Multi-View Stereo

CS4670/CS5670 - Kevin Matzen - April 13, 2016

Video credit: Furukawa, et. al. Towards Internet-scale multi-view stereo, CVPR 2010

Roadmap

- What we've seen so far
  - Single view modeling (1 camera)
  - Stereo modeling (2 cameras)

Roadmap

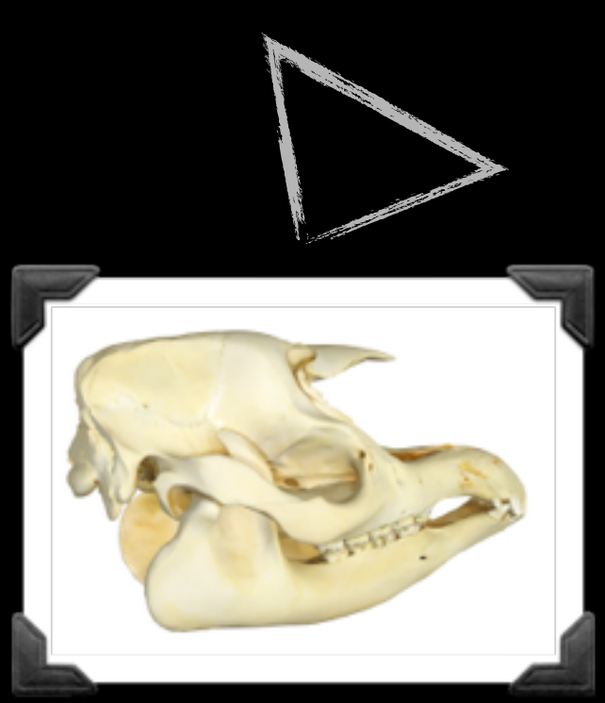
- What we've seen so far
  - Single view modeling (1 camera)
  - Stereo modeling (2 cameras)
- So what's next?
  - More cameras!

Roadmap

- What we've seen so far
  - Single view modeling (1 camera)
  - Stereo modeling (2 cameras)
- So what's next?
  - More cameras! - Which is hard.

Stereo Basics

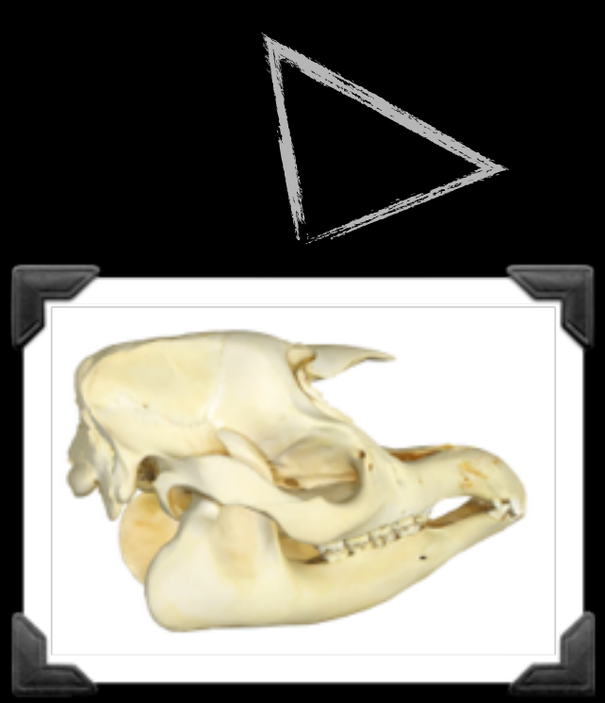
Stereo Basics



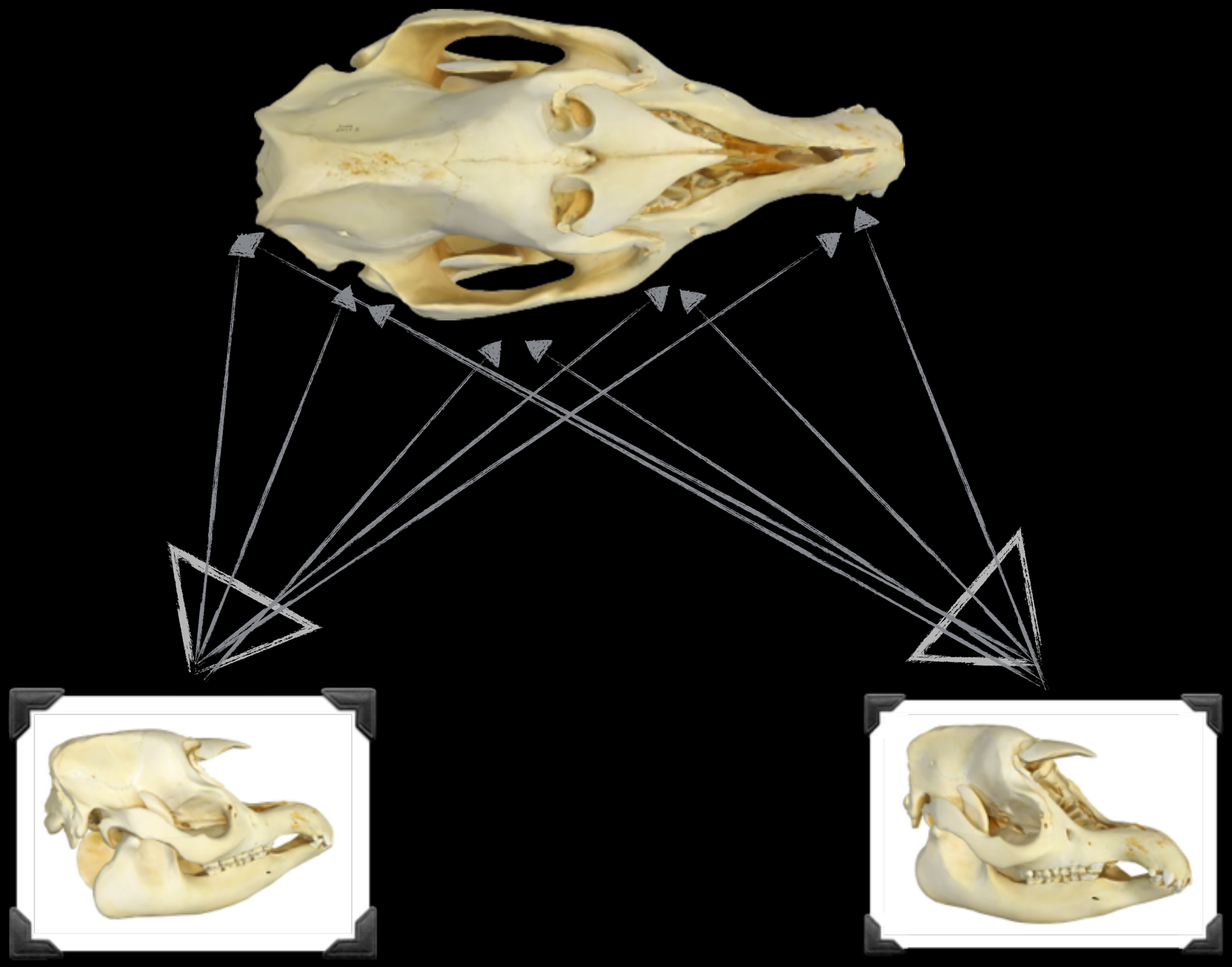
Stereo Basics

For today's lecture, assume we always know the camera calibration.

Stereo Basics



Stereo Basics

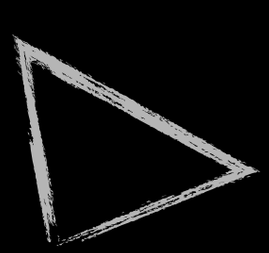


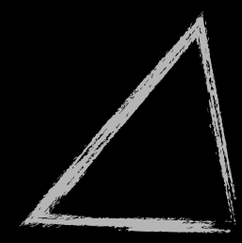
Stereo Basics

Output is a depth map

Stereo Basics

How do we scale this to more cameras?
What more can we do with more cameras?





Today's outline

- What can multi-view stereo do for you?
- How can we extend our two-view depth map recovery algorithm to many views?
- What other representations are appropriate for MVS beyond depth maps?

Recommended Reading

Multi-View Stereo: A Tutorial

Furukawa and Hernandez, 2015

http://www.cse.wustl.edu/~furukawa/papers/fnt_mvs.pdf

Applications


Data SIO, NOAA, U.S. Navy, NGA, GEBCO
Google earth

AUTODESK® 123D® APPS 3D MODELS PROJECTS TOOLS BLOG

でんじろんさん

by じろう でん 266,

POWERED BY matterport

1BR, 1BA Penthouse

Terms

Dollhouse

Floor plan

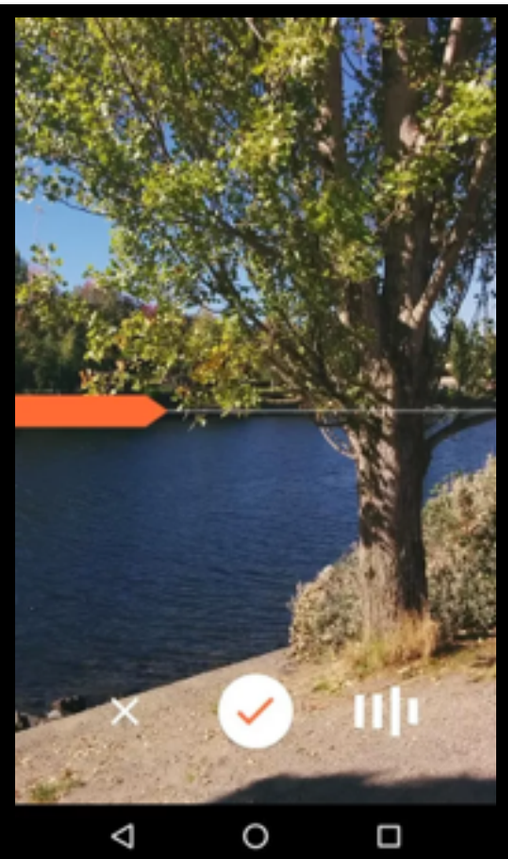
Whistle in the Form of Female Figure *600 AD - 900 AD*

Details Los Angeles County Museum of Art

Los Angeles County Museum of Art | Sculpture | Mexico

Google

JUMP




c|net

http://bit.ly/surround360

How does MVS work?

Recap: Stereo

1

2

Recap: Stereo

1

2

1. Rectify images with epipolar lines along image scanlines

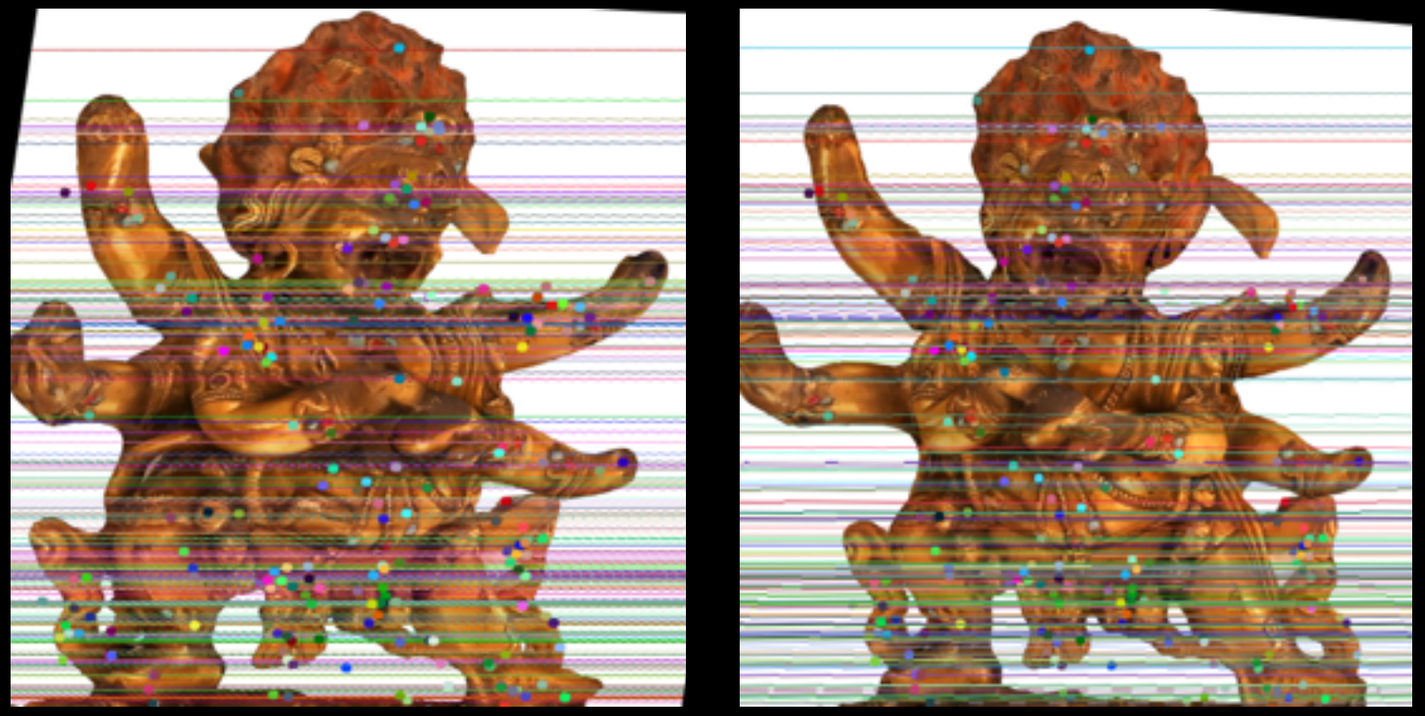
Recap: Stereo

1

2

1. Rectify images with epipolar lines along image scanlines

Recap: Stereo



1. Rectify images with epipolar lines along image scanlines

Recap: Stereo

Q: Why does a homography work here?

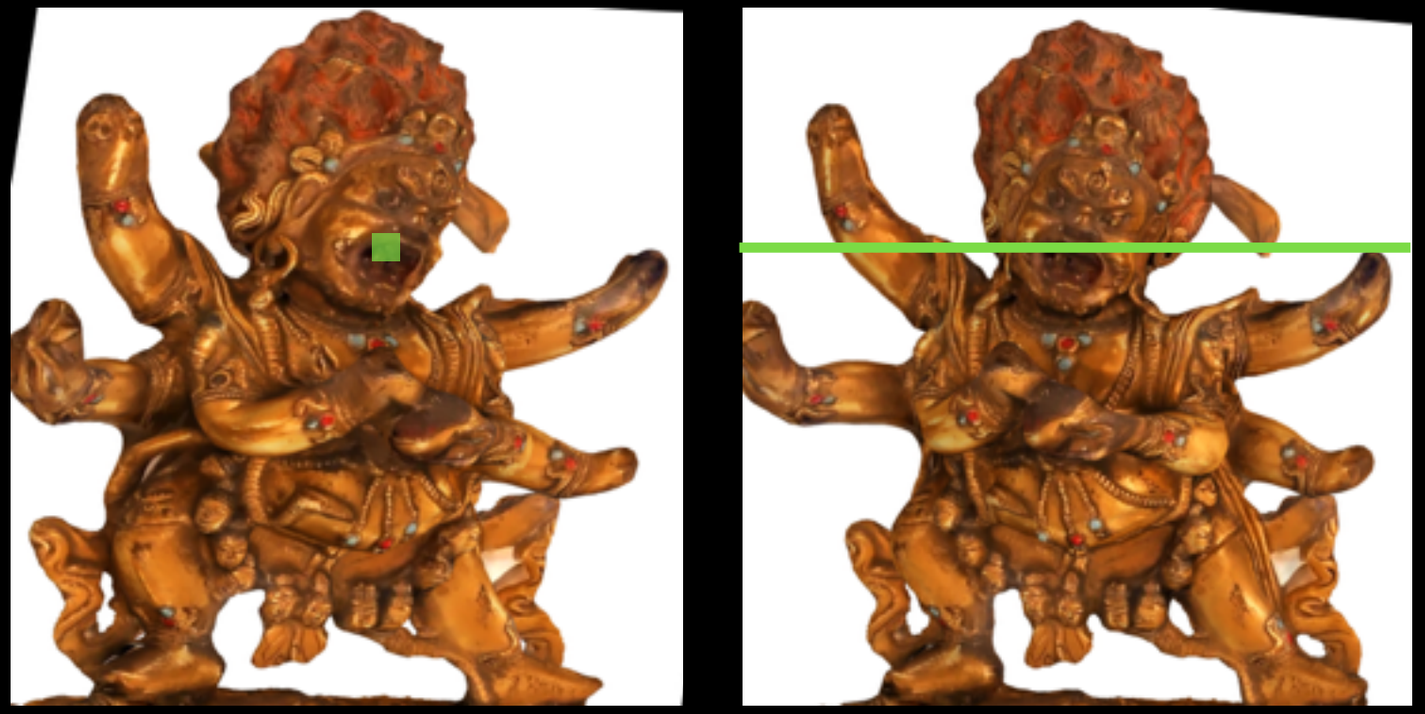
1. Rectify images with epipolar lines along image scanlines

Recap: Stereo

Q: Why does a homography work here?
A: Rotating the camera with no translation.
> Sampling same rays.

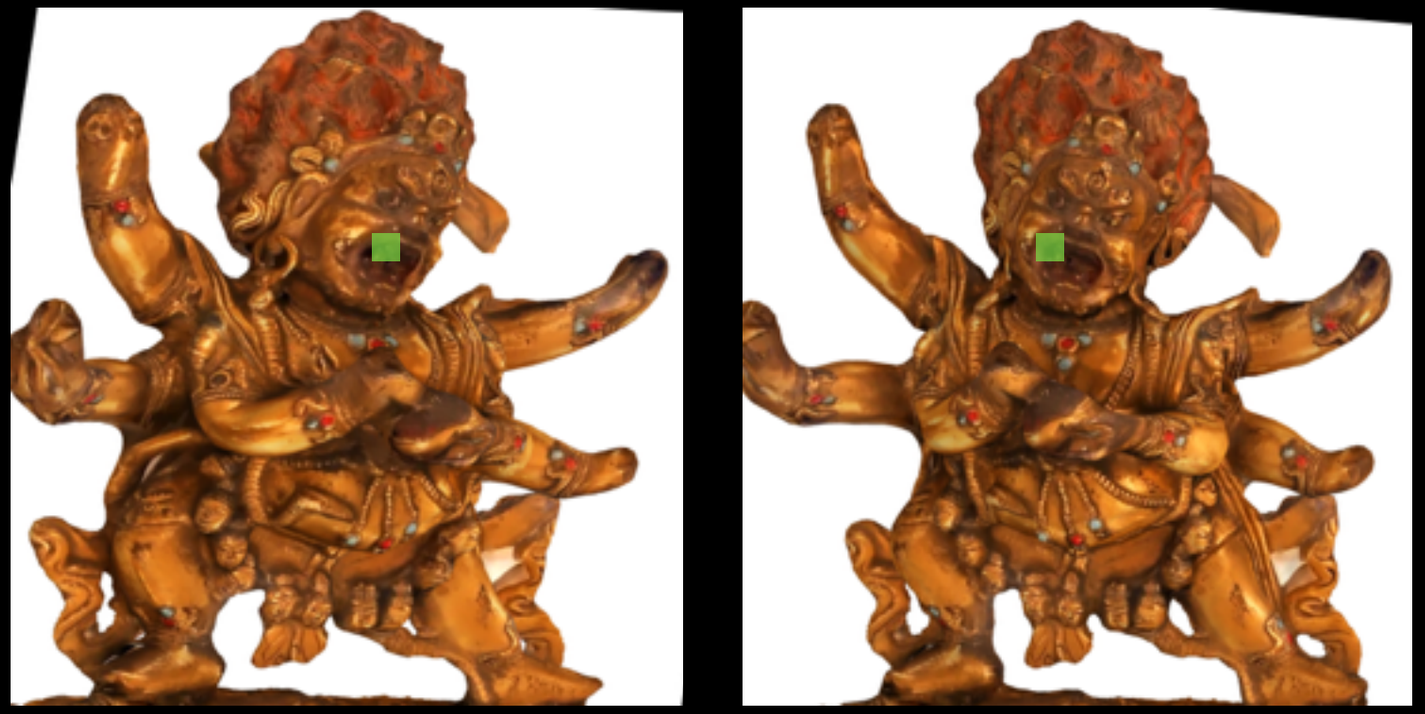
1. Rectify images with epipolar lines along image scanlines

Recap: Stereo



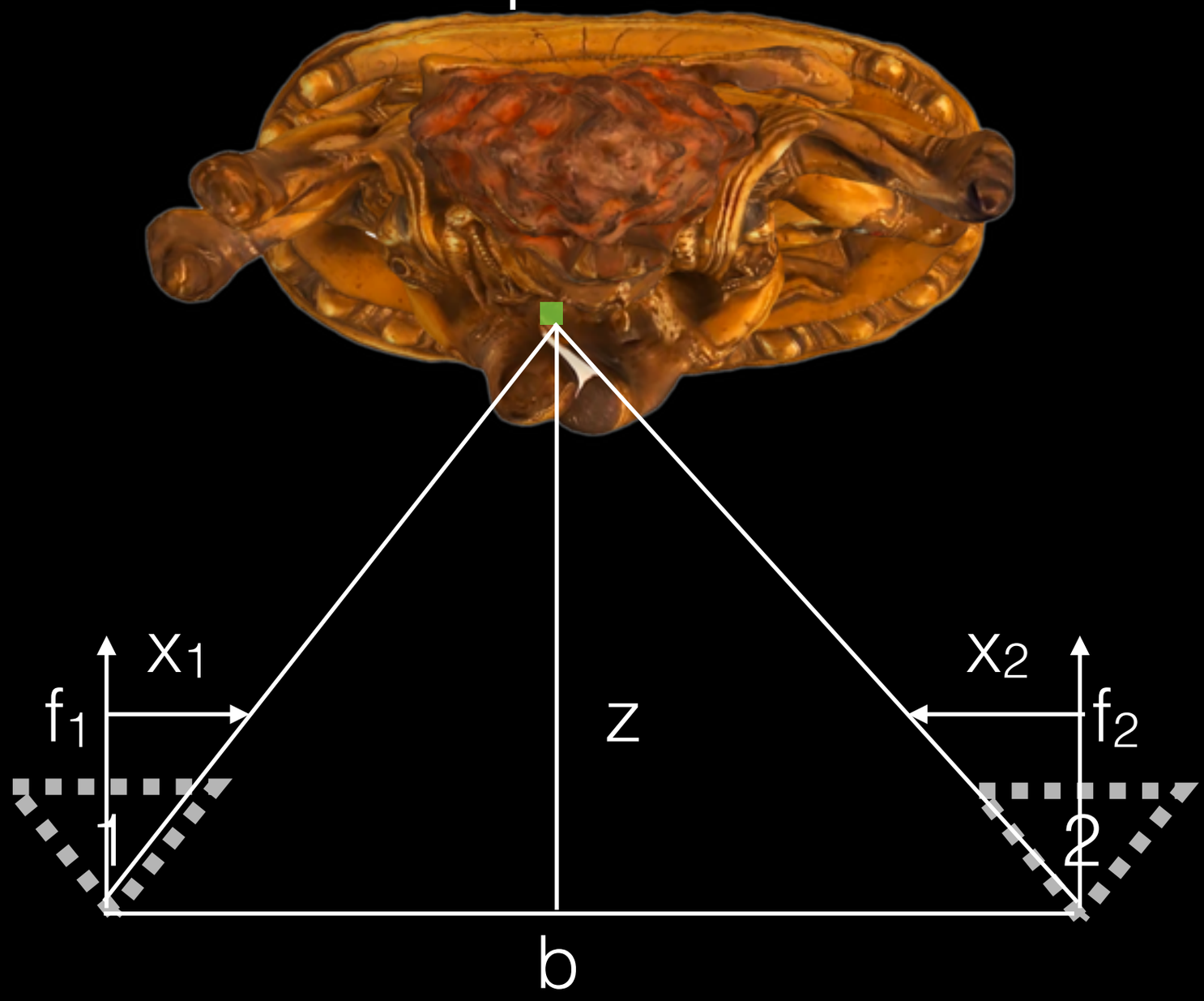
2. Compute photo-consistency score along scanlines

