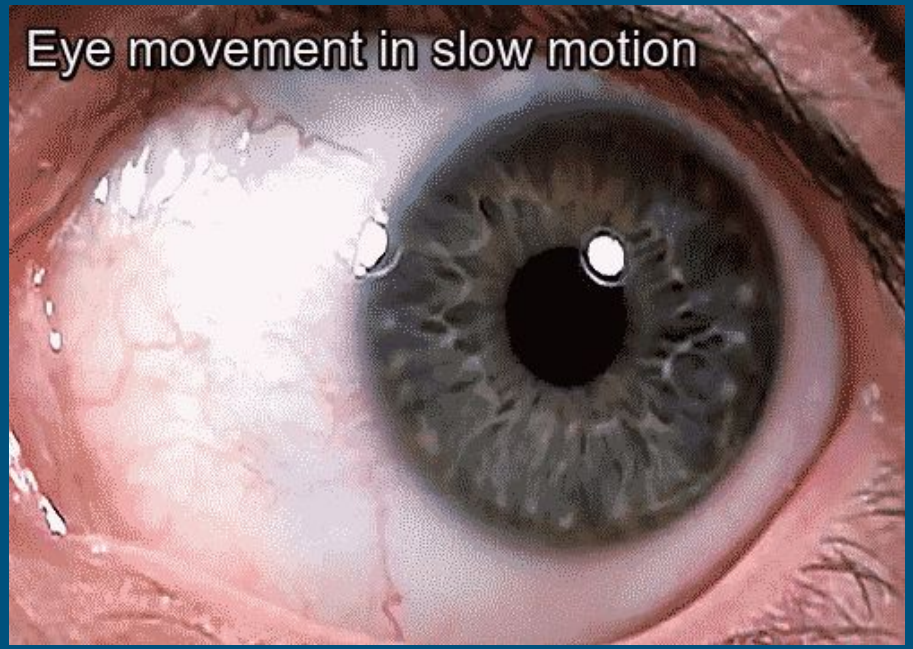
Depth Estimation Error Calculation from Baselines of a Single Event Camera

Huidong Hwang and Haoman Zhong

Or “How accurately can we estimate depth with a single event camera when moving it side to side?”

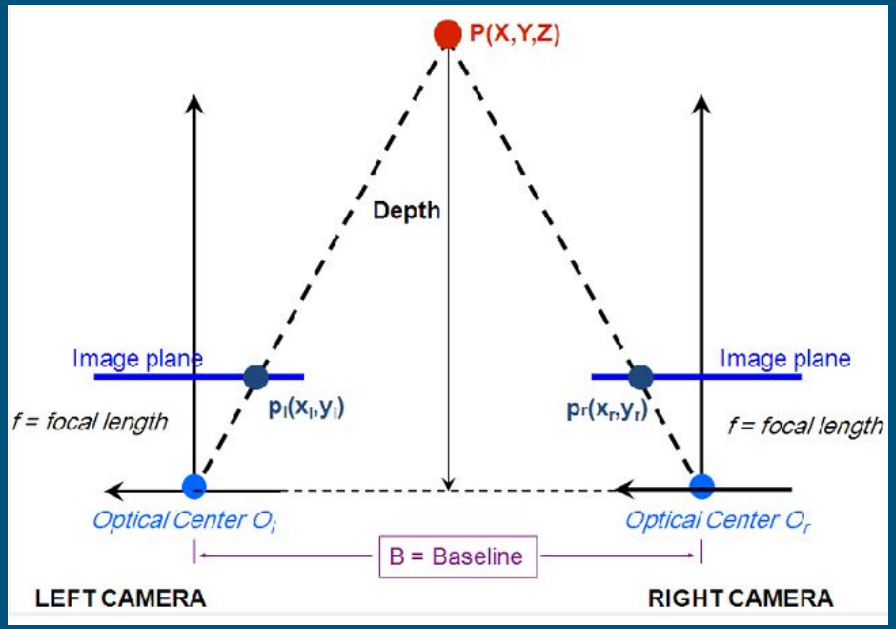
Huidong Hwang and Haoman Zhong

Initial Idea



- We were inspired by the microsaccades of human eyes which happen when we focus
  - We thought the microsaccades might help with depth estimation when seeing from a single eye
- We combined the concept of exaggerating microsaccades with the baseline of stereo vision to estimate depth

Look back to Initial Idea



Experiments

Input Image

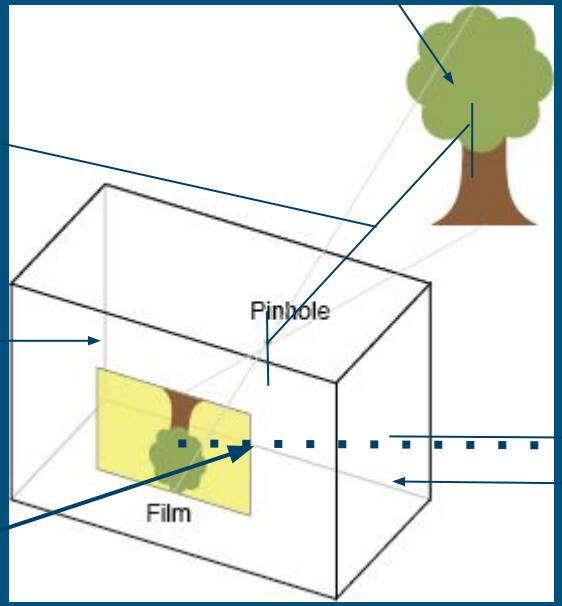
For our camera,
a poster

Ground truth
distance Z,
set by us.

