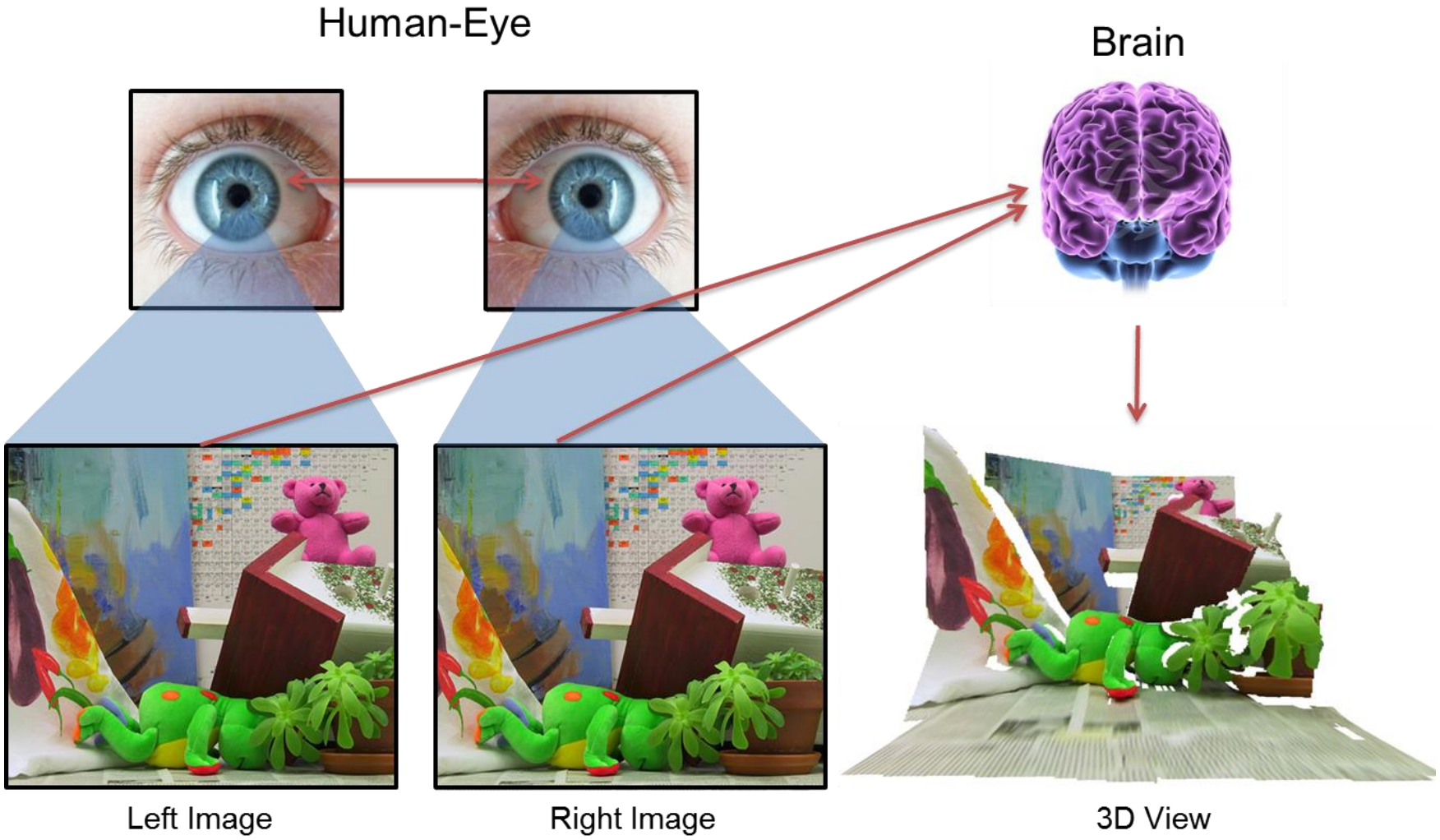
Stereo Matching

Quang-Vinh Dinh
Ph.D. in Computer Science

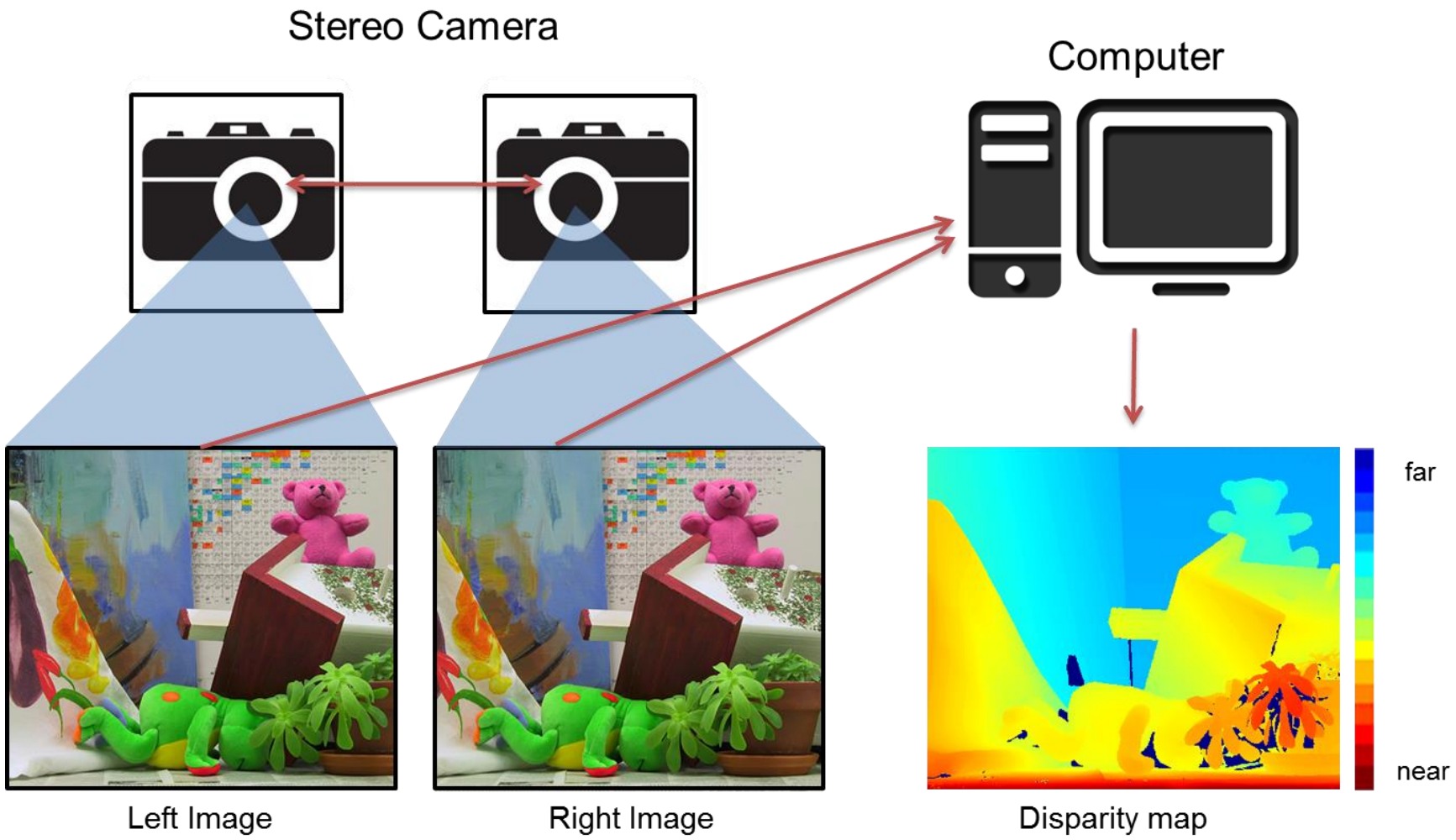
Stereo Matching Algorithm

❖Human perception for 3D information



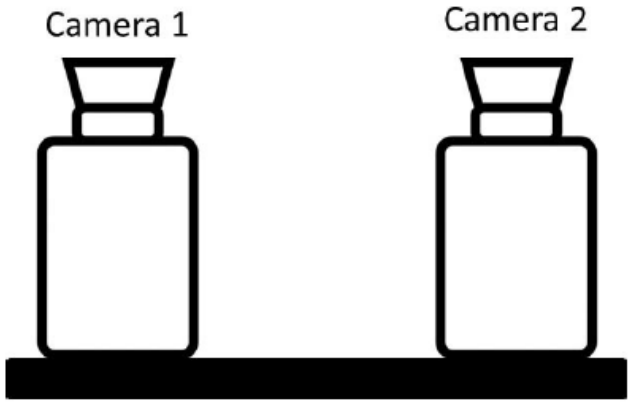
Stereo Matching Algorithm

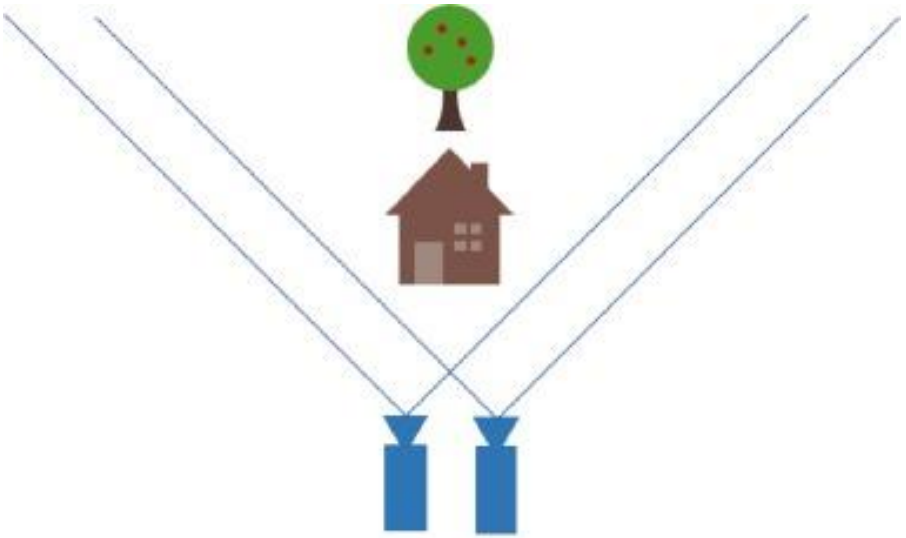
❖ **We generate 3D information using disparity maps obtained from stereo matching algorithm**

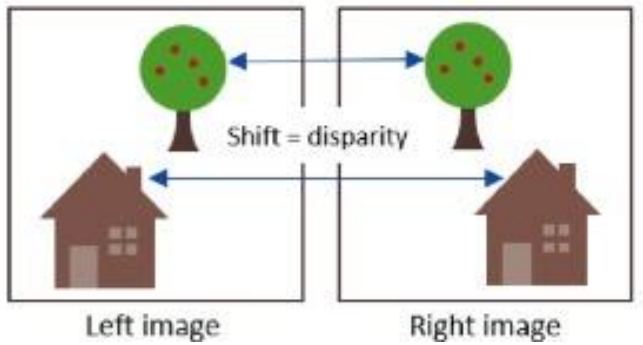


Stereo Matching Algorithm

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

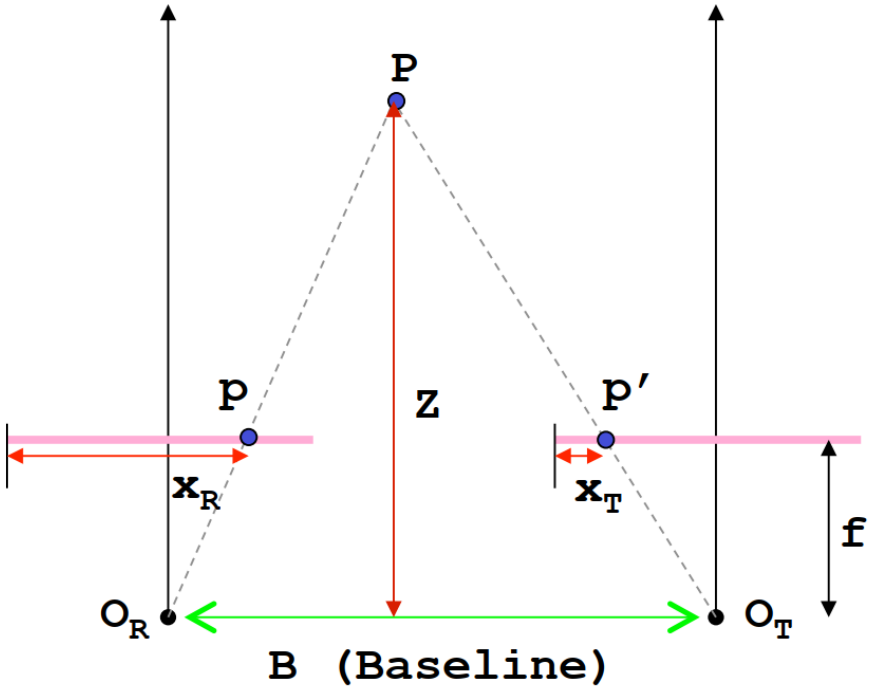


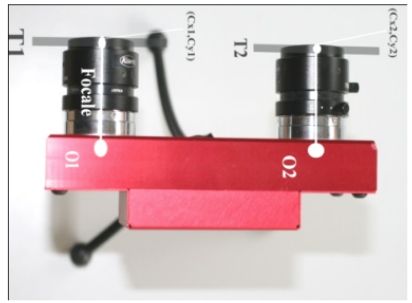




Disparity and depth

http://vision.deis.unibo.it/~smatt/Seminars/StereoVision.pdf





With the stereo rig in standard form and by considering similar triangles (PO_RO_T and Ppp'):

$$\frac{b}{Z} = \frac{(b + x_T) - x_R}{Z - f} \rightarrow Z = \frac{b \cdot f}{x_R - x_T} = \frac{b \cdot f}{d}$$

