Visualizing the Invisible

PyImageConf: August 27, 2018

Joseph Howse, Nummist Media, http://nummist.com

Objectives

- Use OpenCV to track a textured rigid object in 3D in real time
- Superimpose graphics atop the object – **augmented reality!**
- Use various wavelengths of visible or invisible light
    - Make the solution more robust
    - Make the solution more “magical”

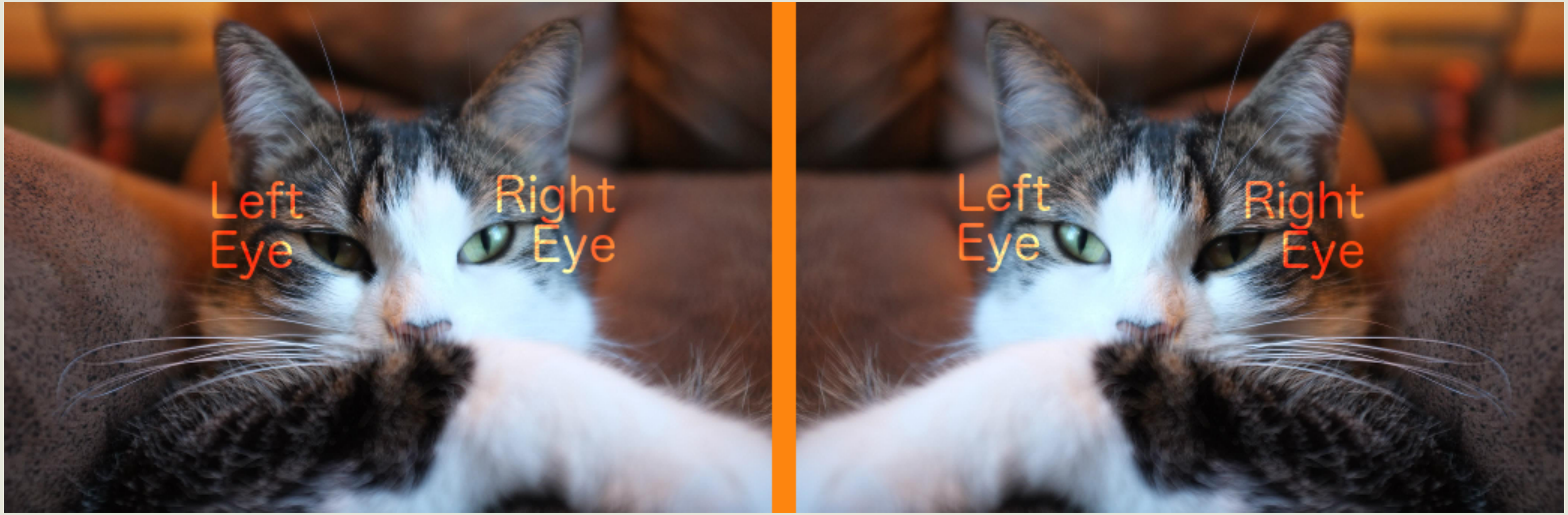
- Building on today’s talk, tomorrow’s workshop goes deeper into implementation
    - Gauge community interest in a new open-source library

Weird things about OpenCV

Weirdness in image space

- Input/output functions demand colour conversion to BGR…
  - …which is rarely a native format
- Hue (in HSV or HLS) only uses range of [0,179] in uchar
- Points are in (x,y) order and rectangles are in (x,y,w,h) order…
  - …but matrix indices are in (y,x) order
- “Left” and “right” are from viewer’s perspective
  - haarcascade_**left**eye_2splits.xml detects subject’s **right** eye
  - haarcascade_**right**eye_2splits.xml detects subject’s **left** eye

Weirdness with mirrors



Definitions based on viewer's left and right are unstable with respect to mirrors

Weirdness in 3D space

Directions in local space

	Object's +X	**Object's +Y**	**Object's +Z**
Right-handed, OpenGL	Object's **right**	Object's **up**	Object's **forward**
Left-handed, DirectX	Object's **right**	Object's **up**	Object's **back**
OpenCV	Object's **left**	Object's **down**	Object's **back**
OpenCV samples with re-flipped Z	Object's **left**	Object's **down**	Object's **forward**

Directions in view space, frontal view

	Object's +X	**Object's +Y**	**Object's +Z**
Right-handed, OpenGL	Viewer's **left**	Viewer's **up**	Viewer's **back**
Left-handed, DirectX	Viewer's **left**	Viewer's **up**	Viewer's **forward**
OpenCV	Viewer's **right**	Viewer's **down**	Viewer's **forward**
OpenCV samples with re-flipped Z	Viewer's **right**	Viewer's **down**	Viewer's **back**

- Relative to OpenGL, OpenCV flips <u>all three</u> axis directions!
- Some of OpenCV's official samples re-flip Z before drawing axes but leave X and Y unchanged

