A Compact Real-Time Thermal Imaging System Based on Heterogeneous System-on-Chip

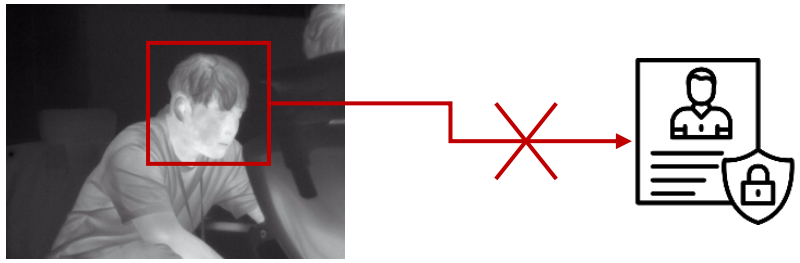
Hyun Woo Oh*, Cheol-Ho Choi, Jeong Woo Cha, Hyunmin Choi, Jung-Ho Shin, and Joon Hwan Han

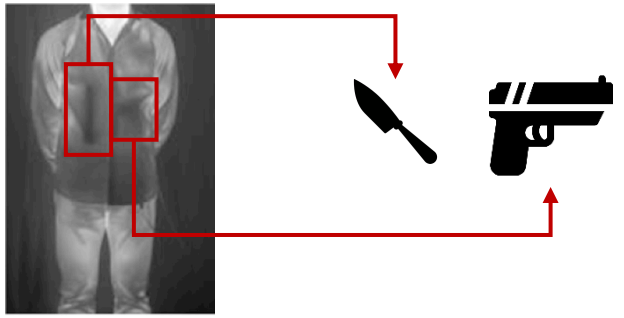
Core H/W Team, Infra Technology R&D Center, Hanwha Systems

Seongnam, Korea

Thermal Imaging

- Adaptability for wide-broad scenarios
  - Robust for low-light conditions
- Privacy protection
  - Hard to identify human visual characteristics
- Detection of human-invisible objects
  - Hidden objects can be inspected





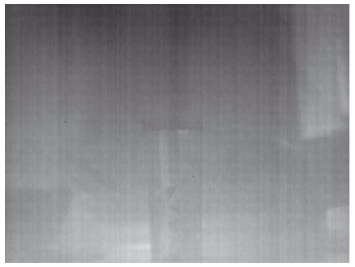
[1]

[1] Kowalski, M. Hidden Object Detection and Recognition in Passive Terahertz and Mid-wavelength Infrared. *J Infrared Milli Terahz Waves* 40, 1074–1091 (2019)

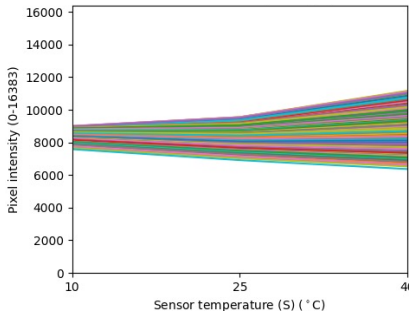
General Challenges in Thermal Imaging

- **Fixed-pattern noise**
  - ROIC noise (IRFPA)
  - Low energy level of IR

Non-uniformity correction (NUC)

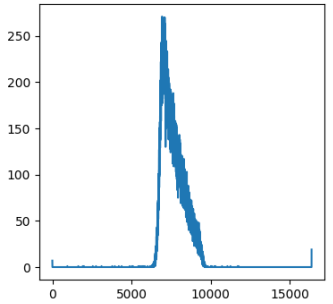


- **Thermal drift noise**
  - Thermistors ← affected by internal temp.



- **Dense intensity distribution**
  - low quantization levels than detectable temp. range

Contrast enhancement is essential
ex) Min-Max Stretching, HistEq, CLAHE, …



Hanwha Systems

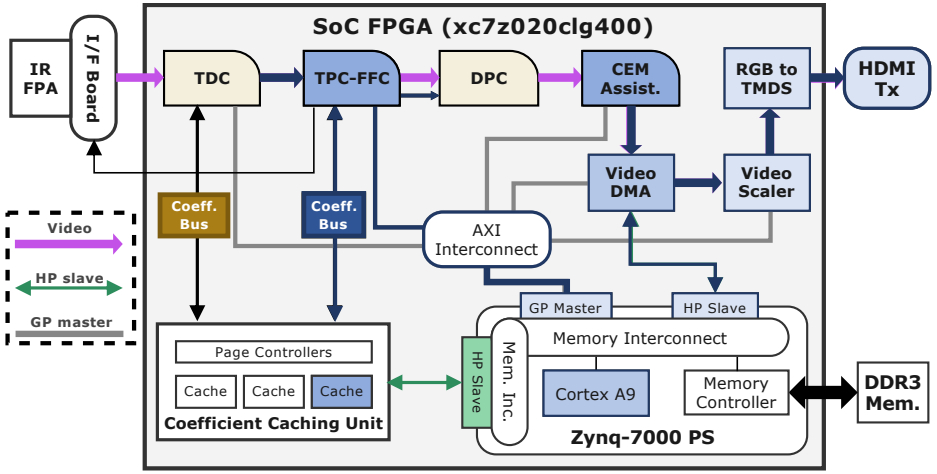
QUANTUM RED
Vision, beyond the limits.