$x_R - x_T$ is the **disparity**

Stereo Matching

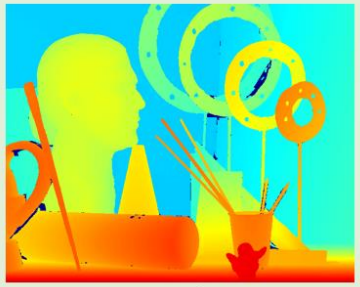
Stereo Matching / Binocular Depth Estimation

Hình từ camera trái

Hình từ camera phải

Thuật toán stereo matching

Hình chiều sâu cho hình trái



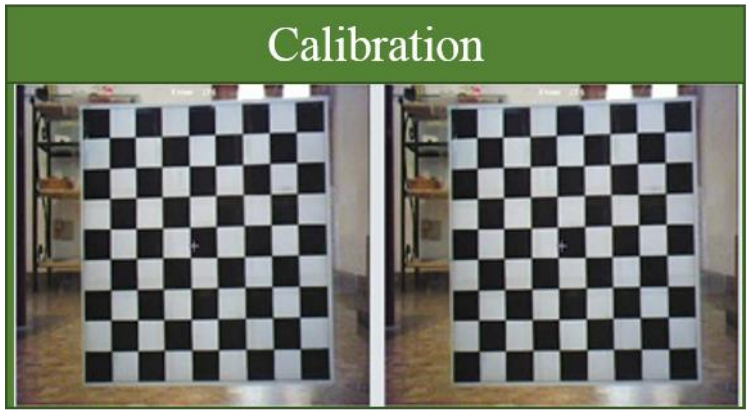
Hình chiều sâu cho hình phải

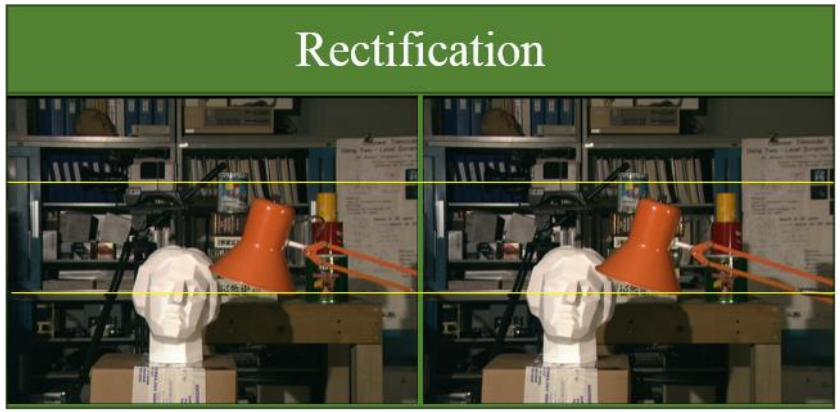
Stereo camera

Ứng dụng

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

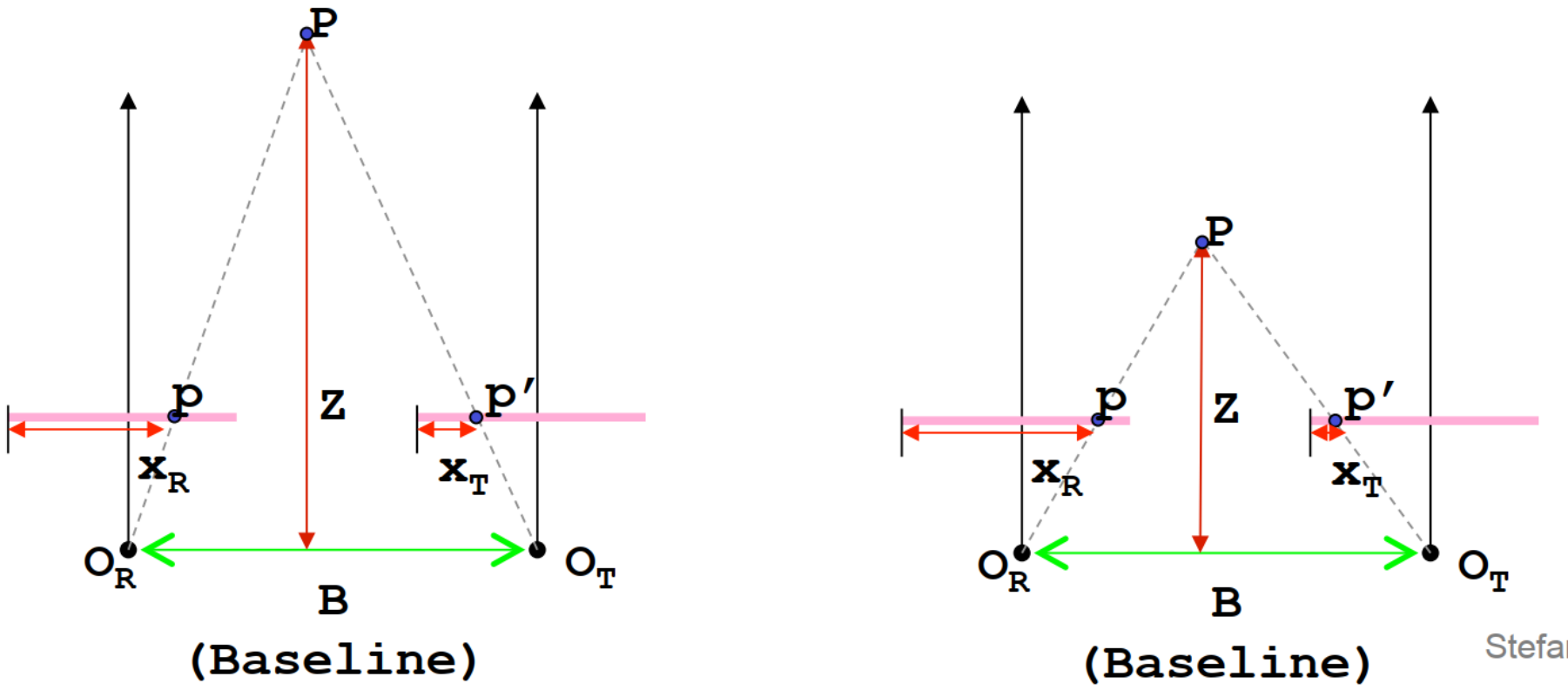
Stereo Matching





Stereo Matching

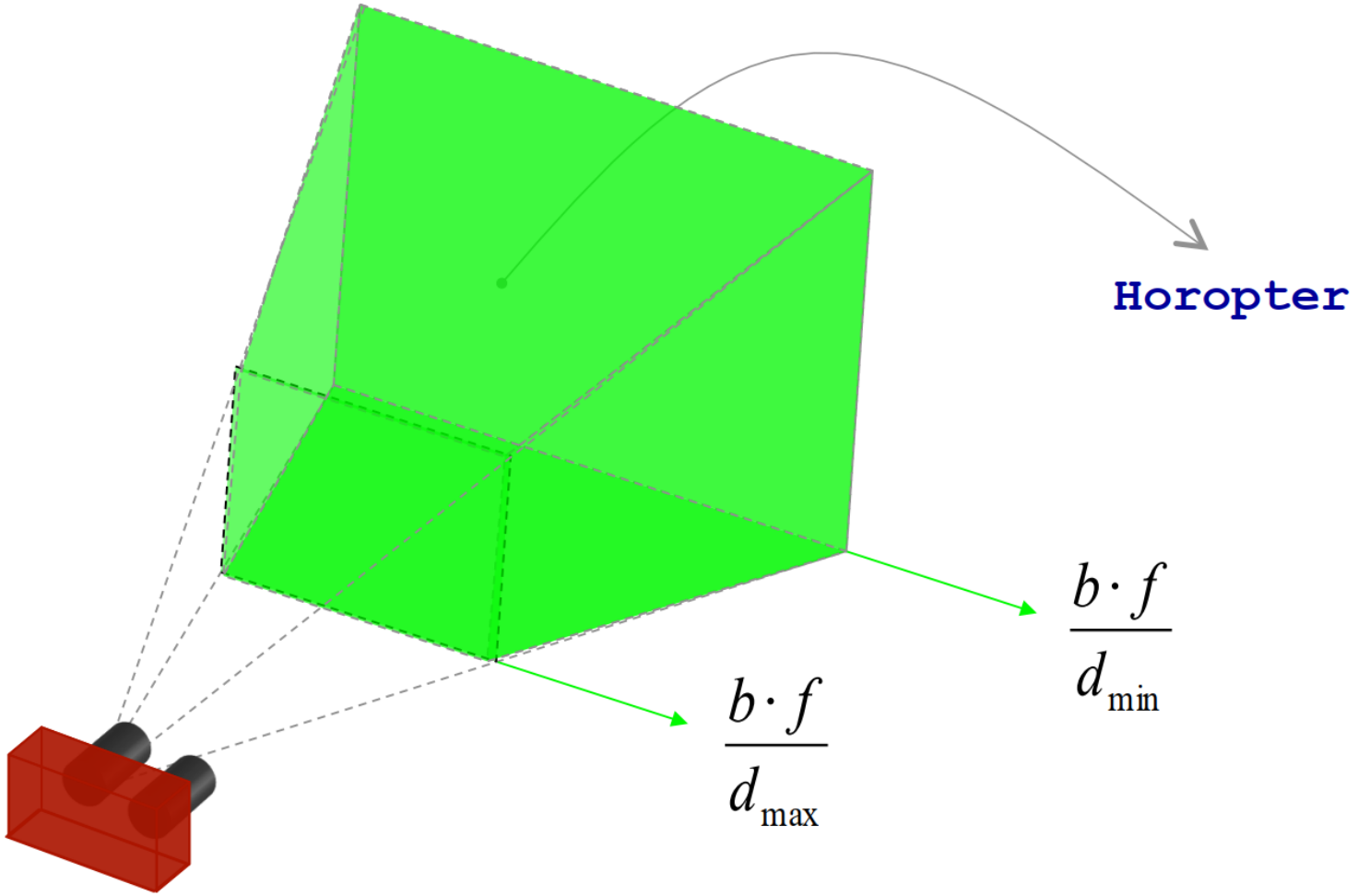
Disparity is higher for points closer to the camera