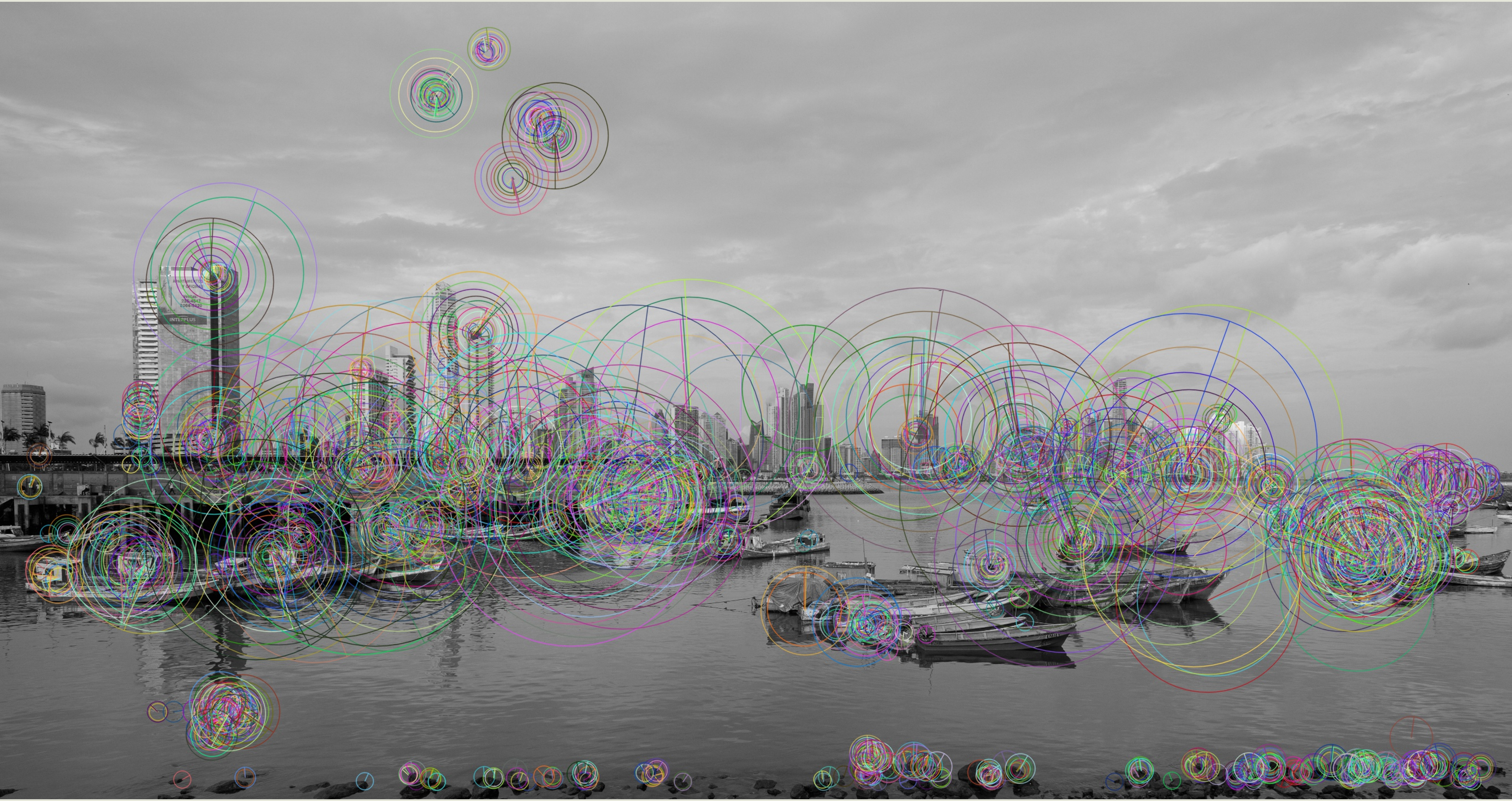
Tracking a textured rigid object in 3D in real time

Demo source code is online at
https://github.com/JoeHowse/VisualizingTheInvisible

Processing a reference image

1. Load image
2. Resize
   - Bicubic
3. Convert to grayscale
   - Intensity: simple average of R, G, B
4. Detect keypoints & compute descriptors
   - ORB
   - Balanced by region
5. Draw visualization of keypoints
6. Save visualization
7. Map keypoints & vertices to 3D reference model
   - Plane
   - Cuboid
   - Cylinder

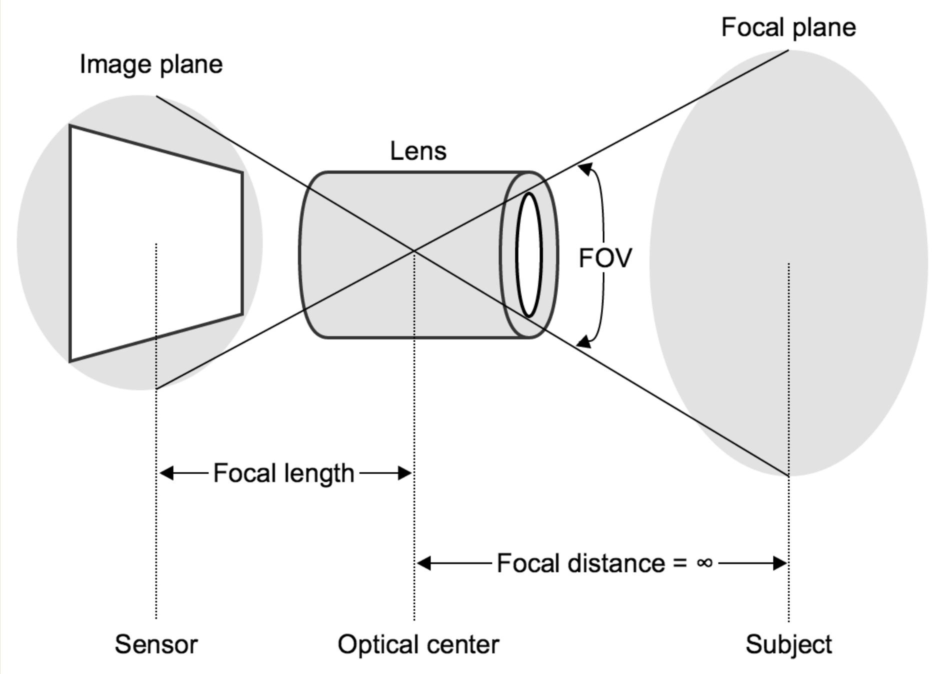
Title: Panama City Fish Market **Photographer:** Joseph Howse January 13, 2018


