Accelerating a learning–based image processing pipeline for digital cameras

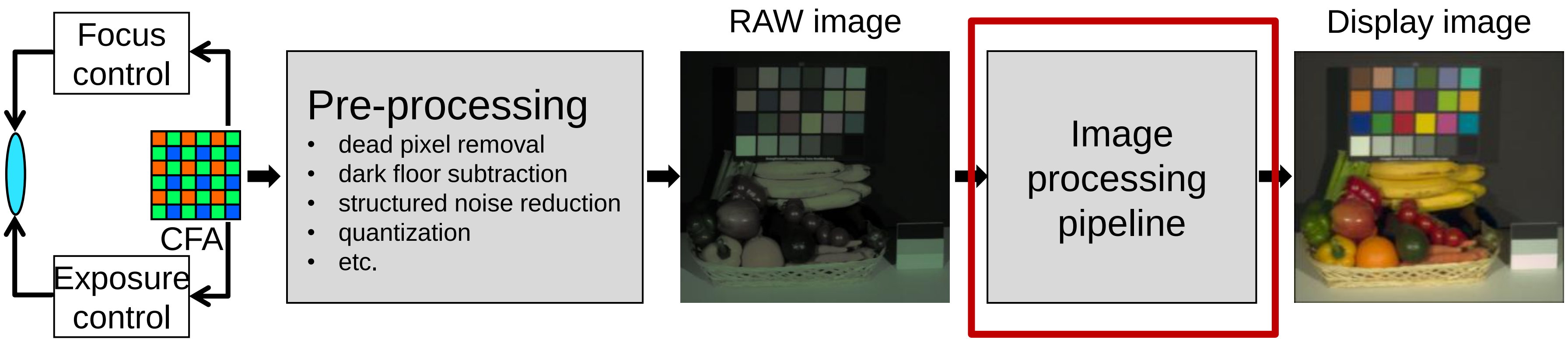
Local, Linear and Learned (L^3) pipeline

Qiyuan Tian and Haomiao Jiang

Department of Electrical Engineering
Stanford University

GPU Technology Conference, San Jose
March 17, 2015

Digital camera sub-systems



Lens, aperture and sensor

Transform the sensor data into a display image

Standard image processing pipeline

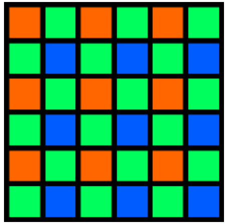
RAW image

Display image

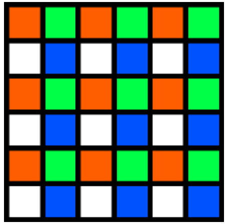
- Requires multiple algorithms
- Each algorithm requires optimization
- Optimized only for Bayer (RGB) color filter array (CFA)

Opportunity

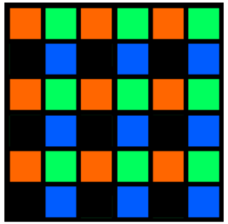
Extra sensor pixels enable new CFAs that improve sensor functionality and open new applications



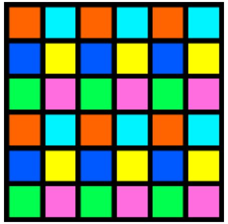
Bayer



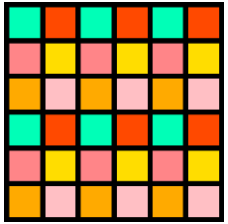
RGBW
low-light sensitivity
dynamic range



RGBX
infrared
light field



RGBCMY
multispectral



Medical
specialized
application

Challenge

- Customized image processing pipeline
- Speed and low power

L^3 image processing pipeline

RAW image

Classify pixels

Retrieve and apply transforms

Display image

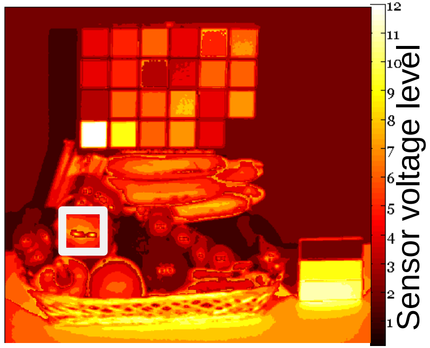
Local, Linear and Learned (L^3)

- Combines multiple algorithms into one
- Rendering is simple, fast and low-power
- Uses machine learning to optimize the class transforms for any CFA

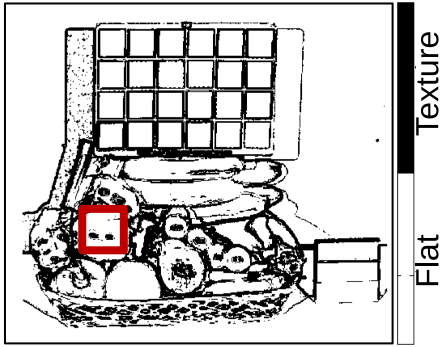
Classify pixels

RAW image

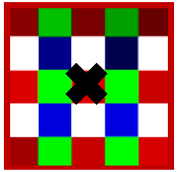
Center pixel color



Intensity



Contrast



“Local” pixel values
(local patch)