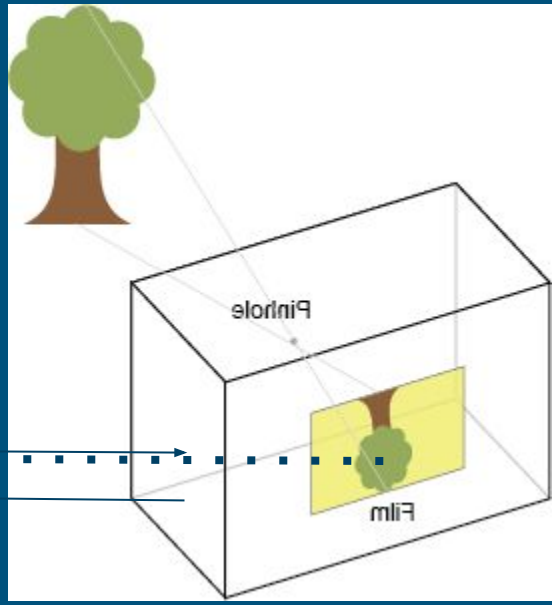
Camera does not
know it

Simulated
Camera

Baseline



Motion,
non linear,
sinusoidal
speed

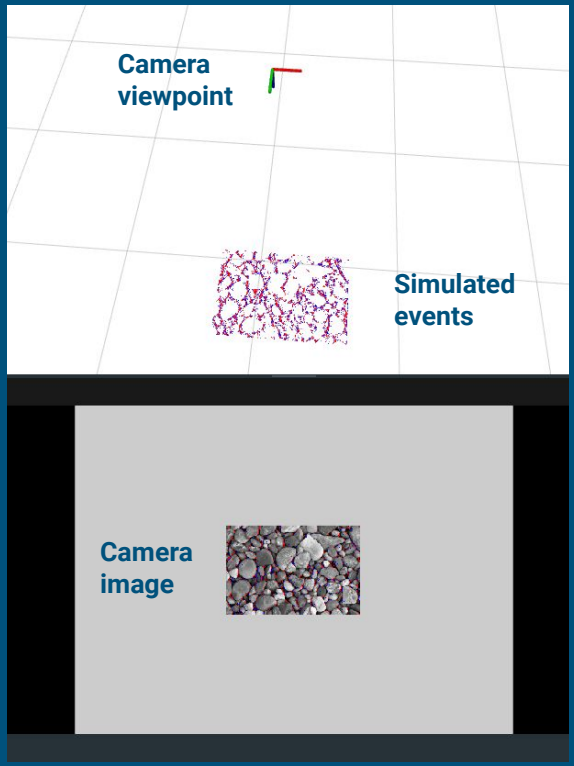


Creating the Dataset

Raw input image: rocks.jpg

ESIM simulation

Custom configuration with two different depths



Screenshot mid-simulation

Calculating the error

$$\Delta z = \frac{z^2}{b \cdot f} \cdot \Delta D \quad \text{Where}$$

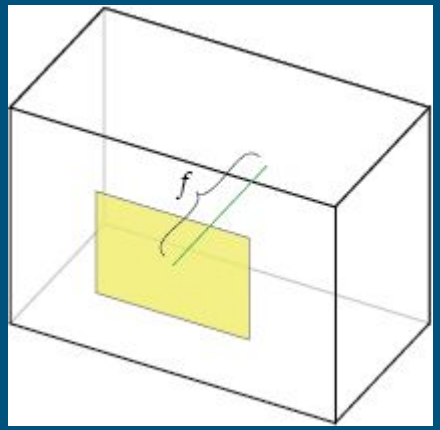
Δz	depth error
z	depth of point of interest
b	baseline
f	focal length
ΔD	disparity error

According to some papers the typical error for ▲D is either 0.5 or 1.0.

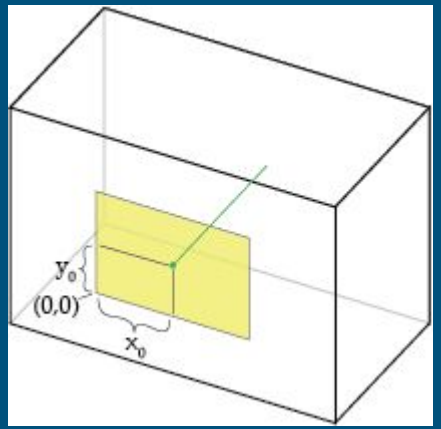
Units for each variable:
▲ Z in meter
Z in meter
b in meter
f in pixel
▲D in pixel