Previous Solutions & Challenges

- **Two-point correction (TPC)-based NUC**
  - Pixel-wise algorithm based on linear expression
    - Great overall image quality
    - Too much coefficients → Hard to apply for real-time high-resolution, high FPS video
  - Requires run-time calibration to maintain the quality
    - Not considering the internal thermal energy → Prevalent interruption is required
- **Scene-based NUC & Neural Network-based NUC**
  - Use real-time image frame as a reference source
    - Does not require many coefficients → Optimized for high bitrate video.
  - Negative effects exist
    - Image blurring, Ghosting, …

Our Previous Solution: SoC FPGA-Based Image Processor (DSD 2023)

- **Algorithm side**
  - Three-layer TPC-based NUC
    - Thermal drift compensation (TDC): Reduce the prevalence of run-time calibration
    - Two-point correction (TPC): Avoid image blurring and ghosting artifacts
    - Defect pixel correction (DPC): Replace defective pixels with appropriate value
  - Contrast enhancement
    - Min-max stretching: Reduce compute load
- **System side**
  - SoC FPGA-based system design
    - FPGA (PL): Execute NUC
      - DRAM caching for raster-scan-ordered processing
        → Eliminate frame buffer BRAMs
    - PS: Execute contrast enhancement
      - Using pre-calculated min/max derived by the PL side



Limitations & Challenges

- **FPGA-based systems** (DSD 2023, Sensors 2024, …)
  - Higher BOM costs compared to the embedded SoC solutions

- **Contrast enhancement algorithms**
  - Min-max stretching (DSD 2023) / Histogram Equalization (Sensors 2024)
    - Pro: Lower compute loads → Appropriate for lightweight embedded systems
    - Con: No locality enhancements → Limited adaptability
  - Clip-Limited Adaptive Histogram Equalization (CLAHE)
    - Pro: Parameterized locality enhancements → Adaptable to diverse scenarios
    - Con: Current real-time implementation relies on FPGAs → Higher cost

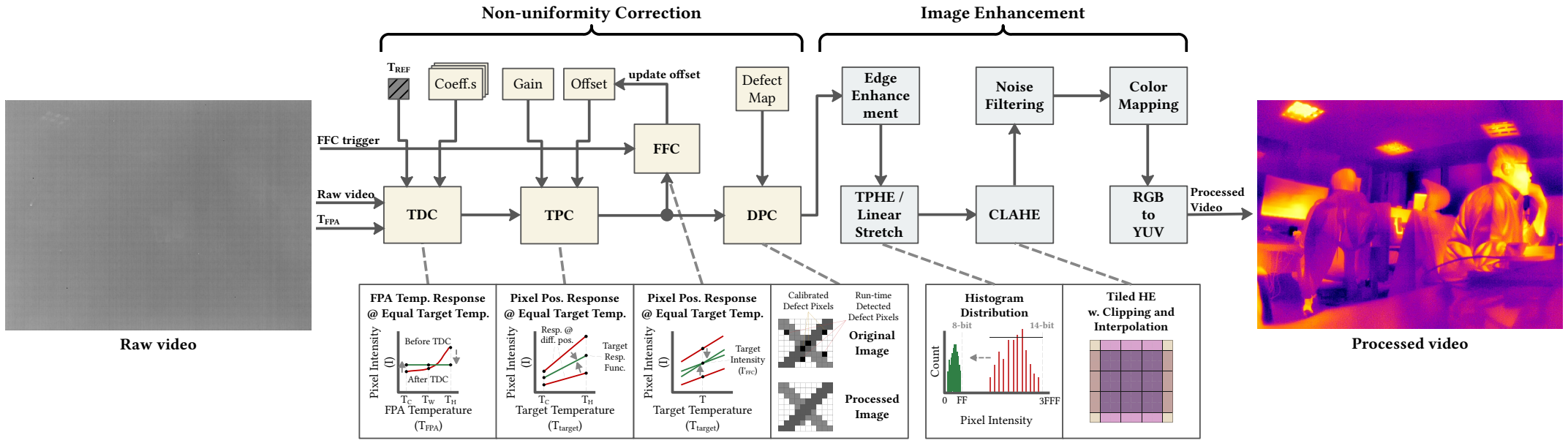
Our Solution

- **Goal**: No FPGAs, Advanced contrast enhancement, Higher throughput

- **Algorithm side**
  - TPC-based Multi-layer NUC algorithm: Similar to our previous solution.
  - Two-layer contrast enhancement
    - Histogram Equalization + CLAHE

- **System Side**
  - Lightweight Heterogeneous SoC
    - Compactness (Even smaller than tiny SoC FPGAs, One LPDDR2 Memory Chip)
    - Optimized Workload Distribution: DSPs, IPUs, Embedded Vision Engine (EVE).
    - SW Optimizations: DSP (SIMD, VLIW), Cache, Hardware IP (ISP).

