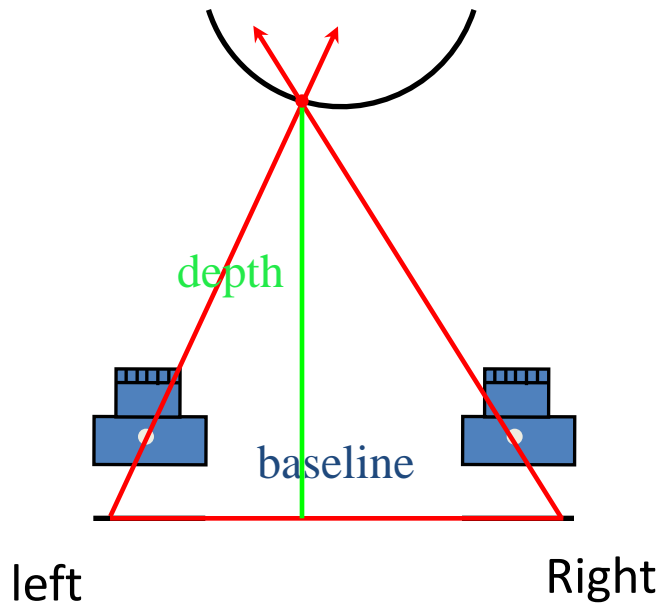


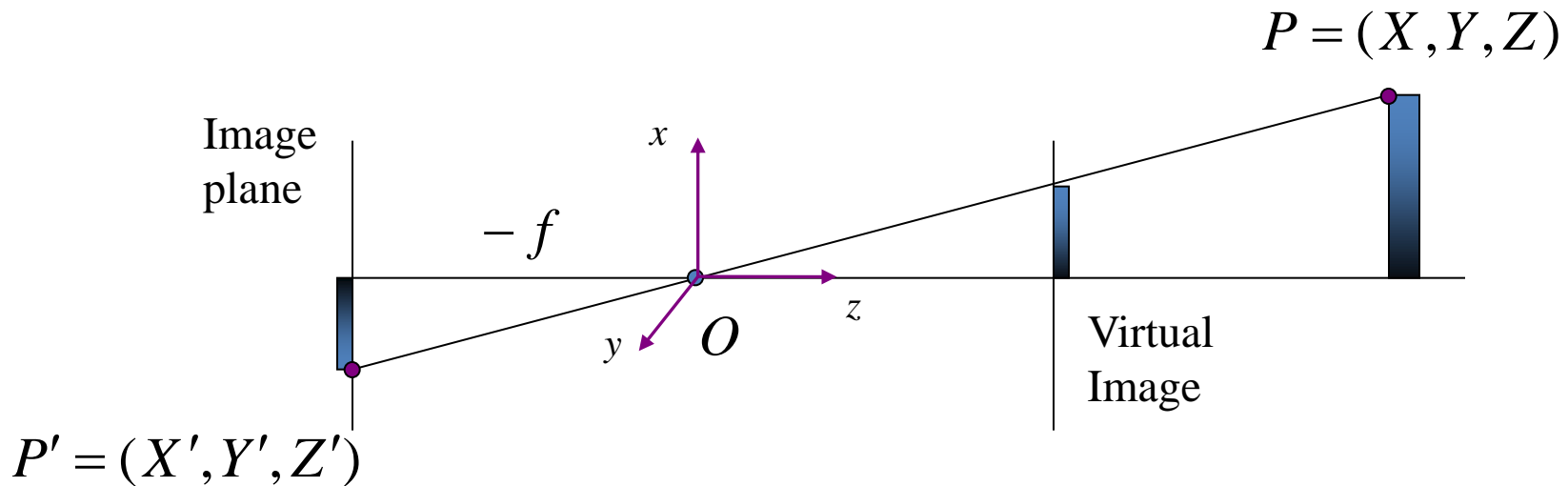
# Bài toán Tìm Tương Ứng Stereo

# Bài toán tìm tương ứng Stereo



*Thông tin ảnh của hai camera có tính chất giống nhau và chụp cùng một đối tượng có thể dùng để tái tạo độ sâu của ảnh*