Reference: https://doc.rc-visard.com/v1.3/en/stereo_matching.html

Other variables in the equation



Focal length in Pixel

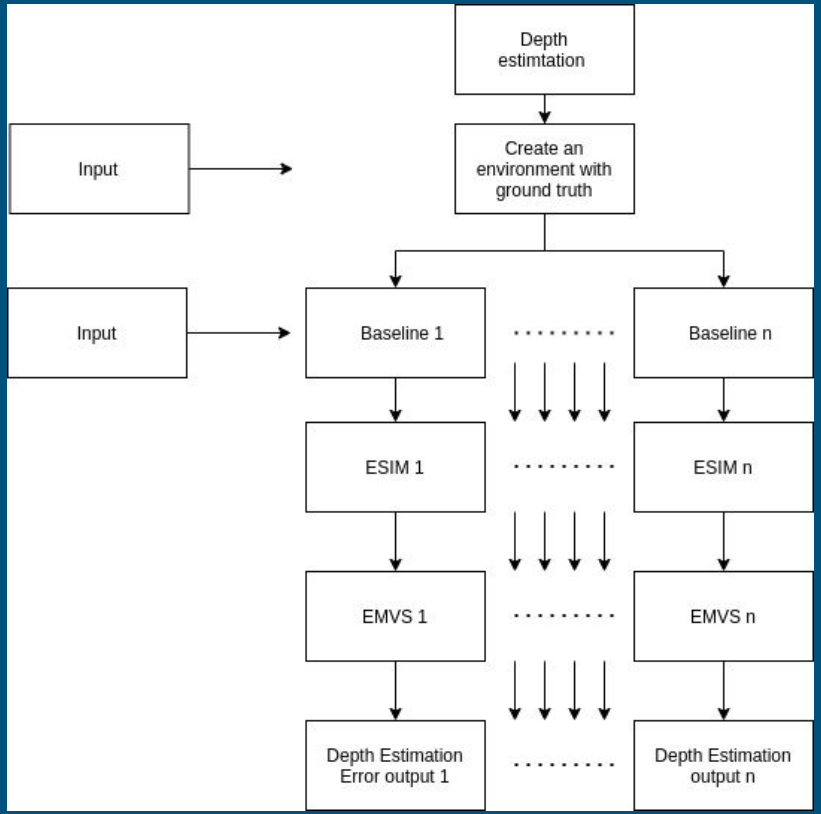


Central point of the image

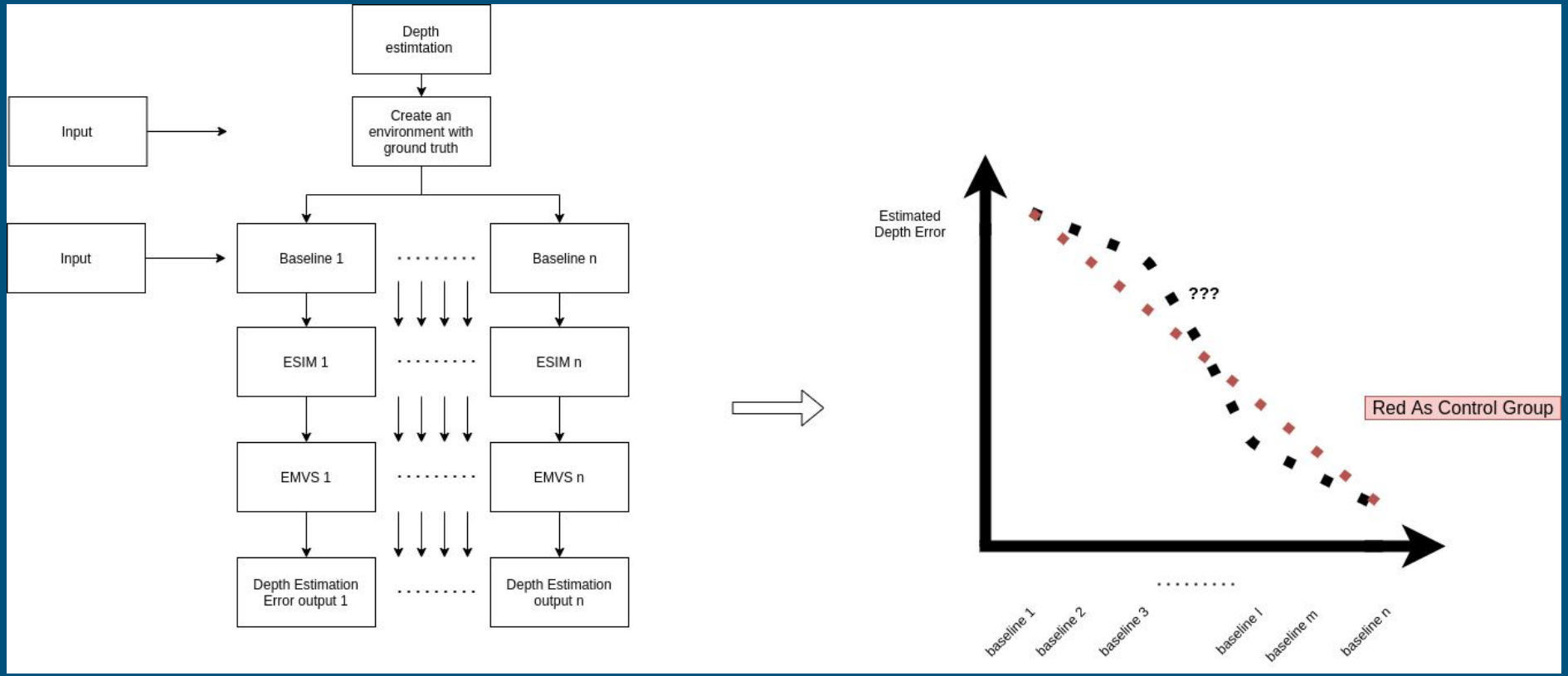
Blockchart


Depth
estimtation
Input
