Concepts of Stereo Vision

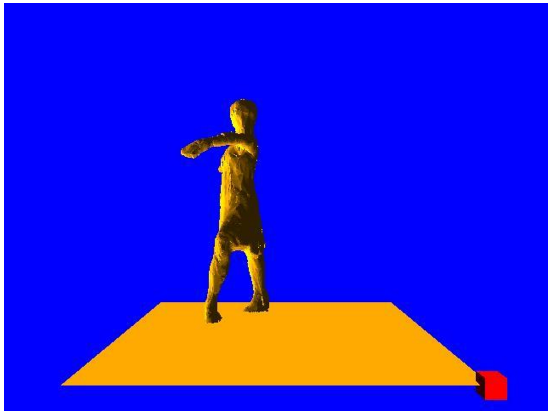
Stereo Vision

Virtual Reality

[Takeo Kanade *et al.*, CMU]

- More than 50 video sequences
- Reconstruct 3D objects form videos

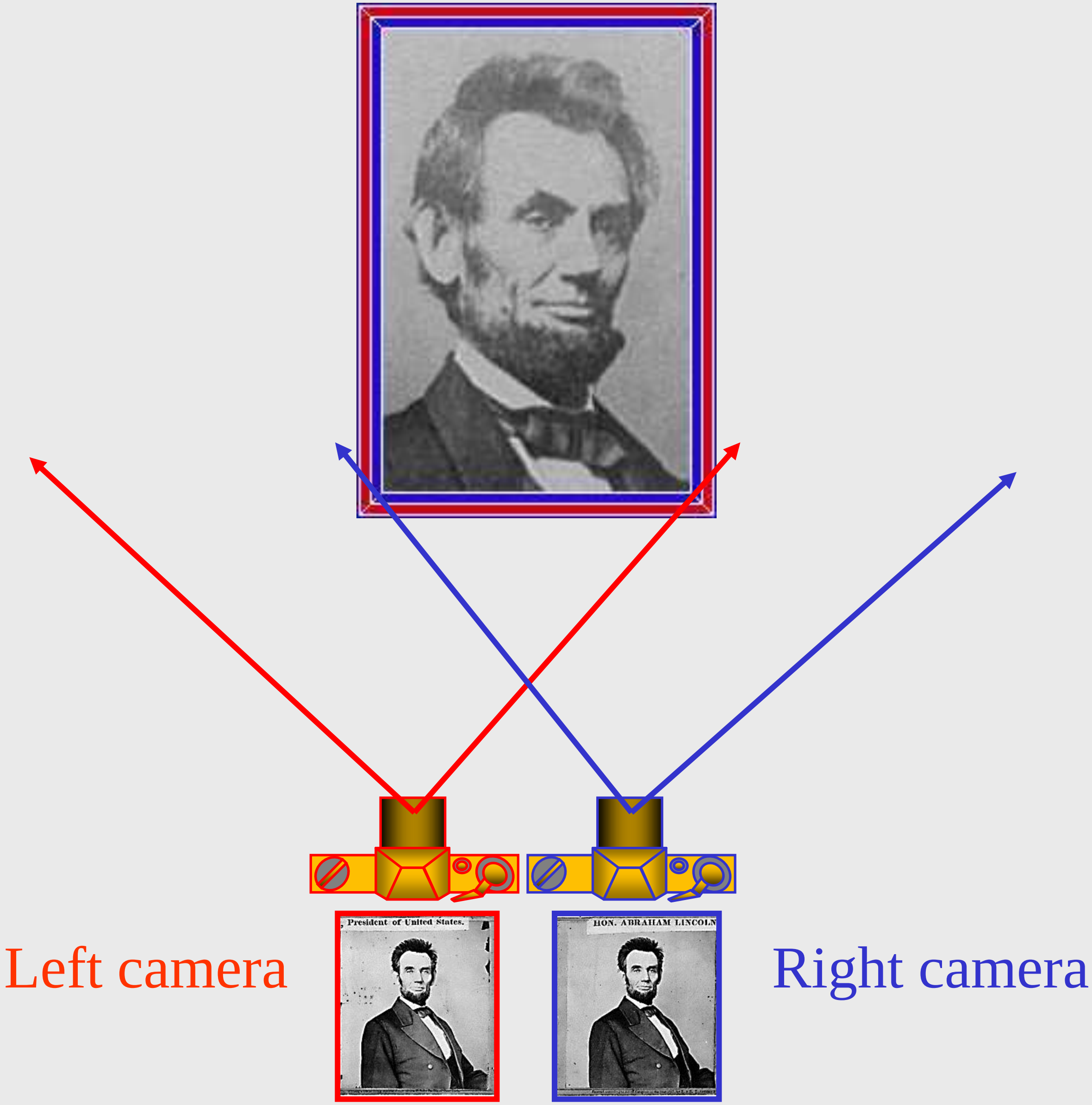




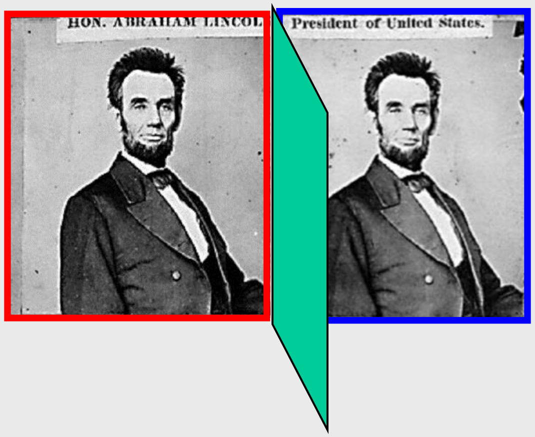
- steerable version used for SuperBowl XXV "[eye vision]"

http://www.cs.cmu.edu/afs/cs/project/VirtualizedR/www/VirtualizedR.html

Example of stereo vision

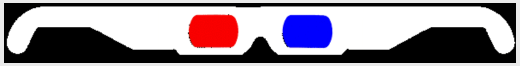


How to display different images to two eyes ?

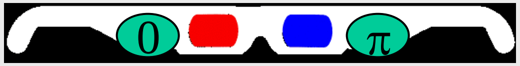


Methods:

1) Color Filter Glasses (濾光眼鏡)

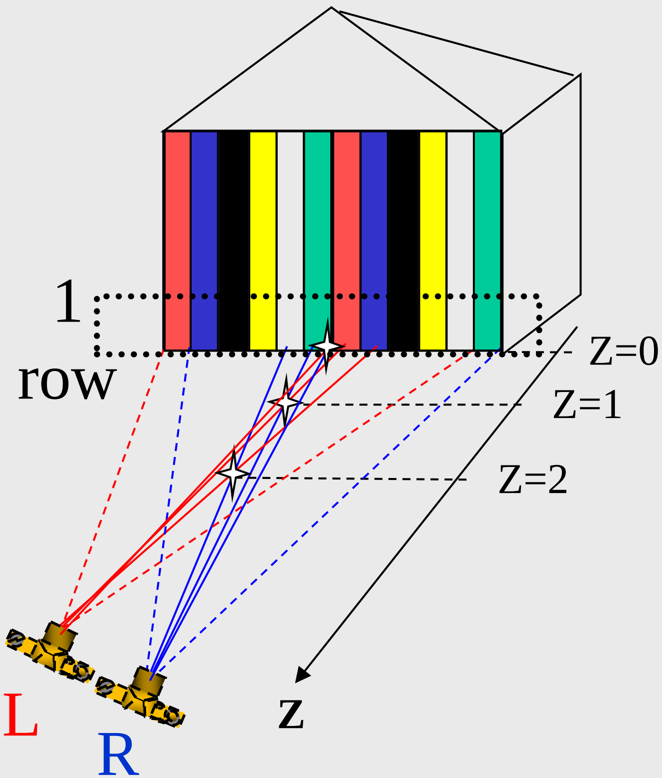


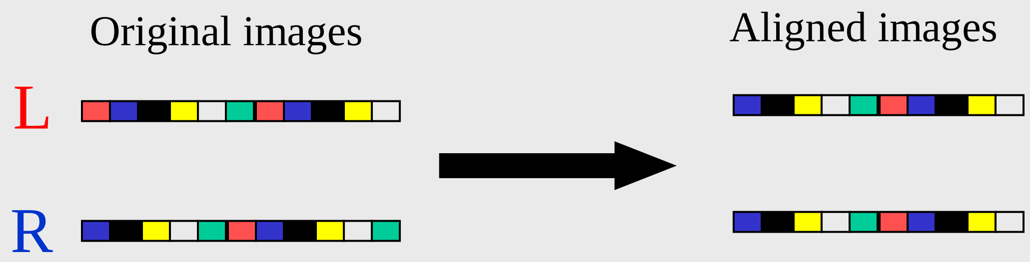
2) Polarized glasses (偏光眼鏡)



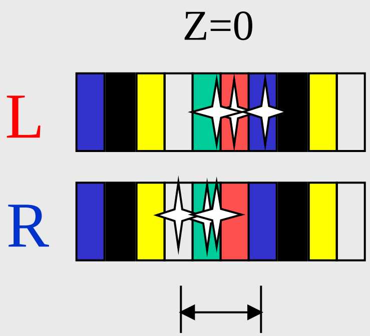
3) Synchronization between the screen and glasses

Parallax and disparity(視差與位差)





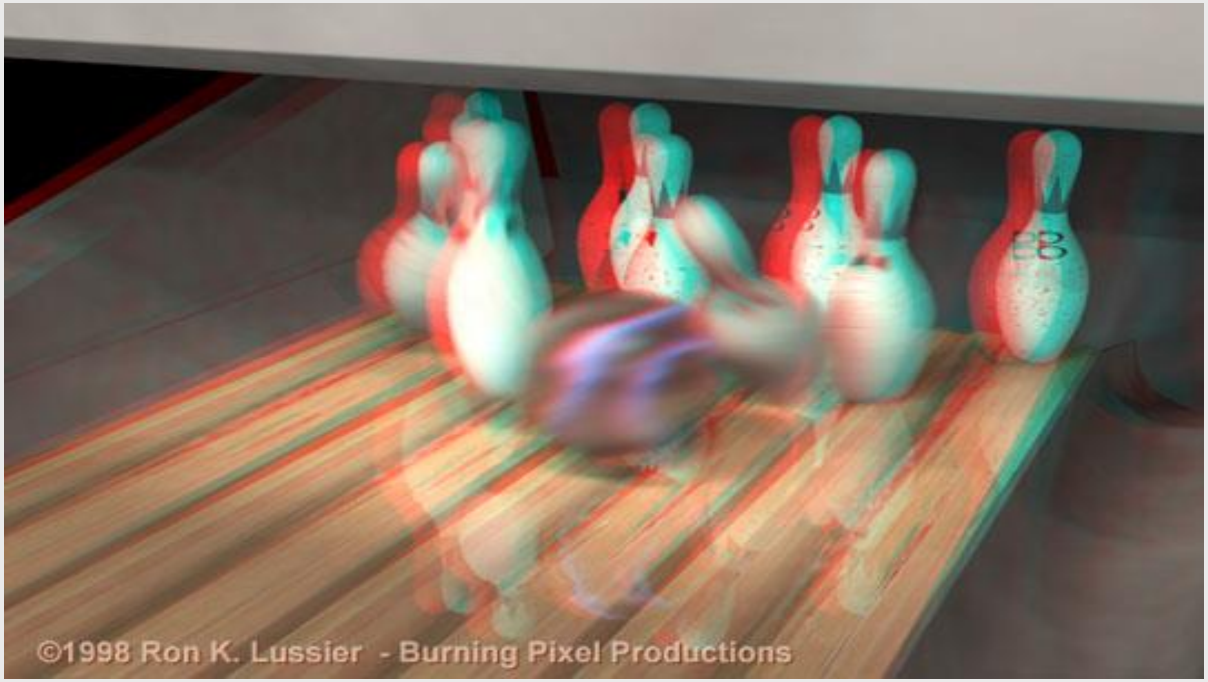
Alignment ➔ Measure depth with respect to this plane