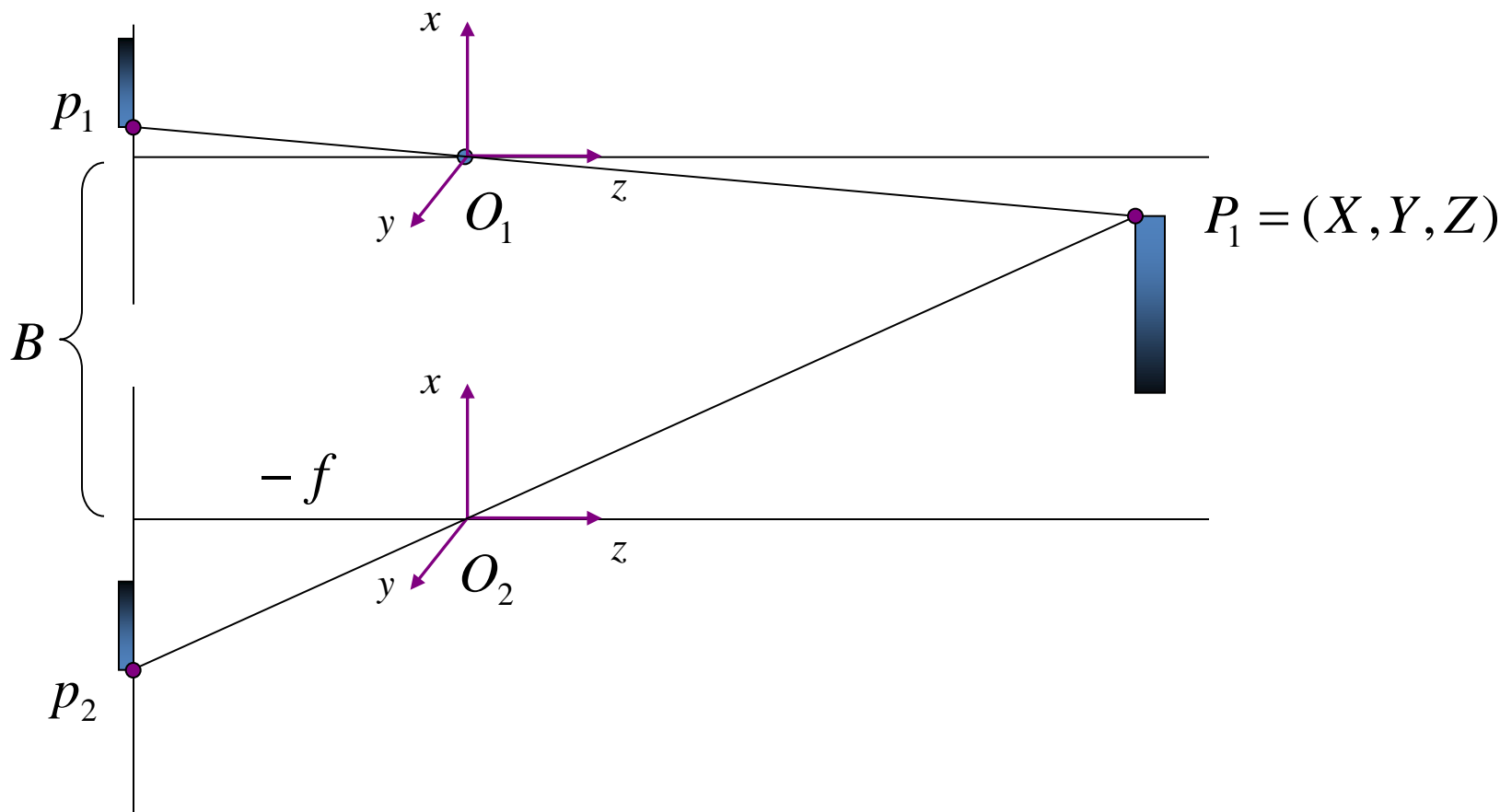
# Nhắc lại mô hình Pinhole camera



$$Z' = -f, \quad X' = -f \frac{X}{Z}, \quad Y' = -f \frac{Y}{Z}$$

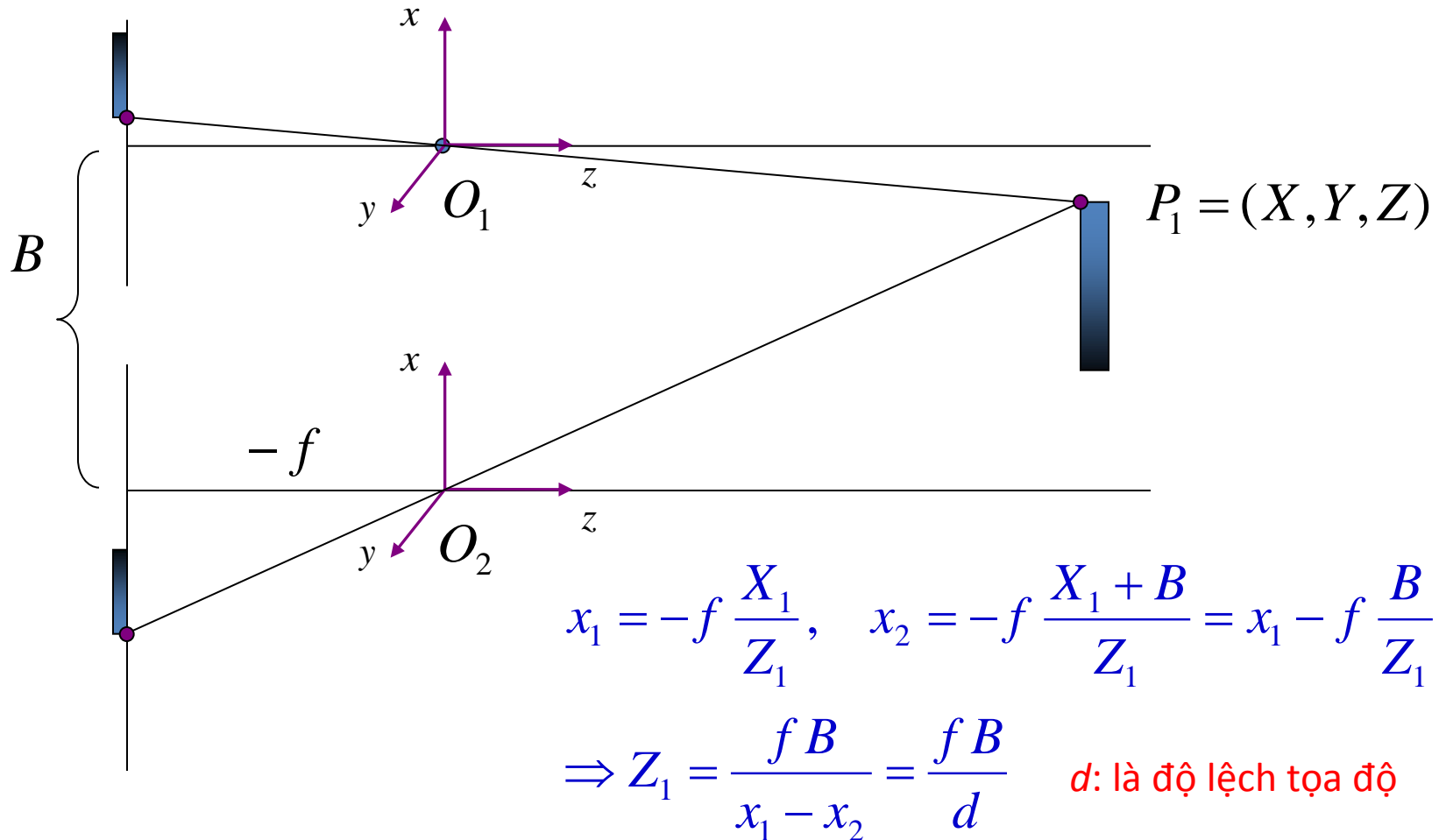
$$(X, Y, Z) \rightarrow \left(f \frac{X}{Z}, f \frac{Y}{Z}, Const\right)$$

# Mô hình ảnh stereo từ hai camera



Thông tin về chiều sâu  $z$  của ảnh có thể được rút ra từ  $x_1, x_2, f, B$

# Biến đổi trên hệ hai camera



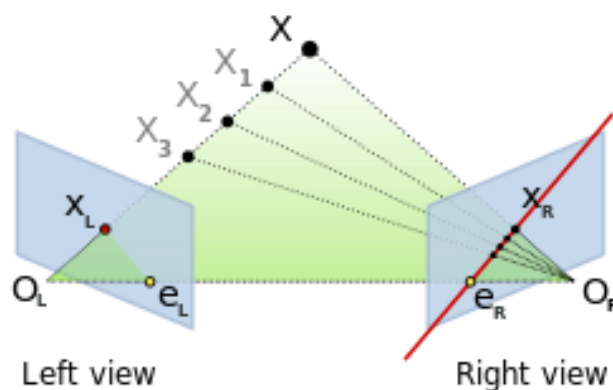
# Điều kiện để giải bài toán tương ứng stereo

- Màu sắc tương đồng
  - Màu của cùng một điểm trên thế giới thực là như nhau trong hai ảnh
  - Màu được tính theo mô hình *Lambertian*



# Điều kiện để giải bài toán tương ứng stereo

- Hình học epipolar
  - Thỏa các điều kiện của hình học epipolar
  - Dùng đường epipolar để hiệu chỉnh hai ảnh cho cùng giá trị trên trục  $y$



# Ảnh sau khi đã hiệu chỉnh

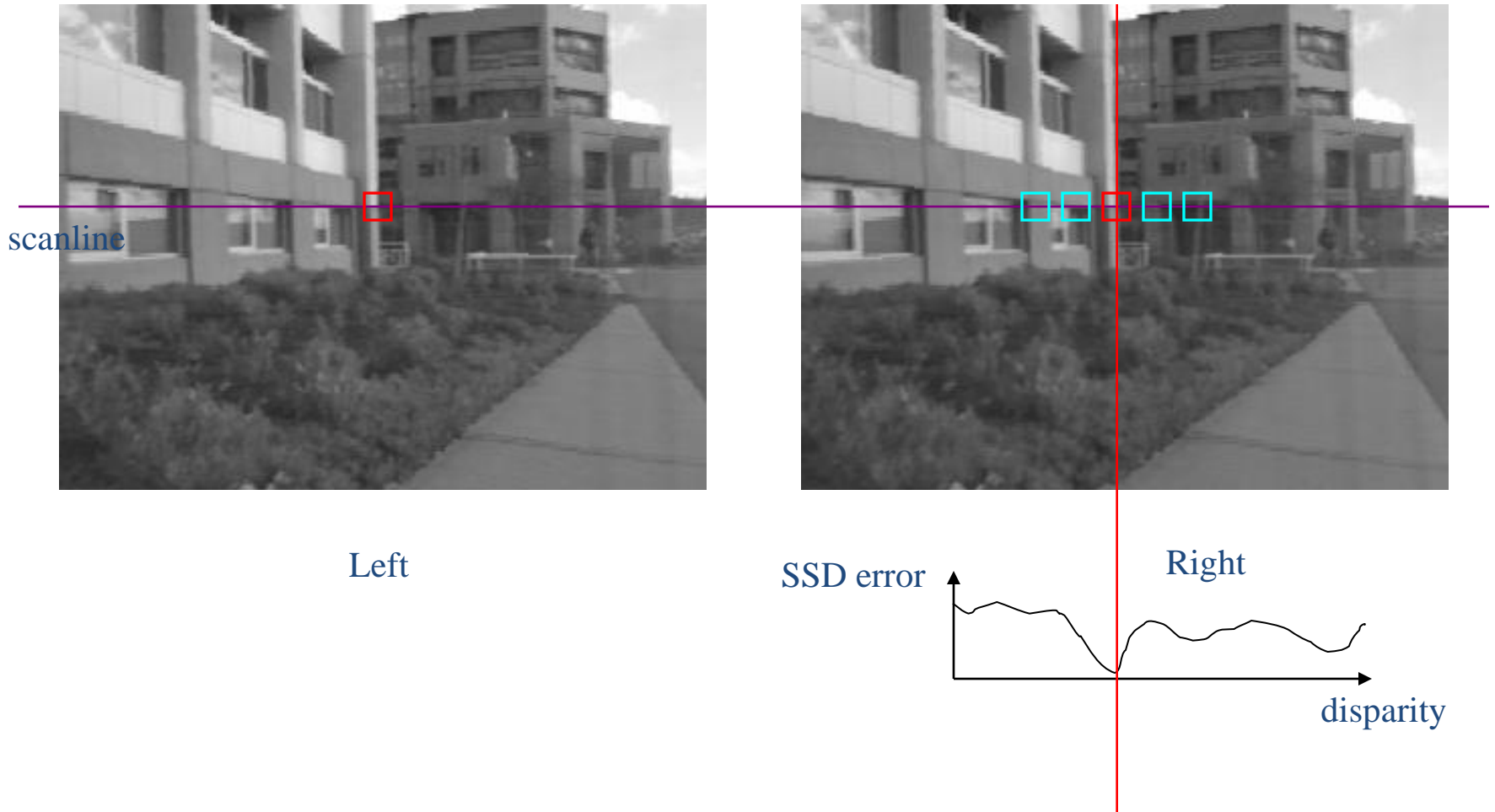




# Điều kiện để giải bài toán tương ứng stereo

- Duy nhất và liên tục (Marr&Poggio)
  - Mỗi điểm ảnh sẽ có thể có một giá trị độ lệch duy nhất và tập hợp các điểm độ lệch sẽ tạo ra một ảnh có độ trơn phù hợp với ảnh gốc.

