Course Overview

ECE 5960 / 6960 Computational Photography

Rajesh Menon, 01/11/22

Project-based course

Lots of independent work with teams.
Build hardware (camera).
Build software.

Learn by doing.

Logistics

- Where: Zoom & *in lab, MEB 1541 (select dates)*.
- When: Tue/Thur 12:30-1:30pm MT
- Zoom: invitations sent via canvas.
- Office hours: Wednesdays 7-8pm MT (but flexible, by appt.).
- Slack: all discussions, ask questions, collaborations, Invitations to be sent via email. *[quick demo]*
- All class notes, software, online resources to be provided. No textbook required.

Learning objectives

- Understand fundamentals of computational imaging.
- Innovate to create new types of cameras (team projects).
- Write a technical journal article & submit for peer review.
- Present your technical article.
- Create open-source software & submit to GitHub.

Topics of interest

- Design of imaging systems.
- Design of computational post processing (linear algebra, machine learning, etc.)
- Human-centric images - how does computation help?
- Machine-centric images - how does computation help?
- What are the fundamental limits?
- Imaging in the visible and infra-red bands.
- Imaging from space.
- Biomedical imaging.
- Imaging for autonomous driving.
- Imaging for biometrics
- … Your interests…

Grading

- Assignments: 30%
- Team presentations: 35%
- Final technical paper (one per team & includes GitHub submission): 35%

Course website: (link)

- All announcements will be posted here. These may not be mirrored on canvas, so please bookmark this.
- Most lectures will be recorded and posted here.
- All course material will be posted here (online content & software).
- Review schedule (tentative, small changes might happen).

Assignments

1. [01/18] Choose team & topic
2. [01/25] Select & order image sensors, optics, relevant hardware.
3. [02/03] Complete simulation model of your system. Submit report.
4. [02/17] Submit 1st version of software tools to GitHub.
5. [03/03] Present 1st experimental results from your camera.
6. [03/17] Complete 1st draft of your technical paper.
7. [03/31] Submit 2nd version of software tools (include ML data as appropriate) to GitHub.
8. [04/14] Submit technical paper for peer review & submit final software (and all data) to Github.
9. [04/21] Final presentation of results.

Journals for peer review

- At least at the quality of IEEE journals.
- Examples are:
  - Optics Letters
  - Optics Express
  - APL Photonics, etc.
  - All of these have overleaf (Latex) & word templates. Please use them.

Open source software & data availability

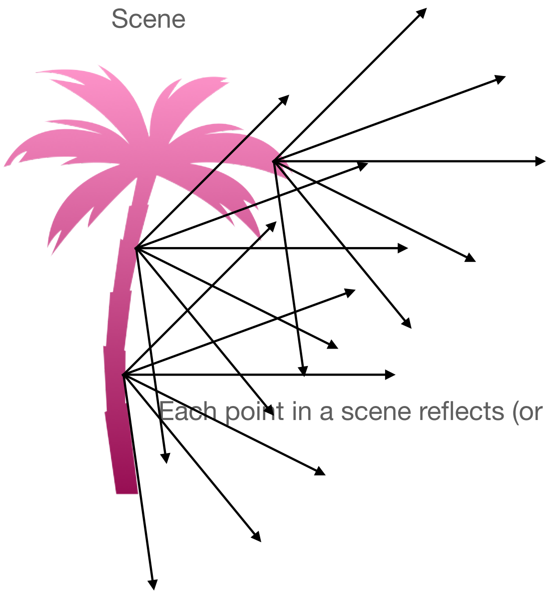
- All data you collect should be uploaded for free availability on GitHub.
- All software should be made open source & freely available on Github.
- Exceptions possible if you want to patent or restrict usage. *Let me know.*

Introductions & interests

- What do you want to get out of this course?
- What are your skill sets ?
- Feel free to discuss any passion projects (if you have any).

Why does a camera need optics?