Stefano Mattoccia

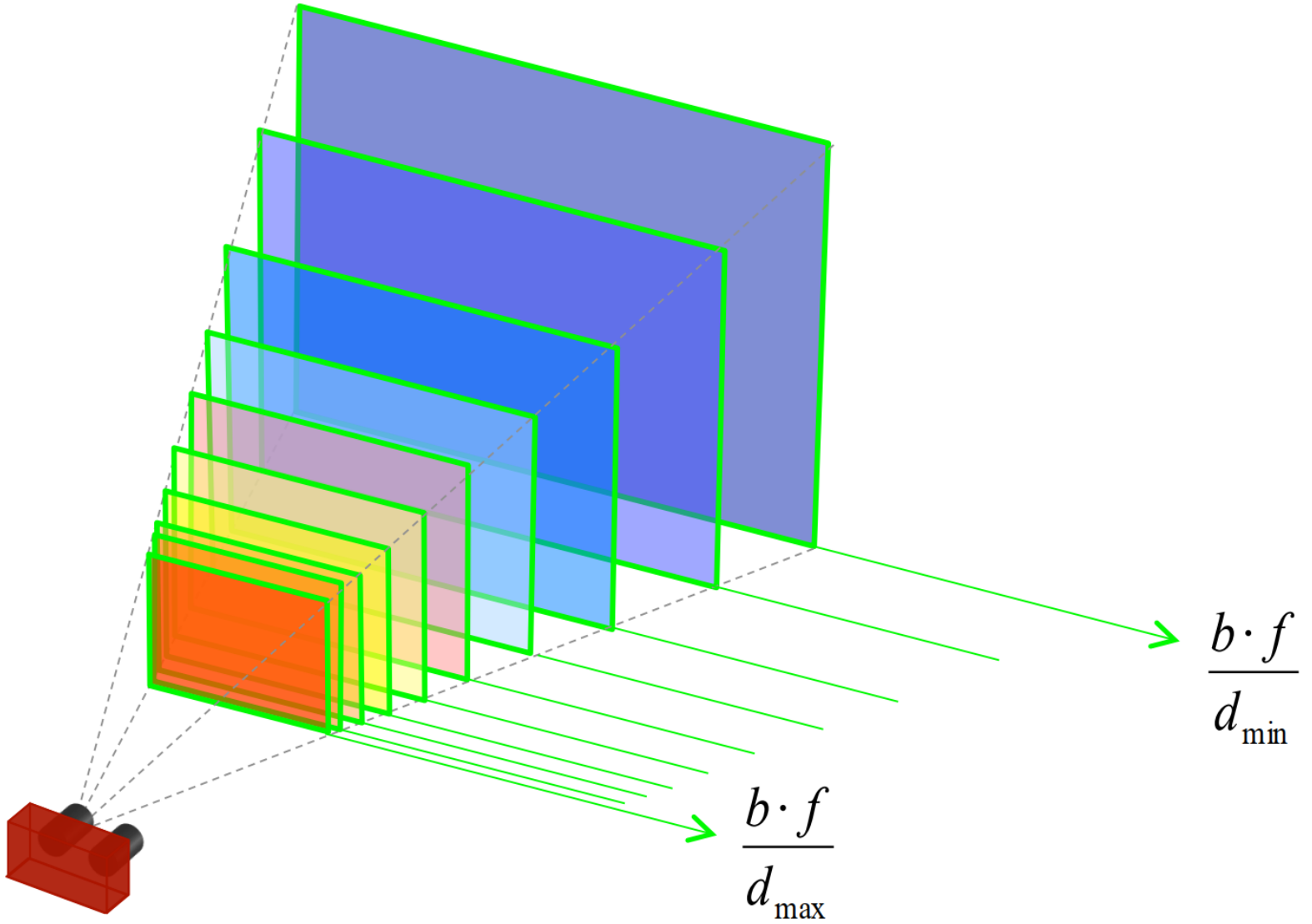
http://vision.deis.unibo.it/~smatt/Seminars/StereoVision.pdf

Stereo Matching

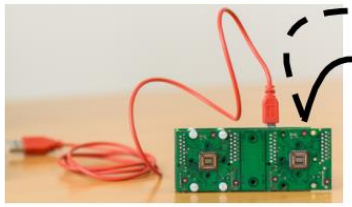


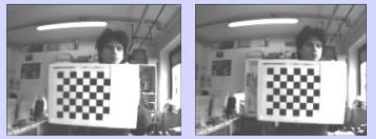
http://vision.deis.unibo.it/~smatt/Seminars/StereoVision.pdf

Stereo Matching



Overview of a stereo vision system





Calibration (offline)

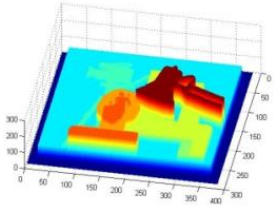
Intrinsic and extrinsic parameters

Stereo pair

Rectified stereo pair

Disparity map

Depth map

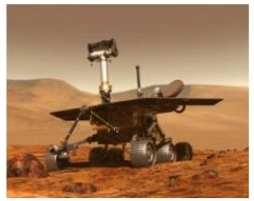


Rectification

Stereo Correspondence

Triangulation

PC, FPGA

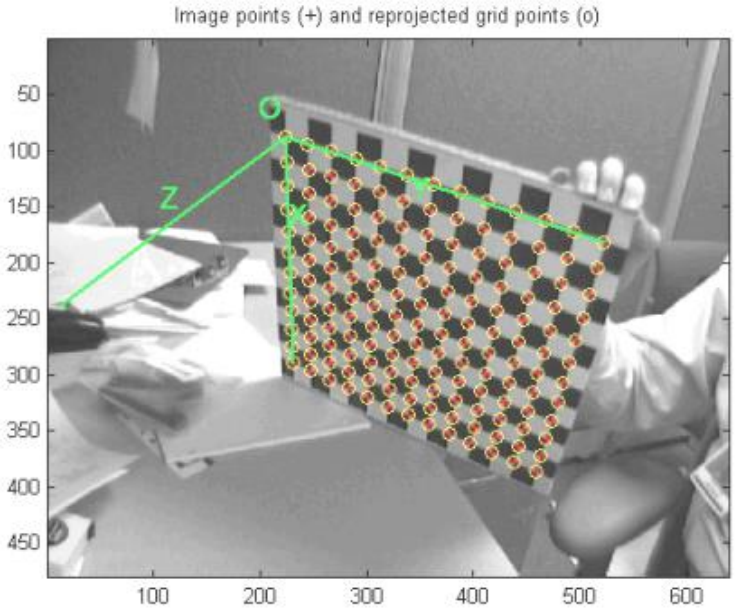


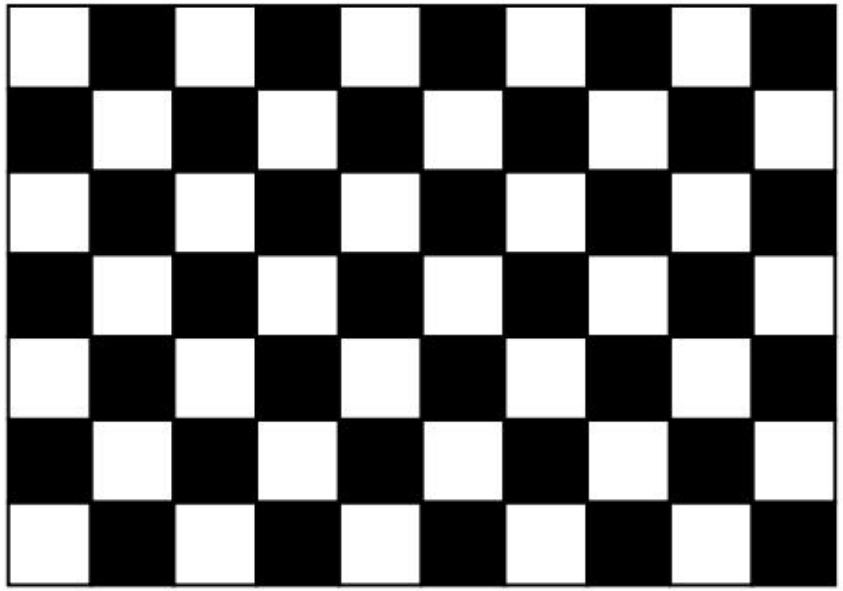
http://vision.deis.unibo.it/~smatt/Seminars/StereoVision.pdf

