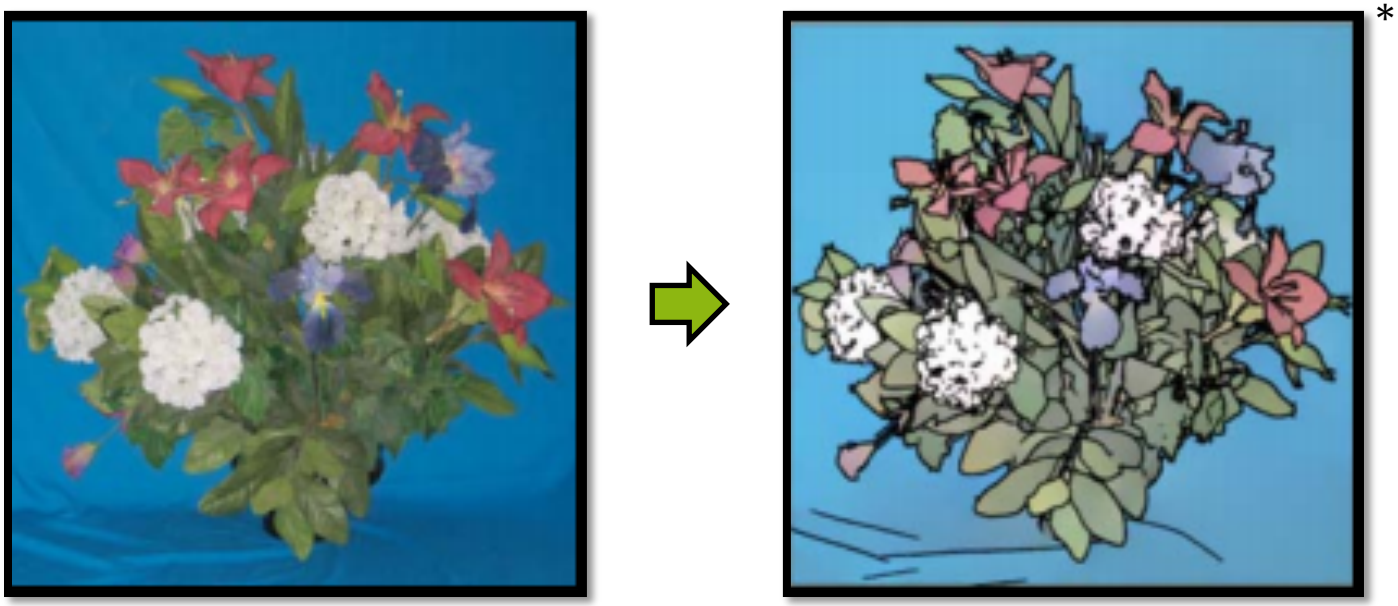
Real-time Non-photorealistic Viewfinder

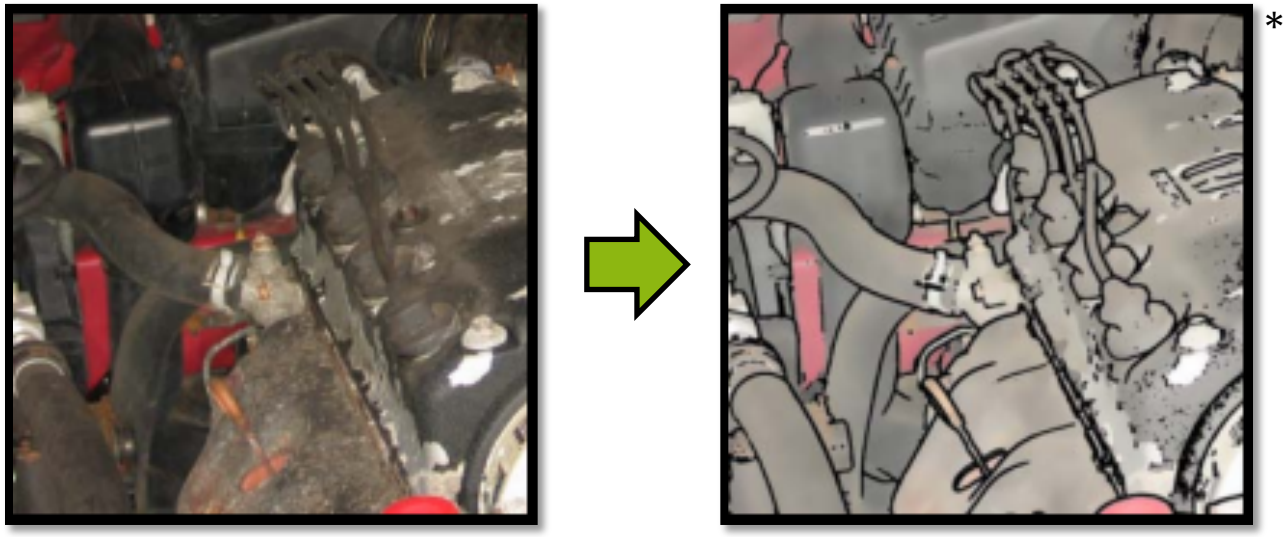
Real-time Non-photorealistic Viewfinder

Tony Hyun Kim and **Irving Lin**
CS478 Computational Photography
Dev. blog: cs478.blogspot.com



*Raskar, *et. al.* (2004)

INTRODUCTION: NPR

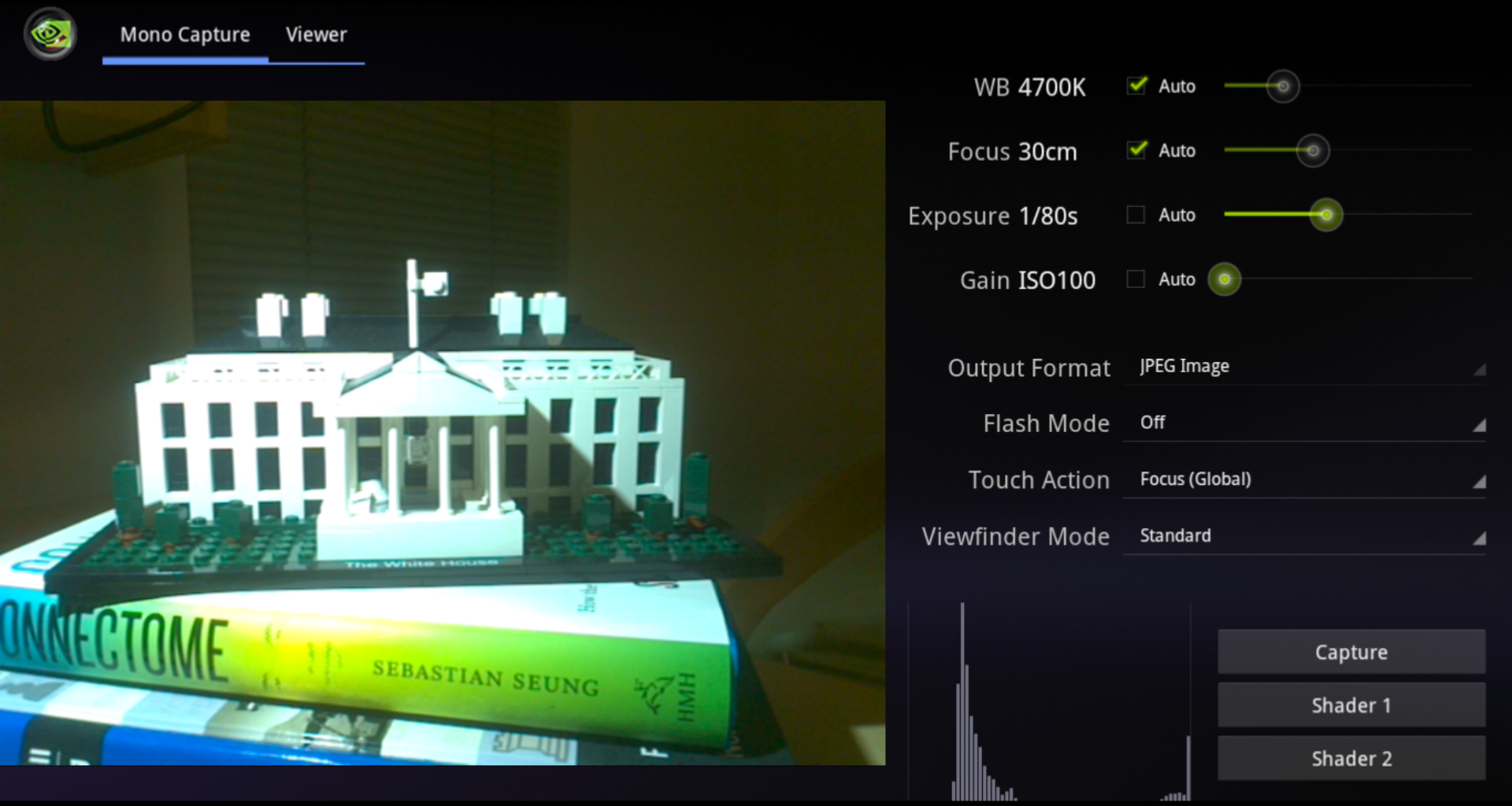

*

- Non-photorealistic rendering (NPR), or "stylization," allows artistic interpretation of an image at the expense of photorealism.
- Promotes simplicity and reduction of detail, better emphasizing the semantic content of the image.
- Applications: visual communication, augmented virtual reality, image compression. Also, just looks interesting.