APARTAMENTOS
Y OFICINAS
Ventas:
6064-5420
INTERPLUS

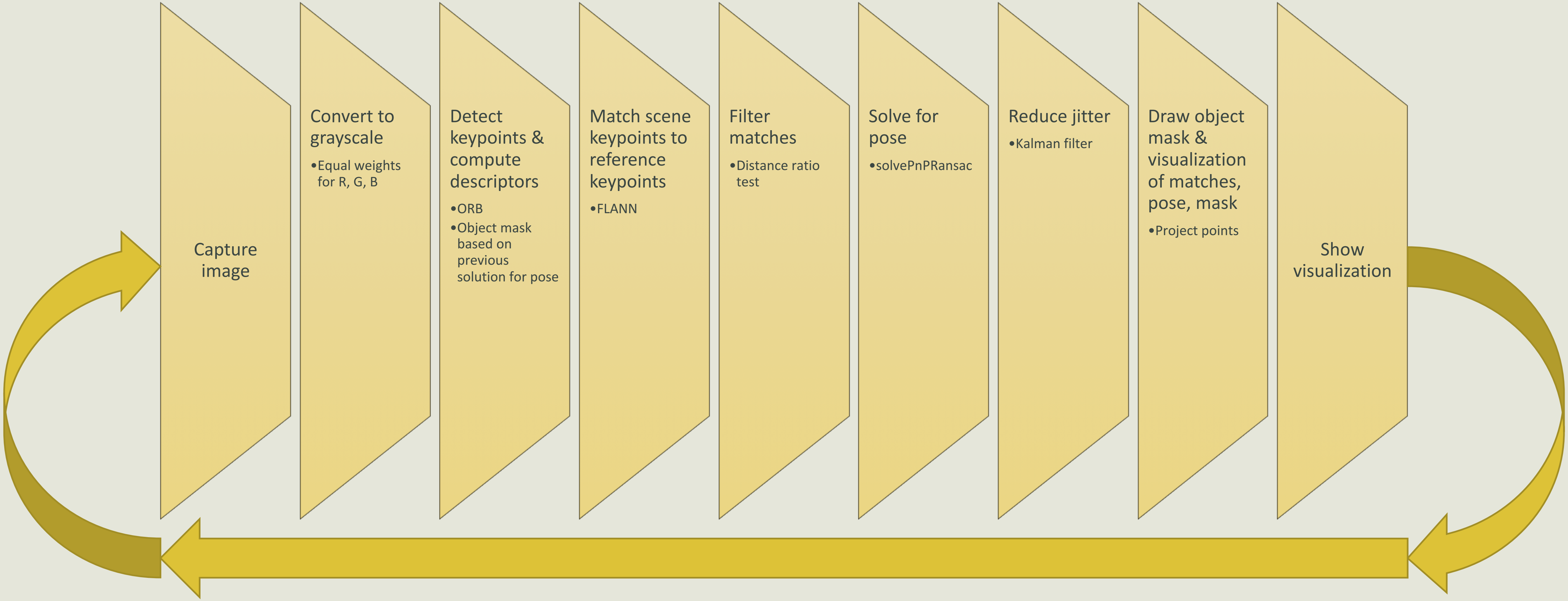
Estimating the camera matrix

The ideal camera matrix		
f	0	$c_x = w/2$
0	f	$c_y = h/2$
0	0	1

- f, c_x, c_y must be in same units, e.g. pixels
- f is focal length
- (c_x, c_y) is center or "principal point" of image within image plane
- (w, h) are width, height of image plane
- α is diagonal field of view (FOV)
- (θ, ϕ) are horizontal, vertical field of view (FOV)
- $f = \frac{\sqrt{w^2+h^2}}{2(\tan\frac{\alpha}{2})} = \frac{\sqrt{w^2+h^2}}{2\sqrt{\left(\tan\frac{\theta}{2}\right)^2+\left(\tan\frac{\phi}{2}\right)^2}}$



Processing a scene



When does tracking deteriorate?

- Lighting is dim
  - Image is noisy
- Lighting is harsh
  - Shadows
  - Specular highlights
- Object is occluded
- Object is angled away from camera
- Object is distant or small
- Object is curved
  - Cylindrical tracking is “experimental” in current implementation
- Lens is out of focus
- Lens distorts
- Matches are nearly collinear
  - Pose estimate spins, as one axis of rotation is indeterminate

Other algorithms not covered in this demo

- Alternatives to intensity grayscale conversion
  - Gamma-corrected conversions may produce more inliers but are slower
    - Samuel Macêdo, Givânio Melo, and Judith Kelner. "A comparative study of grayscale conversion techniques applied to SIFT descriptors". *SBC Journal on Interactive Systems*, vol. 6, no. 2, 2015
  - Equalization, adaptive grayscale conversion, CLAHE – better or worse for producing inliers?
- Alternatives to ORB
  - SIFT, SURF, KAZE, AKAZE may produce more inliers but are slower
    - Zoltan Pusztai and Levente Hajder, "Quantitative Comparison of Feature Matchers Implemented in OpenCV3". 21st Computer Vision Winter Workshop, Rimske Toplice, Slovenia, February 3-5, 2016
- Multiple reference images
- Fallback to optical flow + homography
  - Saves cost by avoiding redetection of keypoints every frame
- Fallback to inertial navigation
  - Rotation from accelerometer, gyroscope, magnetometer

Experiments in ultraviolet

An ultraviolet webcam: XNiteUSB2S-MUV (US$135)