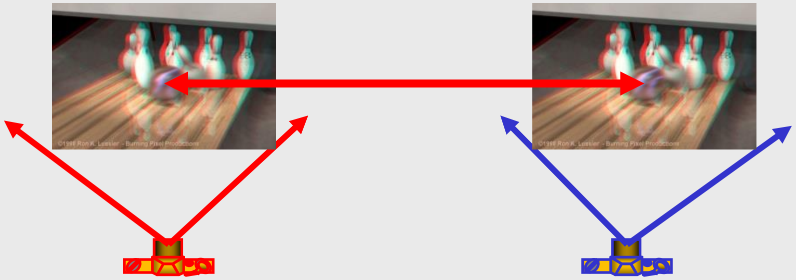


Parallax (3D)

Disparity (1D)

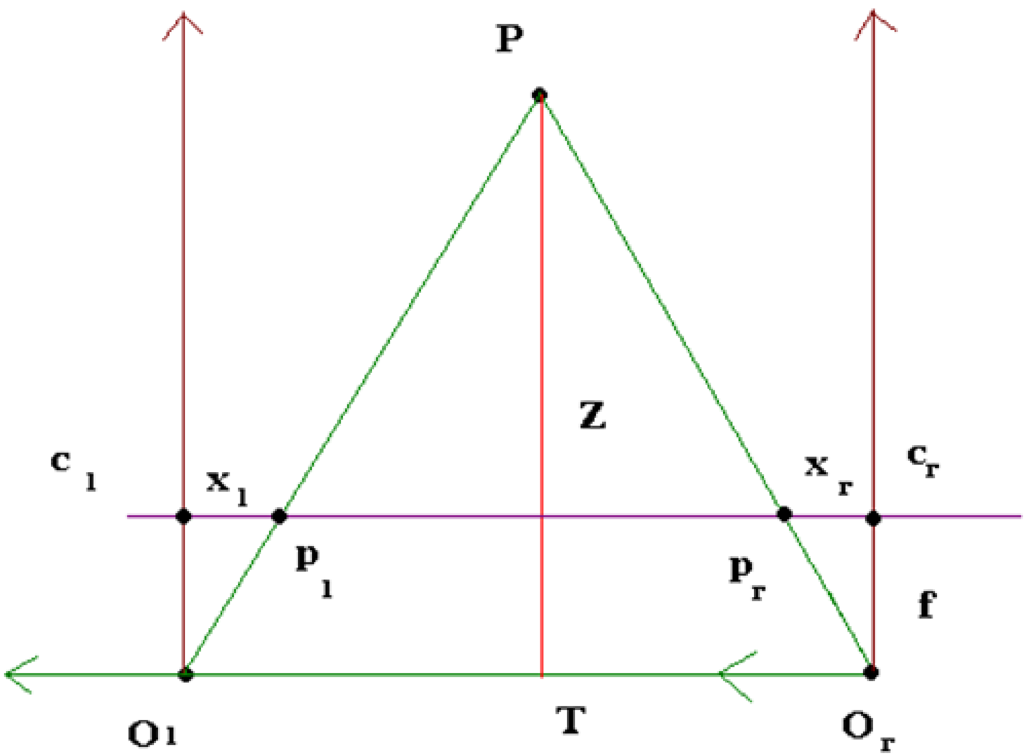
位差顯示
(Displaying disparity)





$$disparity \propto \frac{1}{depth}$$

Recovering depth (reconstruction)



P :its projections p_r, and p_l

$$x_l = f\frac{X_l}{Z_l}, or \quad X_l = \frac{x_l Z_l}{f}$$

$$x_r = f\frac{X_r}{Z_r}, or \quad X_r = \frac{x_r Z_r}{f}$$

Cameras are related by the transformation

$$P_r = P_l - T(translation)$$

Using $Z_r = Z_l = Z$ and $X_r = X_l - T$, we have

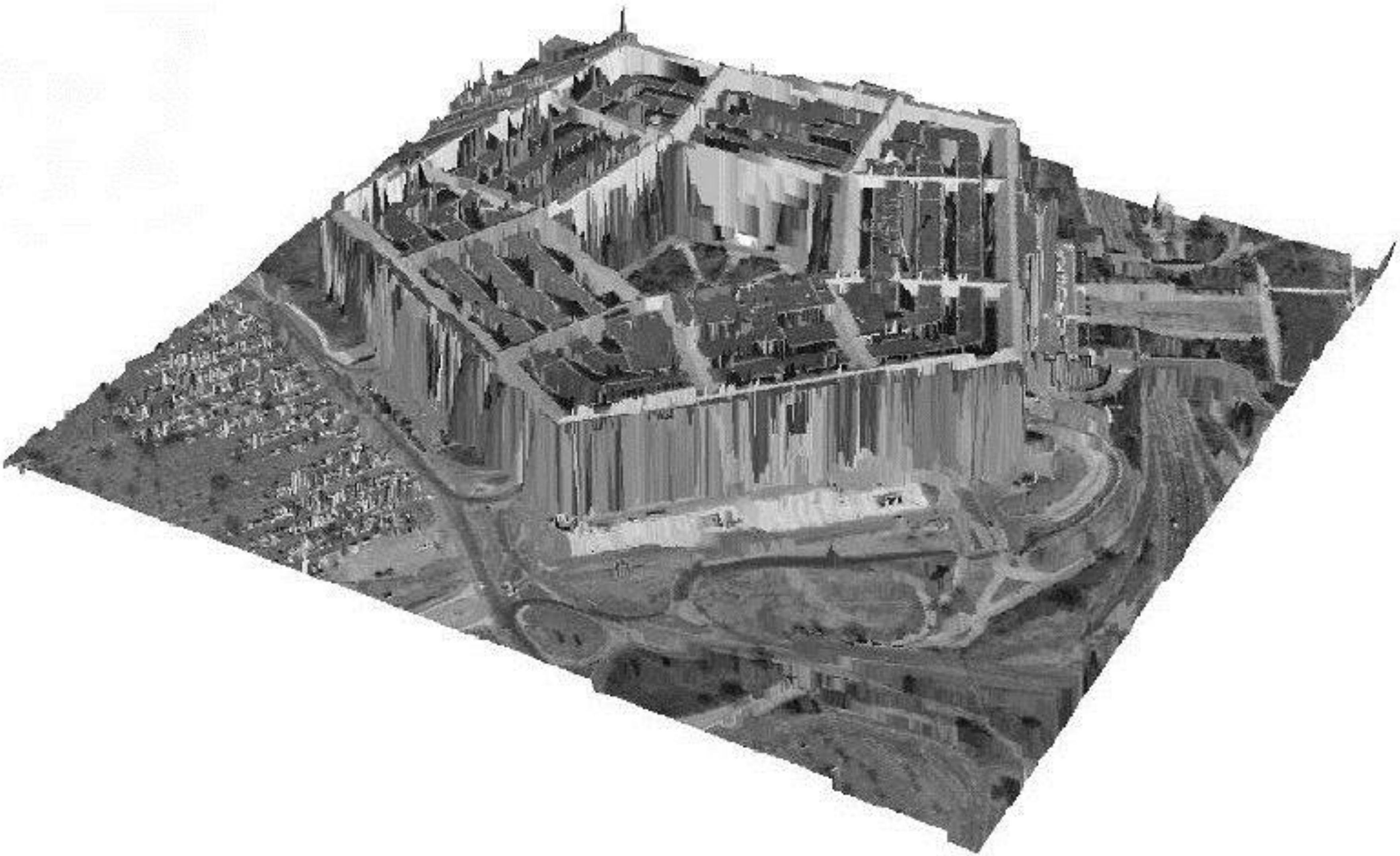
$$\frac{x_l Z}{f} - T = \frac{x_r Z}{f} \quad or \quad Z = \frac{Tf}{d}$$

where $d = x_l - x_r$ is the disparity.

$$disparity \propto \frac{1}{depth}$$

Stereo Vision

$$disparity \propto \frac{1}{depth}$$

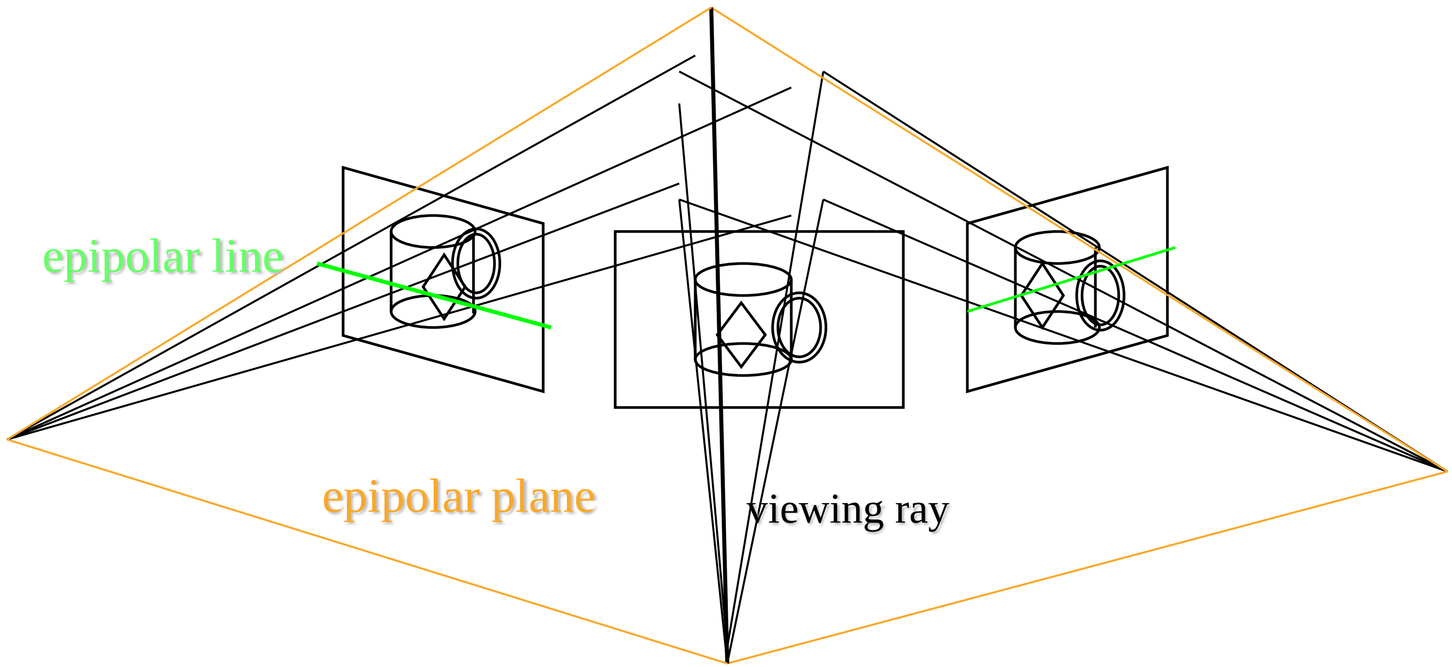


Feature Matching

- What kinds of methods can be used to match
  - Match “point features”, interpolate
  - Match “edge features”, then interpolate
  - Match “intensities”, then interpolate
  - Optimization Method:
    - iterative updating
    - dynamic programming
    - energy minimization (regularization, stochastic)
    - graph algorithms