Image Processing Architecture



Raw video

Processed video

- **NUC stack**: TDC + TPC + DPC
- **Image enhancement stack**
  - Edge Enhancement
  - Two-layer contrast enhancement: TPHE, CLAHE
  - Color mapping & Formatting (RGB to YUV)

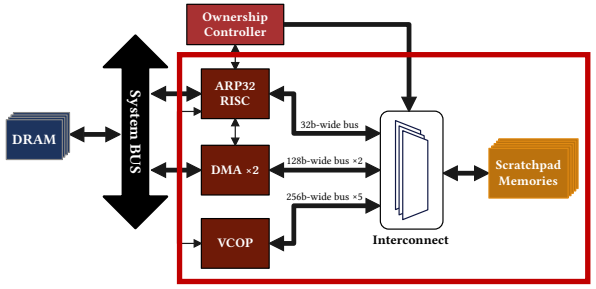
Hanwha Systems

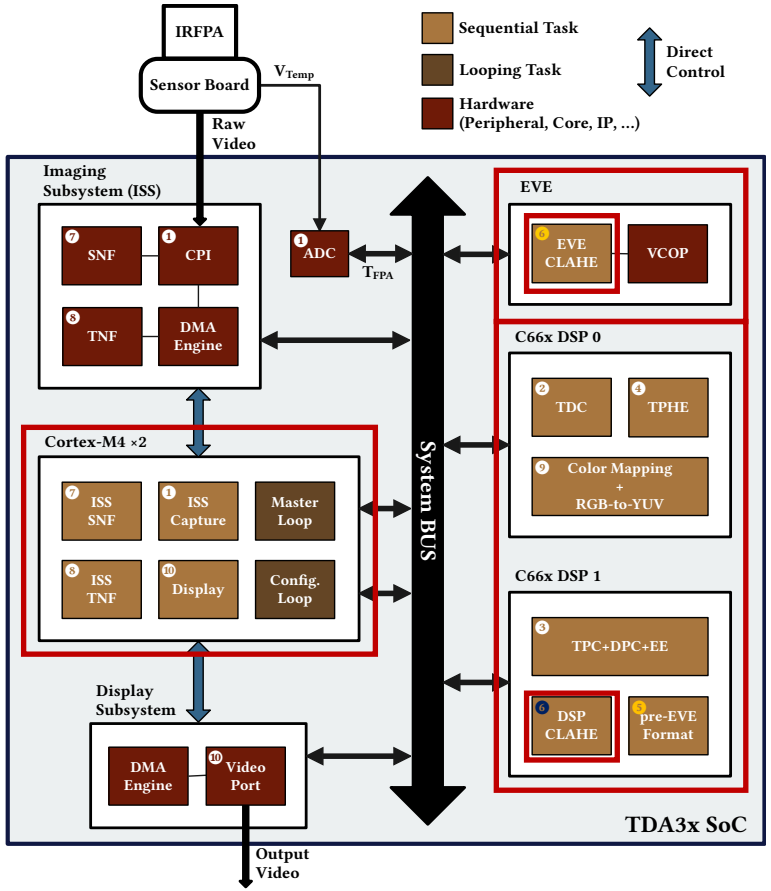
QUANTUMRED
Vision, beyond the limits.

Workload Distribution

- **2x C66x DSPs**
  - Suitable for intensive processing with run-time configuration
  - Most image processing tasks
- **2x IPUs (Cortex-M4)**
  - Not appropriate for intensive operations → managing tasks
- **EVE**
  - Particularly optimized to execute CLAHE.
  - Run-time configuration is limited
    - Isolated data-path requiring system stall
    - Distinctive SW design flow

    → We provide the options to execute CLAHE on the DSP or EVE.

