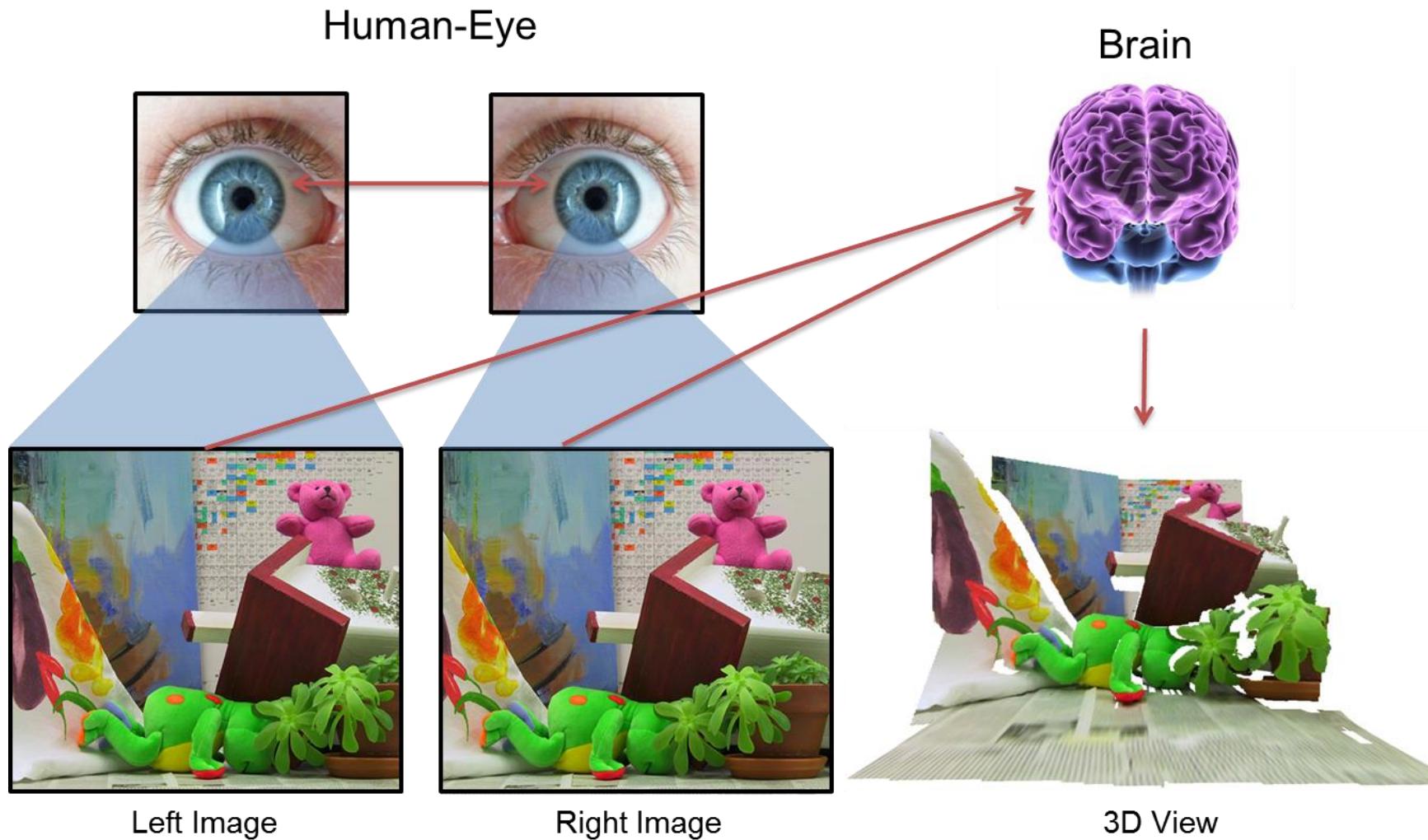


Stereo Matching

Quang-Vinh Dinh
Ph.D. in Computer Science

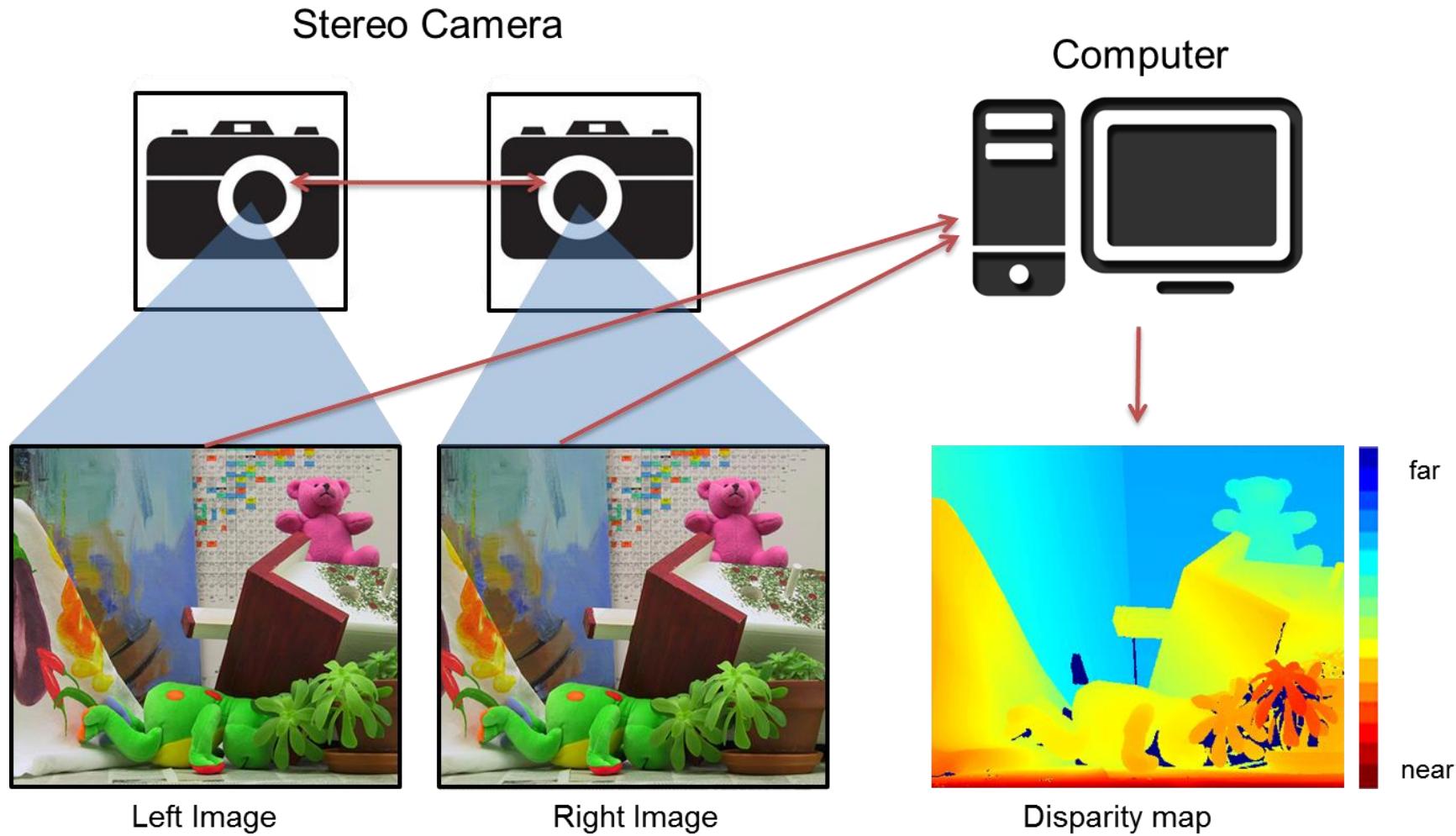
Stereo Matching Algorithm

❖ Human perception for 3D information



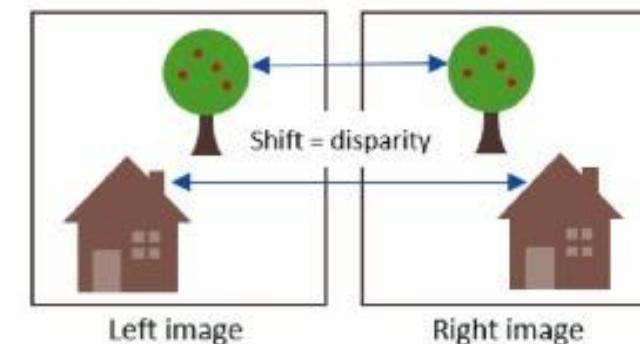
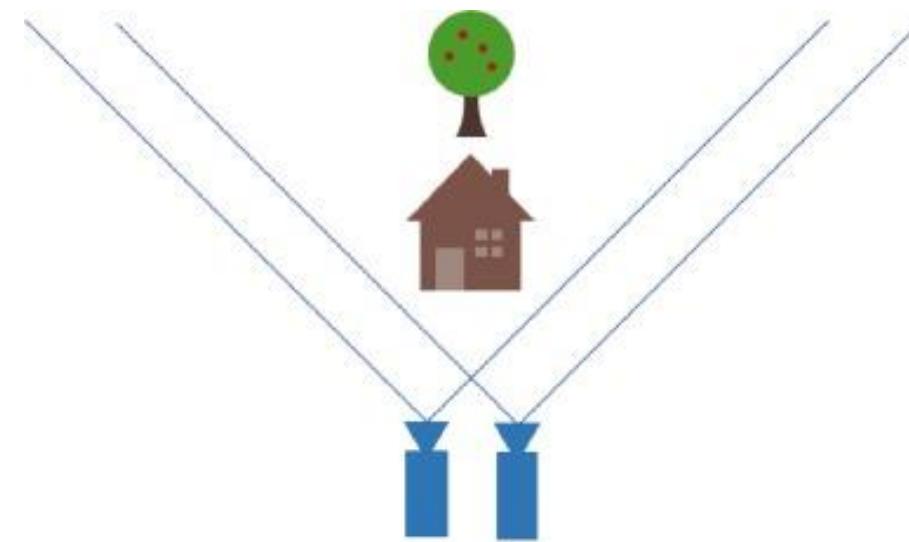
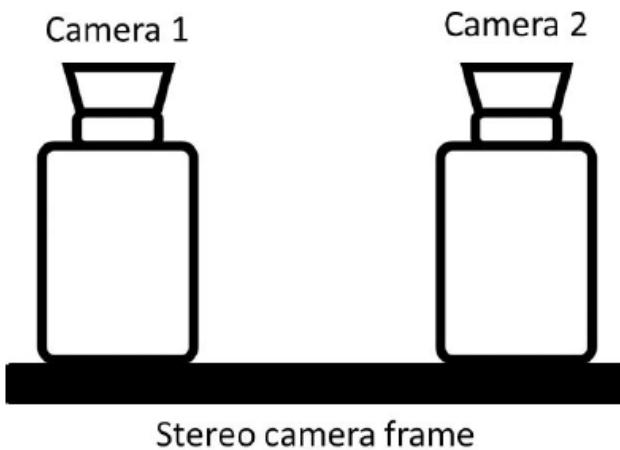
Stereo Matching Algorithm

- ❖ We generate 3D information using disparity maps obtained from stereo matching algorithm



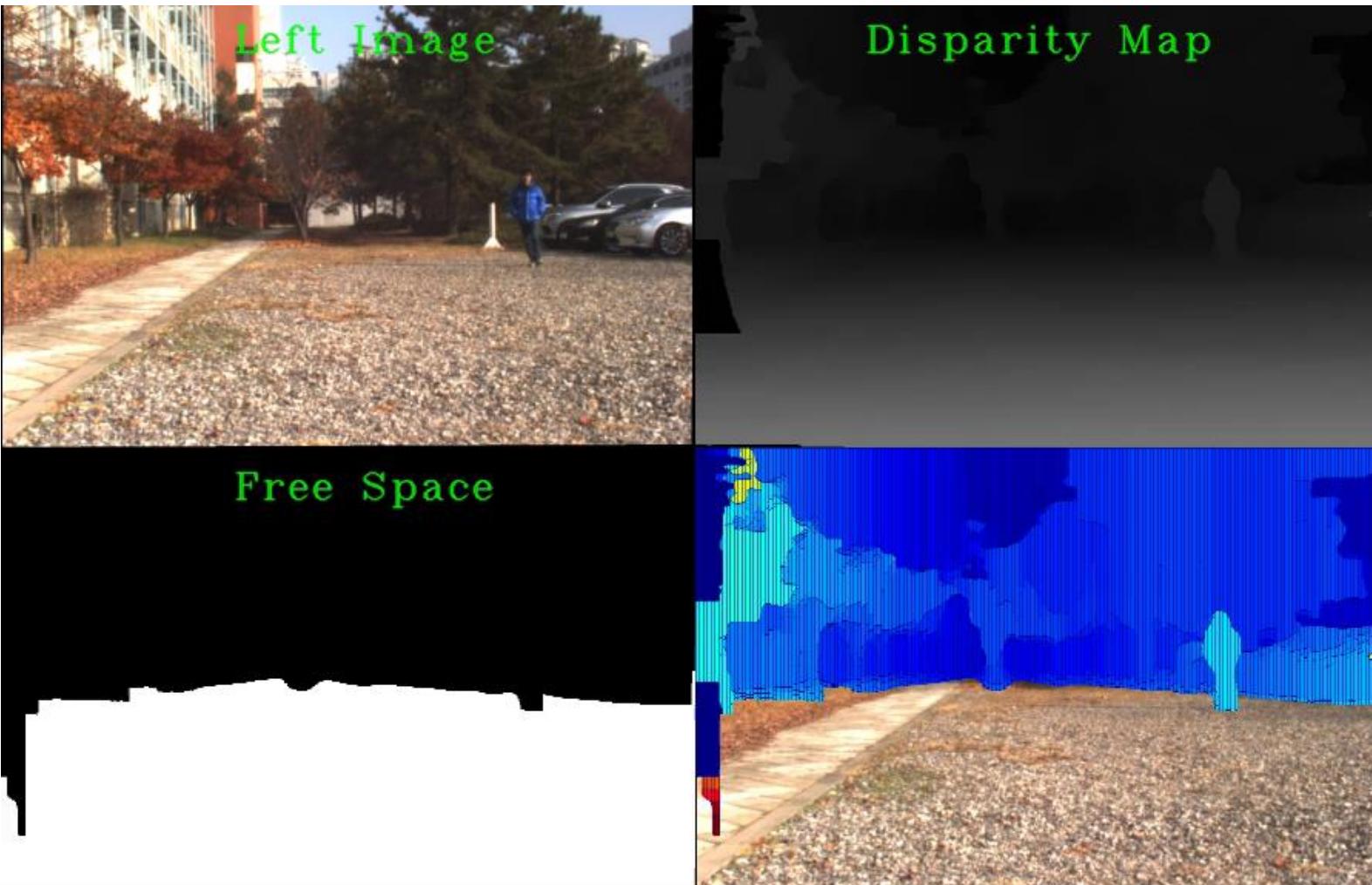
Stereo Matching Algorithm

- ❖ Computer stereo vision is the extraction of 3D information from digital images
- ❖ Stereo cameras