Scene

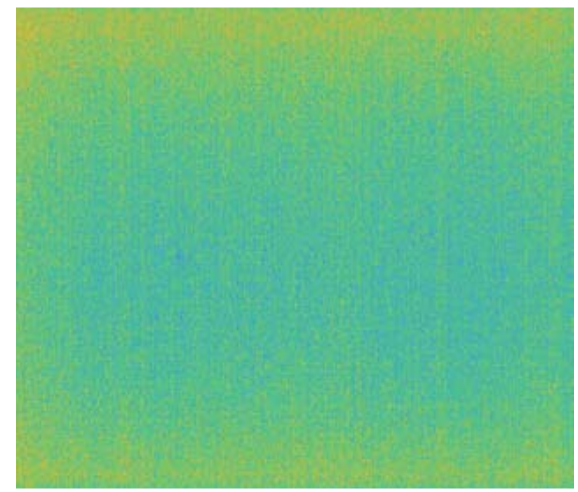


Each point in a scene reflects (or emits) light

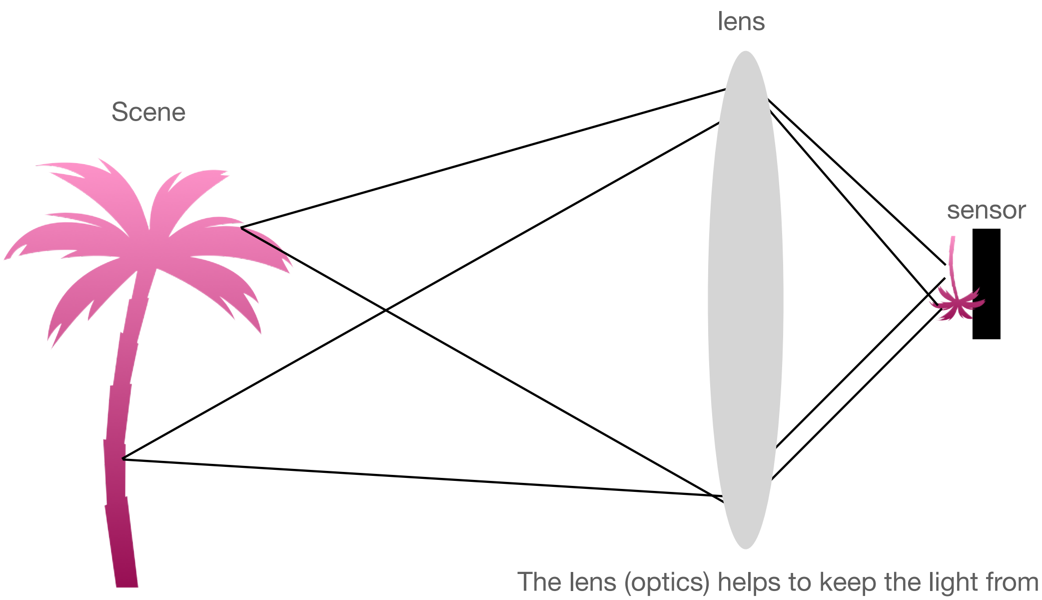
sensor

If we have no optics & just record light on a sensor, light from all points get mixed together & no image is formed!

Example of output from an image sensor with no optics



Why does a camera need optics?

