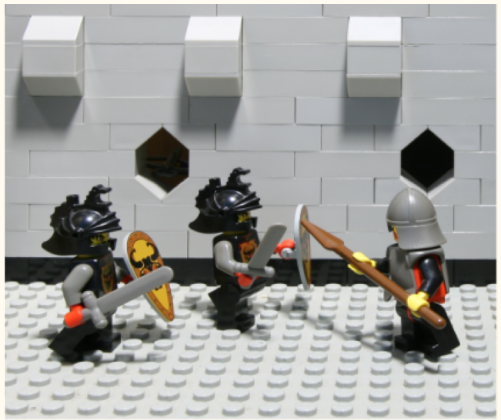
Organization

- Introduction
- Background
- Our Approach – HMLFC
- HMLFC Implementation
- **Results**
- Limitations & Future Work

Results: Data Sets

- Data sets – Stanford new LF archive

Amethyst



Lego Knights

Tarot Cards

Bunny

Results: Compression rate/Quality

LF Dataset (Resolution): Size (MB)	Compression rate (bpp)	PSNR (dB)
Amethyst $(16 \times 16 \times 768 \times 1024) : 576$	0.045	40.7
Bracelet $(16 \times 16 \times 1024 \times 640) : 480$	0.143	40.1
Bunny $(16 \times 16 \times 1024 \times 1024) : 768$	0.027	41
Jelly Beans $(16 \times 16 \times 1024 \times 512) : 384$	0.029	40.5
Lego Knights $(16 \times 16 \times 1024 \times 1024) : 768$	0.157	41
Lego Gallantry $(16 \times 16 \times 640 \times 1024) : 480$	0.155	40.1
Tarot Cards $(16 \times 16 \times 1024 \times 1024) : 768$	0.68	40.3