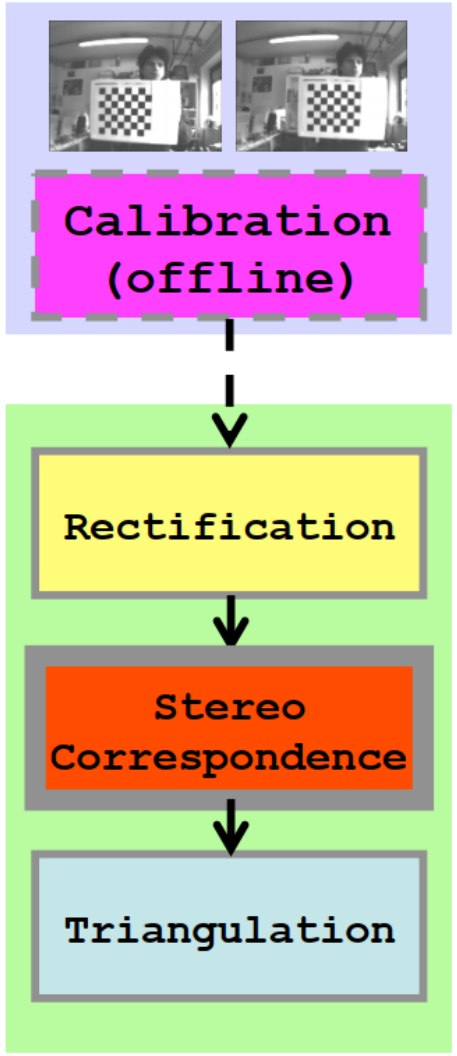
Stefano Mattoccia

Stereo Matching

❖ Calibration



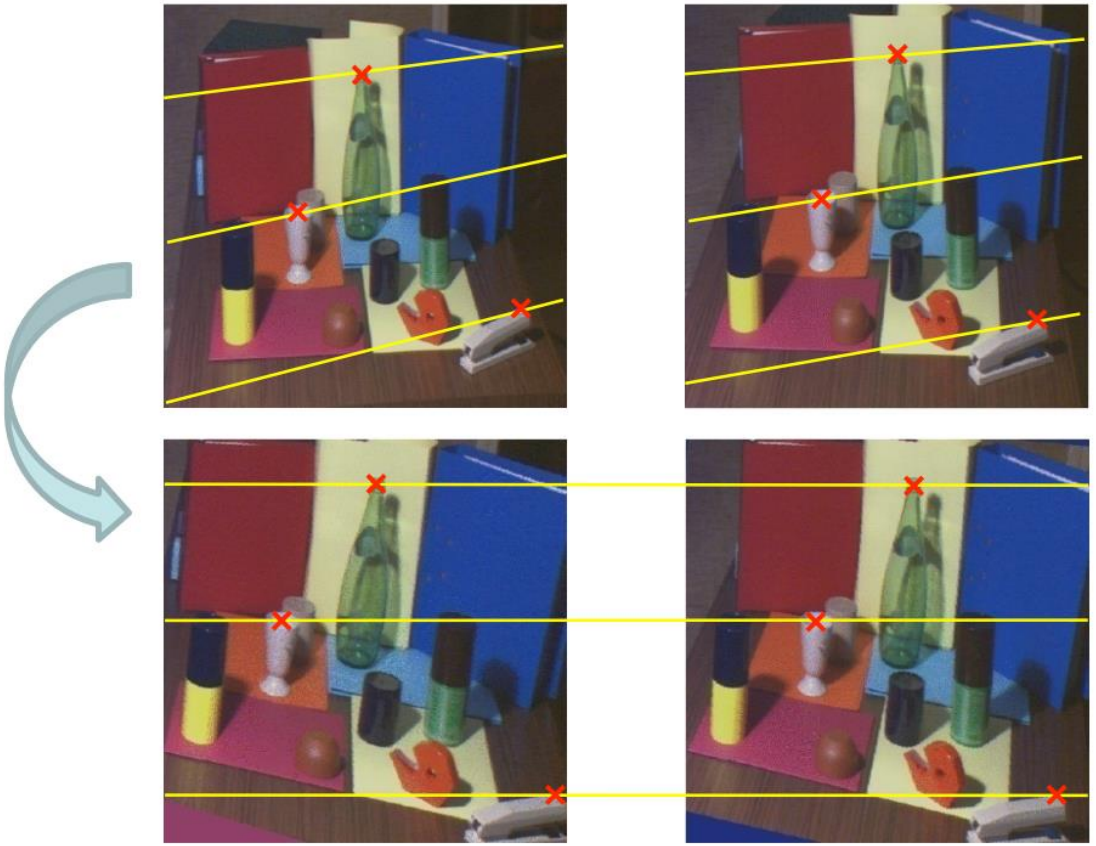


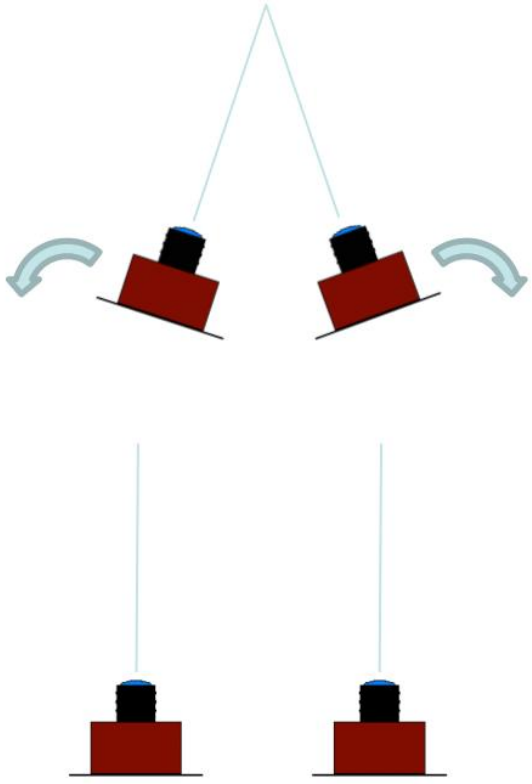


http://vision.deis.unibo.it/~smatt/Seminars/StereoVision.pdf

Stereo Matching

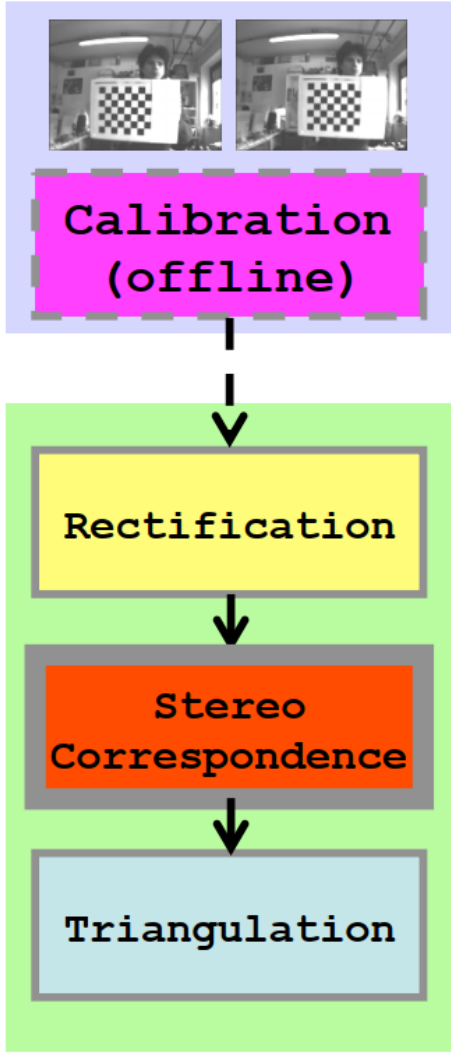
❖ Rectification



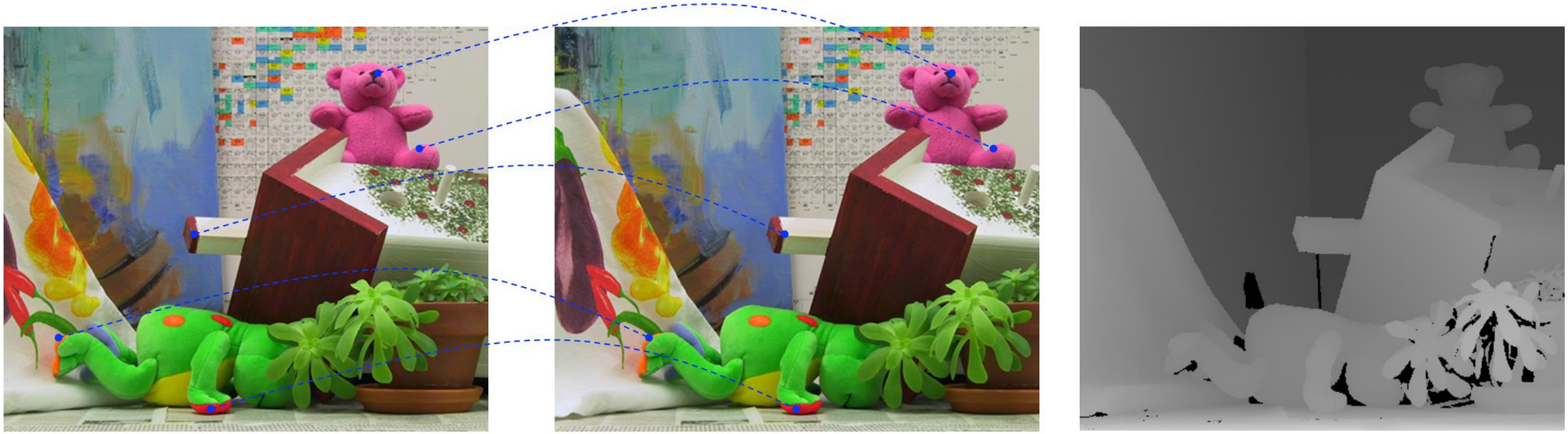


Stereo camera in standard form

Stefano Mattoccia



Stereo Matching



http://vision.deis.unibo.it/~smatt/Seminars/StereoVision.pdf

Stereo Matching

❖ Challenges

Photometric distortions and noise

http://vision.deis.unibo.it/~smatt/Seminars/StereoVision.pdf

Stereo Matching

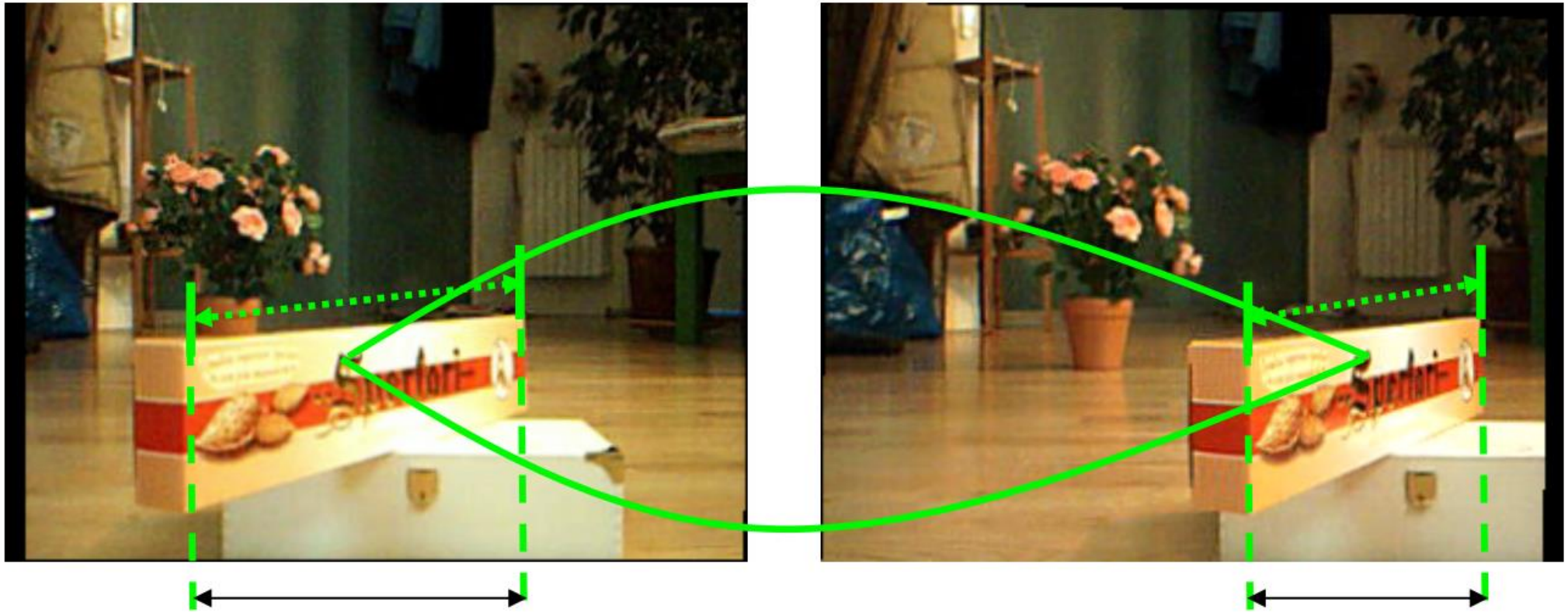
❖ Challenges

Specular surfaces

http://vision.deis.unibo.it/~smatt/Seminars/StereoVision.pdf

Stereo Matching

- **Challenges**
  - **Foreshortening**



http://vision.deis.unibo.it/~smatt/Seminars/StereoVision.pdf

Stereo Matching

❖ Challenges

