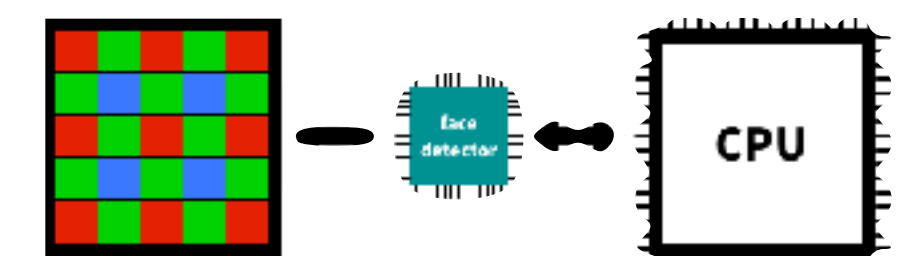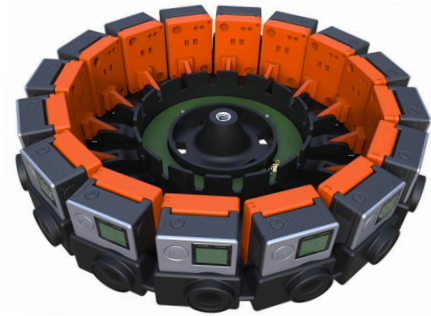# In-camera processing hardware for computer vision and virtual reality

cameras are ubiquitous, the de facto sensor system

modern camera applications are required to be **real-time** and **sophisticated**