- Compression ratio: ~ 30 – 800X
- Analysis on the Stanford LF archive

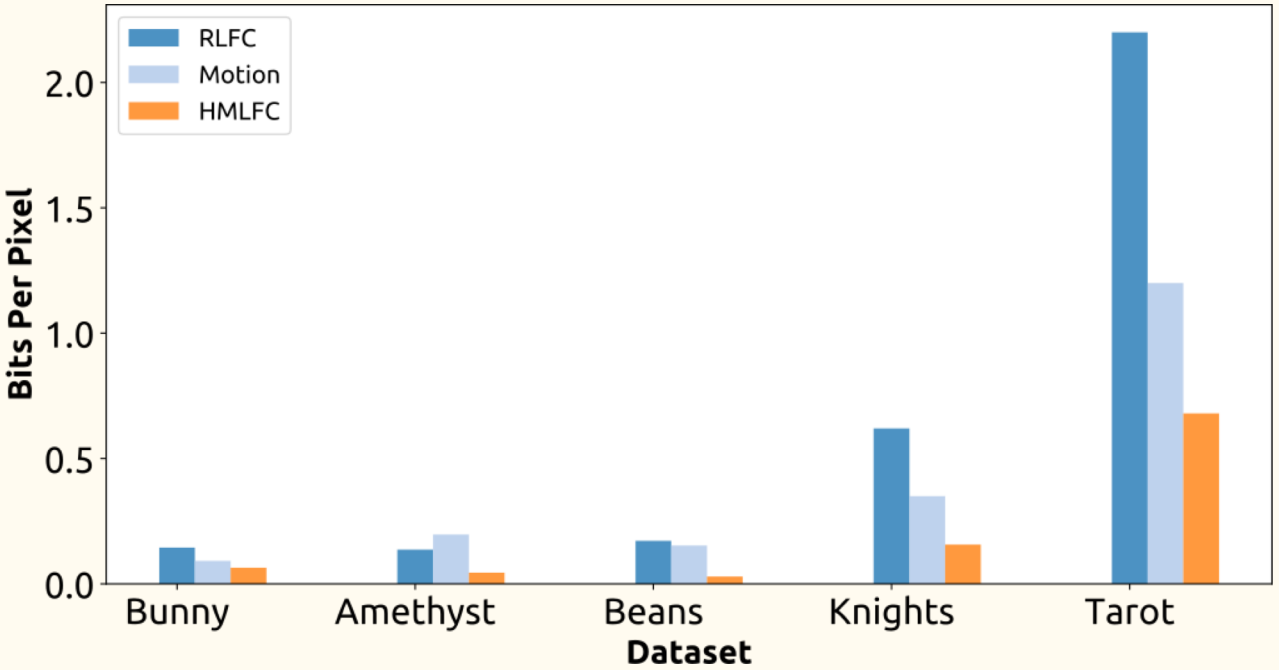
HMLFC: Comparison

- Similar compression quality
- Improvement factor: ~ 2 – 5X

Data set	RLFC (bpp)	HMLFC (bpp)	Improvement factor
Bunny	0.145	0.073	**1.9X**
Amethyst	0.137	0.045	**3X**
Bracelet	0.52	0.143	**3.6X**
Jelly Beans	0.172	0.029	**5.5X**
Lego Knights	0.62	0.157	**4X**
Tarot Cards	2.20	0.68	**3.2X**

HMLFC: Comparsion

- Lower is better
- HMLFC does better than both RLFC and motion-compensation



HMLFC: Visual Quality

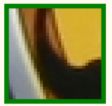
RLFC HMLFC


PSNR: 40.9 PSNR: 40.7
BPP: 0.137 BPP: 0.045 (3X)

RLFC HMLFC


PSNR: 41.5 PSNR: 41
BPP: 0.145 BPP: 0.073 (2X)

RLFC HMLFC

PSNR: 41.7 PSNR: 41
BPP: 0.62 BPP: 0.157 (4X)

RLFC HMLFC


PSNR: 40.9 PSNR: 40.3
BPP: 2.20 BPP: 0.68 (3.2 X)

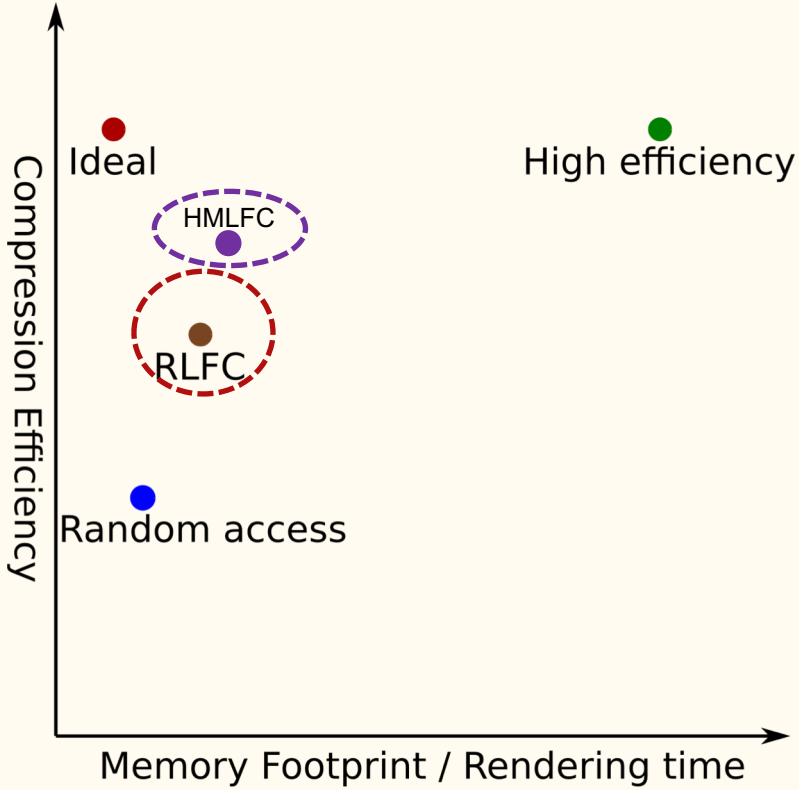
- Visual quality in comparison with RLFC
- Similar quality of compression (PSNR)
- Visual quality same as RLFC

Results: Rendering Speeds

- Generate new views using LF renderer with GPU decoder
- Hardware used: NVIDIA GTX – 960 (2GB) and Intel Xeon (2.4GHz)
- Image resolution (512 X 512)
  - Avg. frames per second: ~200 fps
- Image resolution (1024 X 1024)
  - Avg. frames per second: ~110 fps

HMLFC: Conclusion

- HMLFC: Hierarchical Motion-Comepensated Light Field Compression
- RLFC as the underlying hierarchical scheme
- Phase-shifted motion compensation to all the levels of the hierarchy
- Factor of 2-5X improvement in compression rates for the same quality