Visible: The petals of the black-eyed Susan are solid yellow

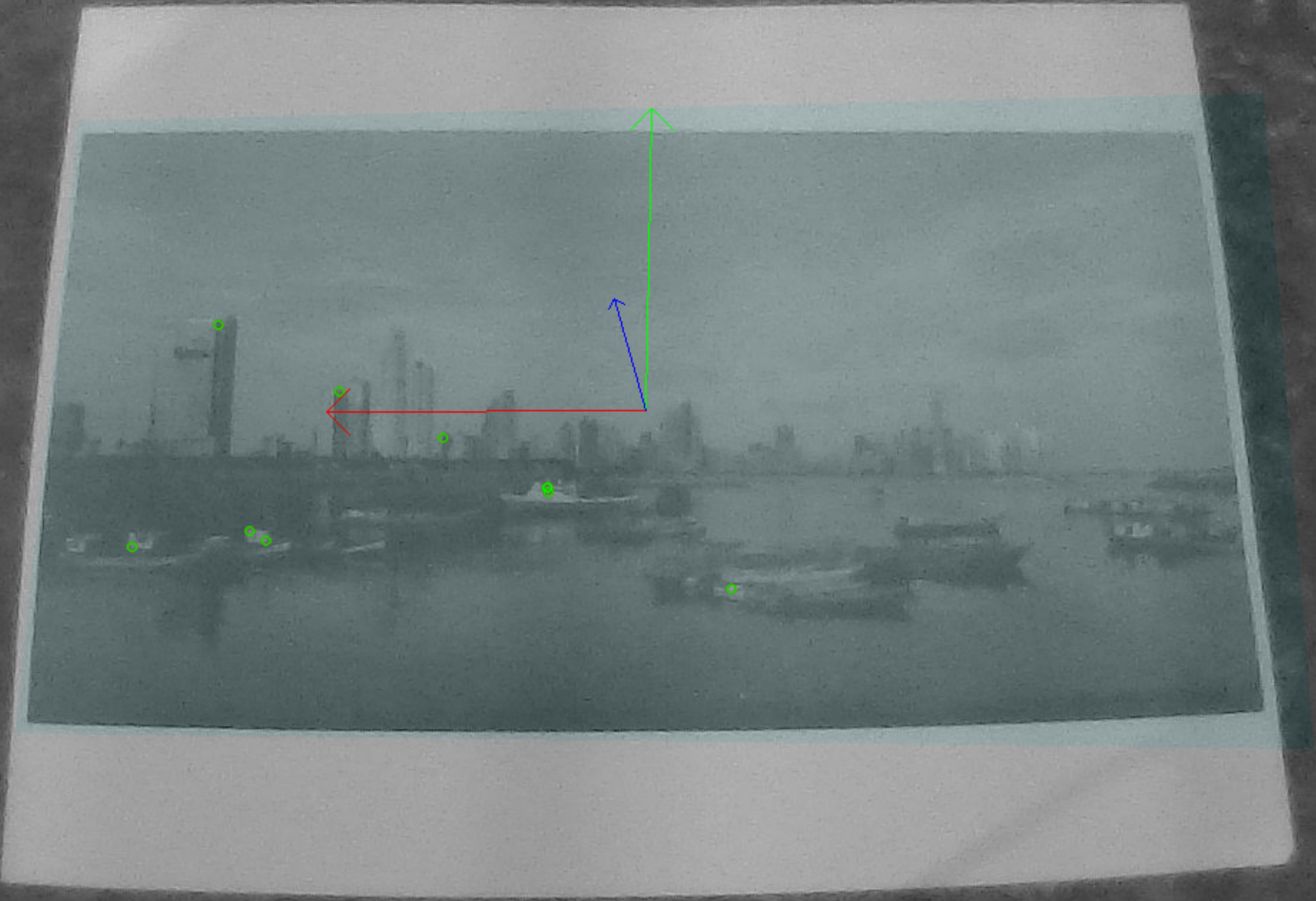
Ultraviolet: The petals of the black-eyed Susan are dark near the centre

- Lens filter (shown above) blocks nearly all visible light but allows UV to pass
- Sold by MaxMax.com: https://maxmax.com/maincamerapage/uvcameras/usb2-small

Subjective evaluation of XNiteUSB2S-MUV

- 3D tracking is feasible under some conditions…
  - …even with textures designed for visible spectrum
- Requires bright sunlight or bright artificial UV light
  - At ground level, sunlight is 53% IR, 44% visible, only 3% UV
  - Normal indoor lights emit too little UV to form images
- Low fidelity
  - Low contrast, haze
  - Noise
  - Barrel distortion
- More experiments needed
  - UV pigments and application techniques
  - Quartz lens – expensive but transmits UV much better

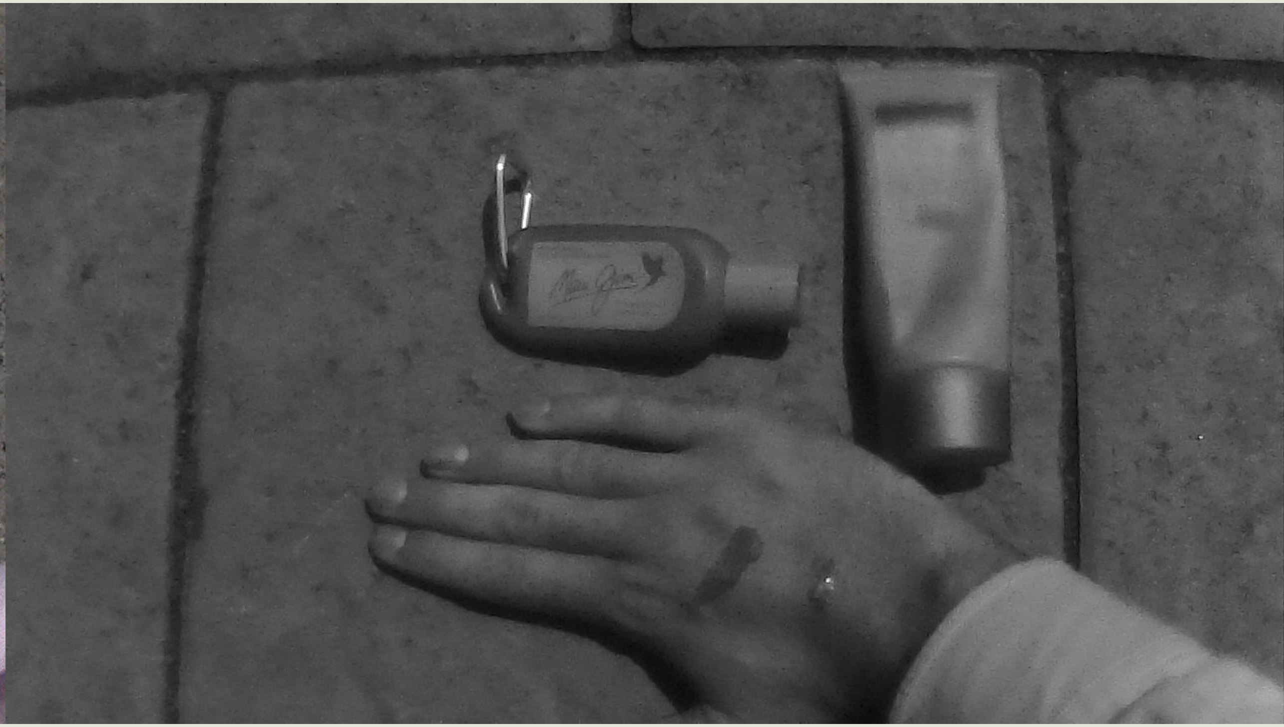
Let’s look at natural UV light, outdoors at 1 p.m. on a sunny day…