Recap: Stereo



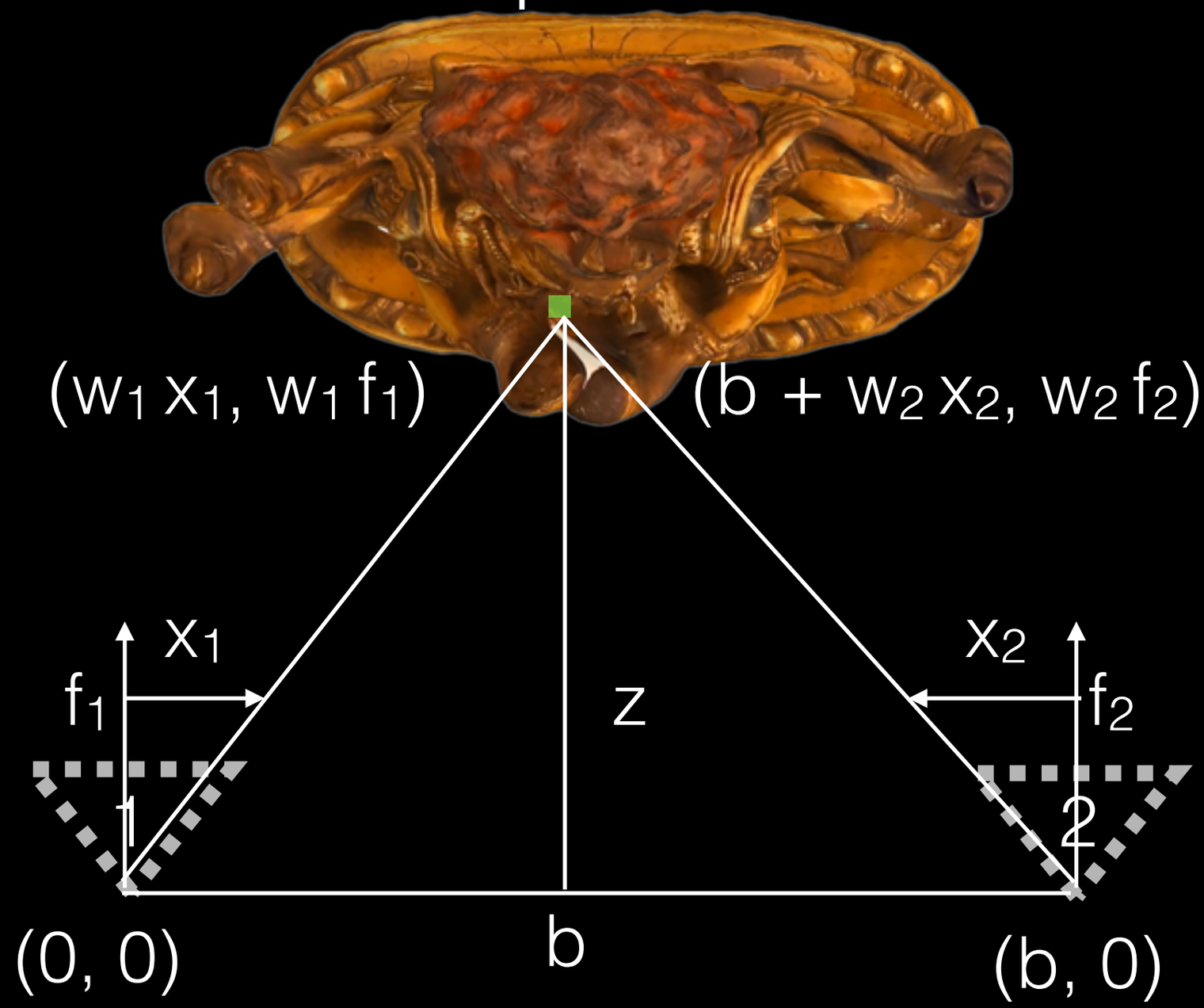
3. Select matches

Recap: Stereo



4. Compute depth

Recap: Stereo



4. Compute depth

Computing Depth

$w_1 x_1 = b + w_2 x_2$
$w_1 f_1 = w_2 f_2$
$\Rightarrow w_2 f_2 / f_1 x_1 = b + w_2 x_2$
$\Rightarrow w_2 f_2 / f_1 x_1 - w_2 x_2 = b$
$\Rightarrow w_2 (f_2 / f_1 x_1 - x_2) = b$
$\Rightarrow w_2 = b / (f_2 / f_1 x_1 - x_2)$

$z = w_2 f_2$
$\Rightarrow z = b f_2 / (f_2 / f_1 x_1 - x_2)$

If $f_1 = f_2 = f$, then simplifies to
$z = b f / (x_1 - x_2)$

Extending 2 cameras to 3+ cameras

- Two cameras
  1. Rectify images with epipolar lines along image scanlines
  2. Compute photo-consistency score along scanlines
  3. Select matches
  4. Compute depth

Extending 2 cameras to 3+ cameras

- Three+ cameras
  1. Rectify images with epipolar lines along image scanlines
  2. Compute photo-consistency score along scanlines
  3. Select matches
  4. Compute depth

Extending 2 cameras to 3+ cameras

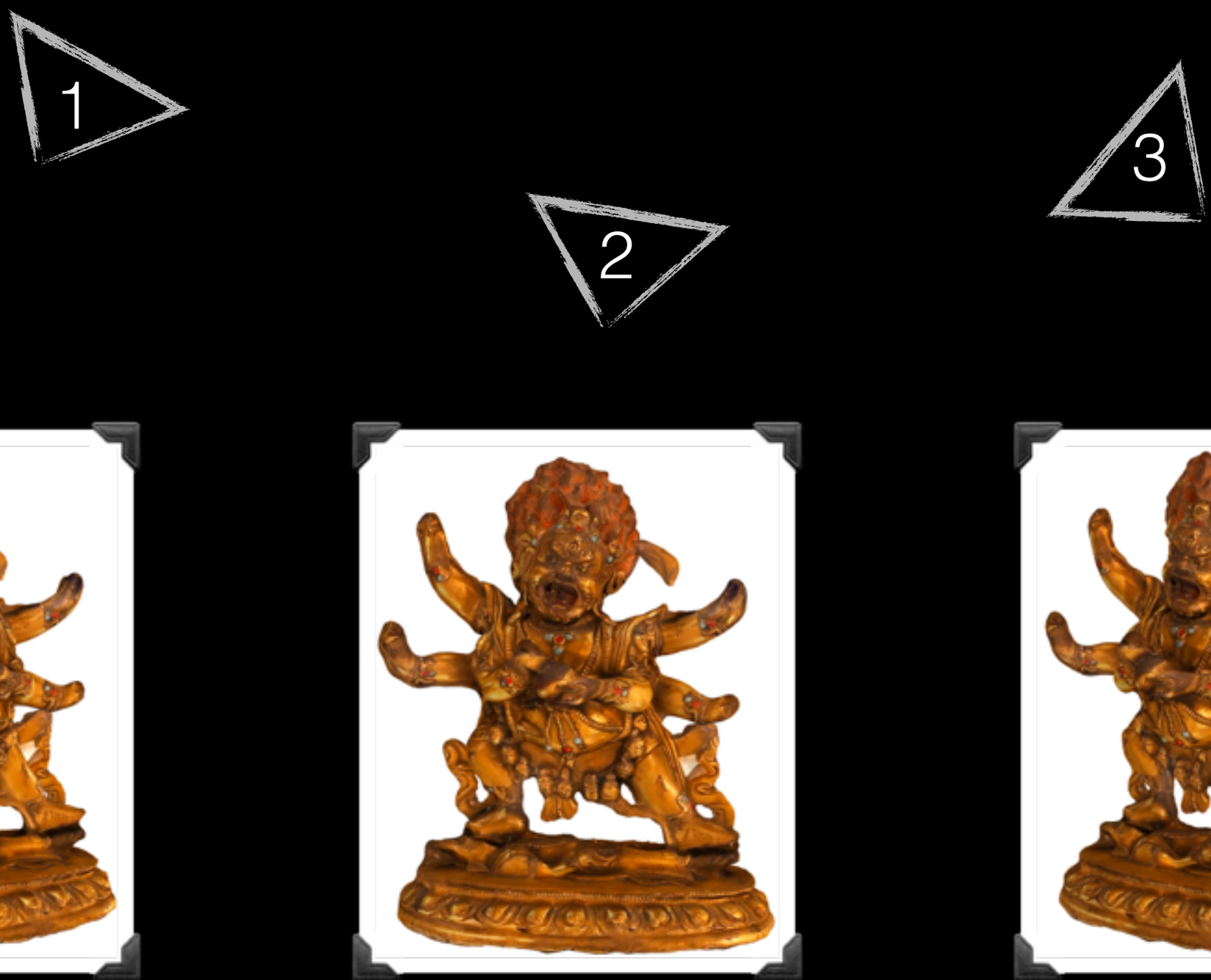
- Three+ cameras
  1. Rectify images with epipolar lines along image scanlines — How do we do this?
  2. Compute photo-consistency score along scanlines
  3. Select matches
  4. Compute depth

1

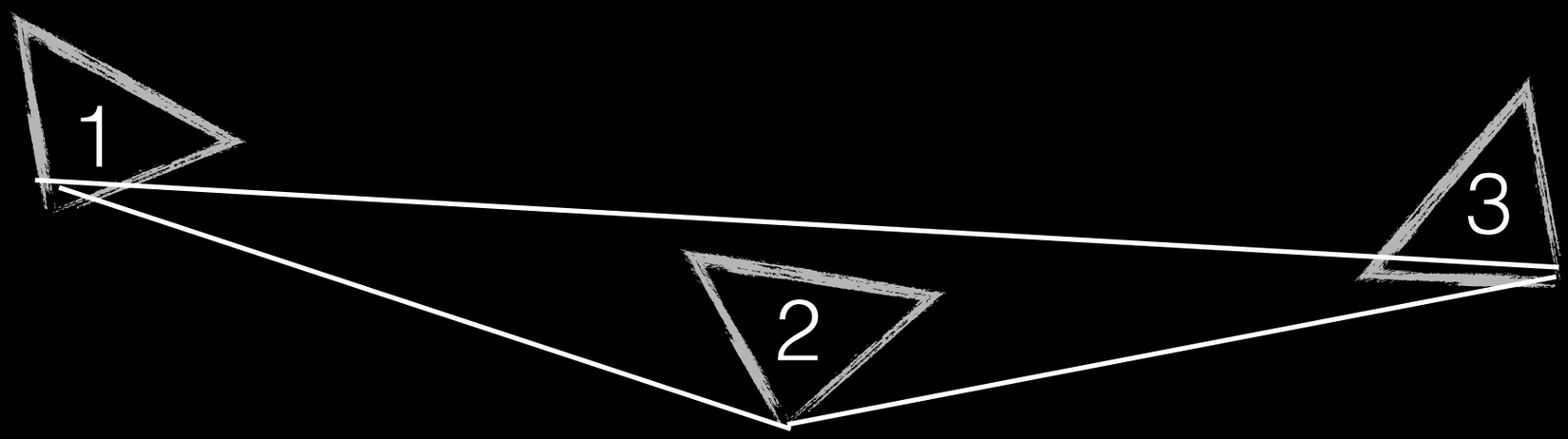
2

3

Idea: Solve several two-view subproblems and merge result



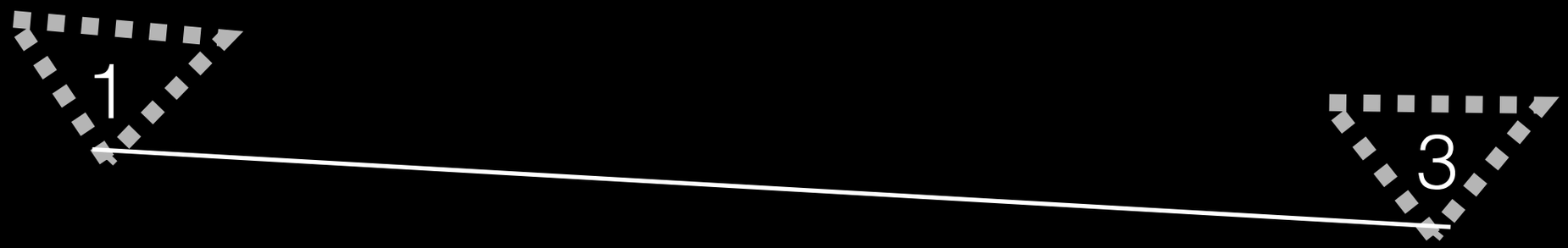
3 cameras => 3 baselines
6 homographies



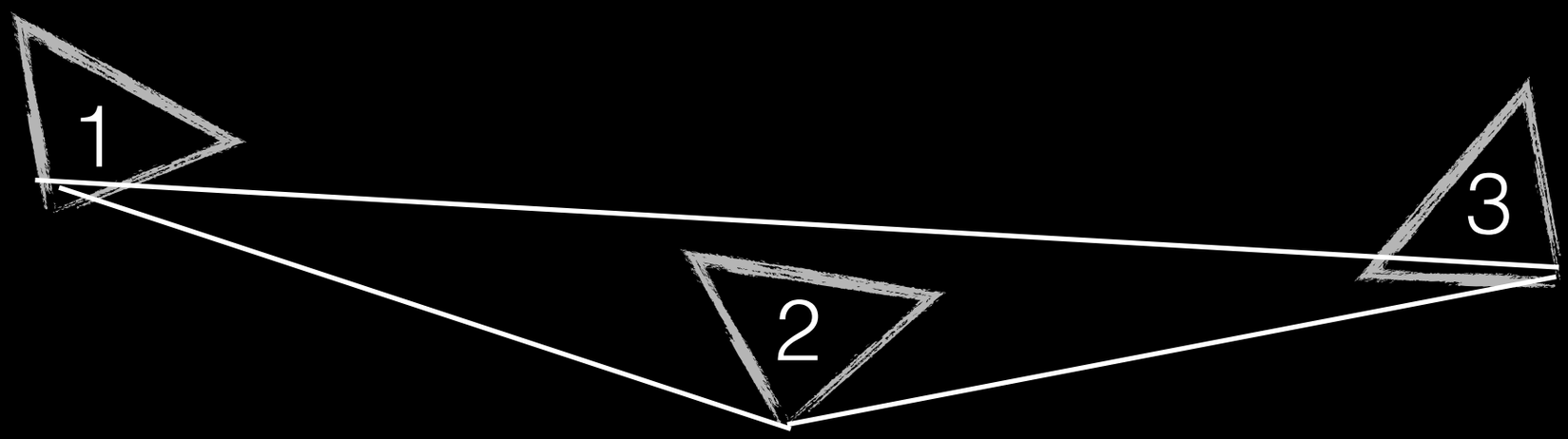
3 cameras => 3 baselines
6 homographies

3 cameras => 3 baselines
6 homographies

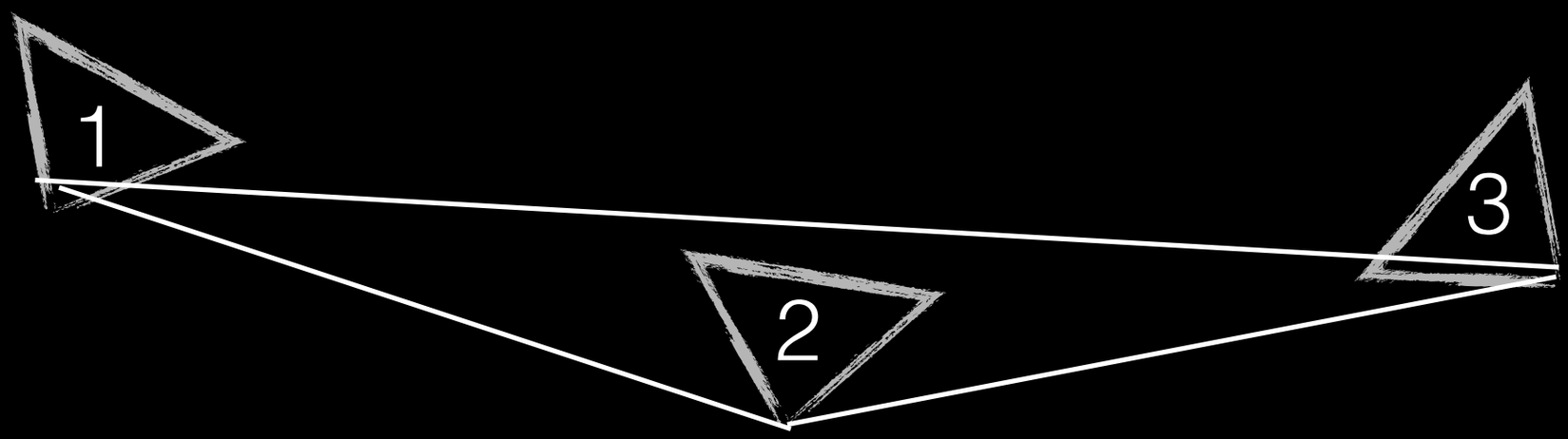
3 cameras => 3 baselines
6 homographies



n cameras => n * (n - 1) / 2 baselines
n * (n - 1) homographies



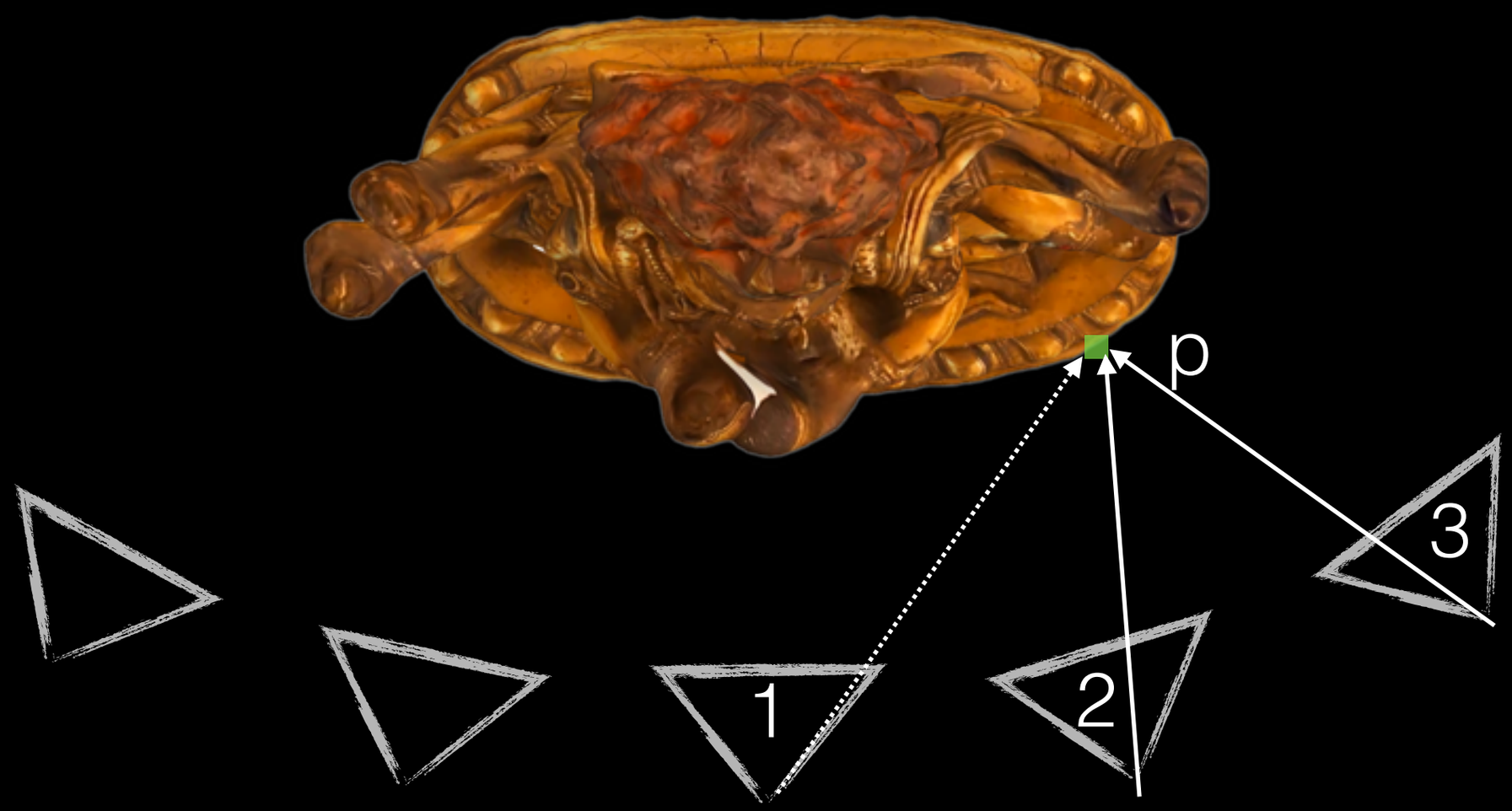
Harder than your regular 2 camera stereo problem!



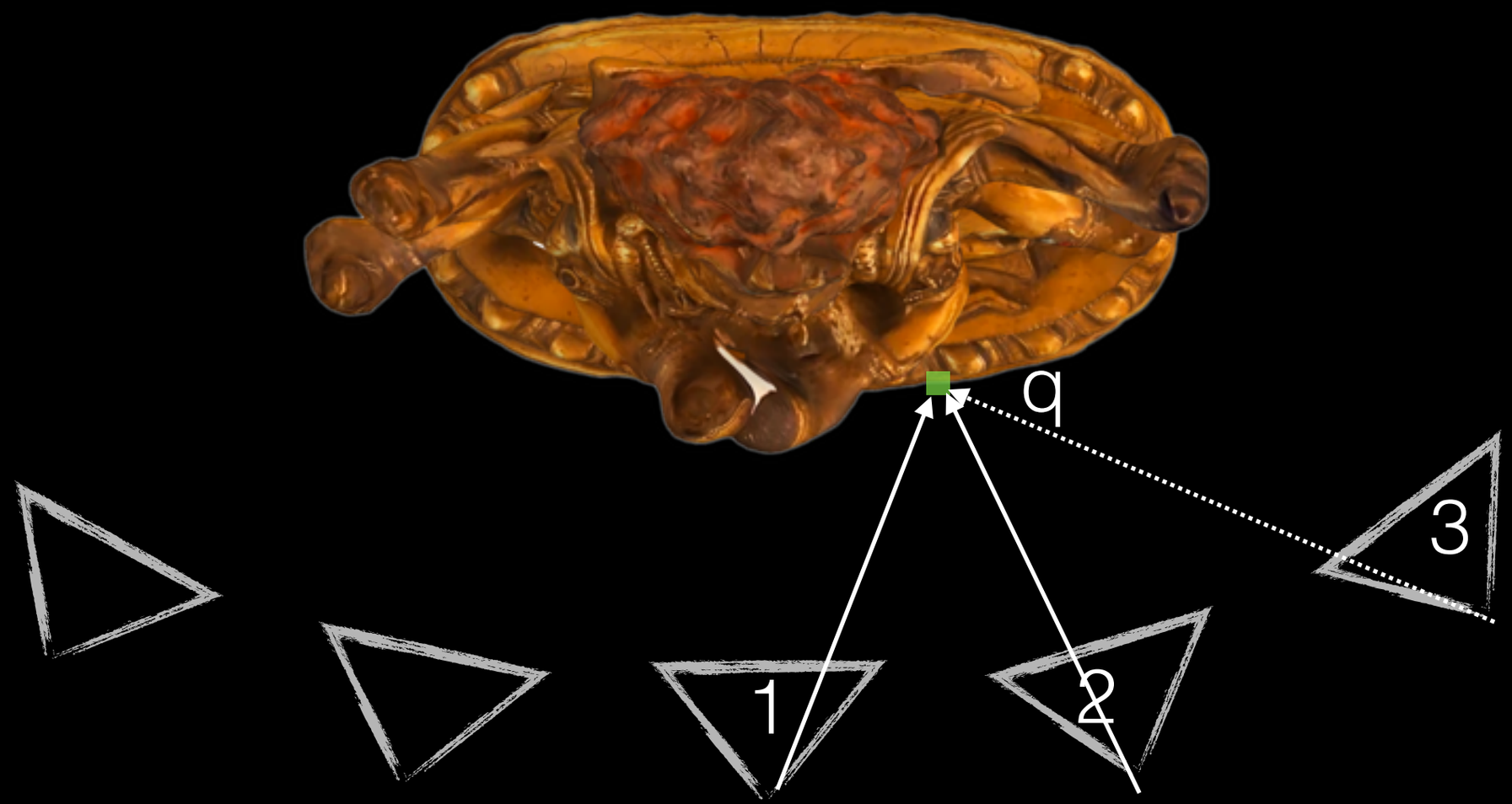
Why MVS?

Why MVS?