Organization

- Introduction
- Background
- Our Approach – HMLFC
- Results
- **Limitations & Future Work**

HMLFC: Limitations

- Benefits are limited to finding a suitable motion-compensation method
- No fine grain control over the encoding parameters
- Unoptimized GPU decoder in terms of memory operations

HMLFC: Future Work

- Extend to other LF parameterizations (spherical & unstructured)
- Use sub-pixel motion compensation to search for a matching block
- Extend the current ideas for light field videos
- Integrate our method in a end-to-end LF rendering system

Acknowledgements

- The research was supported in part by Intel

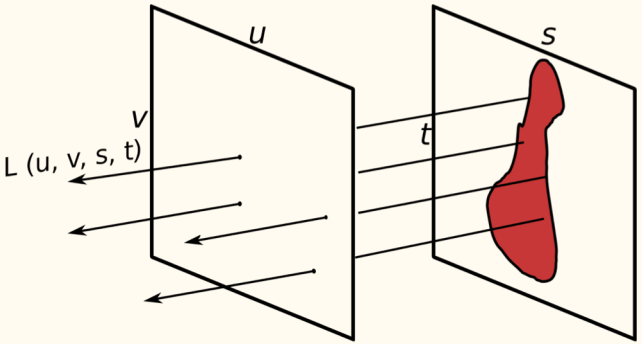
- Reviewers for great feedback

Thank you!

Questions?

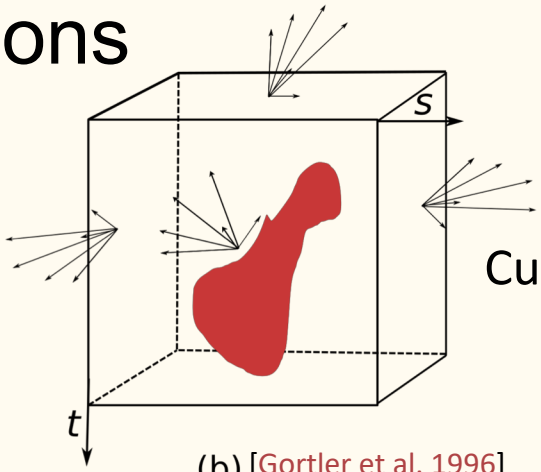
BACK-UP SLIDES

Light Fields: Parameterizations



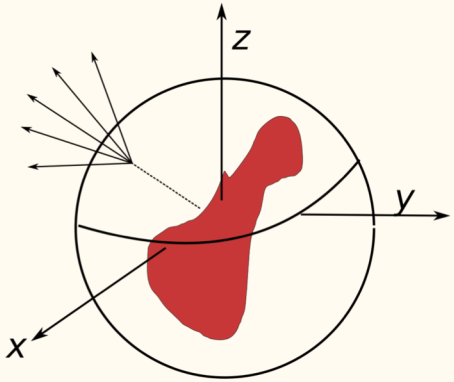
2 Planes

(a) [Levoy et al. 1996]



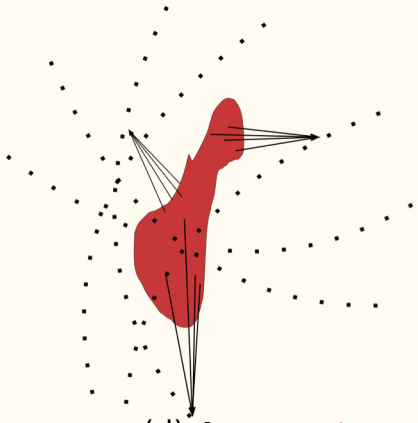
Cube: 6 planes

(b) [Gortler et al. 1996]



Spherical

(c) [Ihm et al. 1997]