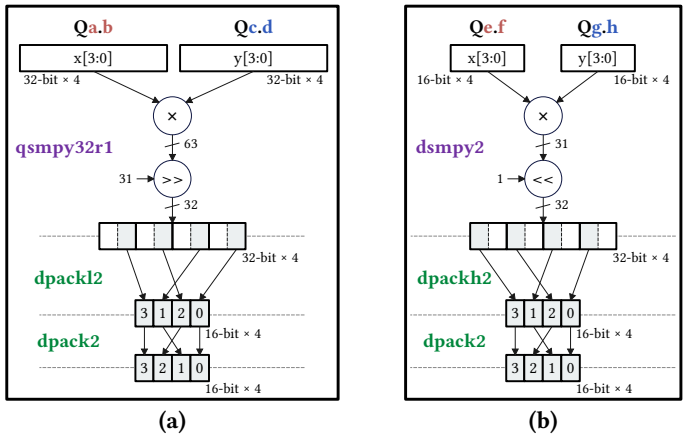


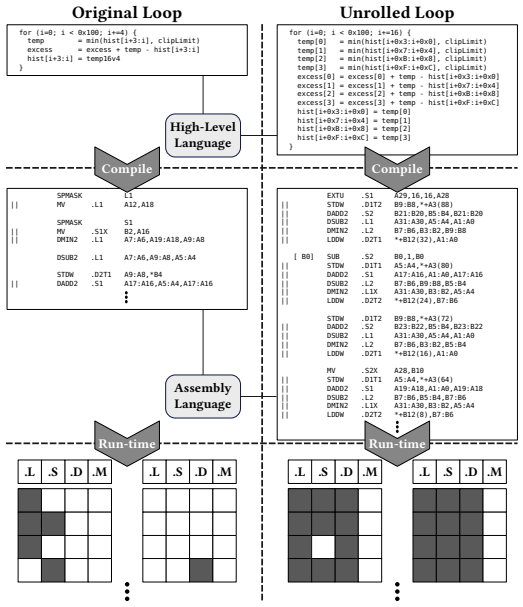


SW Optimization (i) – SIMD / VLIW

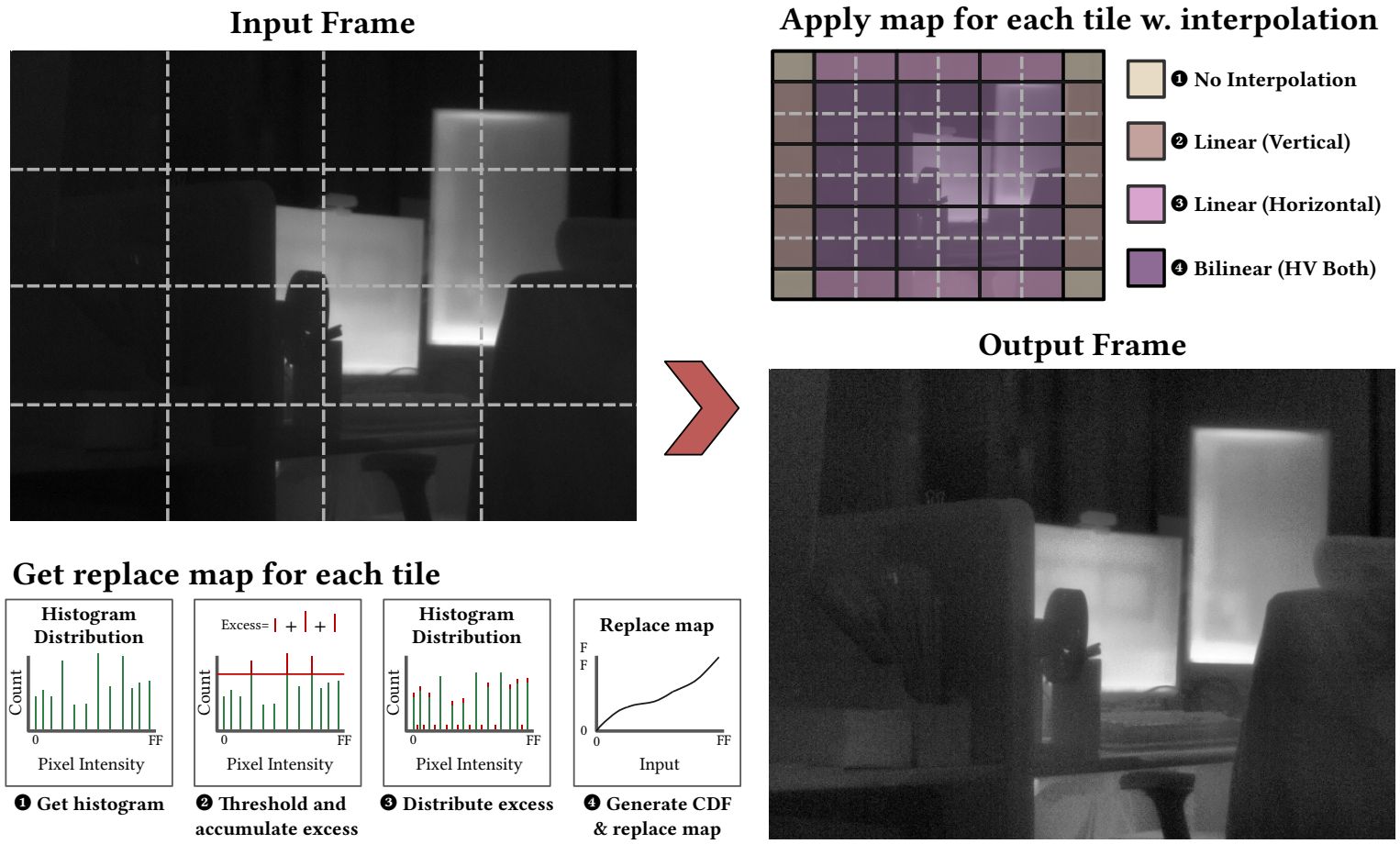
- **SIMD Kernels**
  - 4-way fixed-point multiplication
  - Adaptable to diverse configurations
  - (a): 32 by 32-bit multiply and convert to 16-bit
    - Condition: a + b = 31, c + d = 31, b + d = 31
  - (b): 16 by 16-bit multiply and convert to 16-bit
    - Condition: e + f = 15, g + h = 15, f + h = 15
- **VLIW Optimization**
  - Manual unlooping

