A Hardware-Friendly Bilateral Solver for Real-Time Virtual-Reality Video

Amrita Mazumdar — **University of Washington**
Armin Alaghi
Jonathan T. Barron — **Google**
David Gallup
Luis Ceze
Mark Oskin — **University of Washington**
Steven M. Seitz

virtual reality video with omnidirectional stereo (ODS)

the Google Jump camera rig can capture ODS video easily

16 GoPros x 4K camera feed
3.6 GB/s raw video

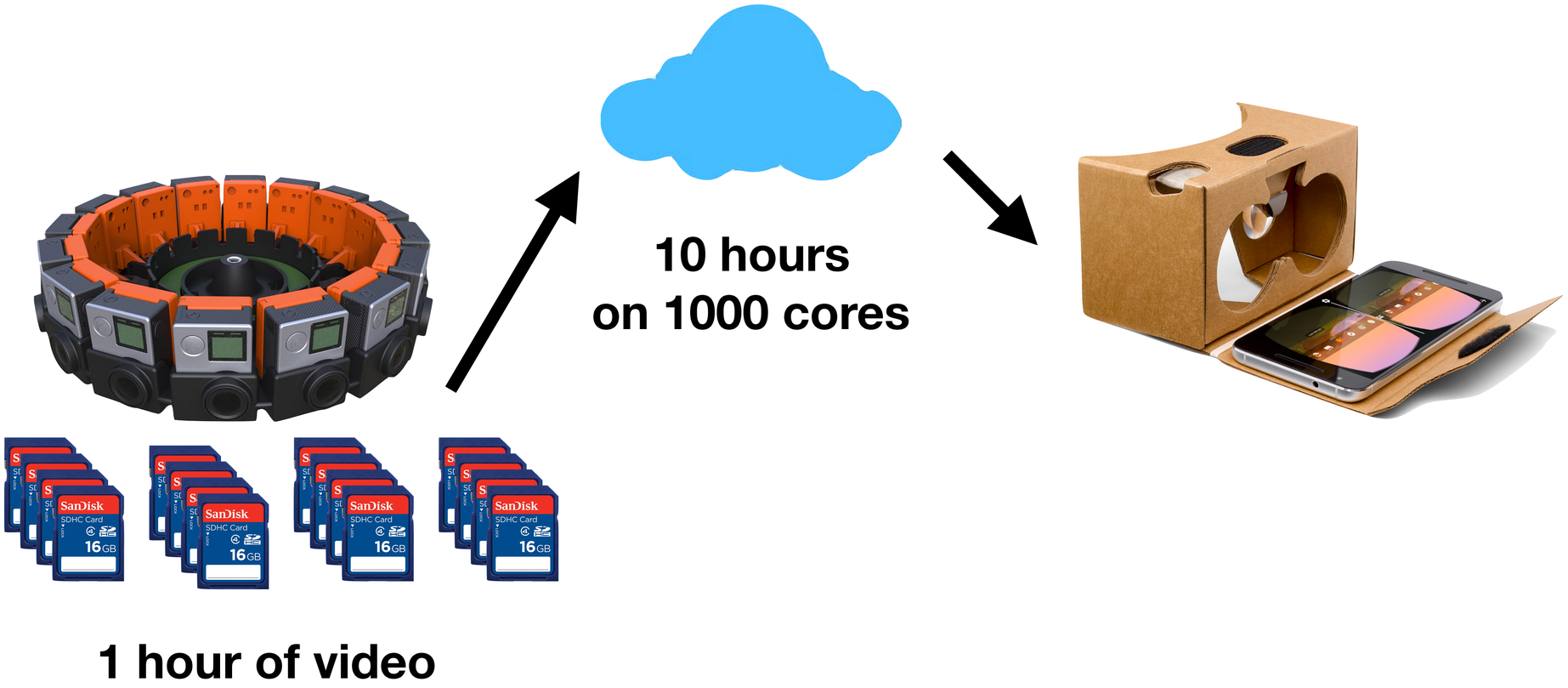
the Google Jump camera rig can capture ODS video easily

Anderson et al., SIGGRAPH Asia 2016

the Google Jump camera rig can capture ODS video easily

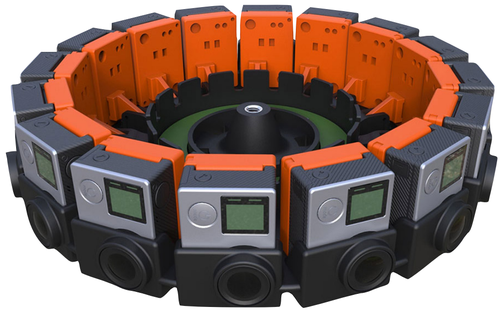
Anderson et al., SIGGRAPH Asia 2016

processing video from Google Jump is slow



Anderson et al., SIGGRAPH Asia 2016

Google Jump pipeline breakdown



sensor

pre-processing

alignment

optical flow

compositing

download to viewer

Anderson et al., SIGGRAPH Asia 2016

Google Jump pipeline breakdown

the bilateral solver dominates processing time

sensor	pre-processing	alignment	optical flow	compositing	download to viewer
	12%	2%	69%	17%	