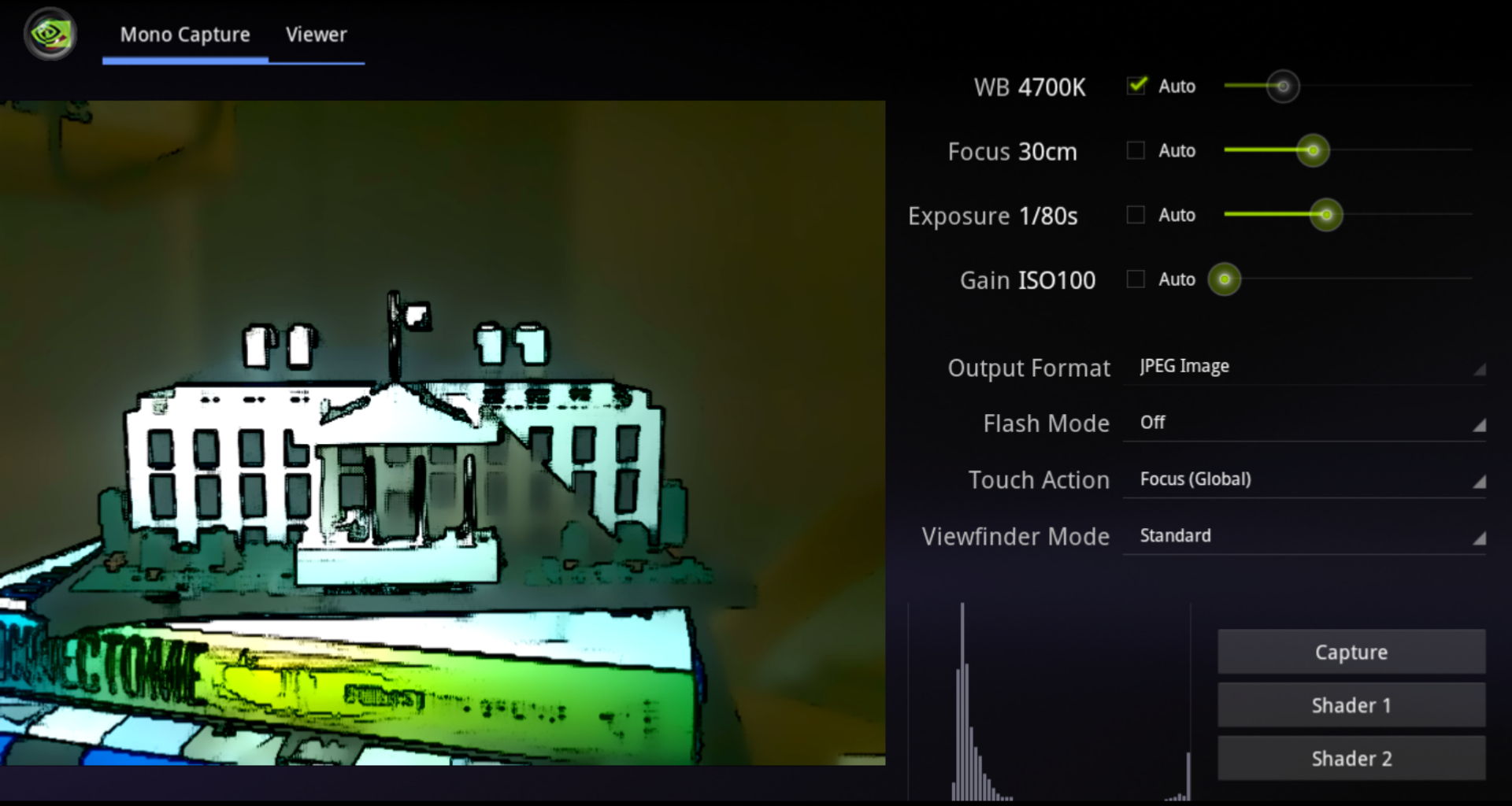
*Raskar, *et. al.* (2004)

OUR WORK: TOPICS TO BE DISCUSSED

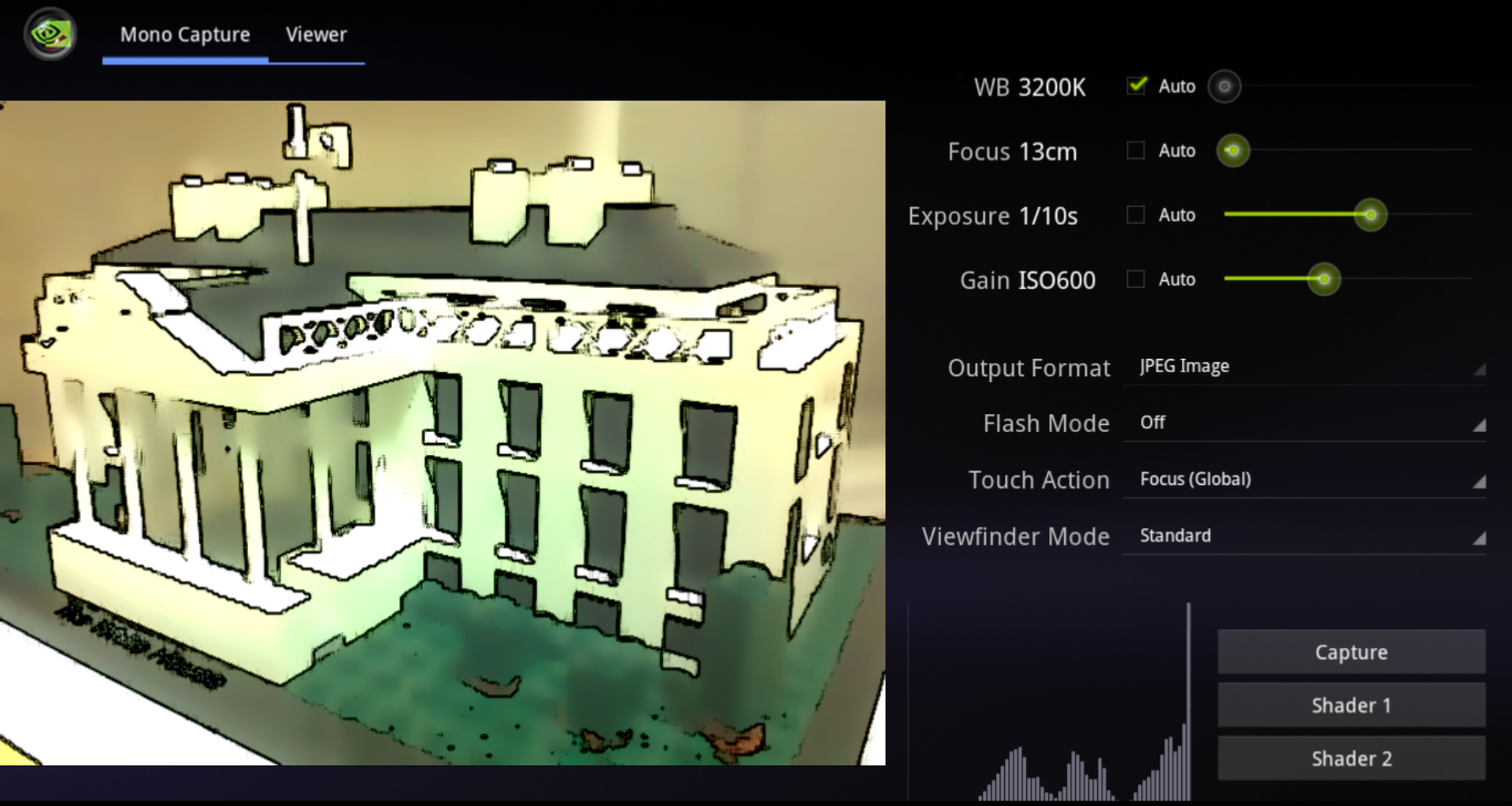
1. Demonstration of NPR viewfinder
2. How it works:
    1. Details of the NPR algorithm
    2. GPU-based “backend” for image processing
3. Integration with tablet hardware (flash)