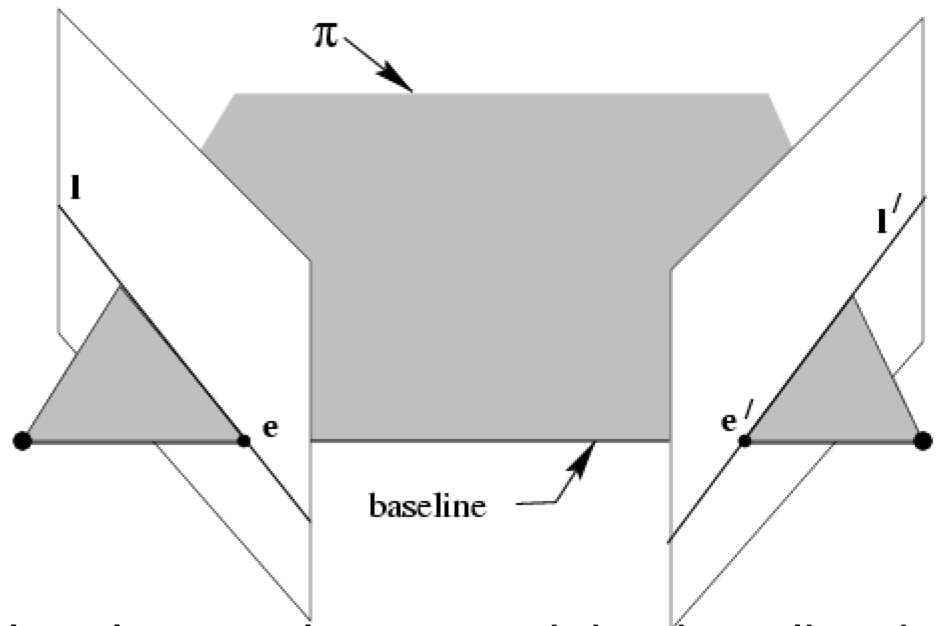
極線幾何 (epipolar geometry)

Try to find the best matching pairs along the epipolar line



極線幾何

epipoles e,e'
= intersection of baseline with image plane
= projection of projection center in other image
= vanishing point of camera motion direction



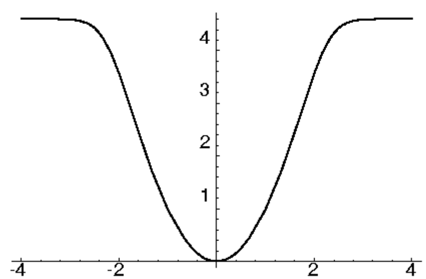
an epipolar plane = plane containing baseline (1-D family)

an epipolar line = intersection of epipolar plane with image
(always come in corresponding pairs)

Matching technique in Stereo Vision

For every disparity, compute *raw* matching .1
costs

$$E_0(x, y; d) = \rho(I_L(x' + d, y') - I_R(x', y'))$$



Why use a robus

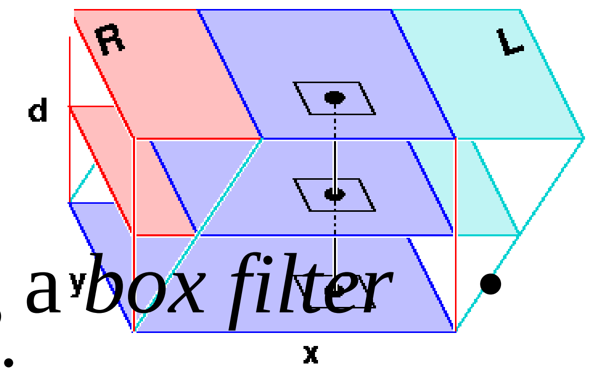
occlusions, oth

Can also use alternative match criteria •

Matching technique in Stereo Vision

2. Aggregate costs spatially

$$E(x,y;d) = \sum_{(x',y')\in N(x,y)} E_0(x',y',d)$$



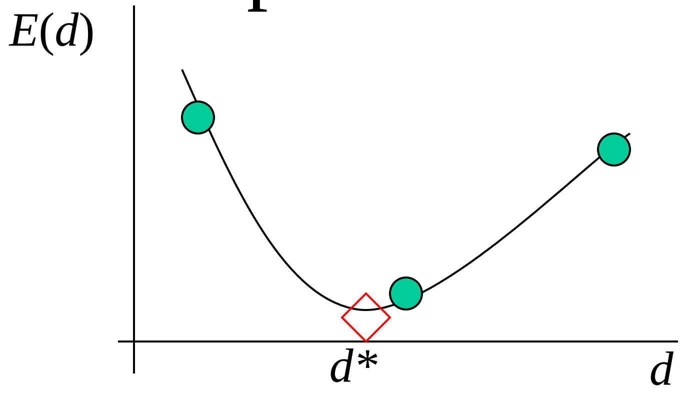
- Here, we are using a *box filter* (efficient moving average implementation)
- Can also use weighted average, [non-linear] diffusion…

Matching technique in Stereo Vision

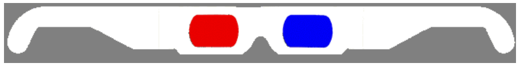
Choose winning disparity at each pixel .3

$$d(x,y) = \arg\min_{d} E(x,y;d)$$

Interpolate to *sub-pixel* accuracy .4



Example



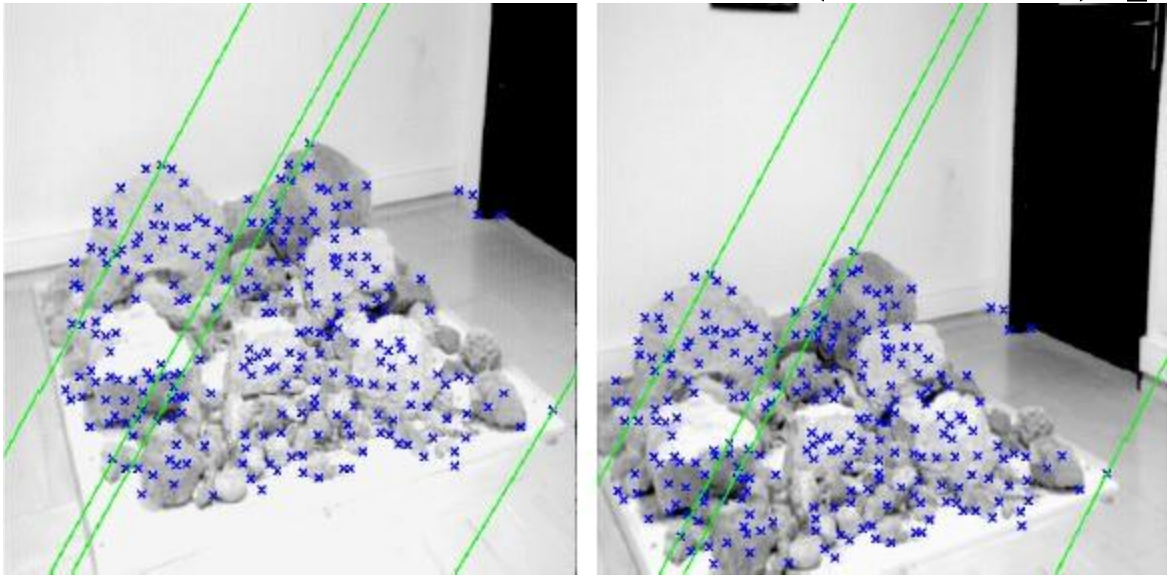
http://www.rainbowsymphony.com/freestuff.html

(Wikipedia for images)

Public Library, Stereoscopic Looking Room, Chicago, by Phillips, 1923

Matching technique in Stereo Vision

- Match "corner" (interest) points



- Interpolate complete solution

Energy minimization

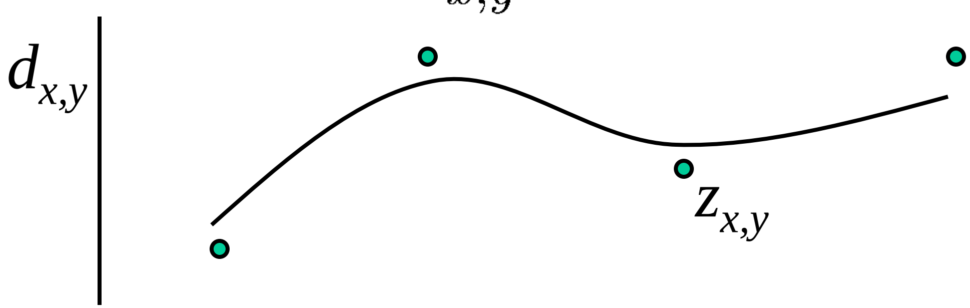
- 1-D example: approximating splines

$$E_{\text{total}}(\mathbf{d}) = E_{\text{data}}(\mathbf{d}) + \lambda E_{\text{smoothness}}(\mathbf{d})$$

$$E_{\text{data}}(\mathbf{d}) = \sum_{x,y} (d_{x,y} - z_{x,y})^2$$

$$E_{\text{membrane}}(\mathbf{d}) = \sum_{x,y} (d_{x,y} - d_{x-1,y})^2$$

$$E_{\text{thin plate}}(\mathbf{d}) = \sum_{x,y} (2d_{x,y} - d_{x-1,y} - d_{x+1,y})^2$$



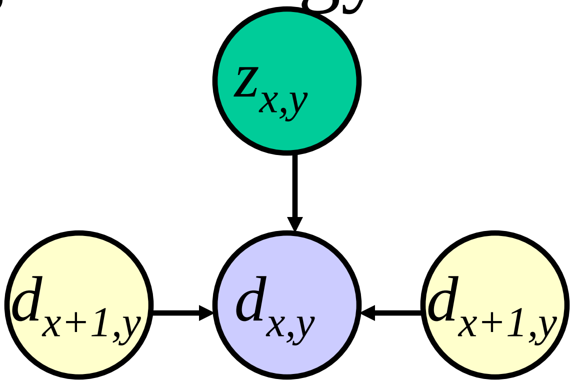
Relaxation

- How can we get the best solution?
- Differentiate energy function, set to 0

$$
\begin{aligned}
\frac{\partial E}{\partial d_{x,y}} &= 2(d_{x,y} - z_{x,y}) + \\
&\quad 2\lambda(2d_{x,y} - d_{x-1,y} - d_{x+1,y}) = 0 \\
d_{x,y} &\leftarrow \frac{1}{1+2\lambda}(z_{x,y} + d_{x-1,y} + d_{x+1,y})
\end{aligned}
$$

Relaxation

- Iteratively improve a solution by locally minimizing the energy: *relax* to solution