Create an
environment with
ground truth

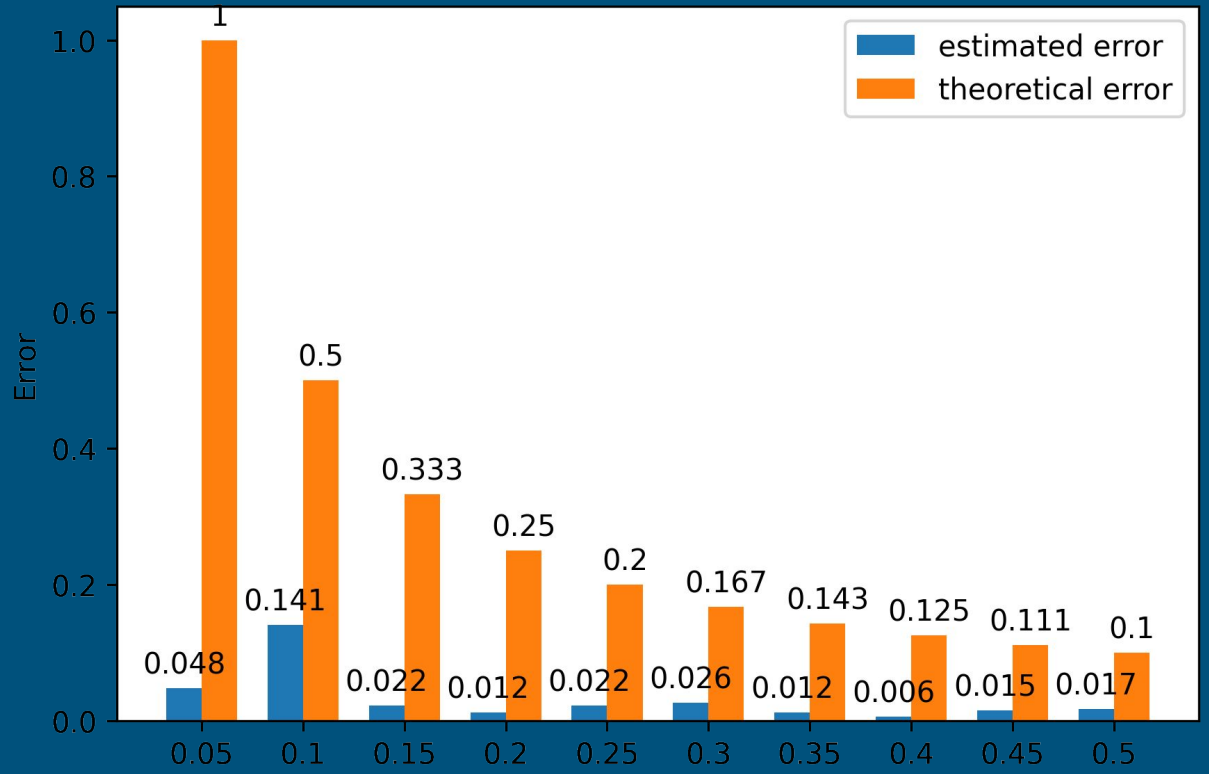
Blockchart


Depth estimtation
Input
Create an environment with ground truth
Input
Baseline 1
Baseline n
ESIM 1
ESIM n
EMVS 1
EMVS n
Depth Estimation Error output 1
Depth Estimation output n