<u>Class</u>

Center pixel color: red

Intensity: high

Contrast: flat

Retrieve and apply transforms

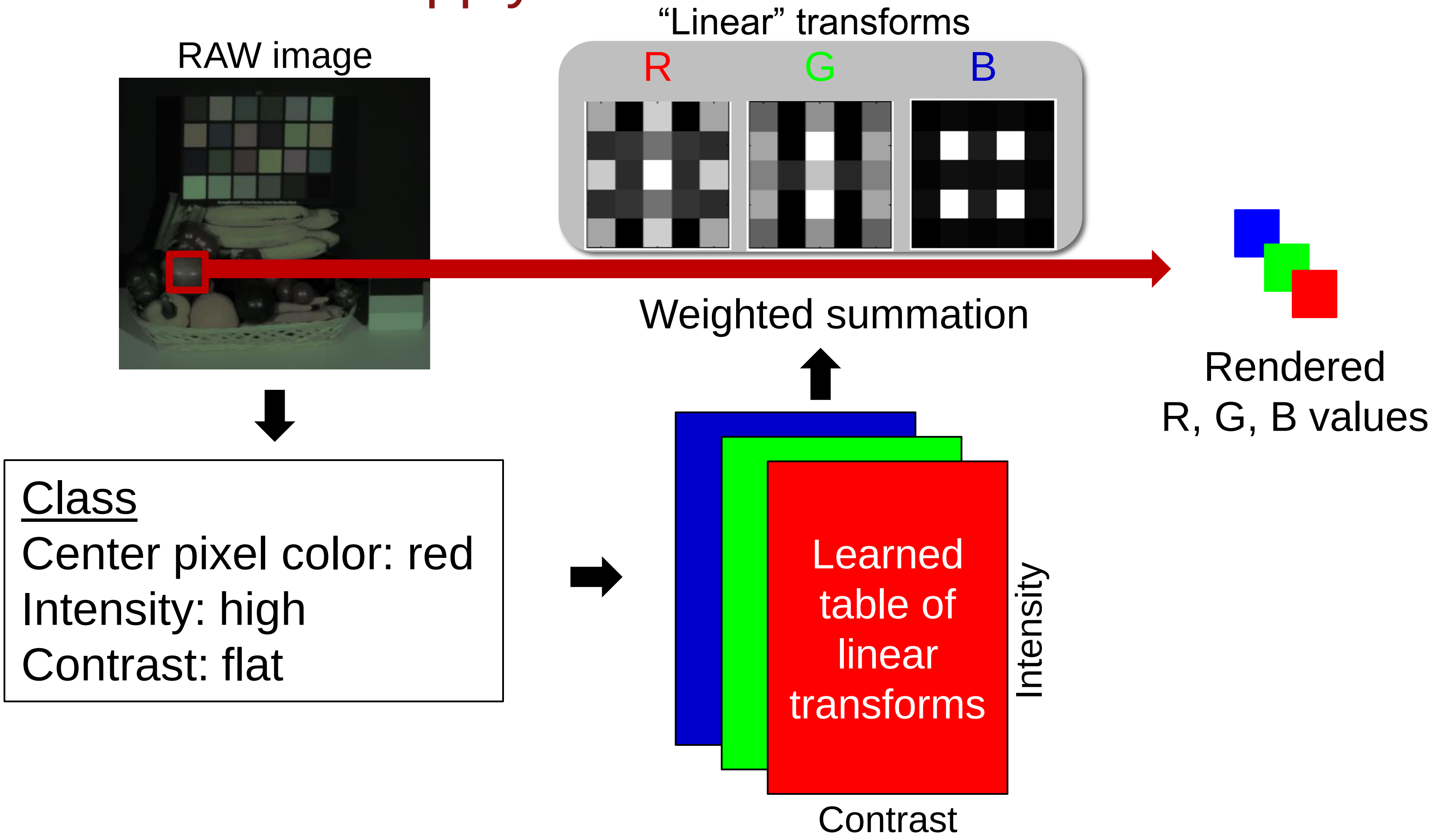
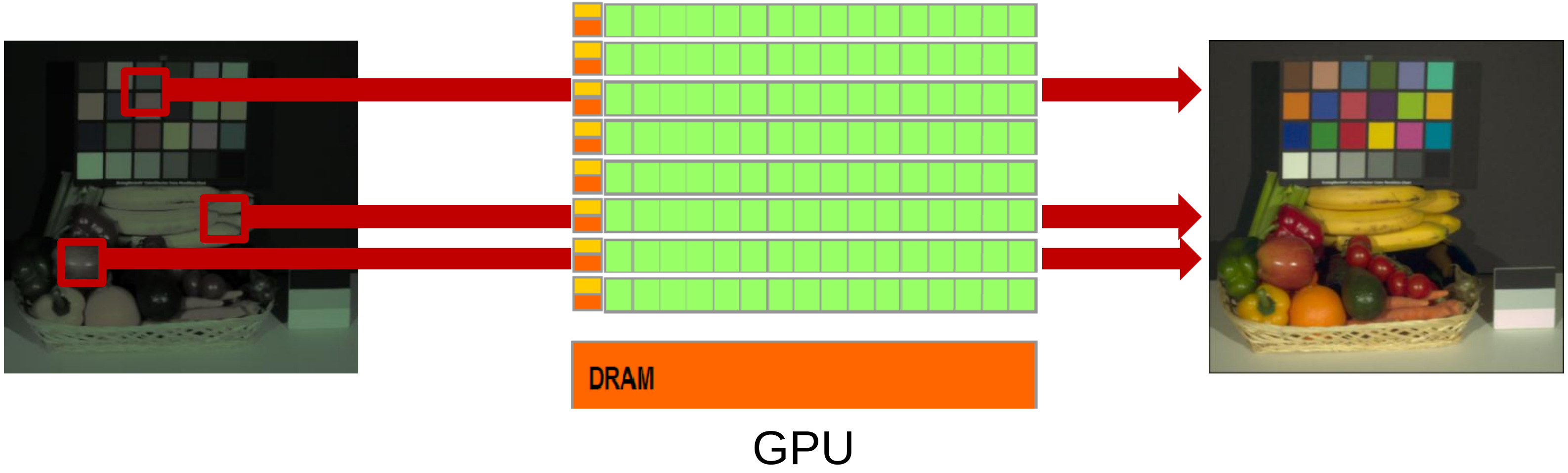