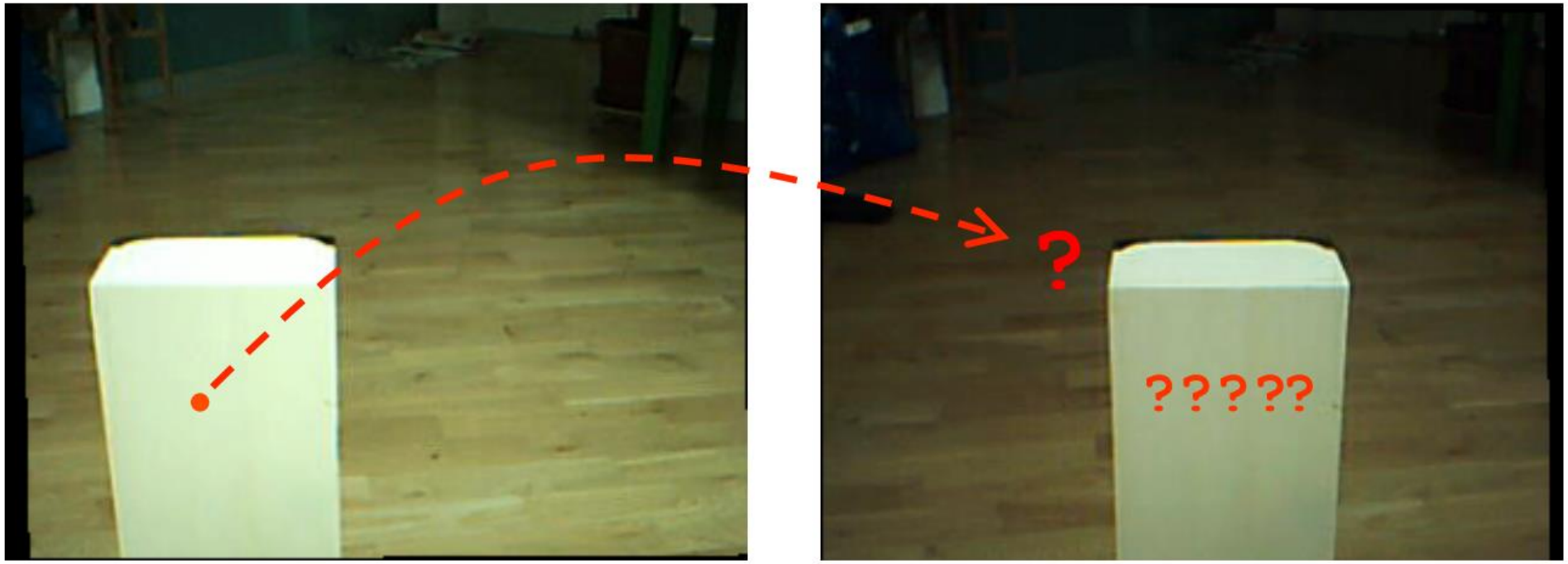
Perspective distortions

http://vision.deis.unibo.it/~smatt/Seminars/StereoVision.pdf

Stereo Matching

❖ Challenges

Uniform/ambiguous regions

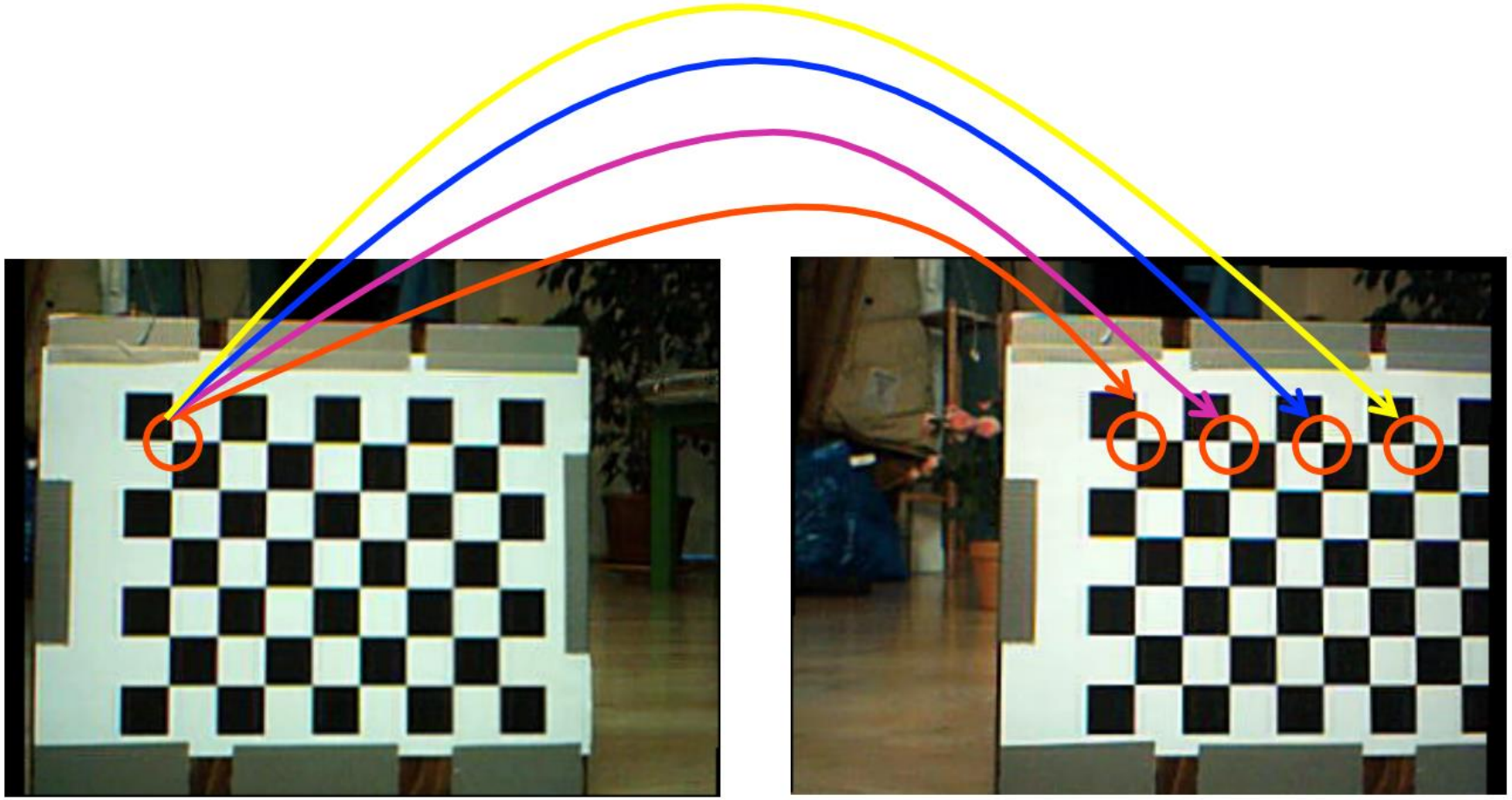


http://vision.deis.unibo.it/~smatt/Seminars/StereoVision.pdf

Stereo Matching

❖ **Challenges**

Repetitive/ambiguous patterns



http://vision.deis.unibo.it/~smatt/Seminars/StereoVision.pdf

Stereo Matching

❖ Challenges

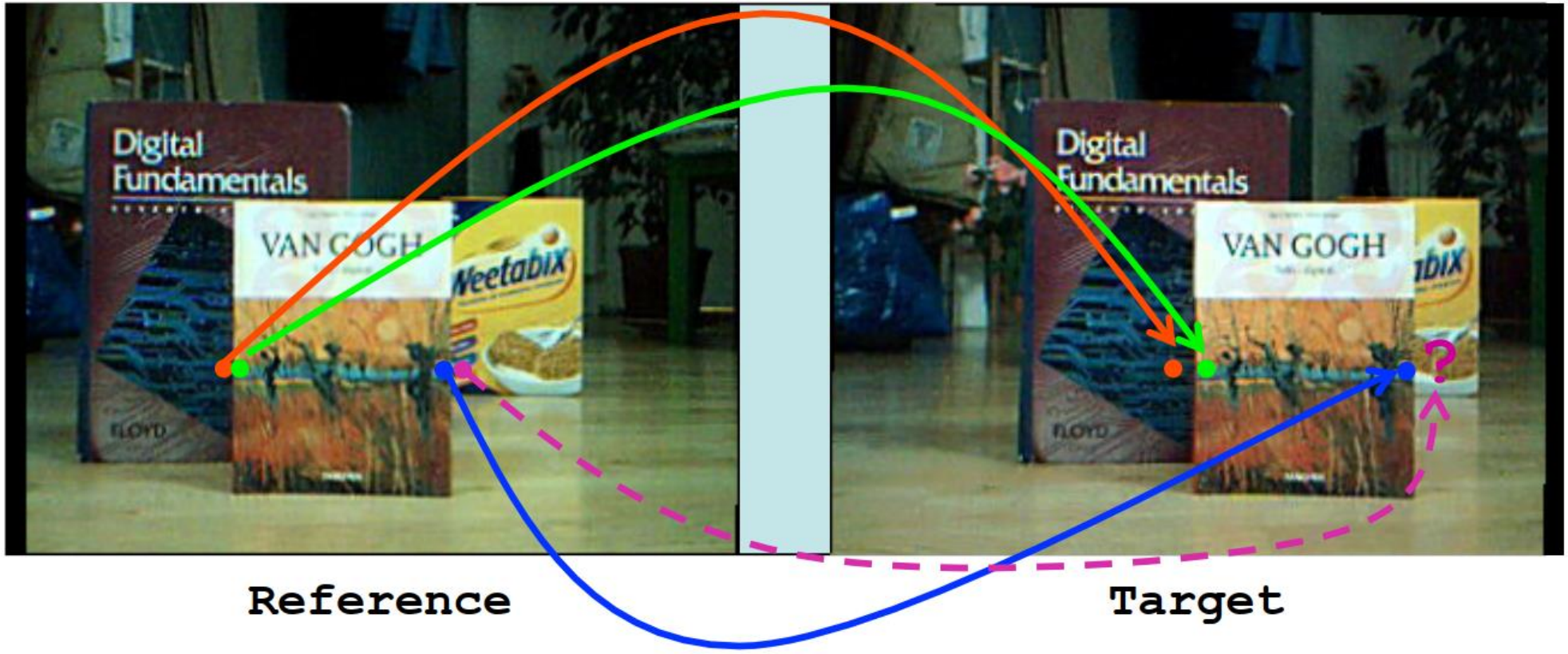
Transparent objects

http://vision.deis.unibo.it/~smatt/Seminars/StereoVision.pdf

Stereo Matching

❖ Challenges

Occlusions and discontinuities



http://vision.deis.unibo.it/~smatt/Seminars/StereoVision.pdf

Stereo Matching

❖ **Left and right images of a stereo pair**

Left Image

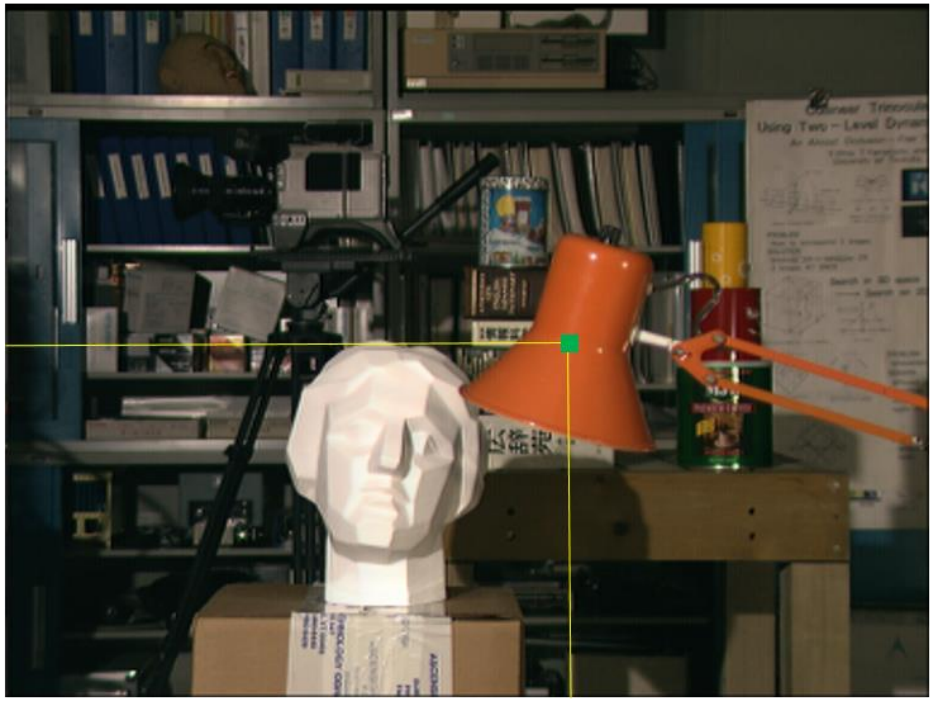
Right Image

Stereo Matching

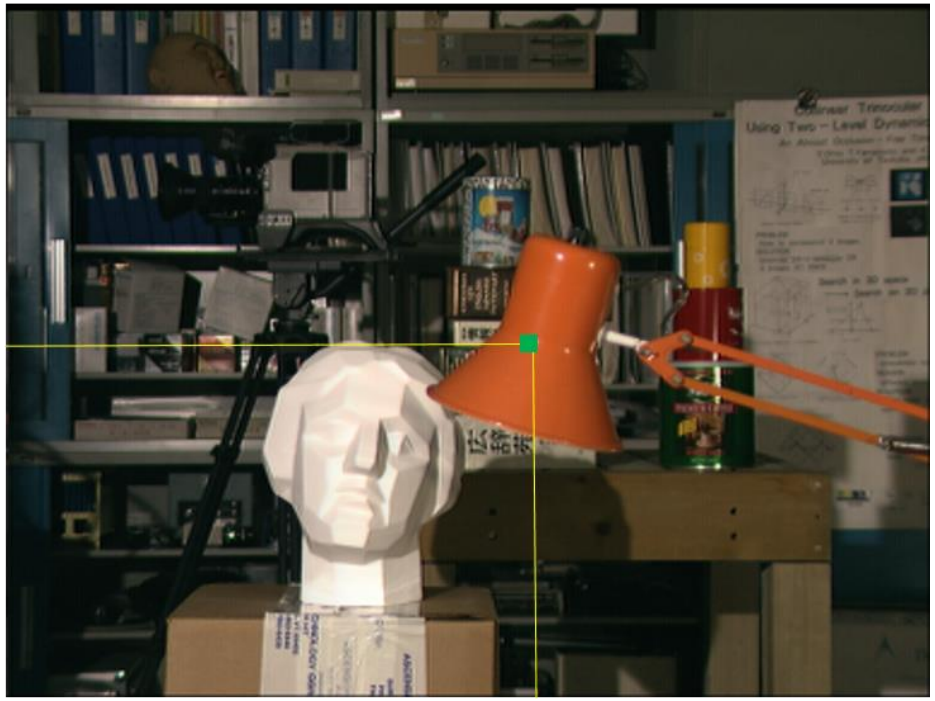
p = (234, 140)

q = (220, 140)