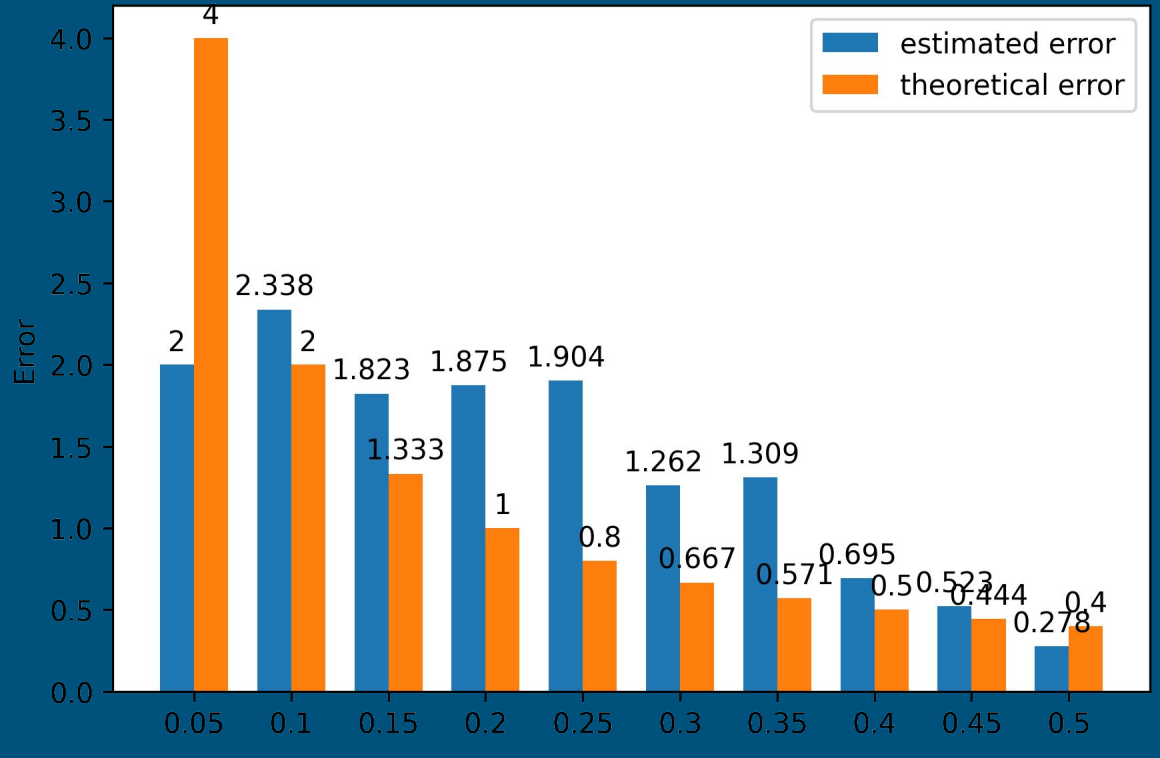
Blockchart


Depth estimtation
Input
Create an environment with ground truth
Input
Baseline 1
Baseline n
ESIM 1
ESIM n
EMVS 1
EMVS n
Depth Estimation Error output 1
Depth Estimation output n
Estimated Depth Error
???
Red As Control Group
baseline 1
baseline 2
baseline 3
baseline l
baseline m
baseline n