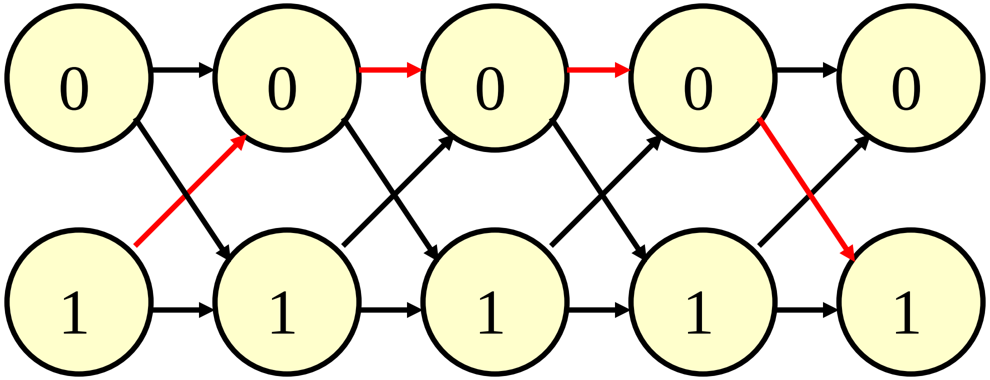


- Earliest application: WWII numerical simulations

Dynamic programming

- Evaluate best cumulative cost at each pixel

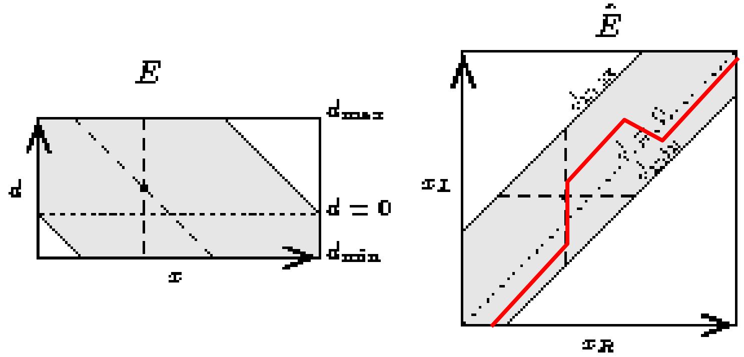
$$
\begin{aligned}
E_{\text{total}}(\mathbf{d}) &= E_{\text{data}}(\mathbf{d}) + \lambda E_{\text{smoothness}}(\mathbf{d}) \\
E_{\text{data}}(\mathbf{d}) &= \sum_{x,y} (d_{x,y} - z_{x,y})^2 \\
E_{\text{smoothness}}(\mathbf{d}) &= \sum_{x,y} |d_{x,y} - d_{x-1,y}|
\end{aligned}
$$



Dynamic programming

- 1-D cost function

$$
\begin{aligned}
E(\mathbf{d}) &= \sum_{x,y} \rho_P(d_{x+1,y} - d_{x,y}) + \sum_{x,y} E_0(x, y; d) \\
\tilde{E}(x, y, d) &= E_0(x, y; d) + \\
&\quad \min_{d'} \left(\tilde{E}(x-1, y, d') + \rho_P(d_{x,y} - d'_{x-1,y}) \right)
\end{aligned}
$$



Dynamic programming

- Disparity space image and min. cost path

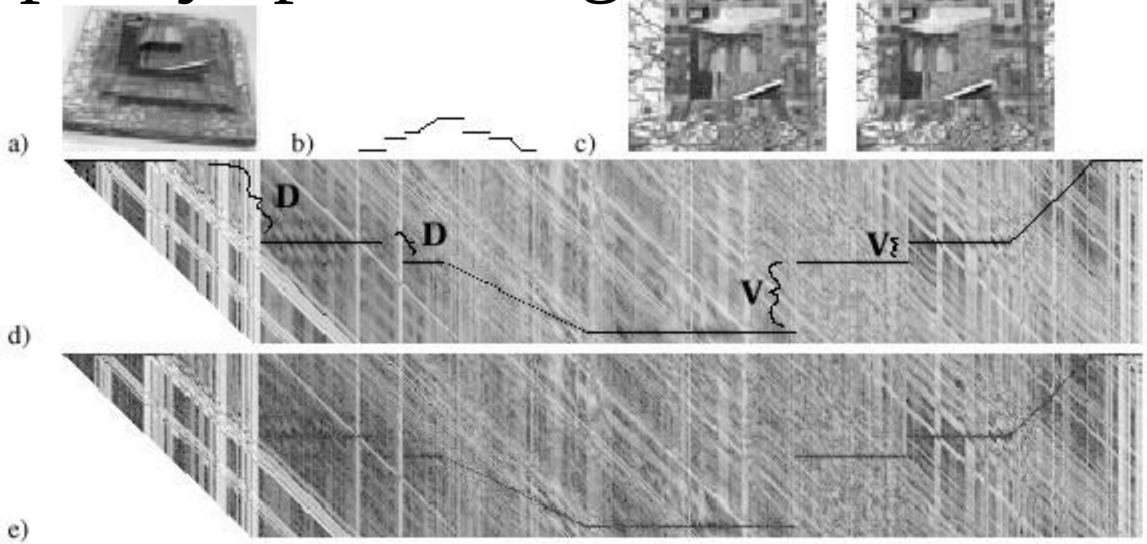
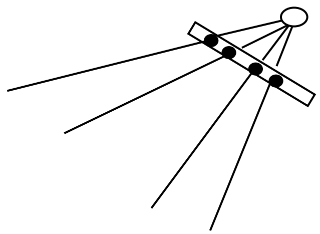


Fig. 4. This figure shows (a) a model of the stereo sloping wedding cake that we will use as a test example, (b) a depth profile through the center of the sloping wedding cake, (c) a simulated, noise-free image pair of the cake, (d) the enhanced, cropped, correlation DSI representation for the image pair in (c), and (e) the enhanced, cropped, correlation DSI for a noisy sloping wedding cake (SNR = 18 dB). In (d), the regions labeled "D" mark diagonal gaps in the matching path caused by regions occluded in the left image. The regions labeled "V" mark vertical jumps in the path caused by regions occluded in the right image.

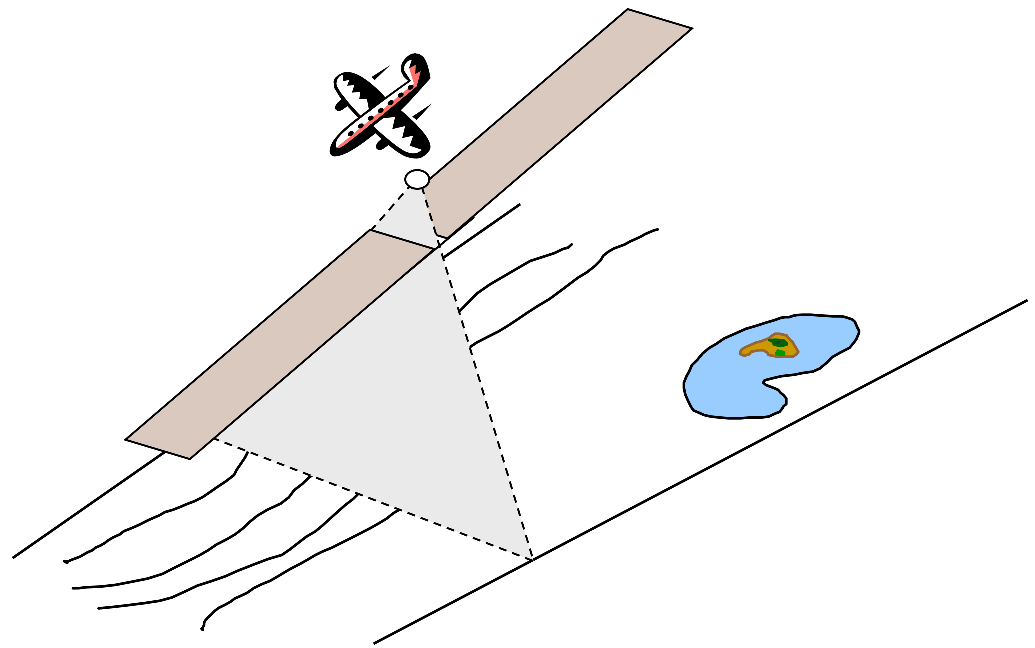
Concepts of Panoramic Images

使用 Pushbroom camera

- 1D projective sensor



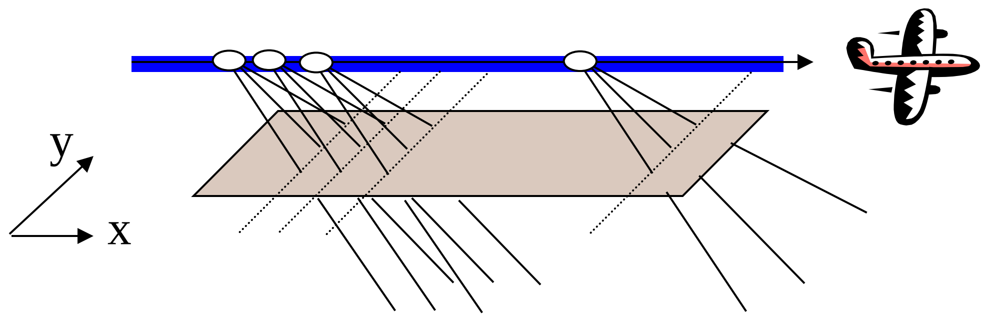
- … translating



Advantage: large field of view in one dimension

使用 Pushbroom camera

The generalized camera model:



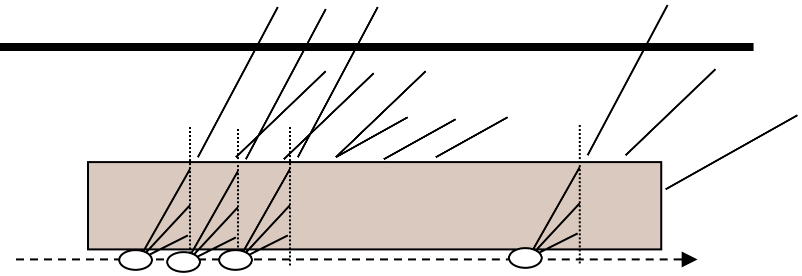
X direction - parallel projection
Y direction - perspective projection

Notion *Generator* – the set of all ray origins

For other cameras, generator can be a 2D surface

Using Pushbroom camera

Imaging process



Images from usual camera

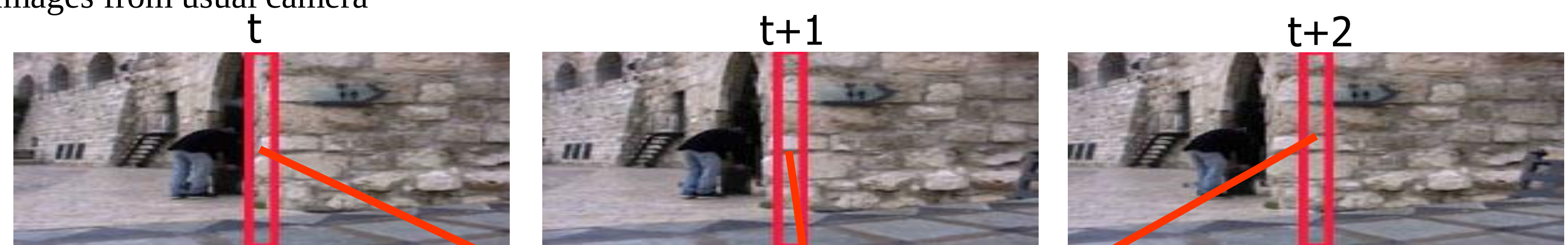
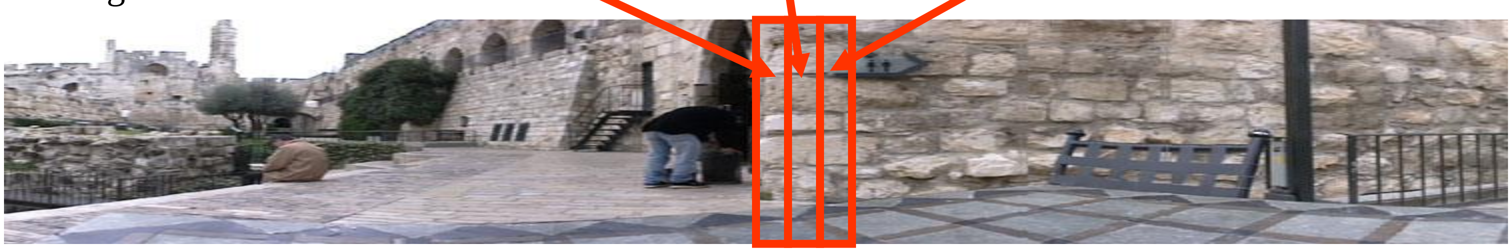