A commonplace UV pigment

Visible: Blobs of sunscreen are white

Ultraviolet: Blobs of sunscreen are dark

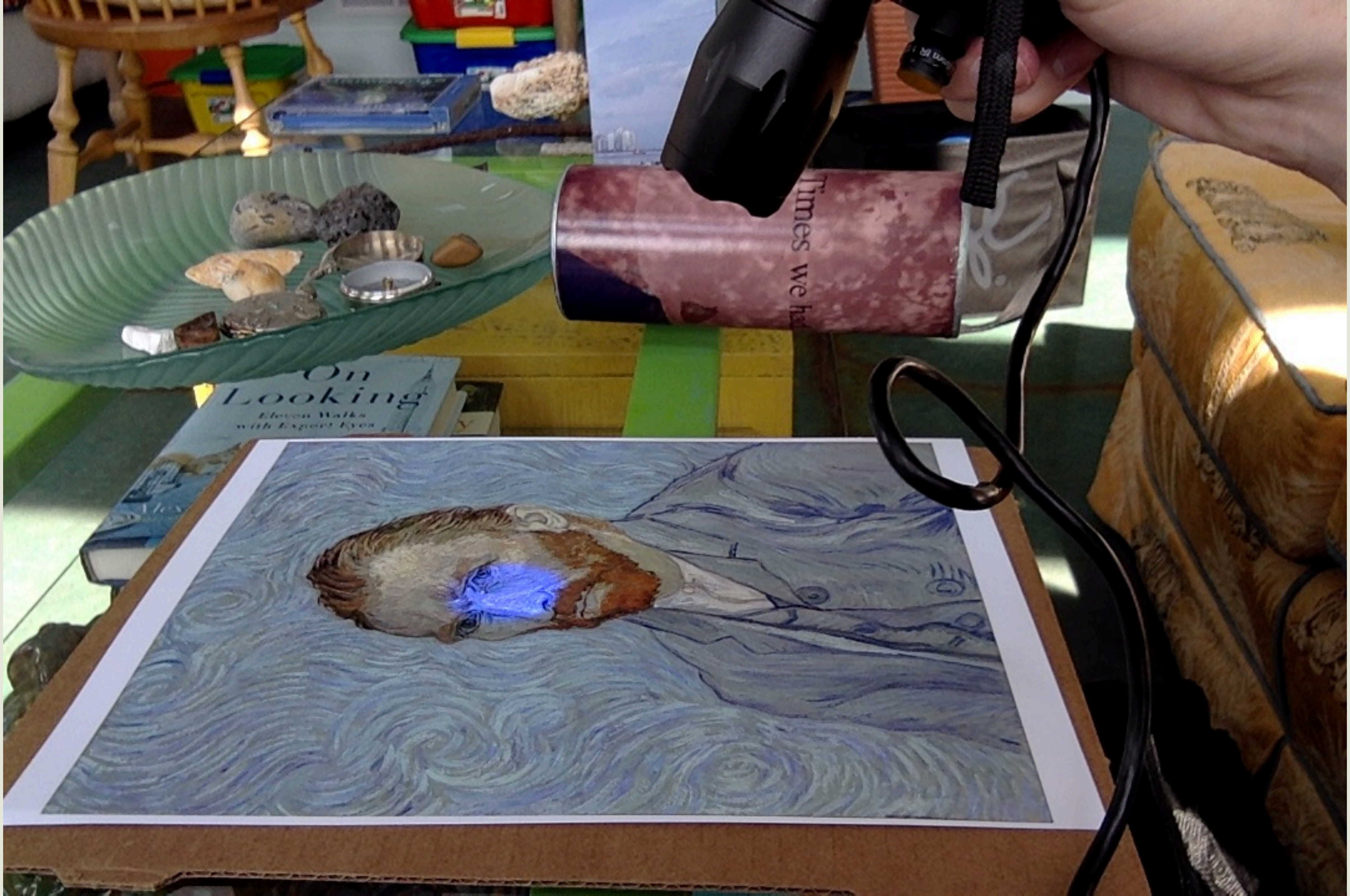


A commonplace UV pigment

Visible: Rubbed-in sunscreen is transparent

Ultraviolet: Rubbed-in sunscreen is dark

Let’s look at light from an XNiteFlashF 365nm UV flashlight (US$115), indoors…


On
Looking
Eleven Walks
with Expert Eyes

UV to red fluorescent ink (US$10 for a pen)

Without UV flashlight: Text is invisible on photo paper, nearly invisible on matboard

With UV flashlight: Text is visible in red

UV to red fluorescent ink (US$10 for a pen)

Outdoor sunlight without UV flashlight: Text is nearly invisible

Outdoor sunlight with narrowly focused UV flashlight: Text is faintly visible in red