- Different points on the object's surface will be more clearly visible in some subset of cameras
  - Could have high res closeups of some regions
  - Some surfaces are foreshortened from certain views



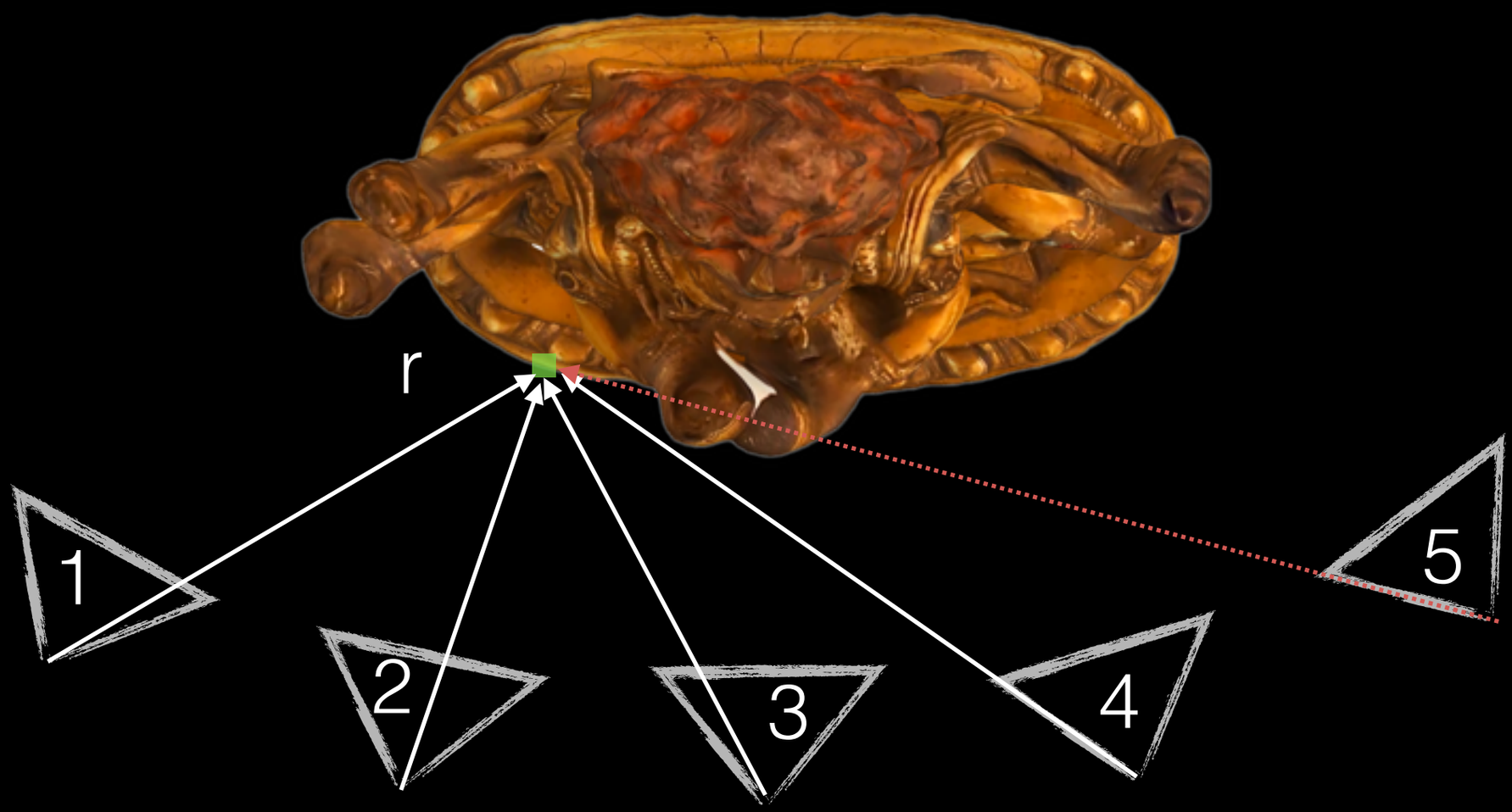
Cameras 2 and 3 can more clearly see point p.



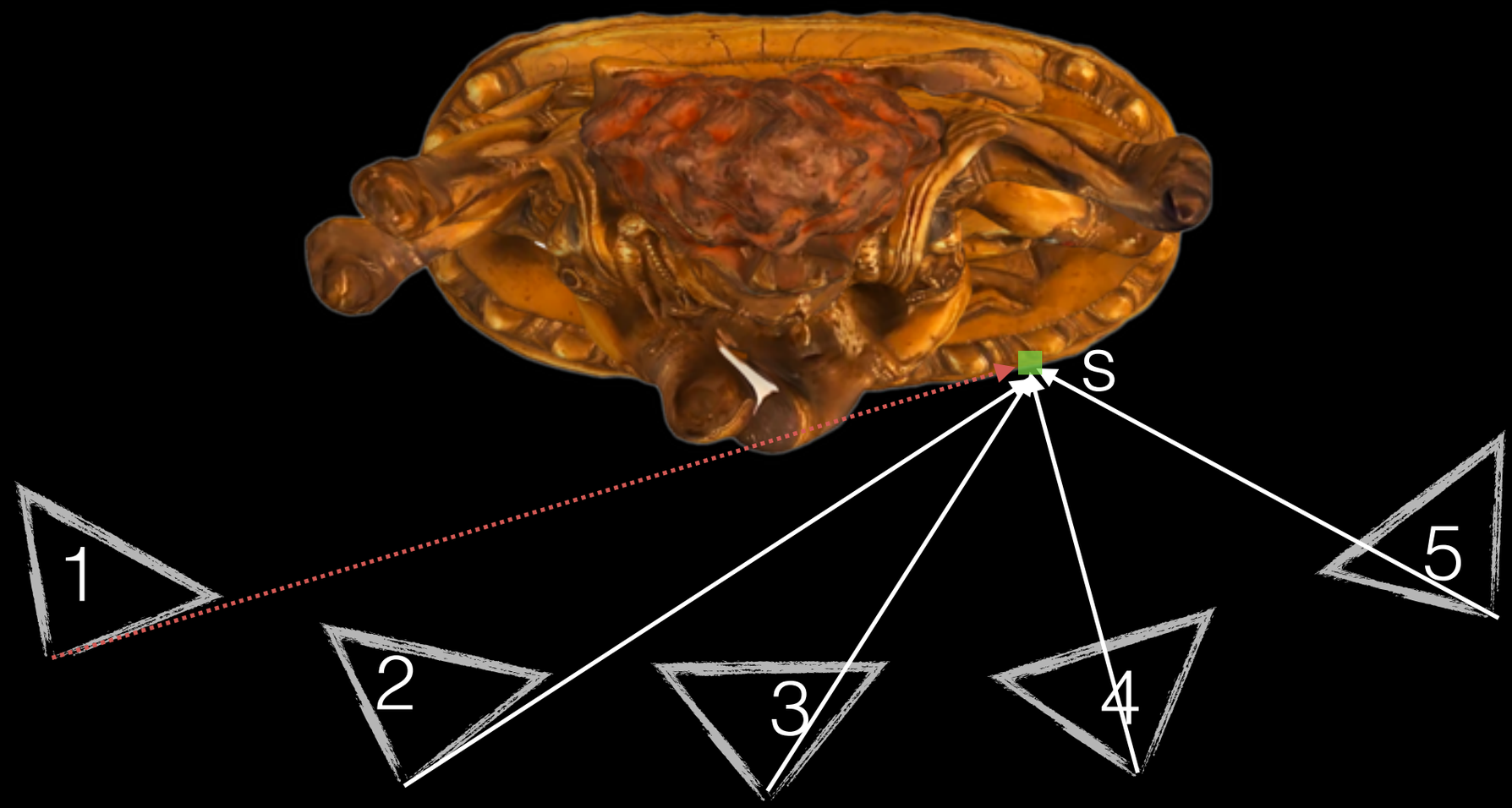
Cameras 1 and 2 can more clearly see point q.

Why MVS?

- Different points on the object's surface will be more clearly visible in some subset of cameras
  - Could have high res closeups of some regions
  - Some surfaces are foreshortened from certain views
- Some points may be occluded entirely in certain views



Camera 5 can’t see point r.



Camera 1 can’t see point s.

Why MVS?

- Different points on the object's surface will be more clearly visible in some subset of cameras
  - Could have high res closeups of some regions
    - Some surfaces are foreshortened from certain views
- Some points may be occluded entirely in certain views
- More measurements per point can reduce error

1

2

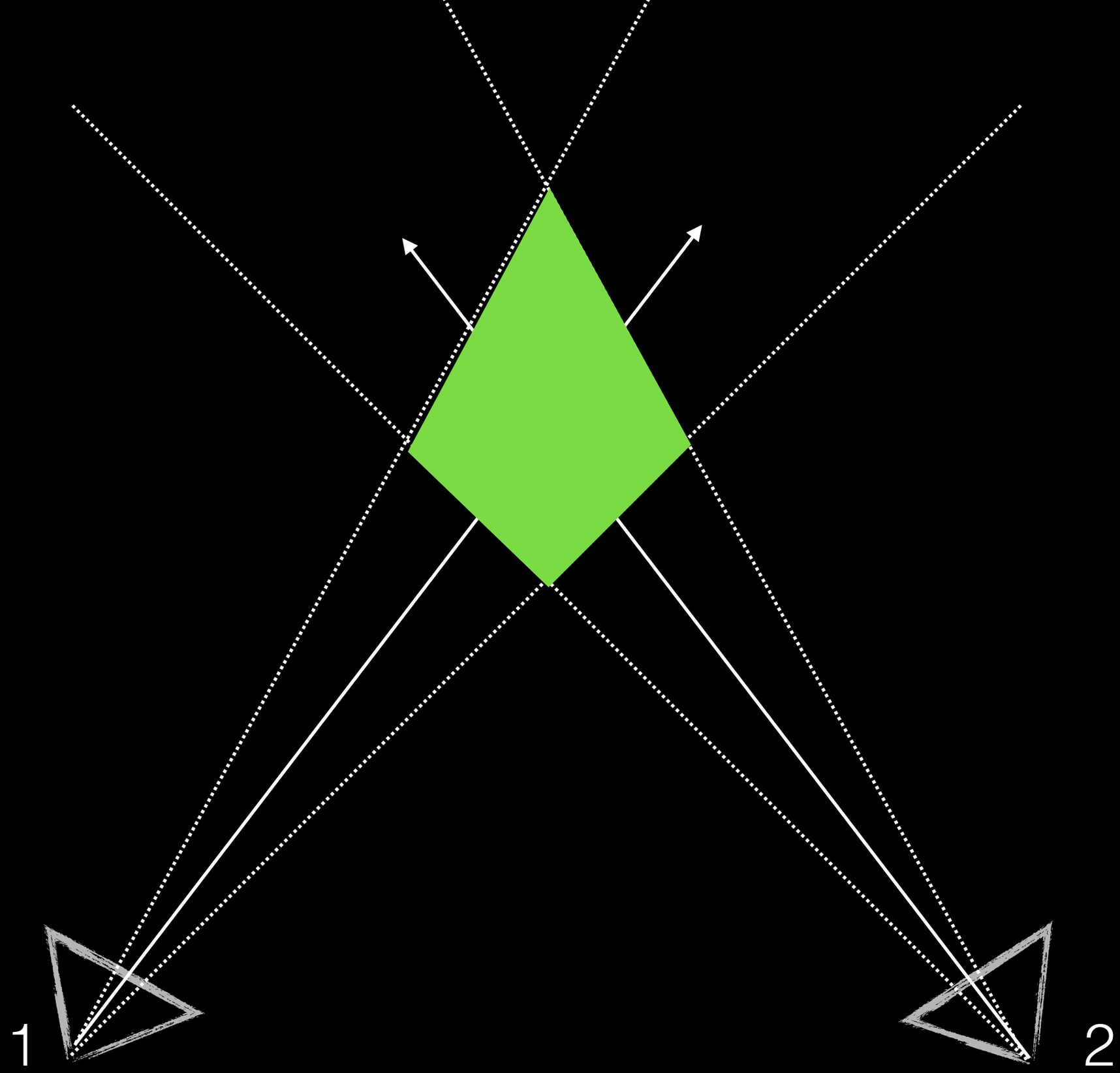
1

2

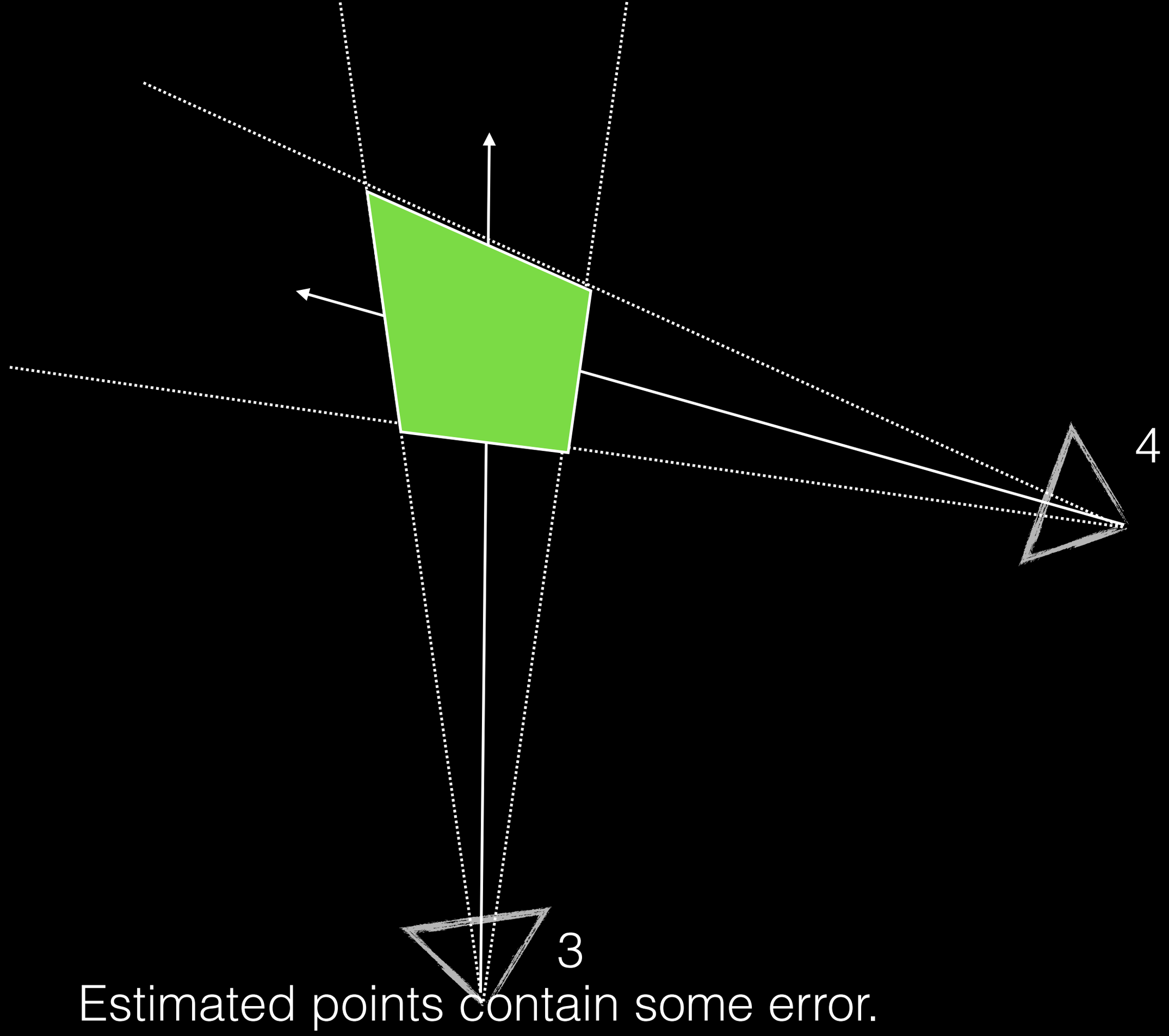
1

2

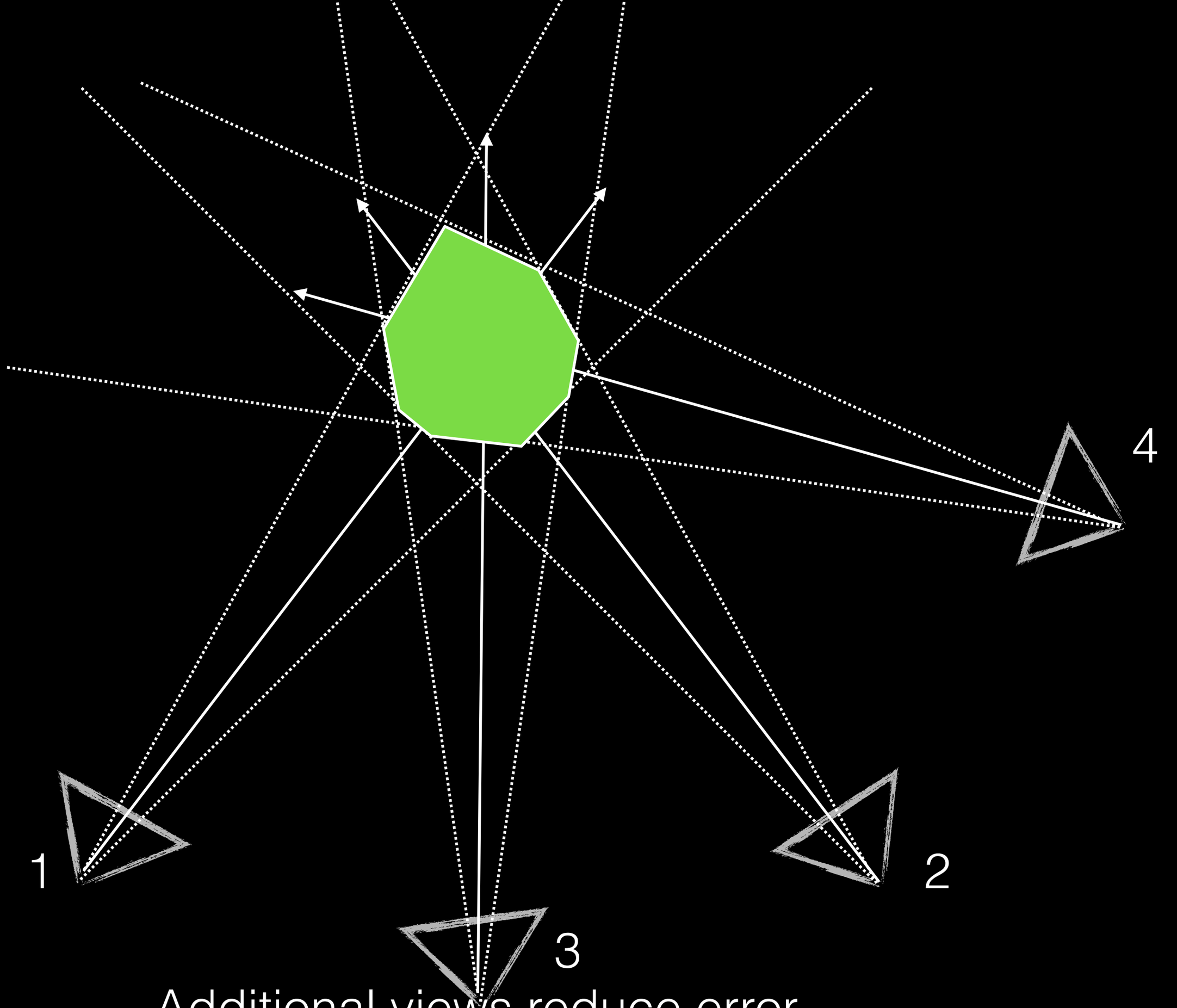
Estimated points contain some error.



Estimated points contain some error.



Estimated points contain some error.



Additional views reduce error.

Why MVS?

- Different points on the object's surface will be more clearly visible in some subset of cameras
    - Could have high res closeups of some regions
        - Some surfaces are foreshortened from certain views
- Some points may be occluded entirely in certain views
- More measurements per point can reduce error
- More measurements per point can **reduce ambiguity**

Extending 2 cameras to 3+ cameras

- Three+ cameras
    1. Rectify images with epipolar lines along image scanlines
    2. Compute photo-consistency score along scanlines
    3. Select matches — Easier with more cameras?
    4. Compute depth

Reducing ambiguity

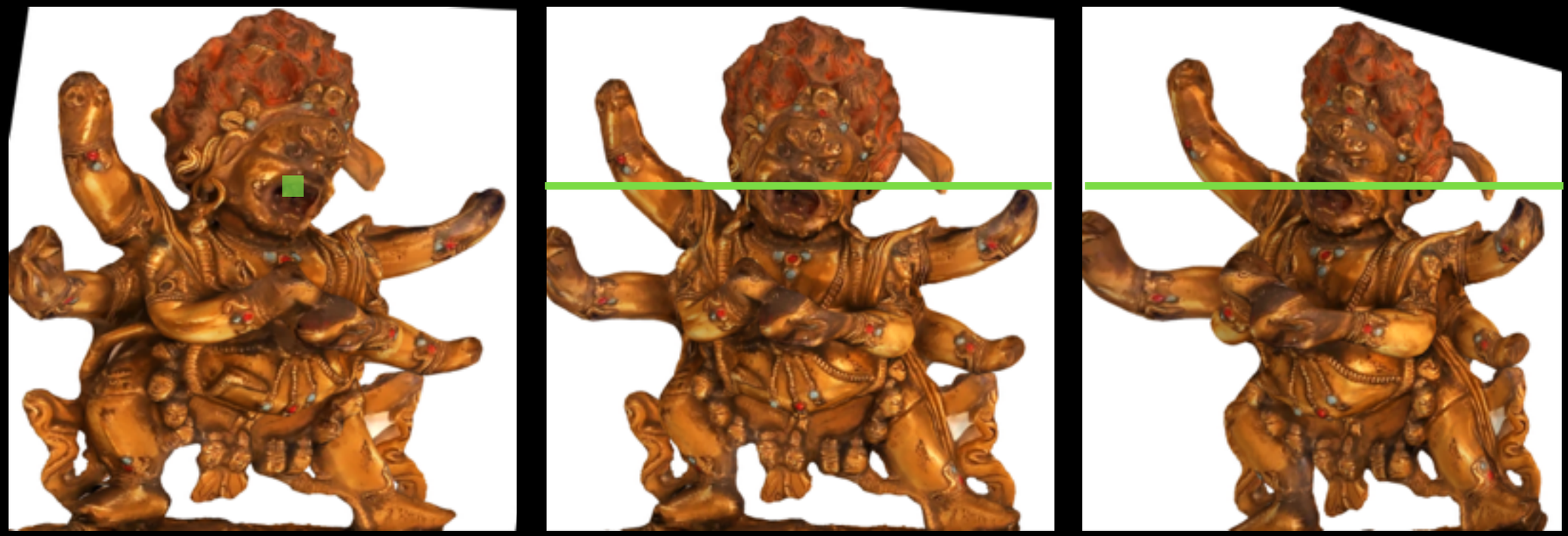


Photo-consistency score
(Higher better)

Reducing ambiguity

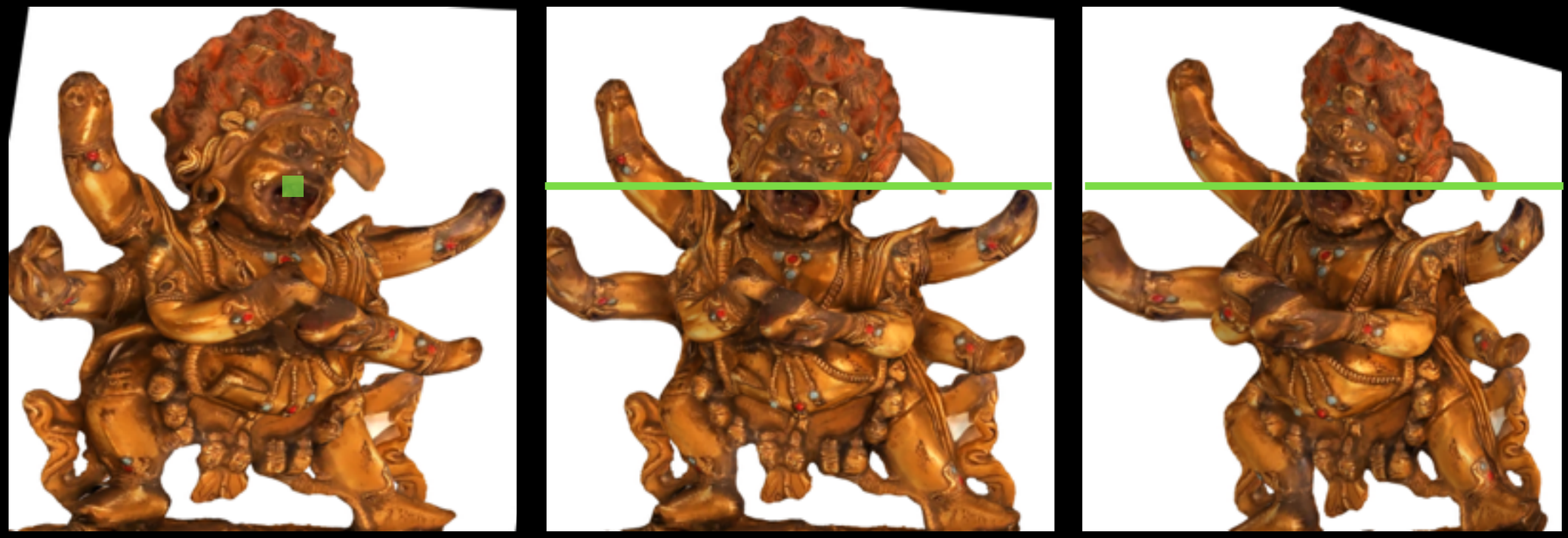
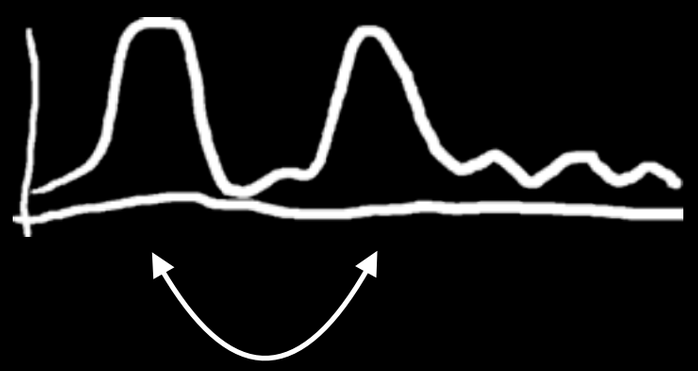
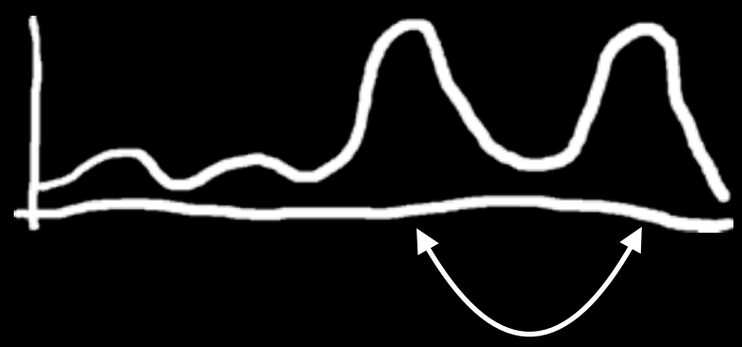