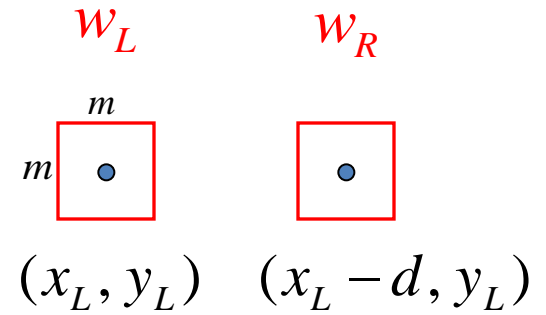
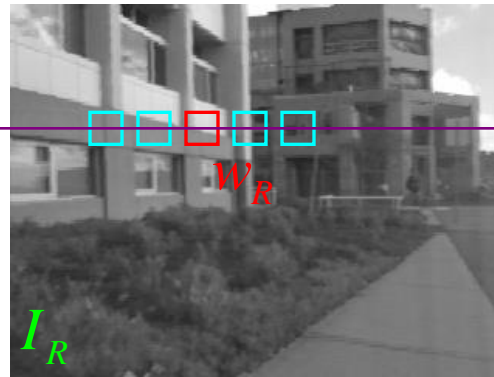
# Phương pháp quét theo dòng



# Độ đo SSD (Sum square of difference)

Left

Right



$w_L$  ,  $w_R$  : windows  $m \times m$  of pixels.

Window function:

$$W_m(x, y) = \{u, v \mid x - \frac{m}{2} \leq u \leq x + \frac{m}{2}, y - \frac{m}{2} \leq v \leq y + \frac{m}{2}\}$$

SSD cost measures:

$$C_r(x, y, d) = \sum_{(u,v) \in W_m(x,y)} [I_L(u, v) - I_R(u - d, v)]^2$$

# Chuẩn hóa ảnh

- Để thỏa tính chất màu không đổi, chúng ta cần tiến hành chuẩn hóa ảnh
- Một số phương pháp phổ biến

$$\bar{I} = \frac{1}{|W_m(x,y)|} \sum_{(u,v) \in W_m(x,y)} I(u,v)$$

Average pixel

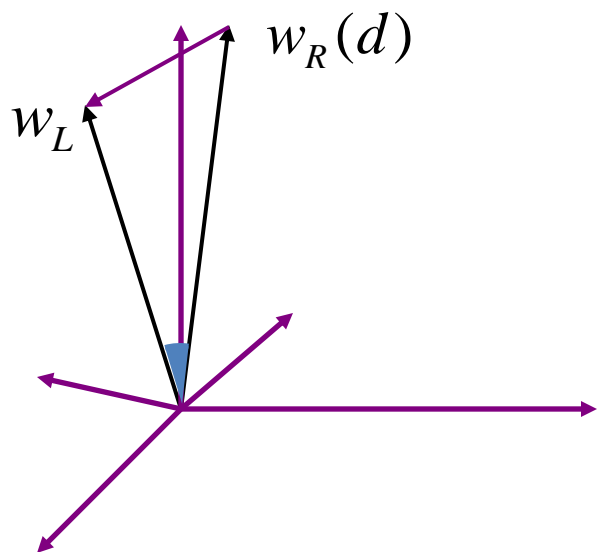
$$\|I\|_{W_m(x,y)} = \sqrt{\sum_{(u,v) \in W_m(x,y)} [I(u,v)]^2}$$

Window magnitude

$$\hat{I}(x,y) = \frac{I(x,y) - \bar{I}}{\|I - \bar{I}\|_{W_m(x,y)}}$$

Normalized pixel

# Độ đo sử dụng góc giữa hai vector



$$\begin{aligned} C_{\text{NC}}(d) &= \sum_{(u,v) \in W_m(x,y)} \hat{I}_L(u,v) \hat{I}_R(u-d,v) \\ &= w_L \cdot w_R(d) = \cos \theta \end{aligned}$$

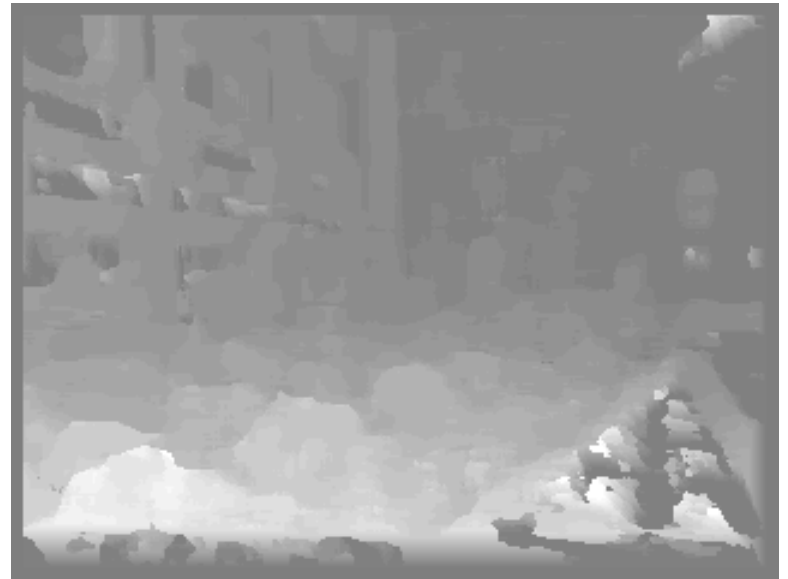
$$d^* = \arg \max_d w_L \cdot w_R(d)$$

# Một vài kết quả

Left



Disparity Map



Images courtesy of Point Grey Research

# Một vài kết quả



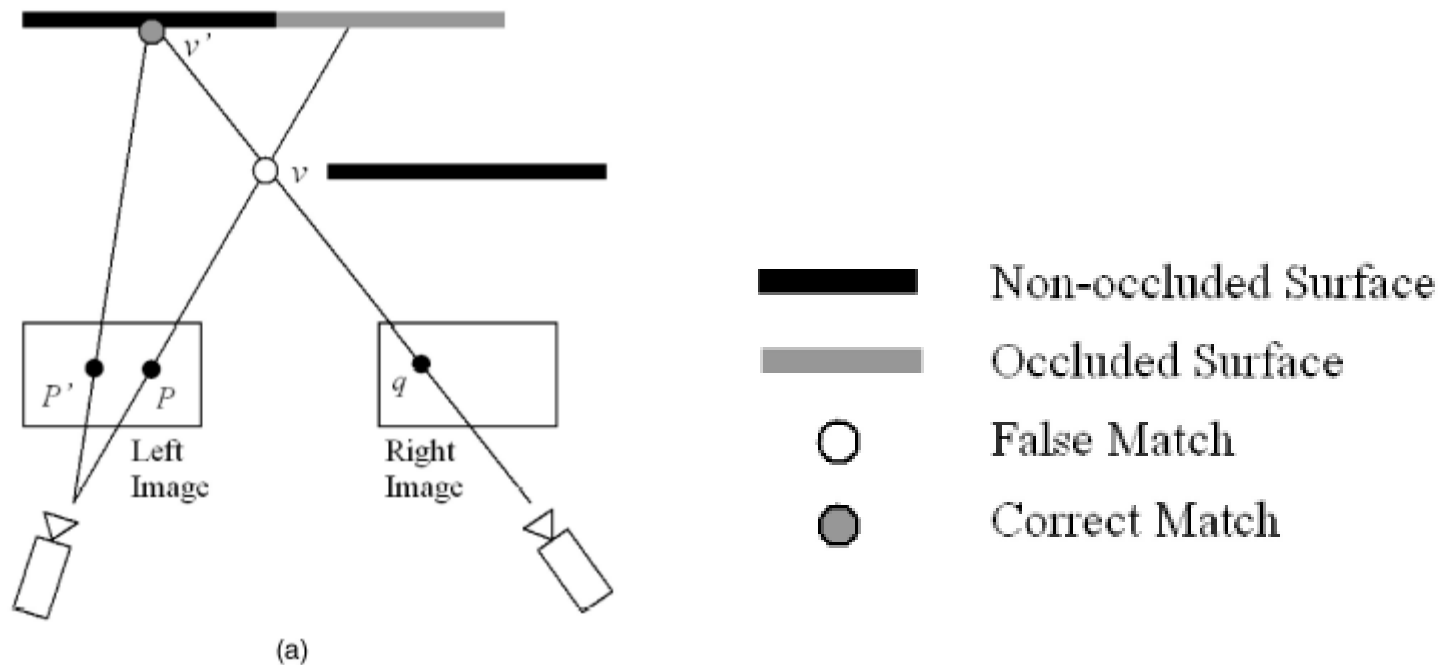
Left



Disparity map

# Dò tìm các vị trí gốc khuất

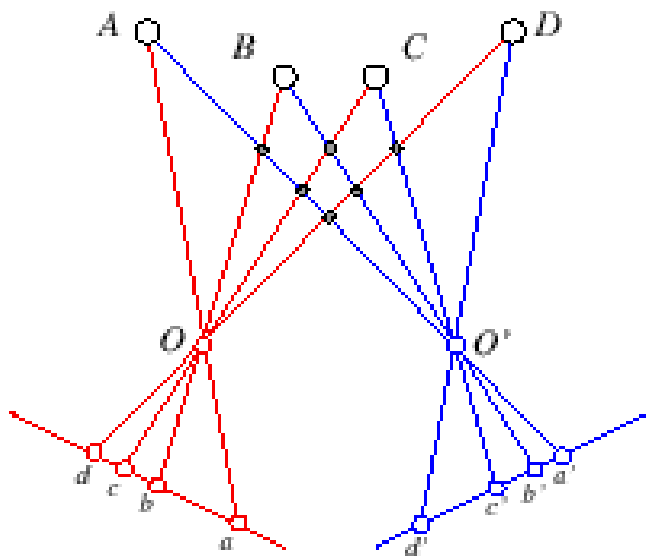
Vị trí gốc khuất thường xảy ra khi các vật sắp xếp theo chiều sâu khác nhau trước góc nhìn của hai camera.