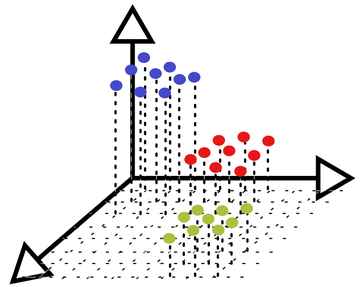
Anderson et al., SIGGRAPH Asia 2016

The bilateral solver produces an image that is smooth and accurate.

input pair (from two cameras)

blocky flow field

upsample into noisy flow field



transform to bilateral grid and solve

output result: smooth flow field

Anderson et al., SIGGRAPH Asia 2016

this work:

a hardware-friendly bilateral solver (HFBS)

The bilateral solver is hard to parallelize

second-order global optimization

global communication prevents aggressive parallelization

high-dimensional, sparse matrices

sparsity results in significant divergence on GPUs

why not a dense grid? too large to store on-chip

HFBS is easier to parallelize

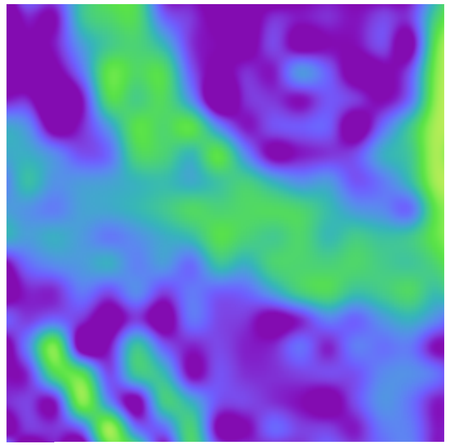
detailed formulation in paper

Barron Poole 2016	HFBS (our work)
✅ includes color	grayscale only
dense matrix too big to fit in memory	✅ dense matrix fits in memory
global communication required	✅ local communication only
iterative bistochastization before solving	✅ partial, non-iterative bistochastization

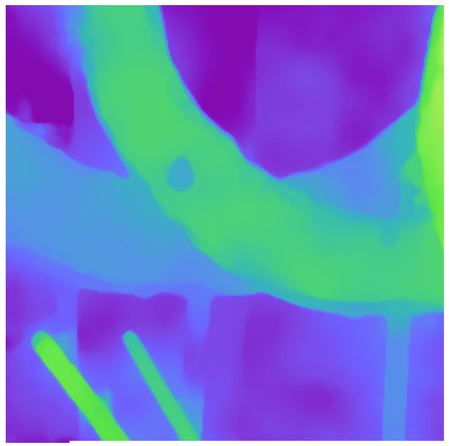
HFBS demonstrates imperceptible accuracy loss

input image

Barron Poole 2016



noisy depth map



HFBS (this work)

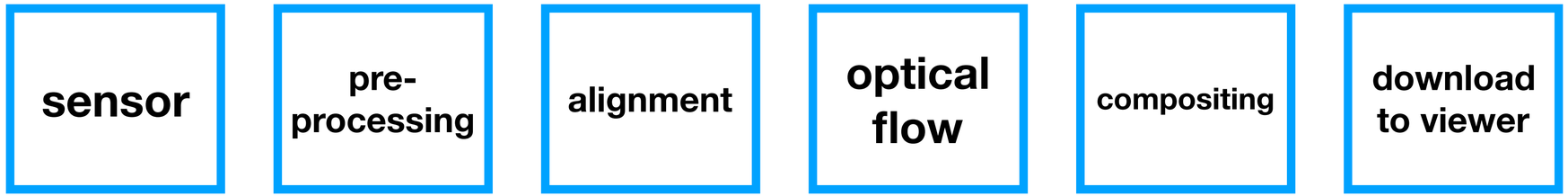
task: Ferstl et al., ICCV 2013,
data: Middlebury stereo dataset

algorithm optimizations make it easier to implement bilateral solver in parallel hardware

algorithm optimizations make it easier to implement bilateral solver in parallel hardware

plan: exploit this parallelism with a custom hardware accelerator

Mapping HFBS to hardware



Mapping HFBS to hardware

sensor | pre-processing | alignment | **optical flow** | compositing | download to viewer