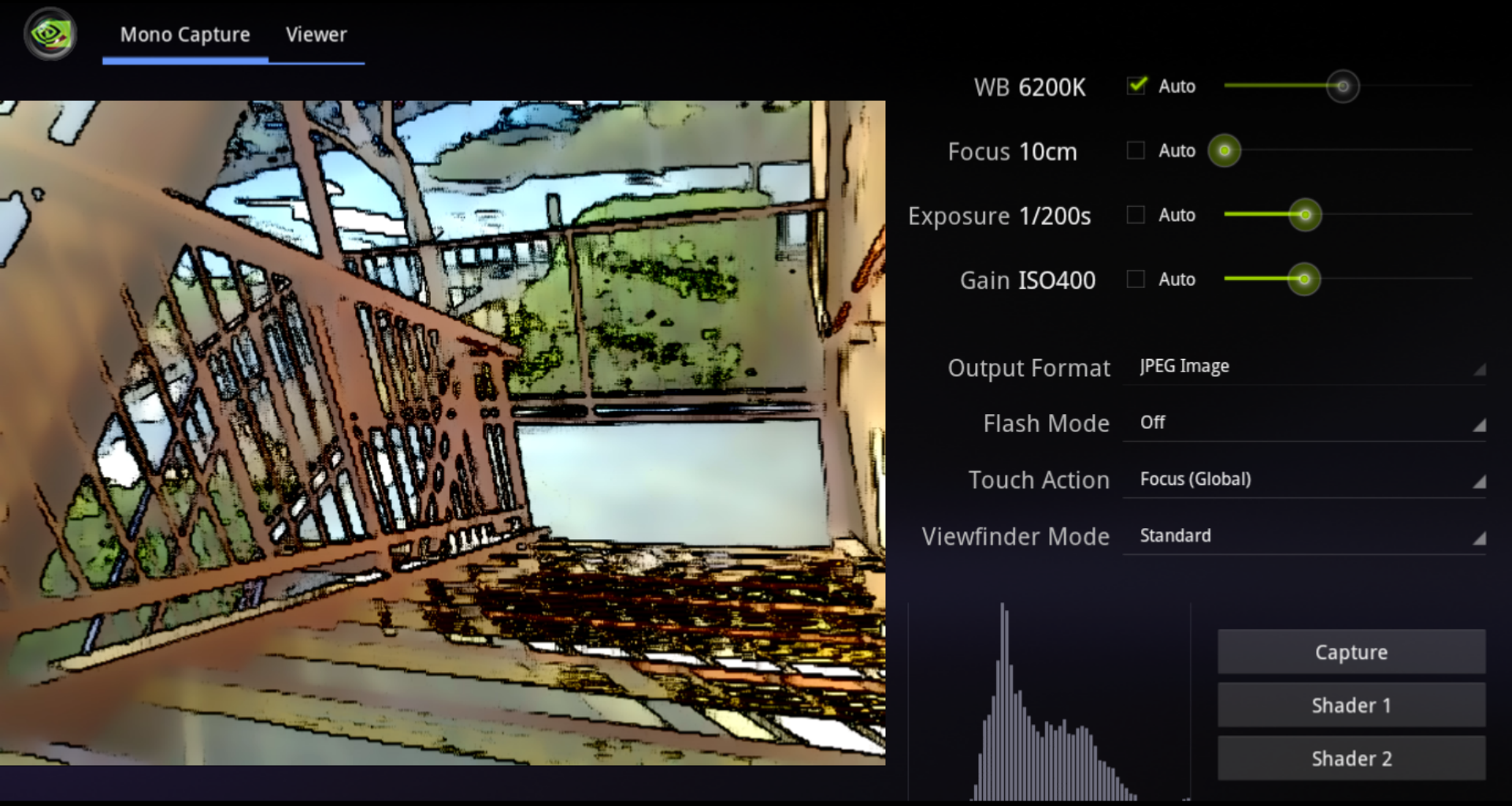
Our typical view of the world. Boring…



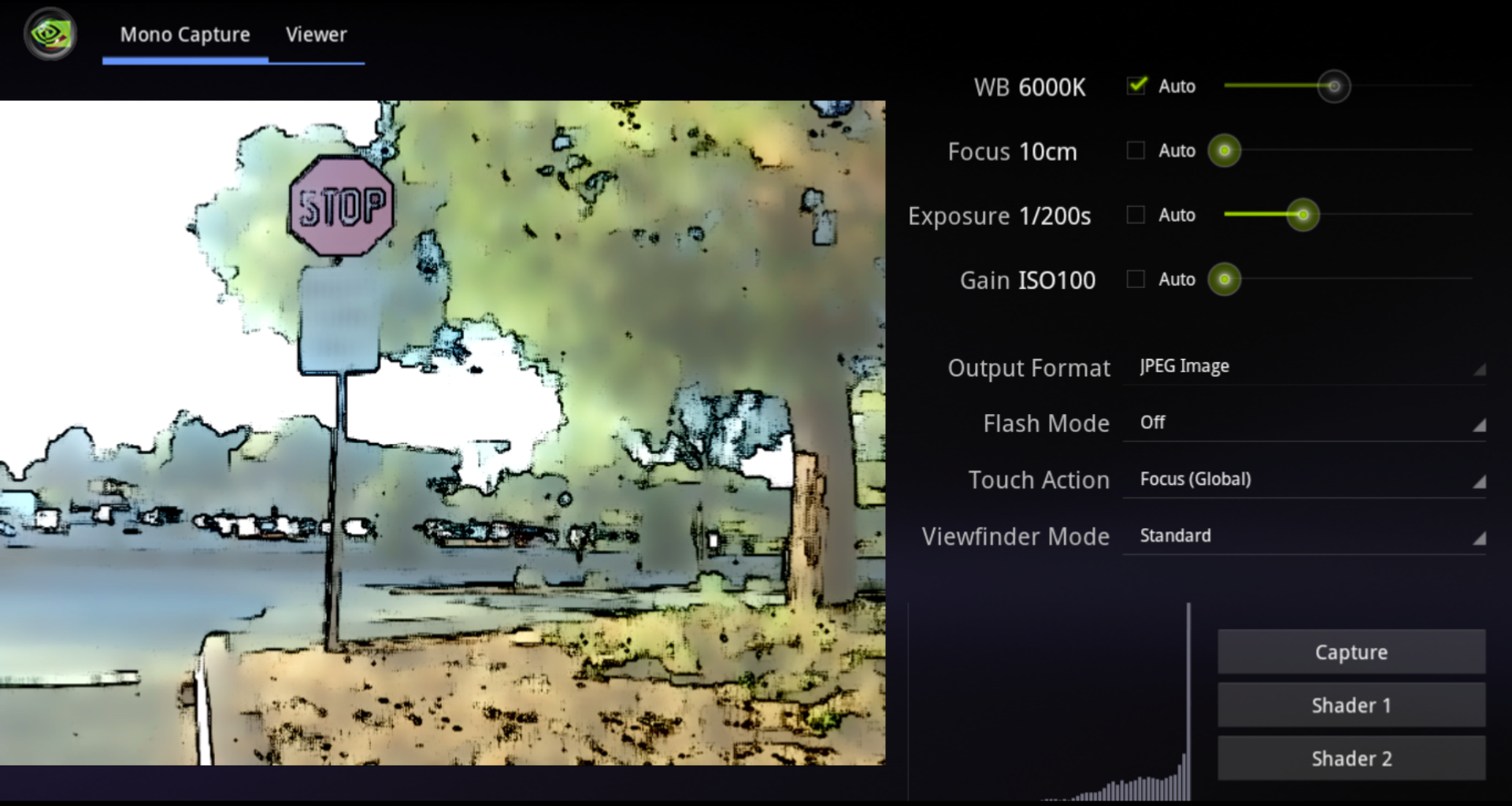
Let’s stylize our worldview!