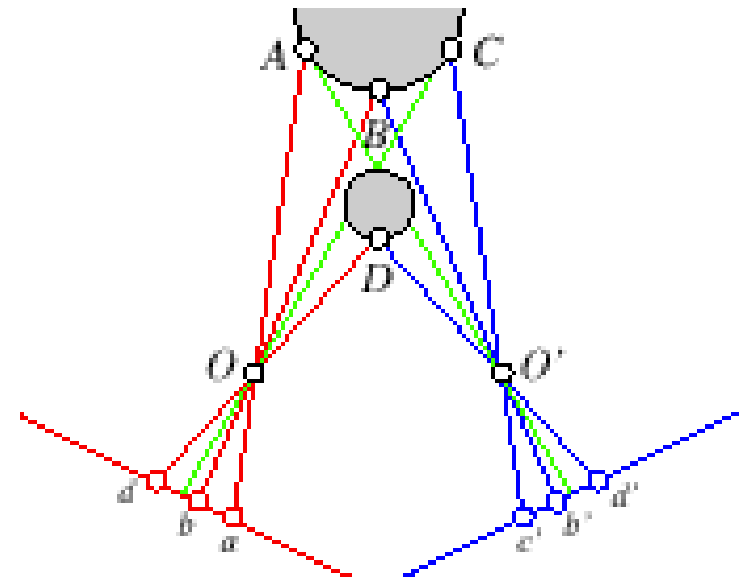


# Điều kiện ràng buộc

- Thứ tự của các điểm thuộc về các đối tượng tương ứng trong ảnh bên trái và ảnh bên phải thường giống nhau, tuy nhiên nếu xảy ra che khuất thì điều kiện trên không còn được thỏa và làm cho kết quả disparity không còn chính xác

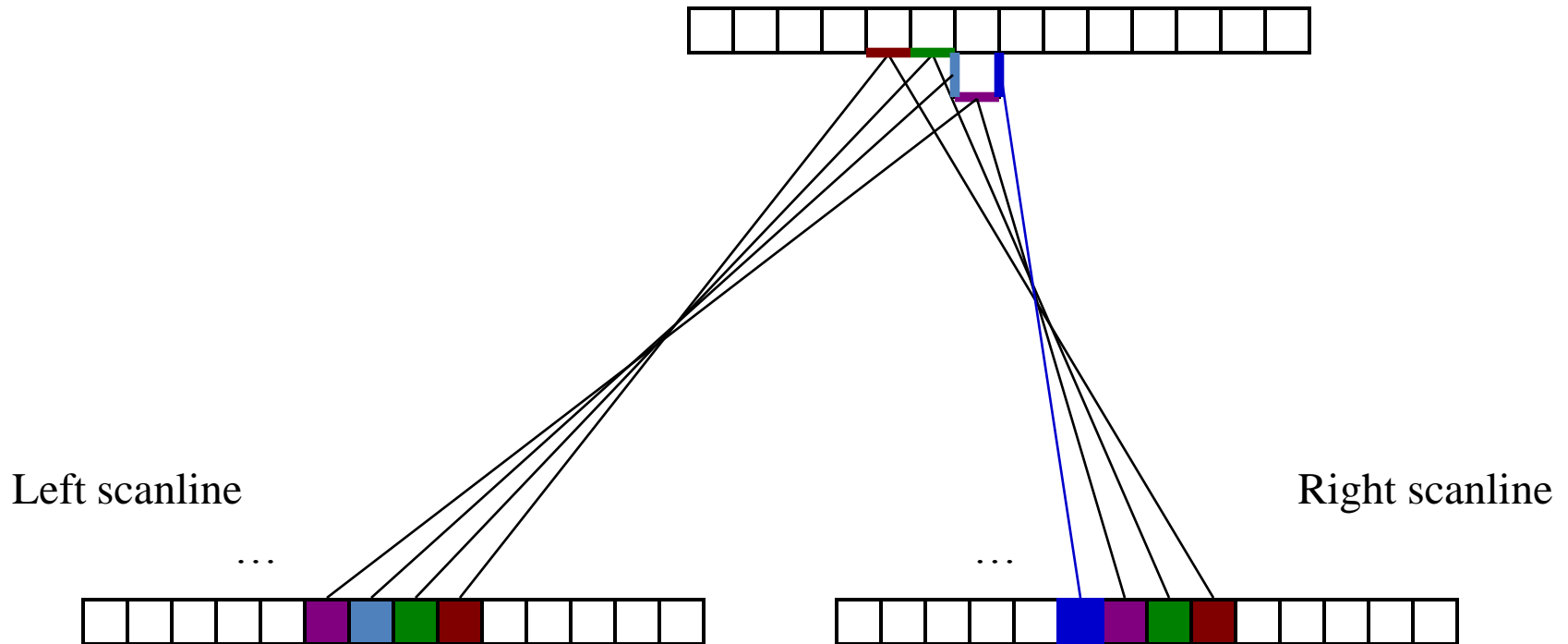


Ordering constraint...