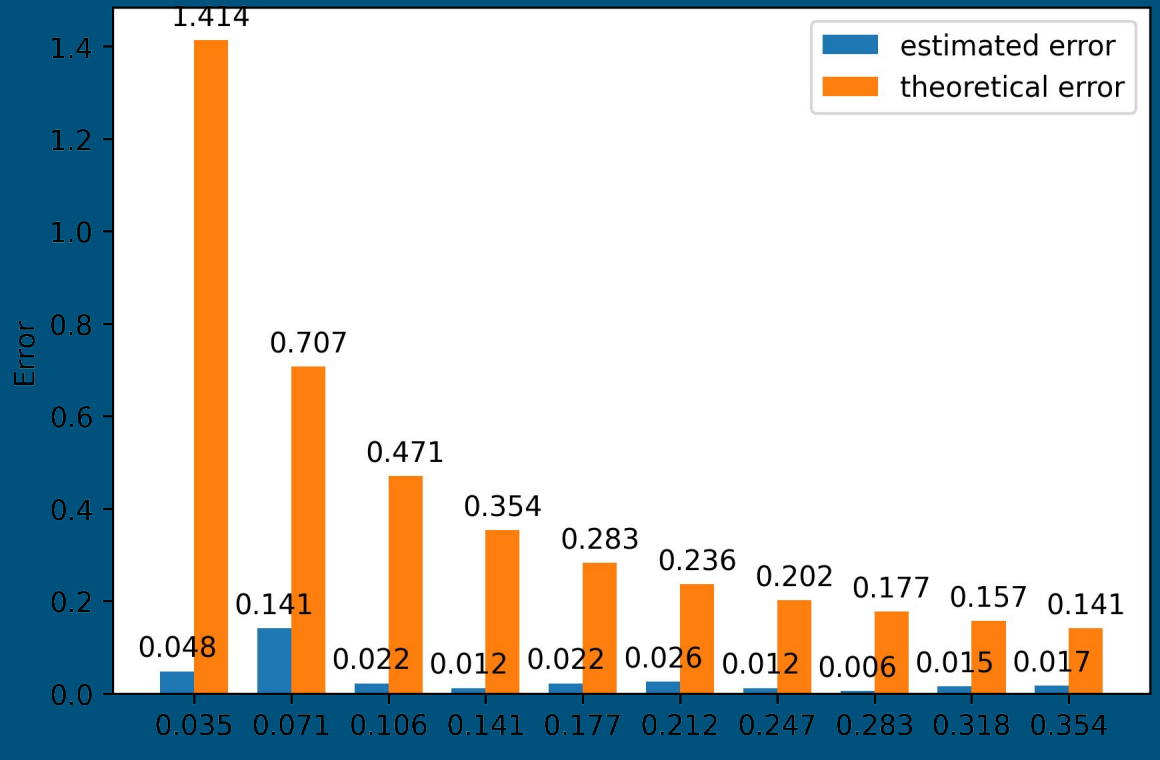
On 5 meter distance Z=5m, Delta D=1px



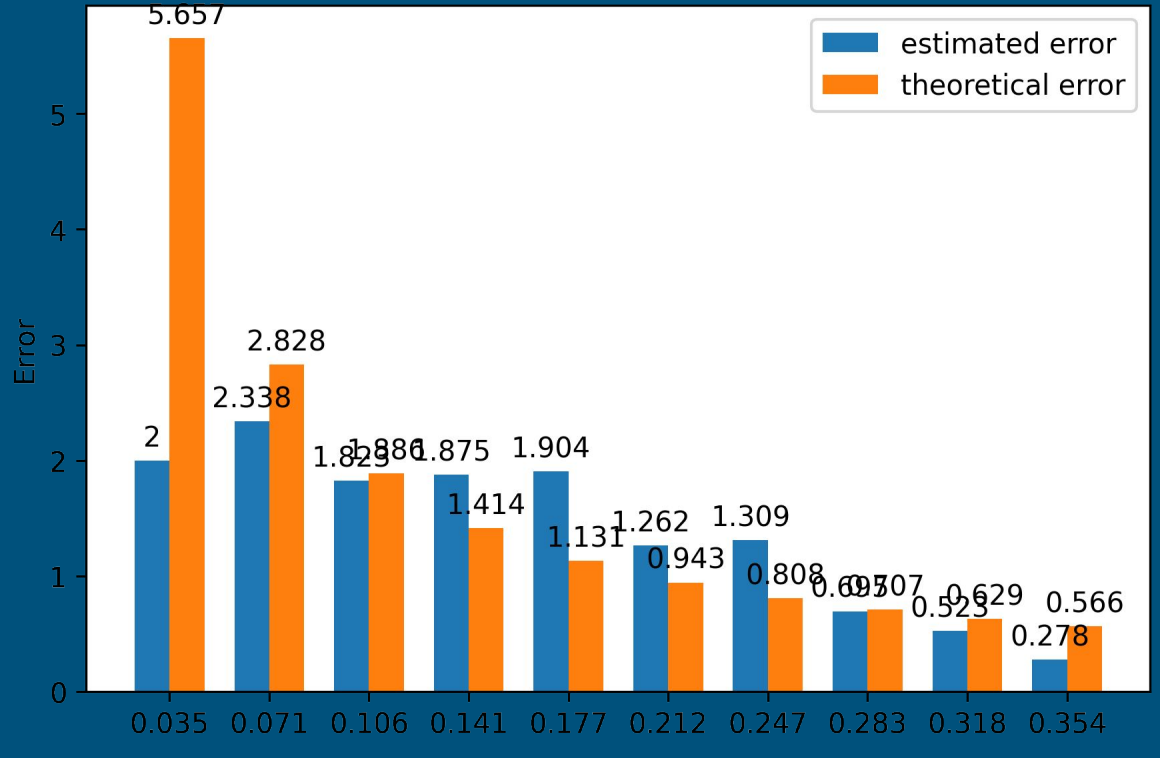
On 10 meter distance Z=10m, Delta D=1px



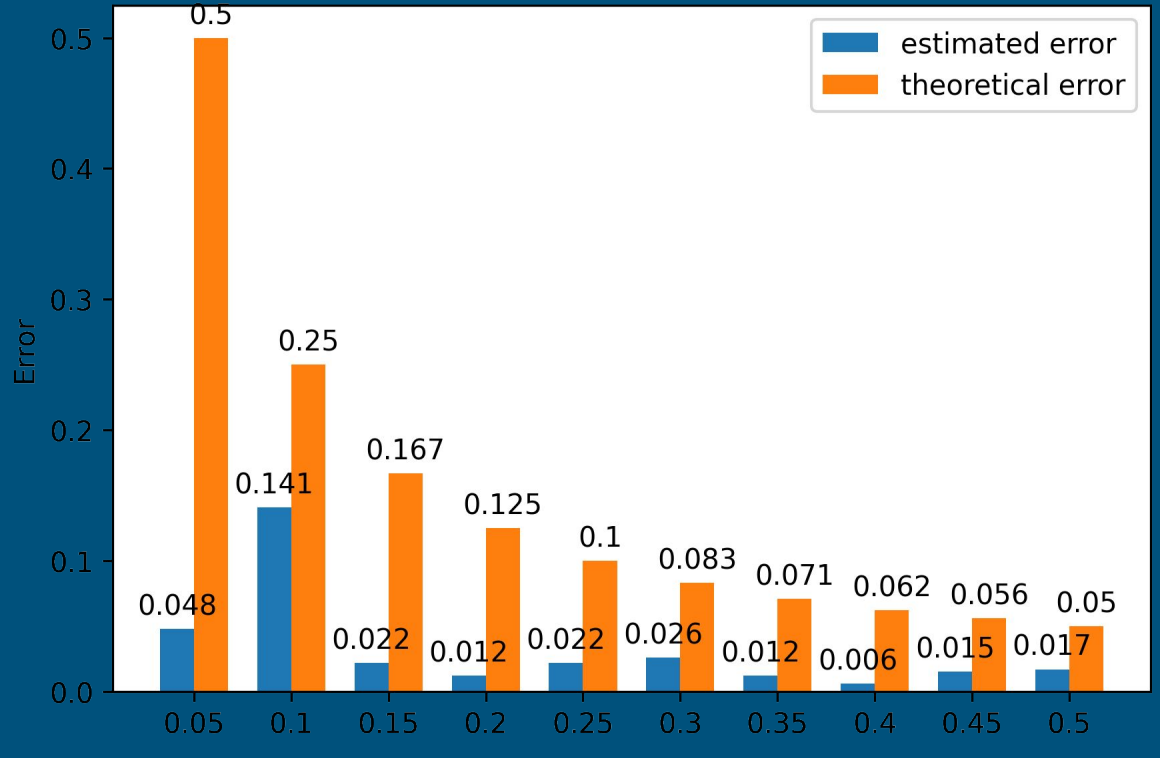
On 5 meter distance.
Z=5m, Delta D=1px, effective baseline = baseline/sqrt(2)