CPU | **FPGA**

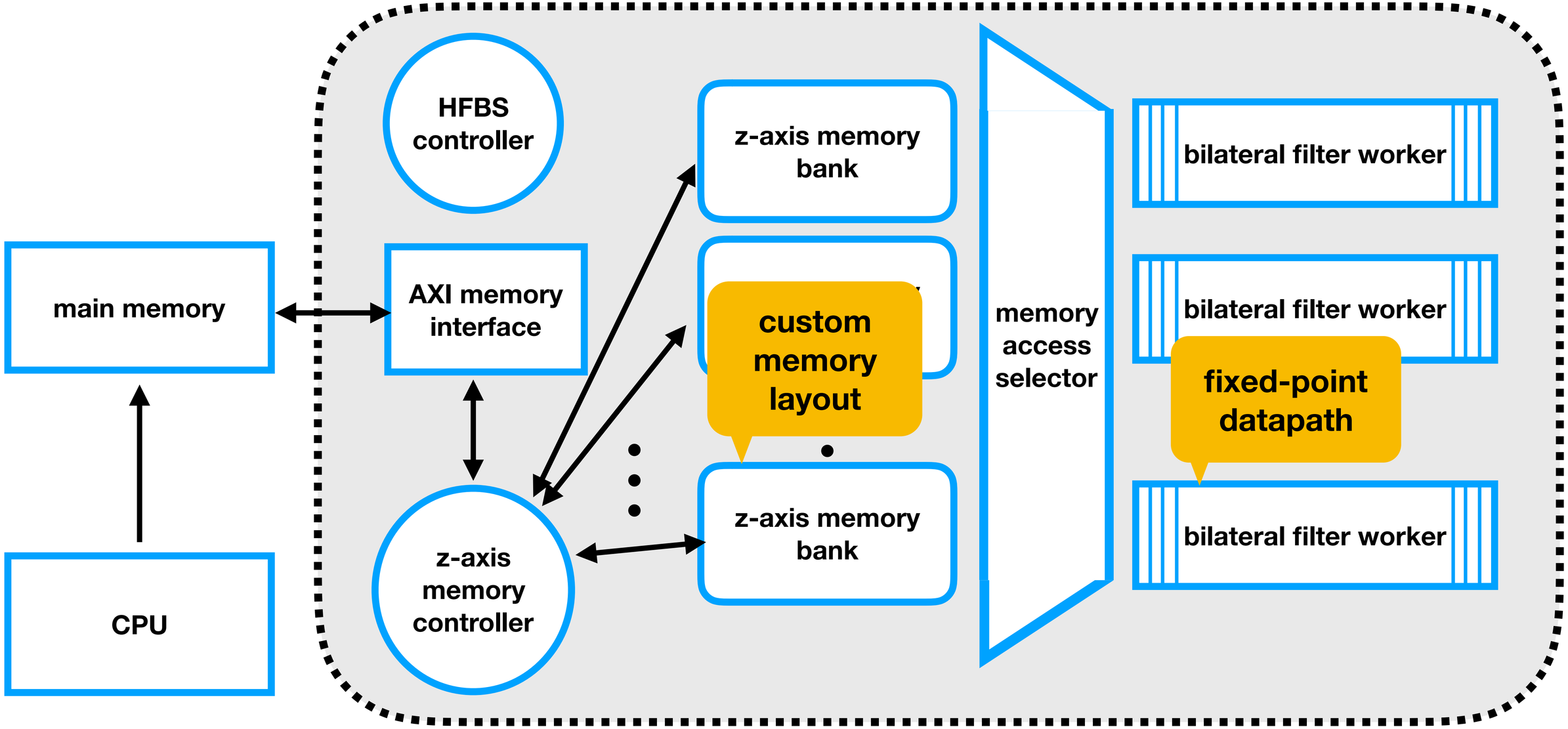
load video pair

construct bilateral grid per pair

perform hardware-friendly bilateral solver

slice out solution into output images

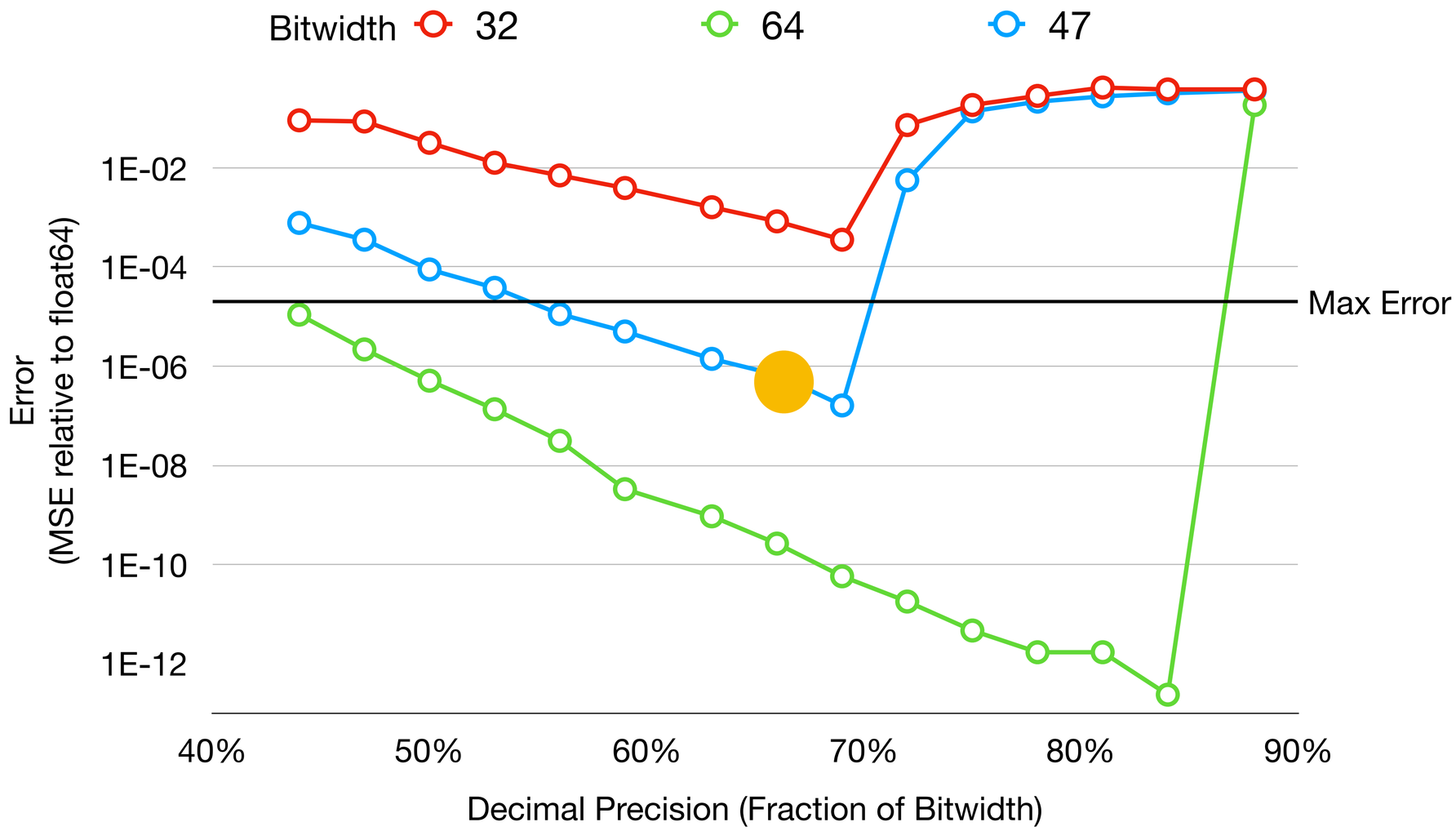
microarchitecture



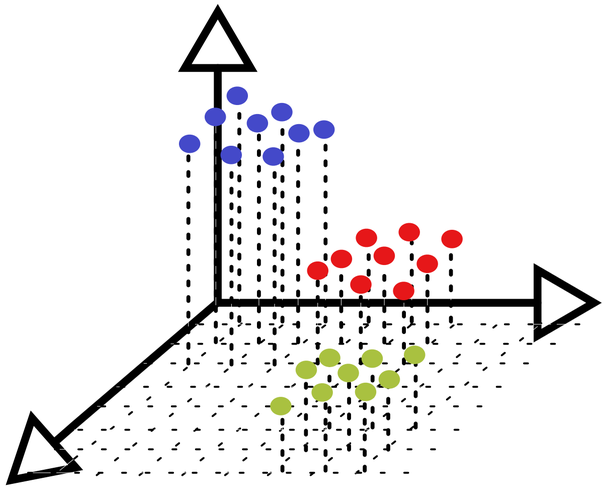