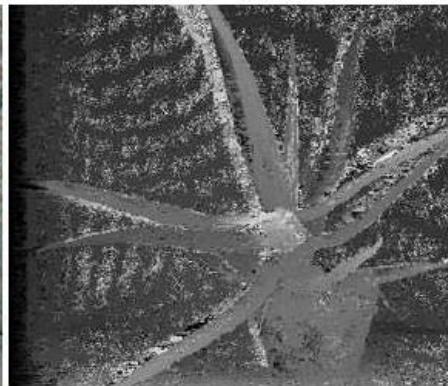
Stereo Matching Algorithm

❖ Applications



Stereo Matching Algorithm

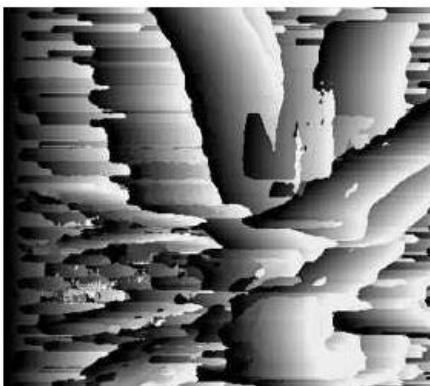
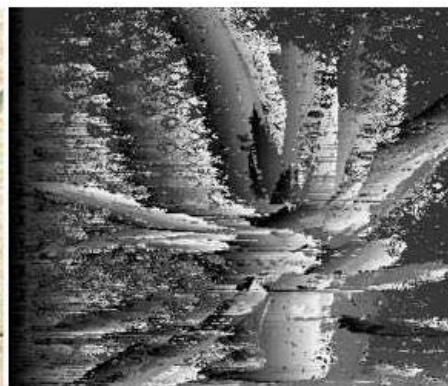
❖ Challenges in stereo matching



Normal stereo pair

ADCensus

SAD



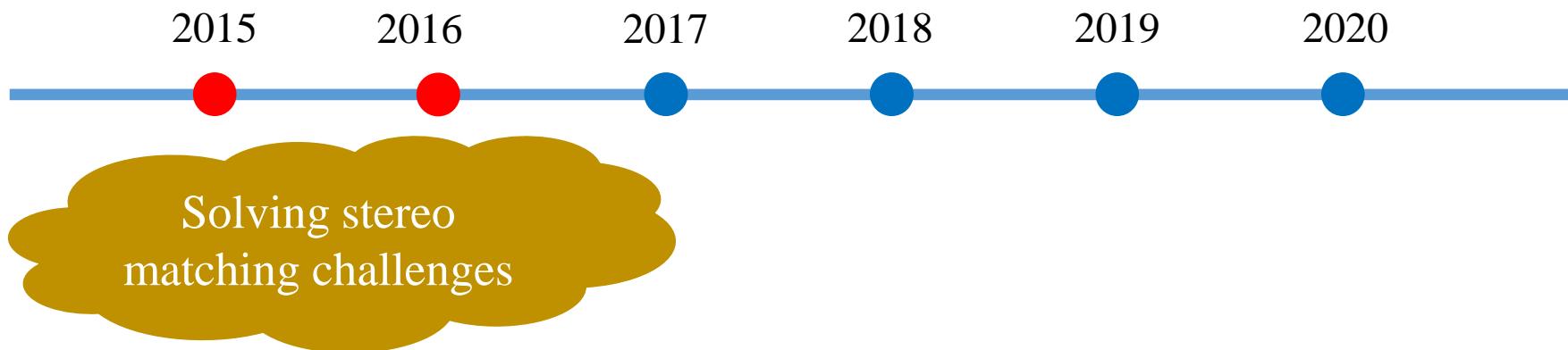
Stereo pair with different illumination

ADCensus

SAD

Stereo Matching Algorithm

❖ Research from 2015 to 2016



Stereo Matching Algorithm

- ❖ Research from 2015 to 2016

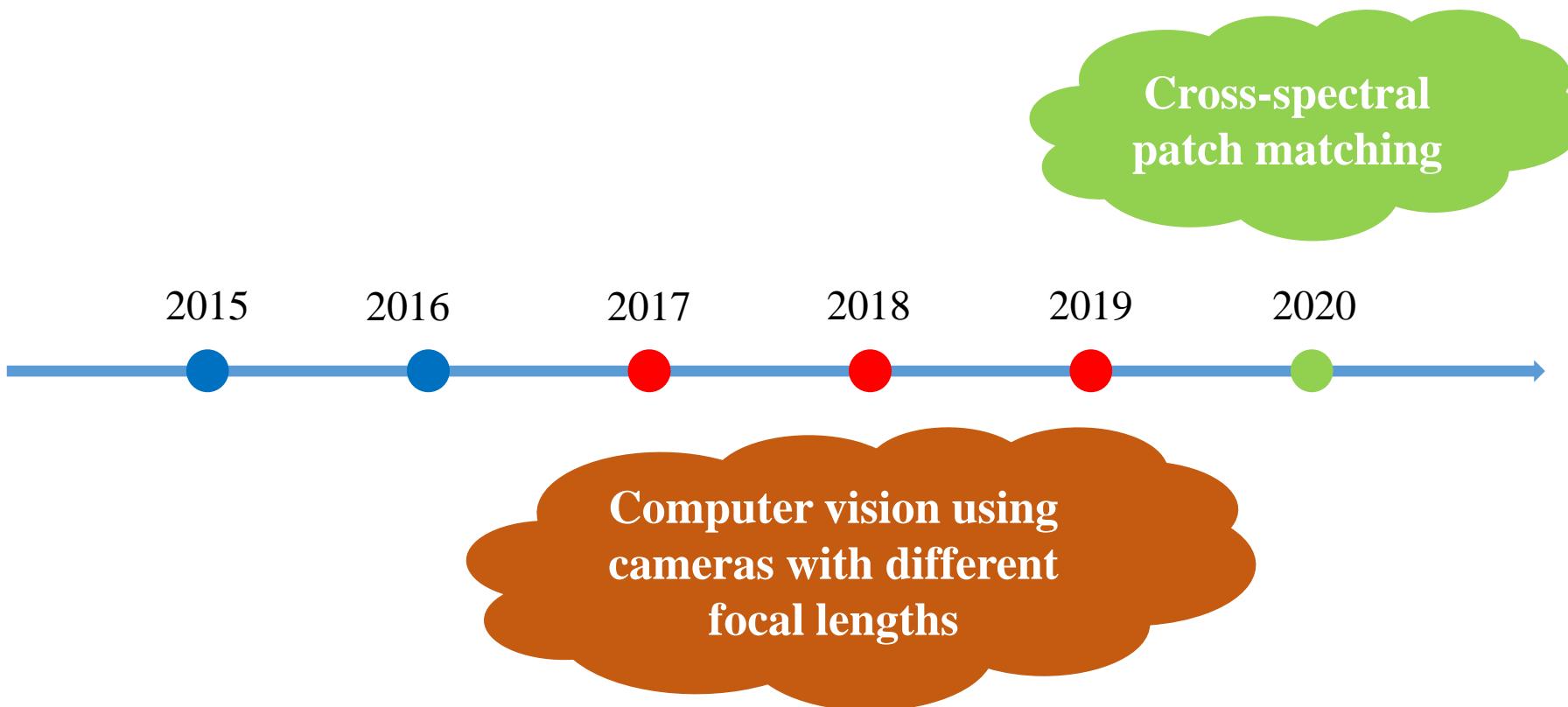
Efficient descriptor

Different
illumination

Using monotonic and non-linear relationship measurement

Stereo Matching Algorithm

❖ Research from 2017 to now

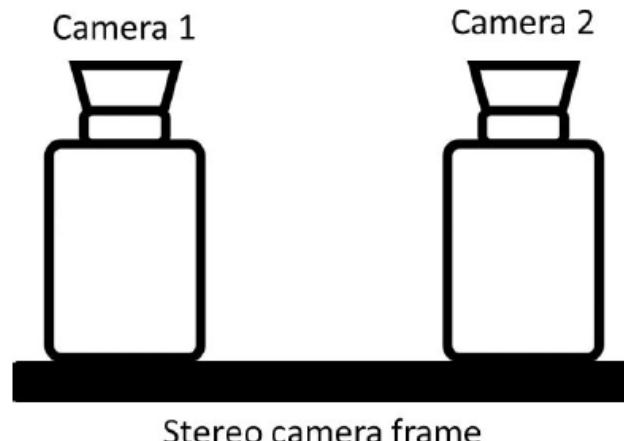


Outline

- Introduction to research topics
- Stereo matching
- Computer vision using cameras with different focal lengths
 - Motivation
- Cross-spectral patch matching

Computer vision using different focal lengths

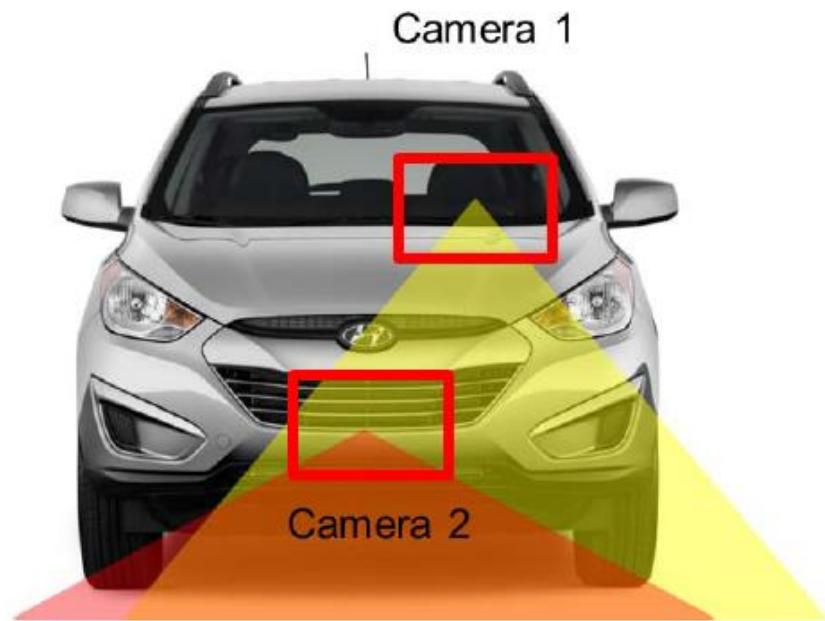
❖ Normal stereo cameras



Stereo cameras

Computer vision using different focal lengths

❖ Motivation

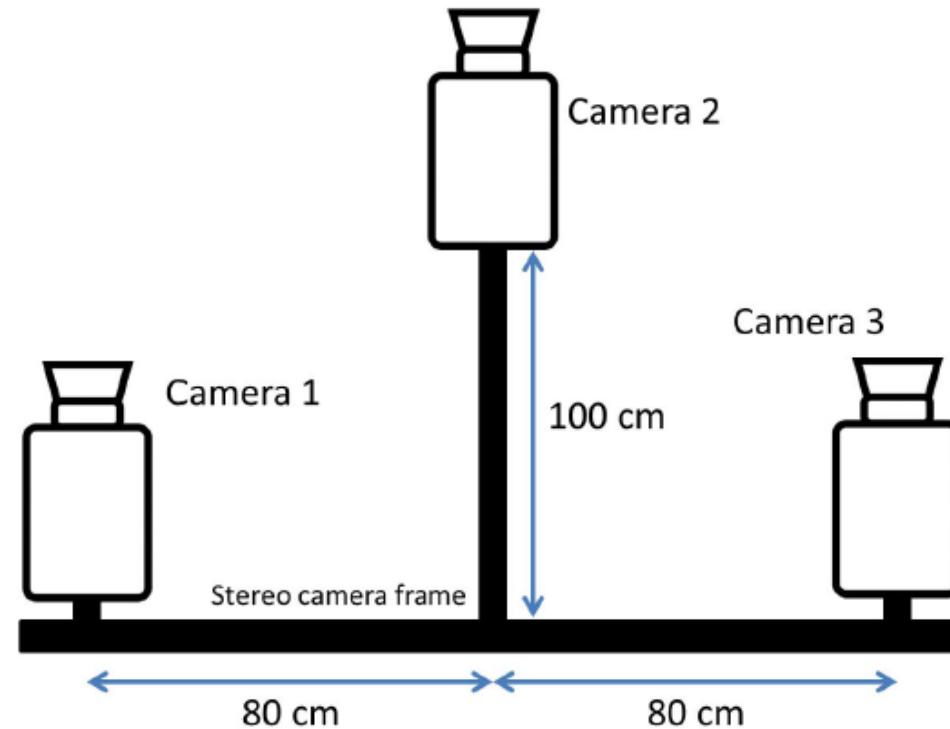


Computer vision using different focal lengths

- ❖ A system setup

- ❖ Applications

- ❖ Image rectification
- ❖ Stereo matching
- ❖ Vehicle detection



Outline

- Introduction to research topics
- Stereo matching
- Computer vision using cameras with different focal lengths
 - Motivation
 - Image rectification
- Cross-spectral patch matching

Image Rectification

- ❖ Image rectification is a transformation process used to project images onto a common image plane

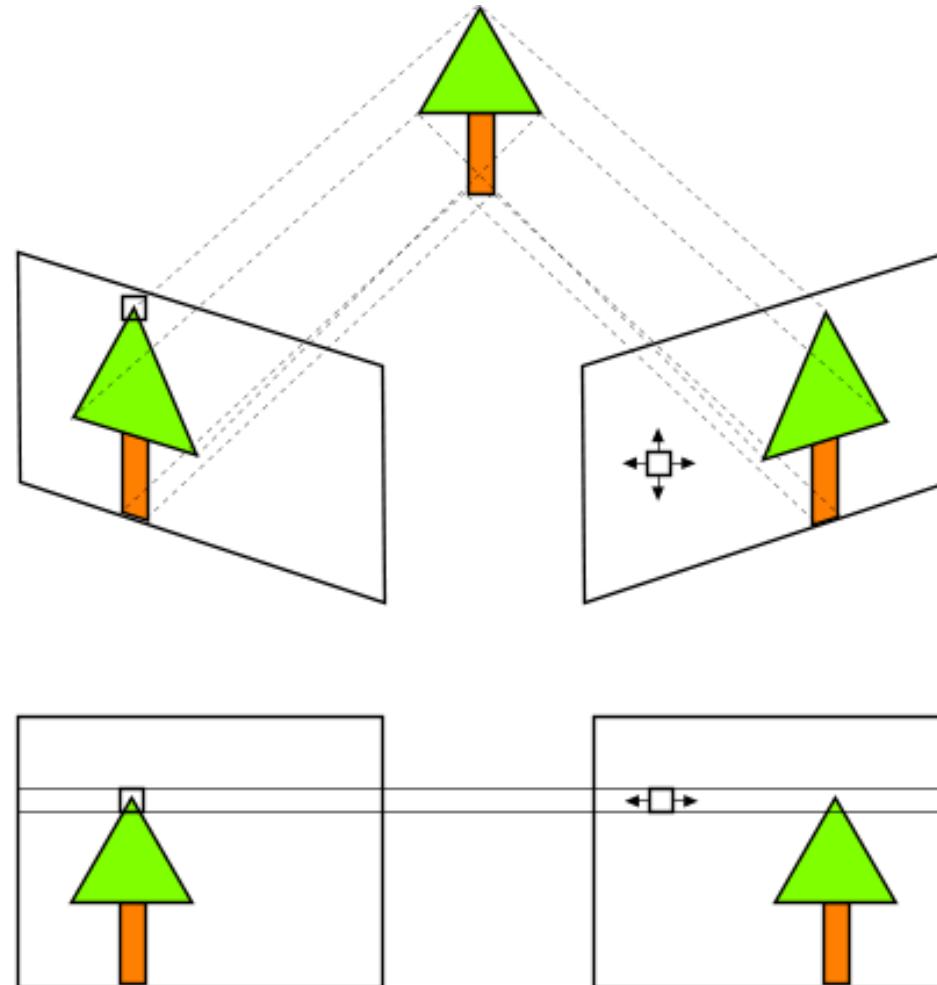
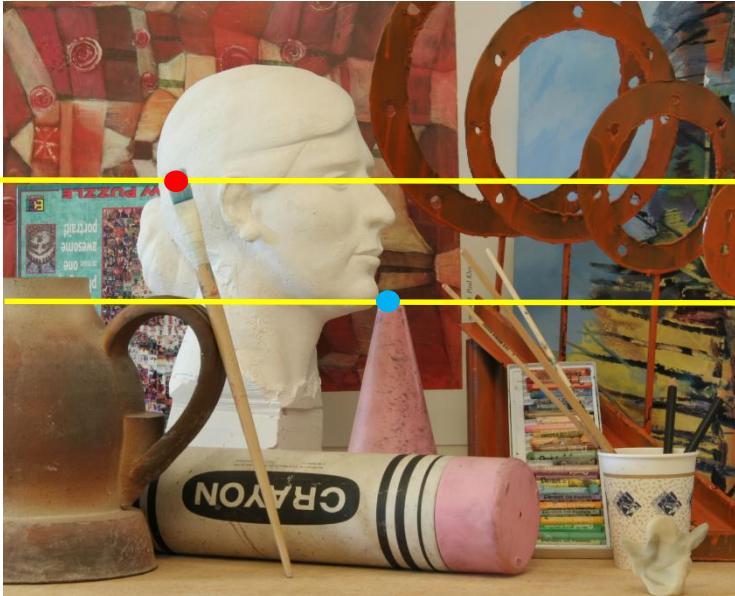
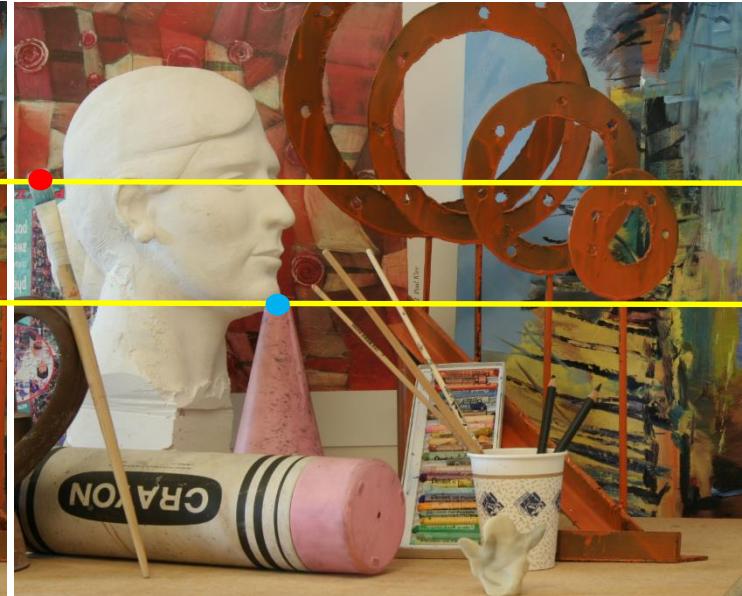