Photo-consistency score
(Higher better)



Ambiguous



Ambiguous

Reducing ambiguity

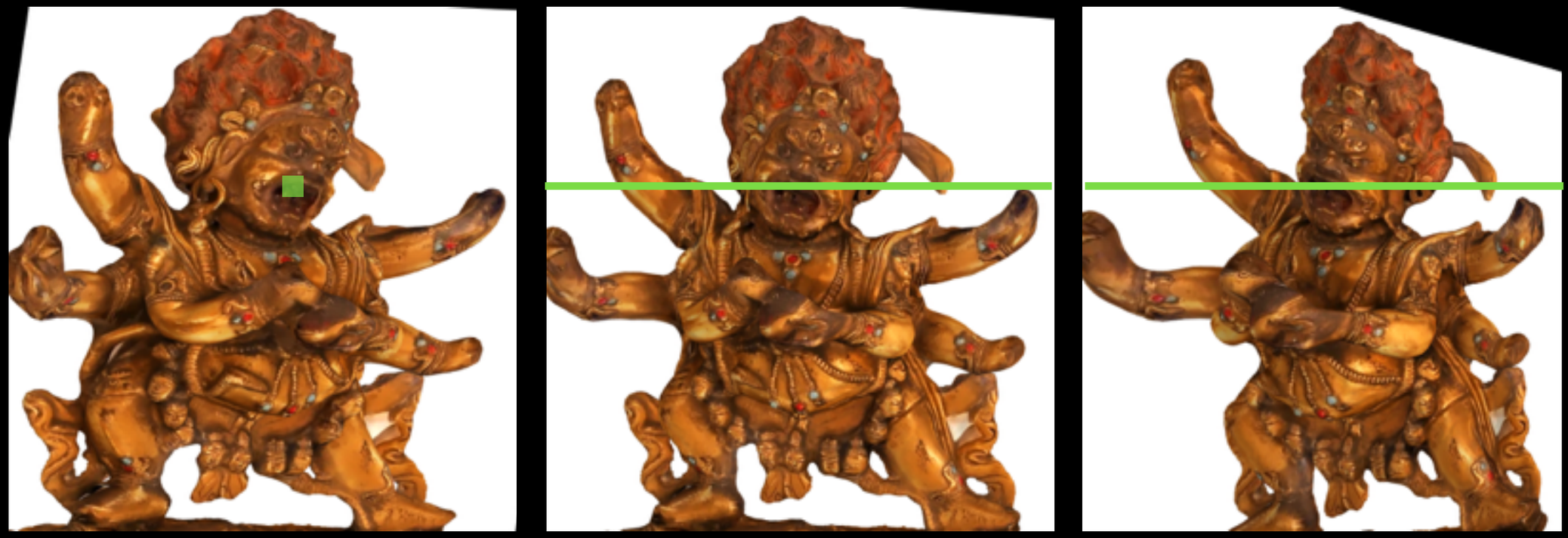


Photo-consistency score
(Higher better)

Constraints mitigate ambiguity

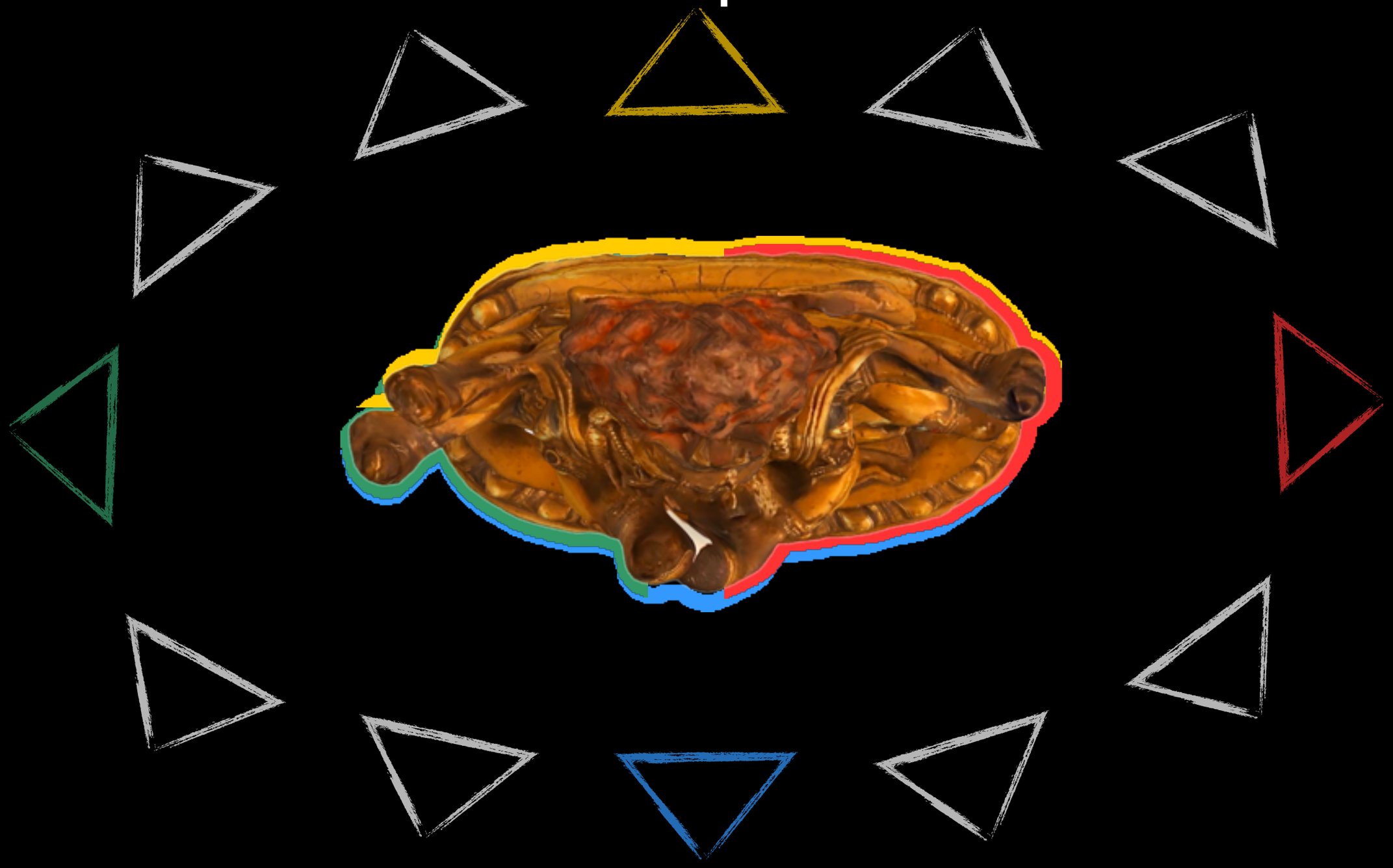
Extending 2 cameras to 3+ cameras

- Three+ cameras
  1. Rectify images with epipolar lines along image scanlines
  2. Compute photo-consistency score along scanlines
  3. Select matches
  4. Compute depth — How do we use **many** depth maps? Build a super depth map, or something else?

Single depth map often isn’t enough

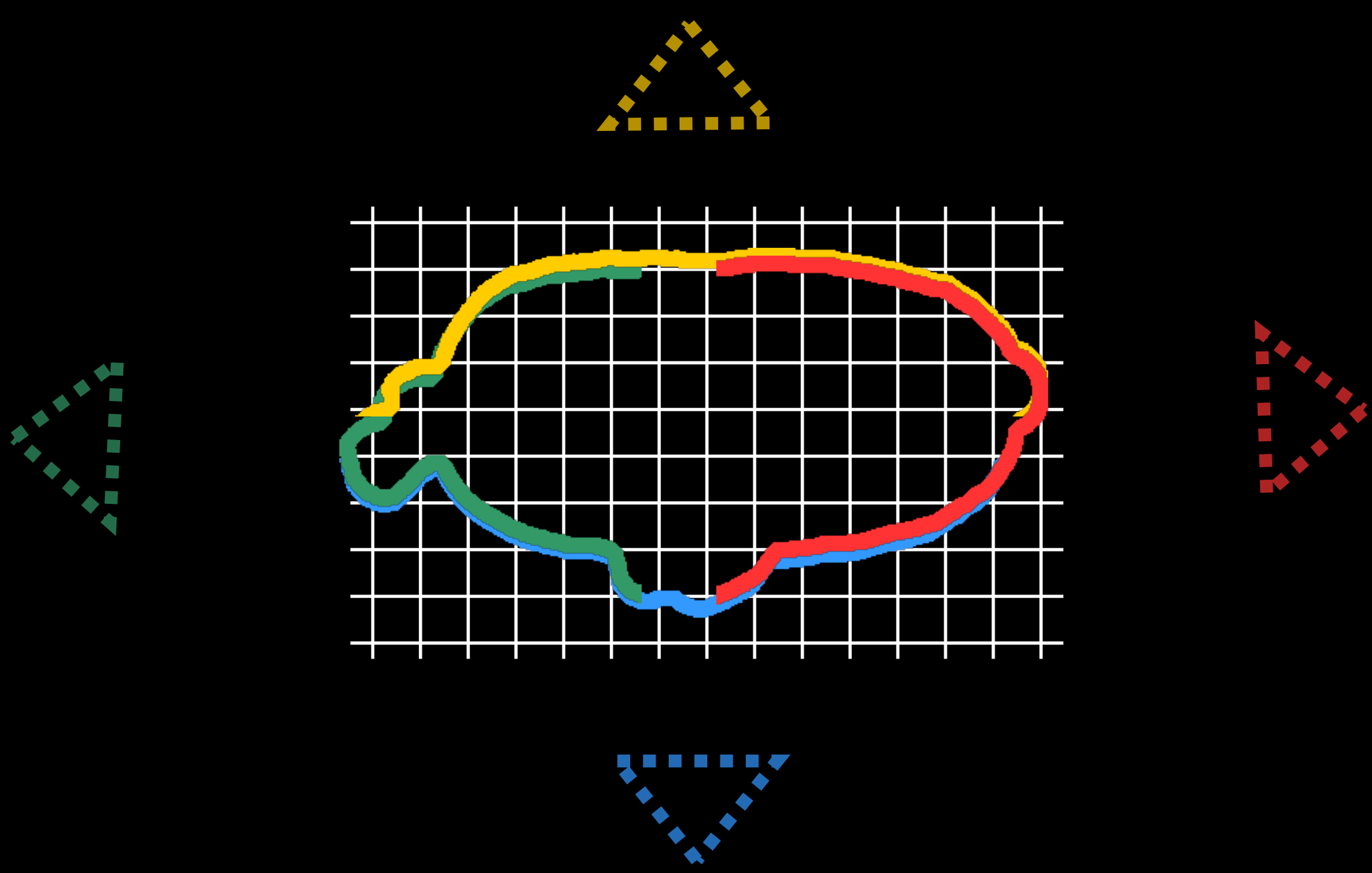
Really want full coverage

Idea: Combine many depth maps



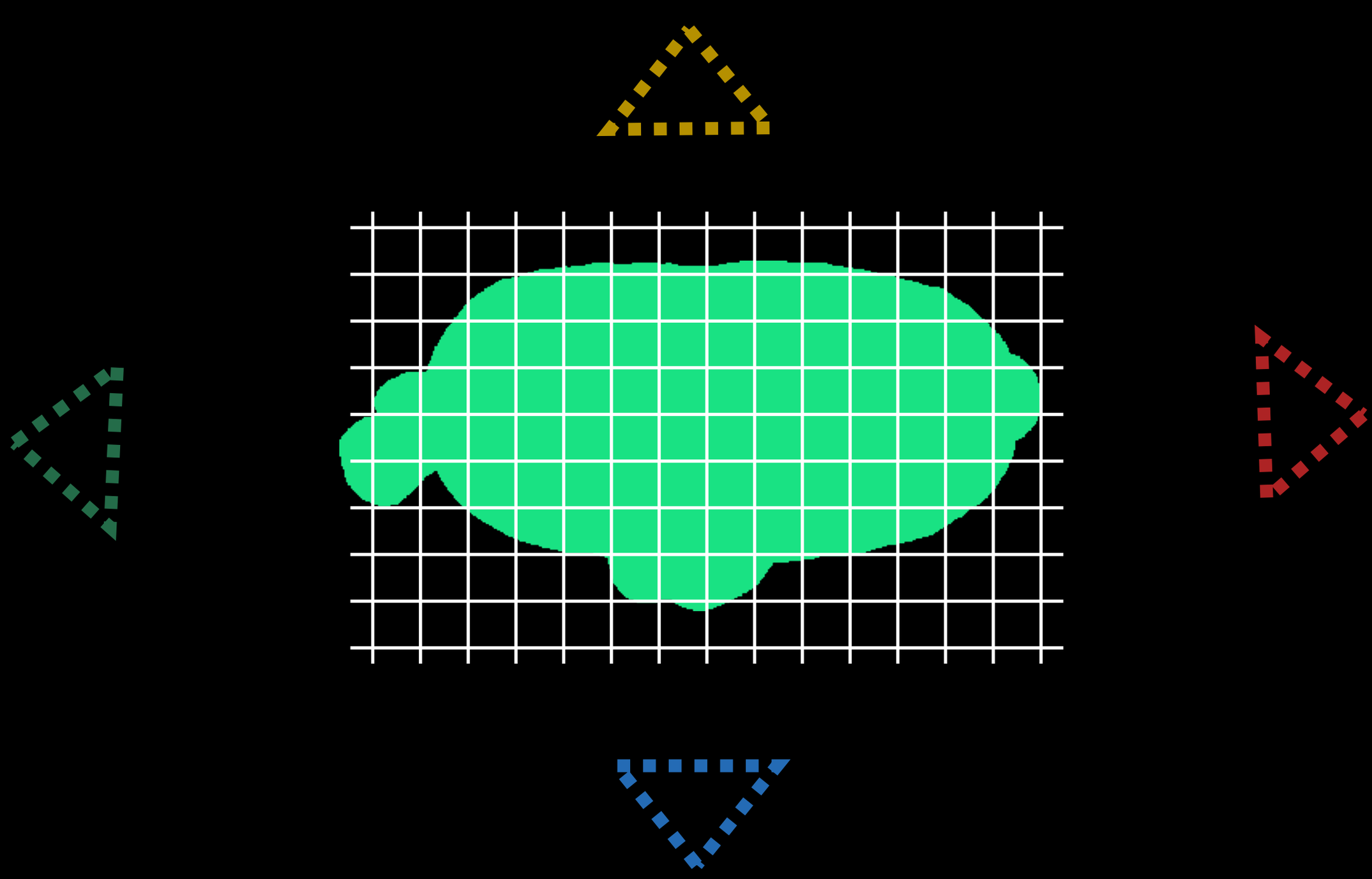
Many depth maps, each with error. How can we fuse these?

Volumetric fusion



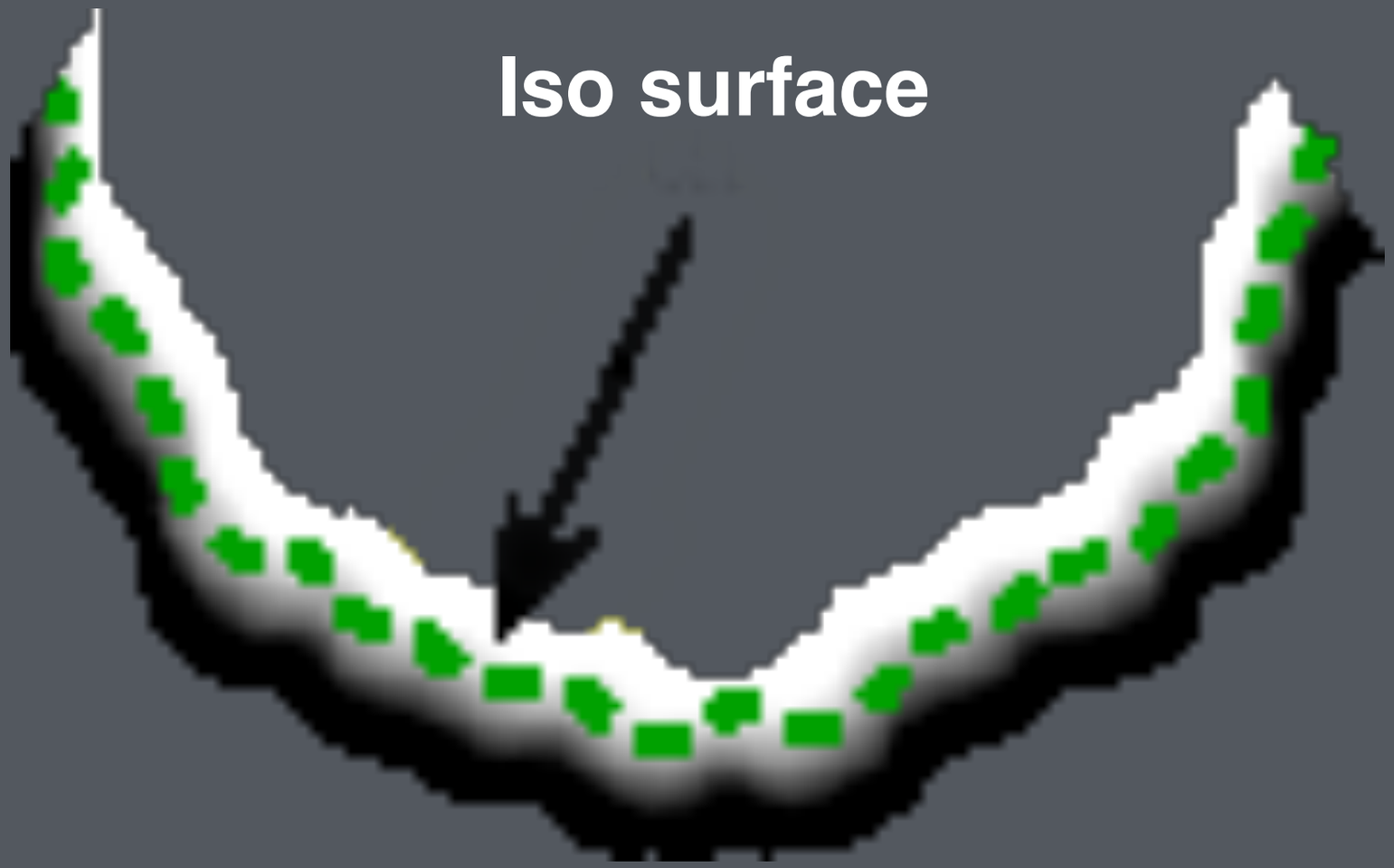
A common world-space coordinate system.

Volumetric fusion



A common world-space coordinate system.

Iso surface



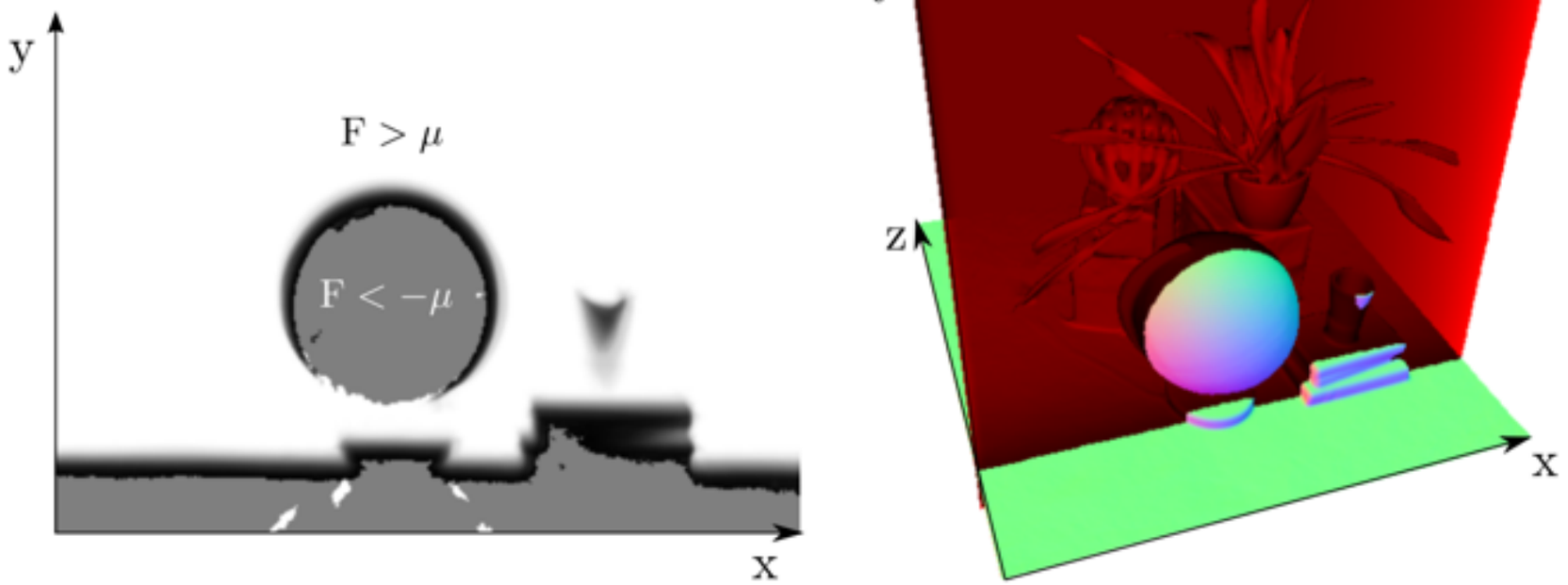
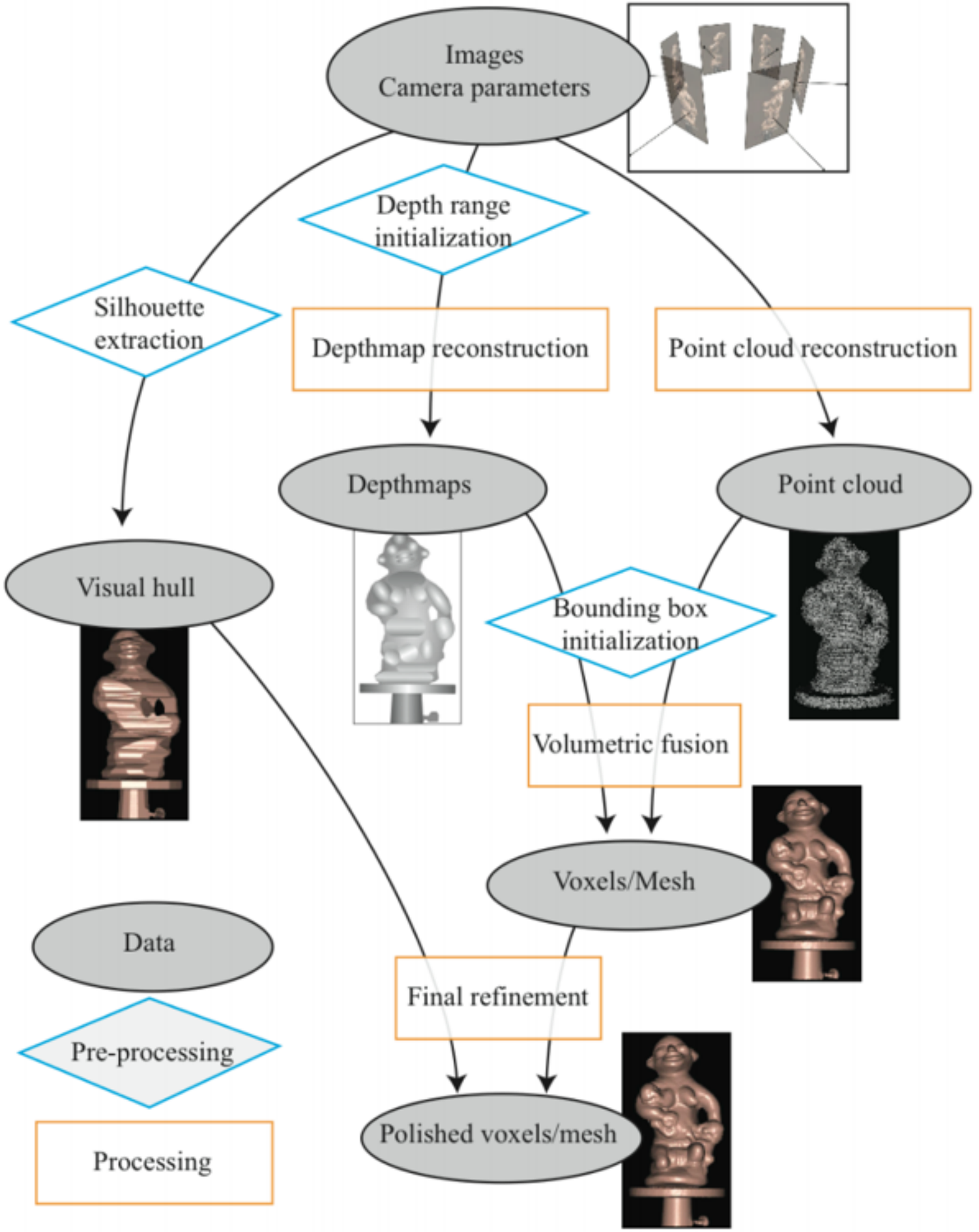
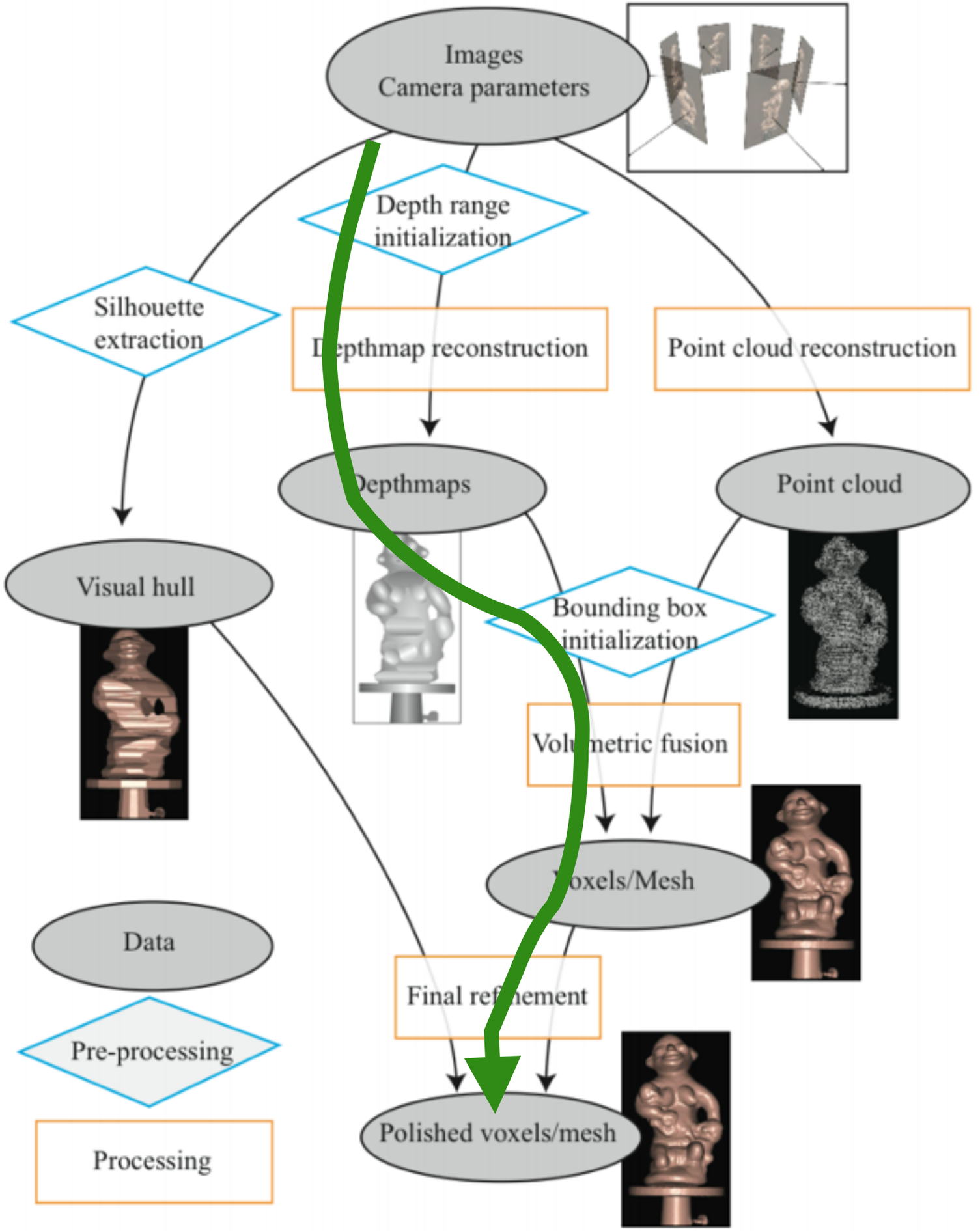