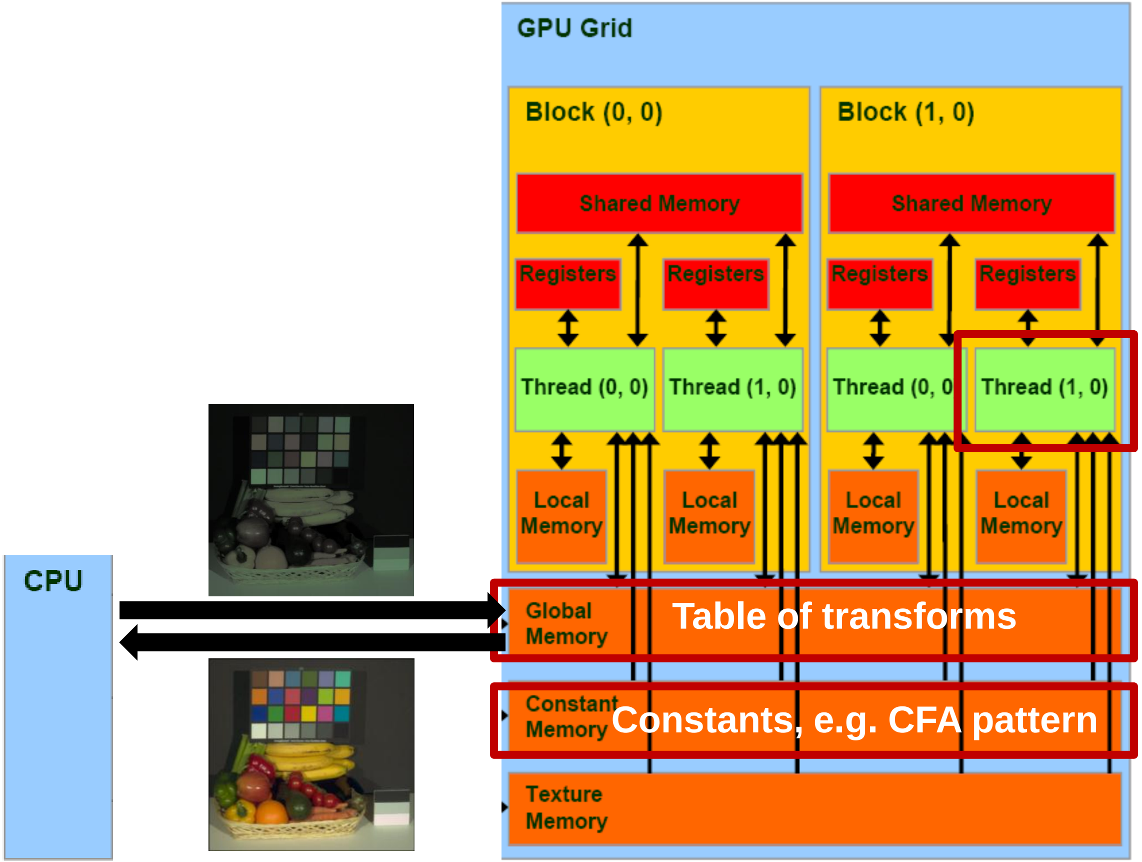
Table-based architecture suits GPU



- Independent calculation for each pixel
- Simple weighted summation

Thus well-suited for parallel rendering using GPU

GPU implementations



Render one pixel (*i, j*)

- Calculate class index
- Retrieve transforms
- Weight sum

GPU acceleration results

- GPU: NVidia GTX 770 (1536 kernels, 1.085 GHz)
- CPU: Intel Core i7-4770K (3.5 GHz)
- CUDA/C programming

Tian et al. 2015

Potential speed improvement

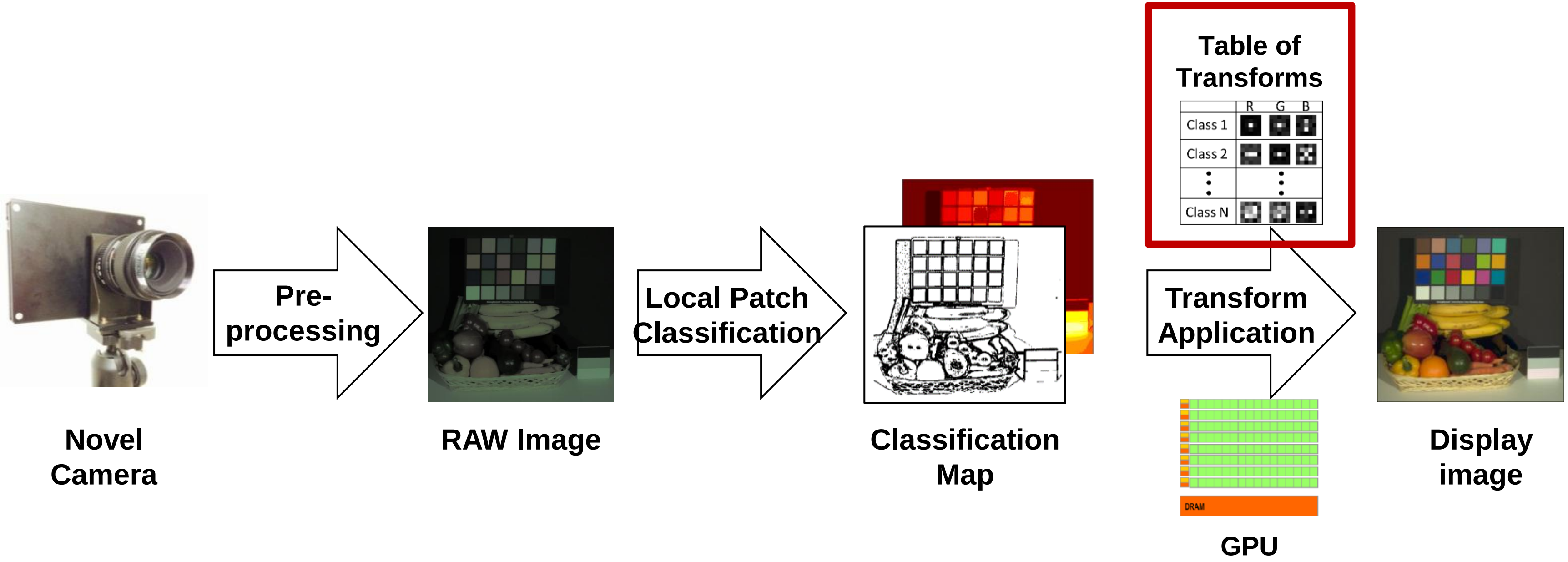
Use shared memory and registers

Specialized image signal processor (ISP)