Image Rectification

- ❖ Image rectification is a transformation process used to project images onto a common image plane



Rectified left image



Rectified right image

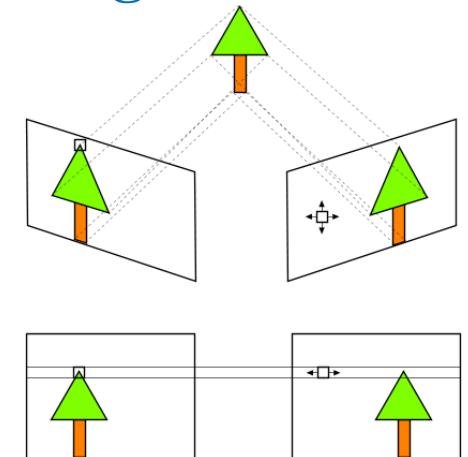


Image Rectification

❖ Introduce to stereo rectification

Before rectification



After rectification

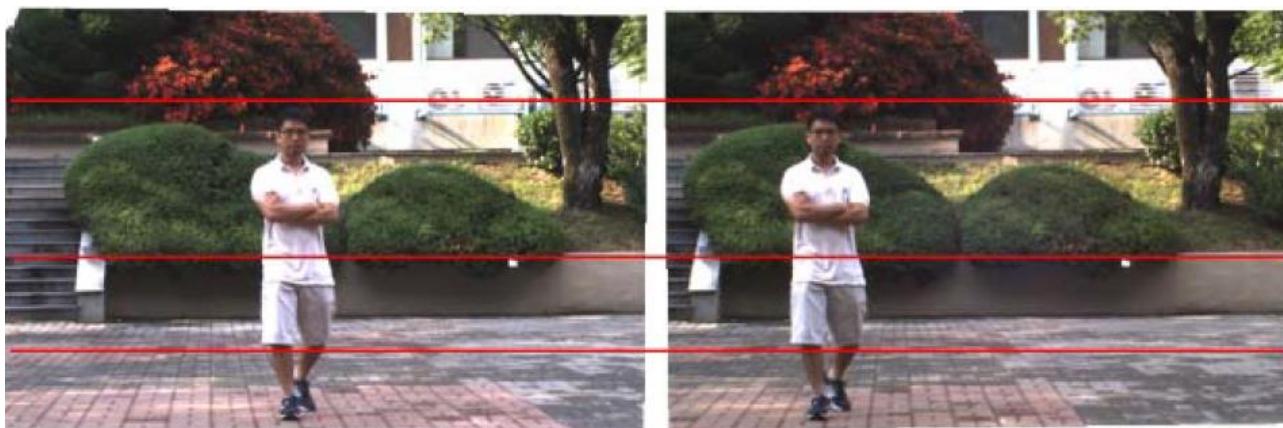


Image Rectification

- ❖ Proposed method
- ❖ Three cameras

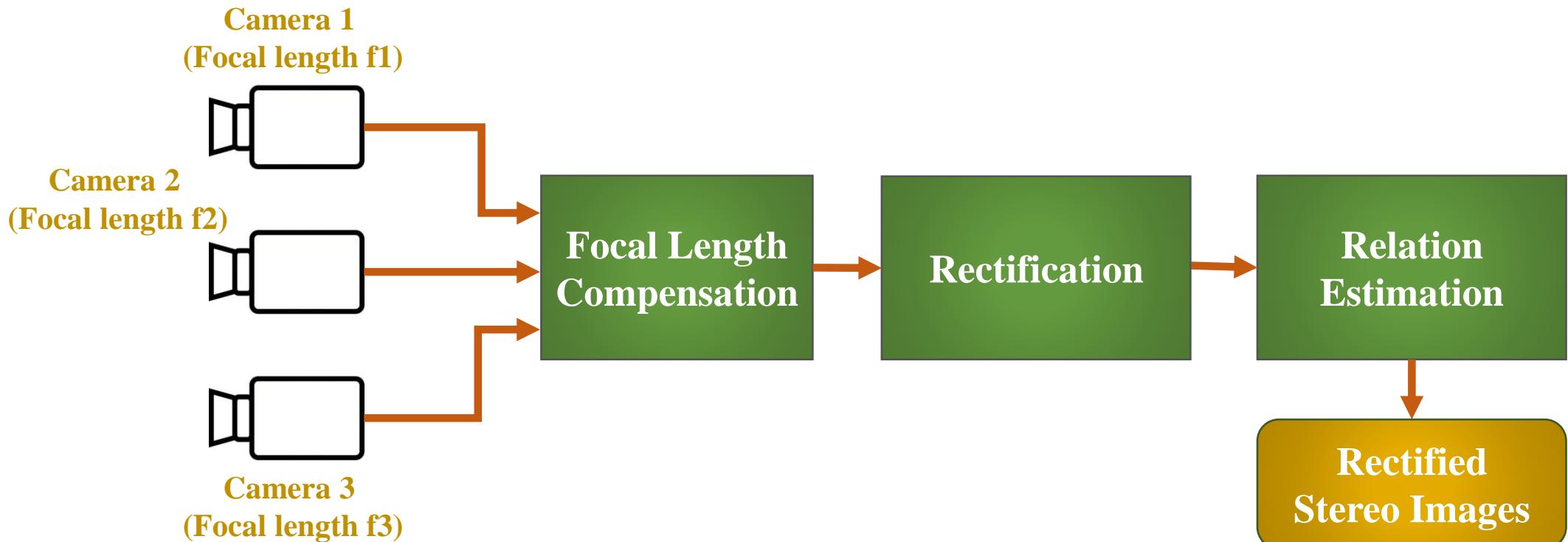


Image Rectification

- ❖ Proposed method
- ❖ Two cameras

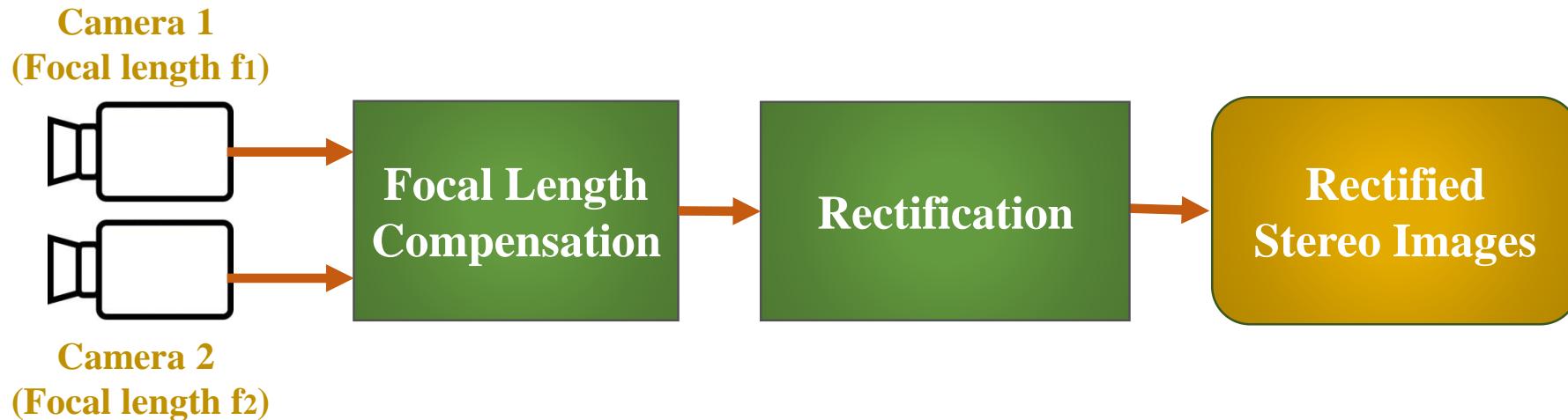
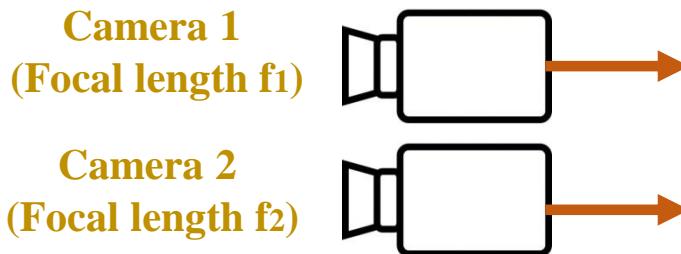


Image Rectification

❖ Proposed method



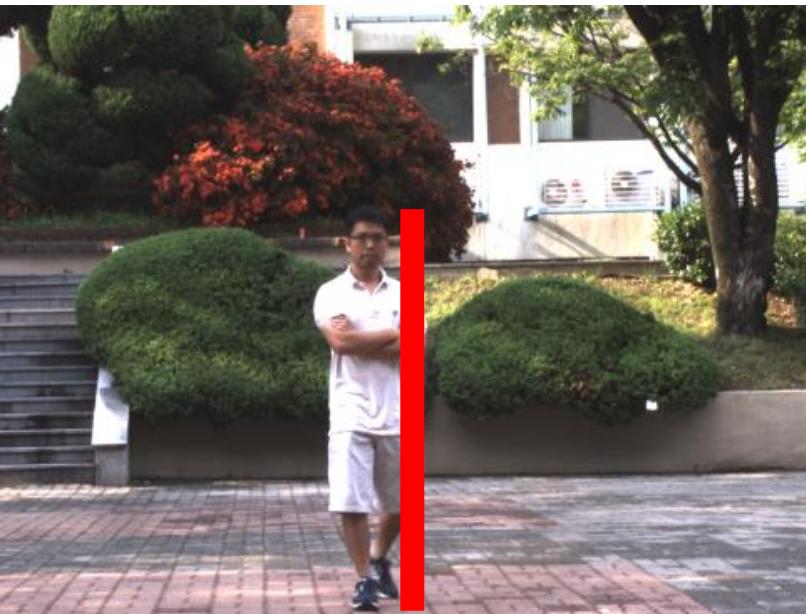
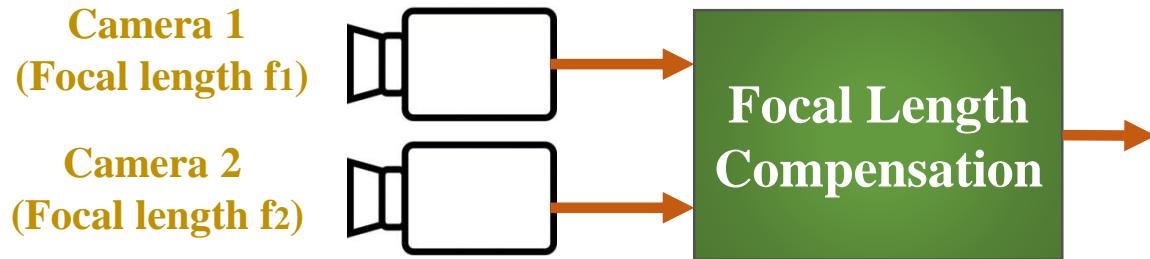
Left image
(longer focal length)



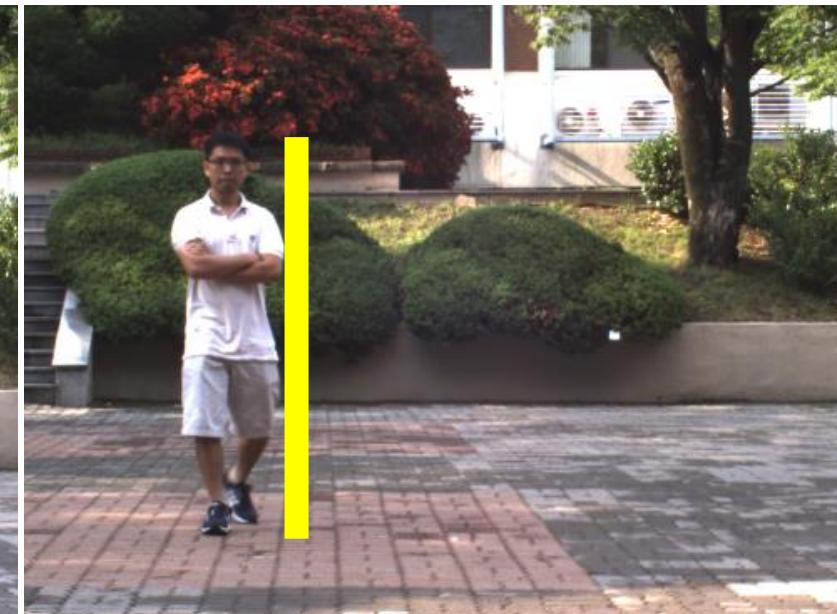
Right image
(shorter focal length)