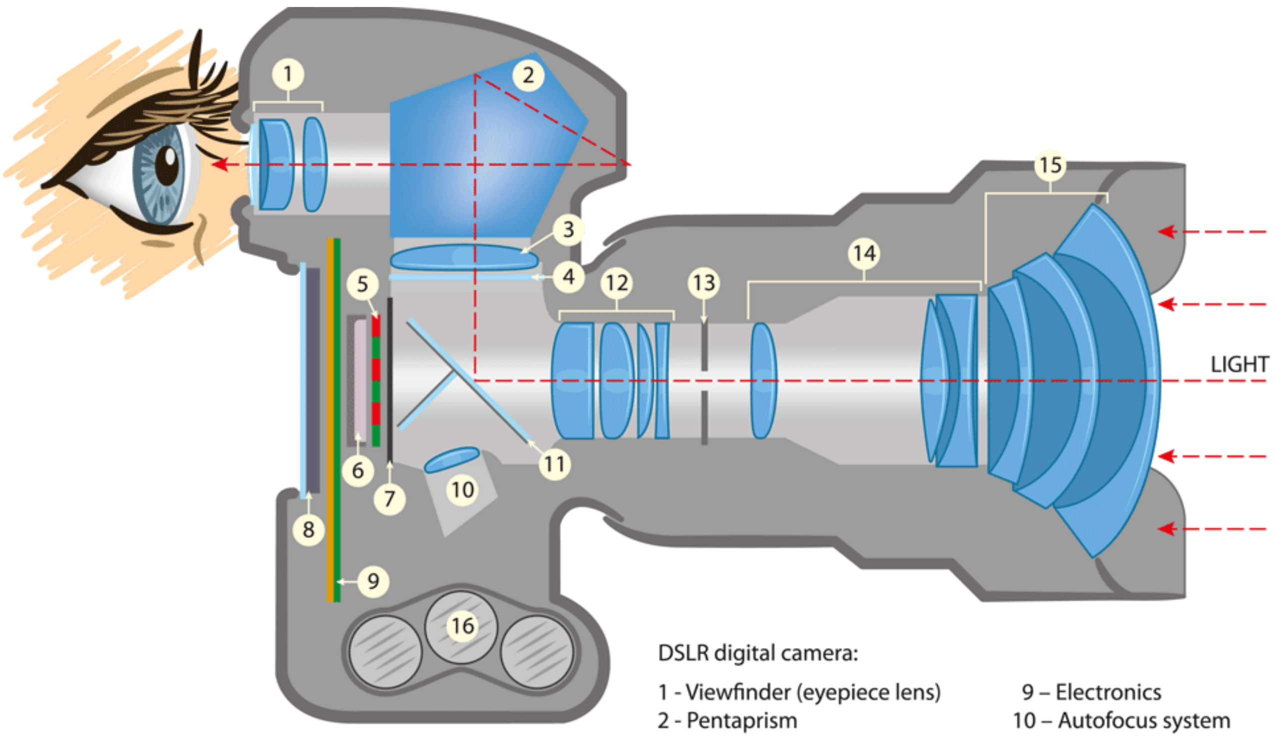


The lens (optics) helps to keep the light from different points separated. Organizes light so as to form an image.

Image is a 1:1 map of the object with (de) magnification & sometimes lateral inversion.

We will learn how to model this process in a computer for relatively complex camera modules.

The optics can be complicated.



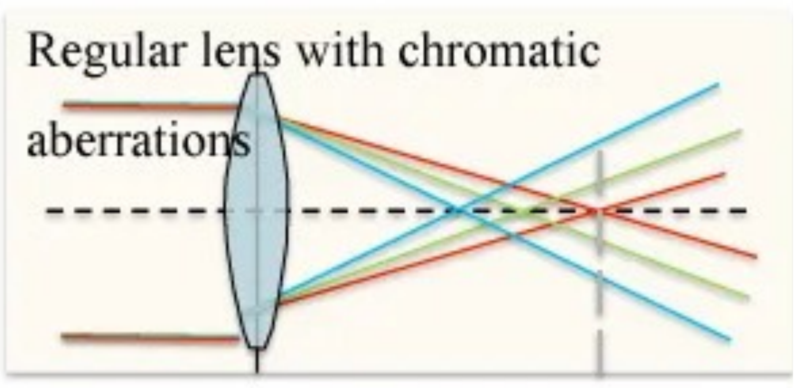
DSLR digital camera:

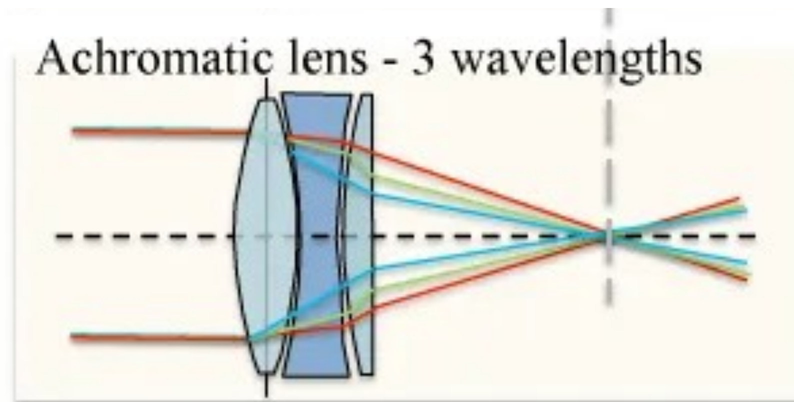
1 - Viewfinder (eyepiece lens)
2 - Pentaprism
3 - Focusing screen
4 - Condenser lens
5 - Color and infrared filter
6 - Digital sensor
7 - Shutter
8 – Display
9 – Electronics
10 – Autofocus system
11 – Reflex and relay mirror
12 – Focusing elements
13 – Aperture
14 – Zoom elements
15 – Front light gathering elements
16 – Batteries

The optics can be complicated.