Image from generalized camera

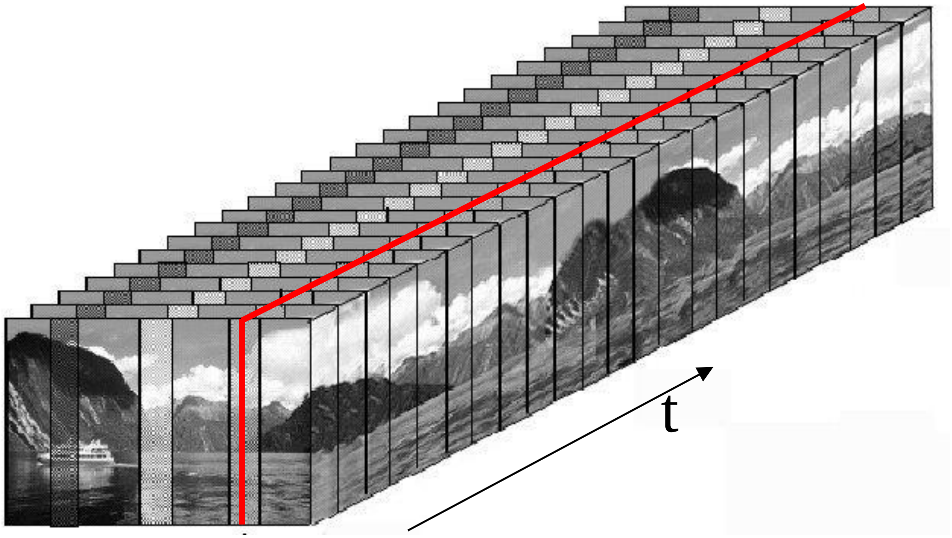


Virtual generalized camera = device (usual camera) + software

Non-classical cameras

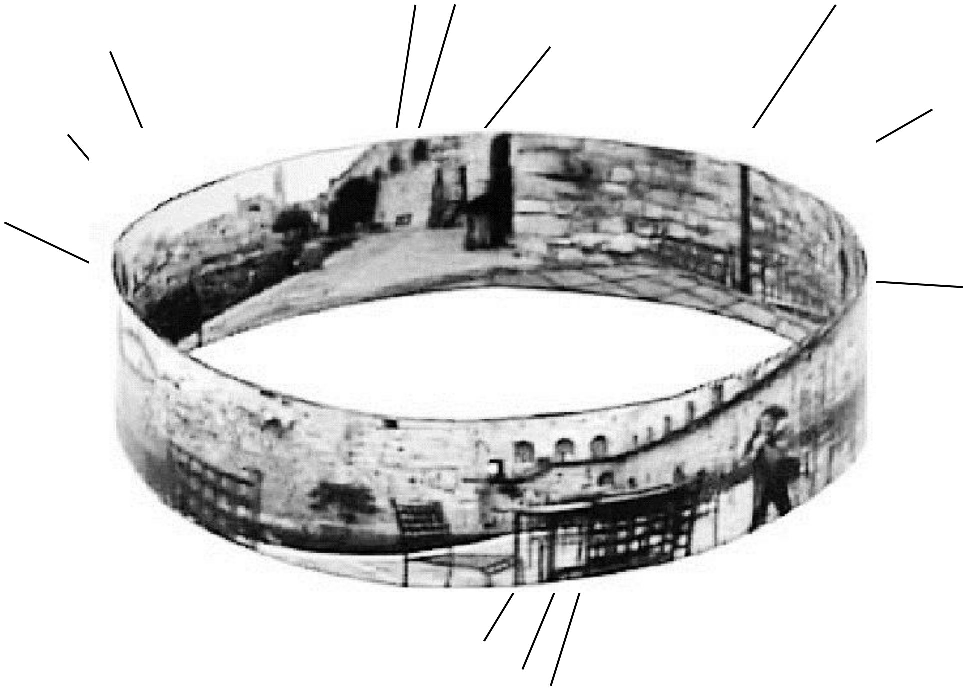
Implementation through cuts of 3D video arrays

- Take images while moving a usual camera
- Stack them into 3D array
- Take a cut along the “time” dimension



Using Circular projective camera

Move “1D sensor” along a circle
record on a cylinder



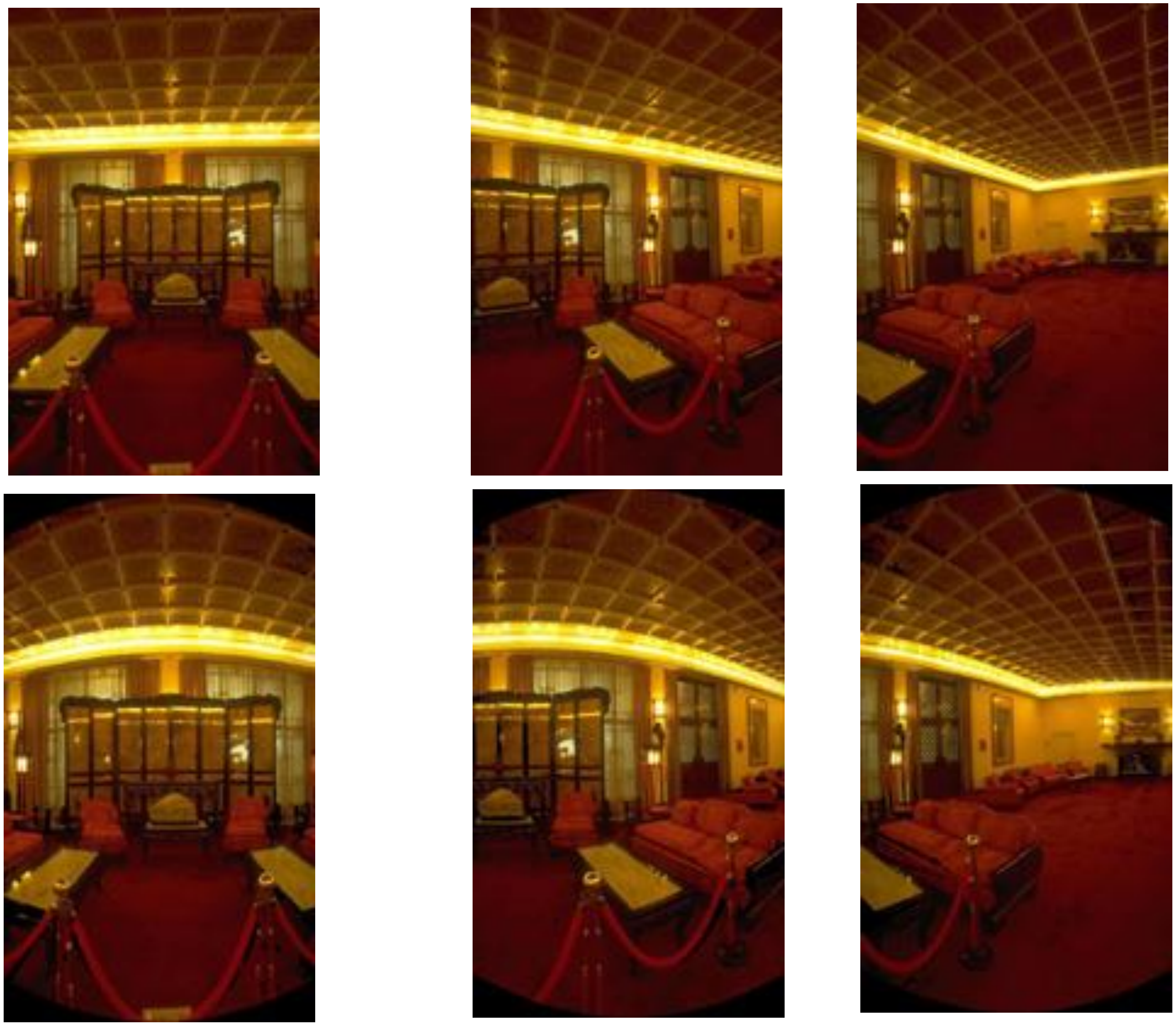
Advantage: complete 360° horizontal view

Using circular projective camera

- Panoramic Image Stitcher
  - Warping
  - Registration
  - Blending
  - Dewarping

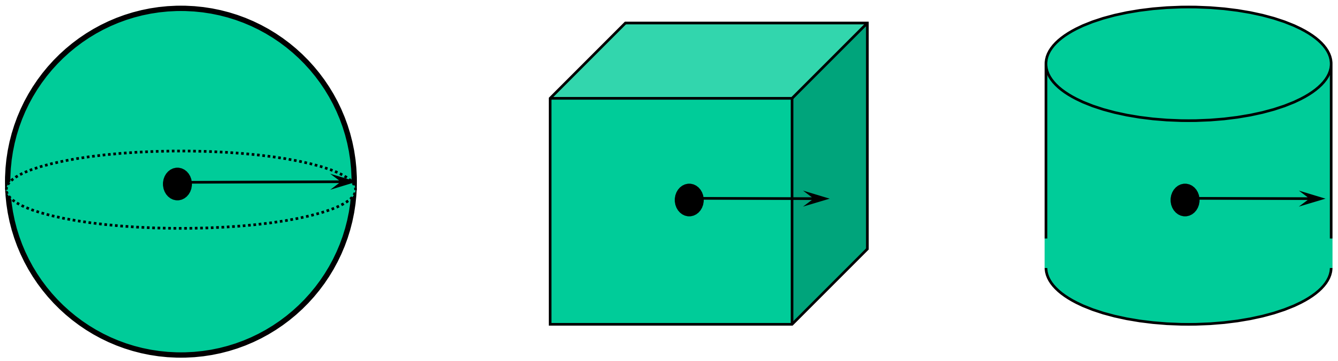
Panoramic Image Stitcher

- Panoramic Images

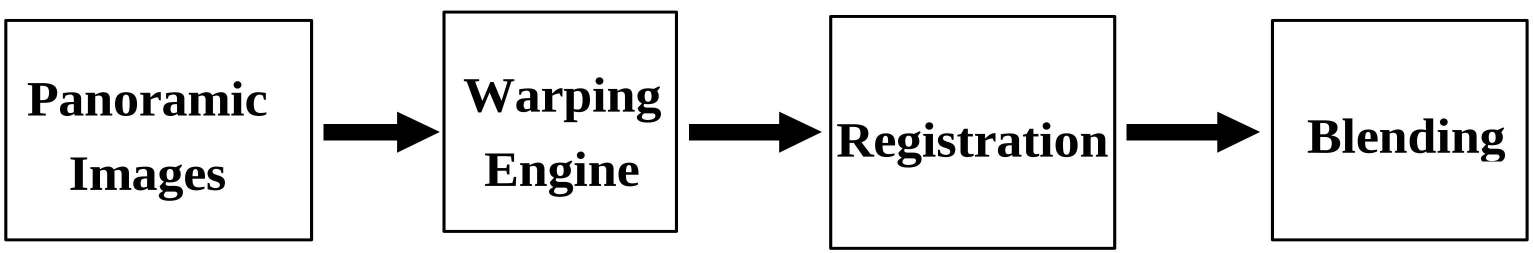


Panoramic Image Stitcher

- Warping Enviroments



- Flowchart of the proposed stitcher



A general system for generating a virtual panoramic world.