Image Rectification

- ❖ Rectification
- ❖ Proposed method



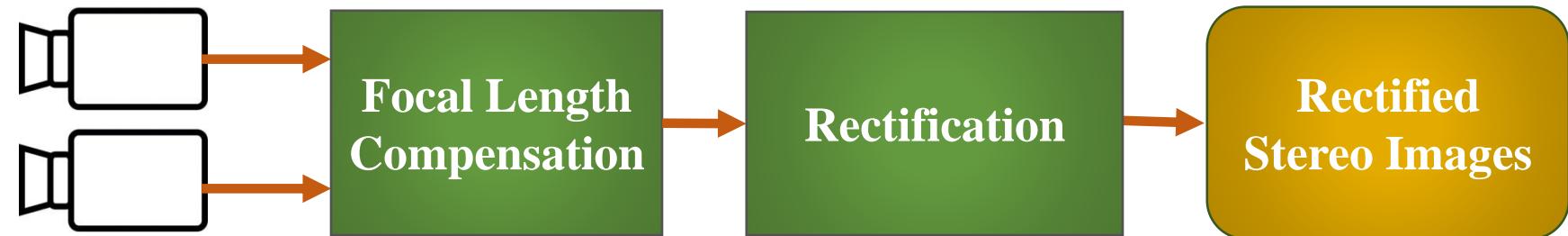
Left image
(longer focal length)



Right image
(shorter focal length)

Image Rectification

❖ Proposed method

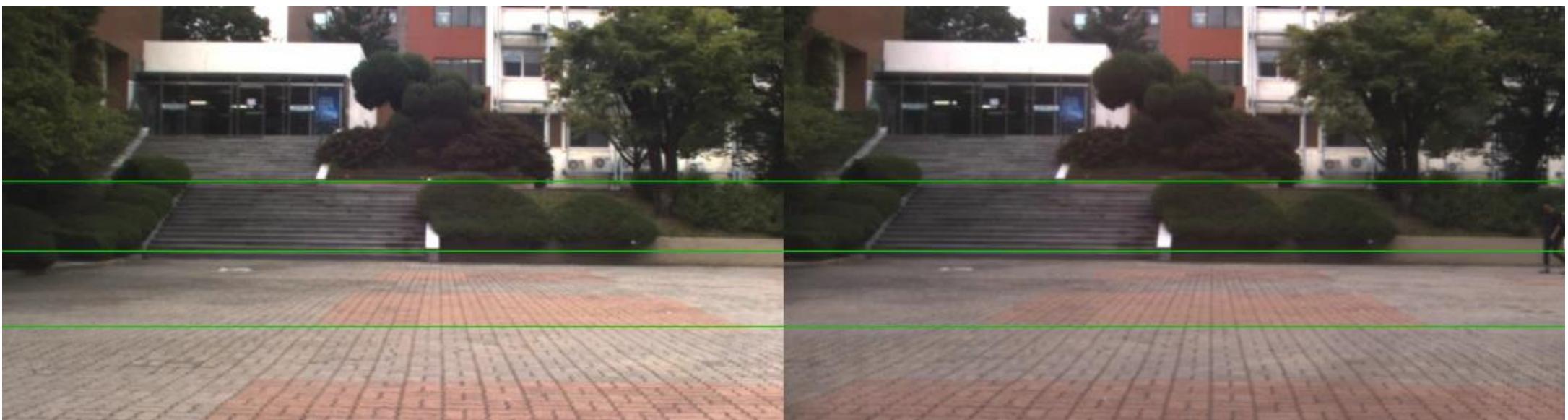


Left image
(longer focal length)

Right image
(shorter focal length)

Image Rectification

❖Qualitative results

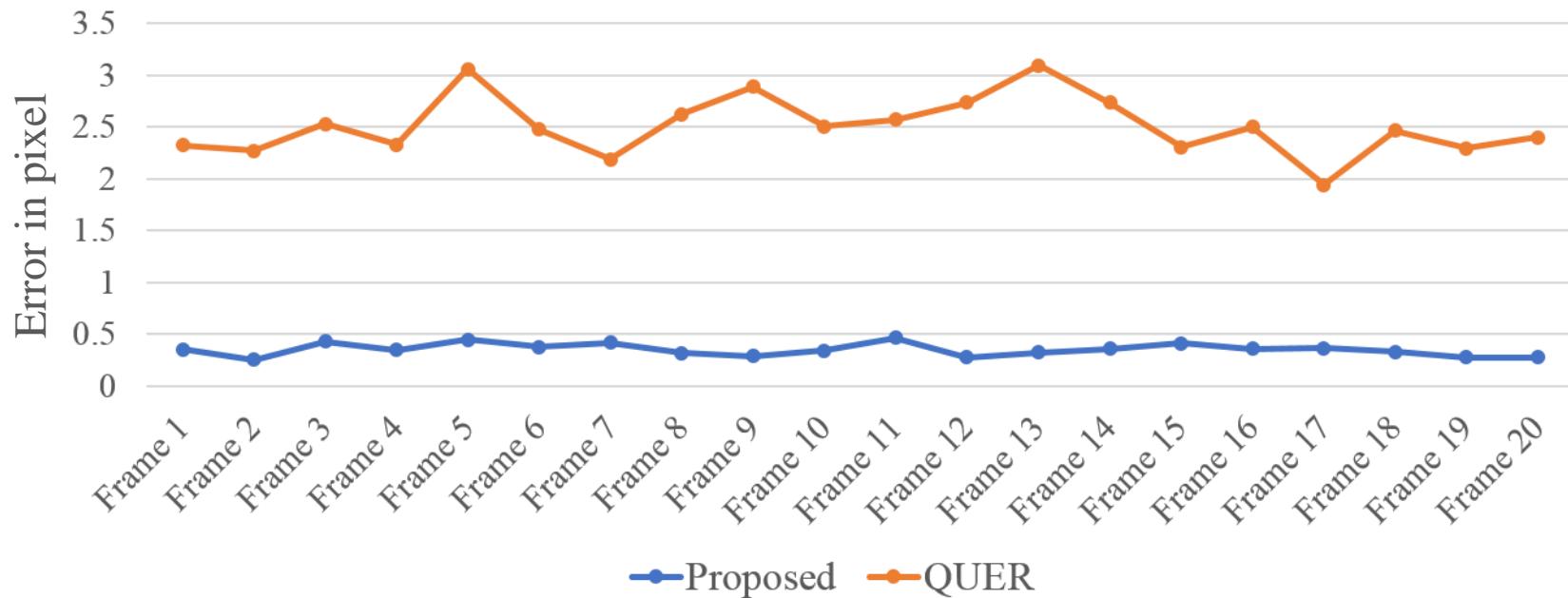


Left image

Right image

Image Rectification

❖ Quantitative results



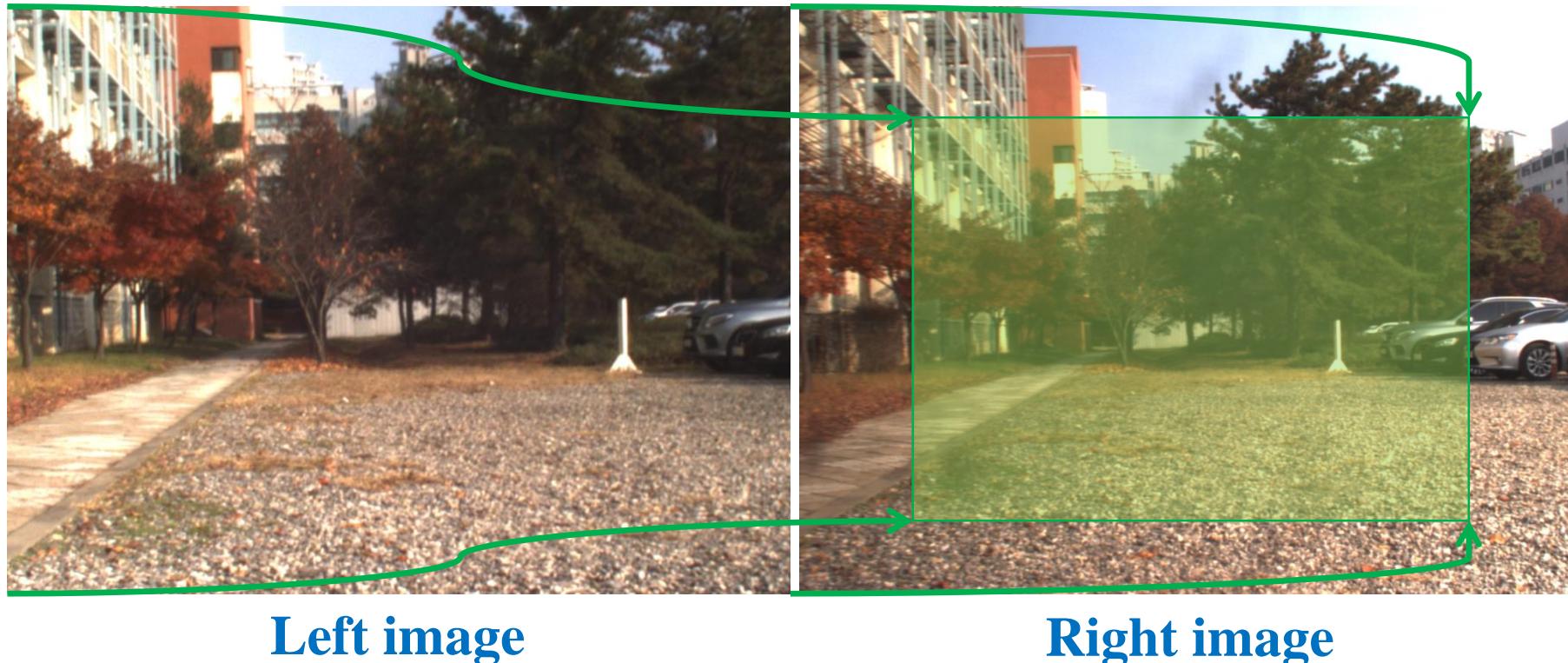
QUER is a state-of-the-art rectification method

Outline

- Introduction to research topics
- Stereo matching
- Computer vision using cameras with different focal lengths
 - Motivation
 - Image rectification
- Stereo matching
- Cross-spectral patch matching