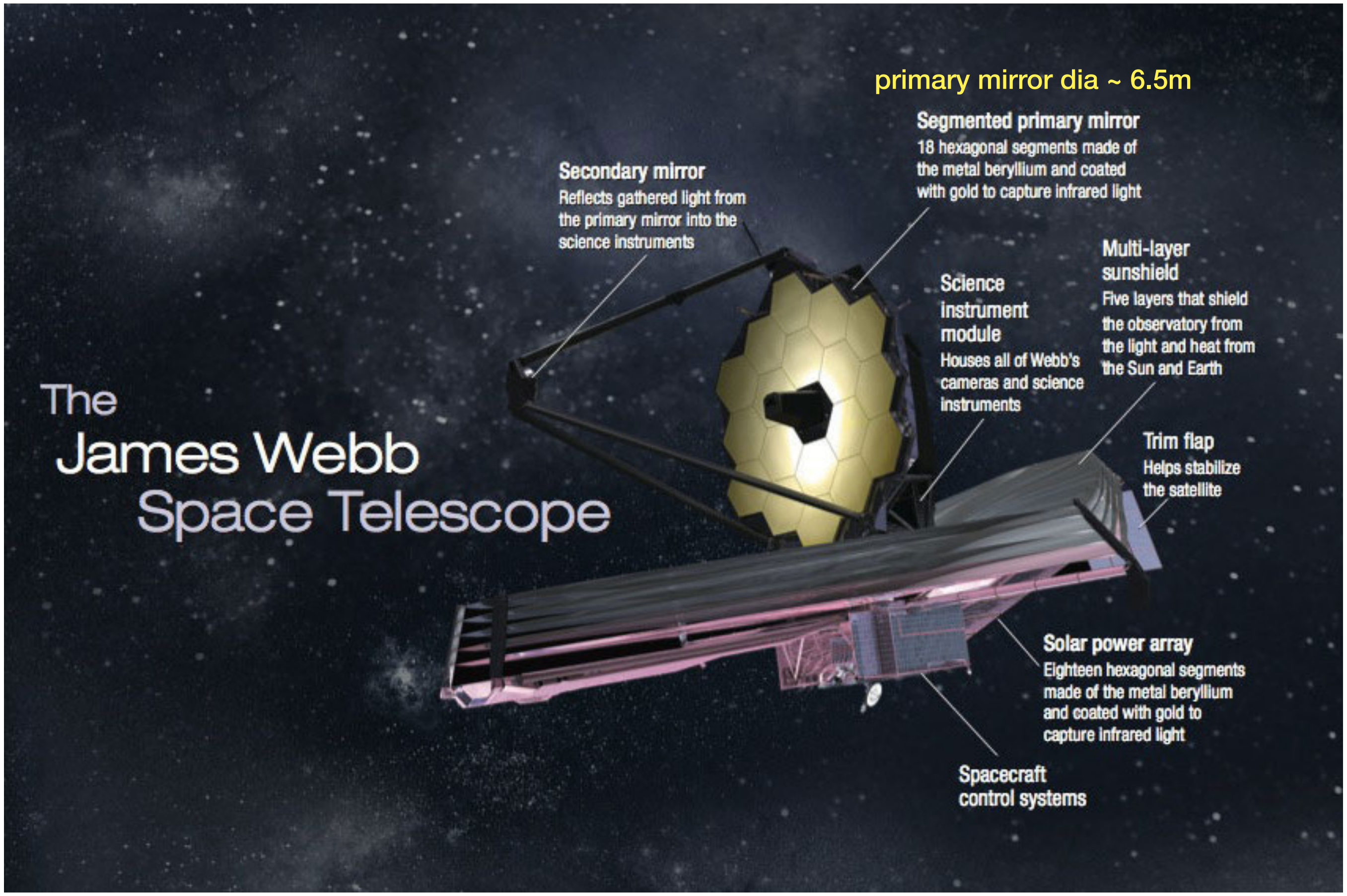
Why do you need many lenses?