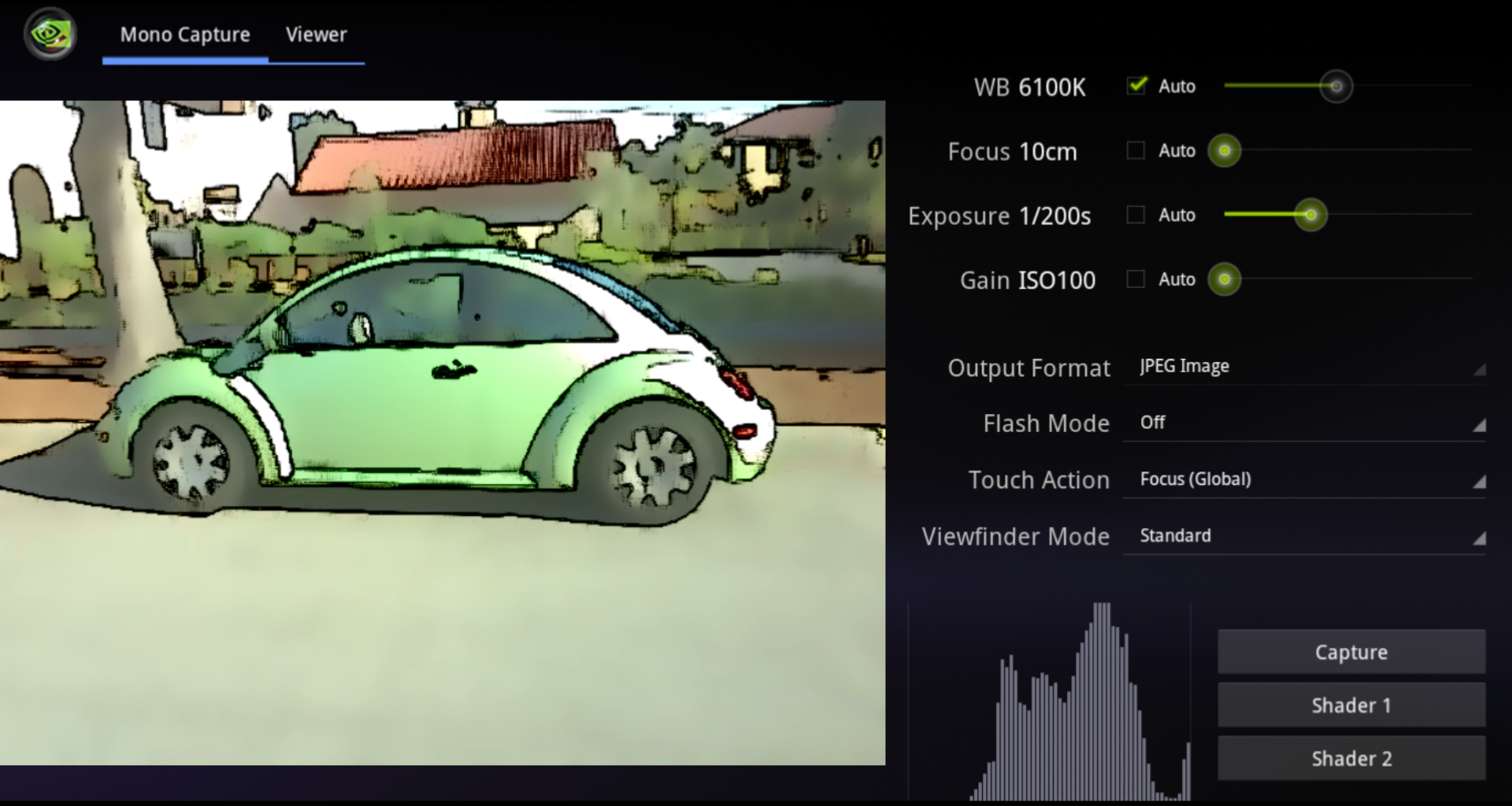
Another perspective…



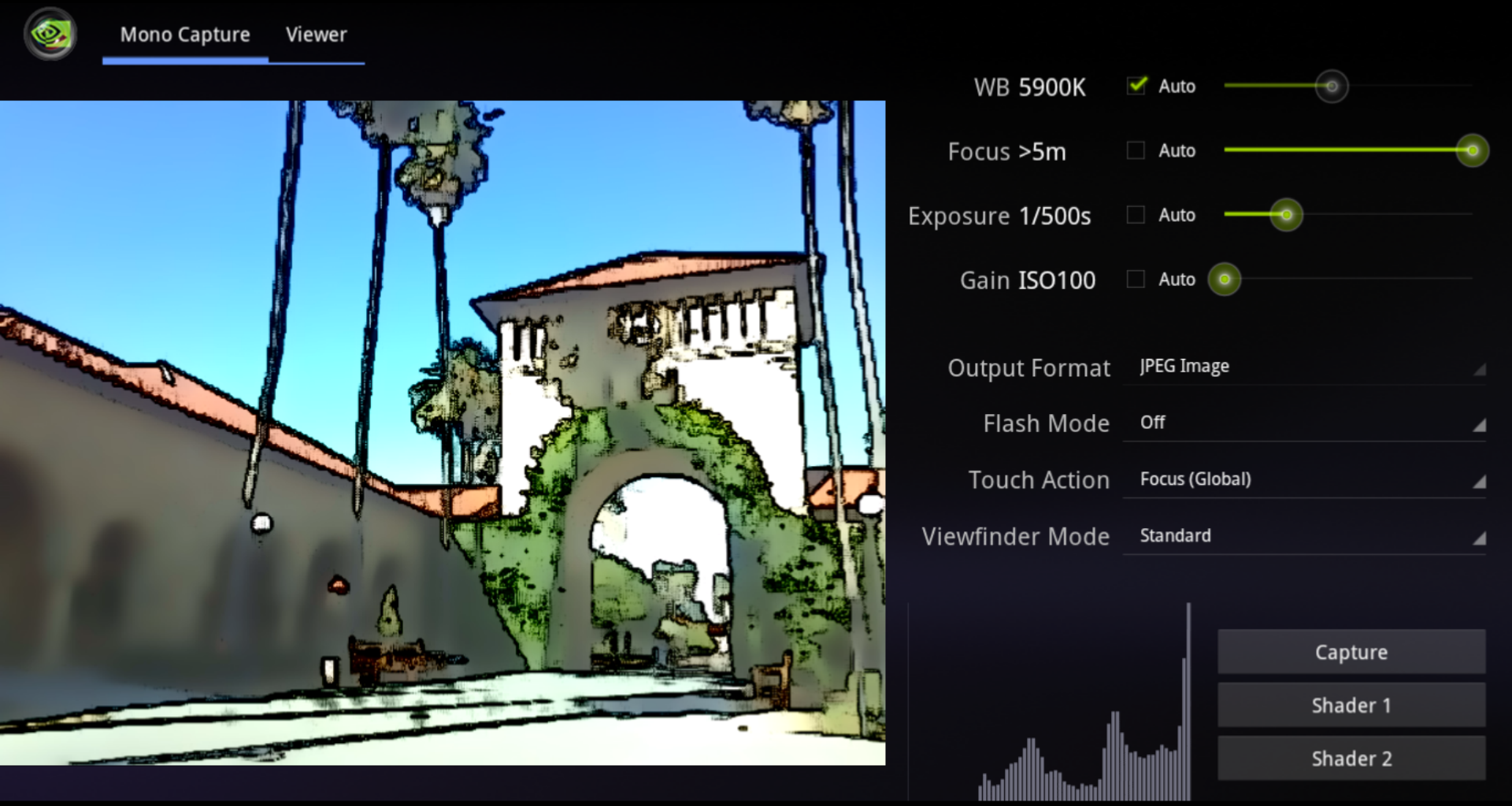
Let's take it outside…



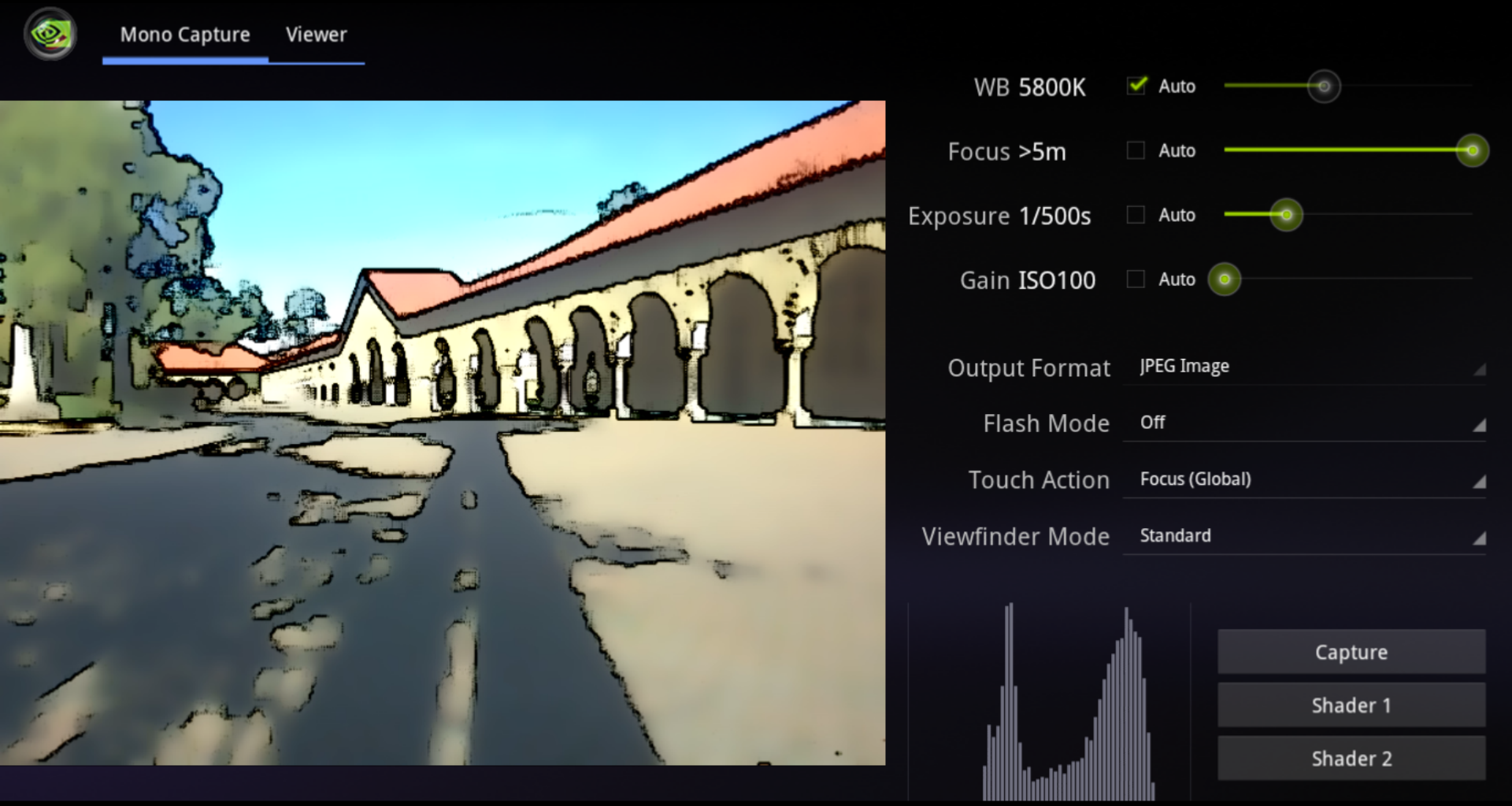
Going towards the university


Mono Capture
Viewer
WB 6100K
Auto
Focus 10cm
Auto
Exposure 1/200s
Auto
Gain ISO100
Auto
Output Format
JPEG Image
Flash Mode
Off
Touch Action
Focus (Global)
Viewfinder Mode
Standard
Capture
Shader 1
Shader 2
4:11