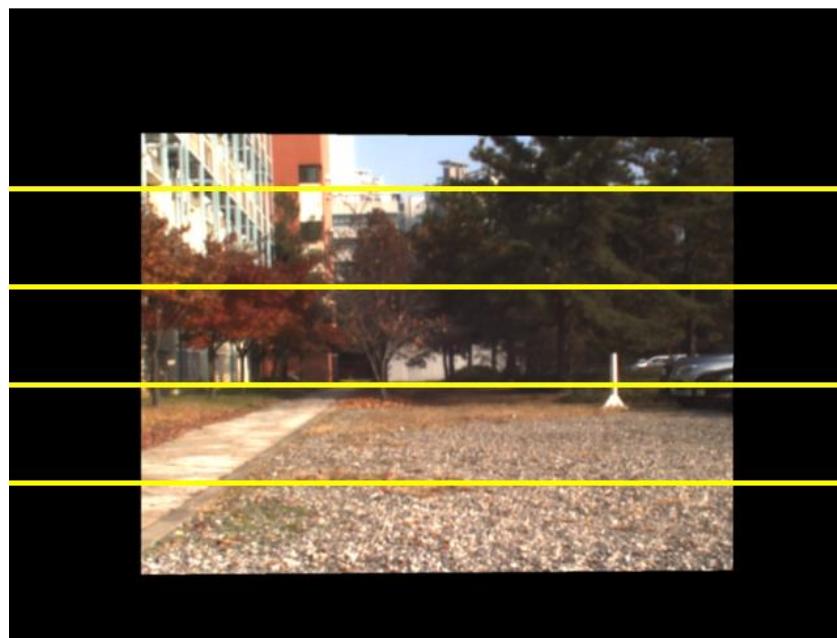
Stereo Matching

❖ Motivation

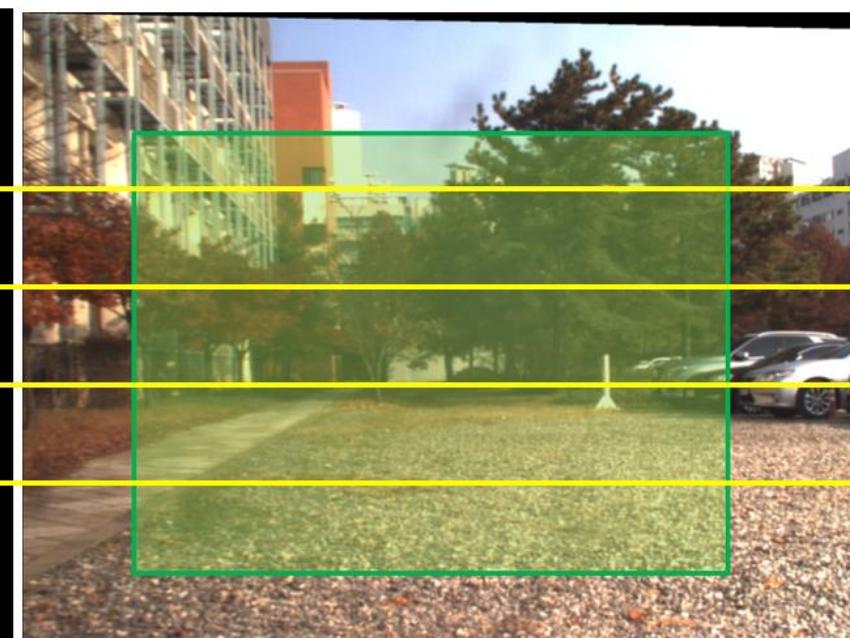


Stereo Matching

❖ Motivation



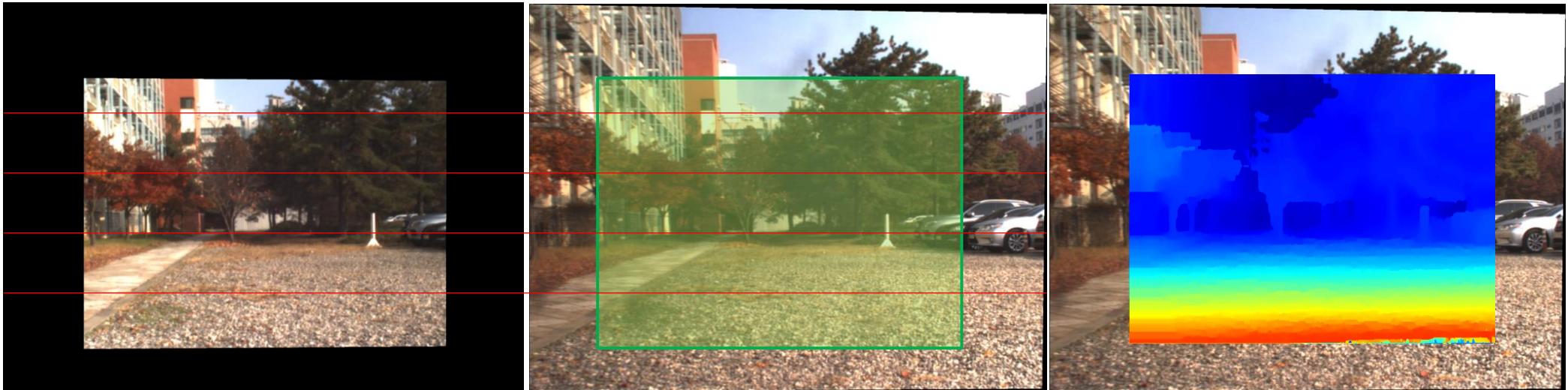
Rectified Left Image



Rectified Right Image

Stereo Matching

❖ Motivation



Rectified Left Image

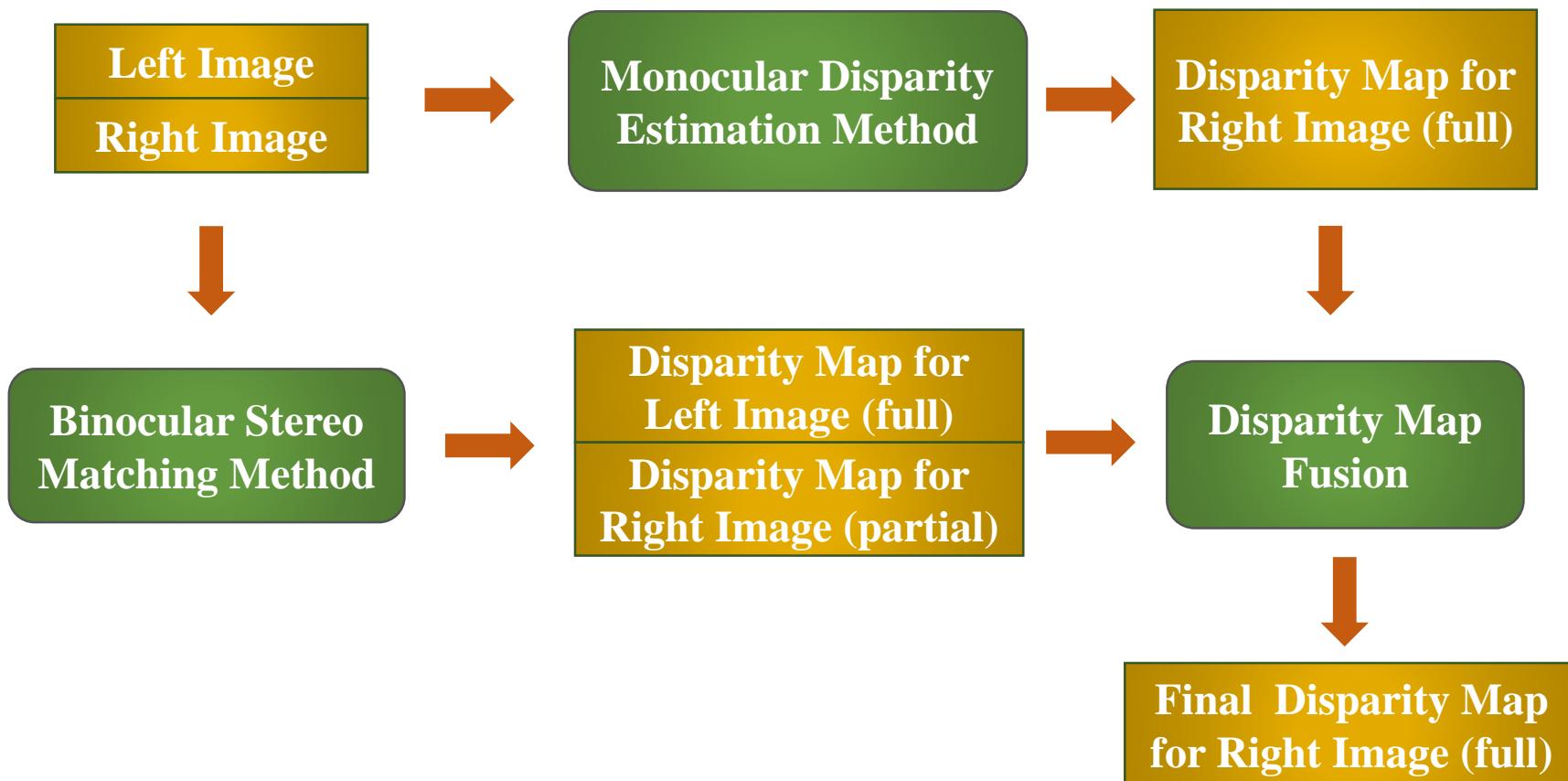
Rectified Right Image

Using a stereo matching
method

How to compute disparity maps for entire images?

Stereo Matching

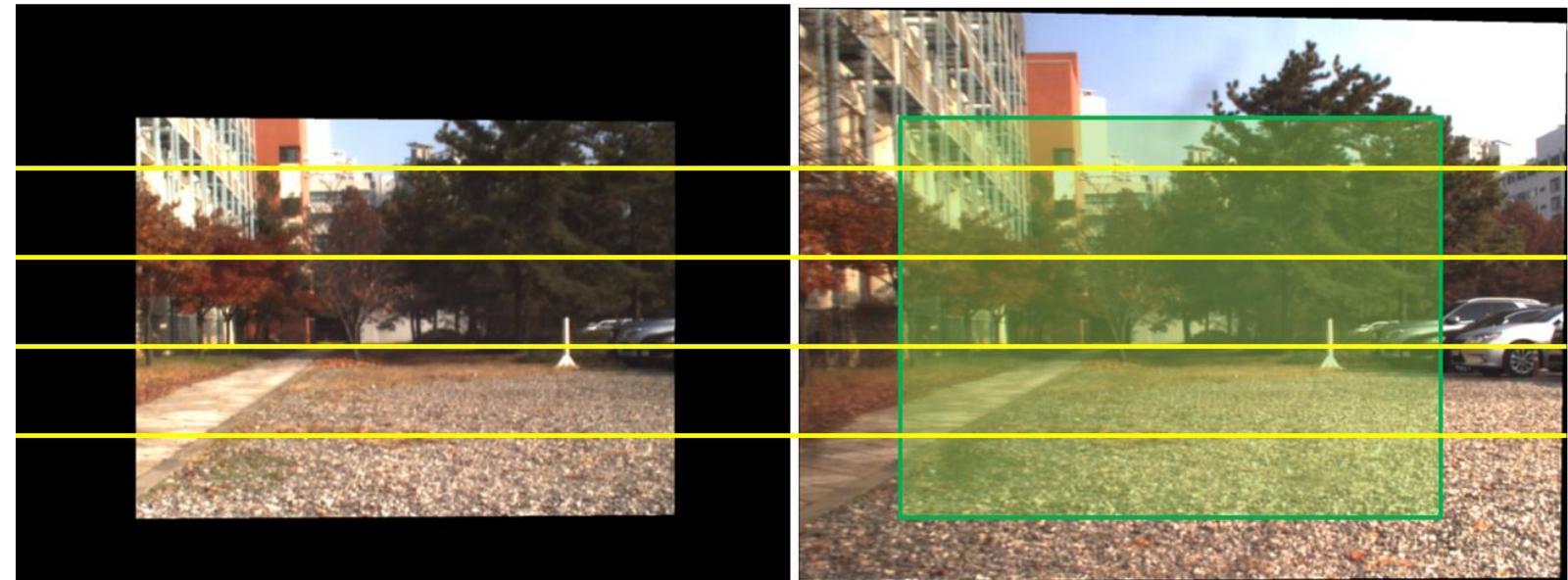
❖ Proposed method



Stereo Matching

❖ Proposed method

Left Image
Right Image

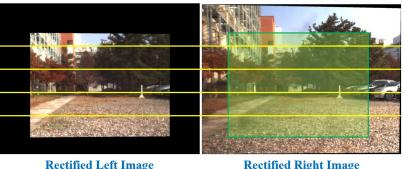


Rectified Left Image

Rectified Right Image

Stereo Matching

❖ Proposed method



Left Image
Right Image



Monocular Disparity
Estimation Method



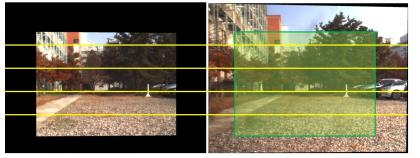
Disparity Map for
Right Image (full)



Disparity Map
using Monodepth

Stereo Matching

❖ Proposed method



Left Image
Right Image

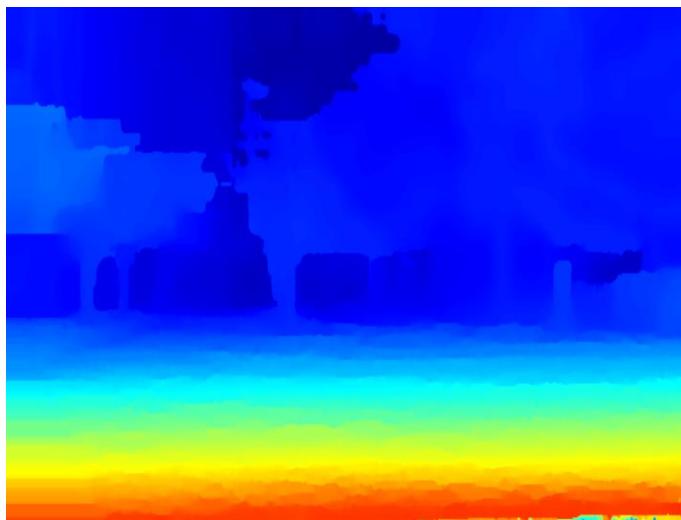
Monocular Disparity
Estimation Method



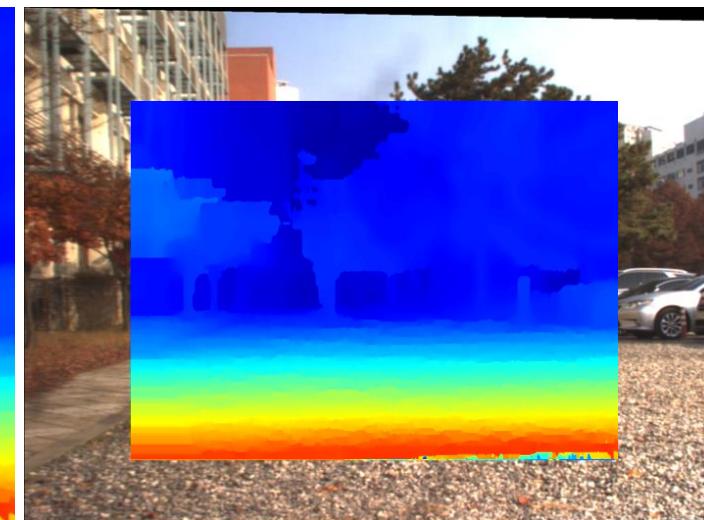
Disparity Map for
Right Image (full)

Binocular Stereo
Matching Method

Disparity Map for
Left Image (full)
Disparity Map for
Right Image (partial)



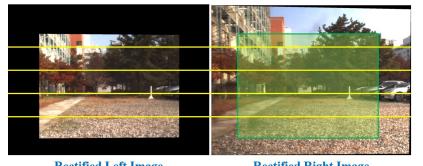
Left disparity map



Right disparity map

Stereo Matching

❖ Proposed method



Left Image
Right Image

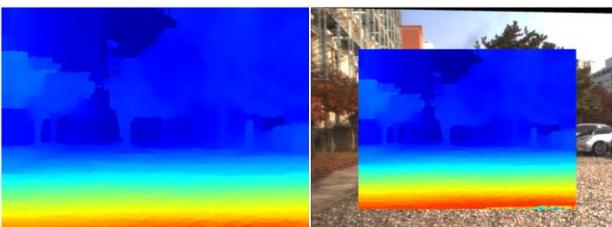
Monocular Disparity Estimation Method



Disparity Map for Right Image (full)

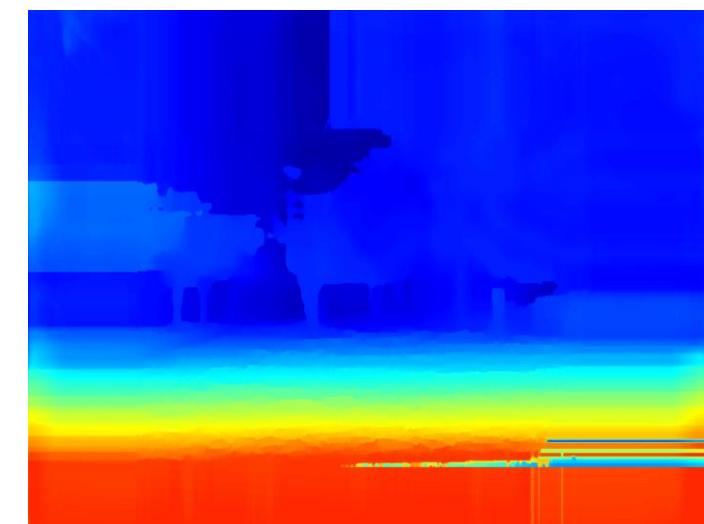
Binocular Stereo Matching Method

Disparity Map for Left Image (full)
Disparity Map for Right Image (partial)



Disparity Map Fusion

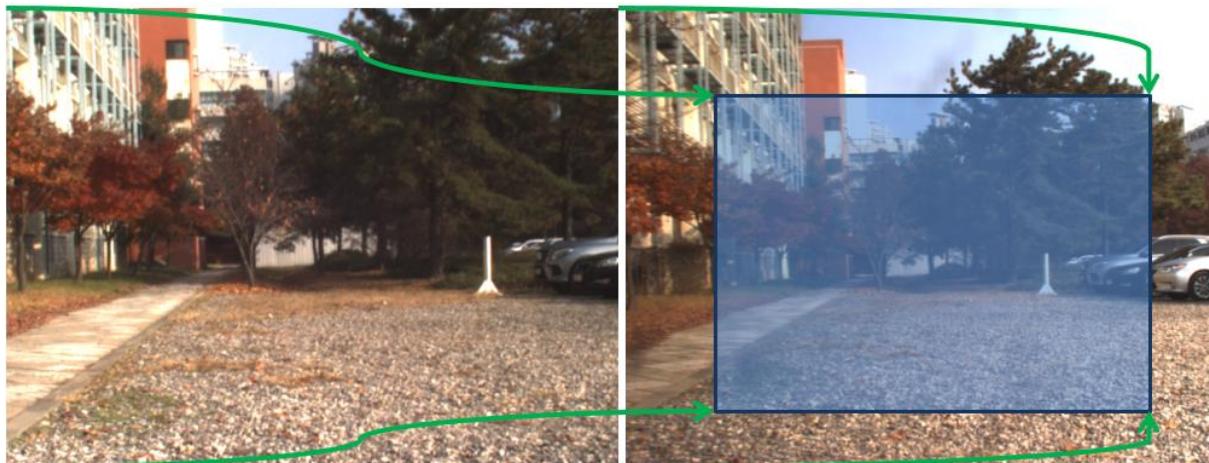
Final Disparity Map for Right Image (full)