- **L^3 ISP**

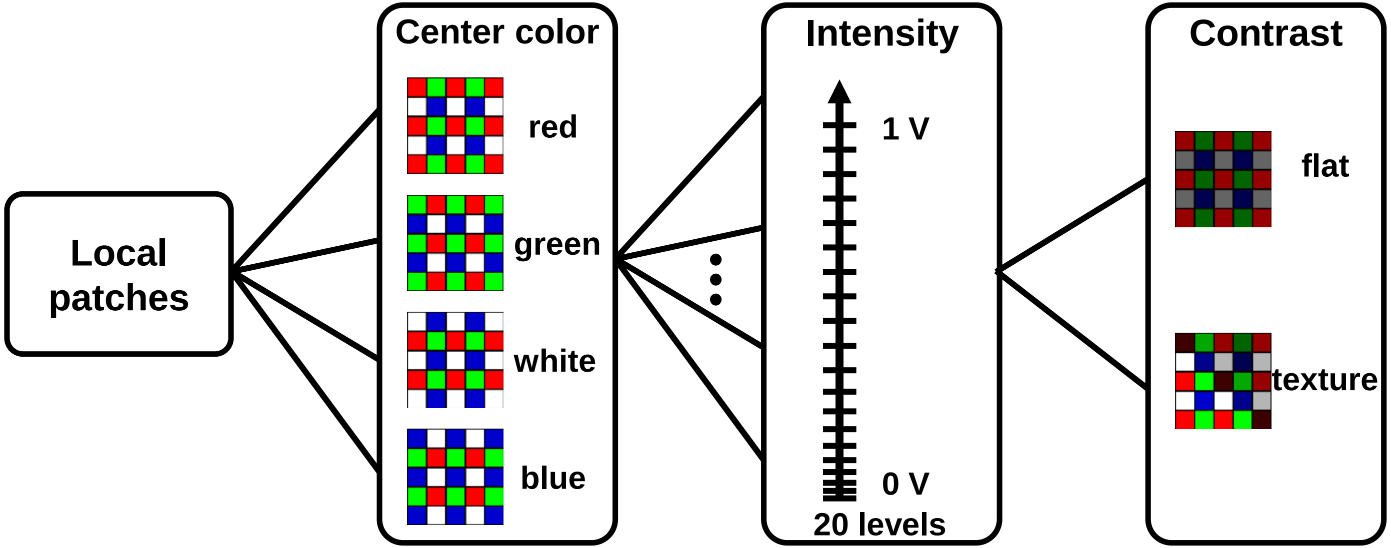
L³ processing

“Learn” the transforms

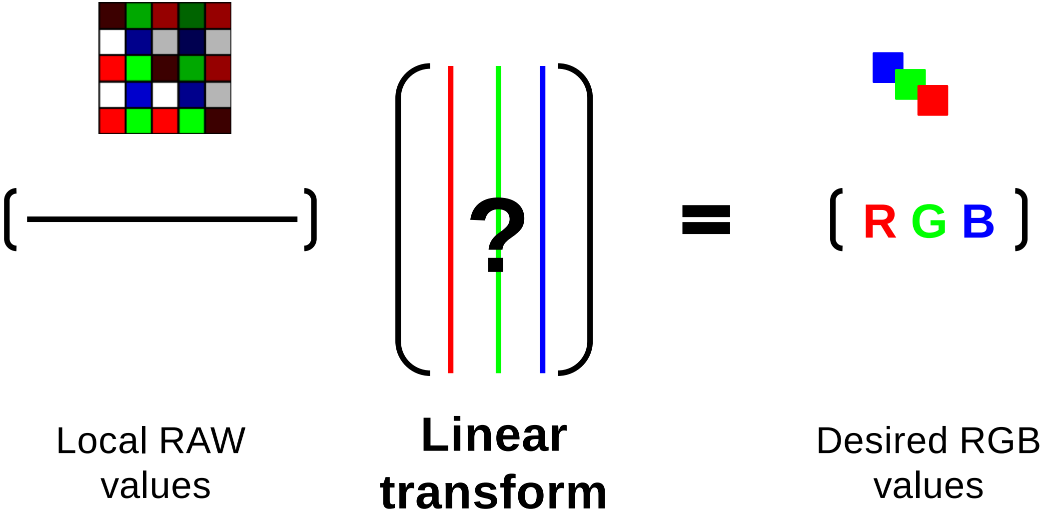


Locally linear transform

- Globally nonlinear for an entire image
- 480 linear transforms in total

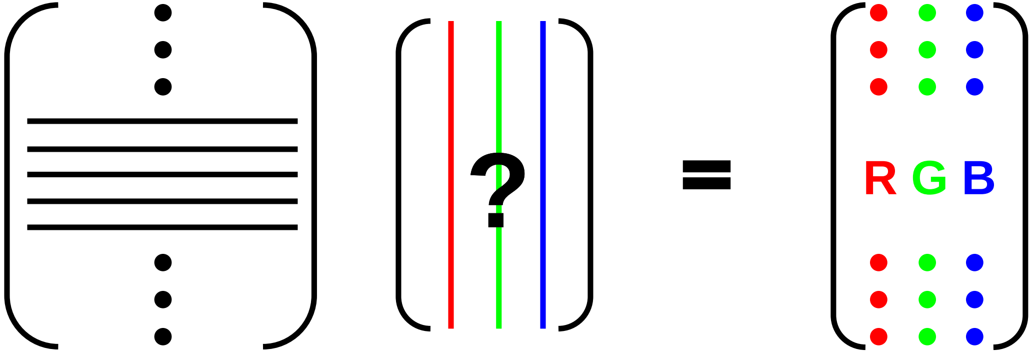


Learn the locally linear transform for each class



$$A \quad \mathbf{x} \quad = \quad \mathbf{b}$$

Solve the transform



Local RAW values

Linear transform

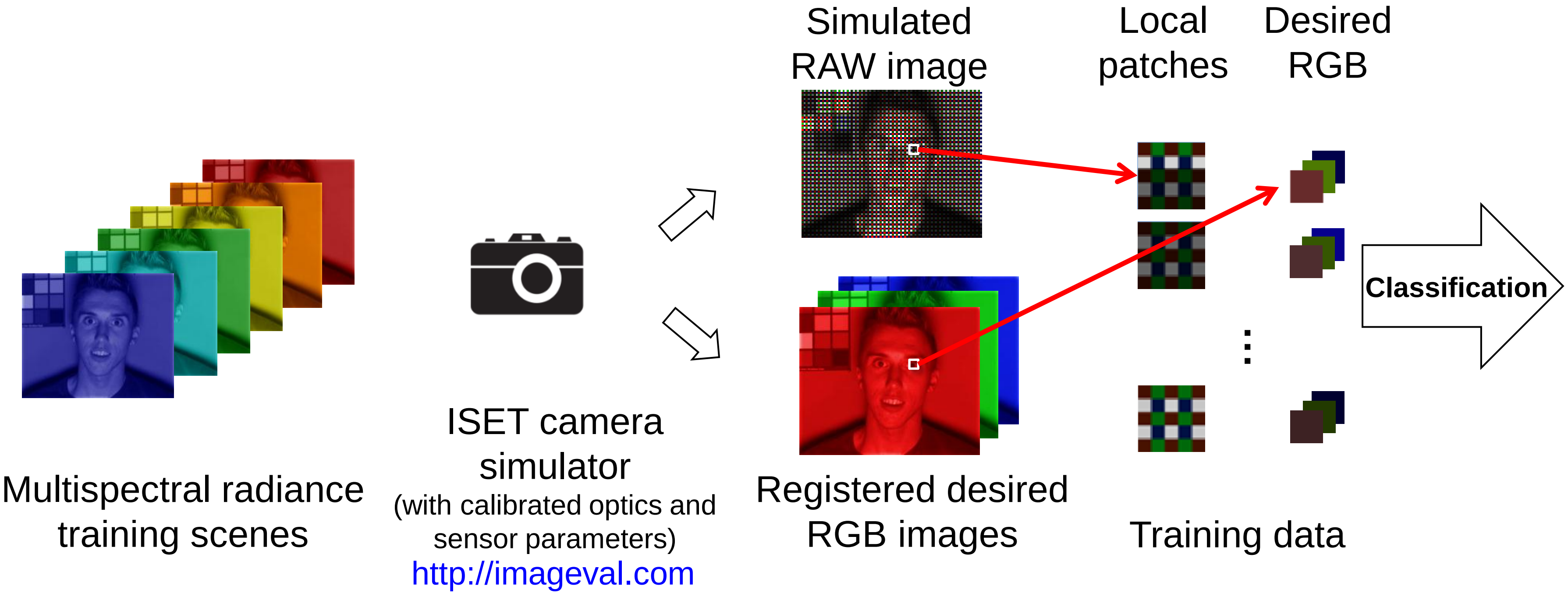
Desired RGB values

$$A \mathbf{x} = \mathbf{b}$$

$$\underset{\mathbf{x}}{\text{minimize}} \quad \|A\mathbf{x} - \mathbf{b}\|^2 + \|\Gamma\mathbf{x}\|^2$$

ridge regression