Panoramic Image Stitcher

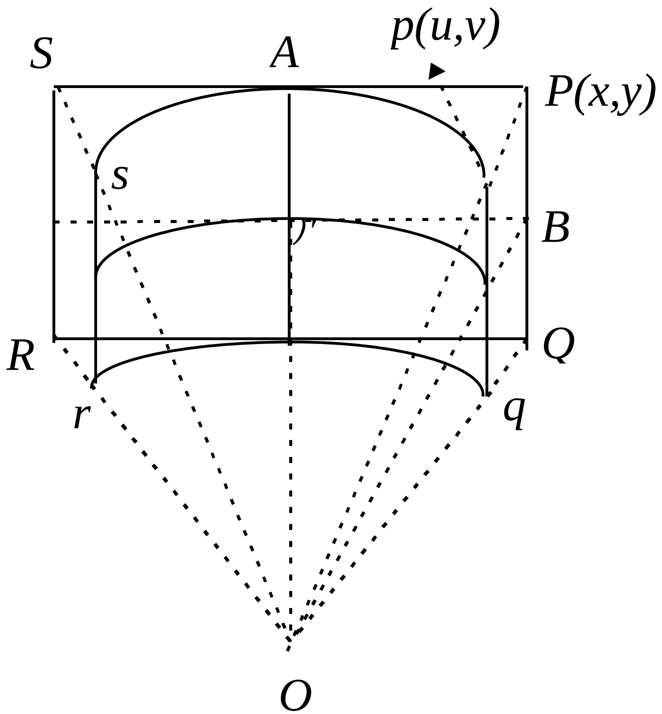
- Cylindrical Environment Map

  - *PQRS* ➡ *pqrs*
  - *P(x,y)* ➡ *p(u,v)*

$$u = f \tan^{-1} \frac{\overline{O'B}}{\overline{OO'}} = f \tan^{-1} \frac{x}{f}$$

$$v = f \frac{\overline{PB}}{\overline{OB}} = f \frac{y}{\sqrt{x^2 + f^2}}$$

f: focal length

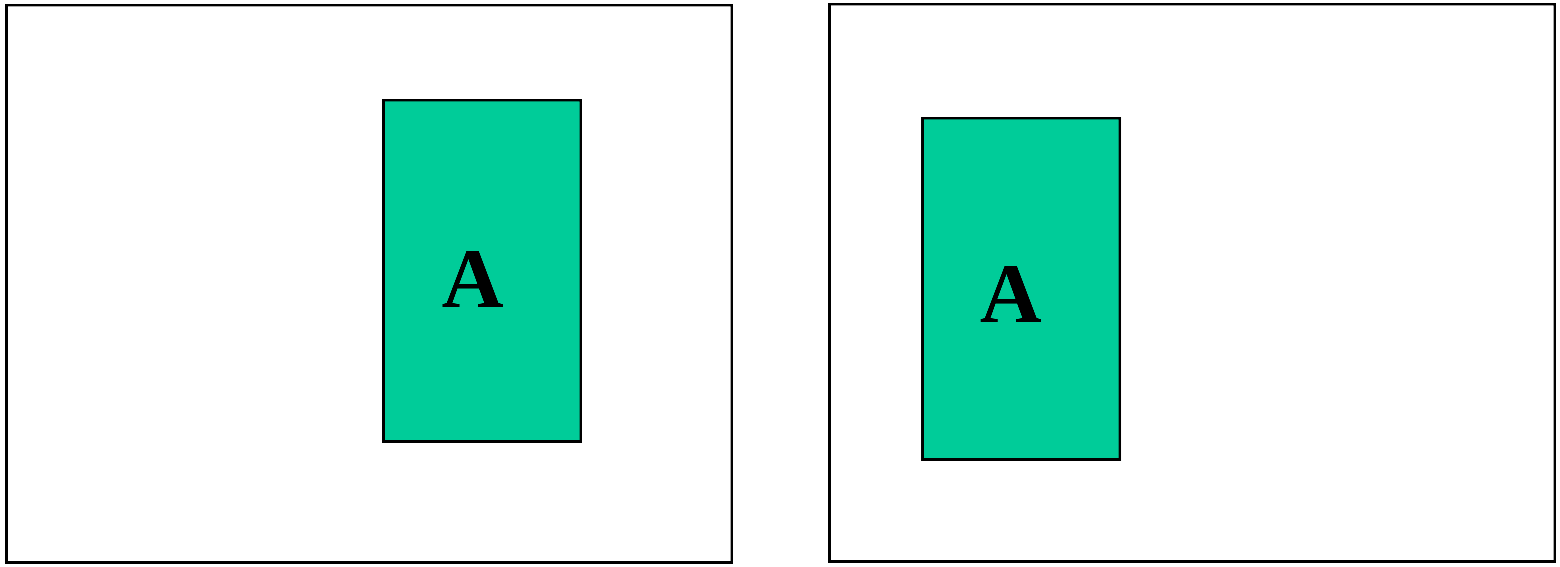


The geometry for cylindrical mapping.

Panoramic Image Stitcher

- Traditional methods:

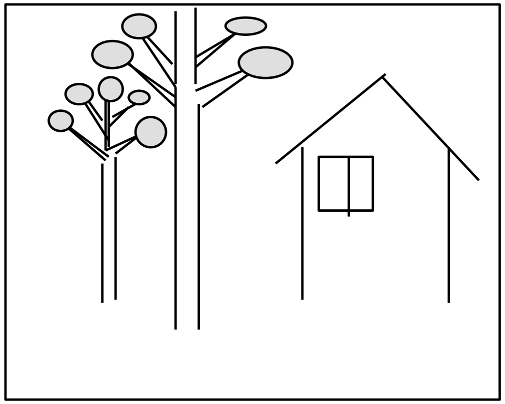
  Using a correlation technique and a full search approach to determine the optimal offsets.



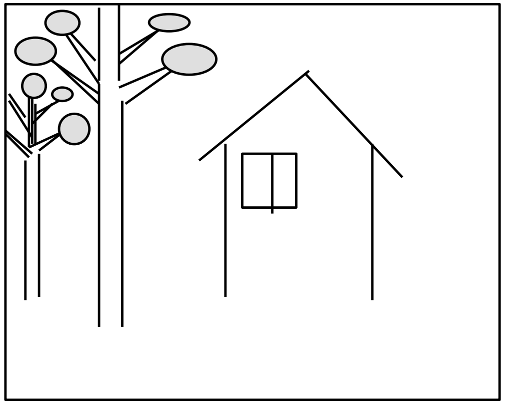
Panoramic Image Stitcher

- Our proposed medthod:

Using edge information to get a set of possible solutions in advance.



(a)



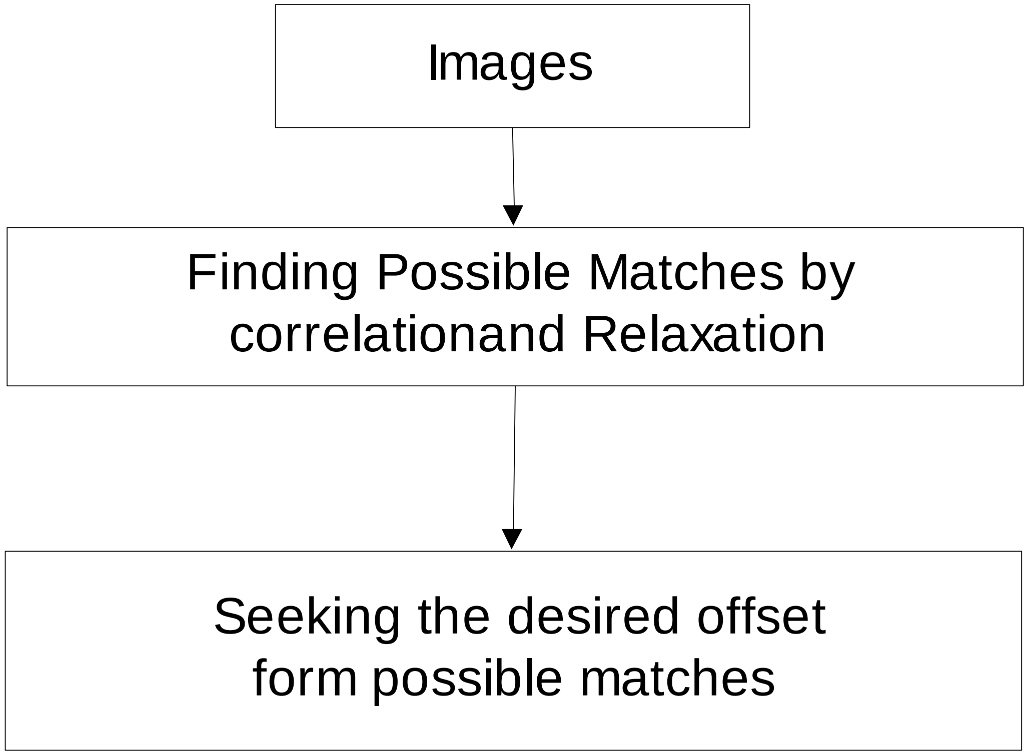
(b)

$$P_a = (100, 115, 180, 200, 310, 325, 360, 390, 470)$$

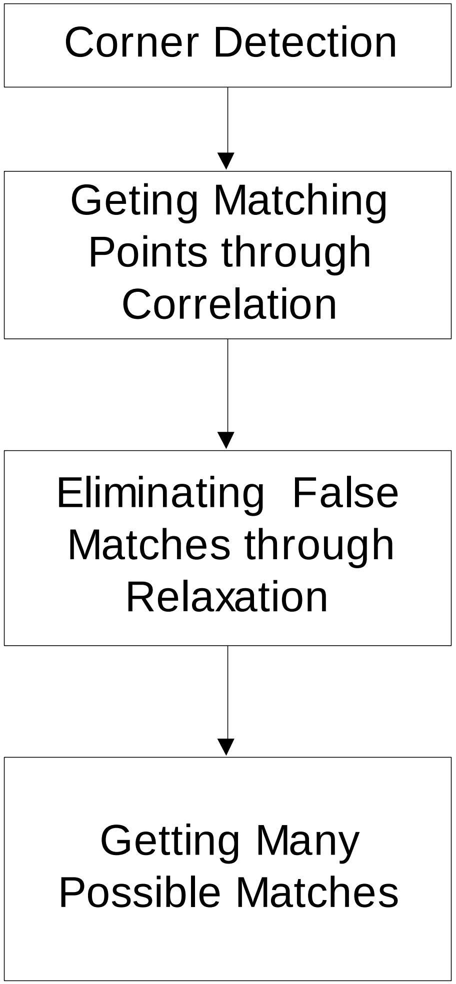
$$P_b = (20, 35, 100, 120, 230, 245, 280, 310, 390)$$