Right disparity map

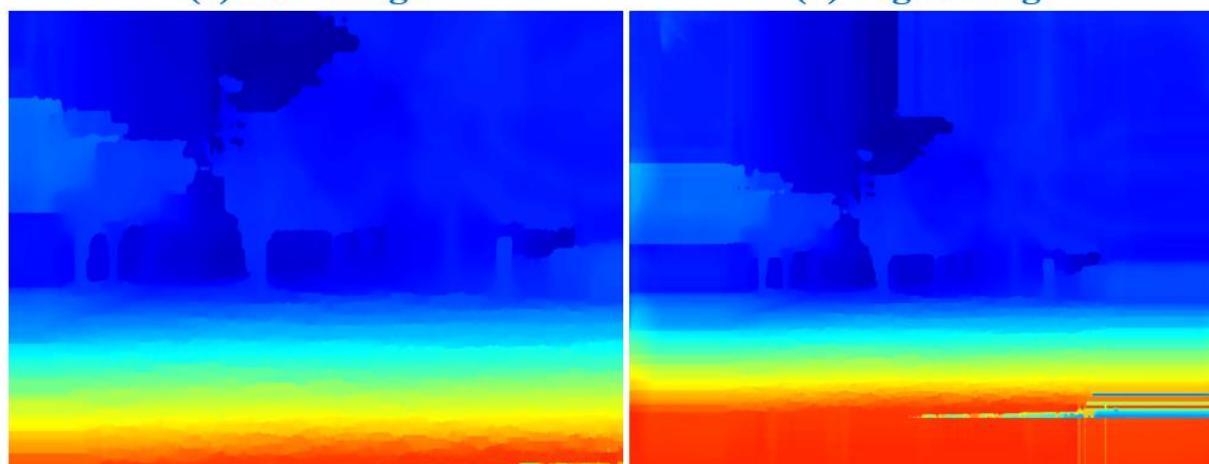
Stereo Matching

❖Qualitative results



(a) Left image

(b) Right image

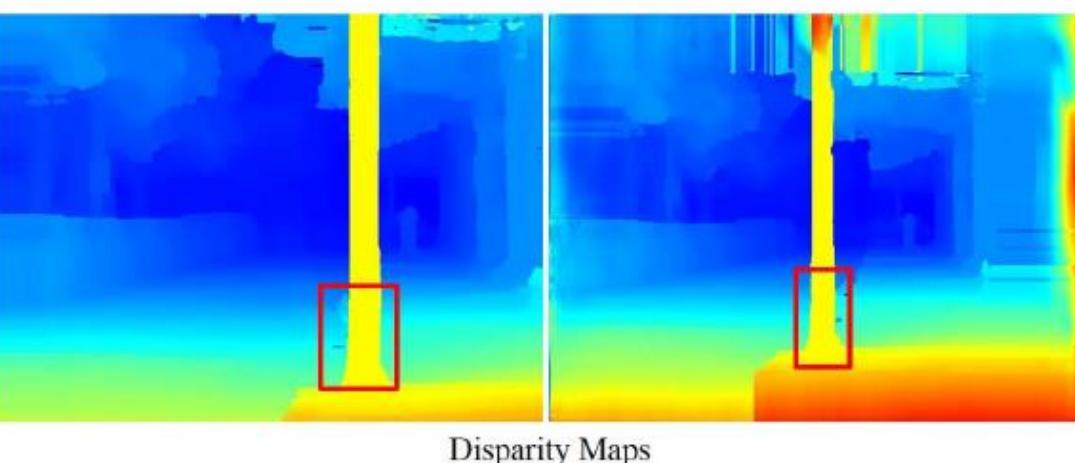


(c) Disparity map for (a)

(d) Disparity map for (b)

Stereo Matching

❖Qualitative results



Stereo Matching

❖ Quantitative results

❖ Using KITTI stereo dataset



(a) Right Image



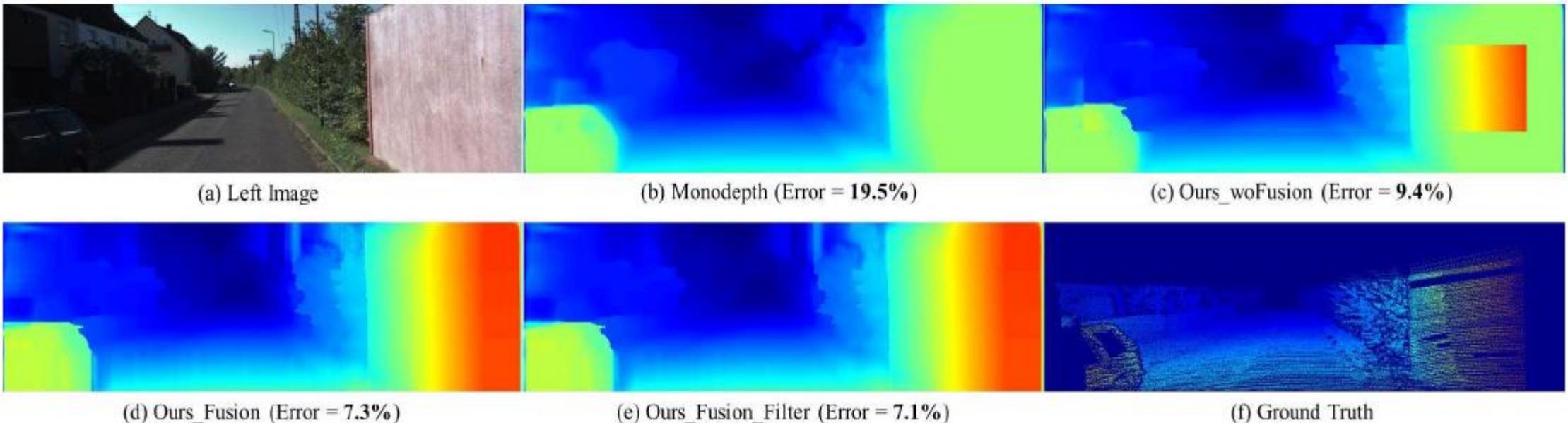
(b) Simulated Right Images with $P=80$



(c) Simulated Right Images with $P=100$

Stereo Matching

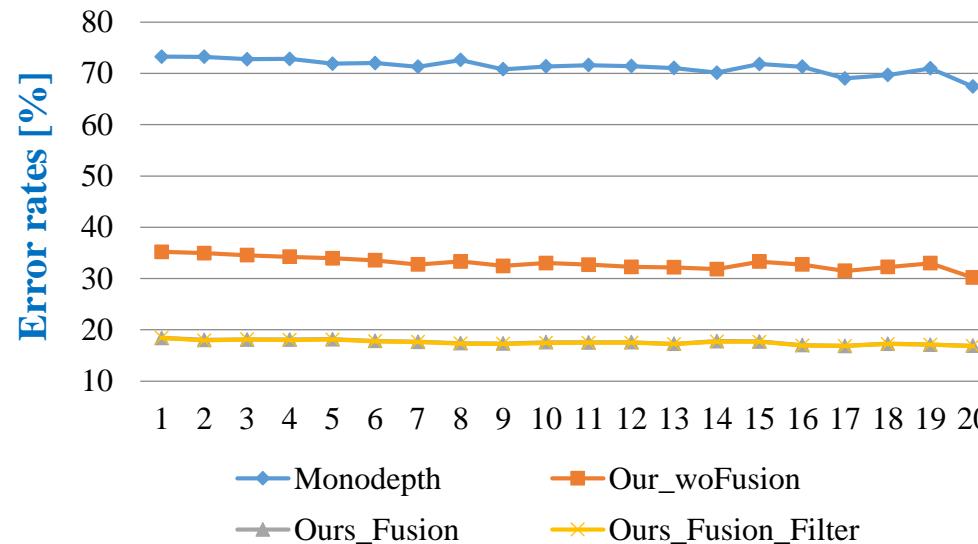
- ❖ Qualitative results
- ❖ Comparison



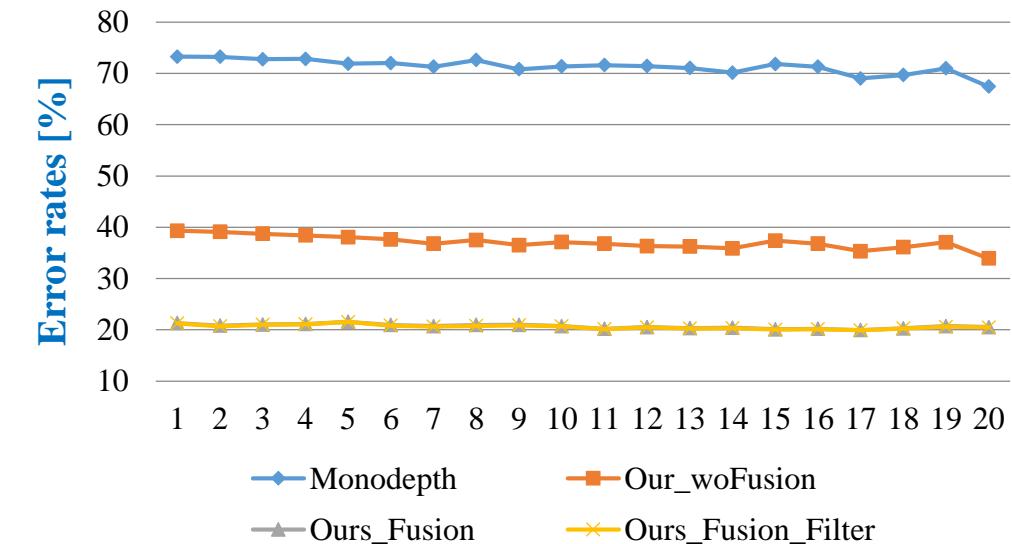
Stereo Matching

❖ Quantitative results

❖ Using KITTI stereo dataset



(a) Padding=80



(b) Padding=100

Stereo Matching

Stereo Matching

Stereo Matching / Binocular Depth Estimation



Hình từ camera trái



Hình từ camera phải

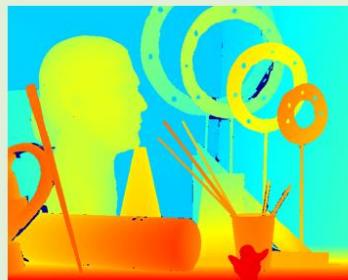


Stereo camera

Thuật toán
stereo matching



Hình chiều sâu
cho hình trái

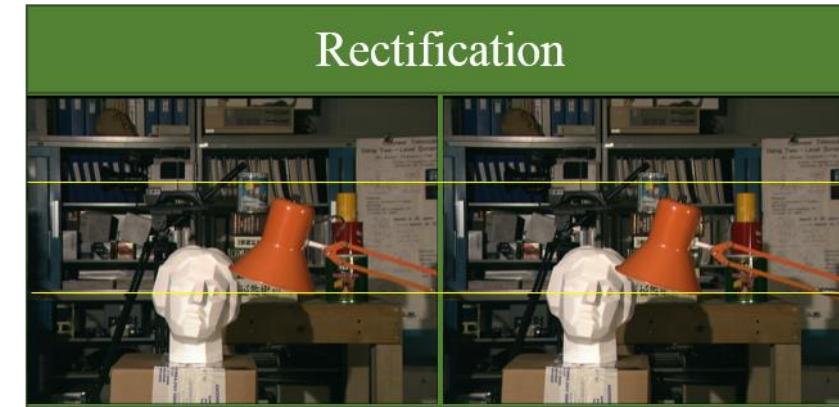
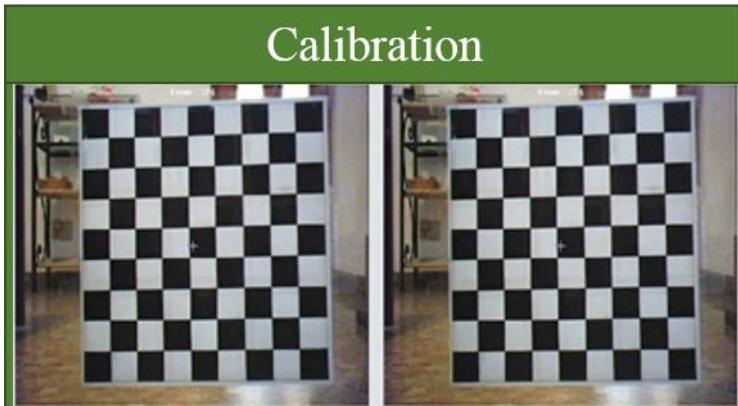


Hình chiều sâu
cho hình phải

Ứng dụng

- Đo khoảng cách các tới các object trong hình
- Xây dựng mô hình 3D
- Thông tin cho các ứng dụng khác: detection, tracking, ...