y_p=140



y_q=140



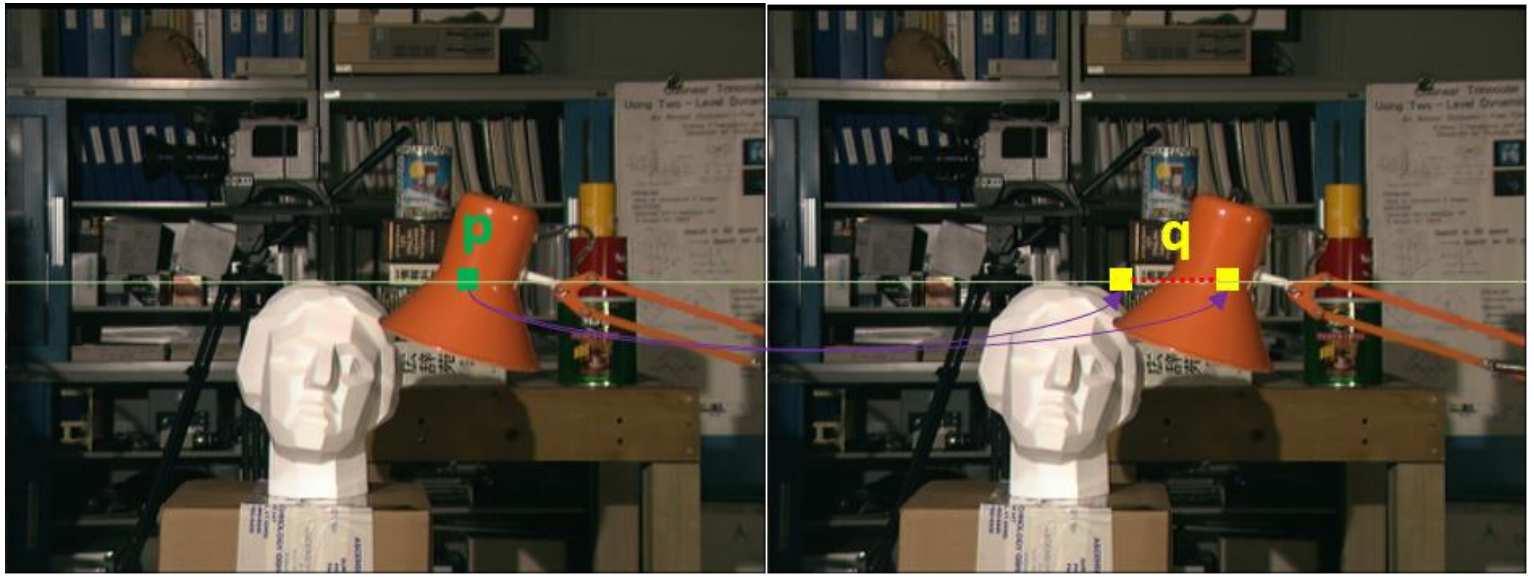
x_p = 234

x_q = 220

disparity_q = x_p – x_q = 14

Stereo Matching

❖ Method 1



Left Image Right Image

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

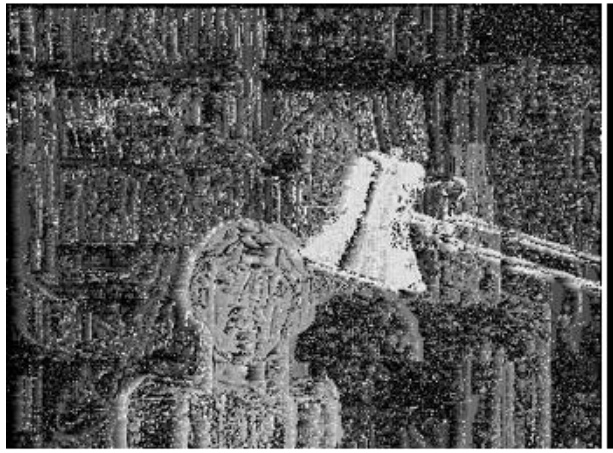
$$where \;\; C(p,q) = (L(p) - R(q))^2$$

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

Stereo Matching

Left Image

Right Image

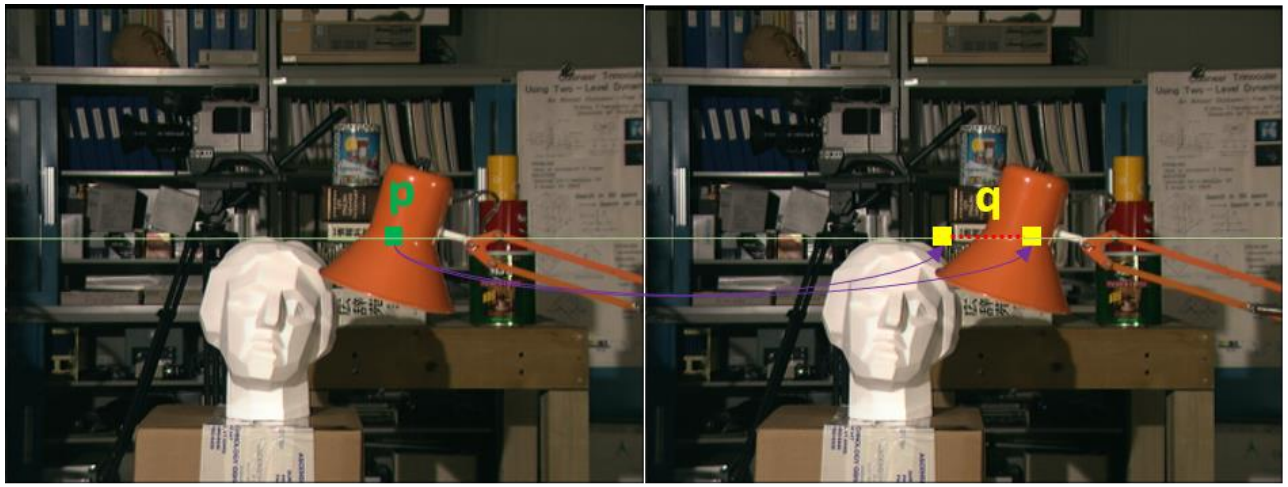


Disparity Map

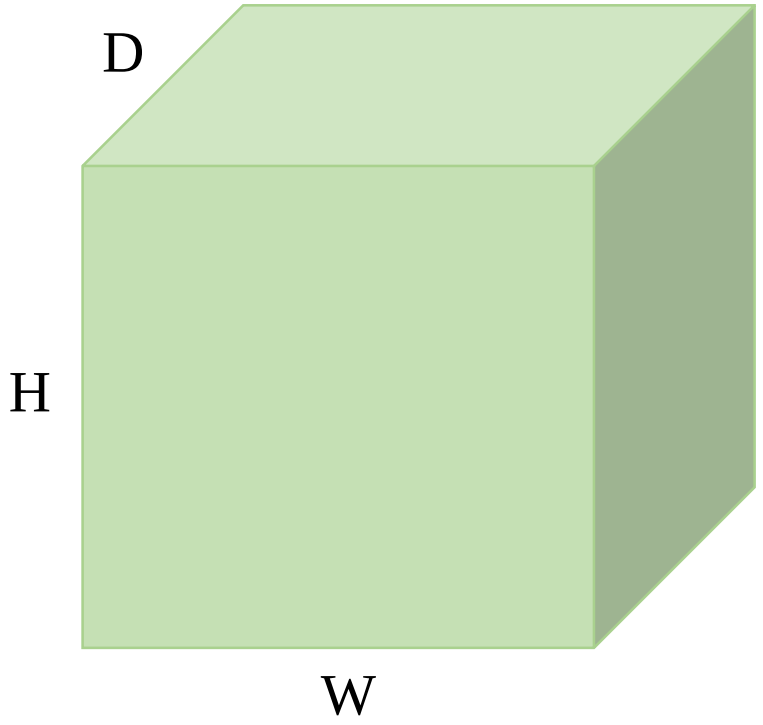
Ground Truth

Stereo Matching

❖ Compute the cost volume C



Left Image Right Image

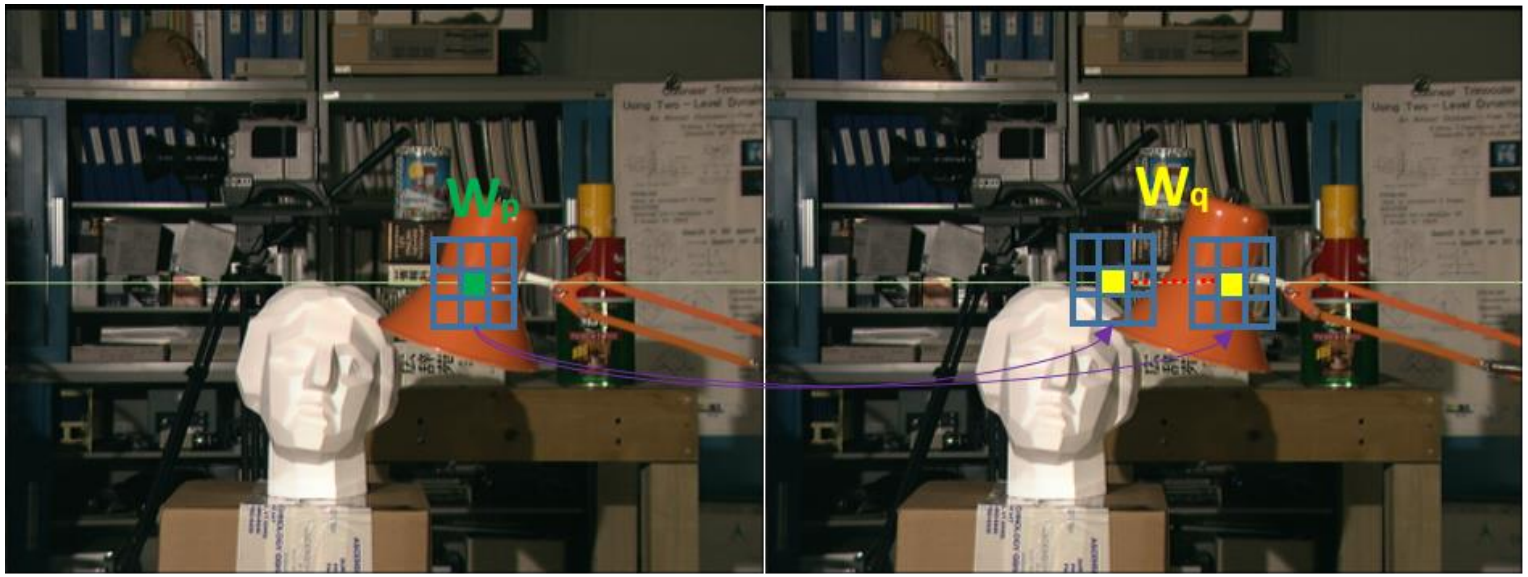


$$\boldsymbol{C}(x, y, d) = |L(x, y) - R(x - d, y)|$$

Stereo Matching

Stereo Matching

❖ Method 2



Left Image Right Image

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

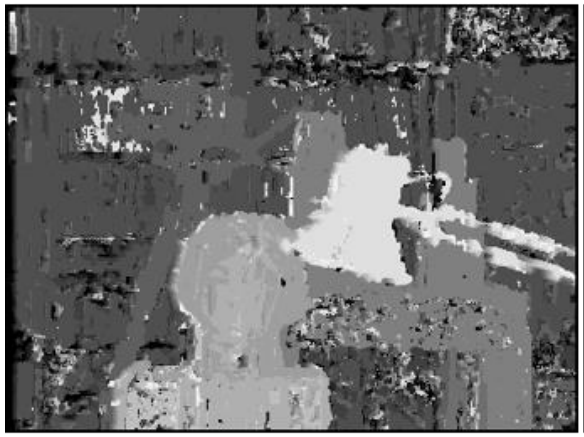
$$where \;\; C(p,q) = \sum_{(u,v)\in(W_p,W_q)} (L(u) - R(v))^2$$