Let’s track a UV-to-red fluorescent drawing on blank photo paper…

Title:
Algorithmic
Squiggle Reality

Artist:
Janet Howse

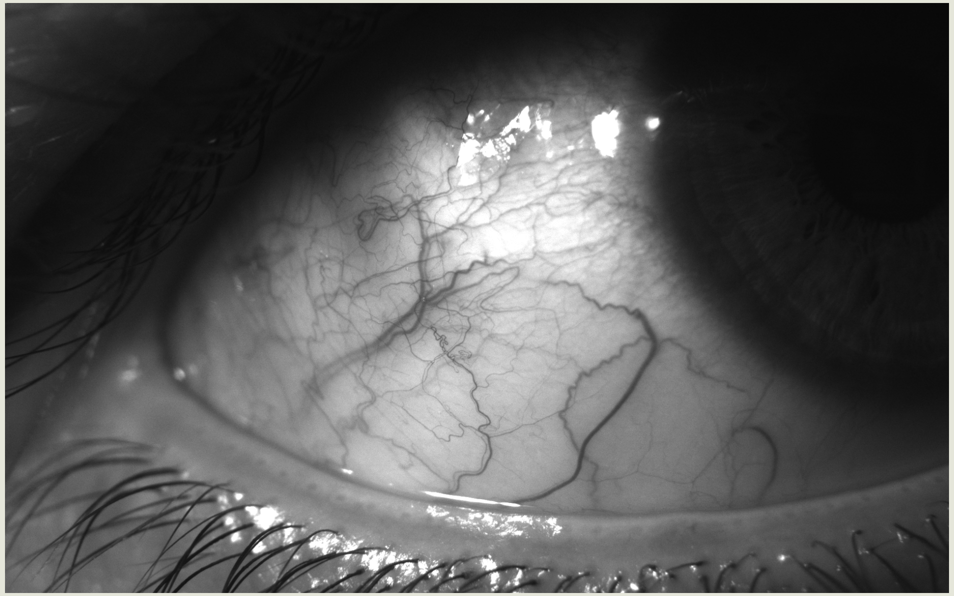
August 24, 2018

Experiments in monochrome

A monochrome industrial camera: Grasshopper3 GS3-U3-23S6M-C (US$995)

Library, shot with 12.5mm lens (US$41)

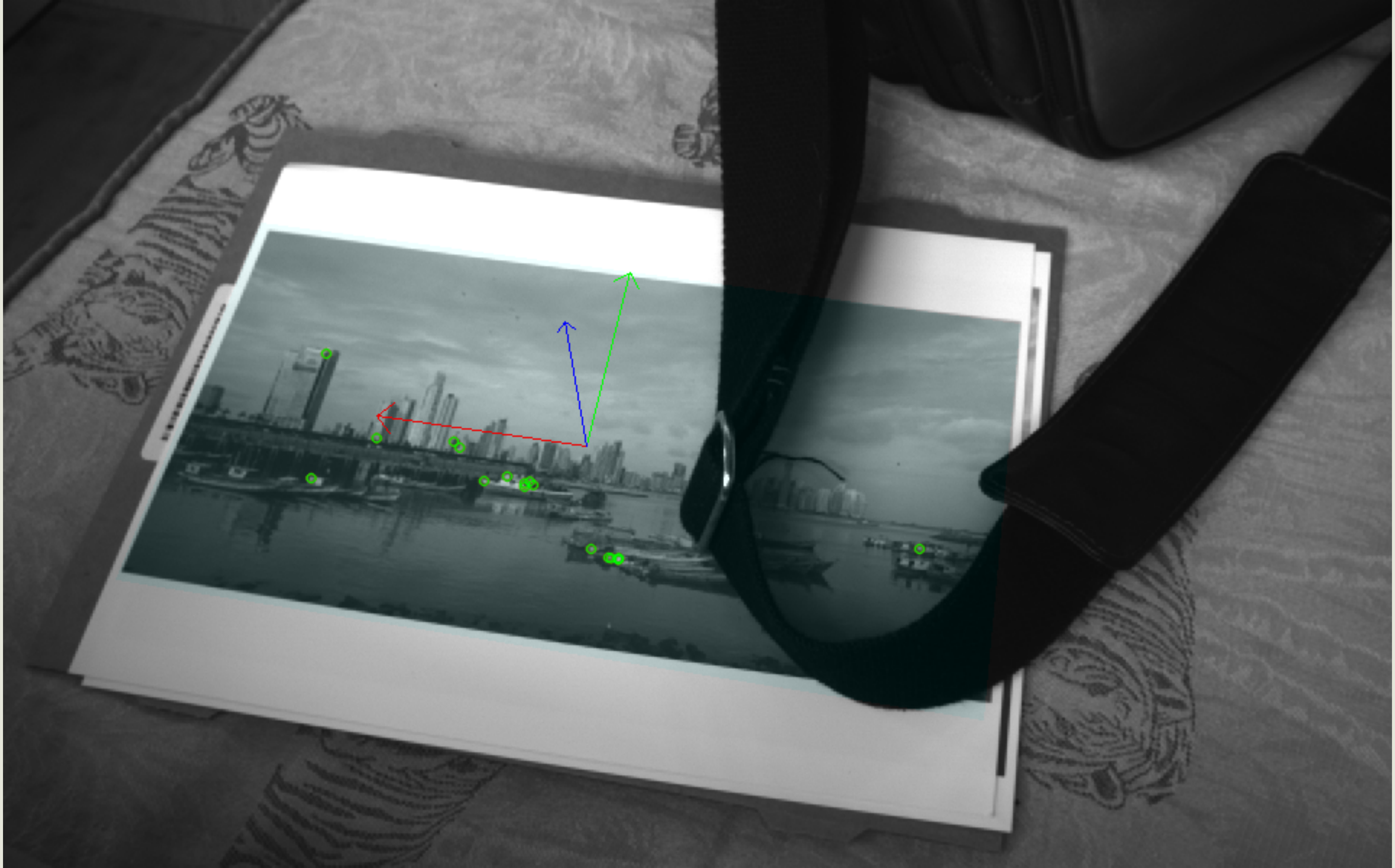
Eyeball, shot with 20mm lens (US$35) and 10mm extension tube (US$10)