Panoramic Image Stitcher

Panoramic Image Stitcher



Panoramic Image Stitcher



Panoramic Image Stitcher

- Corner Detection

$$R(x,y)=\det[C]-0.04\,trace^{2}[C]$$

$$C=\begin{bmatrix}\bar{\hat{I}}_x^2 & I_xI_y\\ I_xI_y & \bar{\hat{I}}_y^2\end{bmatrix}$$

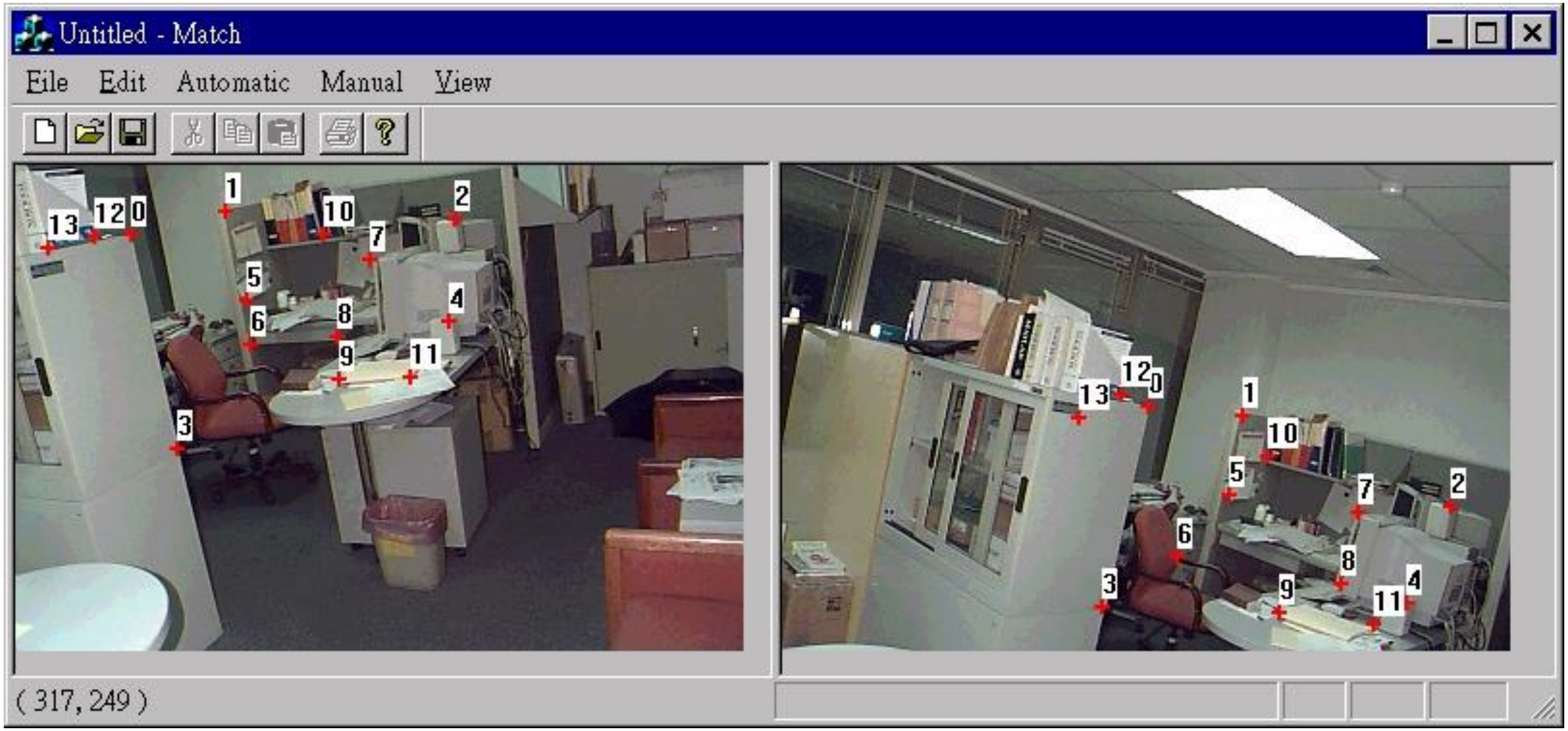
$\bar{I}$: the smoothing version of image I

$\bar{I}_x$: the x directional derivative of image I

$\bar{I}_y$: the y directional derivative of image I

Panoramic Image Stitcher

- ☐ Eliminating False Matches through Relaxation



Panoramic Image Stitcher

- Matching through Correlation

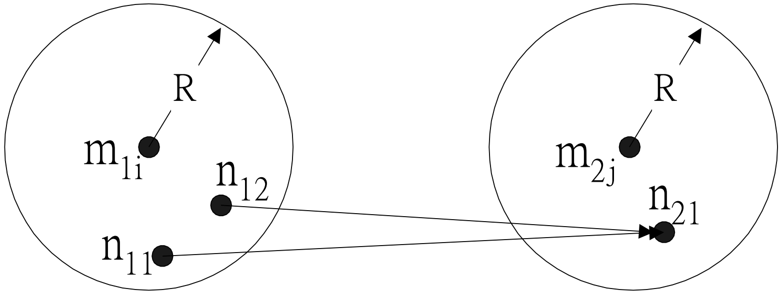
$$Corr(p;q) = \frac{1}{\sigma_1 \sigma_2 (2M+1)^2} \sum_{x,y=-M}^{x,y=M} [I_1(x+p_x, y+p_y) - u_1][I_2(x+q_x, y+q_y) - u_2]$$

u_i and σ_i: local mean and variance of I_i, respectively

$(2M+1)^2$: the area of matching window.

Panoramic Image Stitcher

☐ Eliminating False Matches through Relaxation



Measure of support for a match:

1.Strength of match $$S_M(m_{1i}, m_{2j}) = c_{ij} \sum_{n_{1k} \in N(m_{1i})} \left[\max_{n_{2j} \in N(m_{2j})} \frac{c_{kl}\delta(m_{1i}, m_{2j}; n_{1k}, n_{2l})}{1 + dist(m_{1i}, m_{2j}; n_{1k}, n_{2l})} \right]$$

$$dist(m_{1i}, m_{2j}; n_{1k}, n_{2l}) = [d(m_{1i}, n_{1k}) + d(m_{2j}, n_{2l})]/2$$

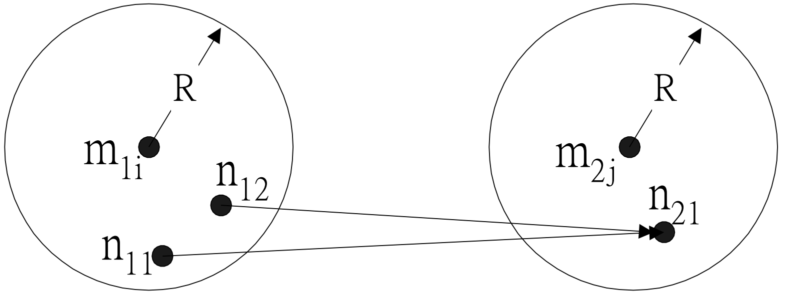
$$\delta(m_{1i}, m_{2j}; n_{1k}, n_{2l}) = \text{a function of } d(m_{1i}, n_{1k}) \text{ and } d(m_{2j}, n_{2l})$$

2. Angle consistence:

The angle between $m_{1i}n_{1k}$ and $m_{2j}n_{2k}$ should be less than a threshold θ.

Panoramic Image Stitcher

- Eliminating False Matches through Relaxation



$$\delta(m_{1i}, m_{2j}; n_{1k}, n_{2l}) = \text{a function of } d(m_{1i}, n_{1k}) \text{ and } d(m_{2j}, n_{2l})$$

$$\delta(m_{1i}, m_{2j}; n_{1k}, n_{2l}) = \begin{cases} e^{-r/\varepsilon} & \text{if } \mathrm{r} < \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

$$r = \frac{|\, d(m_{1i}, n_{1k}) - d(m_{2j}, n_{2l}) \,|}{dist(m_{1i}, m_{2j}; n_{1k}, n_{2l})}$$

$$dist(m_{1i}, m_{2j}; n_{1k}, n_{2l}) = [d(m_{1i}, n_{1k}) + d(m_{2j}, n_{2l})]/2$$

Panoramic Image Stitcher

- Eliminating False Matches through Relaxation

Relaxation Process:

Iterate {

Compute SM for each match

Delete matches whose SM value is less than the k%-smallest matching strength

Check angle consistence

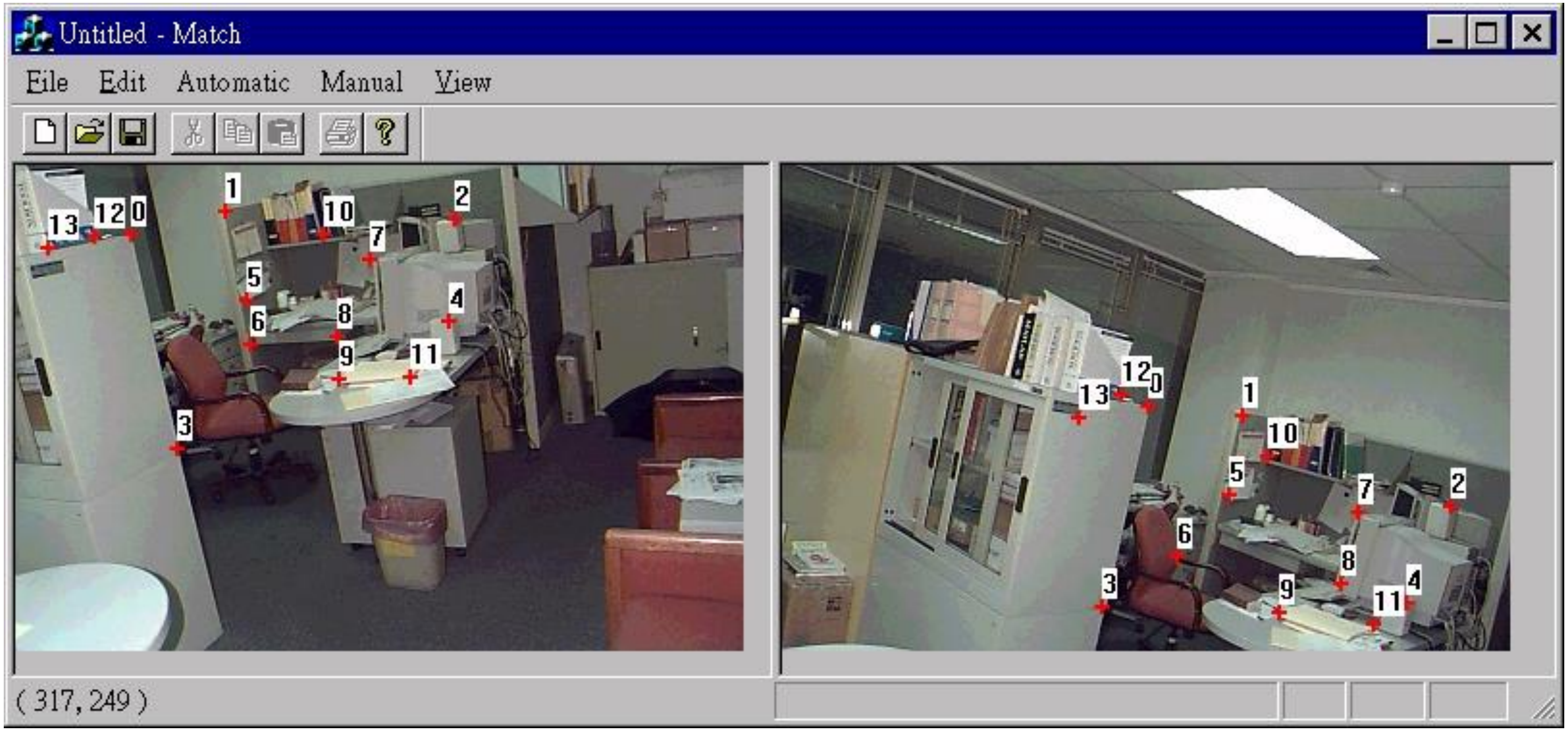
Update SM with the remained matches

}

until iteration times < a threshold

Panoramic Image Stitcher

- Eliminating False Matches through Relaxation



Panoramic Image Stitcher

□ Getting Offsets from Matching Candidates

Using “votes” to get the best offset

1. With matching strength(SM) as a measure, choose top 10 matches as candidates to get a set of possible offsets.

   Offset[I].x = Matches[I].p.x- Matches[I].q.x;