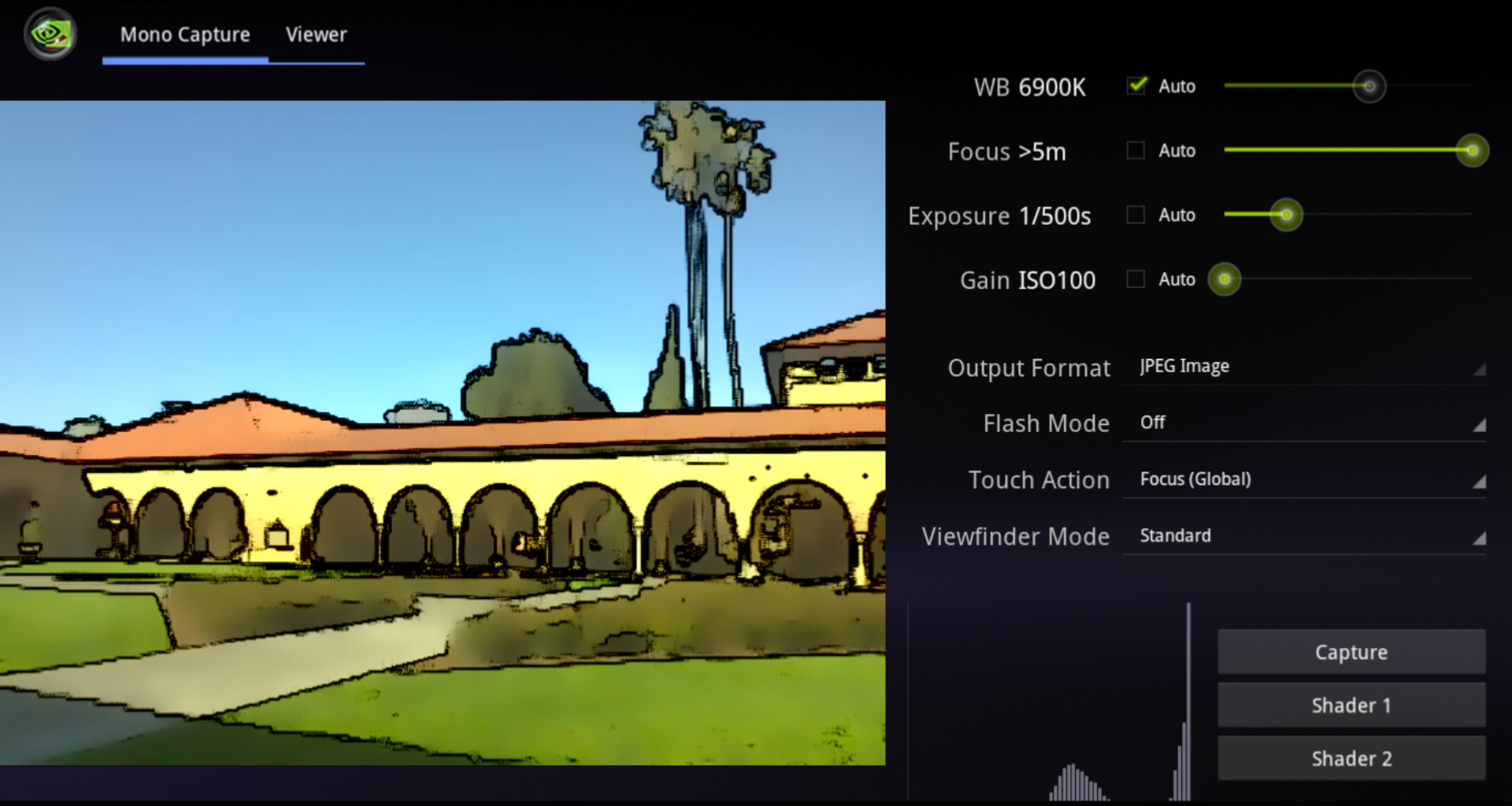
Cool car



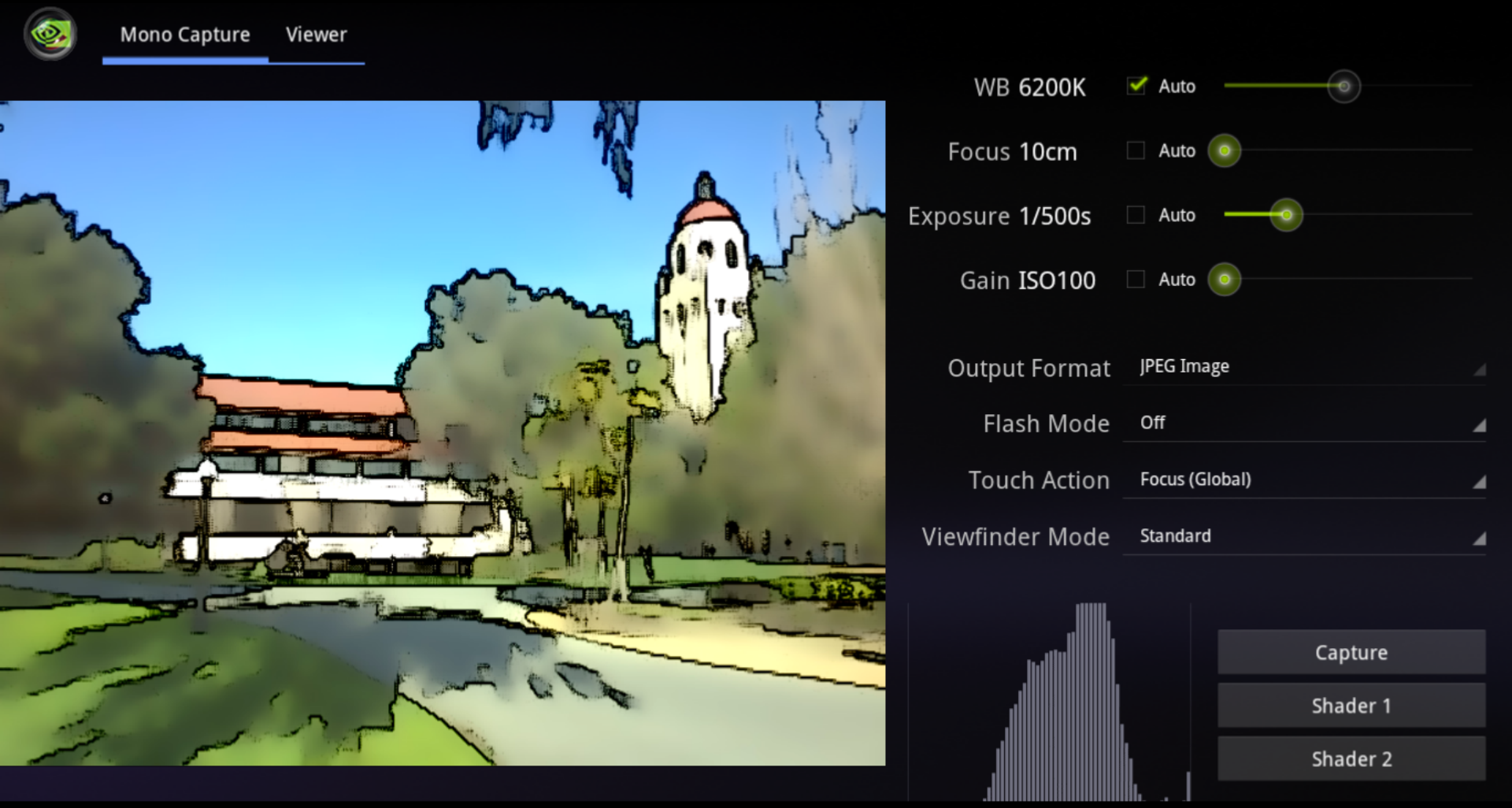
Going to the Quad...