Figure 4: A slice through the truncated signed distance volume showing the truncated function $F > \mu$ (white), the smooth distance field around the surface interface $F = 0$ and voxels that have not yet had a valid measurement(grey) as detailed in eqn. 9.

Newcombe, et. al. ISMAR 2011.


Images
Camera parameters
Depth range initialization
Silhouette extraction
Depthmap reconstruction
Point cloud reconstruction
Depthmaps
Point cloud
Visual hull
Bounding box initialization
Volumetric fusion
Voxels/Mesh
Final refinement
Polished voxels/mesh
Data
Pre-processing
Processing


Images
Camera parameters
Depth range initialization
Silhouette extraction
Depthmap reconstruction
Point cloud reconstruction
Depthmaps
Point cloud
Visual hull
Bounding box initialization
Volumetric fusion
Voxels/Mesh
Final refinement
Polished voxels/mesh
Data
Pre-processing
Processing

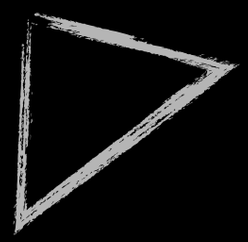
There are **many** variations on this problem.

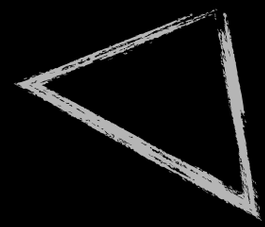
PA4 Part 2
Plane Sweep Stereo

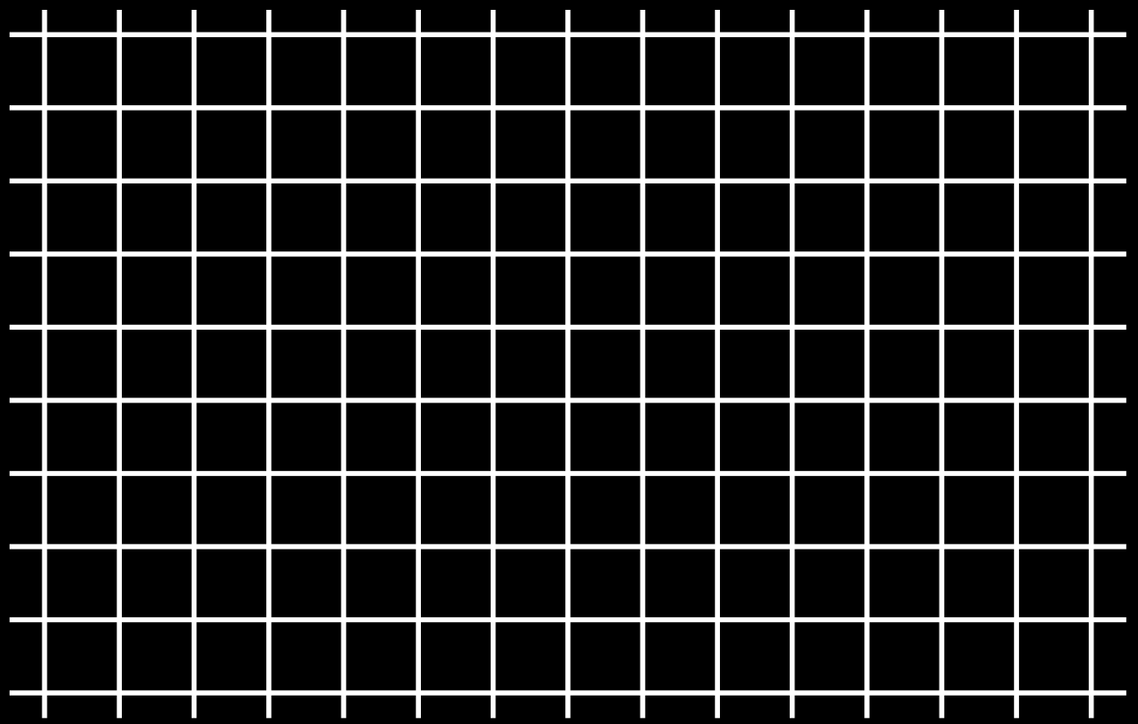
- Can we replace the n * (n - 1) / 2 camera pairs with something more unified?

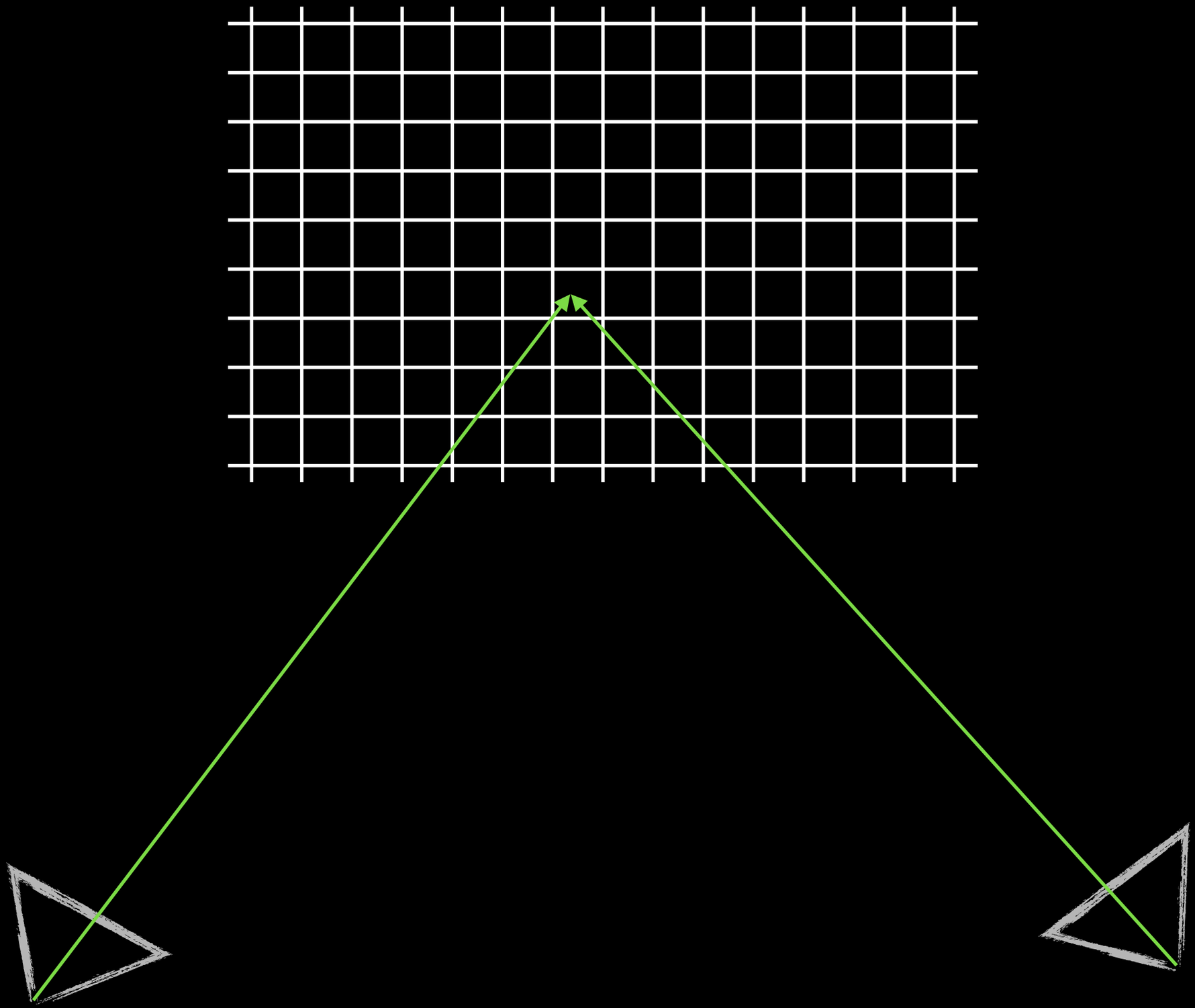
- We ended up using a volumetric representation to merge depth maps.

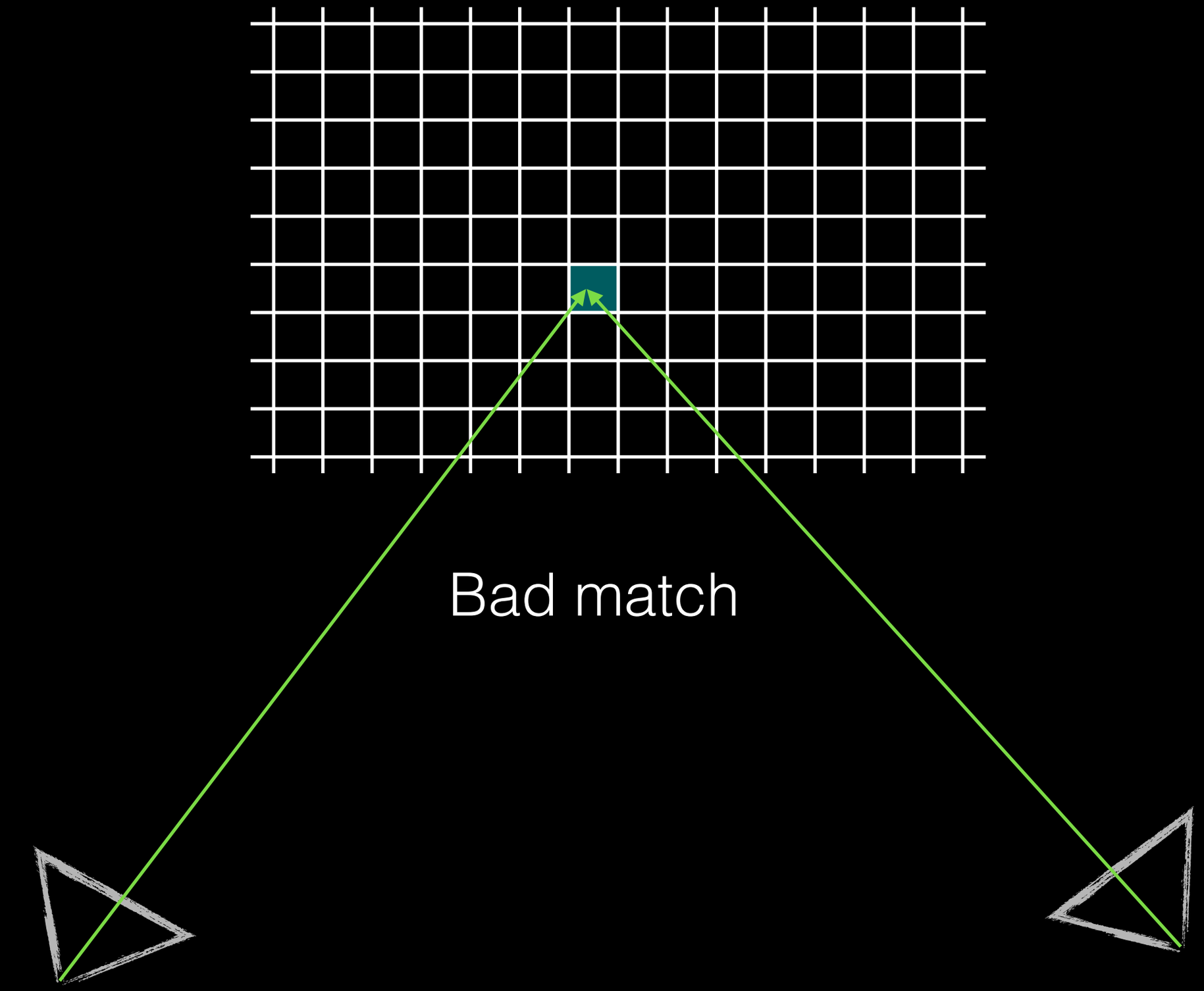
- What if we start with a volumetric representation instead?