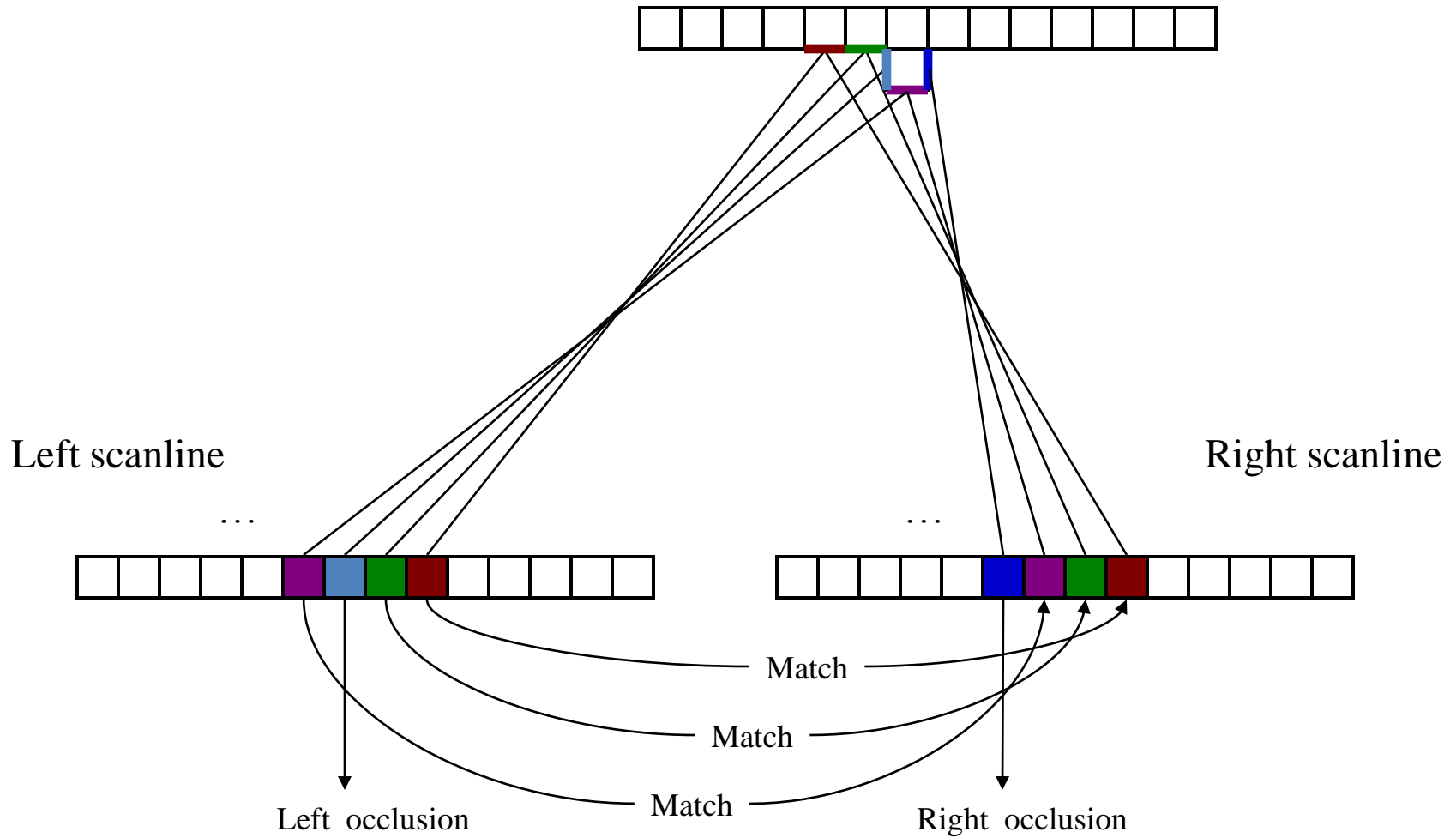
...and its failure

# Ví dụ

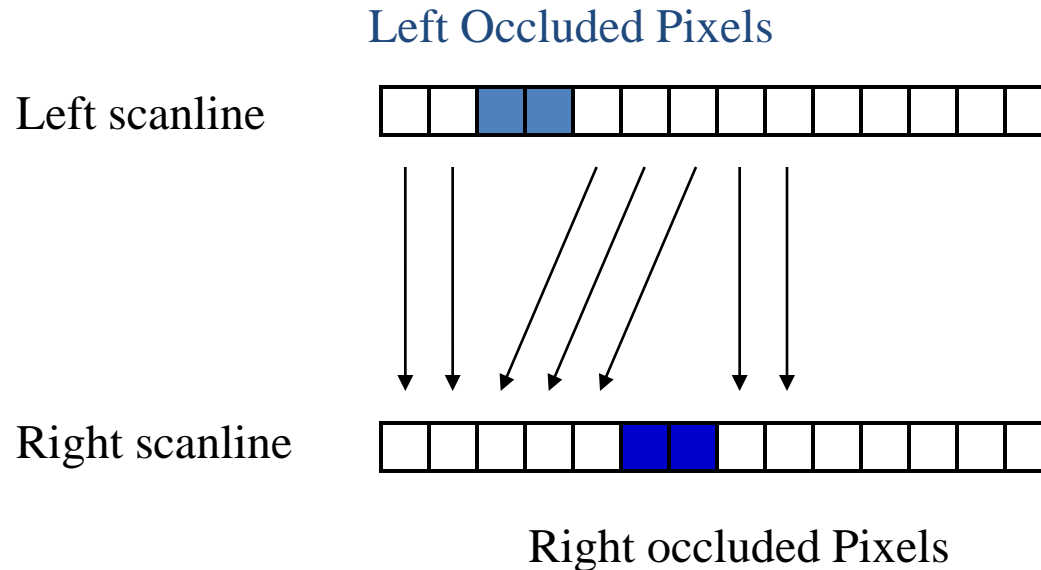


Match intensities sequentially between two scanlines

# Ví dụ



# Quan hệ giữa độ lệch (disparity) và che khuất (Occlusion)

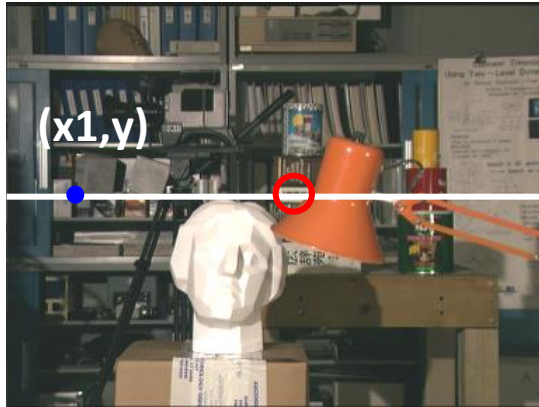


Có 3 trường hợp:

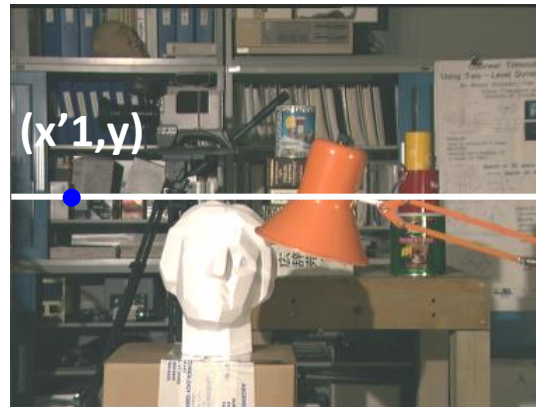
- Có tương ứng trong ảnh trái và phải (không bị che khuất)
- Bị che khuất trái
- Bị che khuất phải

# Độ lệch vs che khuất

left image



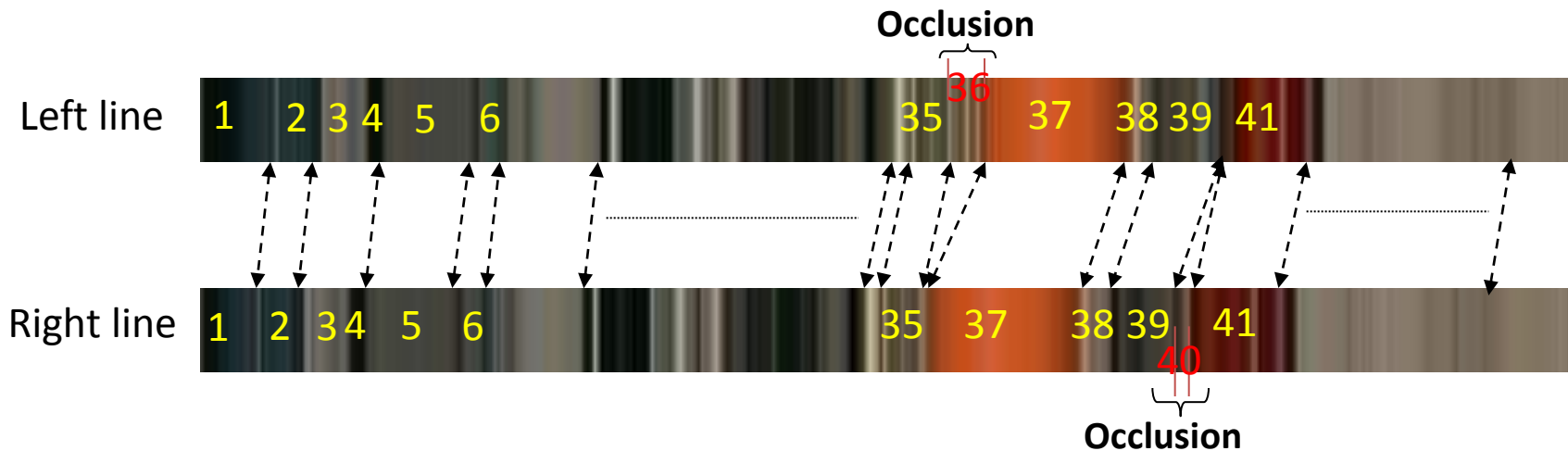
right image



$f$ : focal length  
 $d$ : disparity  
 $b$ : distance of 2 cameras  
 $z$ : depth coordinate