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

Stereo Matching

Left Image

Right Image



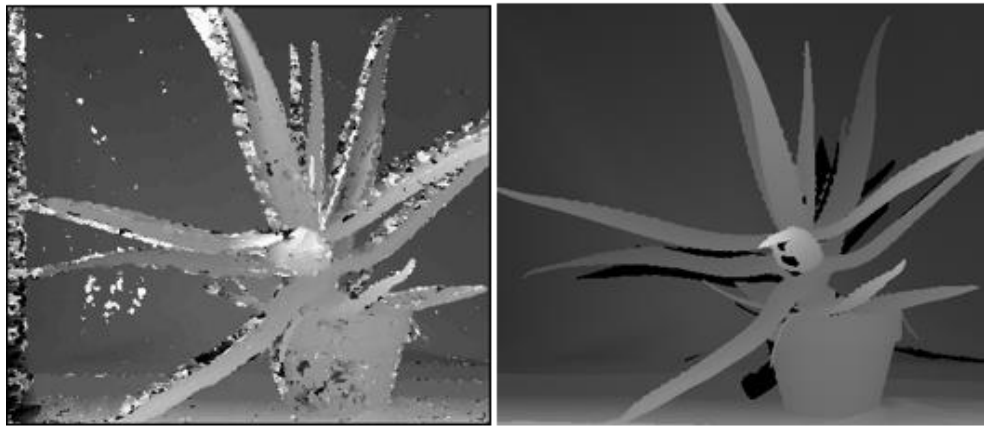
Disparity Map

Ground Truth

Stereo Matching

❖ Aloe stereo pair

Left Image Right Image



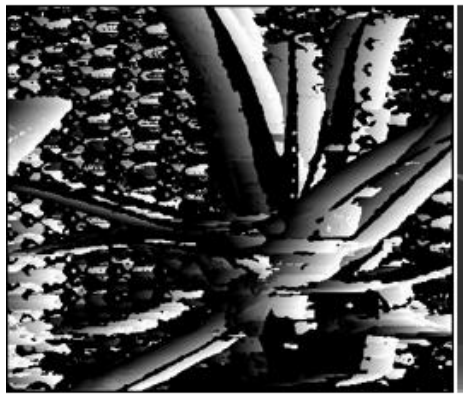
Disparity Map Ground Truth

Stereo Matching

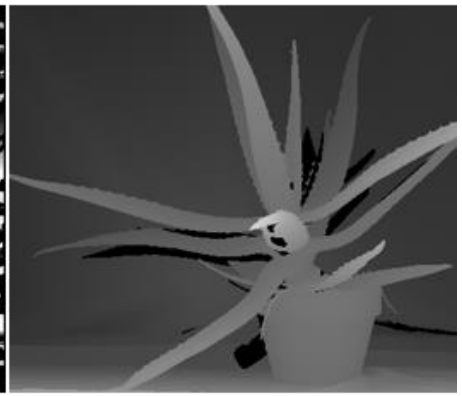
❖ Aloe stereo pair

Left Image

Right Image



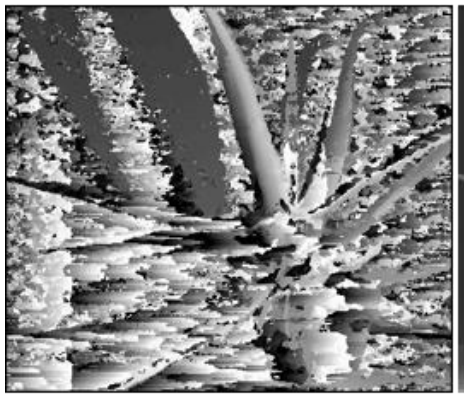
Disparity Map



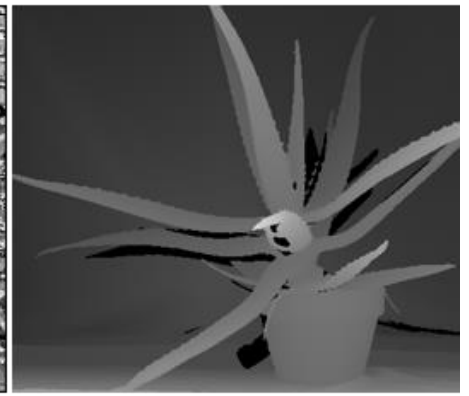
Ground Truth

Left Image

Right Image

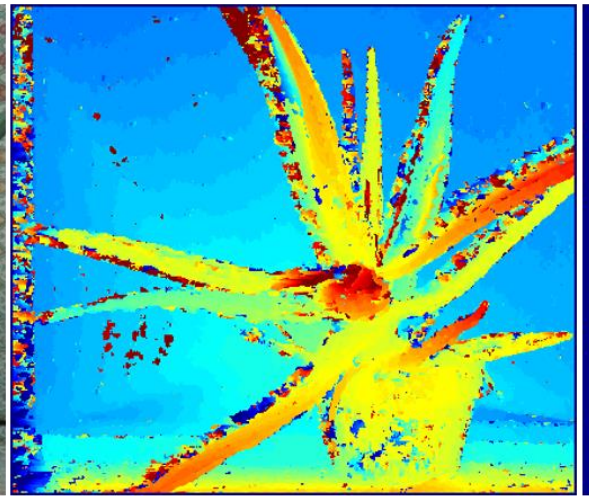


Disparity Map

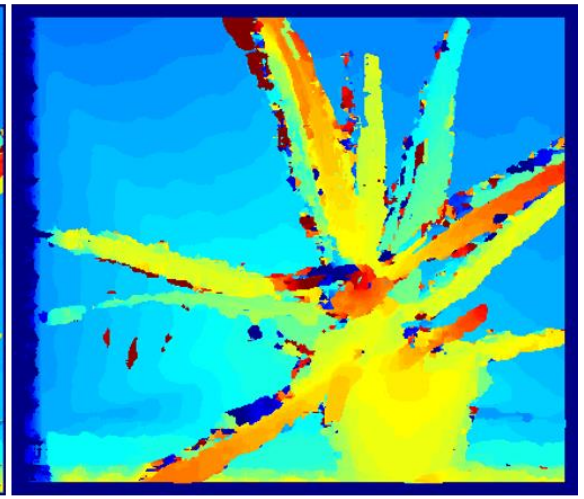


Ground Truth


Ảnh stereo có cùng độ sáng

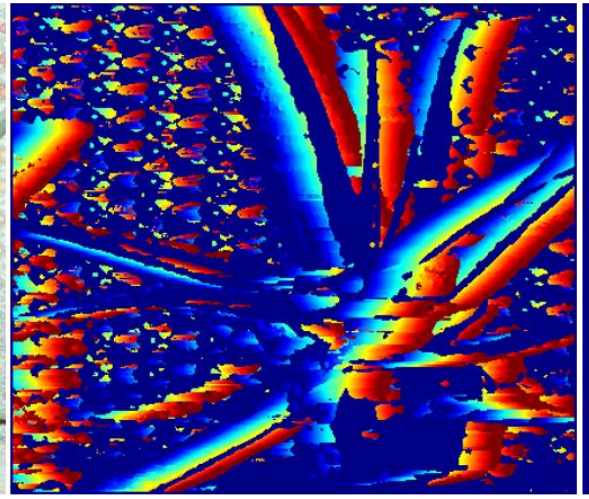


Absolute difference

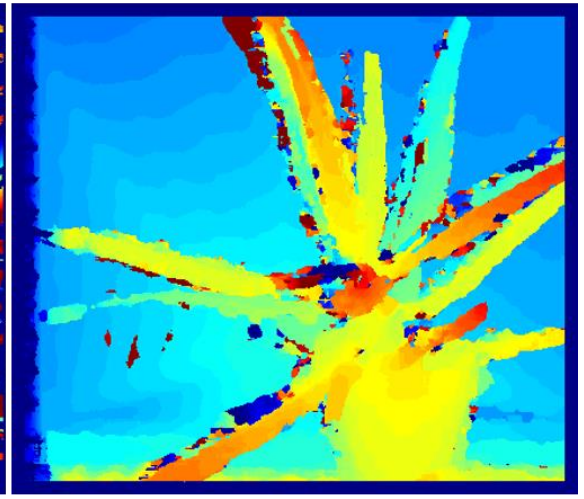


Cosine Similarity


Ảnh stereo khác độ sáng



Absolute difference

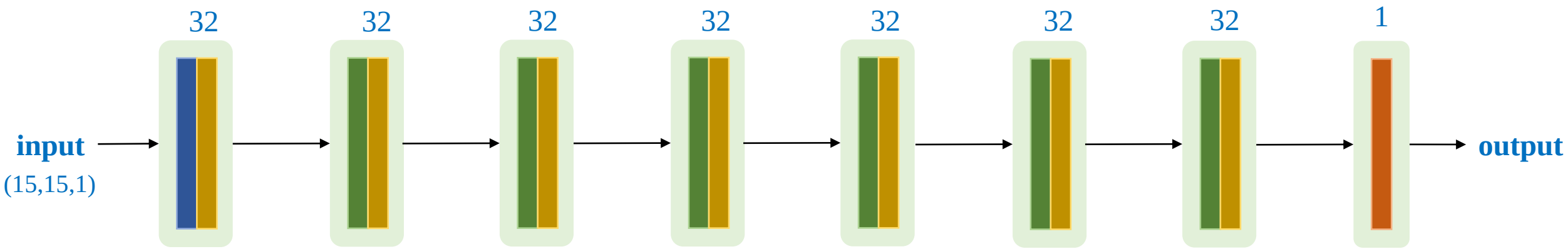


Cosine Similarity

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

Stereo Matching

❖ Using deep learning



Stereo Matching

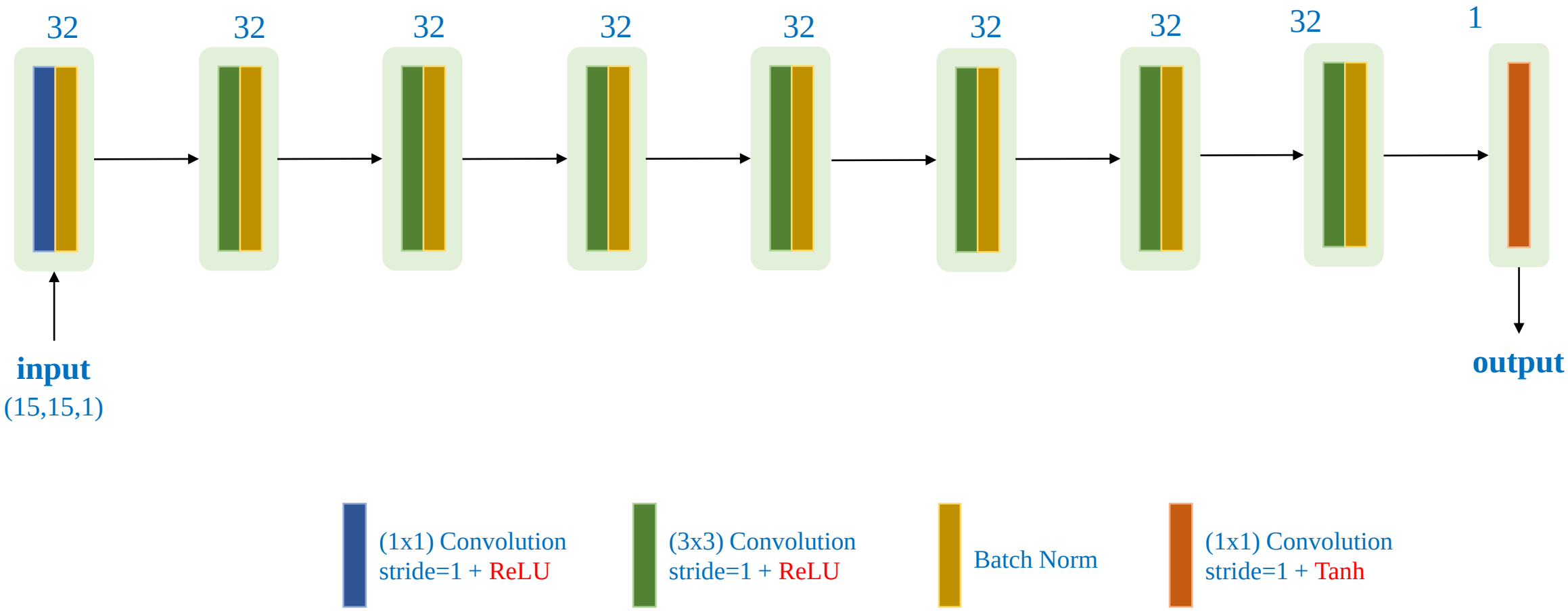
❖ Using deep learning

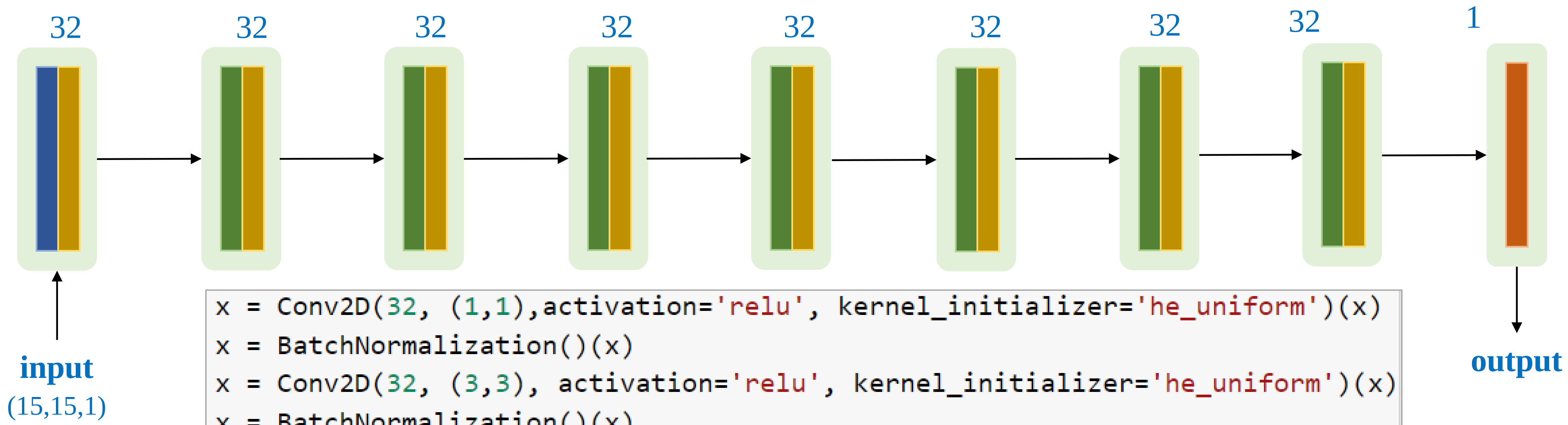
(a)