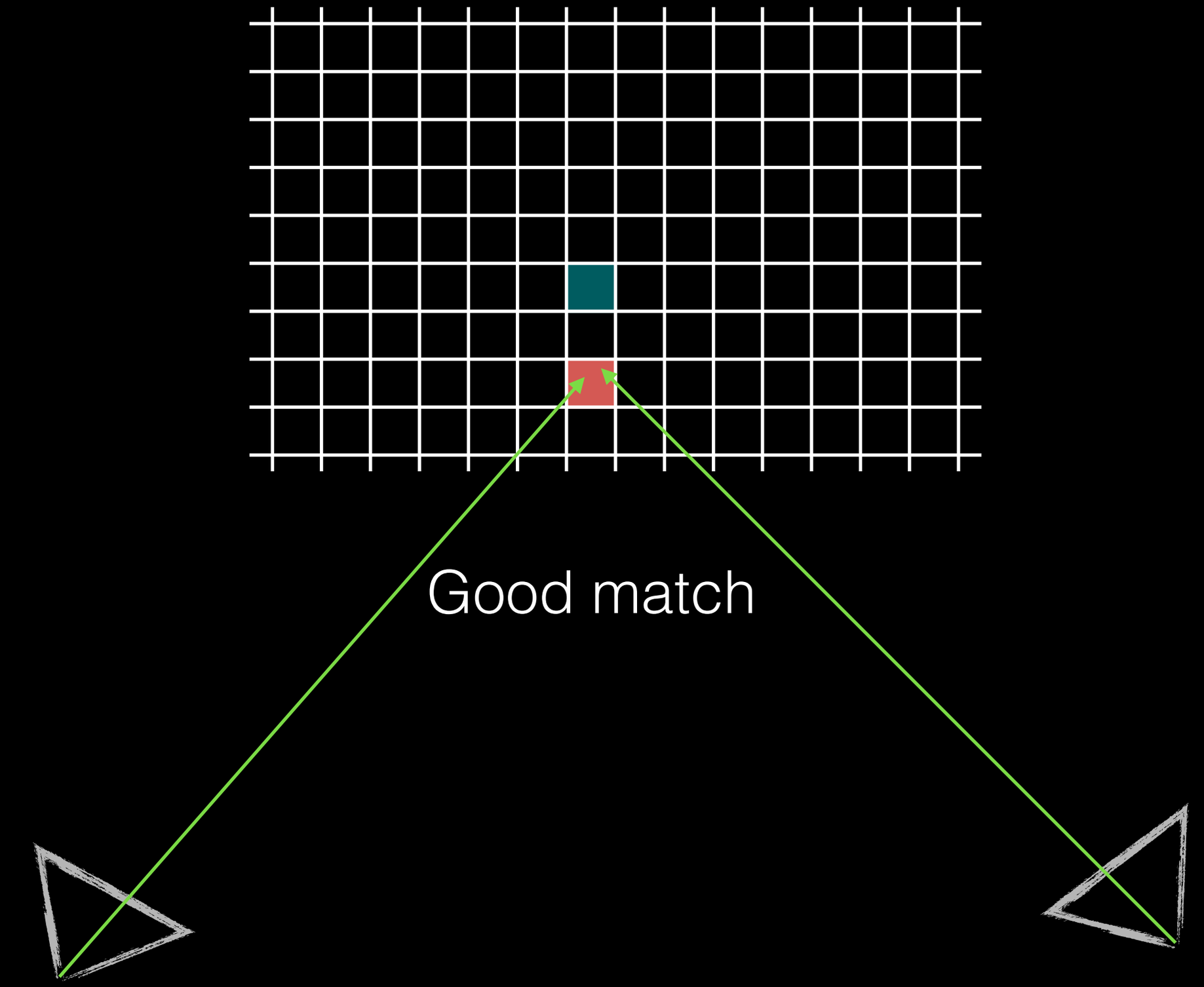


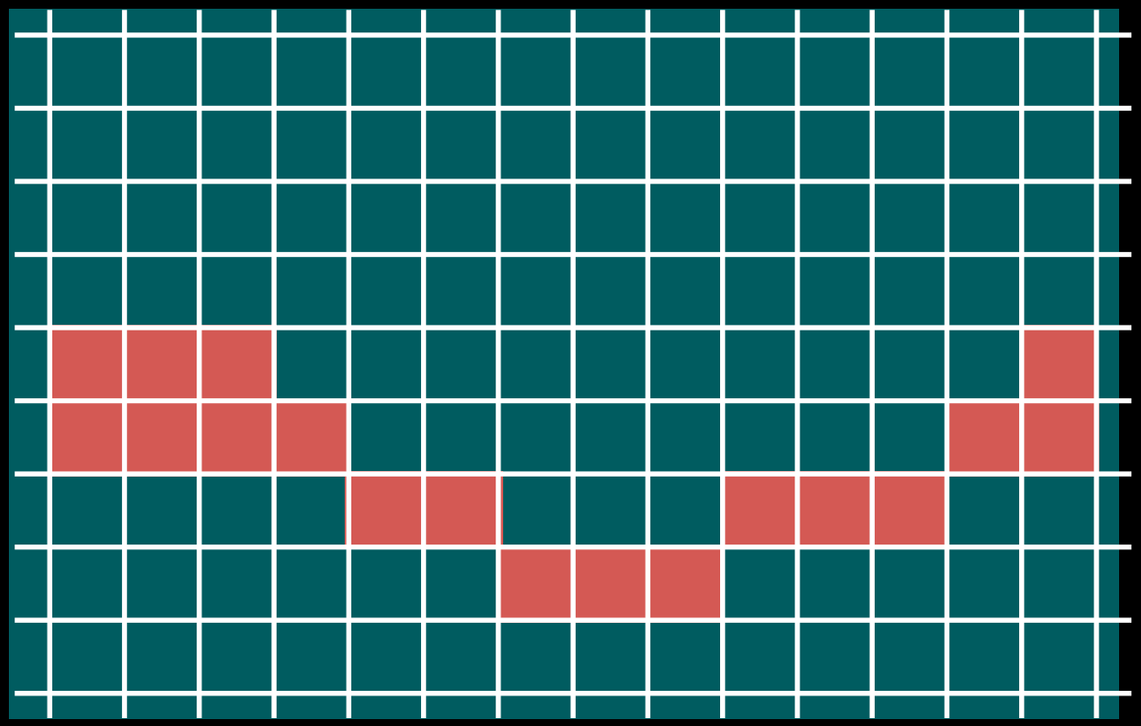


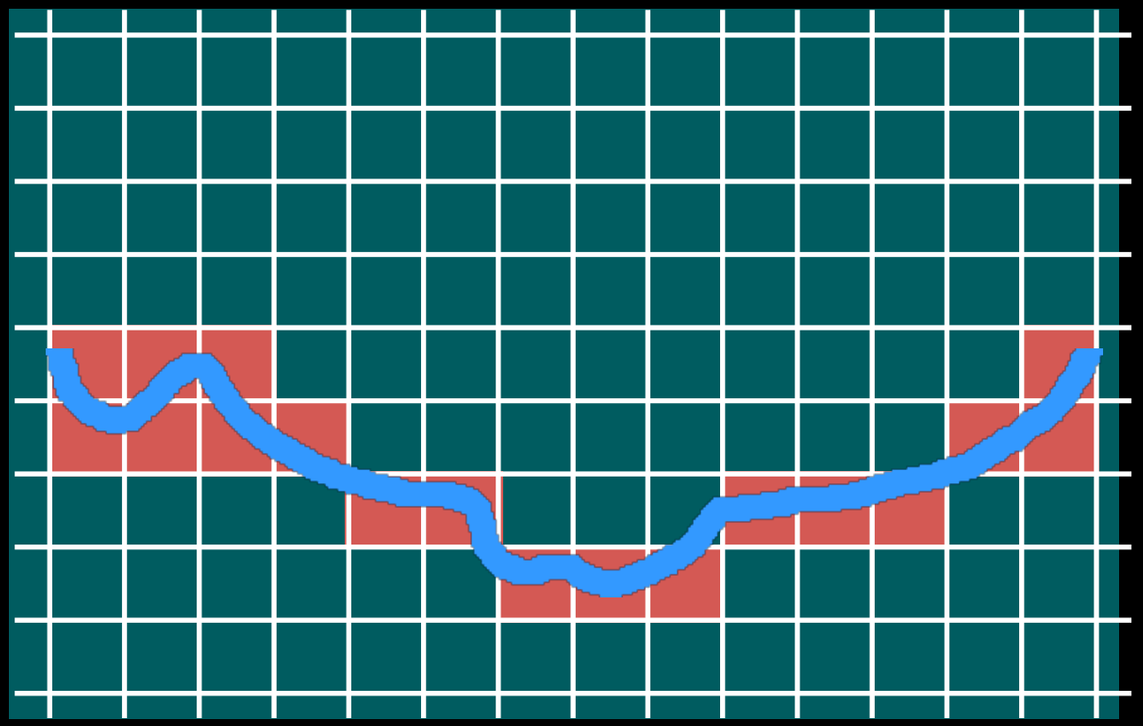


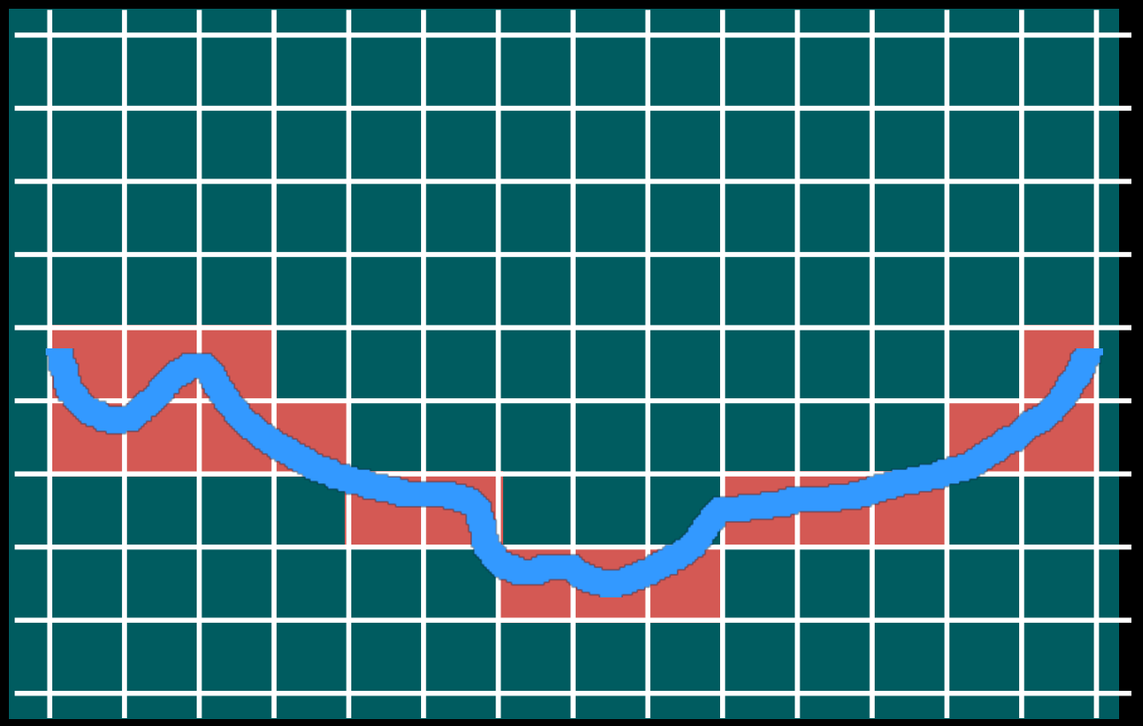



Bad match

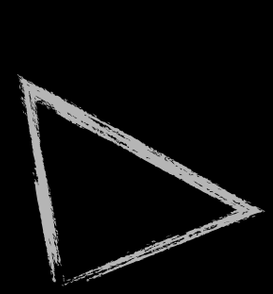

Good match

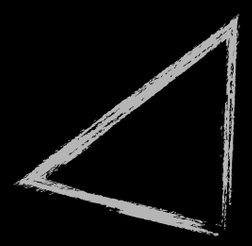


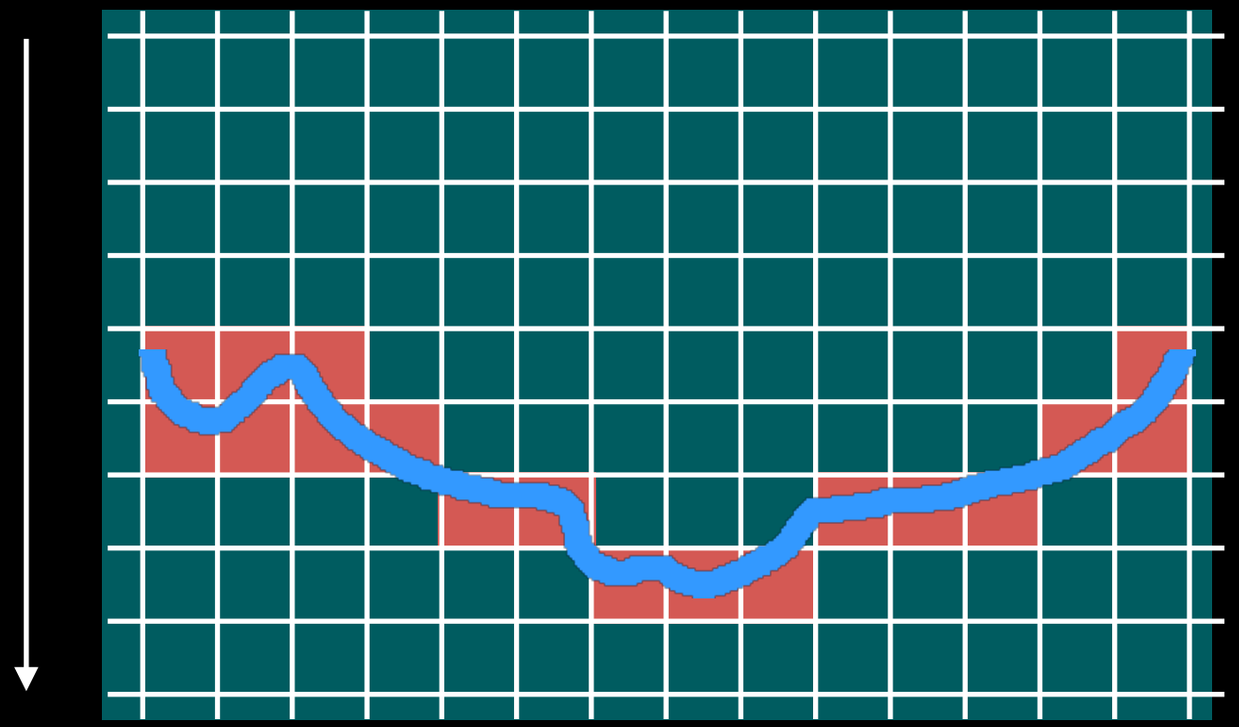




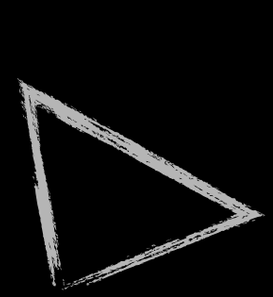
How can we compute this volume efficiently?

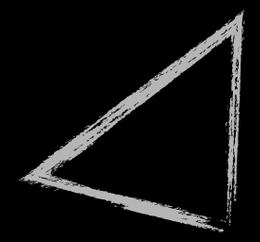


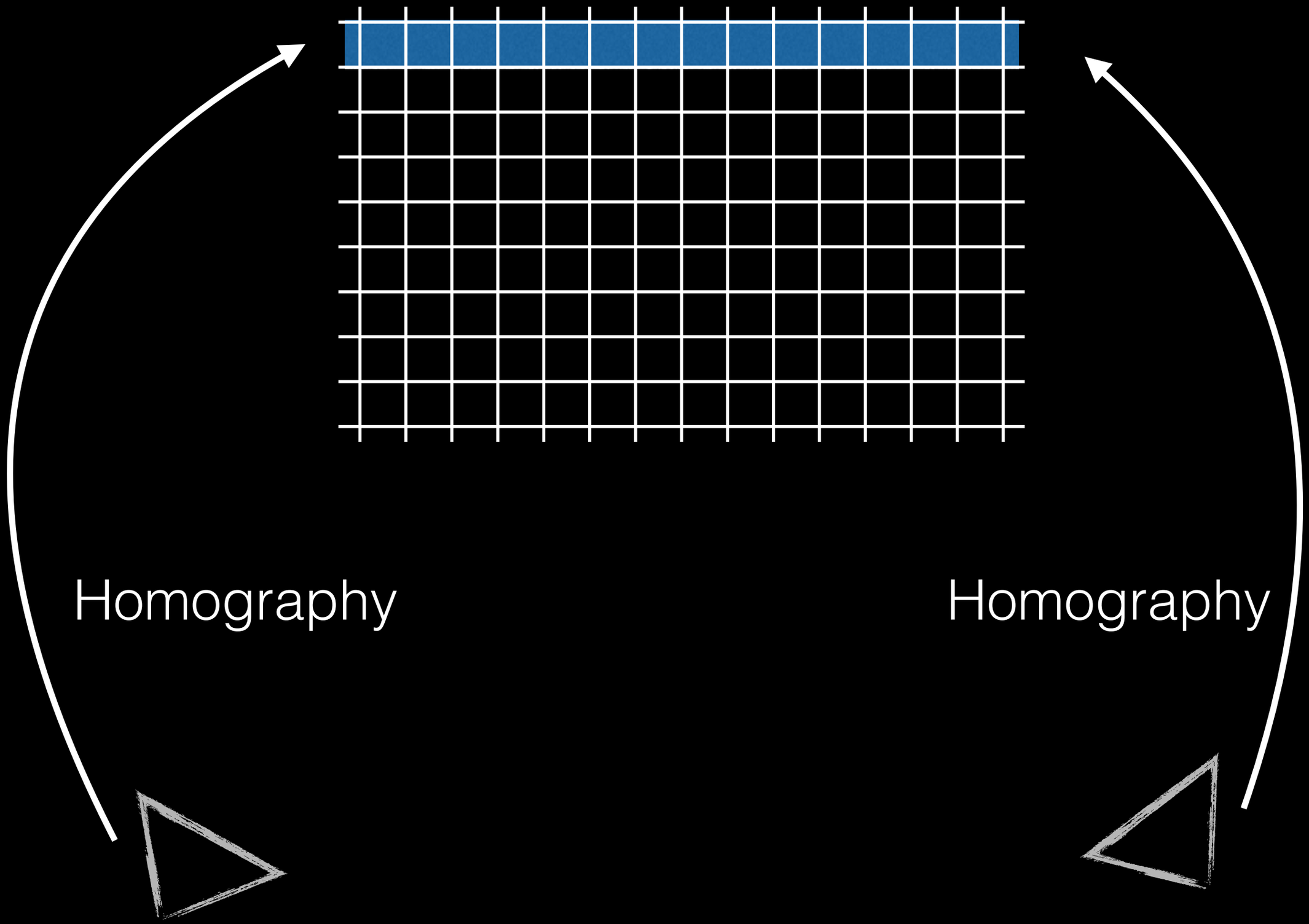




Compute one plane at a time.







Homography

Homography

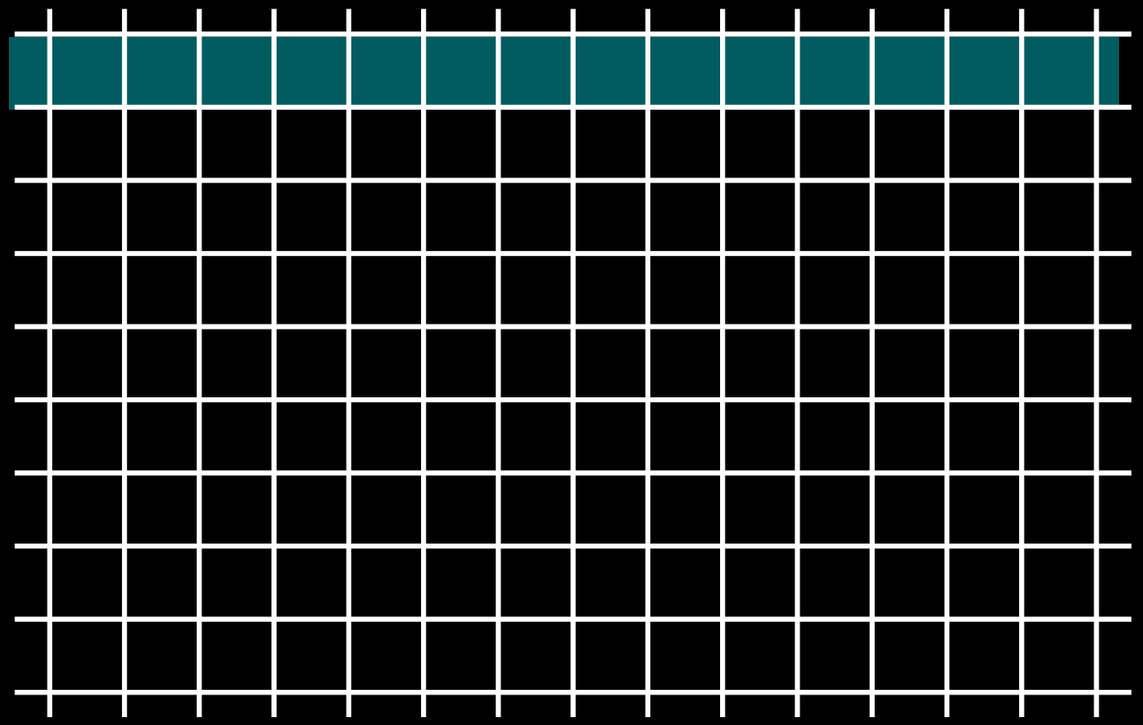
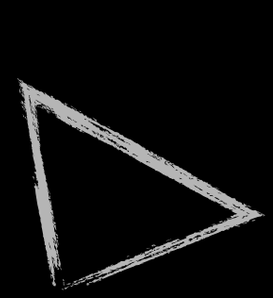
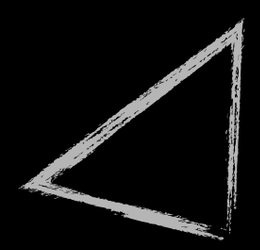
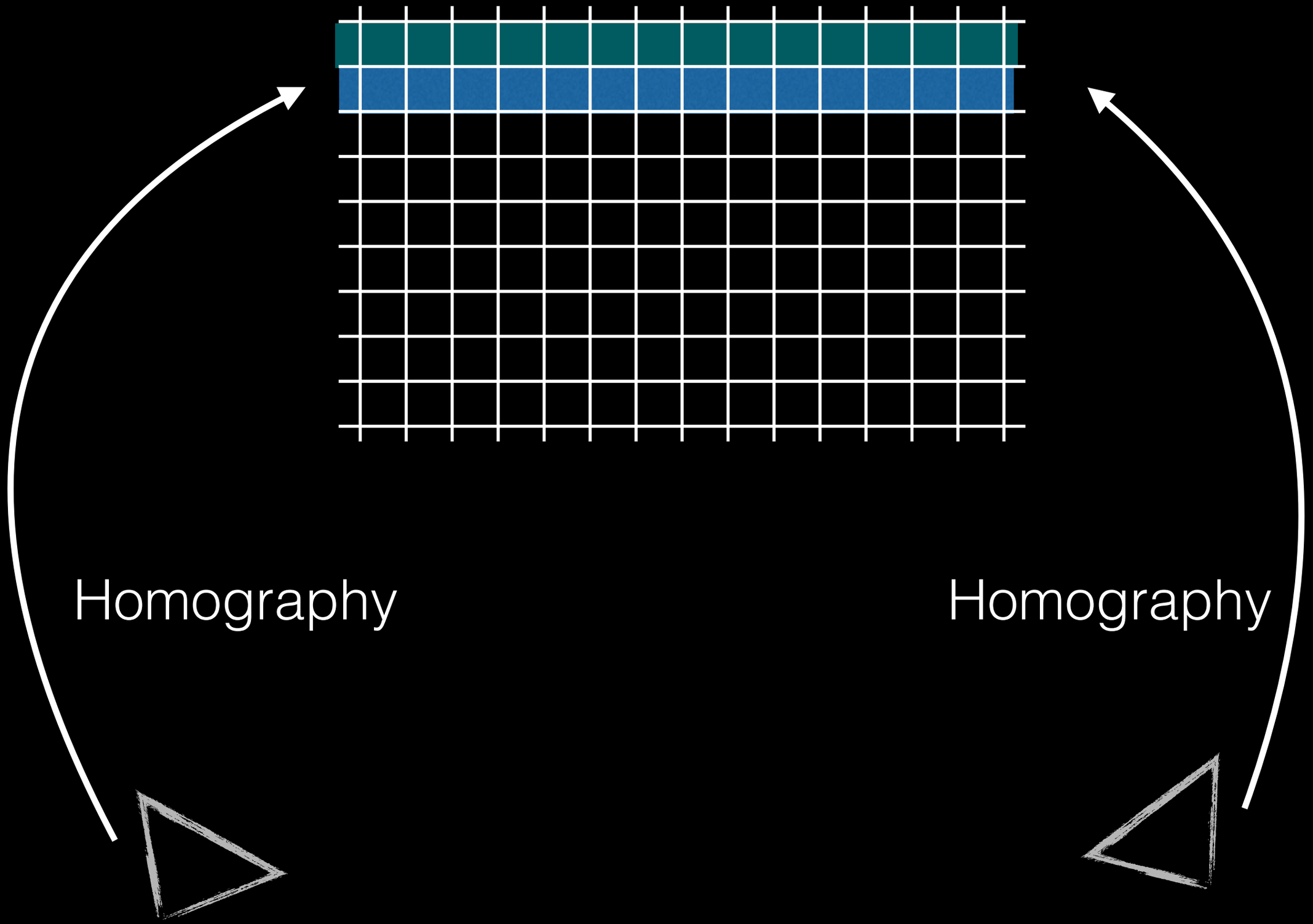


Photo-consistency
score per pixel







Homography

Homography

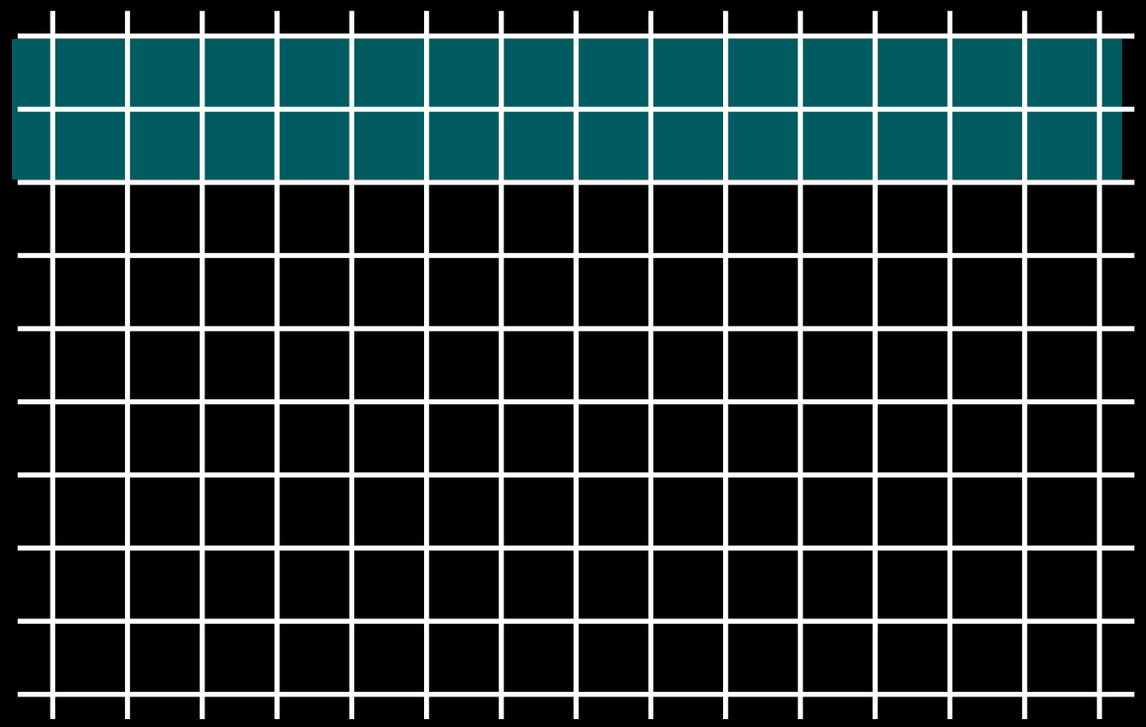
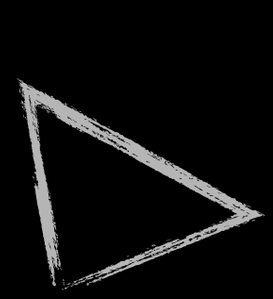
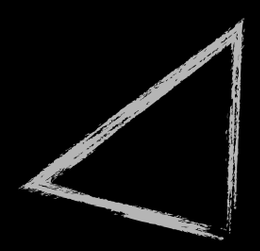
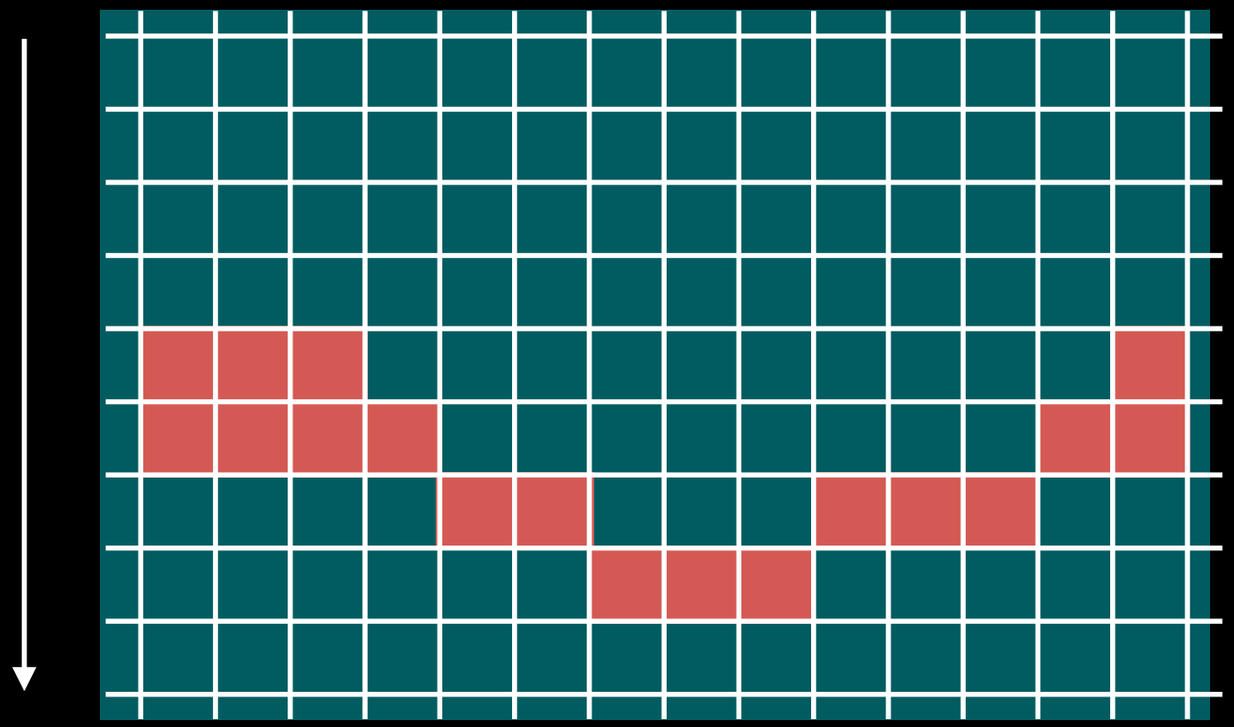