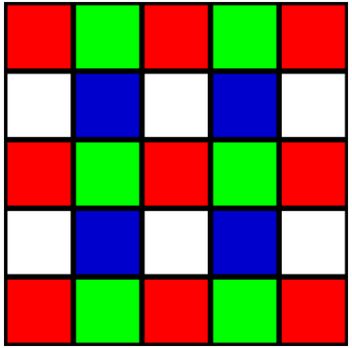
Training data from camera simulation



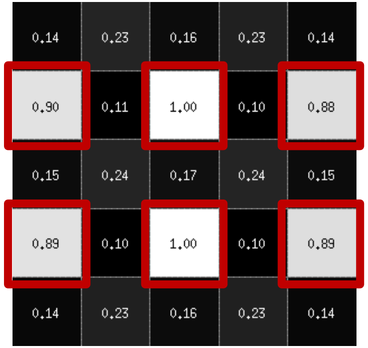
- Simulate any camera designs
- Various training scenes, illuminants and luminances
- Registered and desired RGB images

Learned transforms

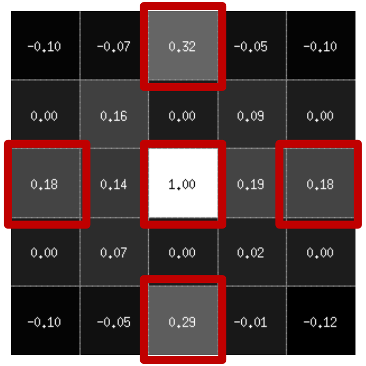
Red-pixel
centered patch



Transforms that
solve for R-channel



Dark class
(use more W)



Bright class
(use more RGB)

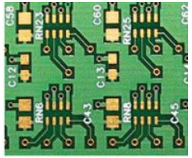
- Accounts for spatial and spectral correlation
- Accounts for sensor and photon noise

Advantages of learning

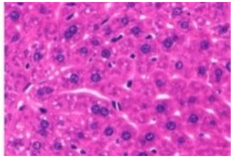
– Adapts to any application and scene content