Floating-point resource requirements limit hardware parallelism

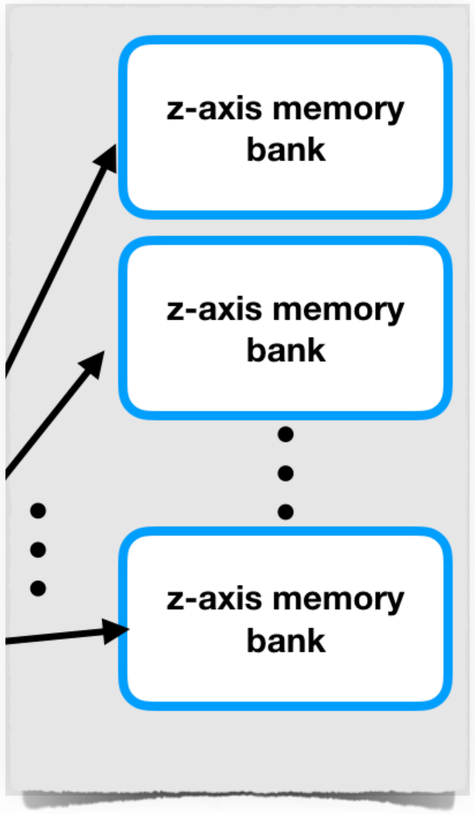
	float64	32-bit fixed	64-bit fixed	47-bit fixed
DSPs per worker	18	1	16	4
Maximum # workers	379	**6840**	**427**	**1710**
Error (MSE)	-	**8.3×10^{-4}**	**7.16×10^{-13}**	**6.69×10^{-7}**