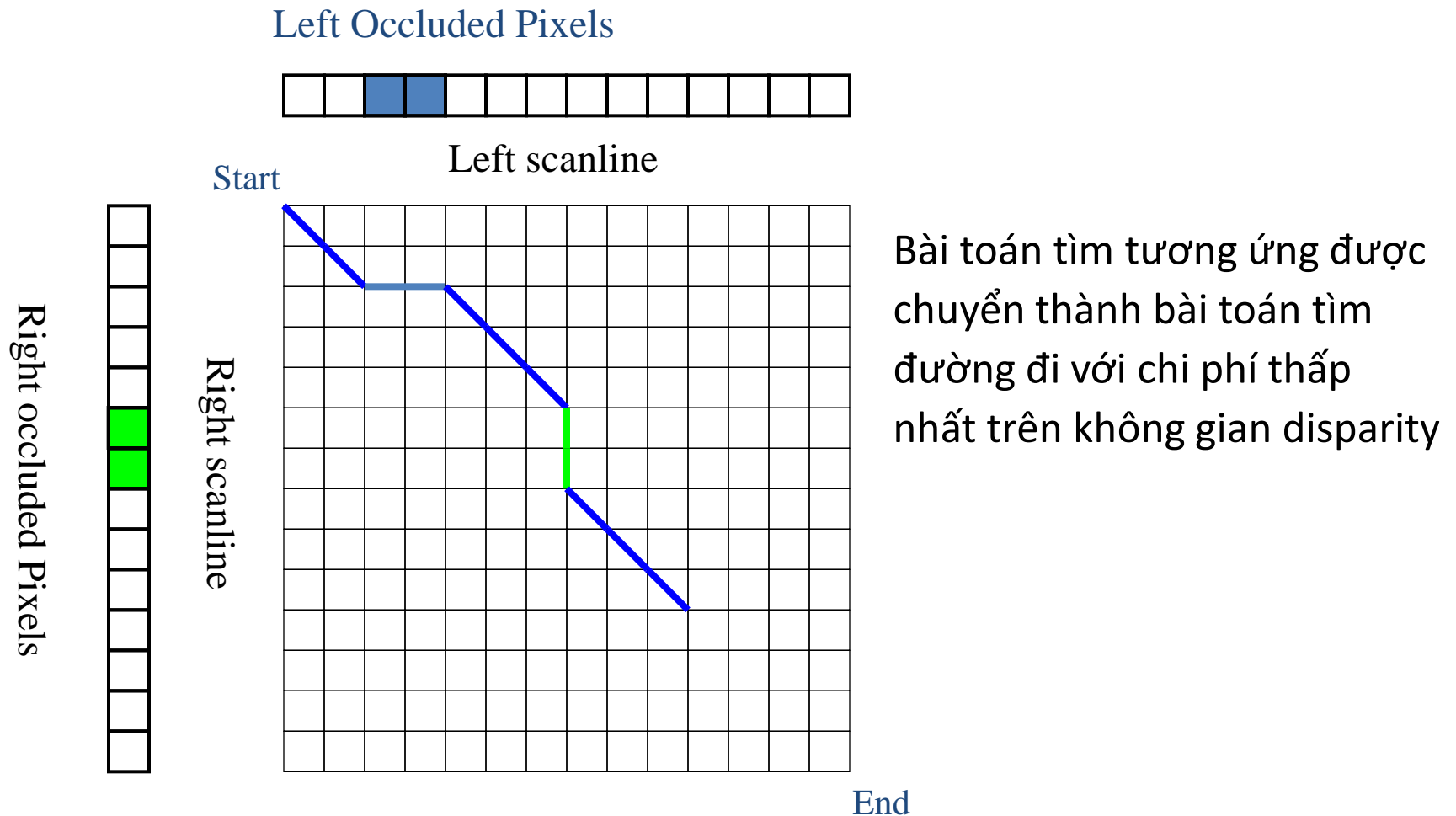
$$\frac{d}{f} \approx \frac{b}{z}$$

**disparity** =  $|x_1 - x'_1|$ : tỉ lệ nghịch với chiều sâu



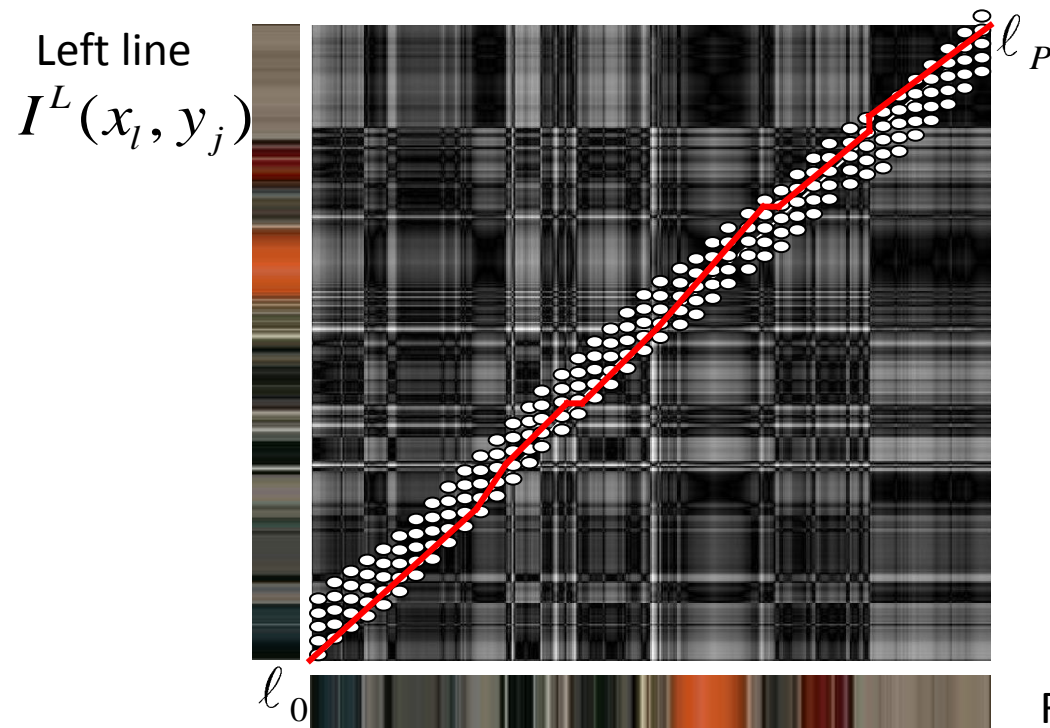
Khó khăn của bài toán stereo là xấp xỉ các giá trị disparity cho các vị trí bị che khuất

# Giải bài toán tìm tương ứng Stereo bằng quy hoạch động



# Không gian độ lệch disparity

- Dynamic programming matching (Global)



Dynamic programming (DP): tìm đường đi trên không gian độ lệch

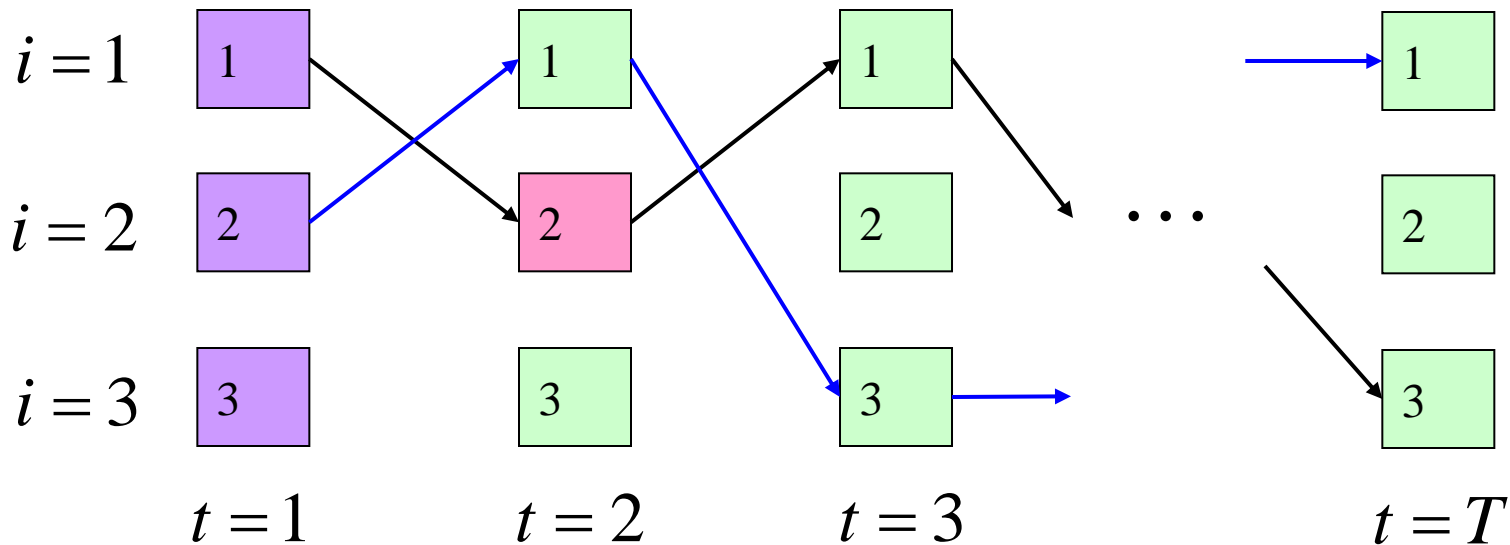
Matching space: Ma trận được tạo ra bằng cách lấy dòng trái trừ cho dòng phải với độ lệch thay đổi từ 0 đến n

$$M(l, r) = |I^L(x_l, y_j) - I^R(x_r, y_j)|$$

Right line  $I^R(x_r, y_j)$

# Quy hoạch động

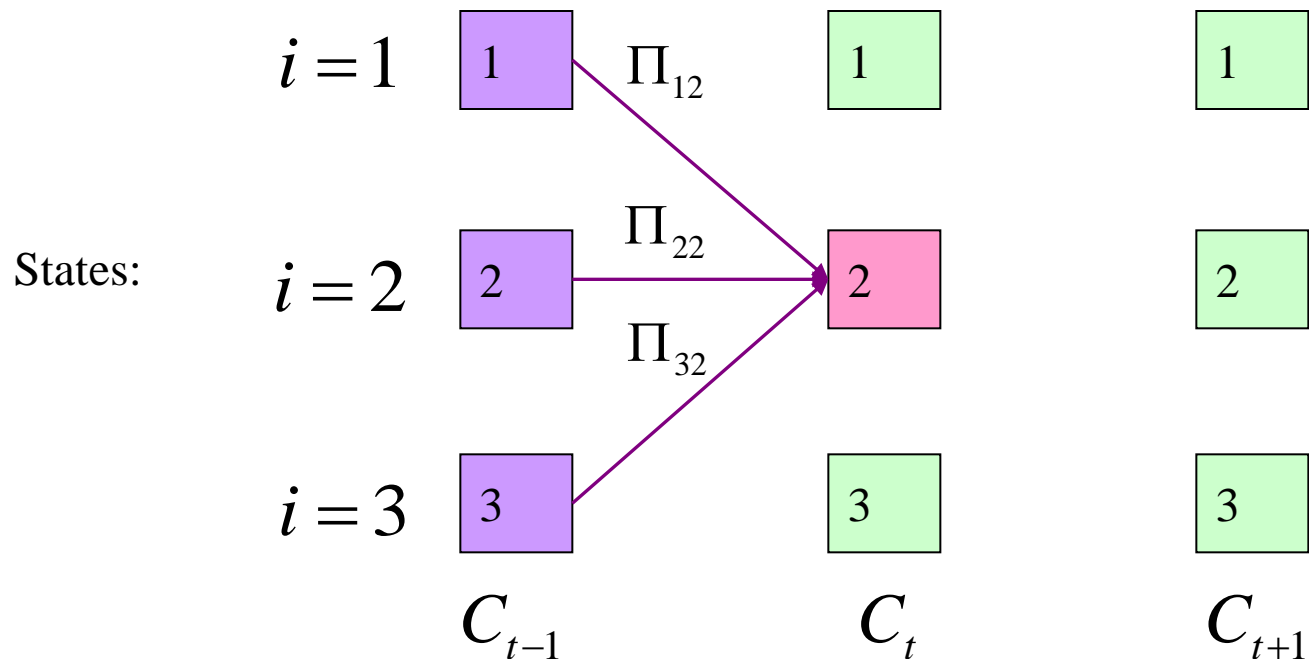
- Xây dựng trellis (lưới) với mỗi node chính là khả năng lời giải của độ lệch disparity.