shifting computation closer to the camera sensor **saves data movement costs**

Hardware-software co-design enables *power-efficient* and *high-performance* camera processing pipelines
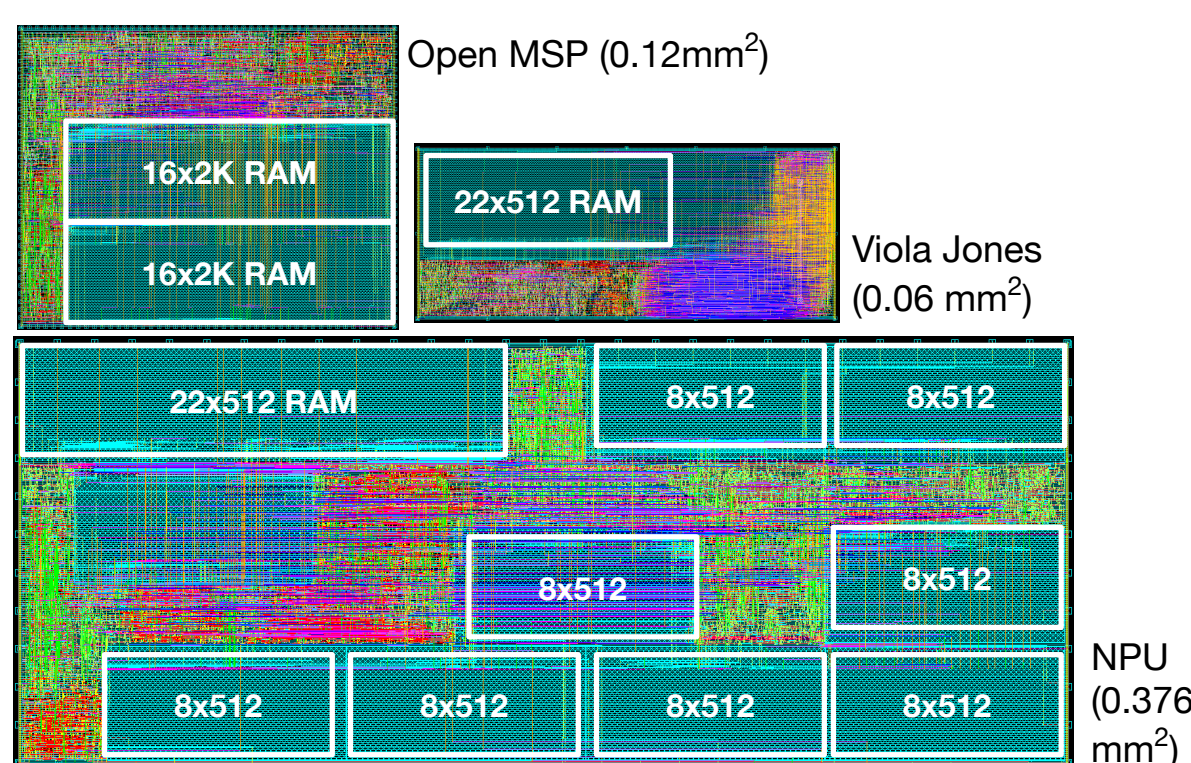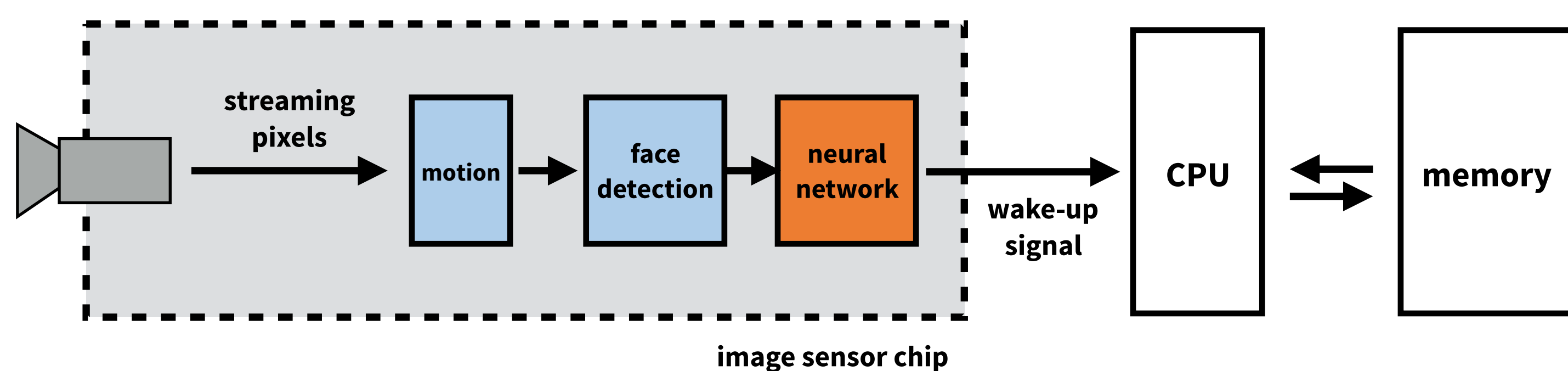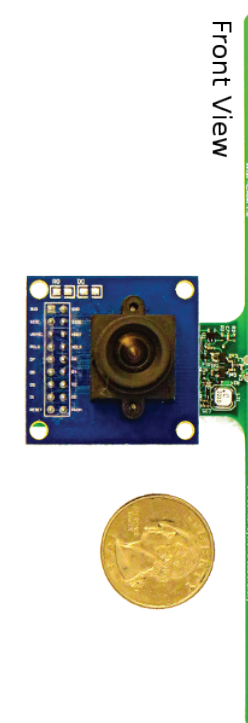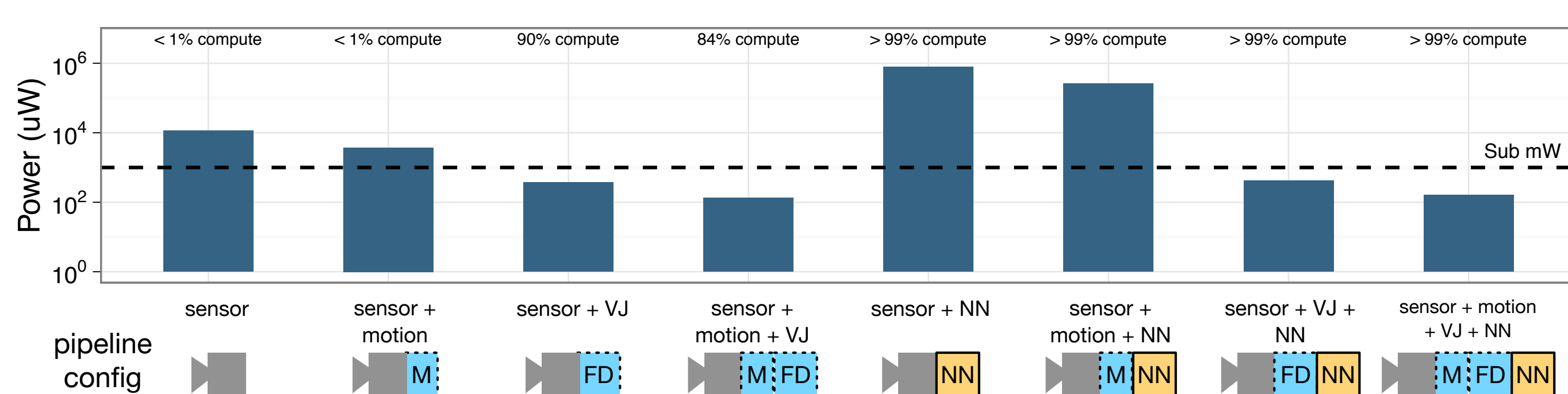
Low-power Face Authentication

Face authentication task:
1. Did we detect any motion?
2. Is there a face?
3. Is this face the one we want?