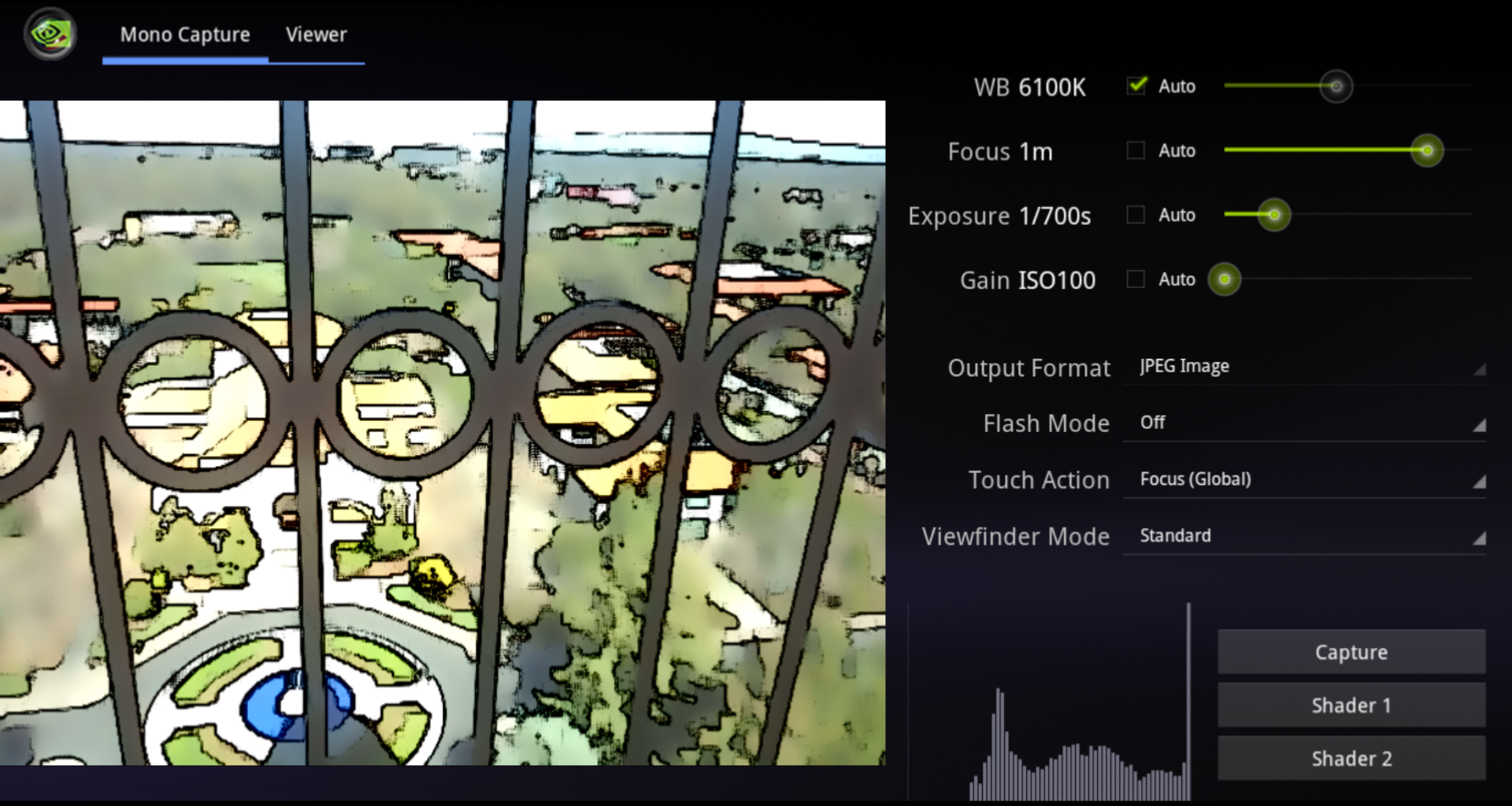
Cartoon pillars



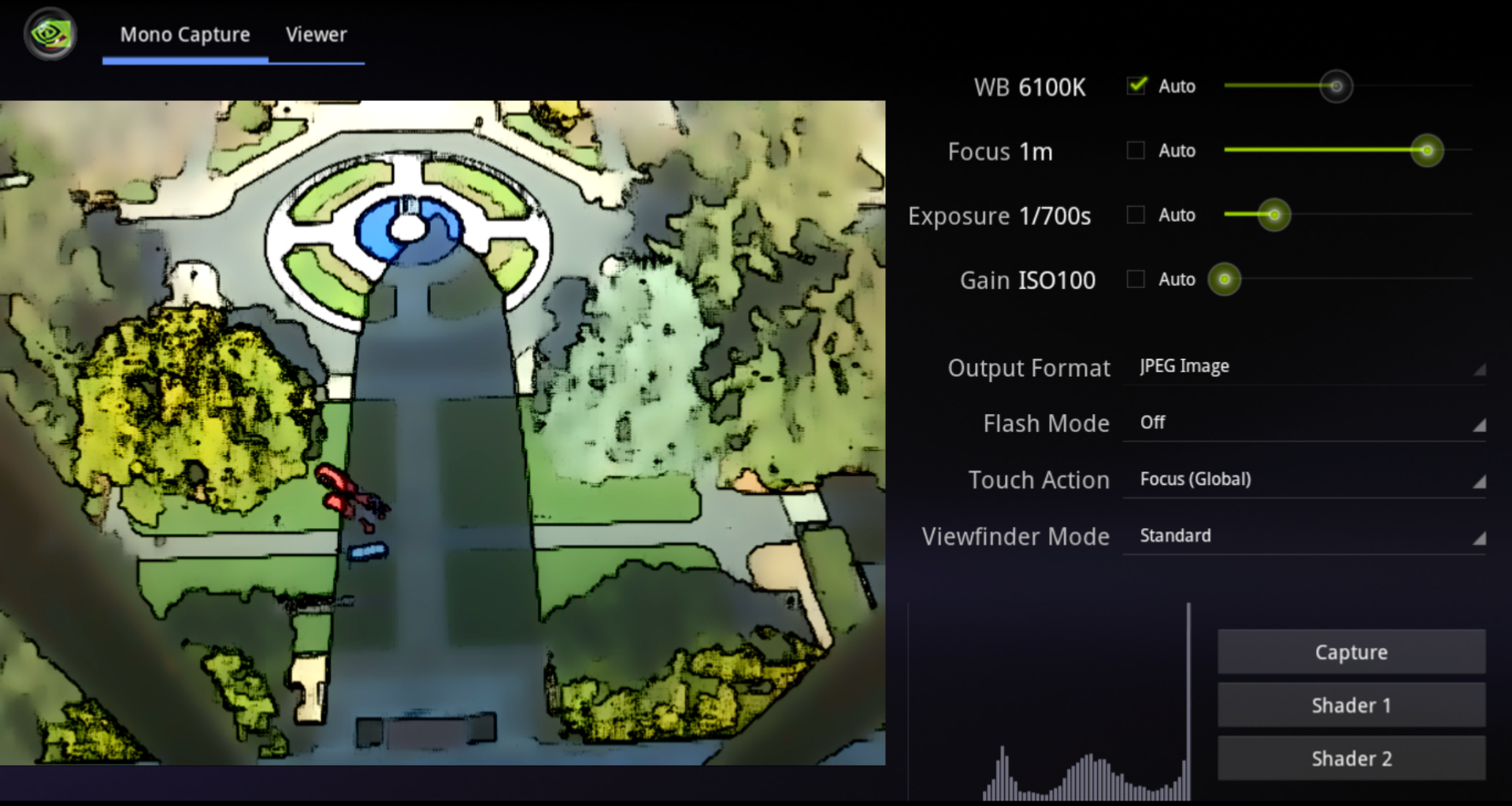
More cartoon pillars!



View of the Hoover tower



Now, in the Hoover tower



The tower's shadow

HOW IT WORKS NPR algorithm

- The world looks “cartoony” with:
  1. Color simplification
  2. Edge enhancement
  3. Color quantization (optional)

HOW IT WORKS NPR algorithm

- The world looks "cartoony" with:
  1. **Color simplification**
  2. Edge enhancement
  3. Color quantization (optional)

Use a **bilateral filter:**

- "Clustering" of nearby pixel colors
- Number of approximations available
  - Separable approx. (Pham, *et. al.* 2005)
  - No spatial weight. (Fischer 2006)

HOW IT WORKS NPR algorithm

- The world looks "cartoony" with:
  1. Color simplification
  2. **Edge enhancement**
  3. Color quantization (optional)

Use **neighbor gradients**:

- Just like the sharpness calculation in autofocus routine ("Hello Camera")
- Up-down-left-right gradients

HOW IT WORKS NPR algorithm

- The world looks "cartoony" with:
  1. Color simplification
  2. Edge enhancement
  3. **Color quantization (optional)**

Bucket the pixel value:

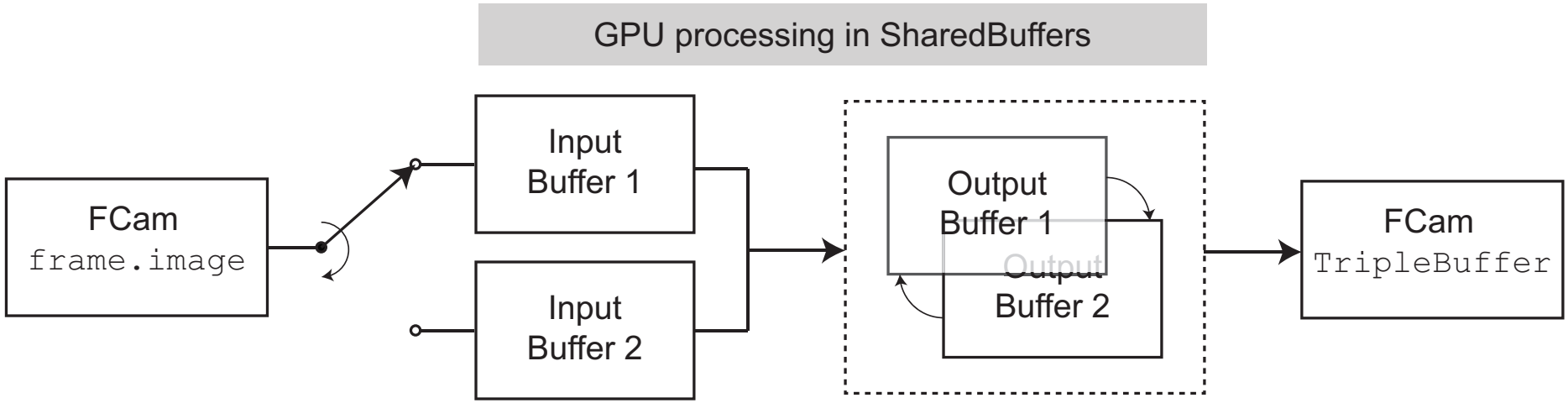
- Quantization of Y → cell-shaded look
- Quantization of U,V → funny colors