    → Increase instruction-level parallelism



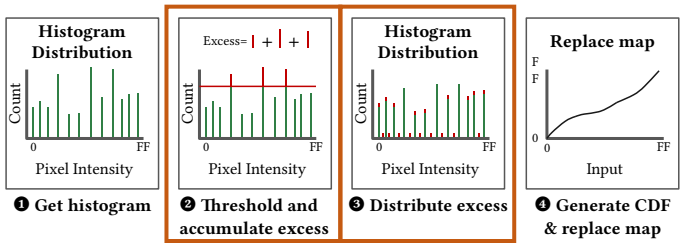


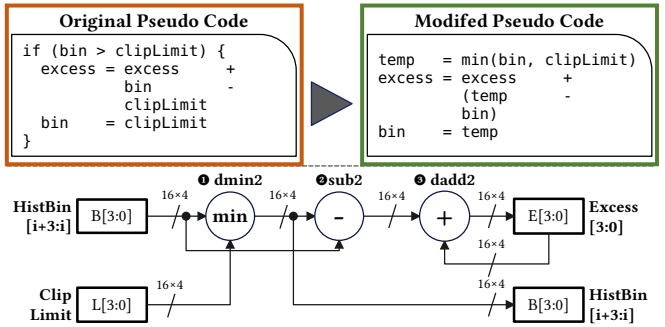
SW Optimization (ii) - CLAHE



SW Optimization (ii) - CLAHE

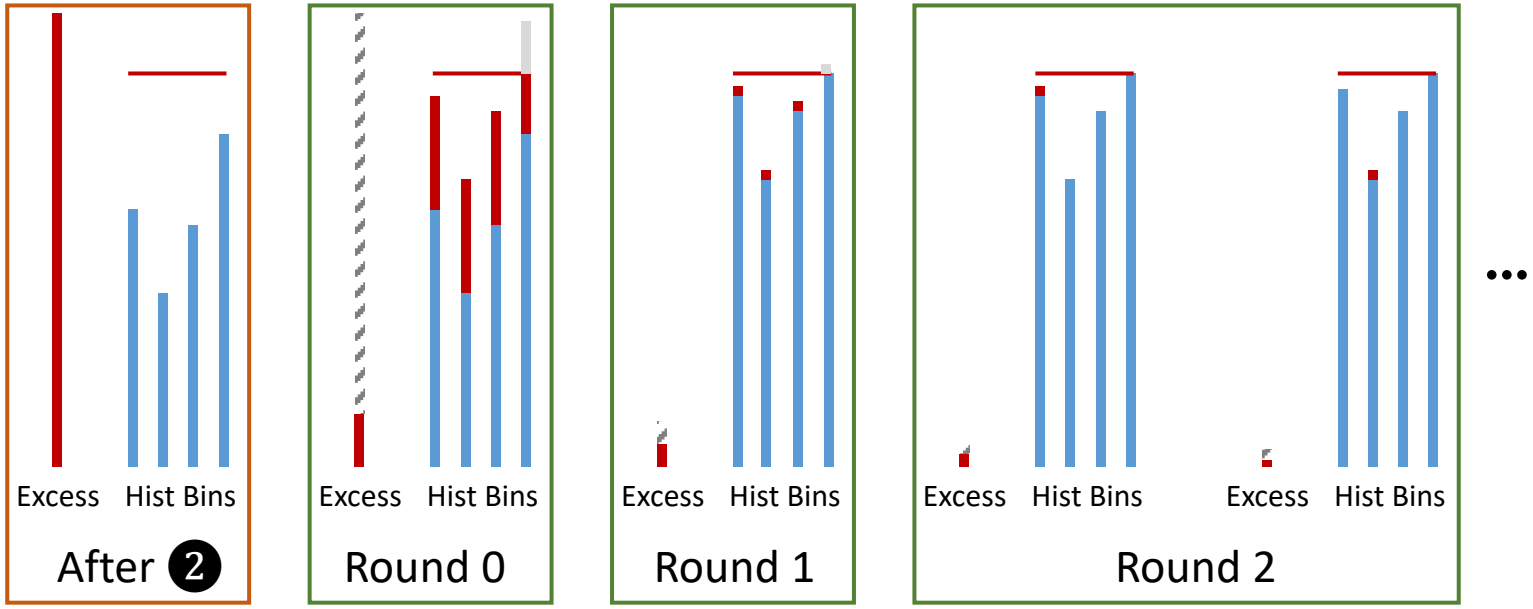
- **Thresholding Kernels**
  - 4-way SIMD operation
  - Replaces conditional branch





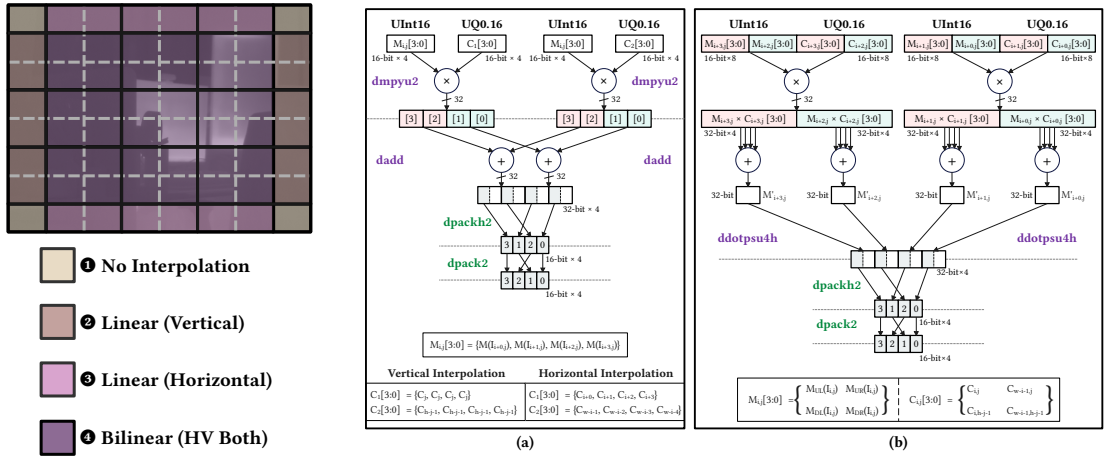
- **Excess Distribution**
  - Three round operation

```
#### Round One:   Fast Distribution
while (excessTotal > 0xFF) {
  dist16v4 = duplicate_by_four(excessTotal >> 8)
  pass16v4 = {0,0,0,0}
  for (i=0; i < 0x100; i+=4) {
    temp16v4 = min(hist[i+3:i] + dist16v4, clipLimit16v4)
    pass16v4 = pass16v4 + temp16v4 - hist[i+3:i]
    hist[i+3:i] = temp16v4
  }
  excessTotal = excessTotal - sum(pass16v4)
}
#### Round Two:   Distribute by Up to Four
for (j=0; excessTotal > 4; j+=4) {    // j is unsigned 8-bit
  temp16v4 = min(hist[j+3:j] + {1,1,1,1}, clipLimit16v4)
  excessTotal = excessTotal - sum(temp16v4 - hist[j+3:j])
  hist[j+3:j] = temp16v4
}
#### Round Three: Distribute Each by One
for (; excessTotal > 0; j+=1) {
  temp = min(hist[j] + 1, clipLimit)
  excessTotal = excessTotal - temp + hist[j]
}
```