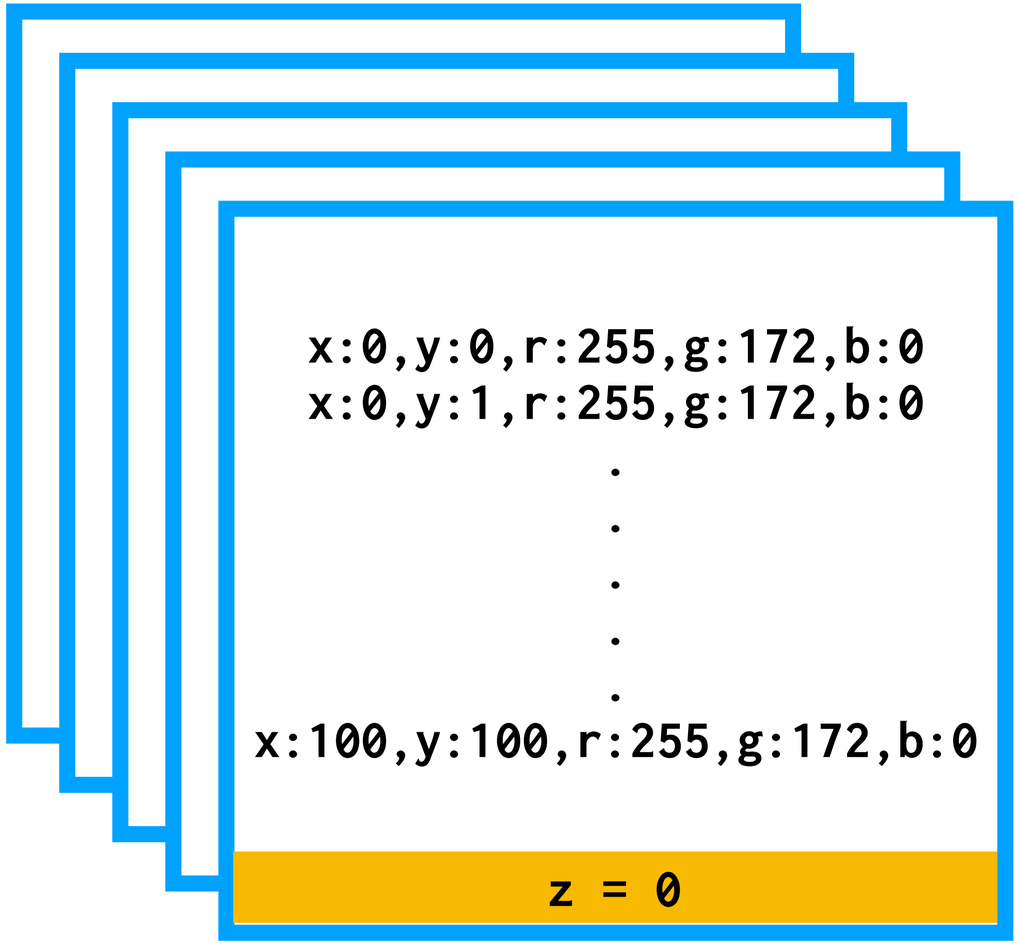
Fixed-point datapath conversion



z-axis slicing for bilateral grid memory layout







Evaluation

Evaluation

Does HFBS improve runtime?
How does parallelization affect power?

sensor	pre-processing	alignment	optical flow	compositing	download to viewer
	12%	2%	69%	17%	