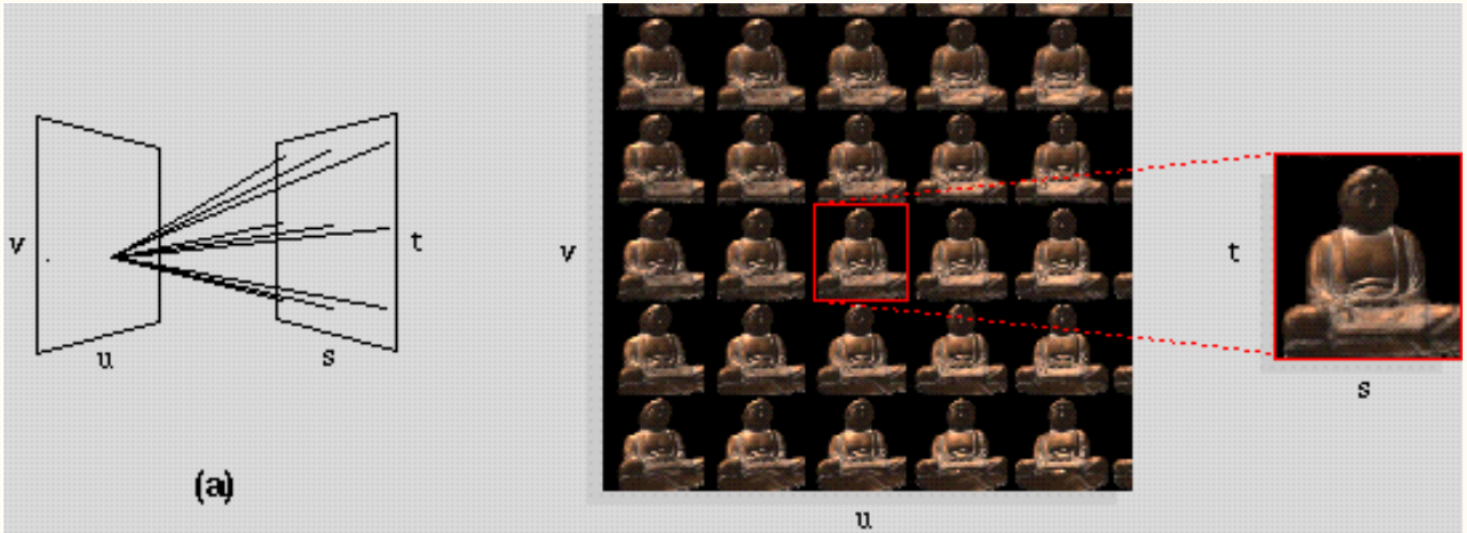


UnStructured

(d) [Davis et al. 2011]

Sampling of the Light Rays

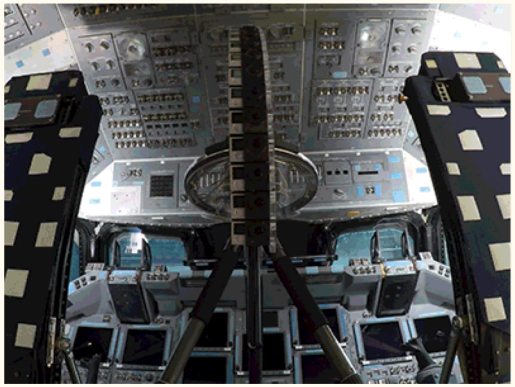
- Sample by taking 2D photographs

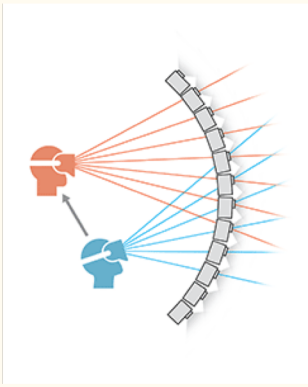


Levoy et al. 96

Light Fields: Revival

- Recent developments in capturing & displays
- Google – Welcome to Light Fields[Overbeck et al. 18]
- Spherical LF representation (360^0 FOV)
- Image samples captured: 8K; Image resolution: 1K; Data size ~8GB





“VR is still a novelty, but Google’s light-field technology could make it serious art”- MIT Technology review

Encoding SRV

- Sparse high frequency images
- Discard insignificant blocks
- Bounded Integer Sequence Encoding (BISE)
- Encodes integer values using BISE from ASTC [Nystad et al. 12]