Consumer Photography

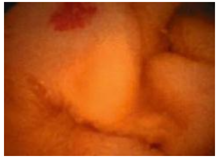
Document Digitization



Industrial Inspection

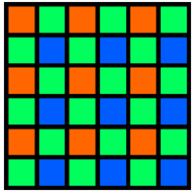


Pathology

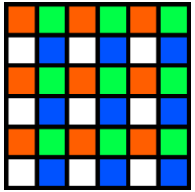


Endoscopy

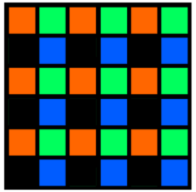
– Adapt to any CFA



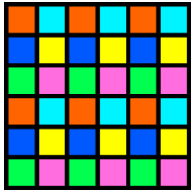
Bayer



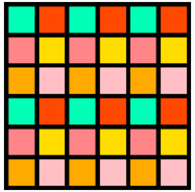
RGBW



RGBX

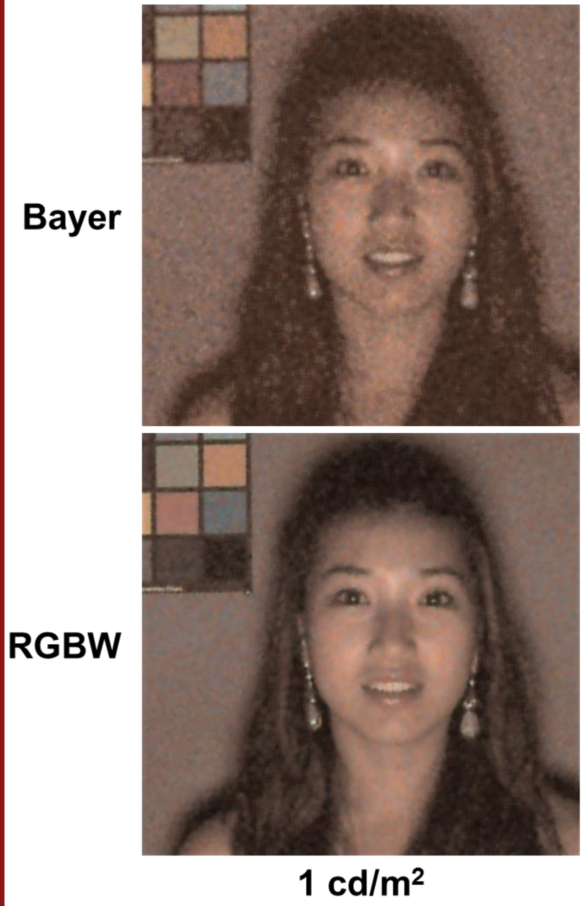


RGBCMY



Medical

Solve RGBW rendering



In dark scene

- Two f-stops gain

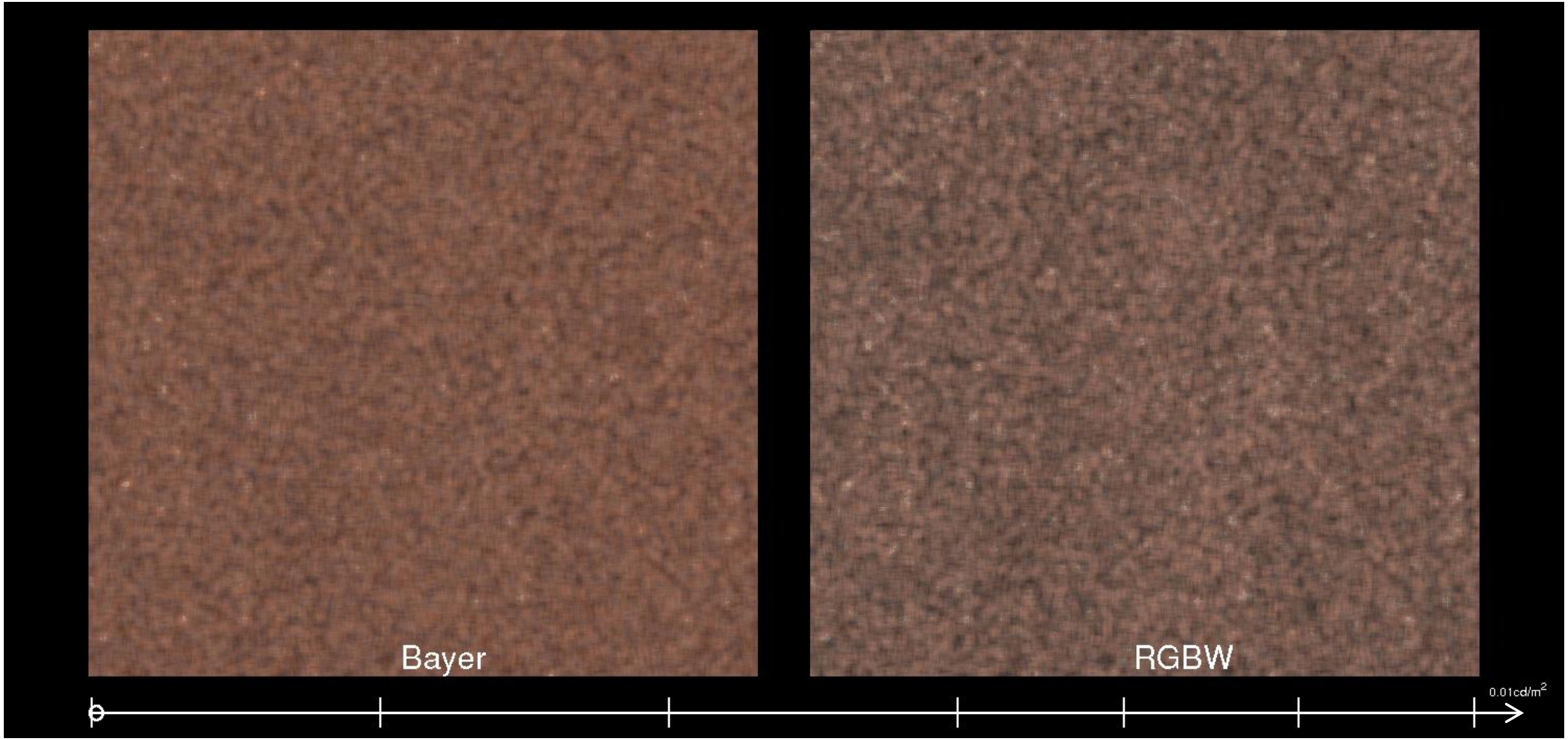
In bright scene

- Same performance