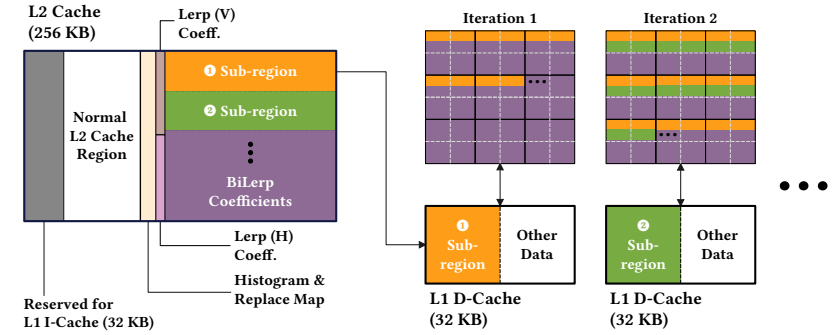


SW Optimization (ii) - CLAHE

- **Interpolation SIMD Kernels**
  - Convert division to fixed-point multiplication
  - Use coefficients from the memory
    - Calculated at
      - System boot time
      - Run-time configuration

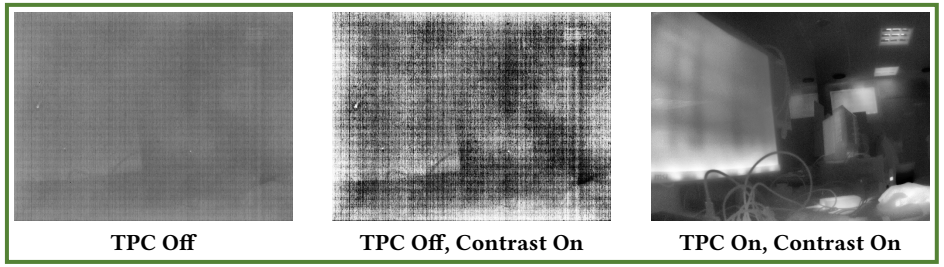


- **Cache Optimization**
  - Static Cache Memory Partitioning
    - Ensure no L2 misses on coefficient access
  - Regional Iteration
    - Split each tile by sub-regions (adjustable)
      → Iteratively access to
    - Increase L1 cache hit rate

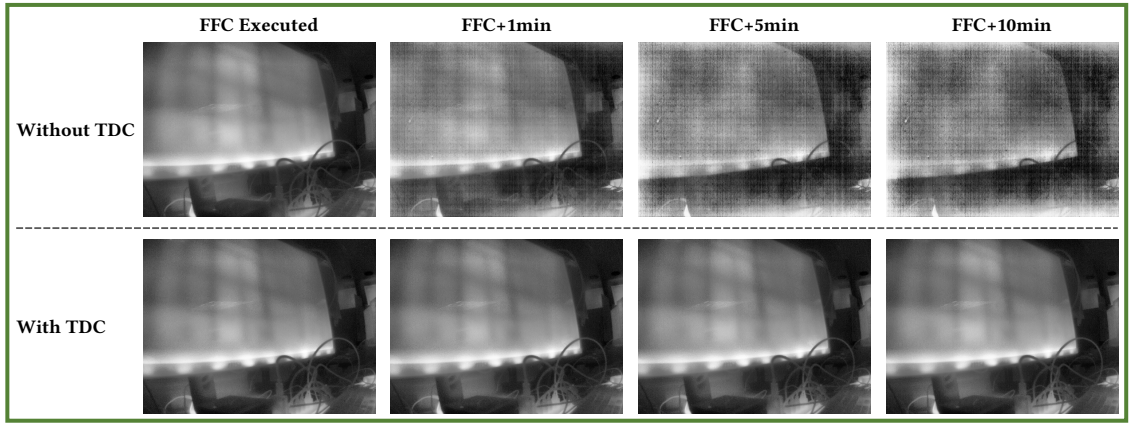


Evaluation – Visualized

- **NUC**
  - **Two-Point Correction (TPC)**
    - Free from ghosting and blurring
    - Fixed-pattern noise removed
  - **Thermal Drift Compensation (TDC)**
    - Without TDC: the thermal drift caused the image degradation starting after 1 minute
    - With TDC: The human-sighted image quality is maintained after 10 minutes