Experimental Setup

CPU: Intel Xeon E5-2620

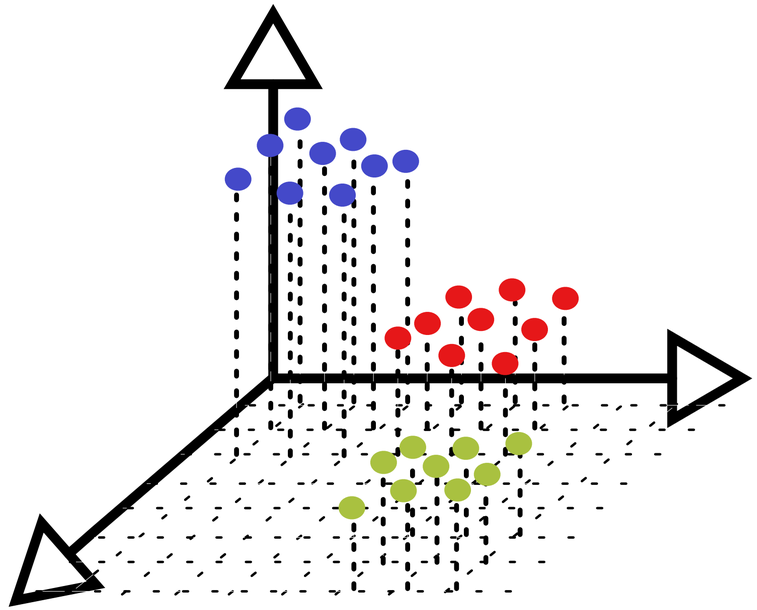
GPU: NVIDIA GTX 1080 Ti

FPGA: Xilinx Virtex Ultrascale+

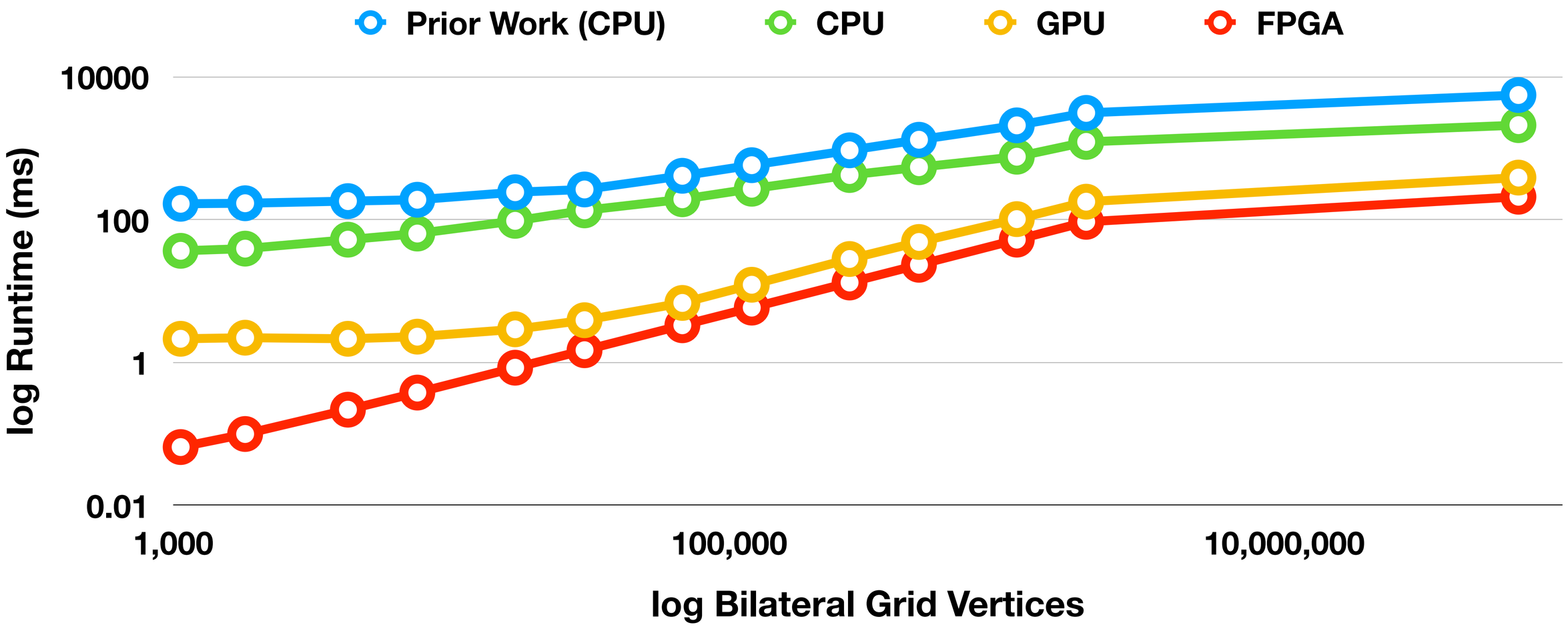
Baseline: Barron Poole et al. 2016 (CPU only)