Có bao nhiêu đường trong một trellis ?  $3^T$



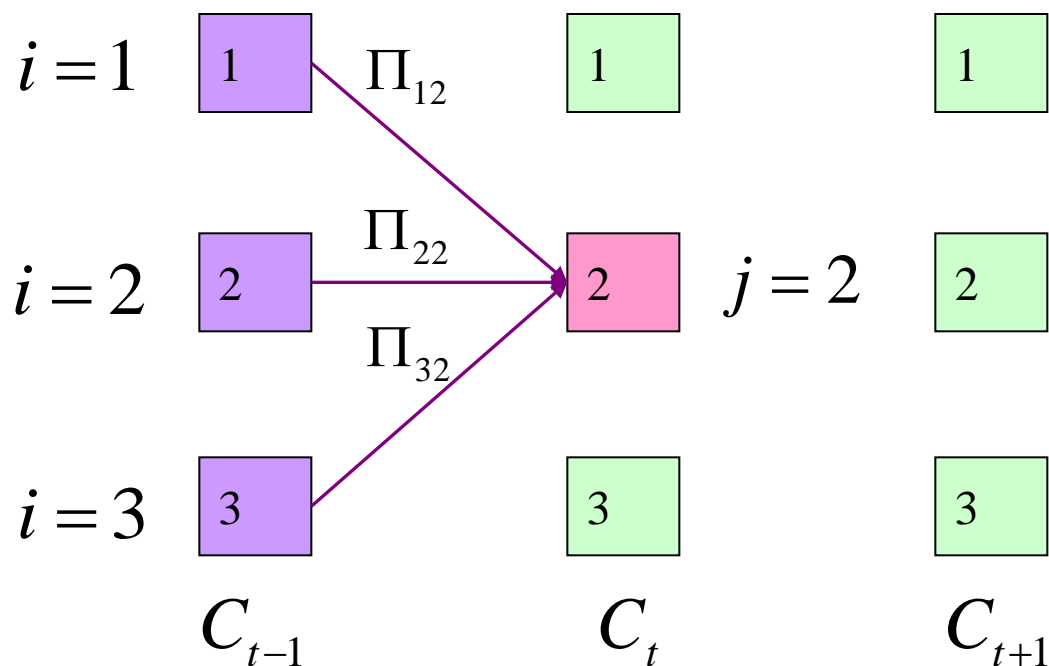
# Quy hoạch động



Chi phí của đường đi có thể chia làm nhiều giai đoạn:

$$\Pi_{ij} = \text{Cost of going from state } i \text{ to state } j$$

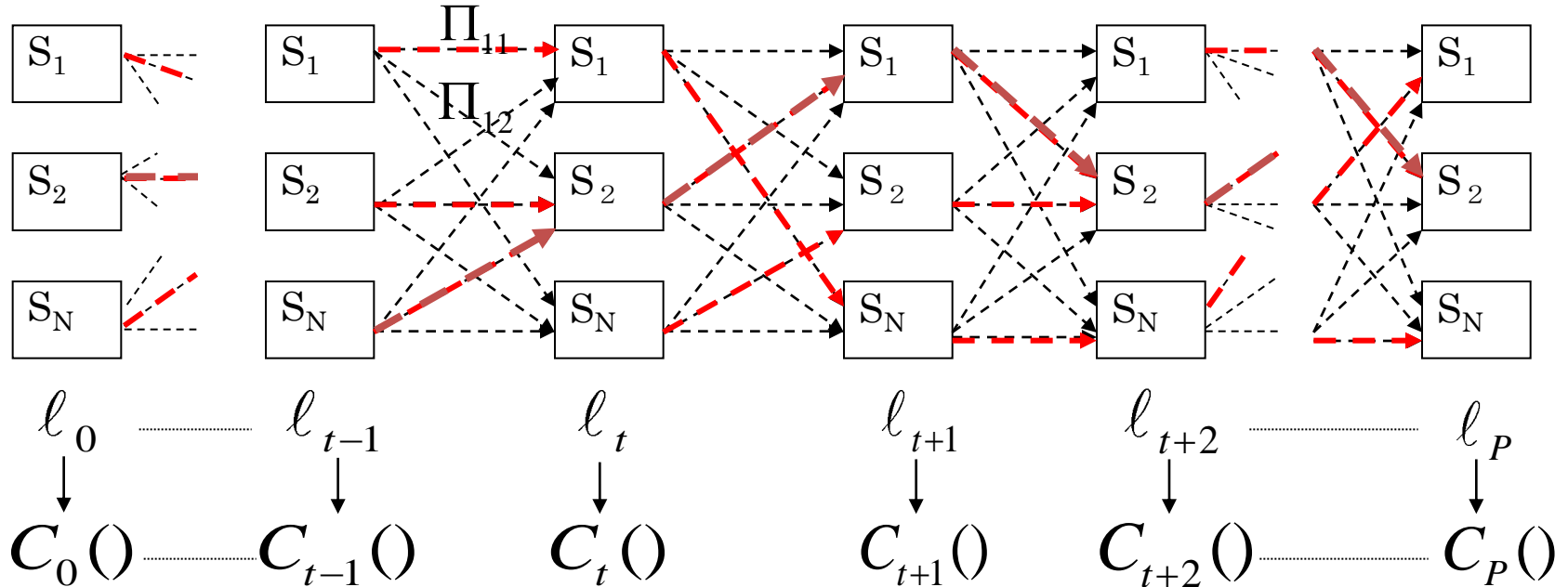
# Quy hoạch động



*Chi phí đường đi đến node thứ  $t^{\text{th}}$   $C_t$*

$$C_t(j) = \min_i (\Pi_{ij} + C_{t-1}(i))$$

# Quy hoạch động



Tính toán tại bước  $t$ -stage: (recursive operation)

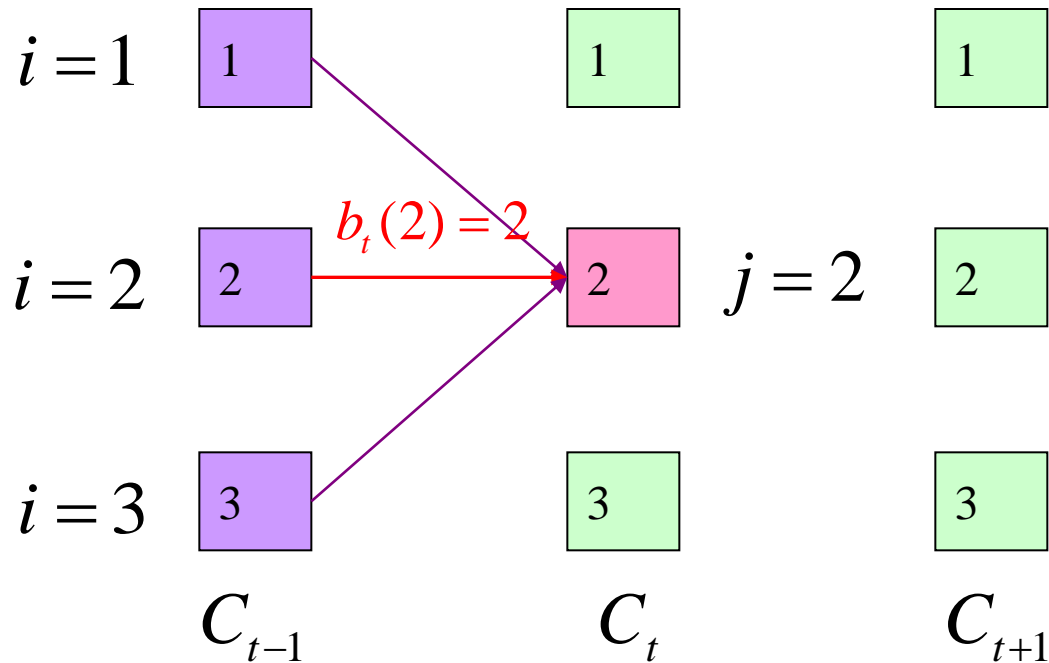
$$C_t(S_i) = \min_j \{ \Pi_{j1} + C_{t-1}(S_j) \}_{j=1, \dots, N; i=1, \dots, N}.$$

$\Pi_{r1} = M(r, 1)$ ;  $S_i$ : candidate of disparity value;

$N$ : the maximum value of disparity.

Nếu không gian trạng thái là xác định thì giá trị min toàn cục sẽ tồn tại.