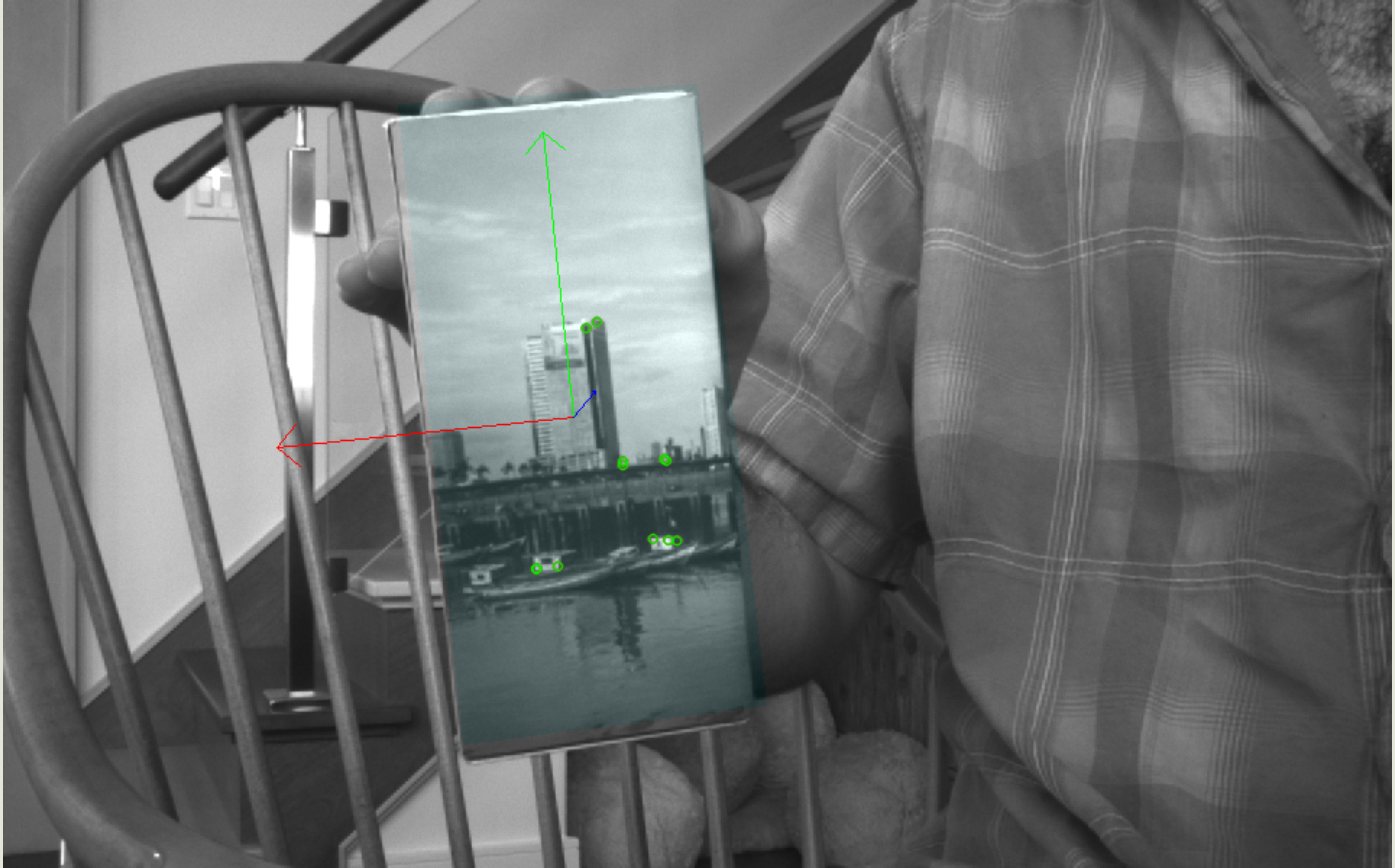


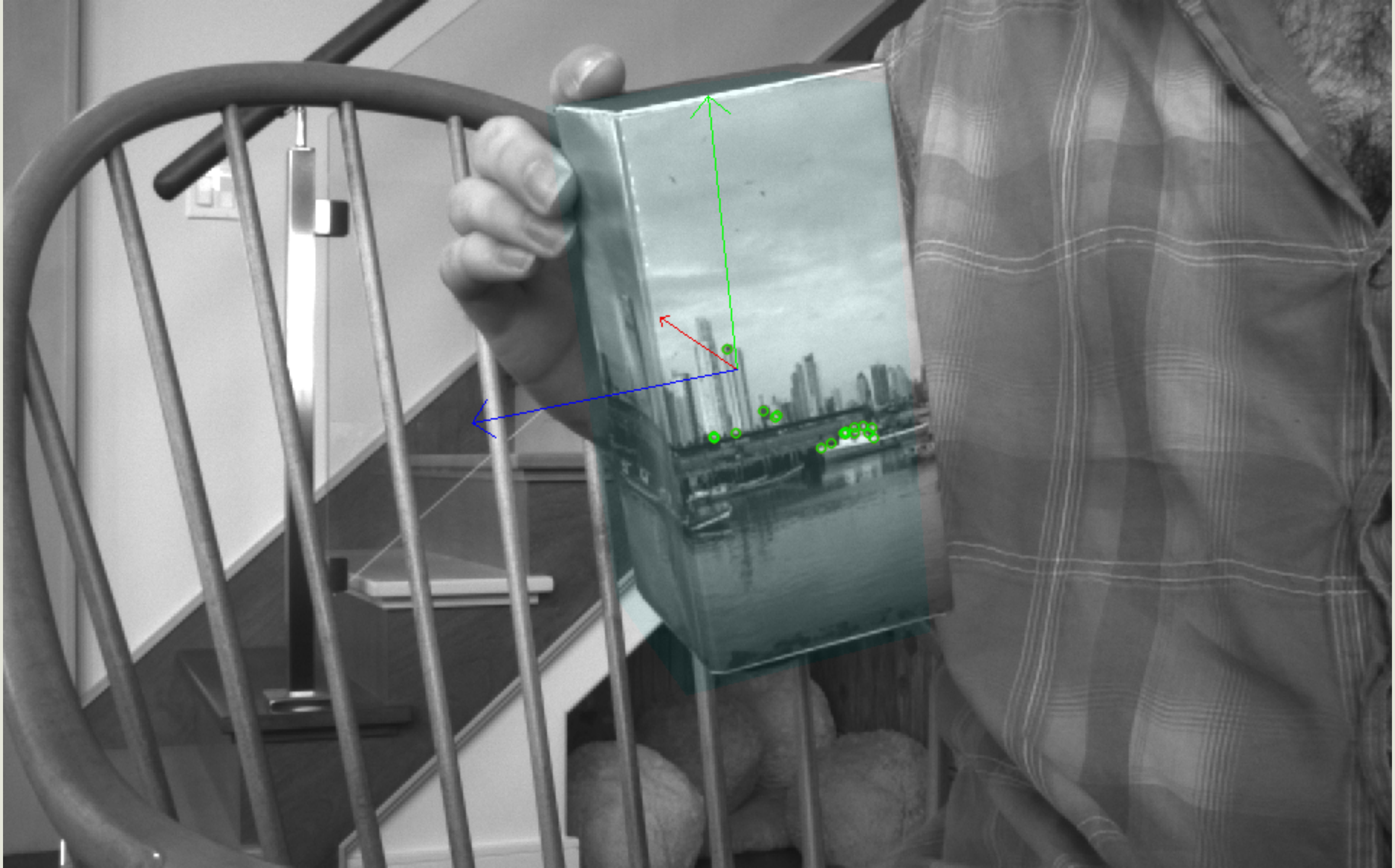
- 1920x1200 resolution, 163 FPS, 1/1.2” sensor format, C-mount interchangeable lenses
  - Works well with cheap, classic lenses from 16mm “cine” systems
- Sold by FLIR / Point Grey: https://www.ptgrey.com/grasshopper3-23-mp-mono-usb3-vision-sony-pregius-imx174-camera