<u>Simulation conditions</u>
Exposure: **100 ms**
F-number: **f/4**

Tian et al. 2014

Smooth transition from dark to bright



.01 .1 1 10 100 200 300 cd/m^2

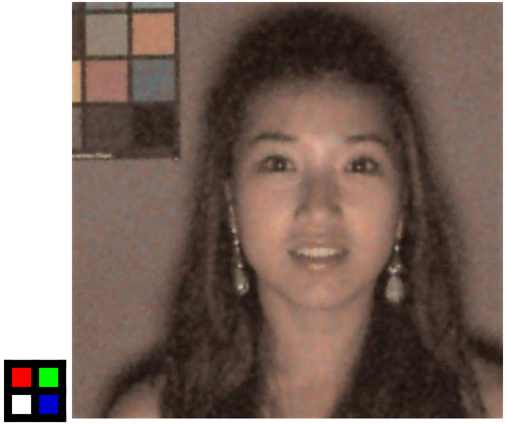
Scene Luminance

Tian et al. 2014

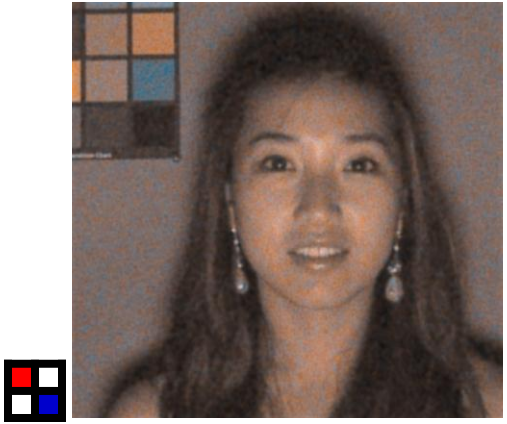
Compare RGBW CFA designs

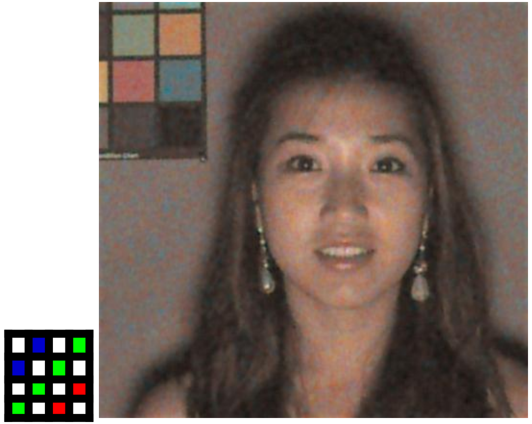
Bayer

Parmar & Wandell, 2009

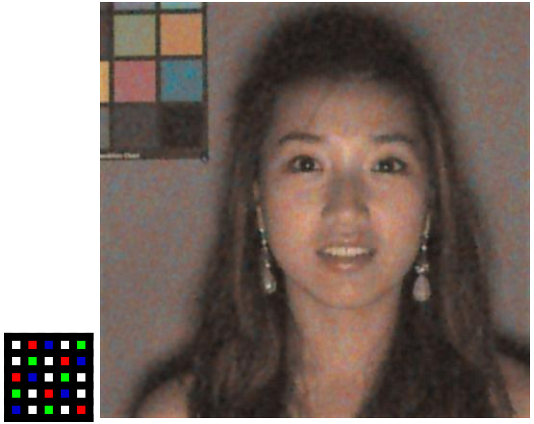


Aptina CLARITY+





Kodak



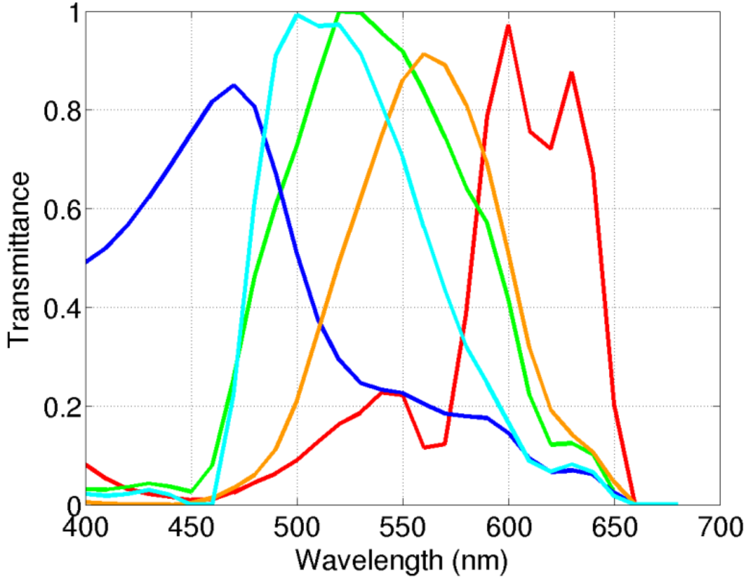
Wang et al., 2011

<u>Simulation conditions</u>
Luminance:**1cd/m^2**
Exposure: **100 ms**
F-number: **f/4**