# Dynamic Programming



*Đường đi của tất cả các node đều được lưu trữ đến node cuối cùng*

$$C_t(j) = \min_i (\Pi_{ij} + C_{t-1}(i))$$

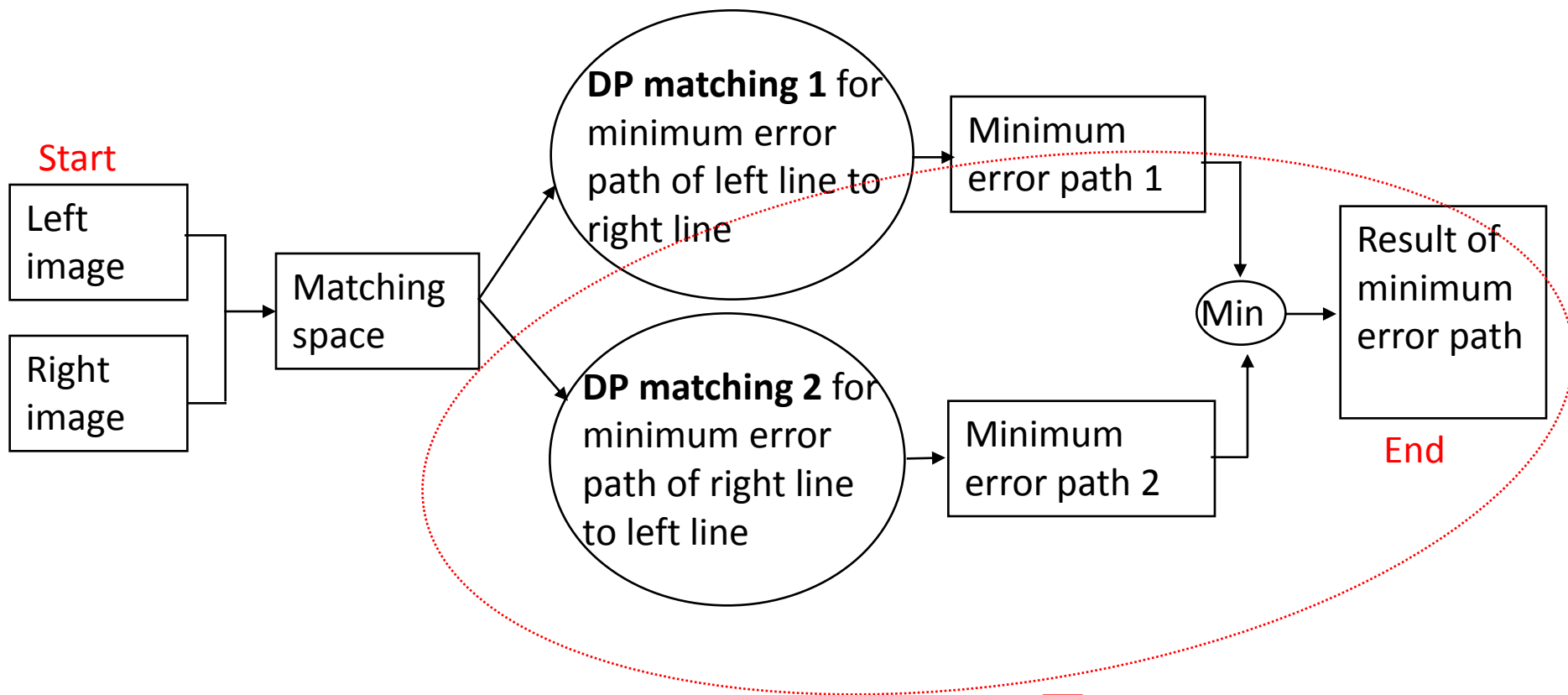
$$b_t(j) = \arg \min_i (\Pi_{ij} + C_{t-1}(i))$$

# Results



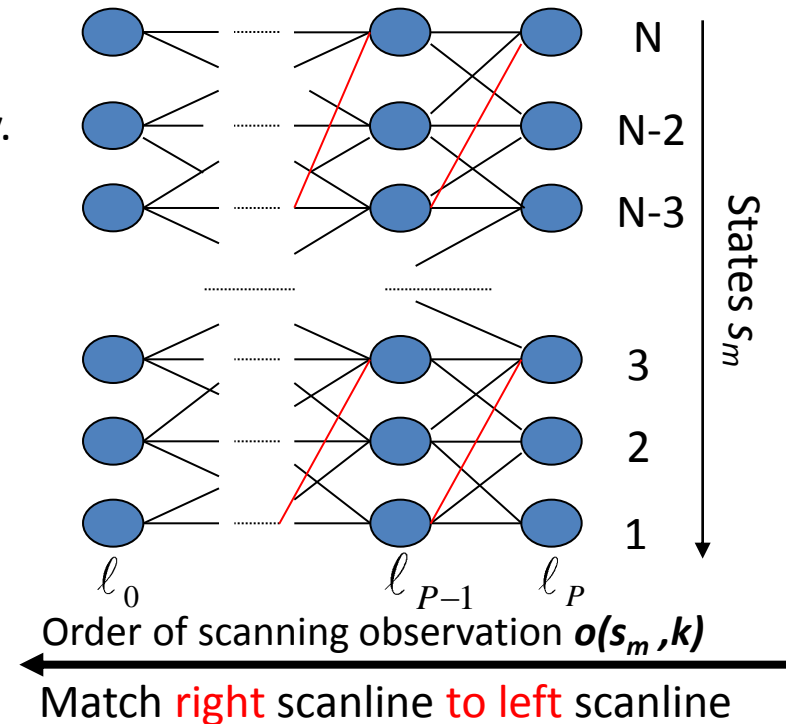
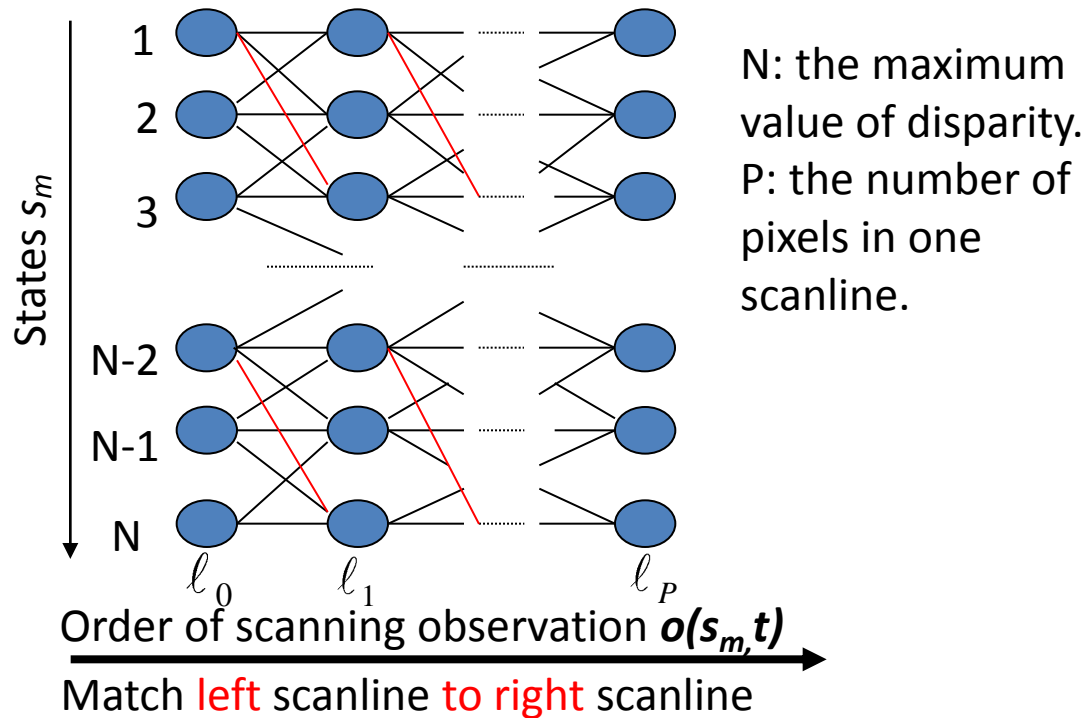
Lỗi có thể bị lan truyền qua các node dẫn đến kết quả không chính xác

# Áp dụng quy hoạch động theo hai chiều



**Tìm đường đi tốt nhất giữa hai nhánh dựa trên hai kết quả thu được.**

# Quy hoạch động theo hai chiều



# Standard benchmark example

Stereo result in Tsukuba picture :



Left Image



Right Image



Ideal disparity ( $d^*$ )

Trắng : ở gần camera  
Đen : ở xa camera

Accuracy:

$$\frac{\sum \text{correct disparities}}{\sum \text{pixels}}$$

Correct disparity:

$$|d^*(i, j) - d(i, j)| \leq 1$$



**SSD Fixed Window Method**  
**89%(±1) , 363 s**



**Proposed Method**  
**95.7 %(±1) , 256 s**



# Giải bài toán Stereo sử dụng phương pháp phân đoạn ảnh

- Tính chất không liên tục trên không gian disparity thường có quan hệ cao với màu của ảnh
- Sự thay đổi của disparity bên trong một segment thường nhỏ
- Tiếp cận dựa trên phương pháp đối sánh cho các segment có các vùng kích thước và vị trí tương đồng

# Ví dụ