Stereo Matching



Stereo Matching

- ❖ Left and right images of a stereo pair



Left Image

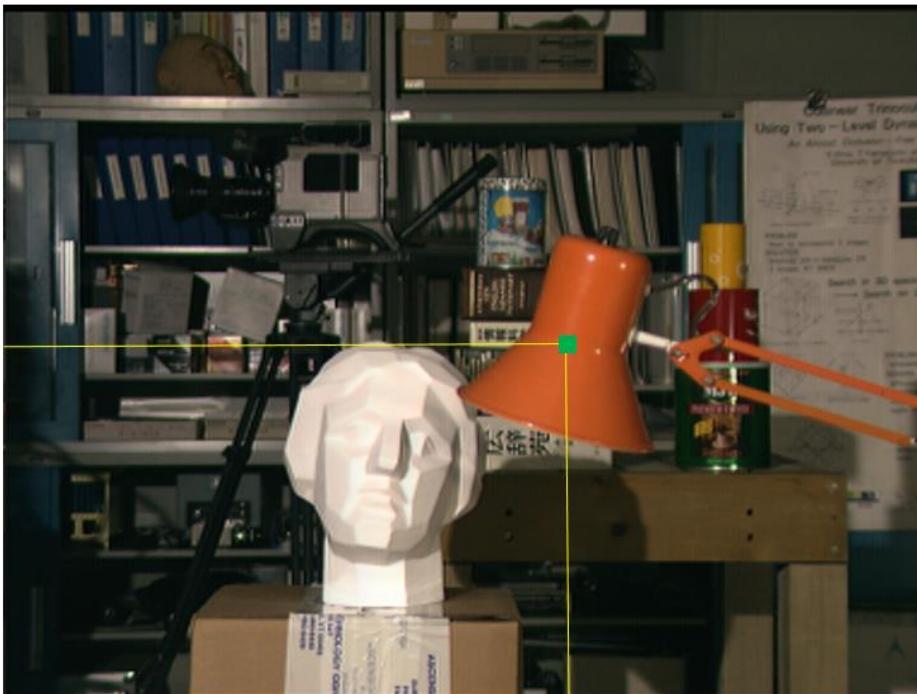


Right Image

Stereo Matching

$p = (234, 140)$

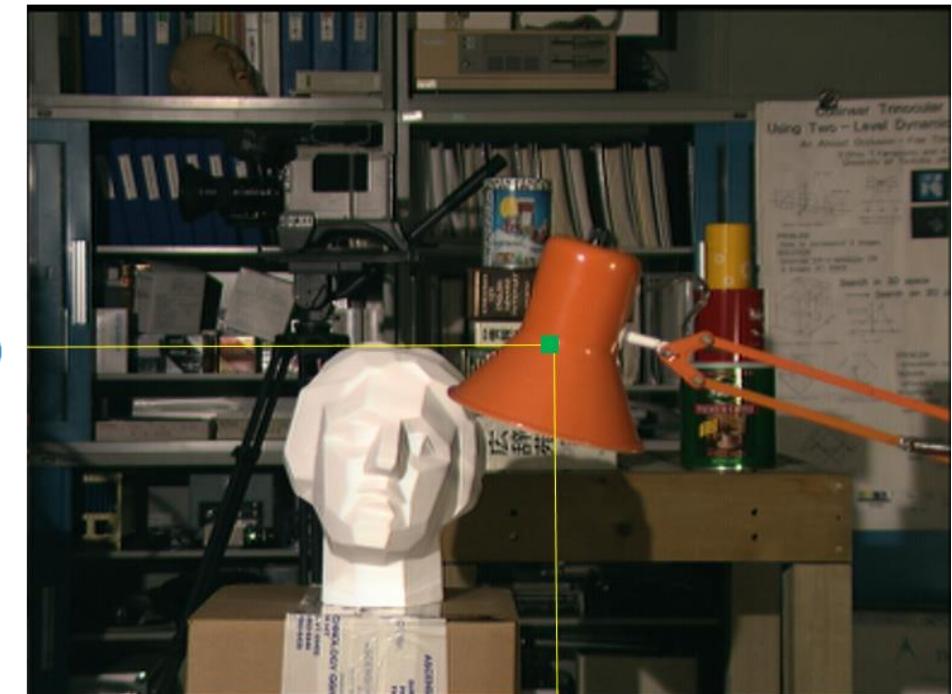
$y_p = 140$



$x_p = 234$

$q = (220, 140)$

$y_q = 140$

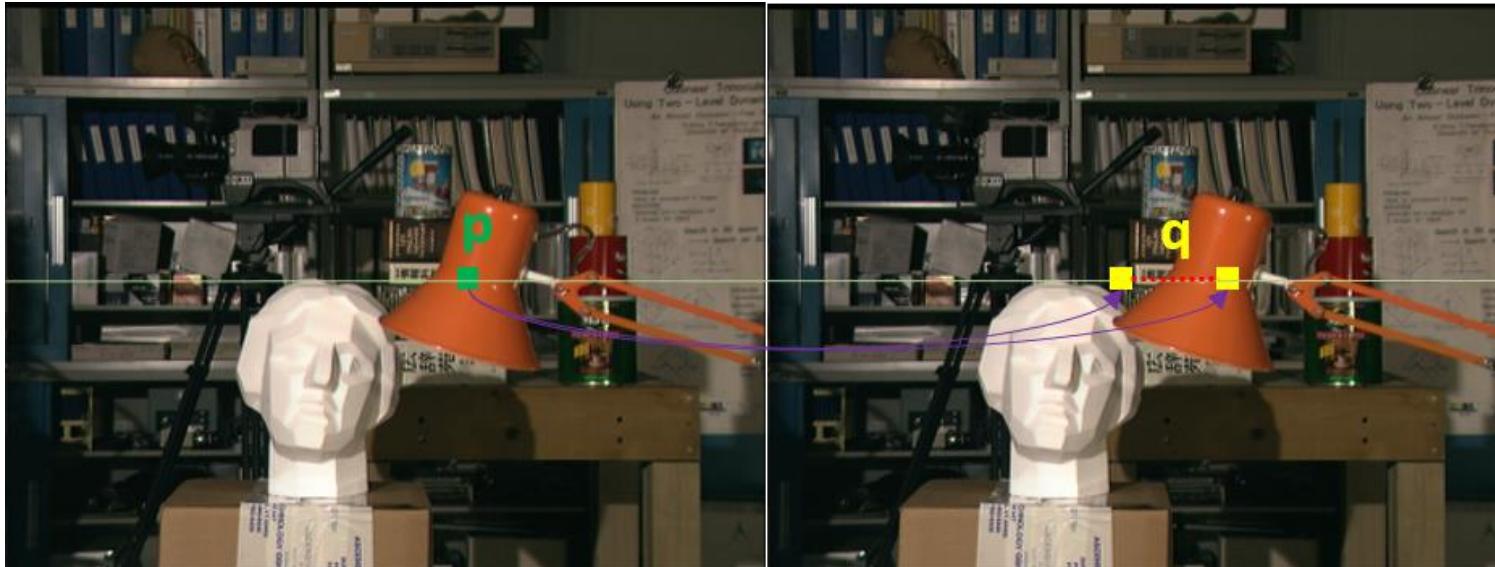


$x_q = 220$

$$\text{disparity}_q = x_p - x_q = 14$$

Stereo Matching

❖ Method 1



Left Image

Right Image

$$d_p = \arg \min_{d \in D} (C(p, q))$$

where $C(p, q) = (L(p) - R(q))^2$

and $q = (x_p - d, y_p)$

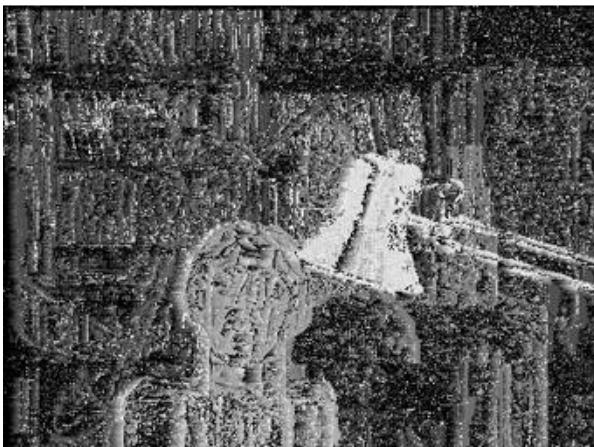
Stereo Matching



Left Image



Right Image



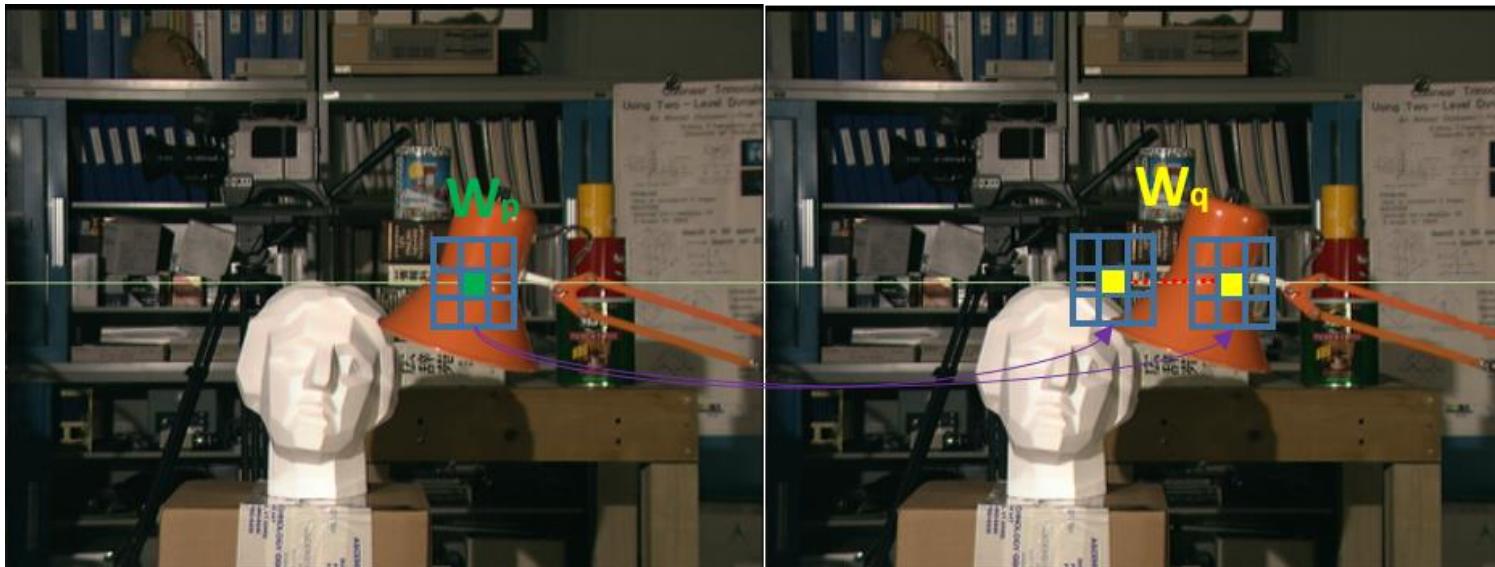
Disparity Map



Ground Truth

Stereo Matching

❖ Method 2



Left Image

Right Image

$$d_p = \arg \min_{d \in D} (C(p, q))$$

$$\text{where } C(p, q) = \sum_{(u, v) \in (W_p, W_q)} (L(u) - R(v))^2$$

$$\text{and } q = (x_p - d, y_p)$$

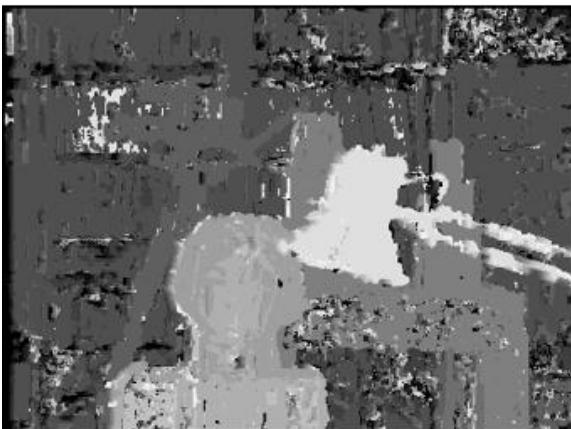
Stereo Matching



Left Image



Right Image



Disparity Map



Ground Truth

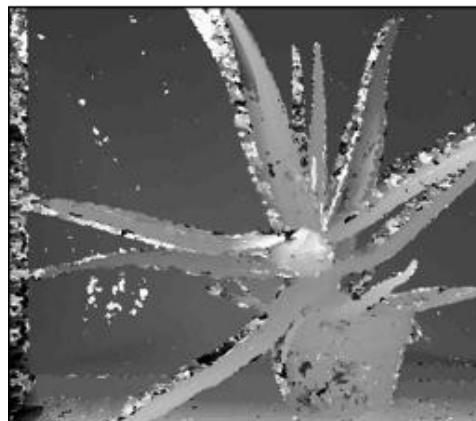
Stereo Matching

❖ Aloe stereo pair

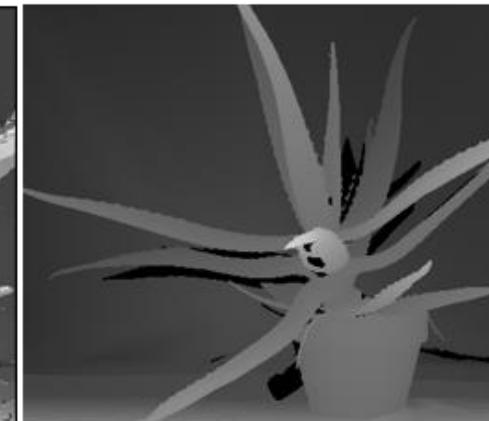


Left Image

Right Image



Disparity Map



Ground Truth

Stereo Matching

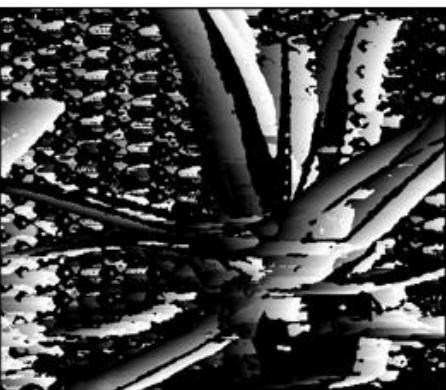
❖ Aloe stereo pair



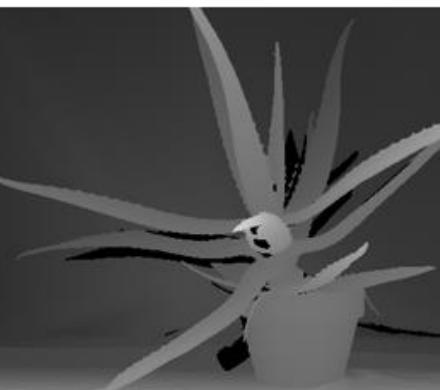
Left Image



Right Image



Disparity Map



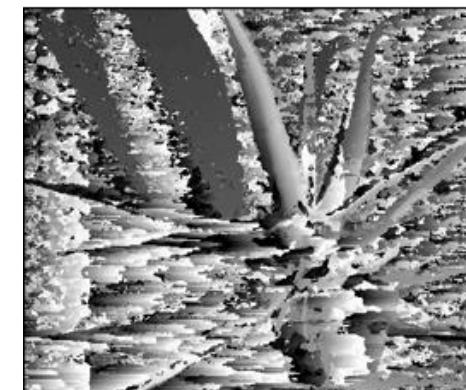
Ground Truth



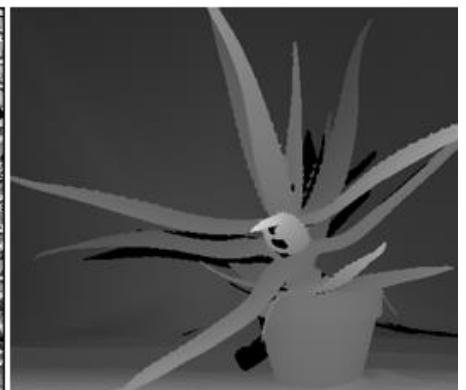
Left Image



Right Image



Disparity Map

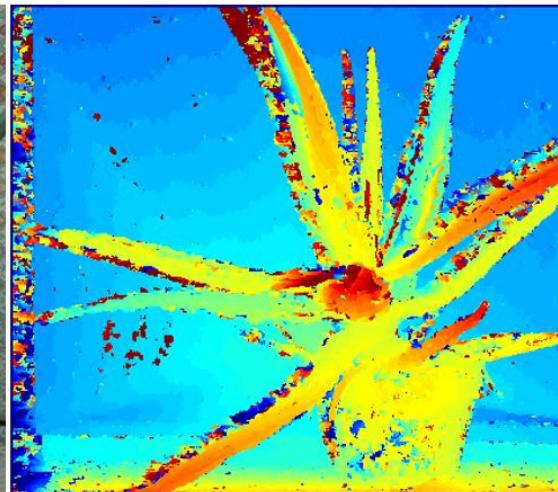


Ground Truth

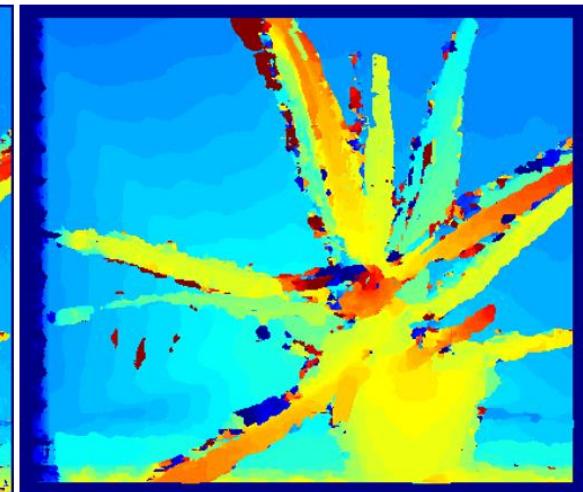
Ứng dụng tìm chiều sâu (depth estimation)