(b)

Stereo Matching

❖ Using deep learning



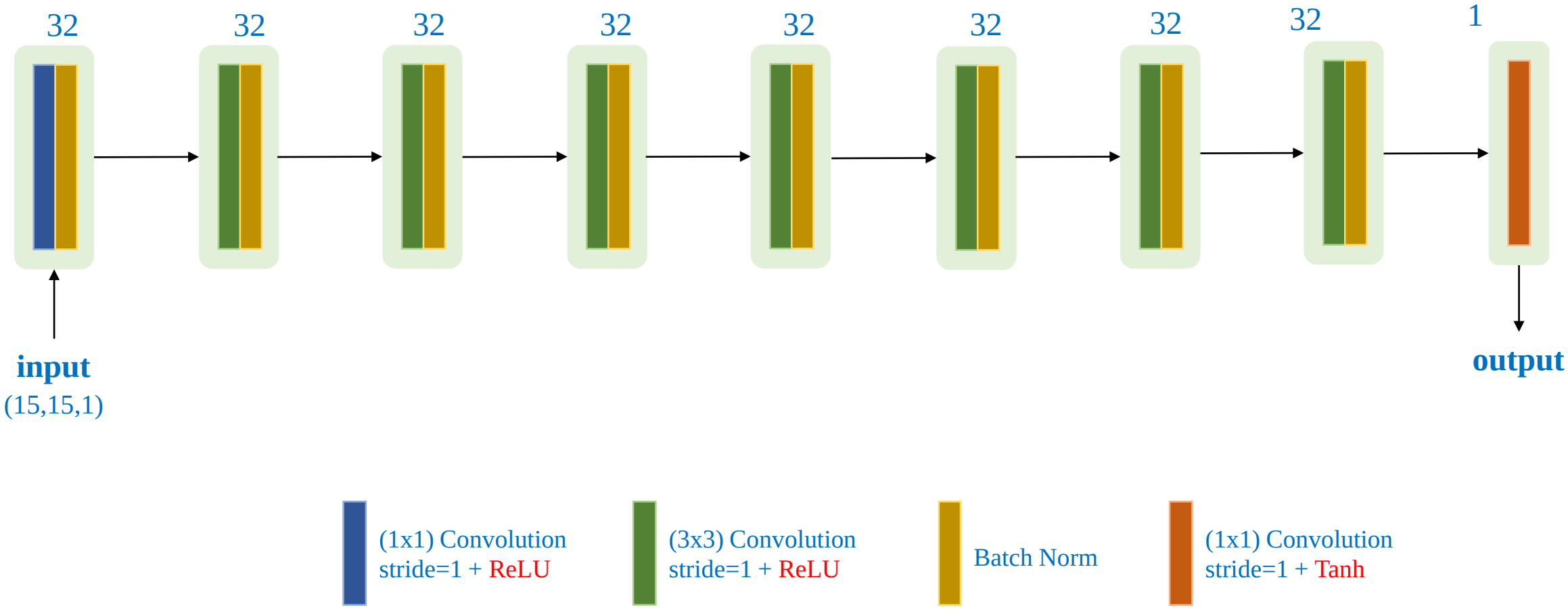



```
x = Conv2D(32, (1,1),activation='relu', kernel_initializer='he_uniform')(x)
x = BatchNormalization()(x)
x = Conv2D(32, (3,3), activation='relu', kernel_initializer='he_uniform')(x)
x = BatchNormalization()(x)
x = Conv2D(32, (3,3), activation='relu', kernel_initializer='he_uniform')(x)
x = BatchNormalization()(x)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

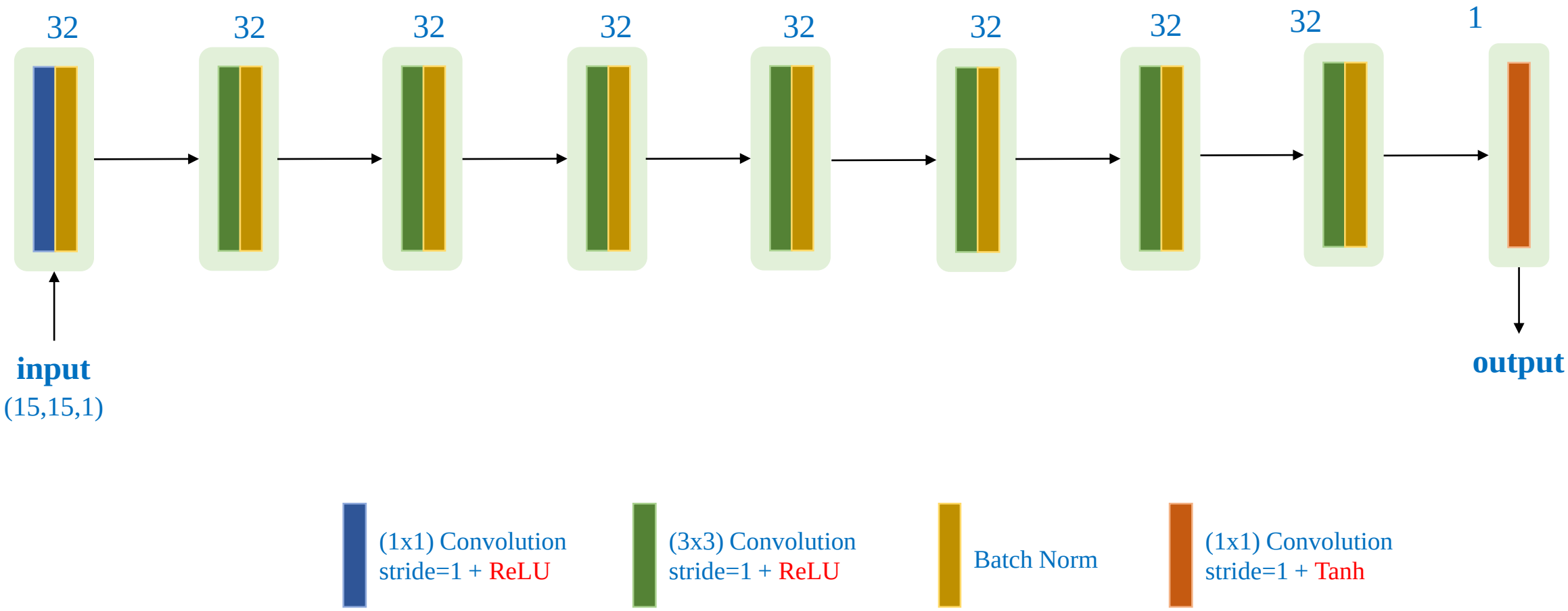
❖ Problem with shapes



Stereo Matching

❖ Implementation

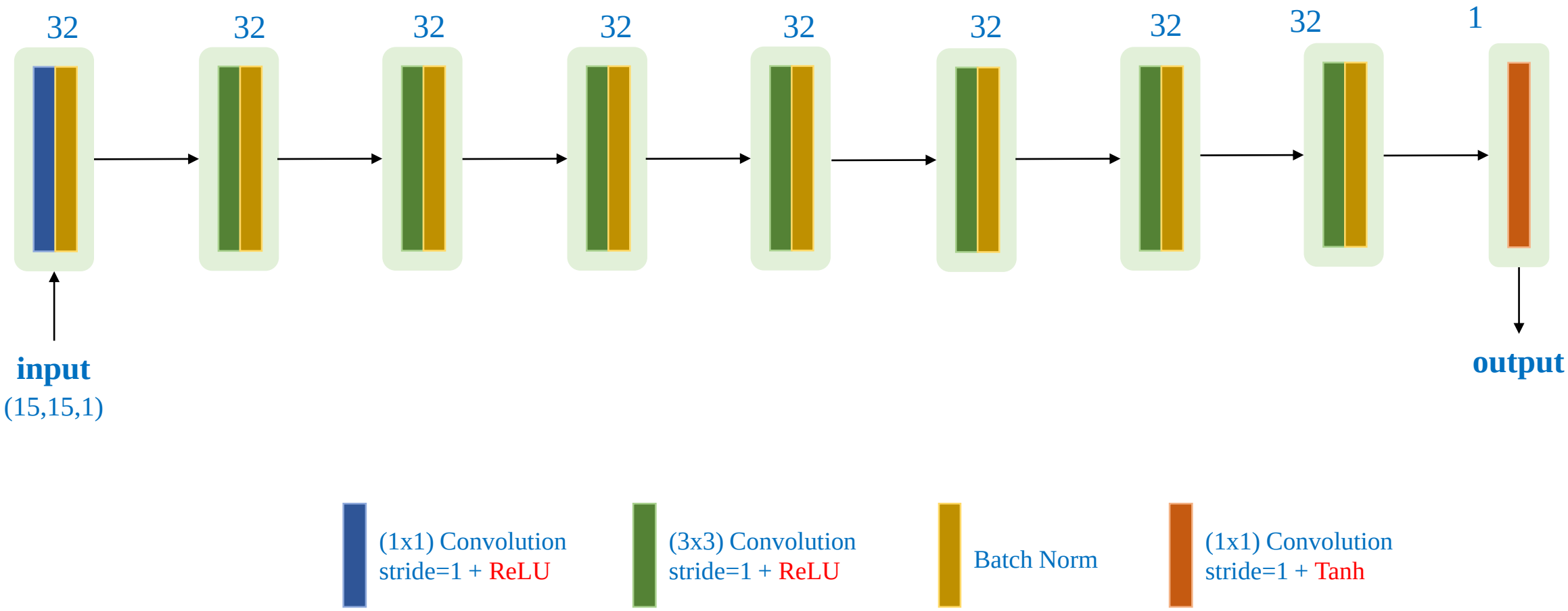
❖ Problem with fixed sizes



Stereo Matching

❖ Implementation

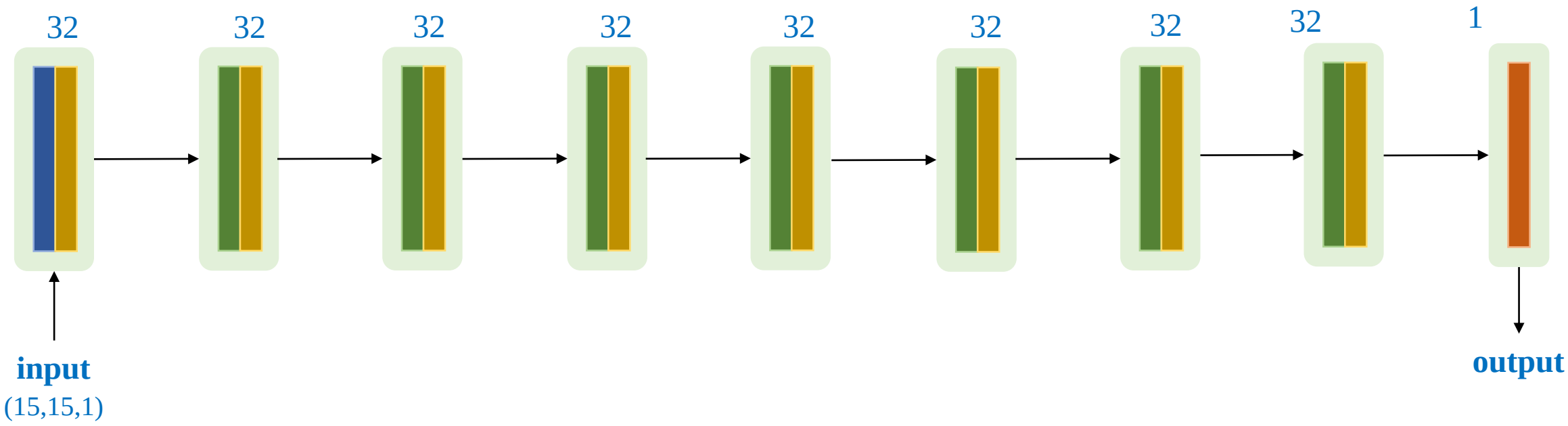
❖ Using the fit function



Stereo Matching

❖ Implementation

❖ Using GradientTape




Thank you!