HOW IT WORKS GPU-based backend

- NPR algorithm is a sequence of image filters
  - For example:
    1. Bilateral filtering (multiple rounds)
    2. Edge detection
- For performance reasons, we perform all image processing on the GPU
  - GPU-backend implemented in `FCamAppThread`

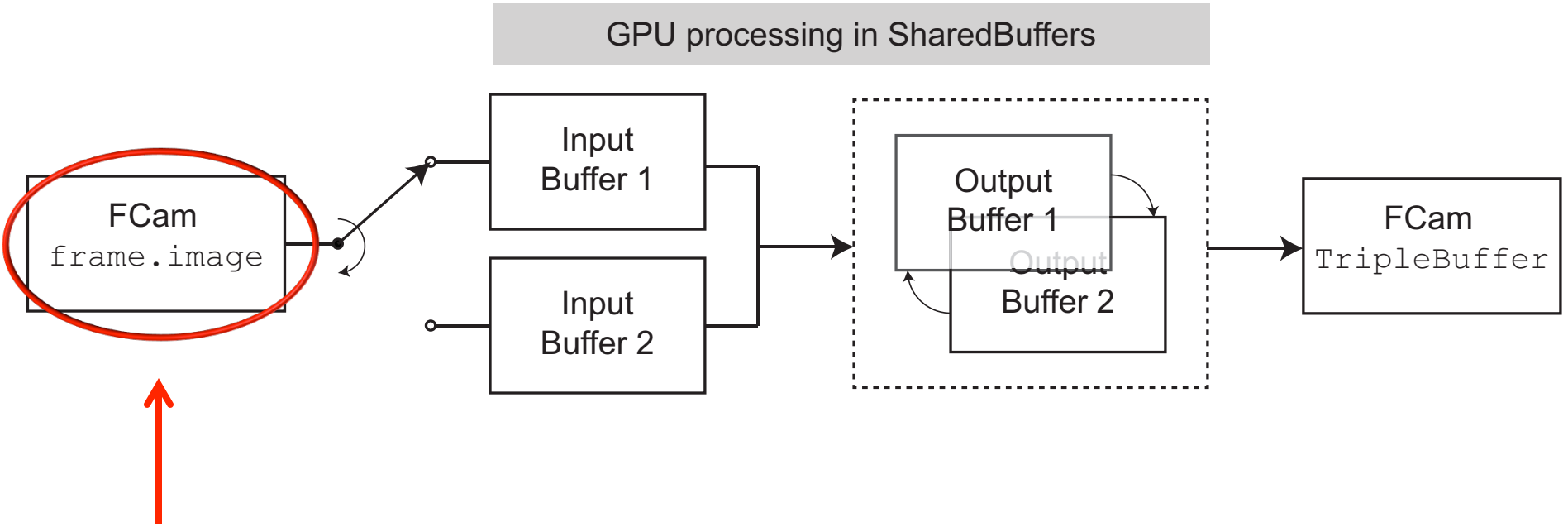
HOW IT WORKS GPU-based backend

Inside of `FCamAppThread`...



HOW IT WORKS GPU-based backend

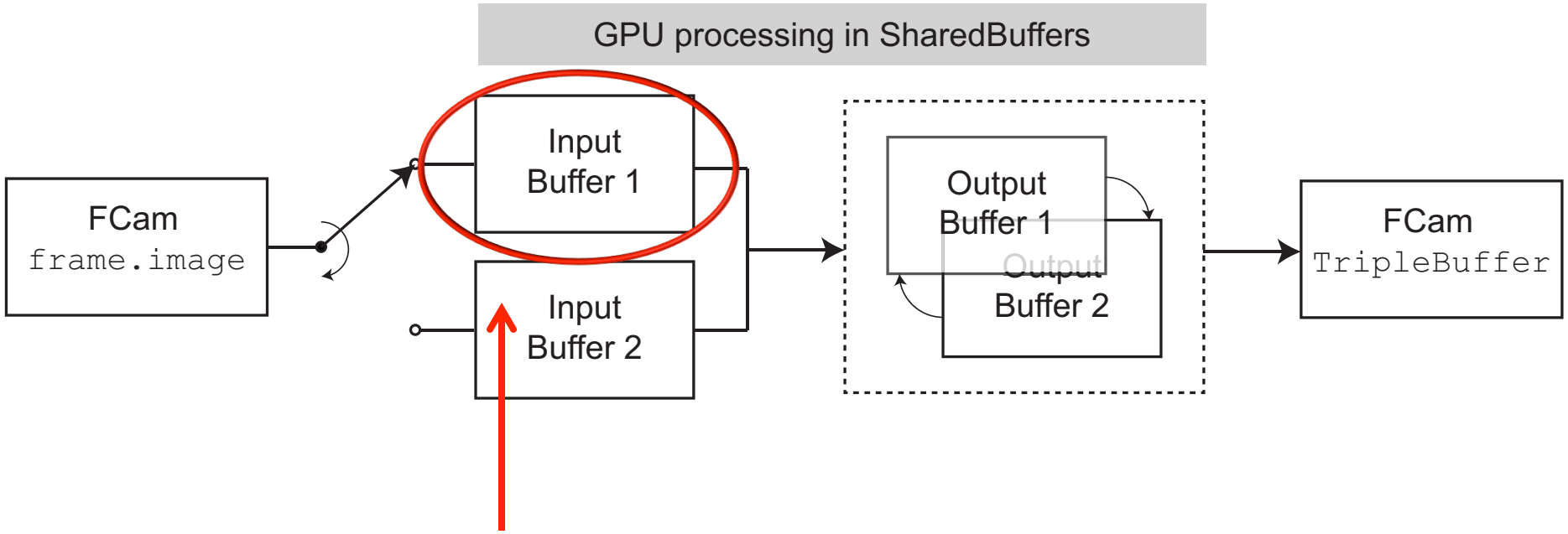
Inside of `FCamAppThread`...



- Get a single 640 x 480 viewfinder frame

HOW IT WORKS GPU-based backend

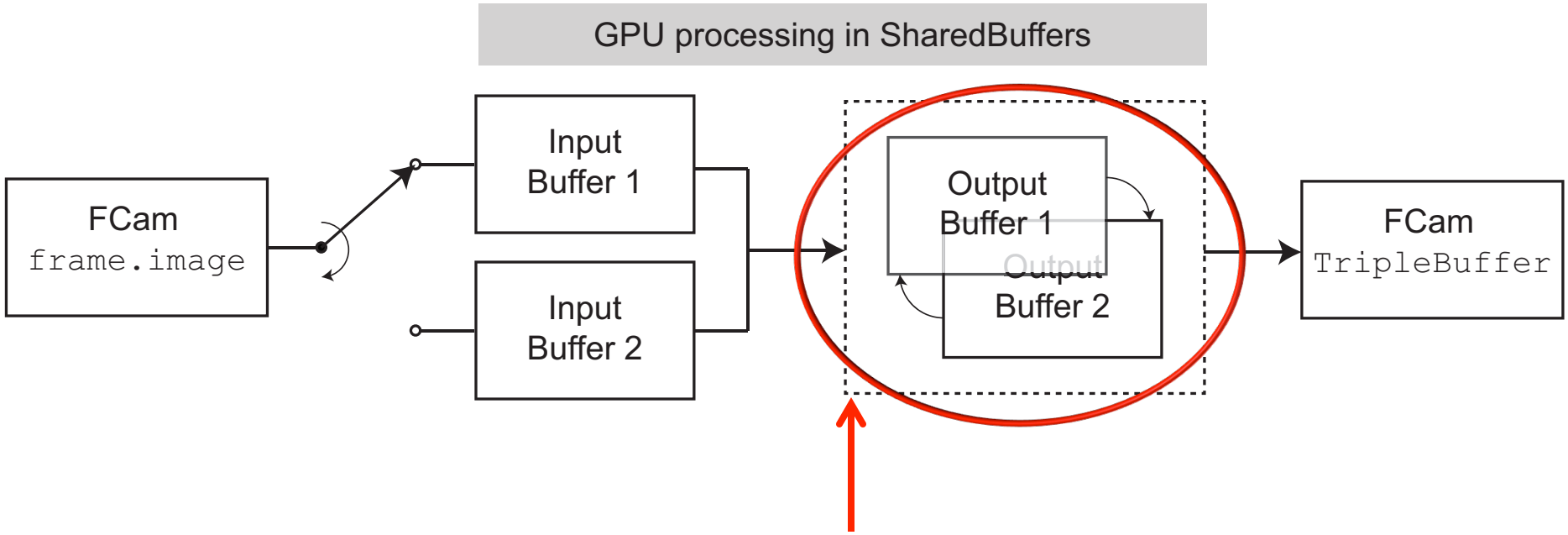
Inside of `FCamAppThread`...