256 iterations of optimization

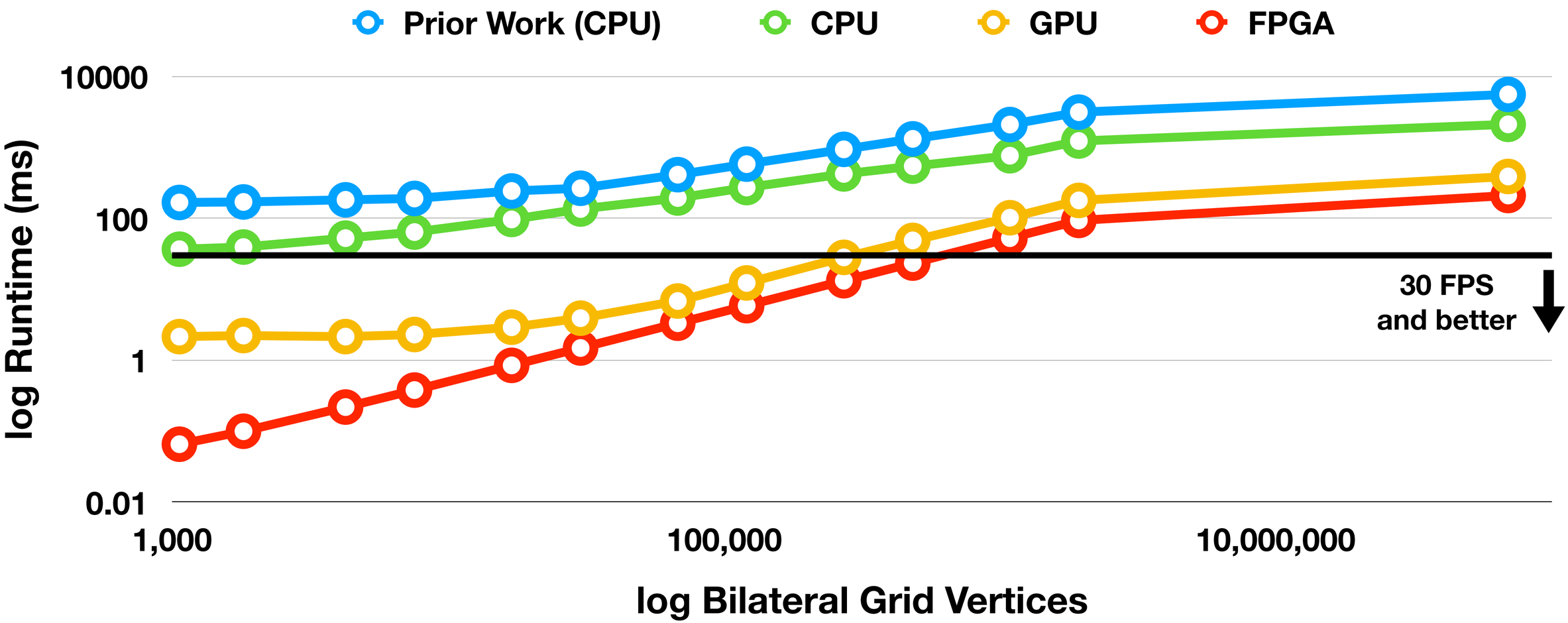
Varied bilateral grid vertices count
⇒ 4 KB - 1.8 GB grid sizes



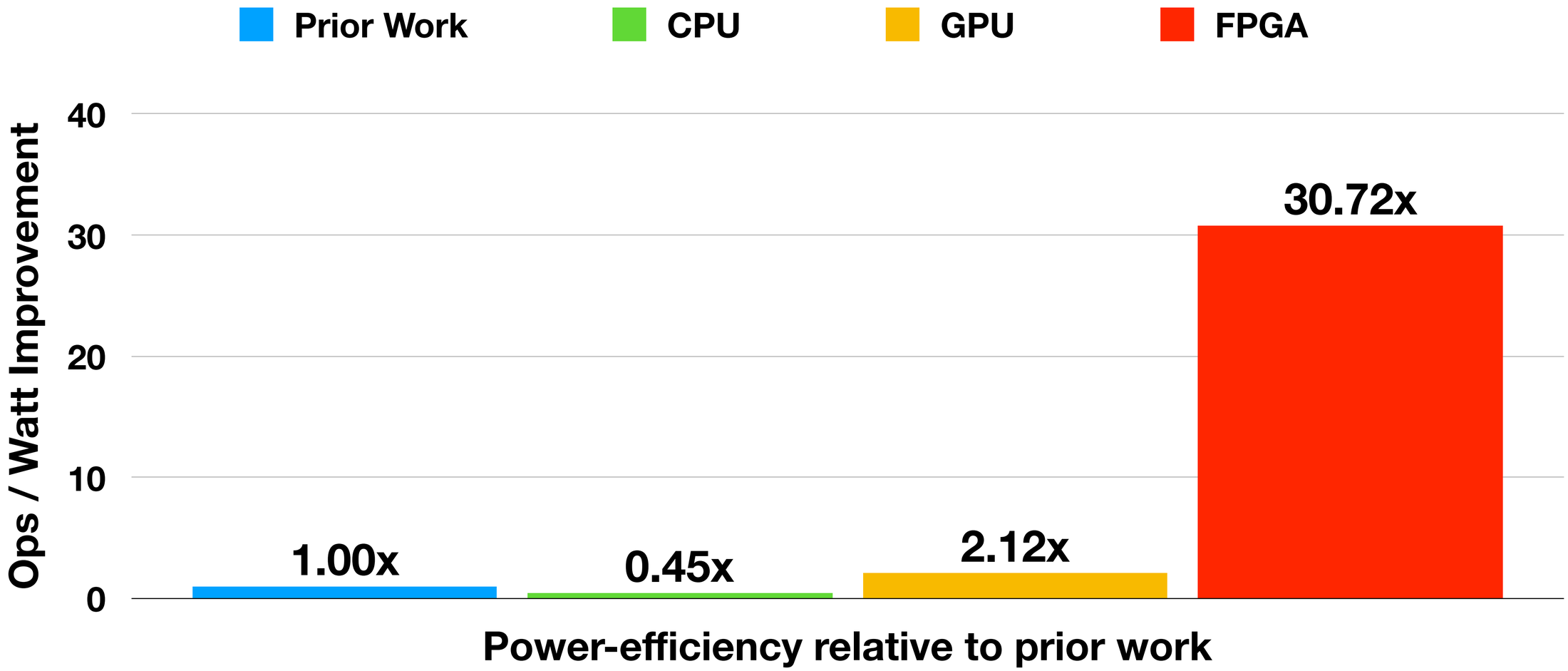
HFBS is faster and more scalable than prior work.