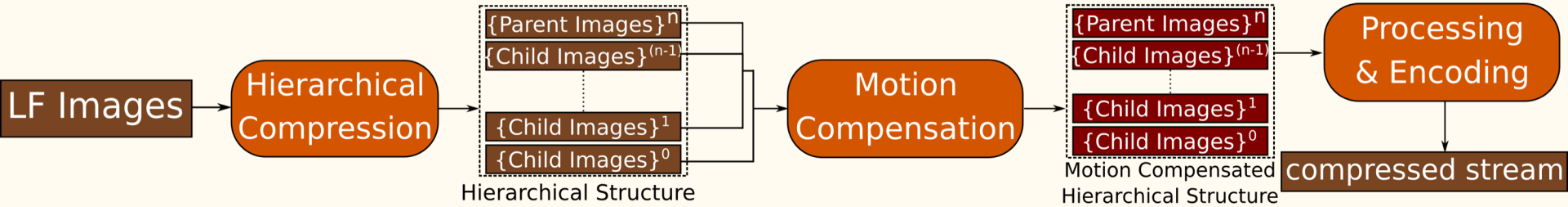
Encoding RKV

- Properties are similar to standard color images

- Compress using JPEG2000

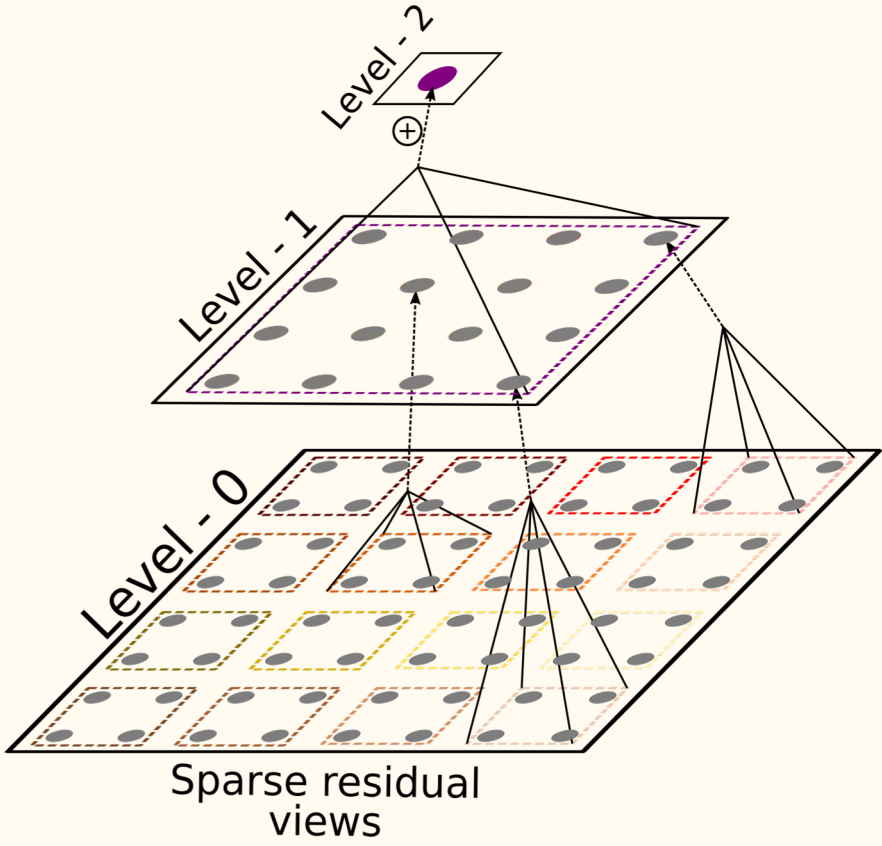
HMLFC : Hierarchical Motion-compensated Light Field Compression

- Improve the compression by combining the orthogonal dependencies
- A hybrid approach to combine both the approaches



HMLFC: Decoding properties

- The hierarchical structure of RLFC remains the same in HMLFC
- Tree traversal decoding to decode the motion compensated blocks
- One additional motion re-compensation step to compute original block
- Decoding properties – Random access, parallel access are preserved