Ảnh stereo có cùng độ sáng



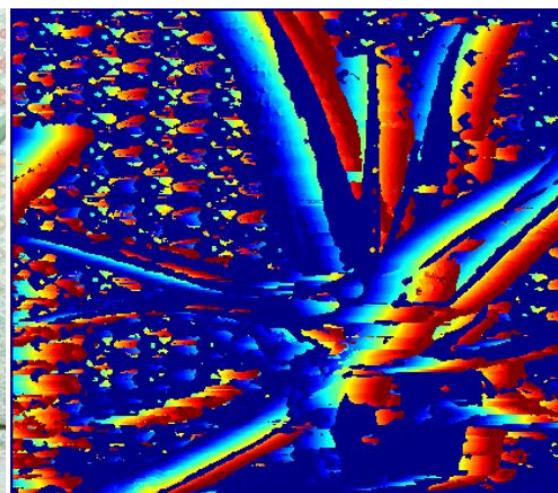
Absolute difference



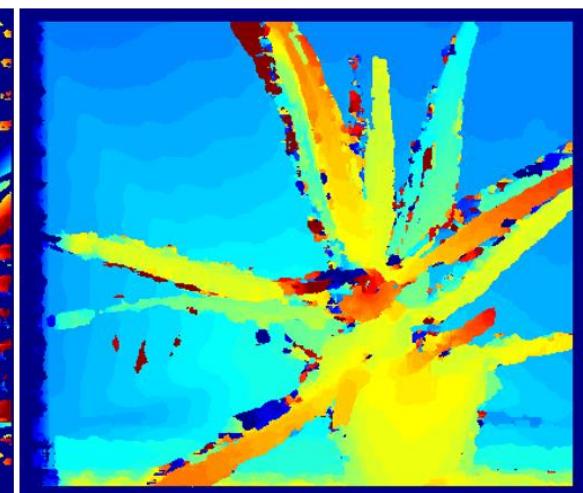
Cosine Similarity



Ảnh stereo khác độ sáng



Absolute difference

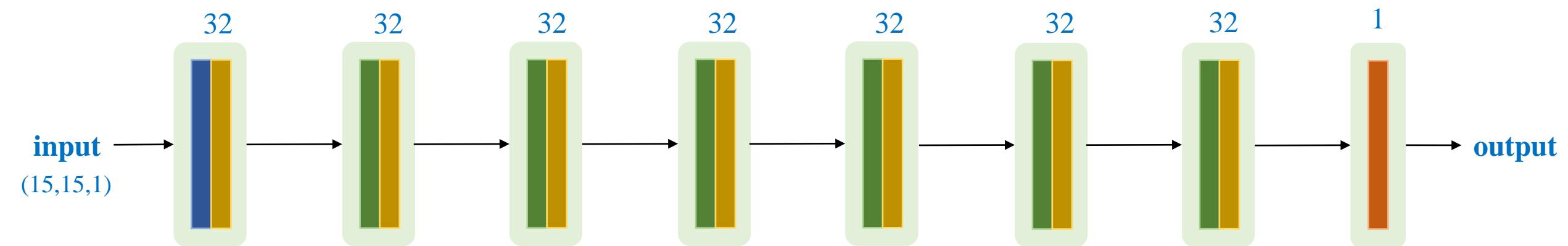


Cosine Similarity

Cosine similarity hoạt động ổn định khi ảnh stereo thay đổi độ sáng

Stereo Matching

❖ Using deep learning



Stereo Matching

❖ Using deep learning



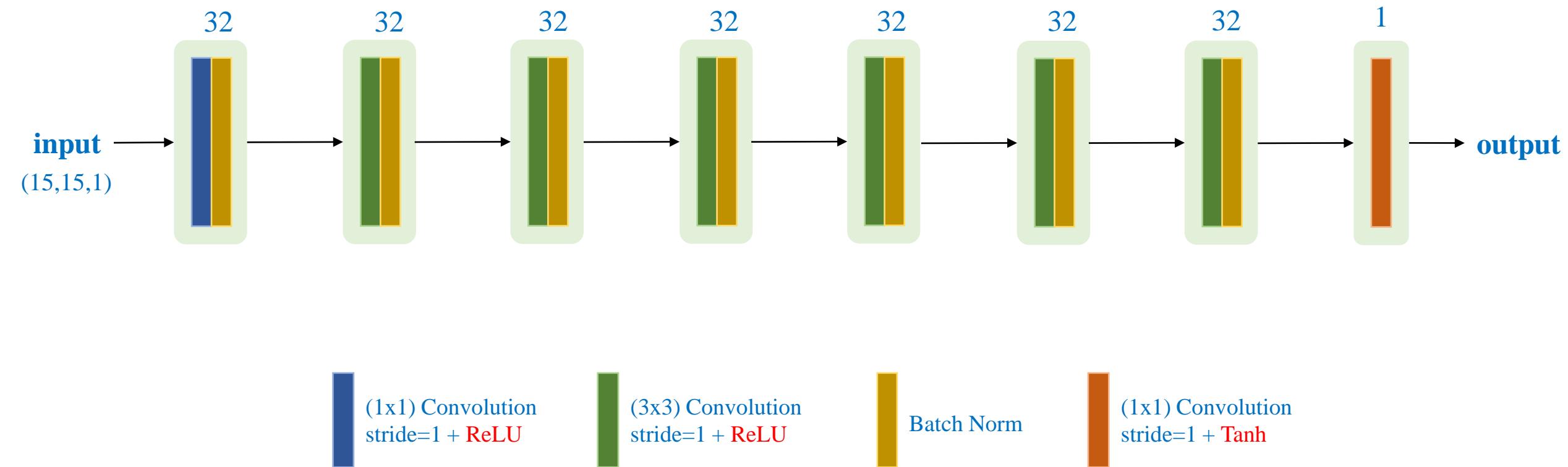
(a)

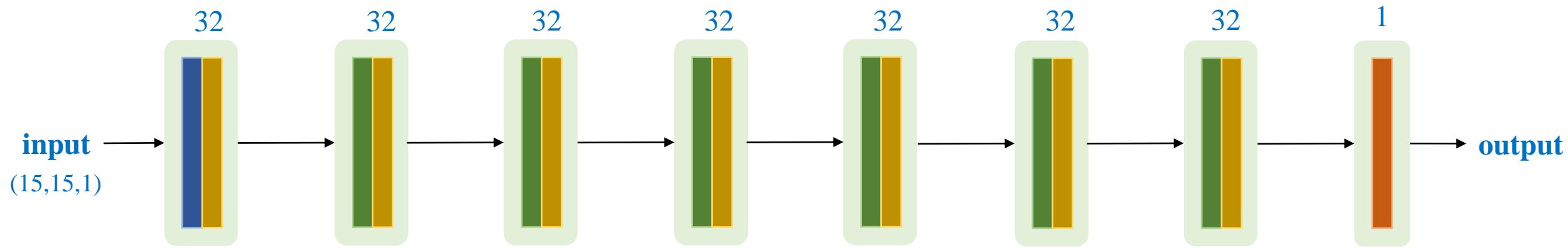


(b)

Stereo Matching

❖ Using deep learning



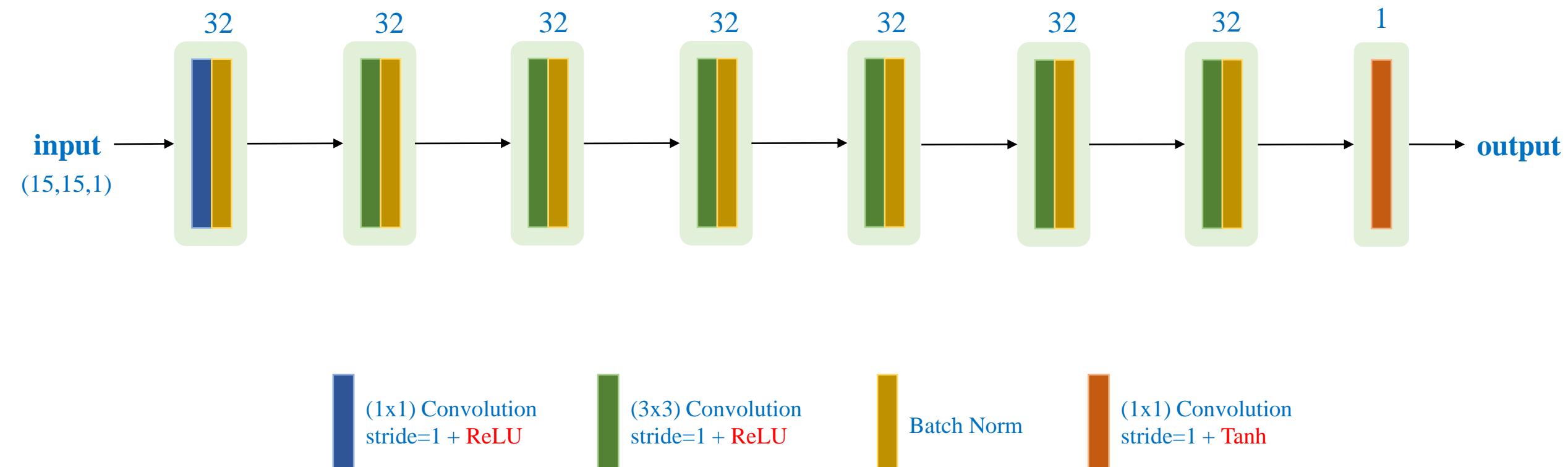


```
x = Conv2D(32, (1,1), activation='relu', kernel_initializer='he_uniform')(x)
x = BatchNormalization()(x)
x = Conv2D(32, (3,3), activation='relu', kernel_initializer='he_uniform')(x)
x = BatchNormalization()(x)
x = Conv2D(32, (3,3), activation='relu', kernel_initializer='he_uniform')(x)
x = BatchNormalization()(x)
x = Conv2D(32, (3,3), activation='relu', kernel_initializer='he_uniform')(x)
x = BatchNormalization()(x)
x = Conv2D(32, (3,3), activation='relu', kernel_initializer='he_uniform')(x)
x = BatchNormalization()(x)
x = Conv2D(32, (3,3), activation='relu', kernel_initializer='he_uniform')(x)
x = BatchNormalization()(x)
x = Conv2D(32, (3,3), activation='relu', kernel_initializer='he_uniform')(x)
x = BatchNormalization()(x)
x = Conv2D(32, (3,3), activation='relu', kernel_initializer='he_uniform')(x)
x = BatchNormalization()(x)
x = Conv2D(1, (1,1), activation='tanh')(x)
```

Stereo Matching

❖ Implementation

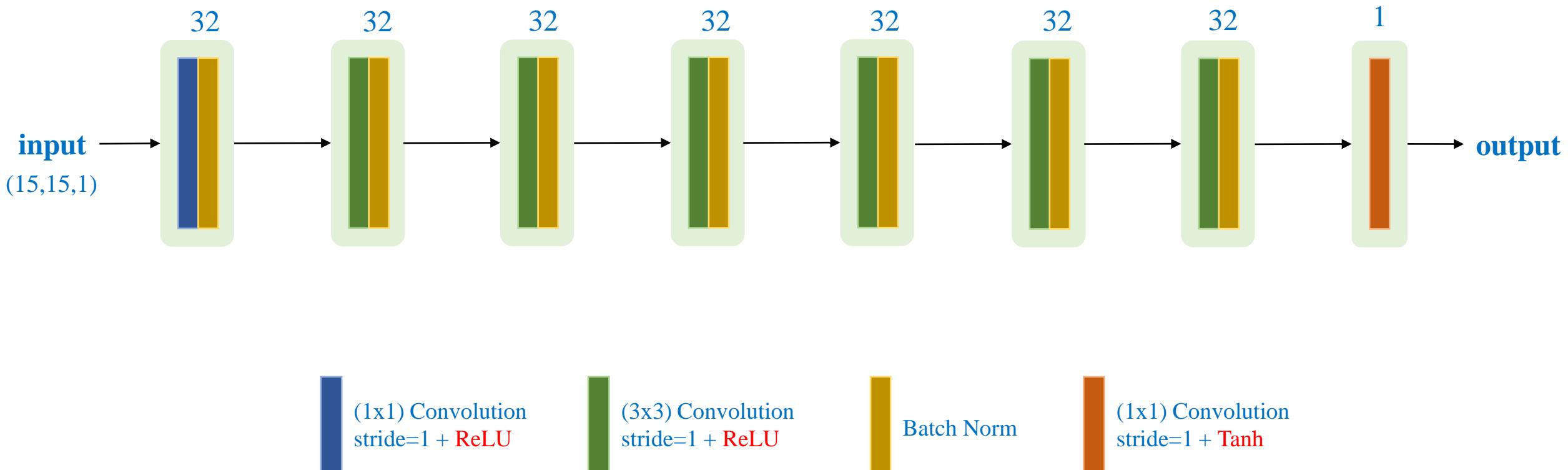
❖ Problem with shapes



Stereo Matching

❖ Implementation

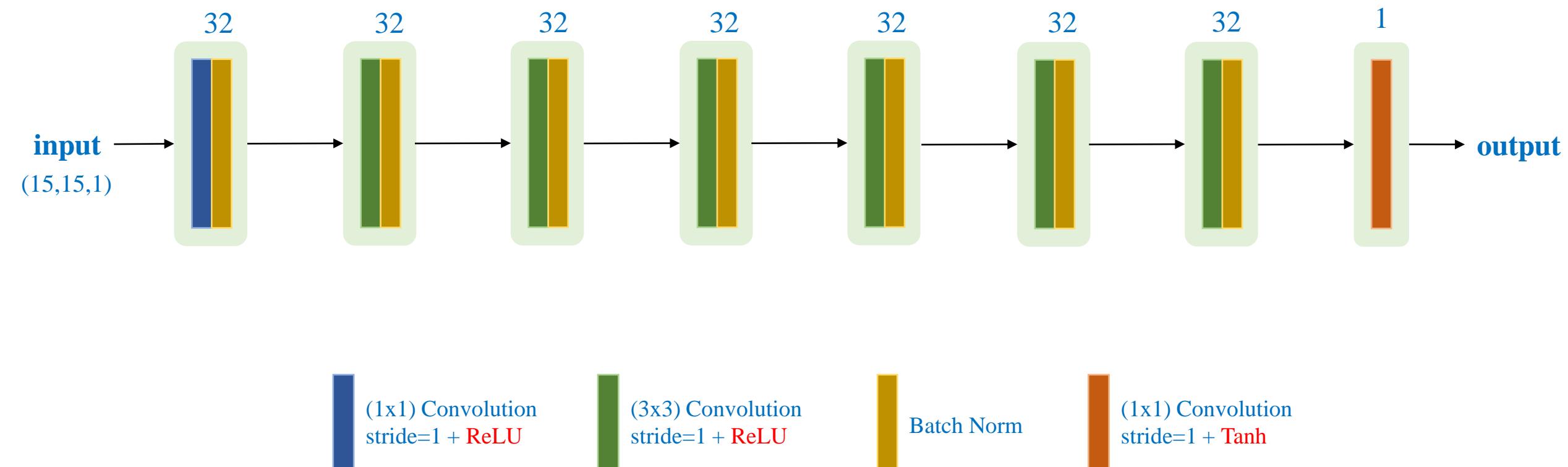
❖ Problem with fixed sizes



Stereo Matching

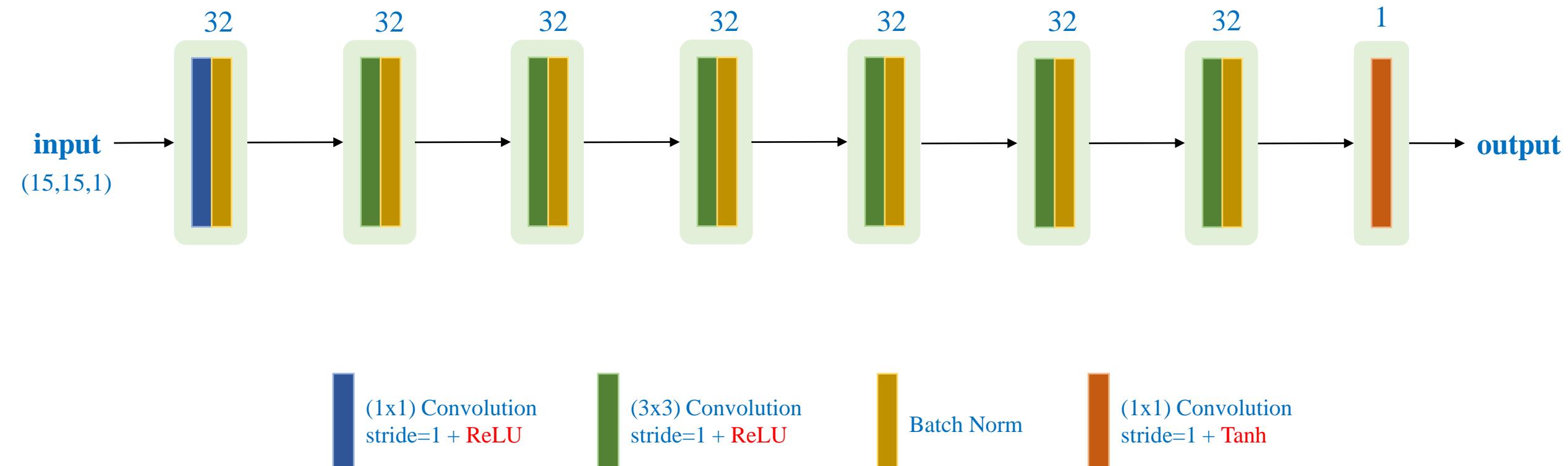
❖ Implementation

❖ Using the fit function



Stereo Matching

- ❖ Implementation
- ❖ Using GradientTape



Thank you!