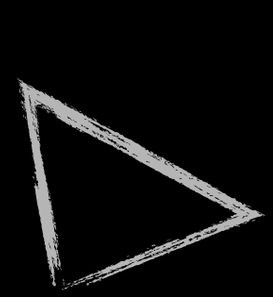
Photo-consistency
score per pixel

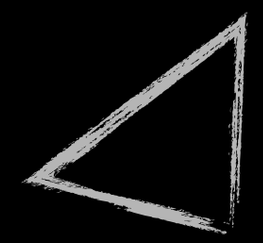


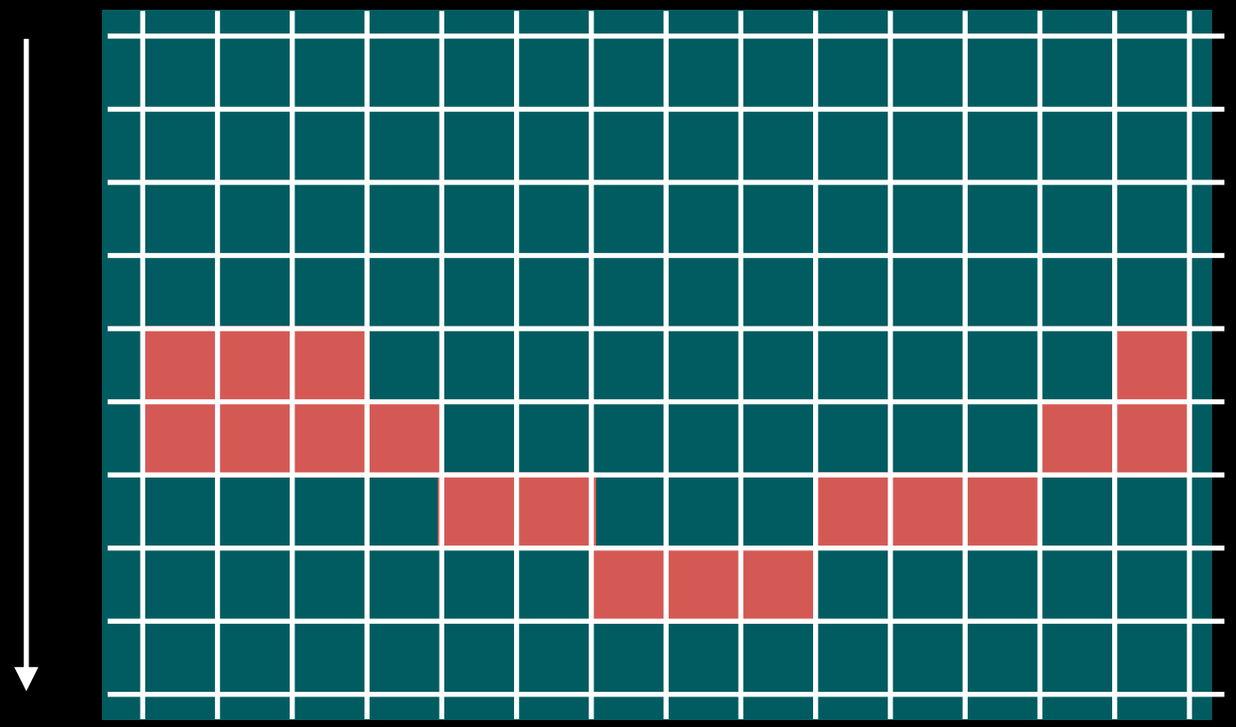




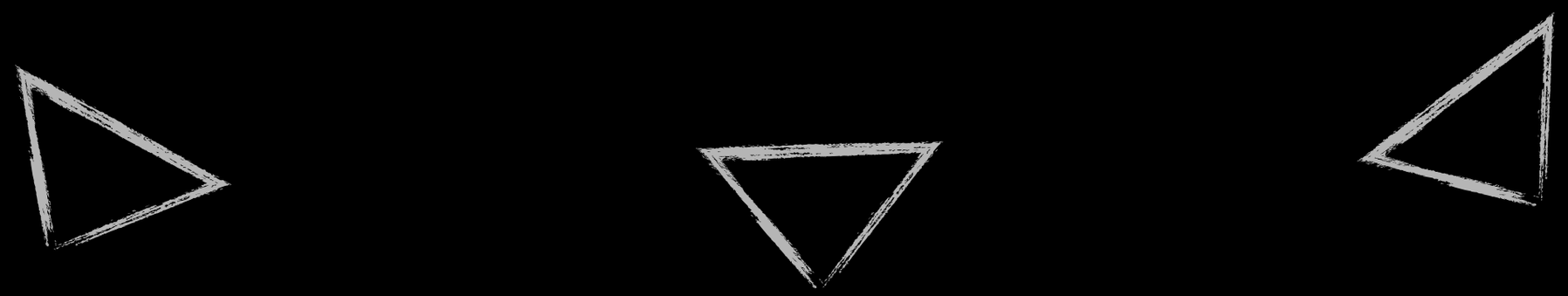
Compute the entire volume

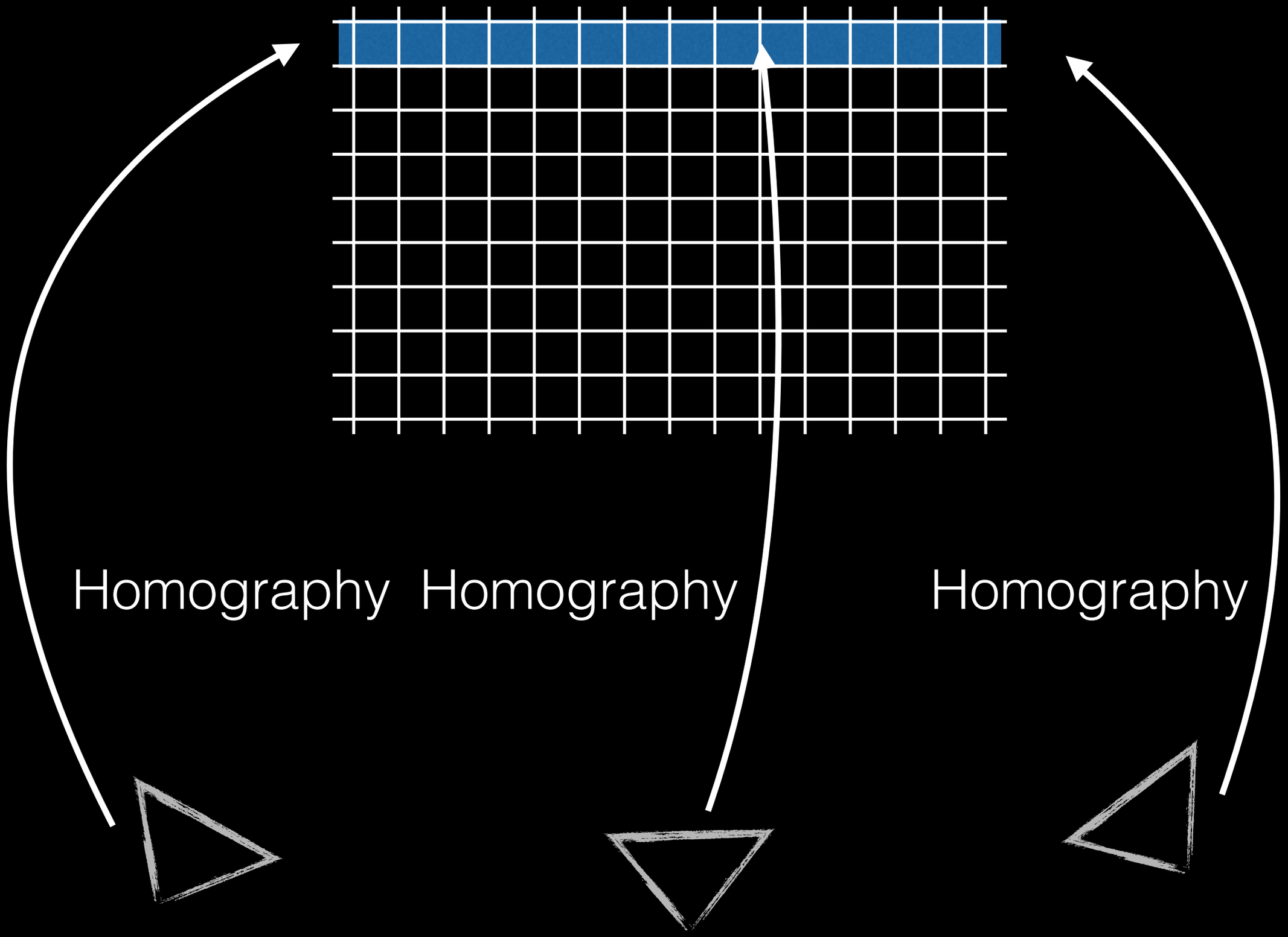


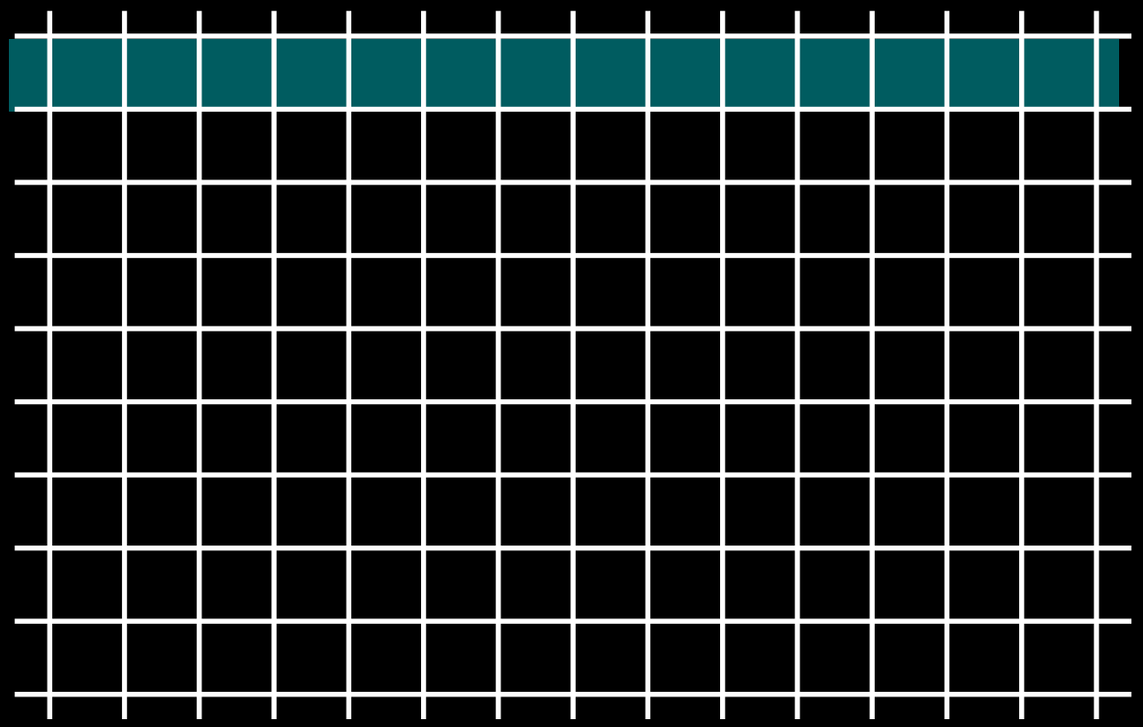




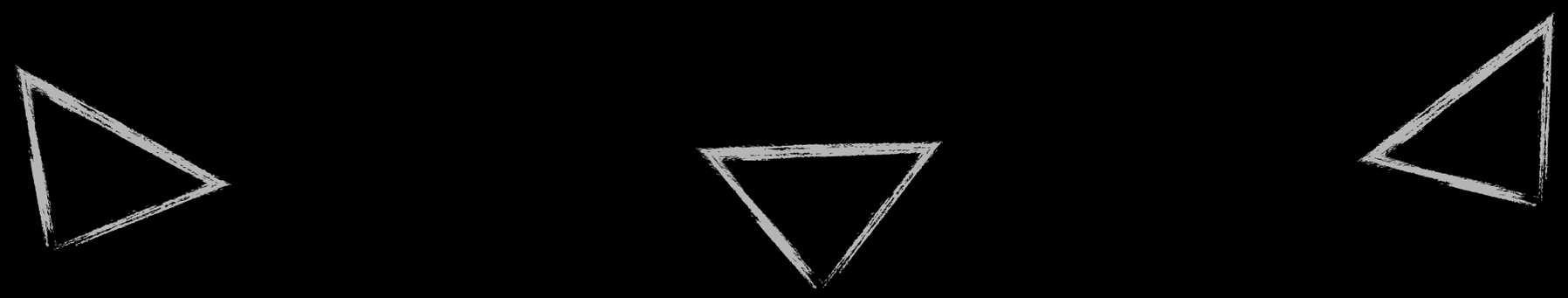
How to extend this to 3+ cameras?

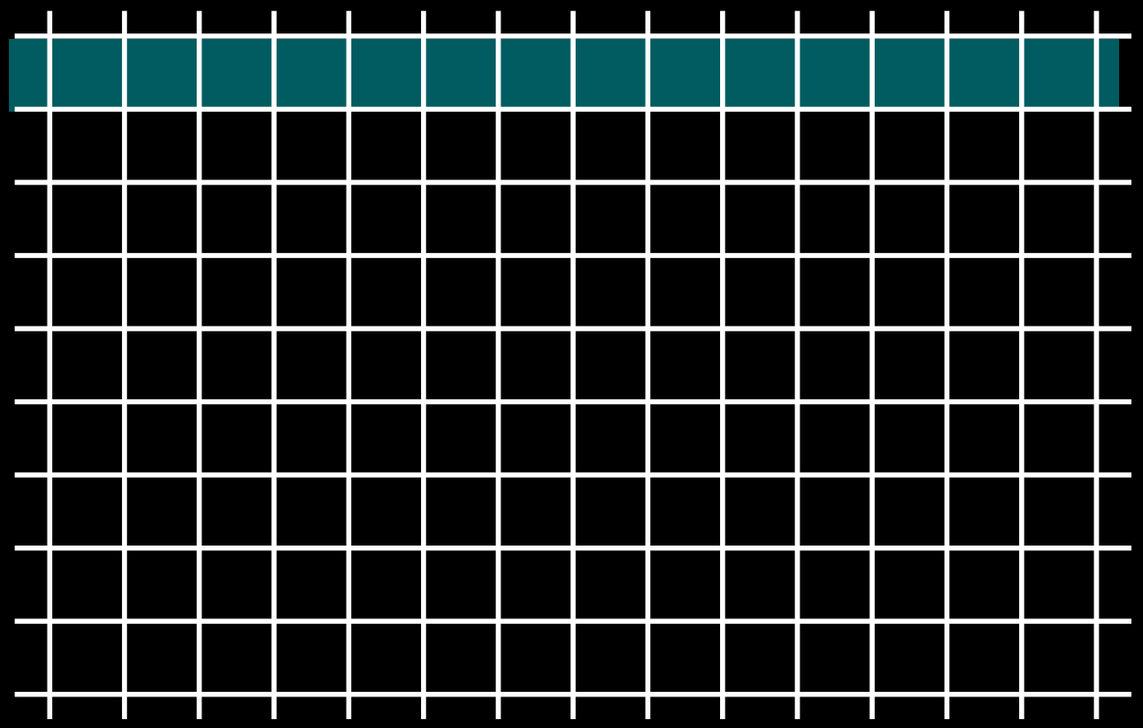



Homography
Homography
Homography



Global photo-consistency score per pixel

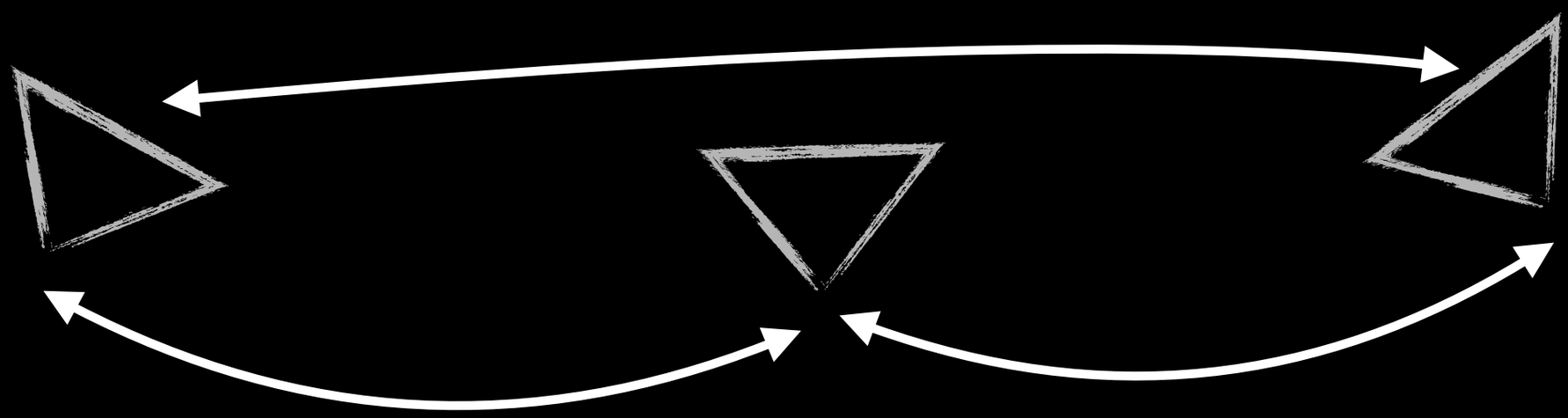


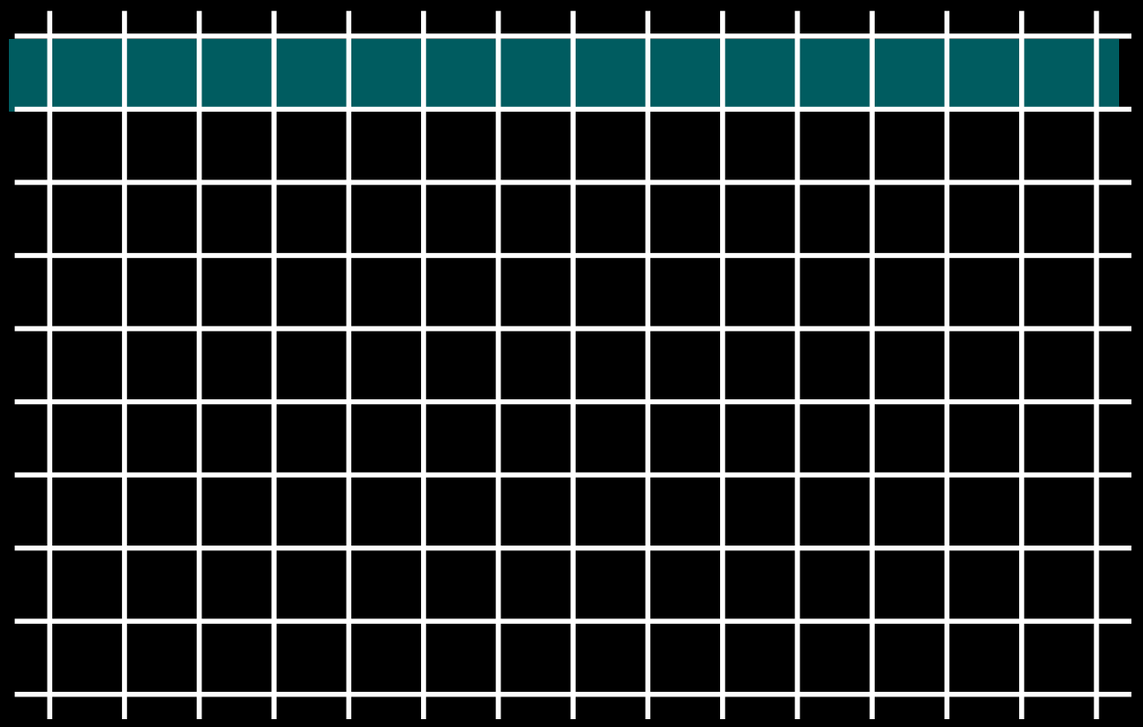


Global photo-consistency score per pixel

Examples:

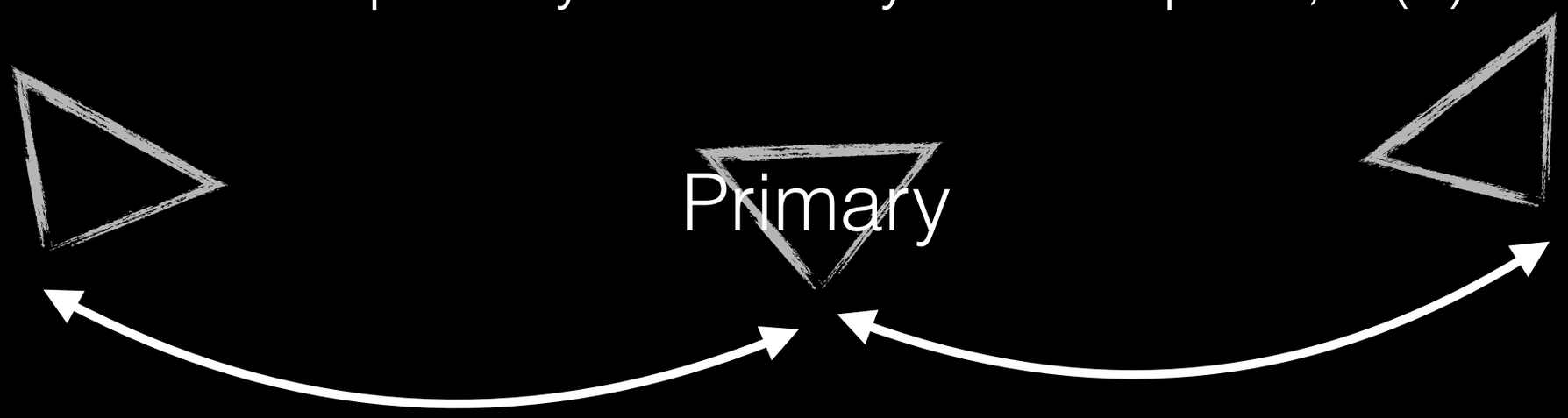
Median of all pairs, $O(n^2)$, photo-consistency scores





Global photo-consistency score per pixel

Examples:
Median of all pairs, $O(n^2)$, photo-consistency scores
Median of primary-secondary camera pairs, $O(n)$



Are depth maps enough?

Are depth maps enough?

Depends on your application…

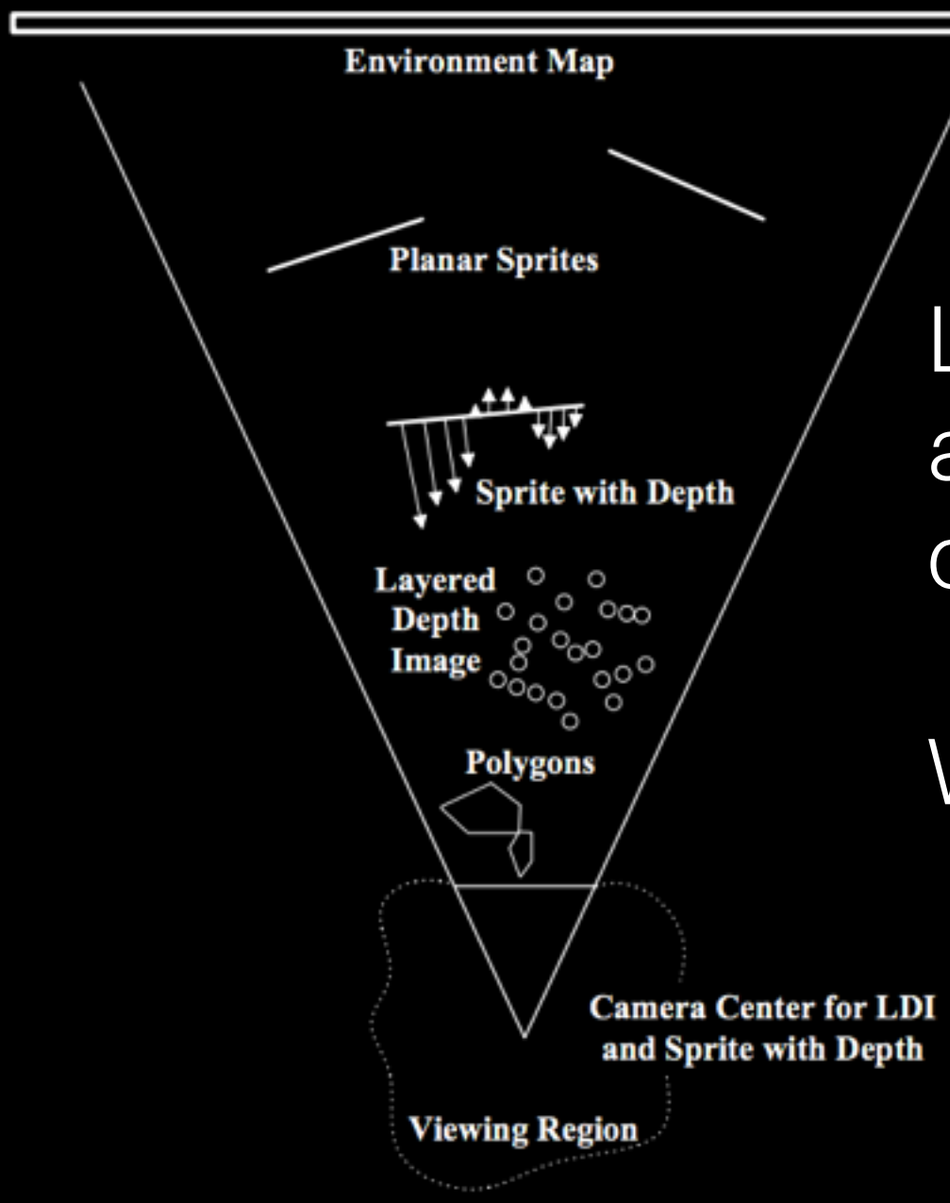


Figure 1 Different image based primitives can serve well depending on distance from the camera

Let's assume you want to render a scene containing your real-world object.

What should you capture?

Shade, et. al. SIGGRAPH 98.

Depends on your application…

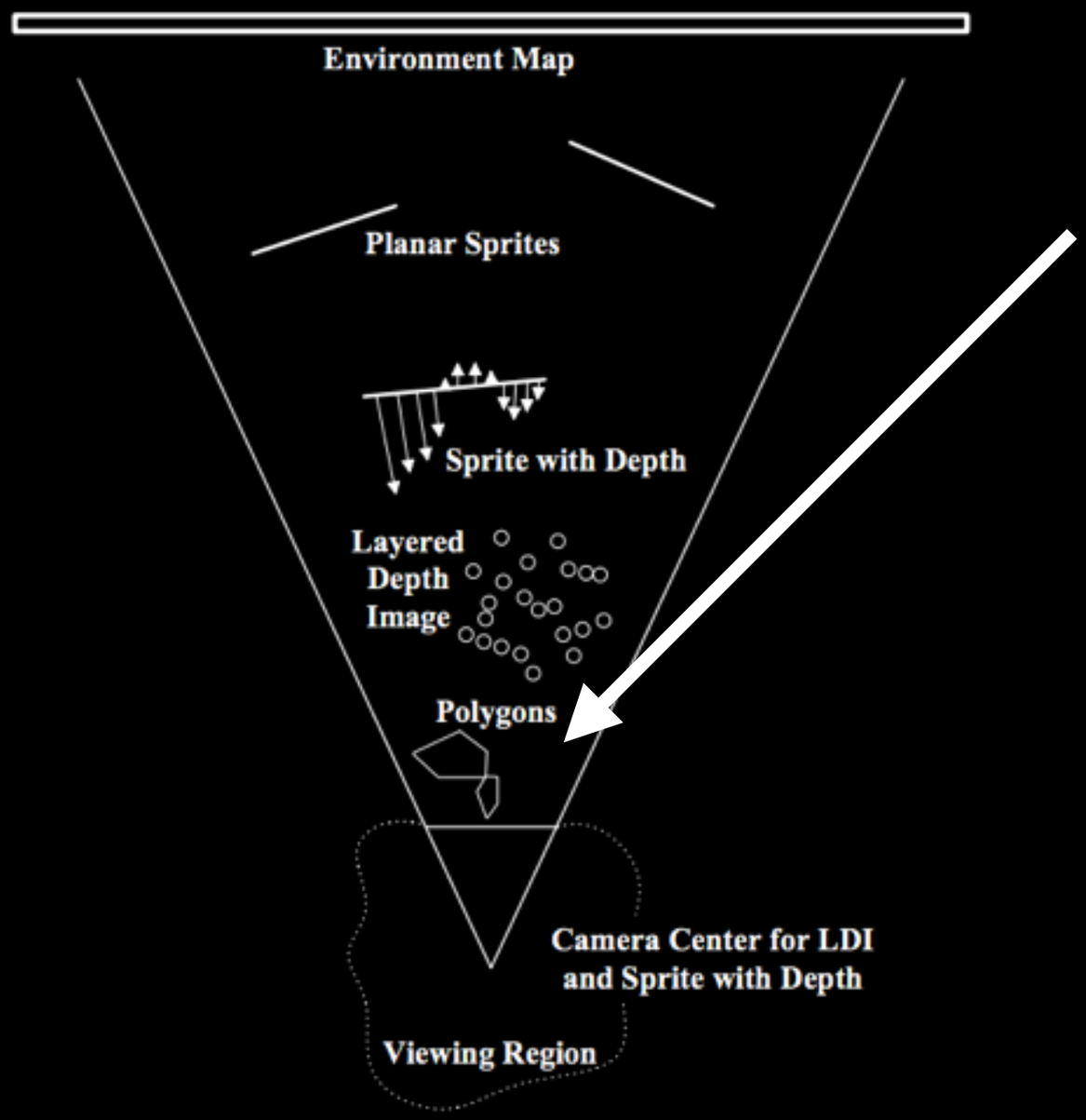


Figure 1 Different image based primitives can serve well depending on distance from the camera

The object is right in front of you and you want to look all around it.