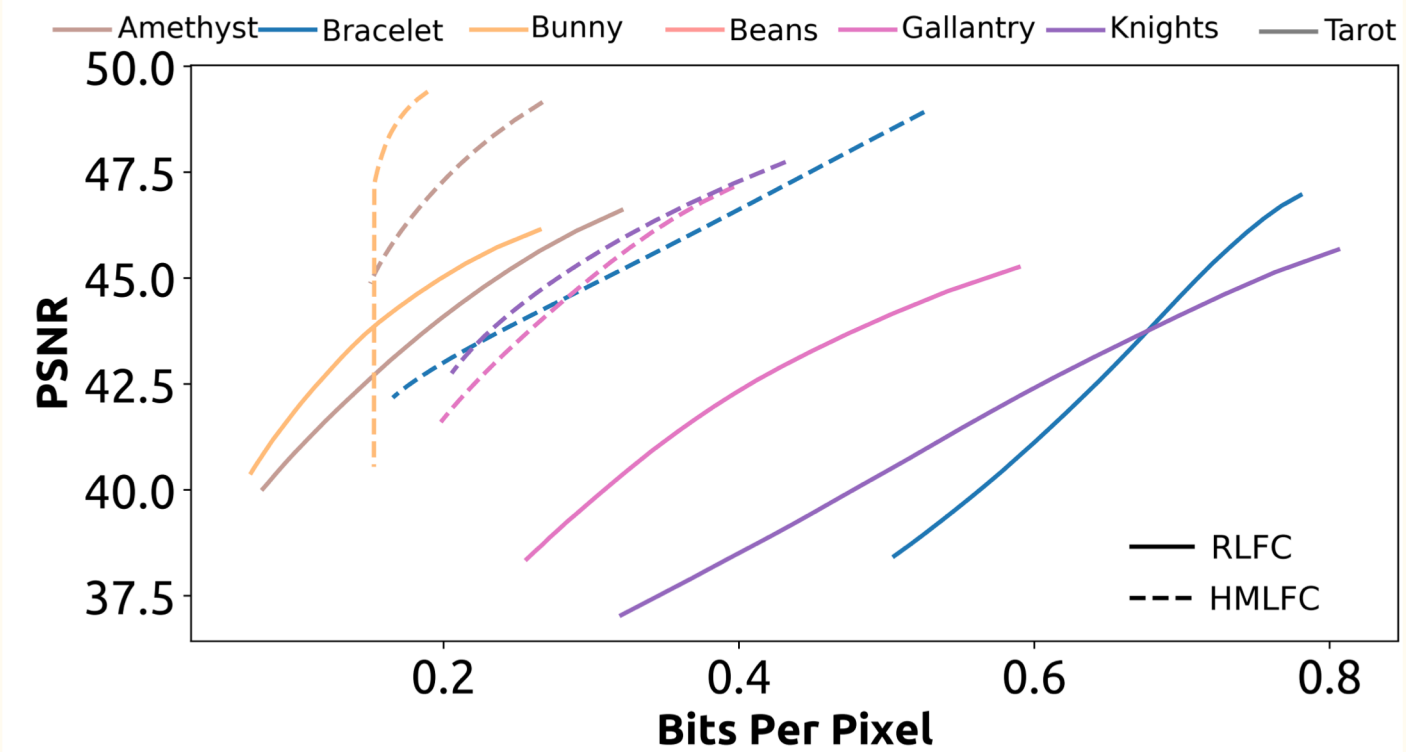
HMLFC: Rate Distortion

- Variation with block size

LF Dataset	Metric	Block Size: 2	Block Size: 4	Block Size: 8
Amethyst	PSNR	38.7	43.69	48.35
	bpp	0.0592	0.106	0.707
Bunny	PSNR	40.52	43.35	47.52
	bpp	0.0173	0.0411	0.548
Bracelet	PSNR	37.03	44.15	48.79
	bpp	0.033	0.35	1.108
Knight	PSNR	38.27	43.046	47.89
	bpp	0.096	0.243	1.15
Beans	PSNR	37.3	44.53	49
	bpp	0.0067	0.052	0.377
Gallantry	PSNR	37.22	42.54	46.95
	bpp	0.0235	0.235	1.08
Tarot	PSNR	36.98	42.14	46.79
	bpp	0.272	0.831	2.6

HMLFC: Rate Distortion

- Variation with block thresholding



HMLFC: Decoding times

- YCoCg-r color space [Malvar et al. 2008]
- Block size 8X8

Channel	RLFC (microseconds)	HMLFC (microseconds)
Y - Channel	2.61	3.32
Co - Channel	1.62	2.86
Cg - Channel	1.42	2.21

NVIDIA GTX – 960 (2GB) and Intel Xeon (2.4GHz)