- **Contrast Enhancement**
  - TPHE / TPHE + CLAHE



Specs

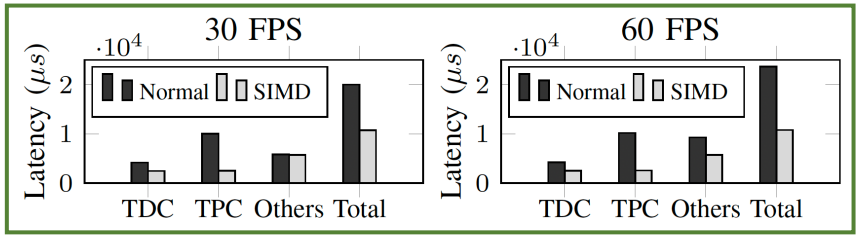
Device: TDA3MVS SoC + LPDDR2 512 MB
Video: 640×480, 14-bit @ 30/60 FPS

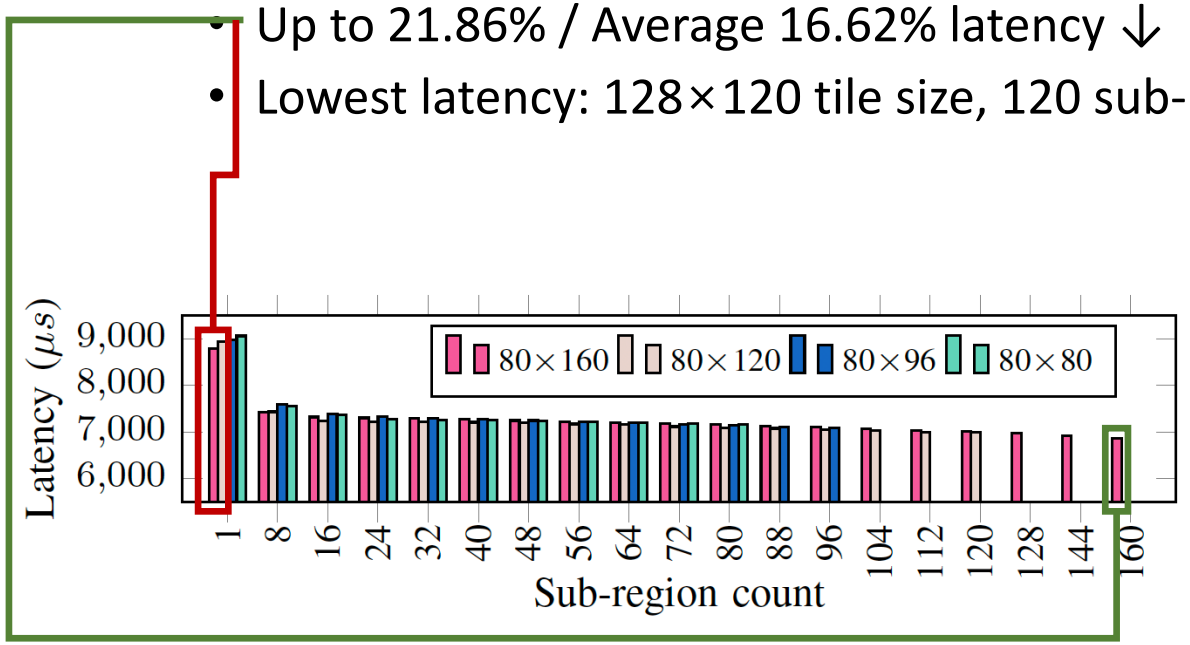
Hanwha Systems

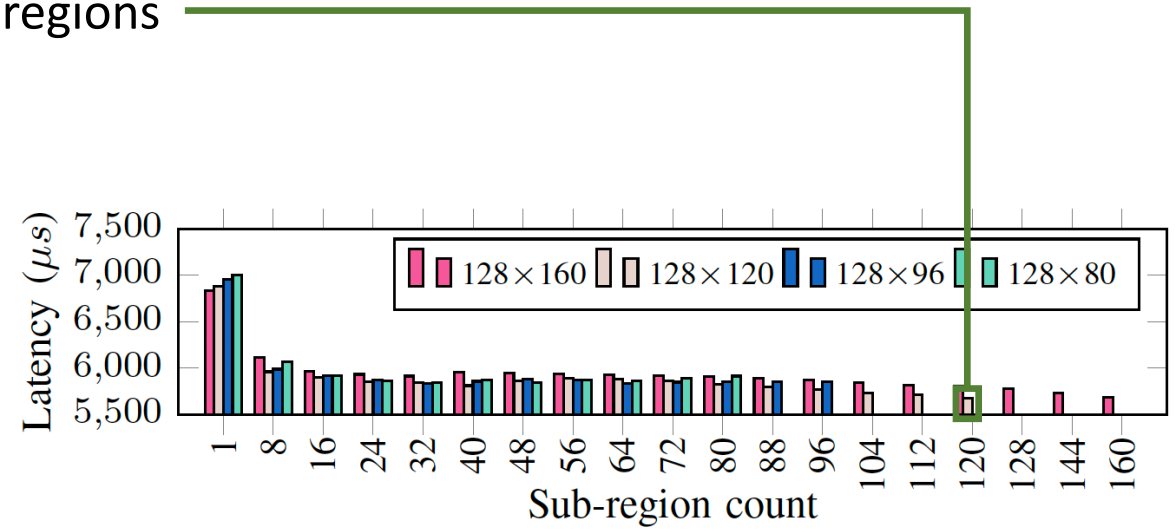
QUANTUMRED
Vision, beyond the limits.