Shade, et. al. SIGGRAPH 98.

Depends on your application…

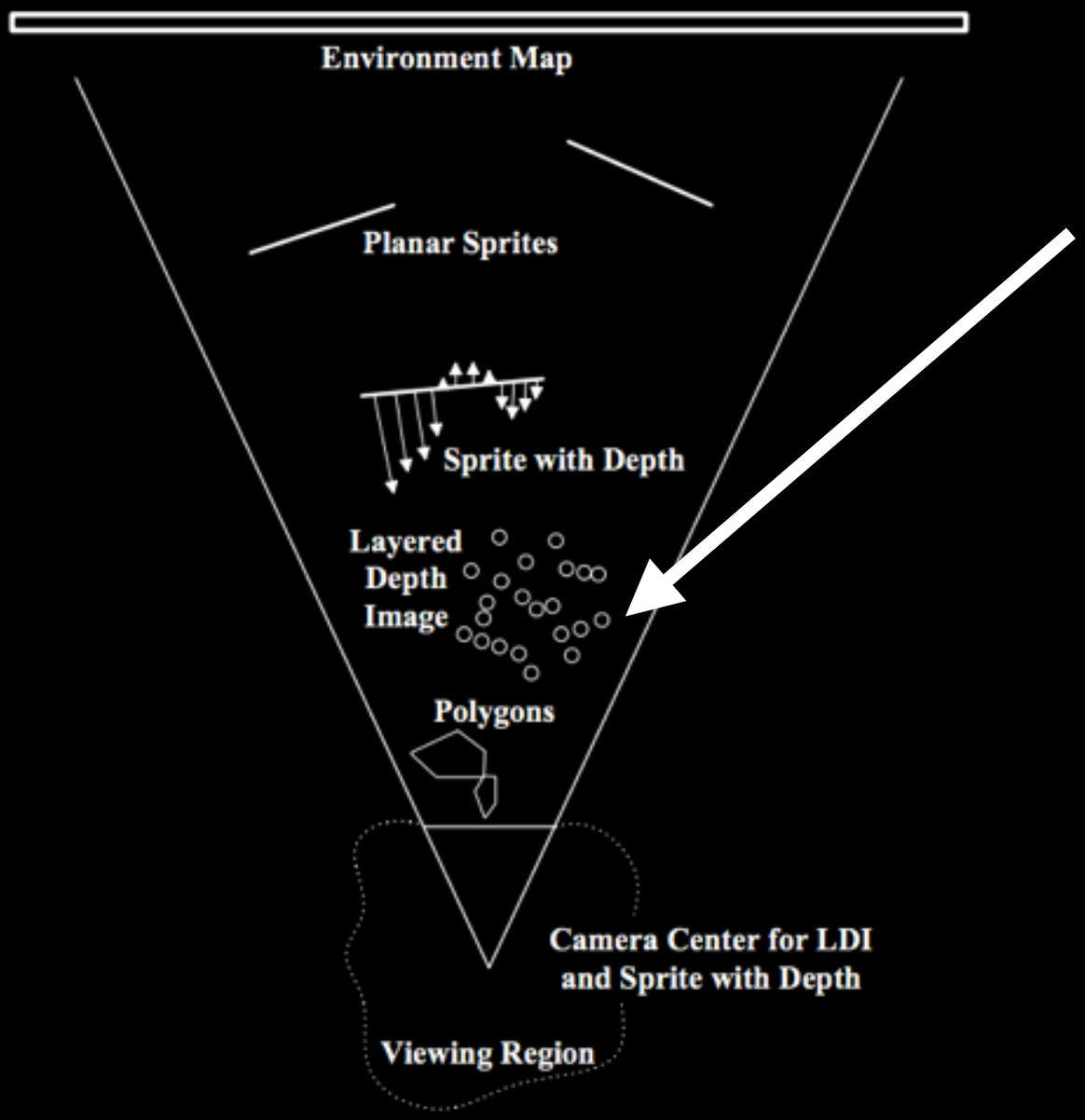


Figure 1 Different image based primitives can serve well depending on distance from the camera

Obvious dis-occlusions if a single depth map is used.

Shade, et. al. SIGGRAPH 98.


Dis-occlusion

Depends on your application…

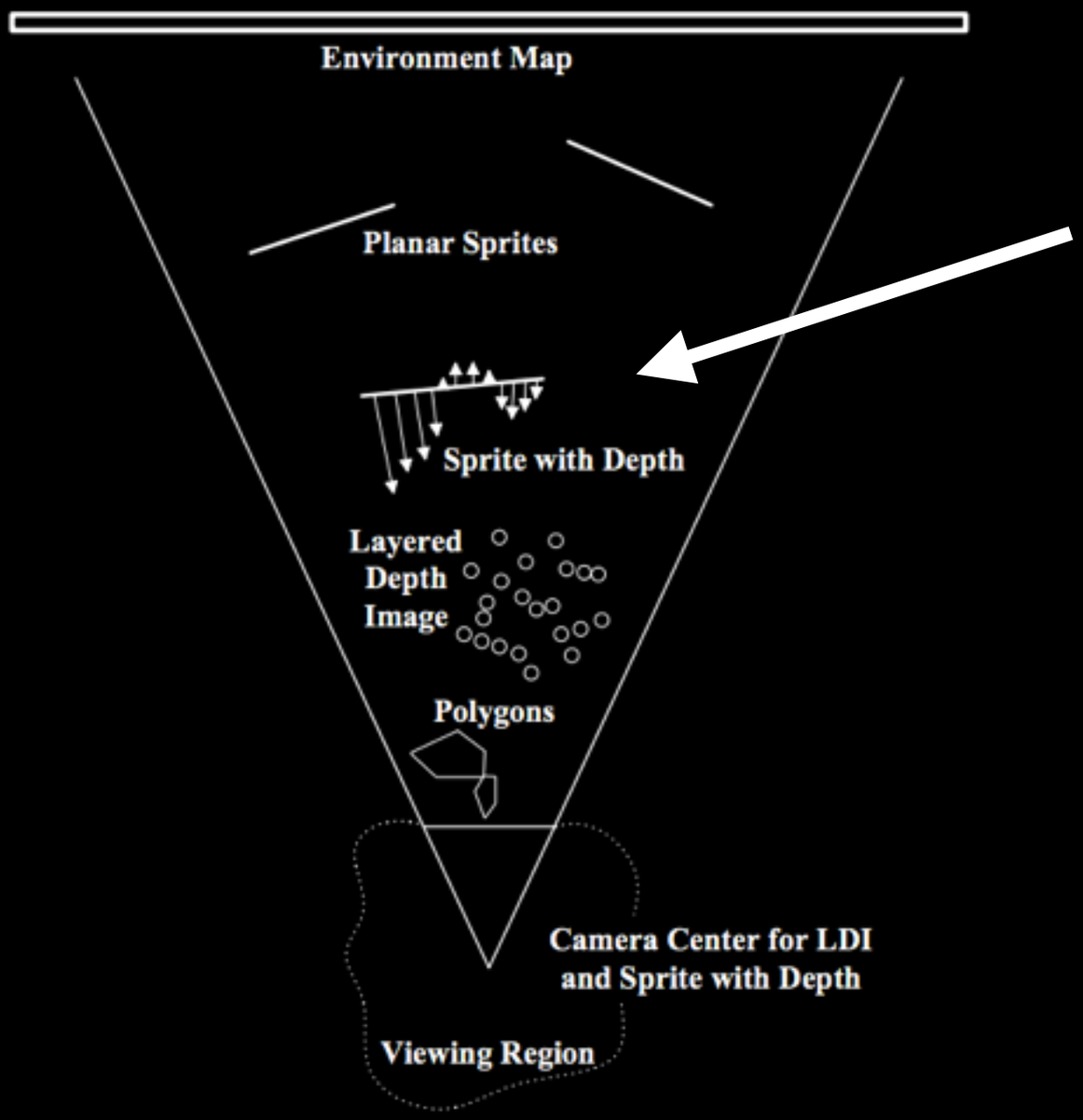


Figure 1 Different image based primitives can serve well depending on distance from the camera

Close enough that you can tell it shouldn't be flat, but **dis-occlusions** are minimal.

Shade, et. al. SIGGRAPH 98.

Depends on your application…

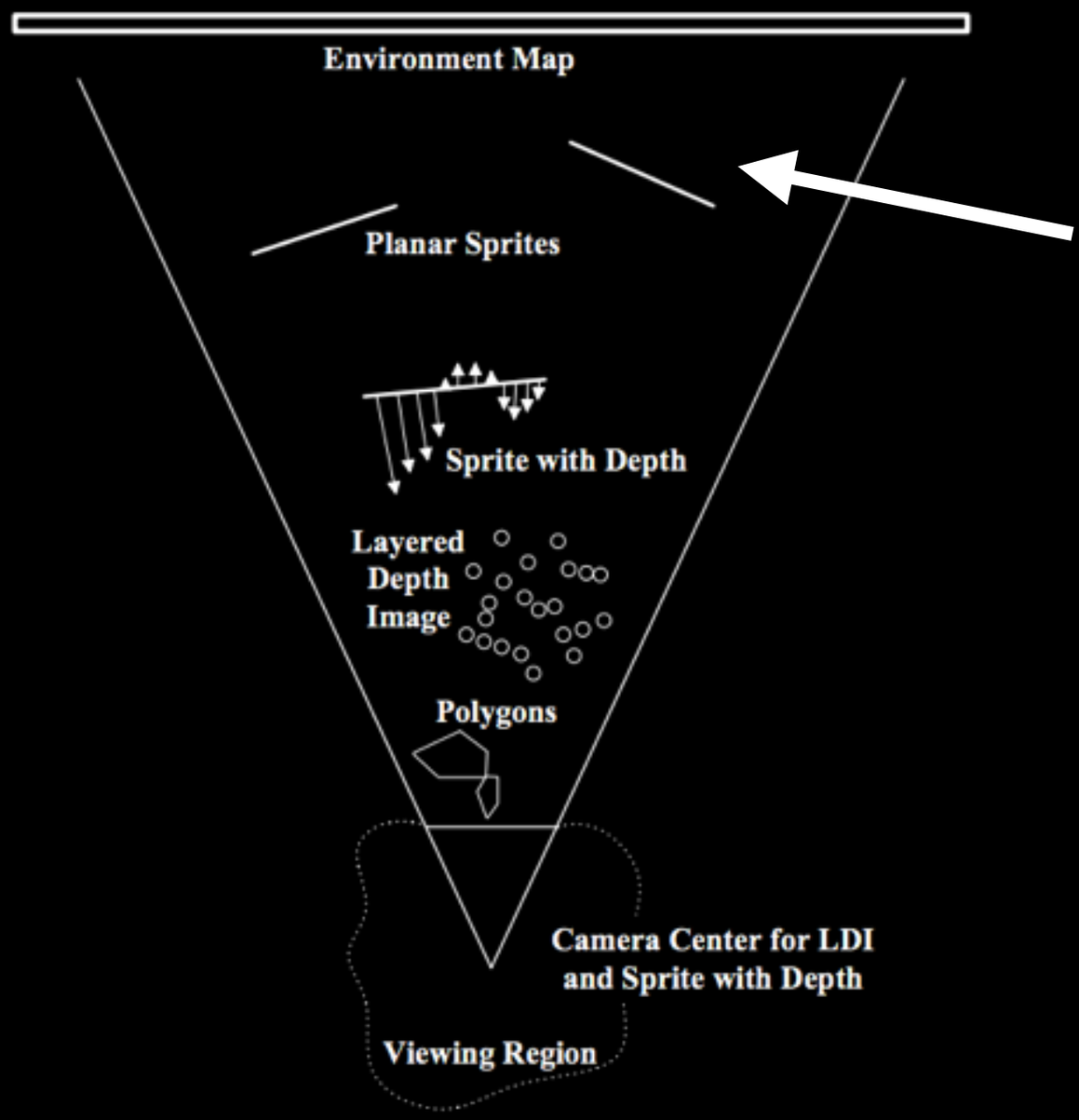


Figure 1 Different image based primitives can serve well depending on distance from the camera

Far enough away that a plane can approximate it.

Shade, et. al. SIGGRAPH 98.

Depends on your application…

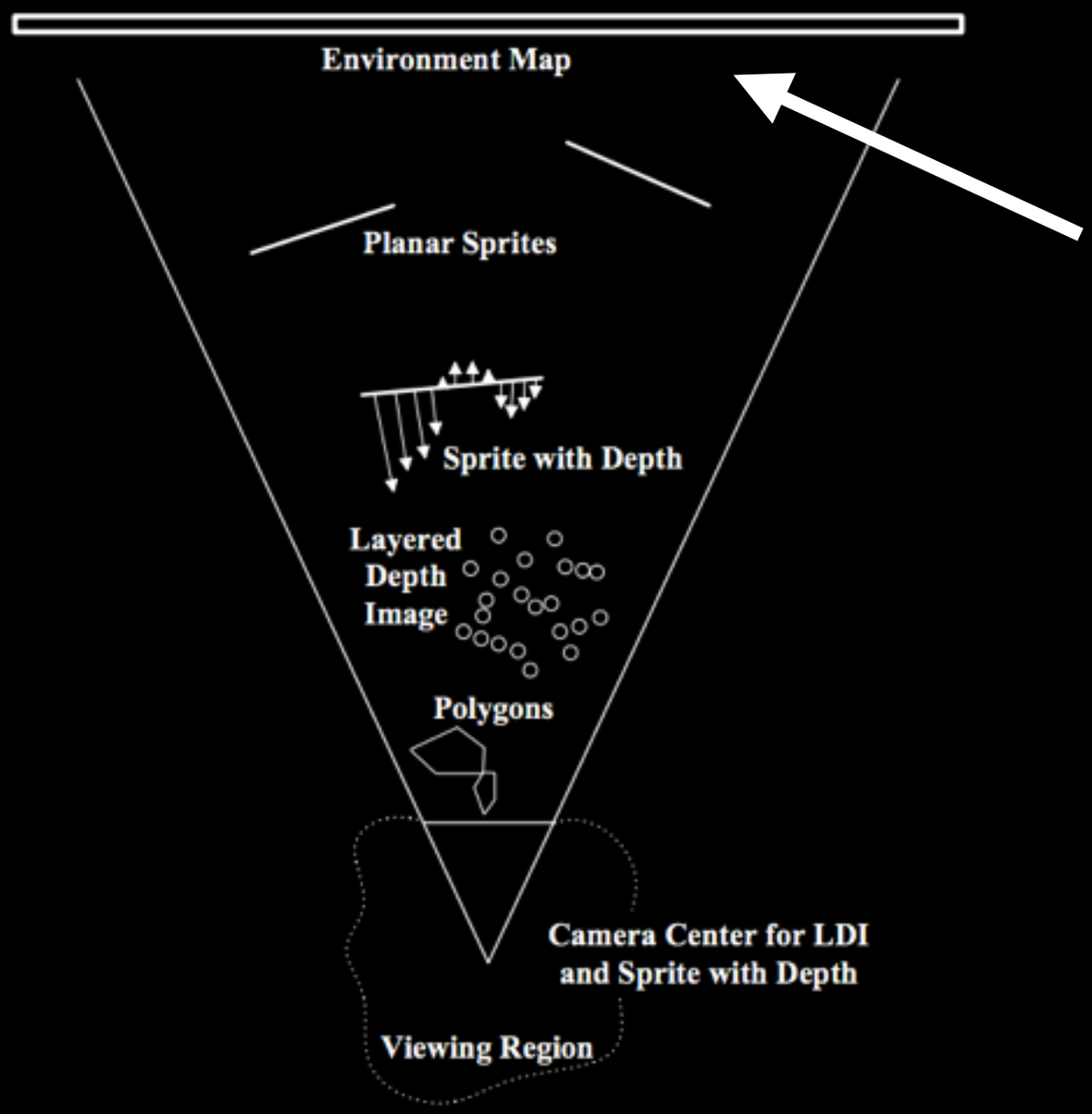


Figure 1 Different image based primitives can serve well depending on distance from the camera

So far away it's effectively at infinity.

Shade, et. al. SIGGRAPH 98.

Depends on your application…

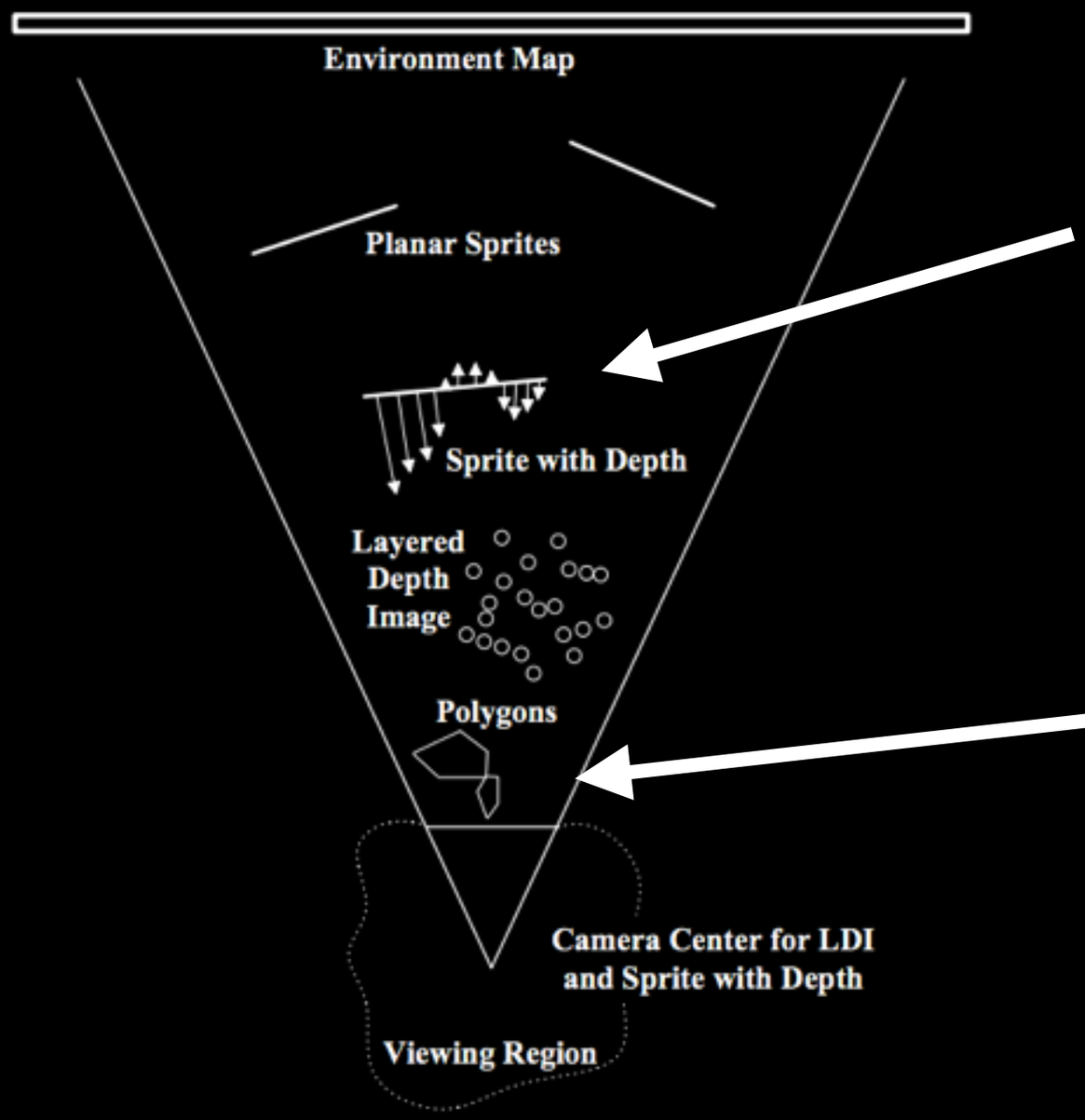
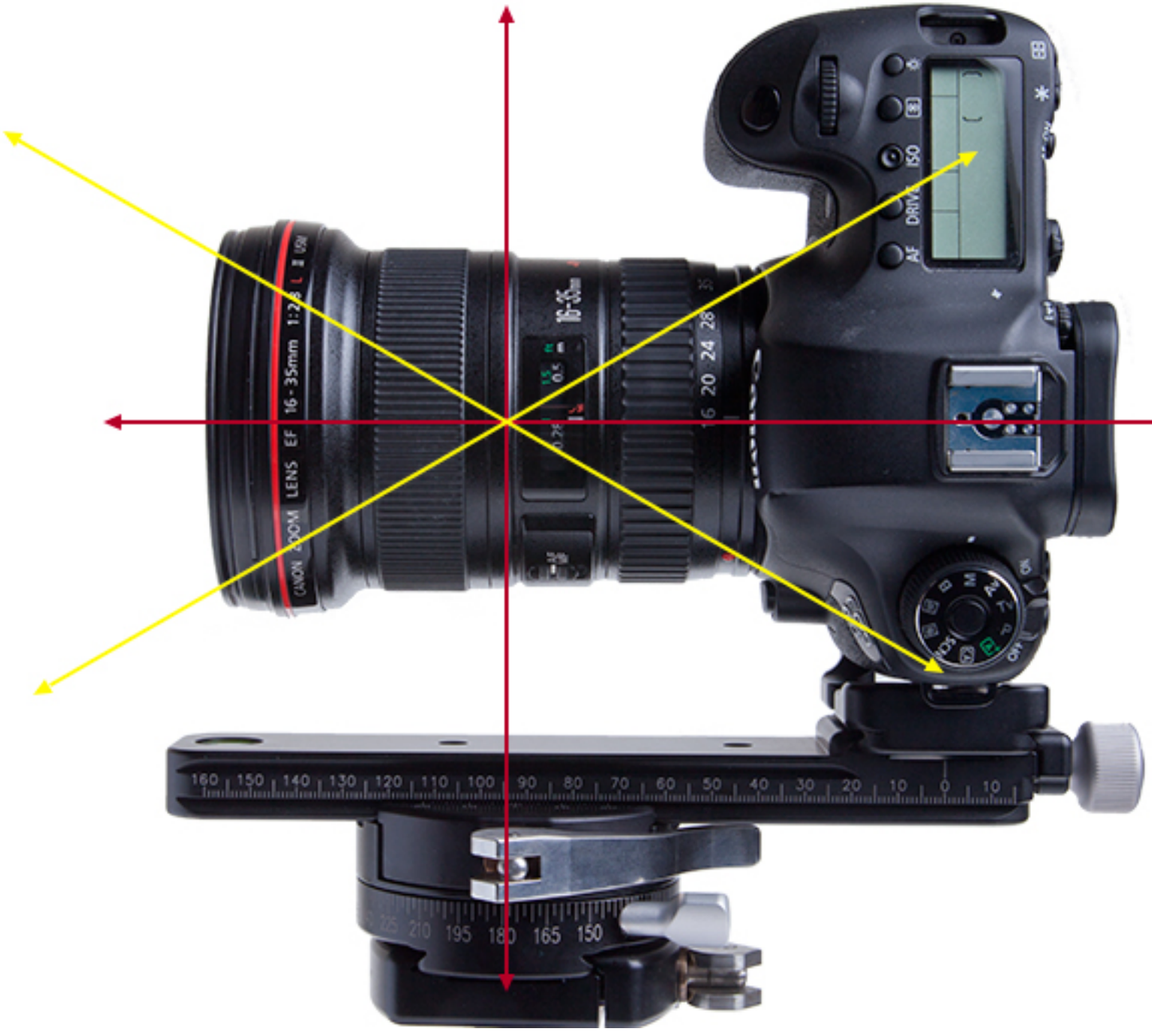


Figure 1 Different image based primitives can serve well depending on distance from the camera

The method you know gets you here.

Now we'll talk about how to get here.

Shade, et. al. SIGGRAPH 98.


CANON ZOOM LENS EF 16-35mm 1:2.8 L USM
16-35mm
AF
DRIVE
ISO
160 150 140 130 120 110 100 90 80 70 60 50 40 30 20 10 0 10
225 210 195 180 165 150