For example for correcting aberrations.



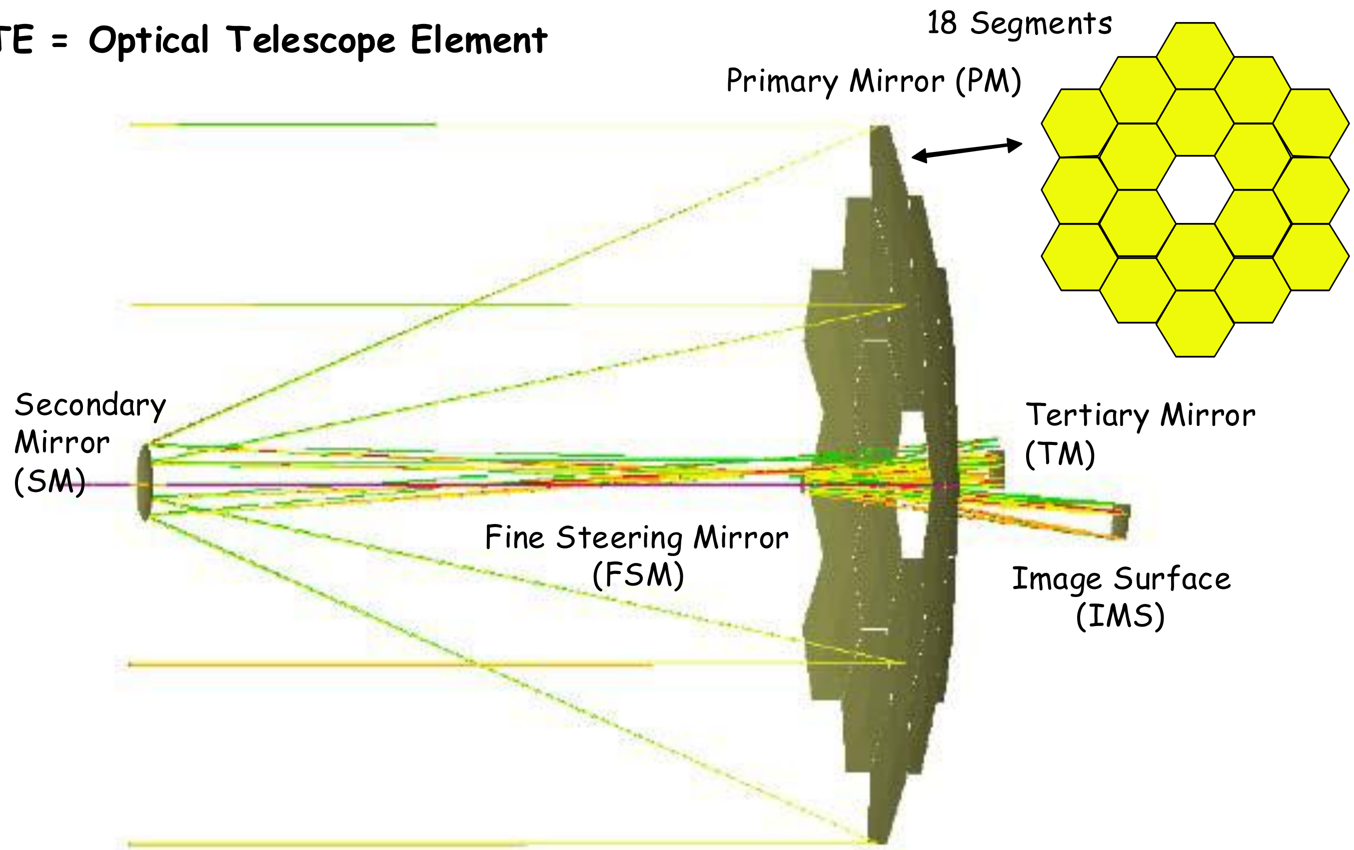


The optics can be in reflection.