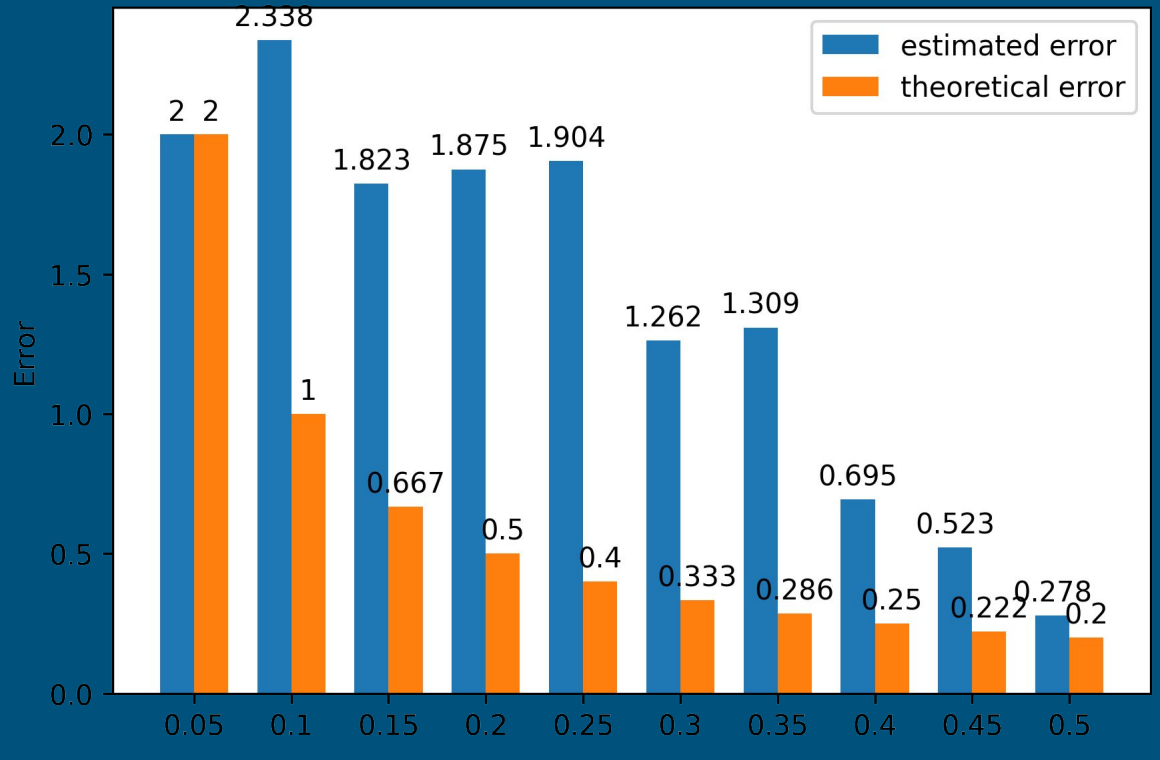
On 10 meter distance
Z=10m, Delta D=1px, effective baseline= baseline/sqrt(2)