- Copy frame data to "`SharedBuffer`"
- Data is visible to both CPU and GPU

HOW IT WORKS GPU-based backend

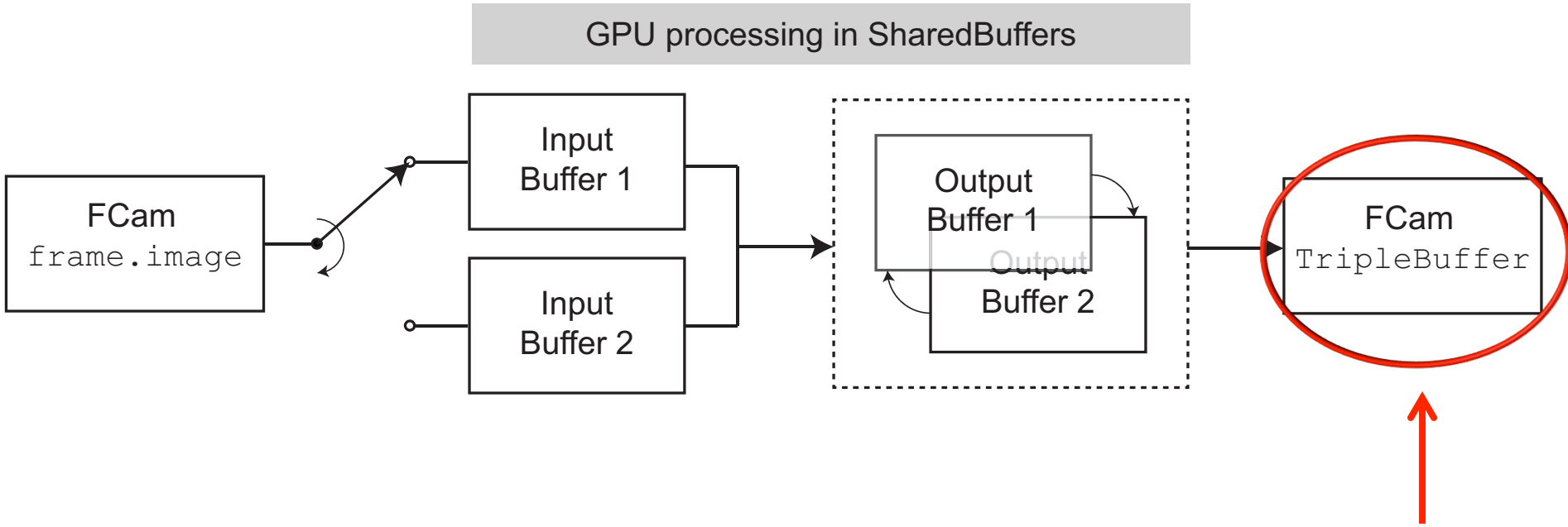
Inside of `FCamAppThread`...



- Apply sequence of shaders in GPU

HOW IT WORKS GPU-based backend

Inside of `FCamAppThread`...



- Copy result back to FCam renderer

HOW IT WORKS GPU-based backend

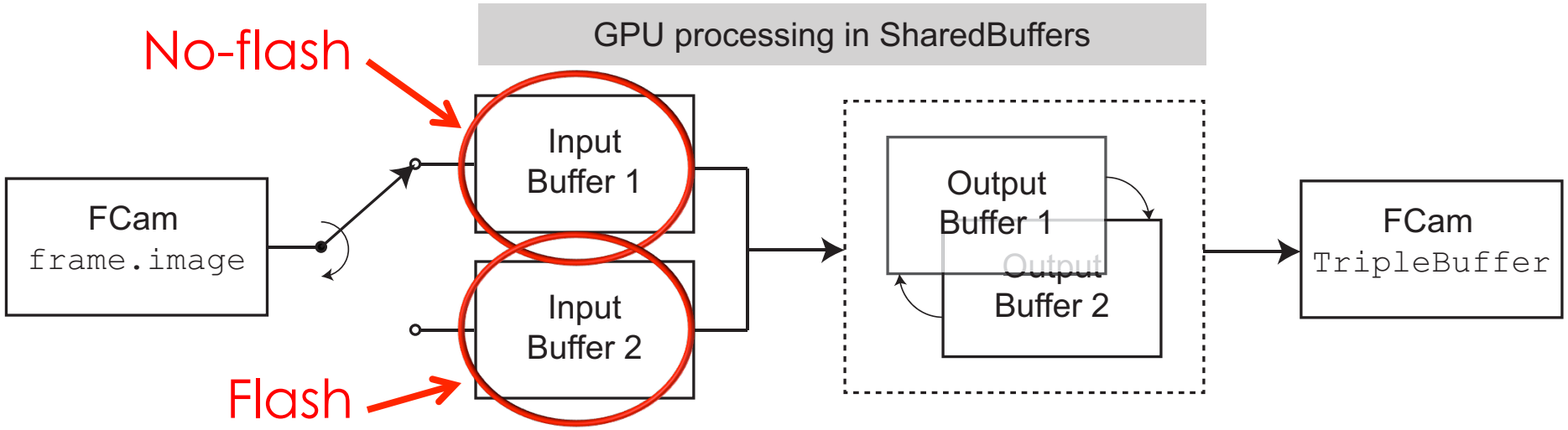
Task	Elapsed time (ms)
Copy-to-GPU (`frame.image`→`SharedBuffer`)	4.9 ± 1.7
Bilateral filtering (Single xy-pass; Kernel radius $K = 3$)	30.3 ± 9.9
Edge detection	8.2 ± 5.3
Color quantization	15.9 ± 3.6
Image mux	29.9 ± 8.7
Copy-to-CPU (`SharedBuffer`→`TripleBuffer`)	31.9 ± 8.6

Table 1: *Typical elapsed times for various sub-steps of the NPR algorithm on a single* 640×480 *viewfinder frame. Error bars are standard deviations over* 100 *measurements.*

- Typical perf: ~100 ms per frame (10 fps)

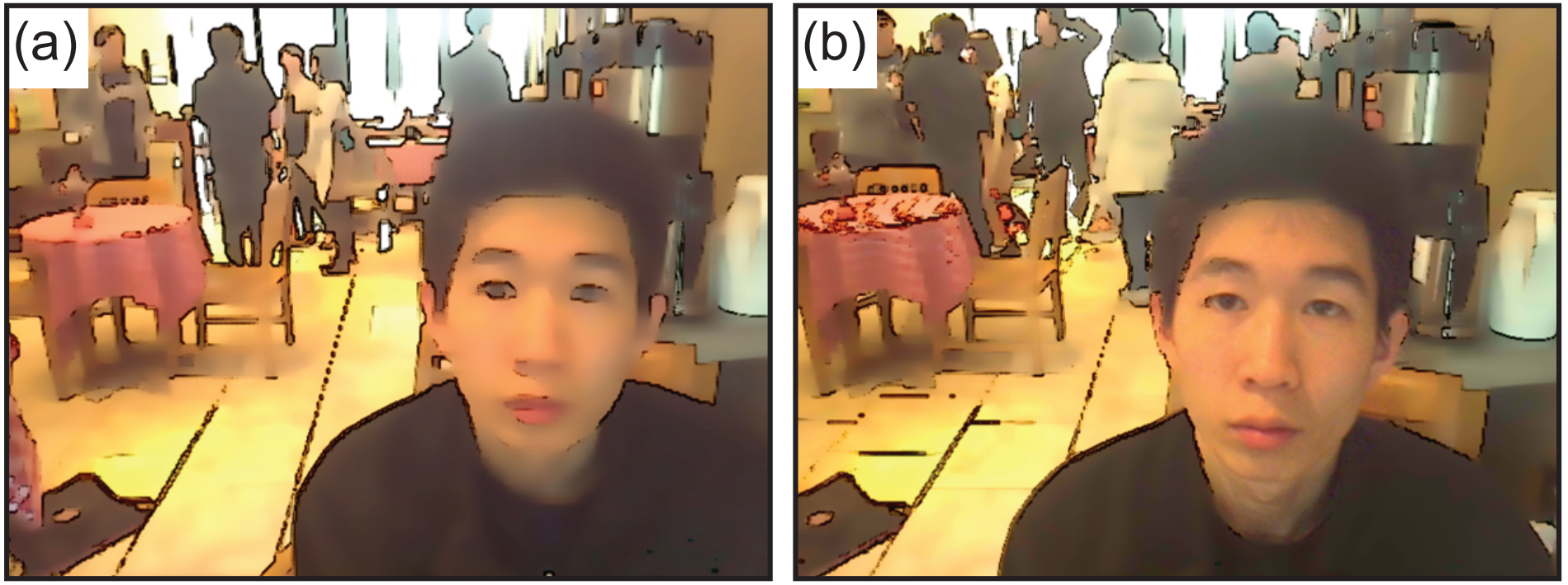
Switching gears…

FLASH-BASED SELECTIVE NPR



- Request a stream of alternating flash/no-flash video. Steer it to the right destination.
- Use flash/no-flash pair to segment foreground and background (very naïve!)

FLASH-BASED SELECTIVE NPR



CONCLUSIONS

1. Demonstration of NPR viewfinder
   - Stylizes 640 x 480 video at 10 fps
   - Bilateral filtering, edge detection, quantization
   - See: **cs478.blogspot.com** for more results

2. Backend for GPU-based image processing:
   - Generic, can be used with other filters

3. Proof-of-principle integration with hardware
   - Flash-based, background-selective NPR