HFBS is faster and more scalable than prior work.



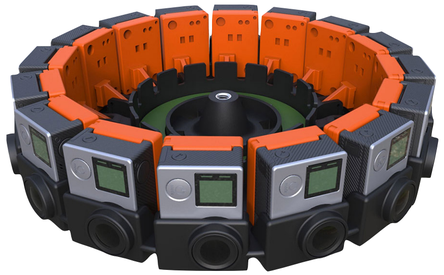
HFBS-FPGA is more power-efficient than other platforms



building a VR video camera rig with HFBS

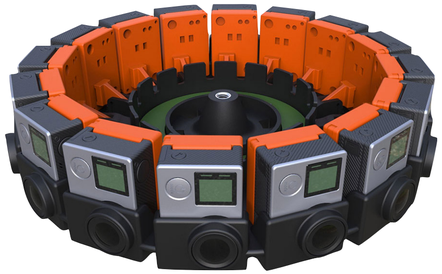
this work

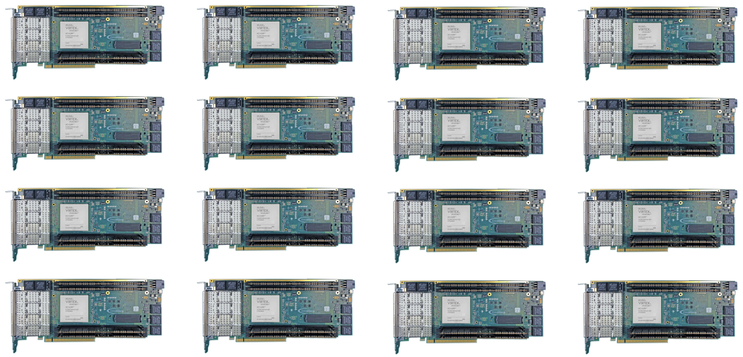
full system



HFBS-FPGA consumes much less power than a GPU for the same task

16 FPGAs = 400 W





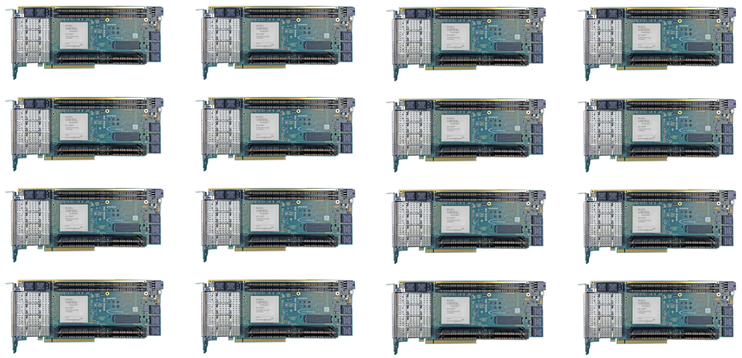
16 GPUs = 4,560 W

HFBS makes real-time VR video more feasible with FPGAs

on-node with FPGAs

~~**offloaded to cloud**~~

sensor	pre-processing	alignment	optical flow	compositing	download to viewer



to conclude

fast, parallel implementation of bilateral solving with little accuracy loss

fixed-point datatypes and a custom bilateral-grid memory layout for improved FPGA performance

hardware-software codesign to reduce latency and improve quality for future VR applications

A Hardware-Friendly Bilateral Solver for Real-Time Virtual Reality Video

parallel algorithm for bilateral solving
FPGA architecture
50x faster, 30x more power-efficient