Tian et al. 2014

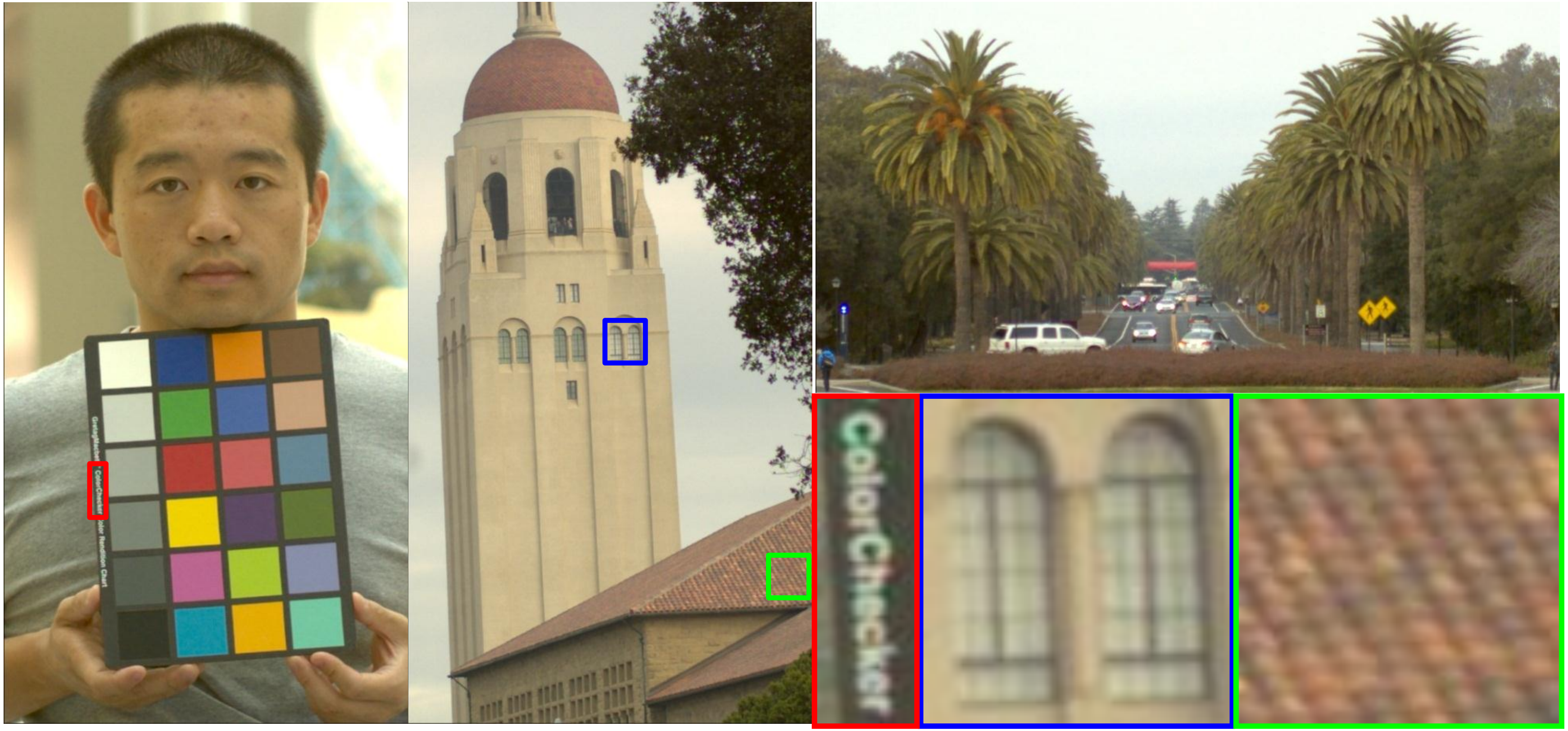
Five-band camera prototype

RGB Cyan Orange
4×4 super-pixel



Tian et al. 2015

L^3 solves five-band prototype rendering



Tian et al. 2015

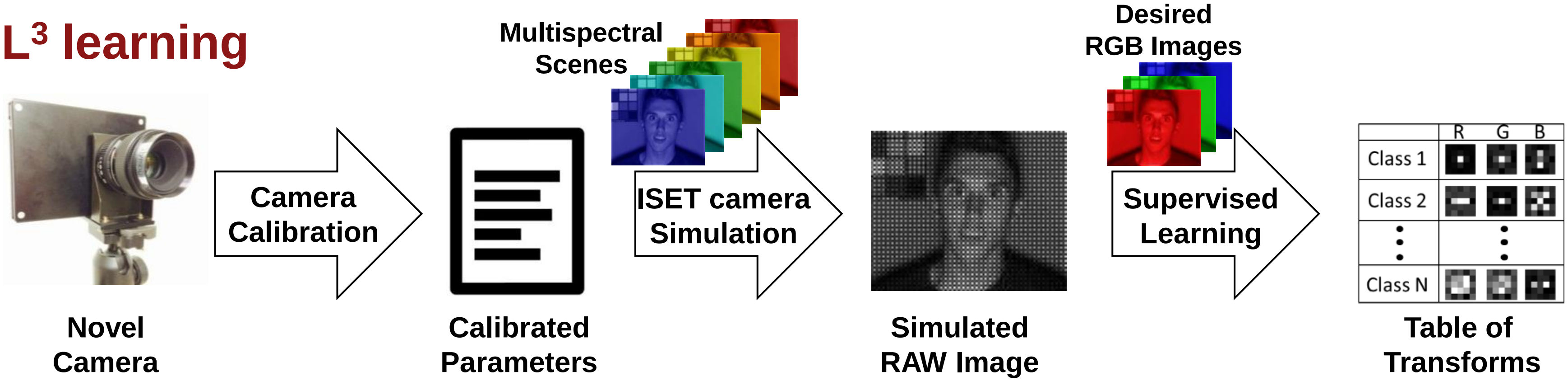
GPU acceleration results

Results	**GPU**	**CPU**
Image (1280×720)	0.062s (16 fps)	12.4s
Video (1280×720×1800)	163.2s (11 fps)	

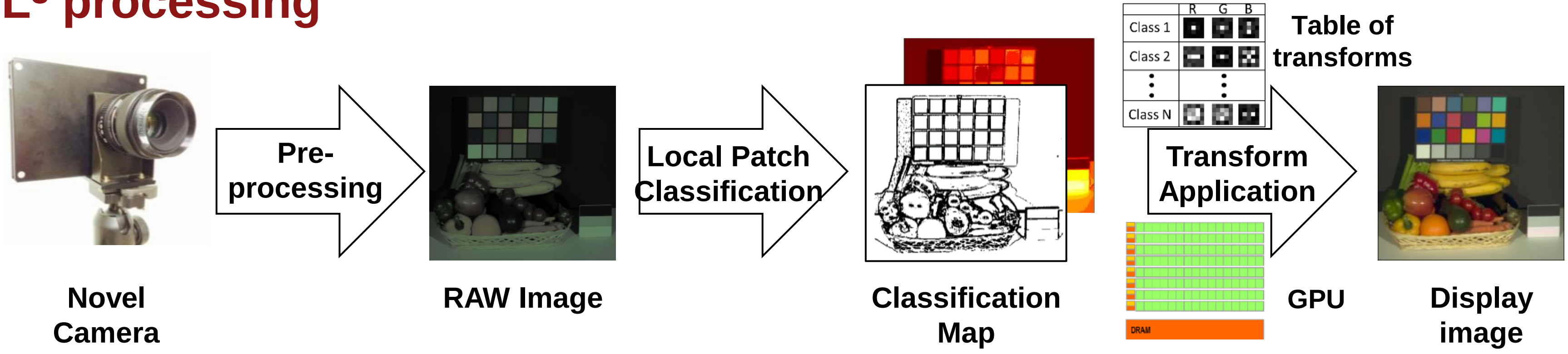
- GPU: NVidia GTX 770 (1536 kernels, 1.085 GHz)
- CPU: Intel Core i7-4770K (3.5 GHz)
- CUDA/C programming

Tian et al. 2015

L^3 learning



L^3 processing



Local, linear and learned pipeline (L^3) summary

- Table-based rendering architecture is ideal for GPU acceleration

- Machine learning automates image processing for any CFA and scene content

Rethink image processing pipeline

Acknowledgement

Advisors

- Brian Wandell, Joyce Farrell

Group members

- Henryk Blasinski, Andy Lin

Stanford collaborators

- Francois Germain, Iretiayo Akinola

Olympus collaborators

- Steven Lansel, Munenori Fukunishi

References

Tian, Q., Lansel, S., Farrell, J. E., and Wandell, B. A., “Automating the design of image processing pipelines for novel color filter arrays: Local, Linear, Learned (L^3) method,” in [IS&T/SPIE Electronic Imaging], 90230K–90230K, International Society for Optics and Photonics (2014).

Tian, Q., Blasinski, H., Lansel, S., Jiang, H., Fukunishi, M., Farrell, J. E., and Wandell, B. A., “Automatically designing an image processing pipeline for a five-band camera prototype using the local, linear, learned (L^3) method,” in [IS&T/SPIE Electronic Imaging], 940403-940403-6, International Society for Optics and Photonics (2015).

End

Thanks for your attention!

Questions?

Contacts

qytian@stanford.edu

hjiang36@stanford.edu

Potential speed improvement

- Local vs Global
  - L3 is locally linear: can use local memory to speed up
  - Locality in memory: writing output as RGBRGB is faster than writing as image plane
- Device based optimization
  - CFA pattern and other parameters are fixed: Constant Memory & no need to pass in
  - Symmetry and other properties
- CUDA, GLSL, FPGA, Hardware
  - L3 rendering is based on linear transforms and can be implemented with shaders or hardware circuits to achieve further acceleration