The optics can be in reflection.

OTE = Optical Telescope Element



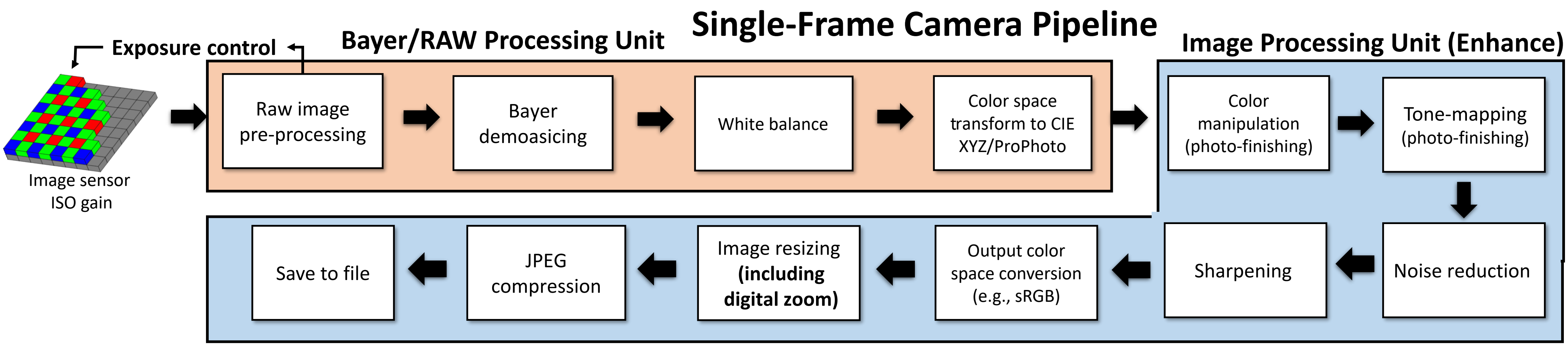
Three-Mirror-Anastigmat (TMA) design per D. Korsch

Resources

- OpenCV - free course to get started. Open source software that you can modify.
- Arducam or similar cameras.
- Online lectures & simulation tools.
- Introduction to Machine Learning on Coursera (Andrew Ng) (link)

What happens after the image is recorded?

https://arxiv.org/abs/2102.09000



Correcting Vignetting

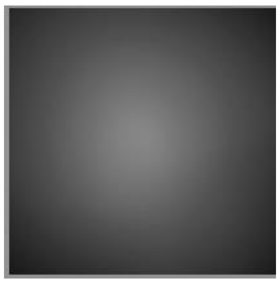


Image of a uniformly illuminated surface. The light falling on the sensor is reduced in a radial pattern.

Bayer image before lens shading correction.

Lens shading mask required to correct the radial fall-off.