Goal: mW-scale execution at 1FPS
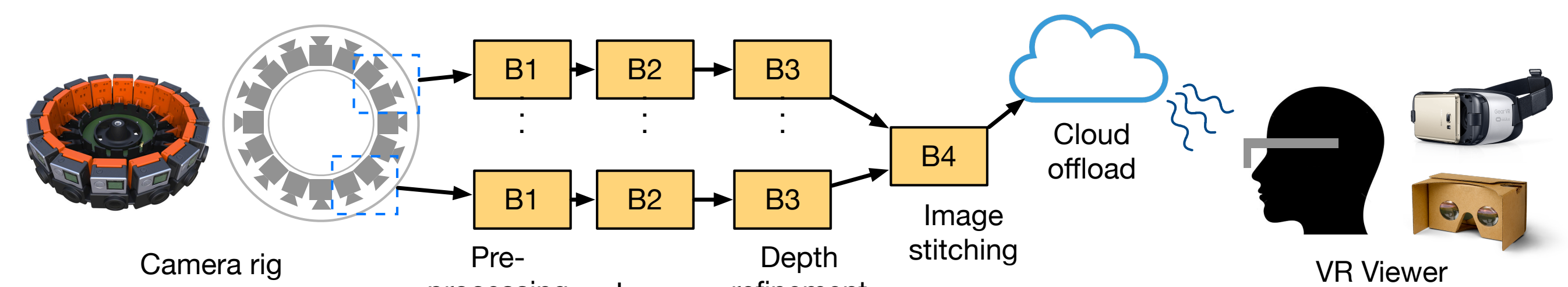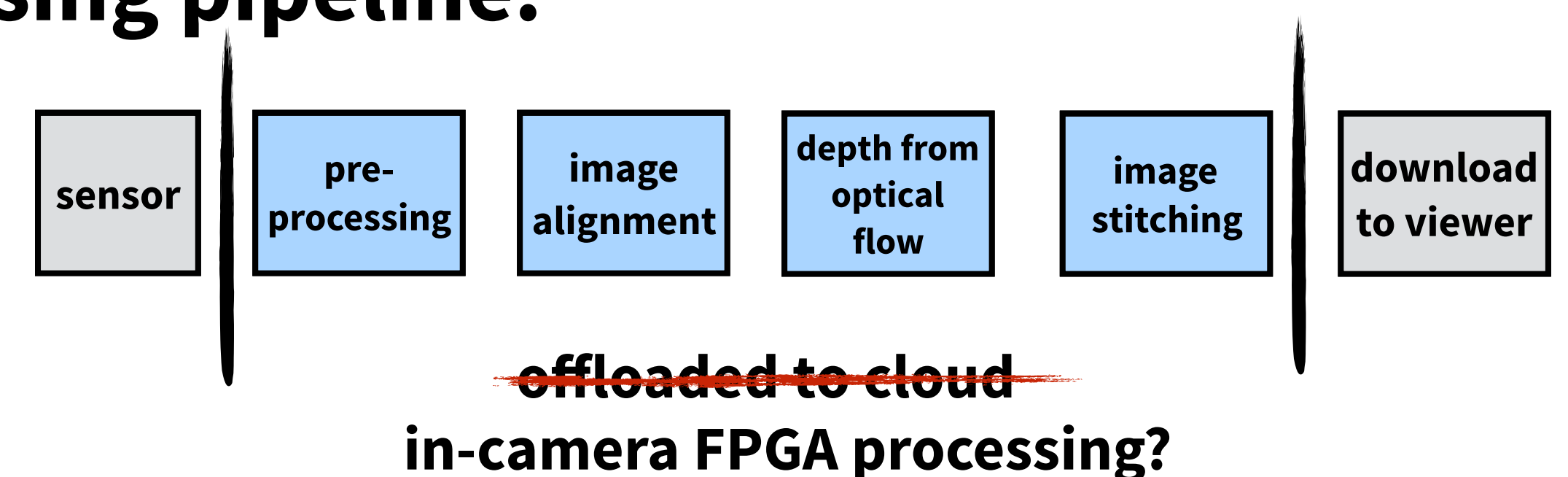
Is this Armin? ✅

streaming pixels → **motion** → **face detection** → **neural network** → *wake-up signal* → **CPU** ↔ **memory**

image sensor chip

Open MSP (0.12mm²)
16x2K RAM
22x512 RAM
Viola Jones (0.06 mm²)
22x512 RAM
8x512
8x512
8x512
NPU (0.376 mm²)
8x512

- Face detection accelerator: 337 µW
- Neural network accelerator: 393 µW
- Full system power: < 1 mW!