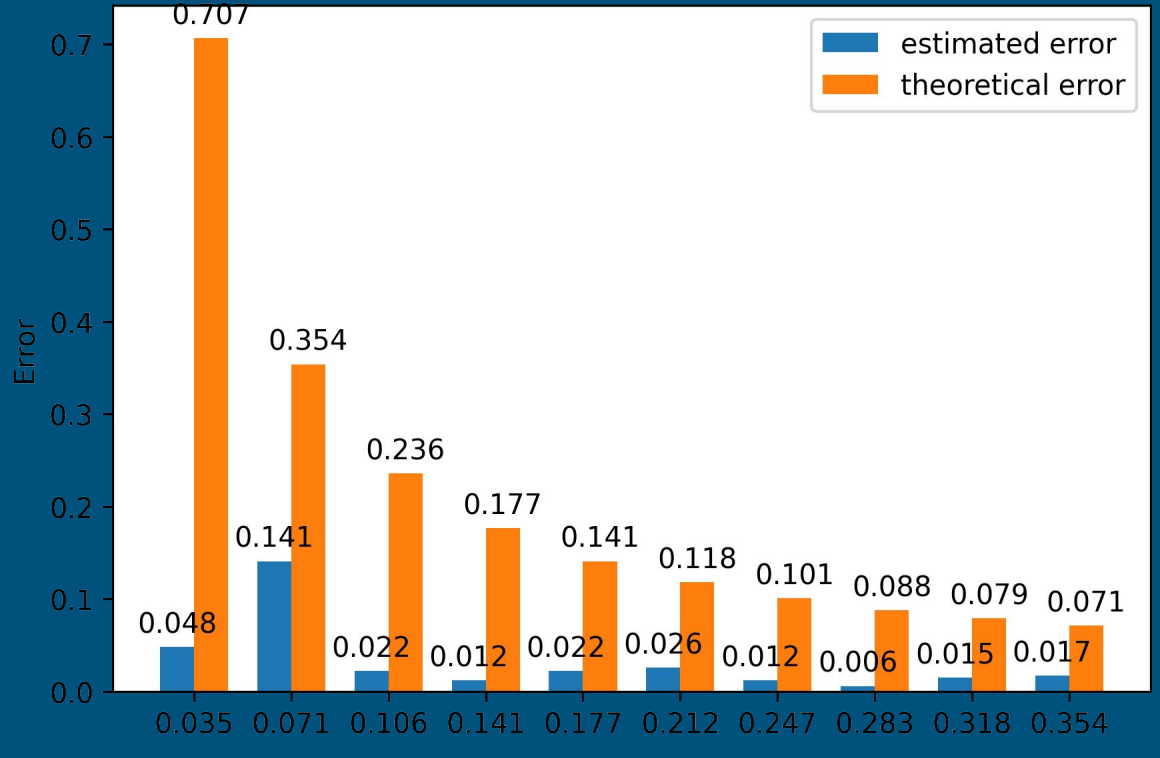
On 5 meter distance Z=5m, Delta D=0.5px



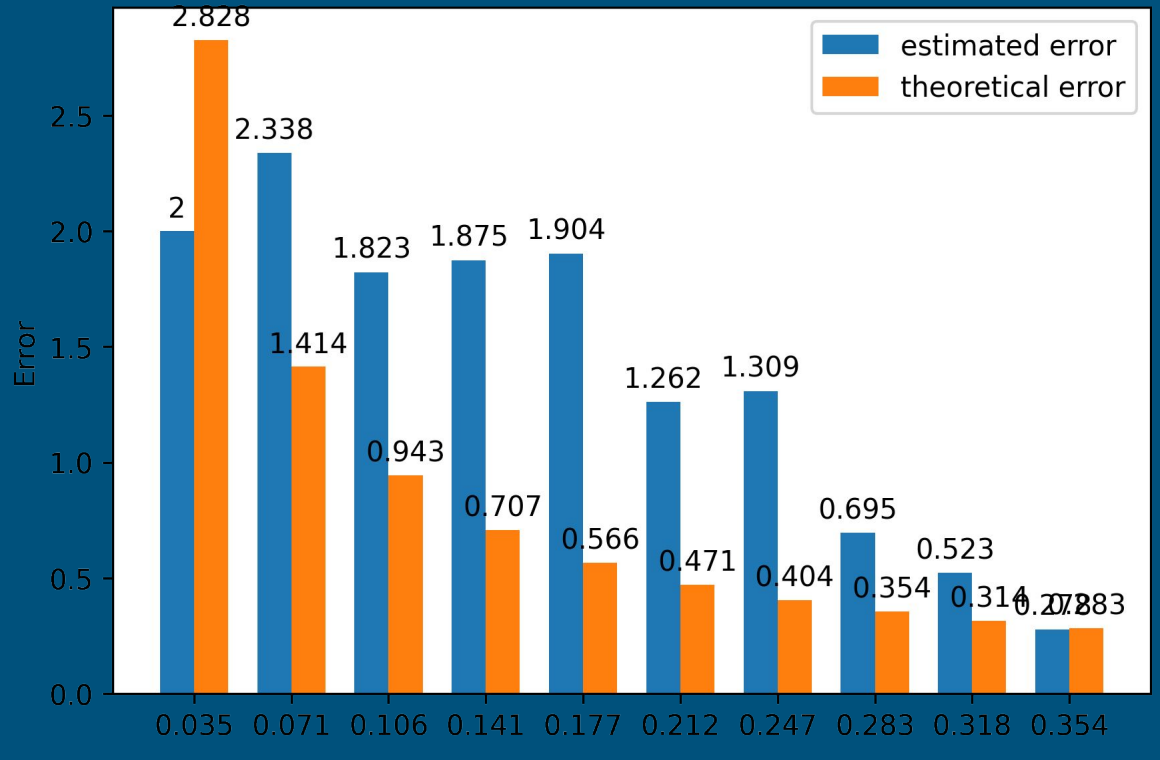
On 10 meter distance Z=10m, Delta D=0.5px



On 5 meter distance.
Z=5m, Delta D=0.5px, effective baseline = baseline/sqrt(2)



On 10 meter distance
Z=10m, Delta D=0.5px, effective baseline= baseline/sqrt(2)



Conclusions

Difficulties During Experimentation

- Learning two separate codebases and their configuration files
- Realizing that the renderer only does 2D camera work
- Modifying code to fit our purposes
- Creating a smooth pipeline to speed up experimentation
- Figuring out adequate testing parameters

Outlooks

The gap between 5 and 10 meters needs to be examined. E.g. more experiments on 6/7/8/9 meters.

Different experiments on different images (posters) to eliminate other factors.

More complex scenes with 3D rendering for more interesting data

Questions and Comments?

Thank you for your attention