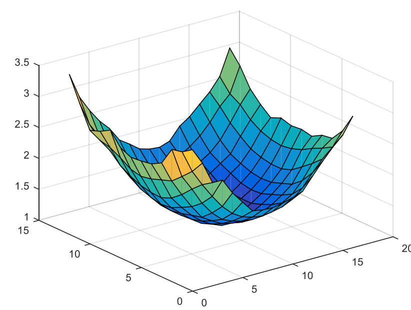
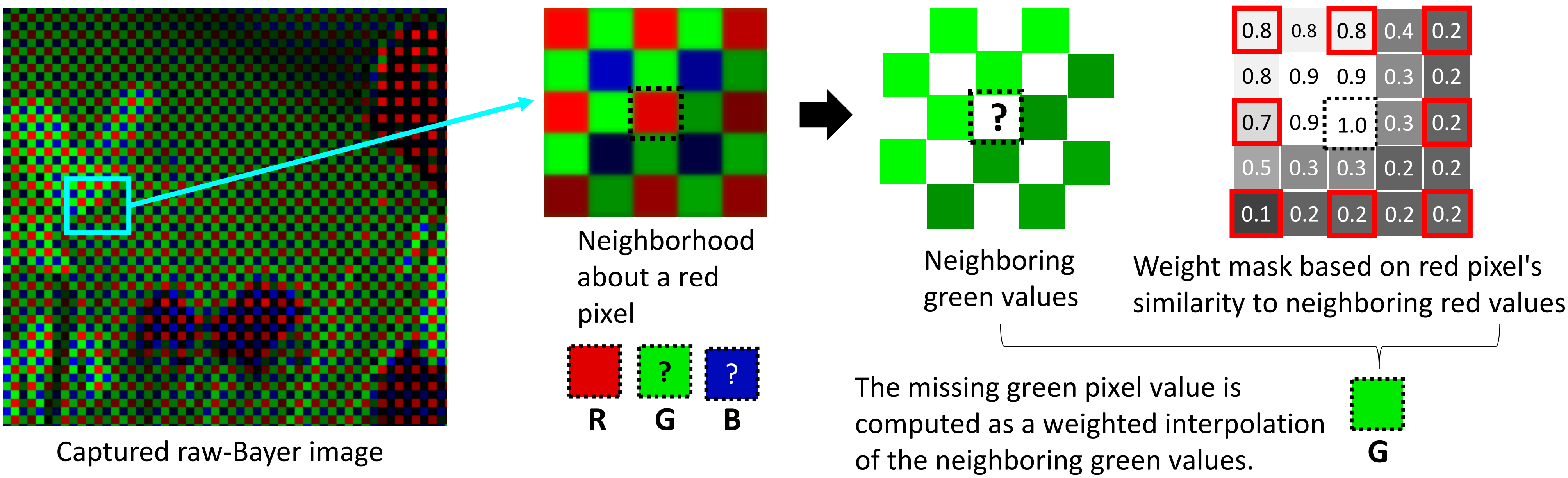


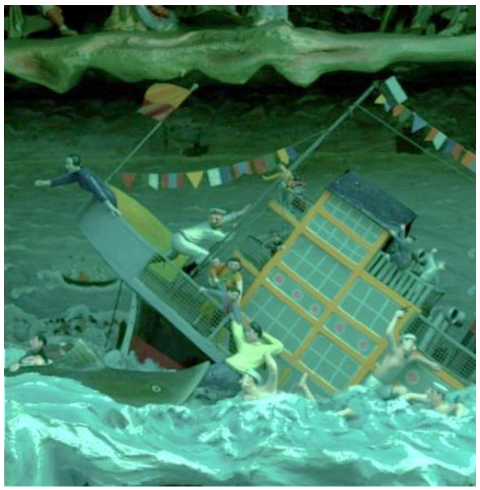
Image of a uniformly illuminated surface after lens shading correction has been applied.

Bayer image after lens shading correction.

Demosaicing



White Balancing



raw sensor image before white-balance correction

Sensor's RGB response to scene illumination (ℓ)

$$\begin{bmatrix} \ell_r \\ \ell_g \\ \ell_b \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.8 \\ 0.8 \end{bmatrix}$$

White-balance correction matrix

$$\begin{bmatrix} r_{wb} \\ g_{wb} \\ b_{wb} \end{bmatrix} = \begin{bmatrix} 1/\ell_r & 0 & 0 \\ 0 & 1/\ell_g & 0 \\ 0 & 0 & 1/\ell_b \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

Auto white balance (AWB) algorithm estimates the illumination from the input image.

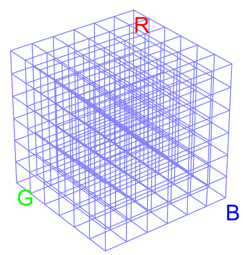
raw sensor image after white-balance correction

Photo finishing

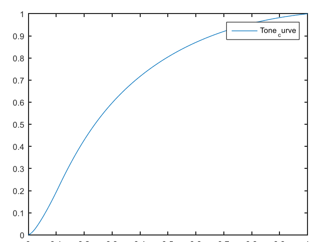
Different photo-finishing picture styles

Photo-Finishing

Before photo-finishing



Color manipulation
3D LUT



Tone manipulation
as a 1D LUT

After photo-finishing

Many other topics …

- Tone-mapping, sharpening, Denoising,
- Low-light imaging, super-resolution, Bokeh
- All of these are for humans, what should we do for machines (cars, robots, drones, etc.) ?