Evaluation – SW Optimizations

- **SIMD Kernel**
  - 46.4% / 54.5% latency ↓ @ 30 / 60 FPS
    → 1.87× / 2.2× improved throughput
- **Regional Iteration**
  - Up to 21.86% / Average 16.62% latency ↓
  - Lowest latency: 128×120 tile size, 120 sub-regions

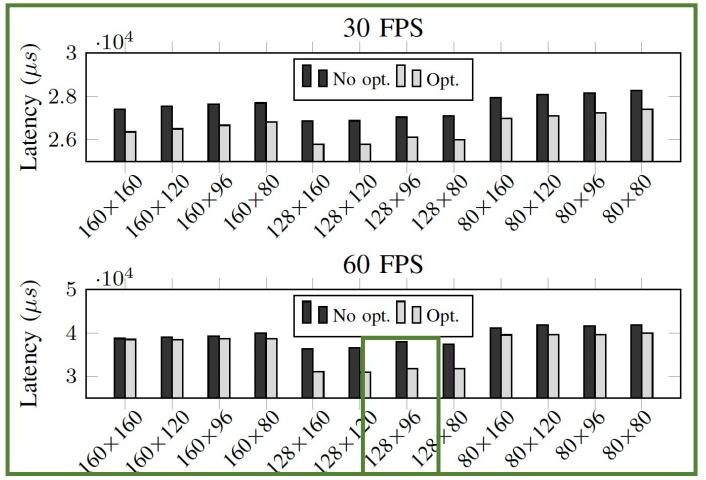




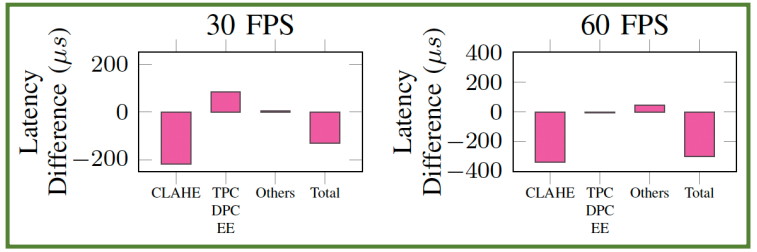


Evaluation – SW Optimizations

- **VLIW Optimization**
  - Avg. 3.72% / 7.18% latency ↓ @ 30 / 60 FPS
  - Up to 16.43% latency ↓ @ 60 FPS, 128×96



- **Static Cache Memory Partitioning**
  - Avg. 0.54% / 1% latency ↓ (131 μs, 299 μs)
  - Up to 4.27% latency ↓ @ 60 FPS, 128×120 (1265 μs)



Evaluation – Comparative Analysis

- **vs. SOTA**
  - Maximum power under 2.2 *W*
  - 9.42% / 9.96% power efficiency ↑ @ 30 / 60 FPS
  - 2.8% power efficiency ↓ with CLAHE

Works	Syst. Compon.	Resolut.	FPS	NUC Stack	Contrast Enhanc.	Power (W)	Power Effic. (FPS / W)
This work	SoC + IRFPA	640×480	30	TPC+TDC+FFC+DPC	HE-variant+CLAHE HE-variant	1.837[a] / 1.865[b] 1.656	16.33[a] / 16.09[b] 18.12
			60	TPC+TDC+FFC+DPC	HE-variant+CLAHE HE-variant	2.145[a] / 2.160[b] 1.908	27.97[a] / 27.78[b] 31.45
[23]	FPGA + IRFPA	640×480	30 / 111	TPC+FFC+DPC	HE	1.812* / 6.970*	16.56* / 15.96*
[24]	FPGA + IRFPA	640×480	60 / 142	TPC+TDC+FFC+DPC	Min-max stretching	2.098*(60 FPS)	28.60*(60 FPS)
[43]	MCU + Camera	160×120	8.7	Camera dependent	HE-variant	0.920	9.46

[a]EVE CLAHE mode / [b]DSP CLAHE mode. *FPGA on-chip power, other system devices are not measured.

- **Potential for Further Improvement**
  - Peak core load 57.5% (DSP 0)
  - Lower EVE utilization → Room for further distribution

Core	30 FPS (%)			60 FPS (%)		
	HE[a]	DC[b]	EC[c]	HE[a]	DC[b]	EC[c]
Coretx-M4 0	6.90	14.20	12.30	9.80	21.20	21.00
Coretx-M4 1	1.10	5.60	1.20	1.20	11.80	1.60
DSP 0	23.80	26.40	26.30	46.70	50.50	57.50
DSP 1	8.60	26.20	12.80	17.70	55.70	30.80
EVE	0.70	0.70	6.30	0.70	0.90	12.00

[a]HE-only mode / [b]DSP CLAHE mode / [c]EVE CLAHE mode.

Conclusion

- The proposed **heterogeneous SoC**-based imaging system provides compact yet high-performance thermal imaging, enabling the **exclusion of the FPGA** chip, which was previously essential for **CLAHE**.

- Our system seamlessly processes the **14-bit**, **640×480 (VGA)** resolution, **60 FPS** IR video with the thermal-drift-aware **three-layer NUC** (TDC, TPC, DPC) and **two-layer contrast enhancements** (TPHE, CLAHE).

- Our design consumed a maximum of 2160 mW of power at DSP CLAHE mode, showing its feasibility in resource-constrained systems.

- The maximum DSP core load of 57.5% and room for workload distribution (low EVE load) allow for adaptation to higher throughput videos (1280×1024) with further optimization.

Thank You!

cheoro1994@hanwha.com
hyunwooo@uci.edu