Power (uW)

| < 1% compute | < 1% compute | 90% compute | 84% compute | > 99% compute | > 99% compute | > 99% compute | > 99% compute |

Sub mW

pipeline config: sensor | sensor + motion (M) | sensor + VJ (FD) | sensor + motion + VJ (M FD) | sensor + NN (NN) | sensor + motion + NN (M NN) | sensor + VJ + NN (FD NN) | sensor + motion + VJ + NN (M FD NN)

Real-time Virtual Reality

Camera rig → Pre-processing (B1) → Image alignment (B2) → Depth refinement (B3) → Image stitching (B4) → Cloud offload → VR Viewer

processing pipeline:

sensor | **pre-processing** | **image alignment** | **depth from optical flow** | **image stitching** | **download to viewer**
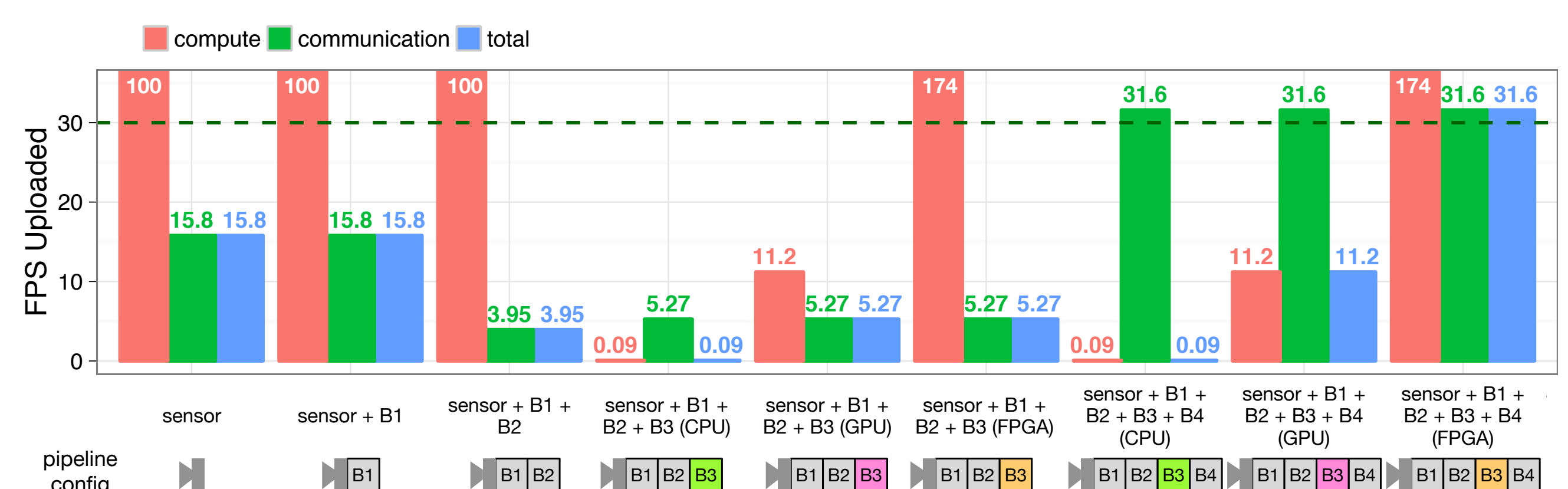
~~offloaded to cloud~~
in-camera FPGA processing?

- Prototyped with Xilinx Zynq SoC
- Evaluated against CPU and GPU implementations
- Software code written and tuned in Halide
- Assumed 2GB/s network link

FPS Uploaded

Legend: compute | communication | total

pipeline config	compute	communication	total
sensor	100	15.8	15.8
sensor + B1	100	15.8	15.8
sensor + B1 + B2	100	3.95	3.95
sensor + B1 + B2 + B3 (CPU)		5.27	5.27
sensor + B1 + B2 + B3 (GPU)	11.2	5.27	5.27
sensor + B1 + B2 + B3 (FPGA)	174	0.09	0.09
sensor + B1 + B2 + B3 + B4 (CPU)		31.6	11.2
sensor + B1 + B2 + B3 + B4 (GPU)	11.2	31.6	11.2
sensor + B1 + B2 + B3 + B4 (FPGA)	174	31.6	31.6

sampa

Amrita Mazumdar   Luis Ceze
Armin Alaghi   Mark Oskin

contact:
amrita@cs.washington.edu
homes.cs.washington.edu/~amrita