Subjective evaluation of Grasshopper3 GS3-U3-23S6M-C

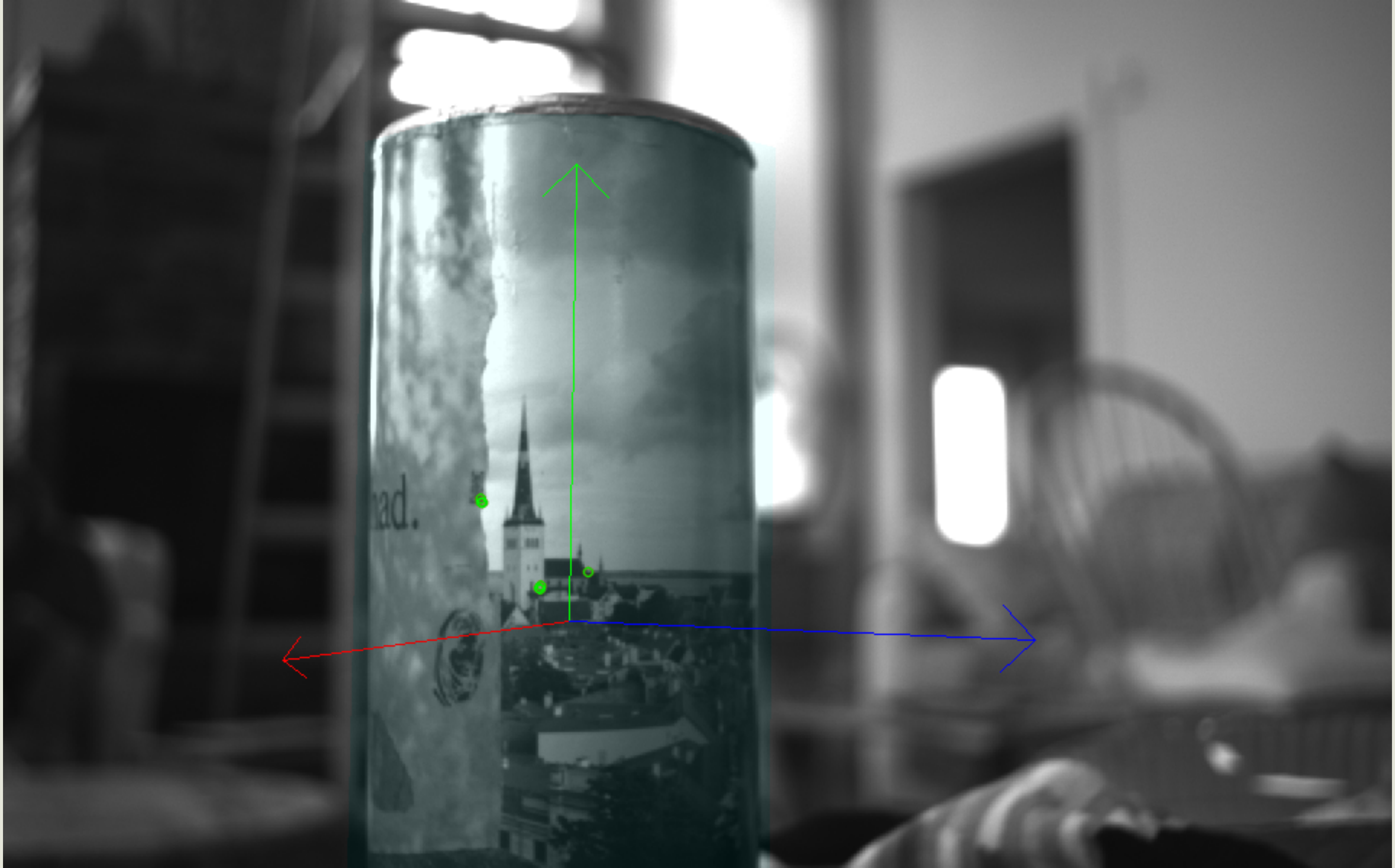
- 3D tracking works well, feels fluid
  - ~40 FPS, compared to ~25 FPS with built-in webcam (MacBook Pro, Late 2013)
  - As distance increases, captures more inliers than built-in webcam
- Requires good manual focusing
- High fidelity
  - Detailed when in focus
  - Good contrast
  - Low noise, even in dim light








nad.



Future Work

My Wish List

- Improve cylindrical tracking
  - Apply distortion to reference image before detecting keypoints?
- Use multiple reference images for different viewpoints
- Optimize **grayscale contrast*** (tonality) to produce more inliers
- Increase **frame rate*** to make Kalman filter run more smoothly
- Create “invisible” textures that only UV or IR camera can see
- Port to other platforms, including mobiles
  - Integrate visual tracking & rendering with other kinds of sensing

* Contrast and frame rate are affected by lighting, materials, lens, camera, processing

Your Wish List?

- Come to tomorrow’s workshop to learn and give your input
- What features and architecture would you like to see in an open-source library?
- What chapters would you like to see in a book?