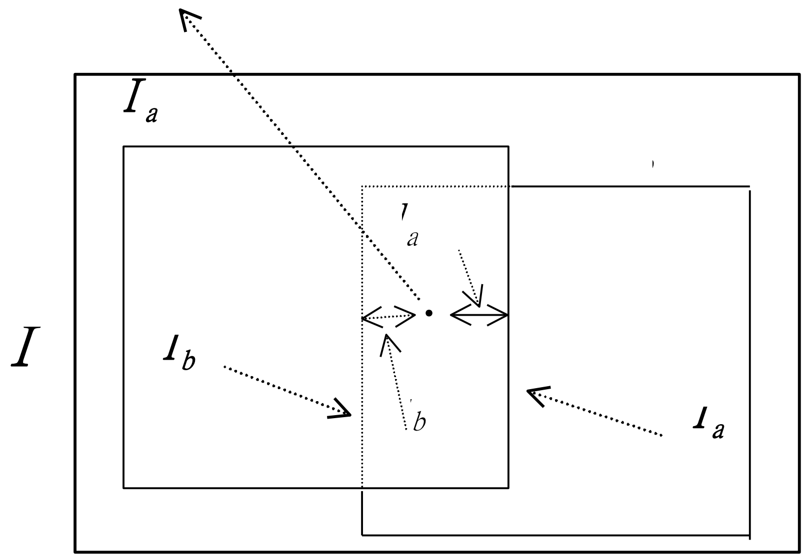
   Offset[I].y = Matches[I].p.y- Matches[I[.q.y;

2. With the set of possible offsets, get the best offset with voting from all the matches.
3. Fine-tuning with correlation.

Panoramic Image Stitcher

□ Blending Method

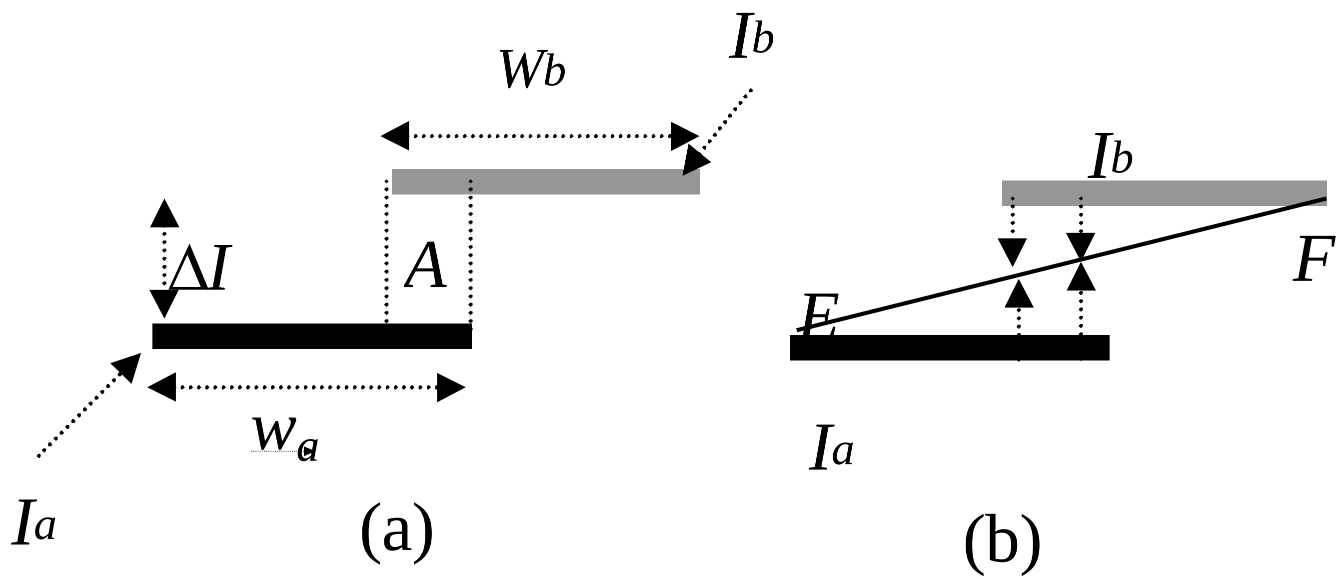
p_i, q_i, r_i (located at the same location)



$$I(r_i) = \frac{d_b^t I_a(p_i) + d_a^t I_b(q_i)}{d_a^t + d_b^t}$$

Panoramic Image Stitcher

- Blending Method

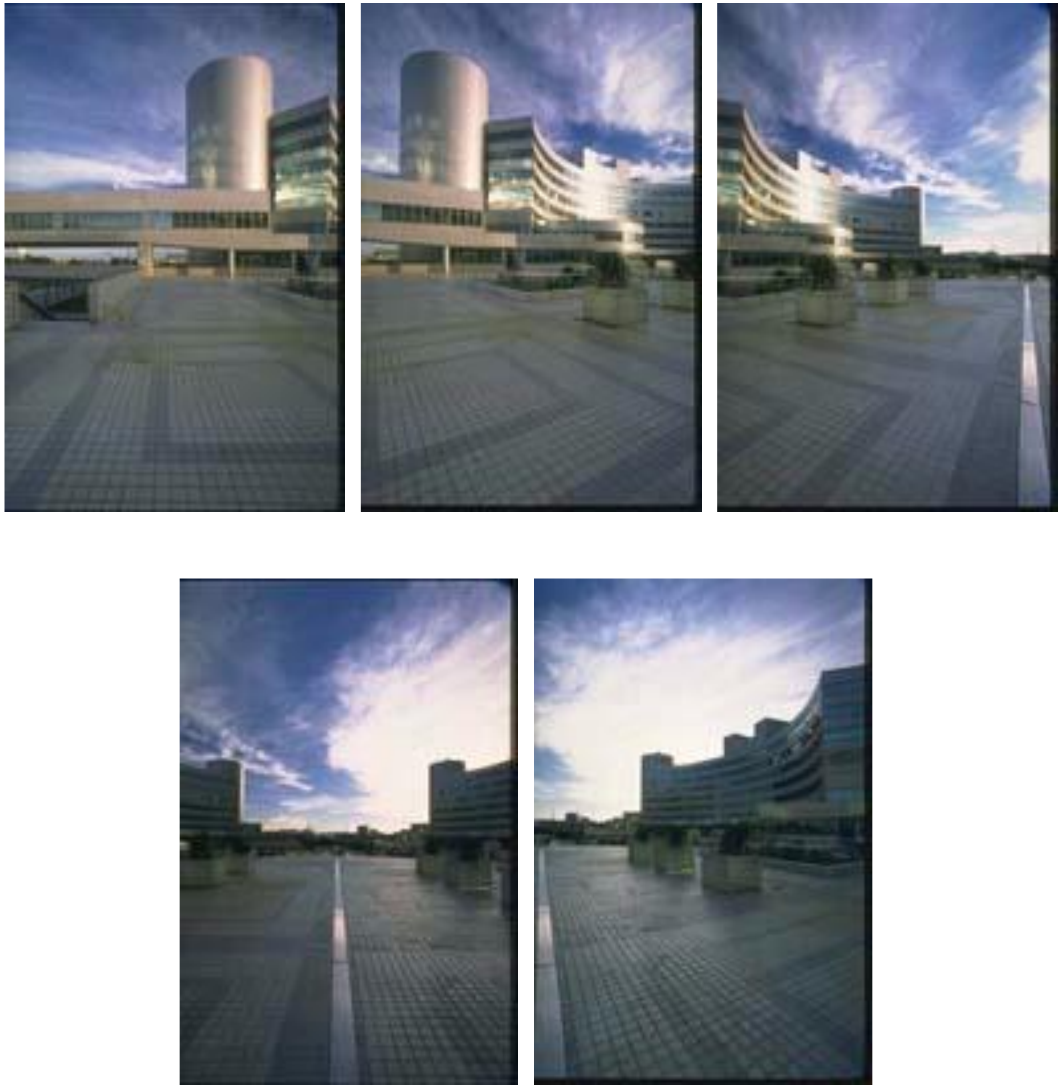


$$\Delta I = \frac{1}{|A|}\sum_{i \in A}(I_b(q(i)) - I_a(p(i))),$$

$$I_a(p(x,y)) = I_a(p(x,y)) + x\frac{\Delta I}{2w_1}$$

$$I_b(q(x,y)) = I_b(q(x,y)) + (x - w_2)\frac{\Delta I}{2w_2}$$

Panoramic Image Stitcher



A series of panoramic images for testing.

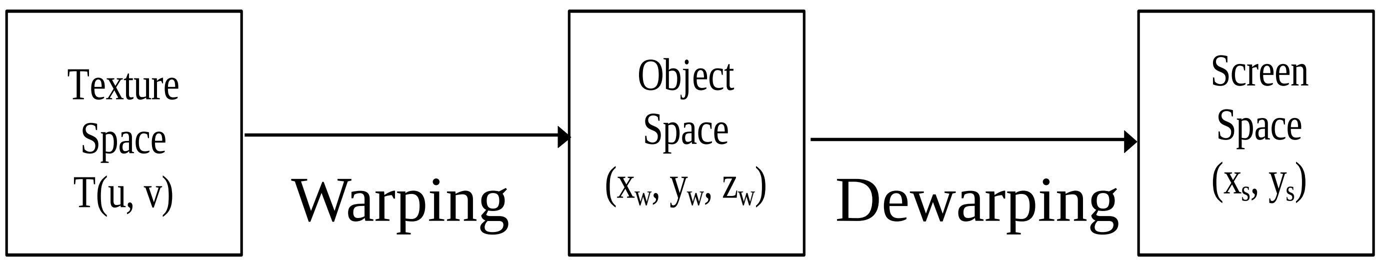
Panoramic Image Stitcher

The stitching result obtained by our intelligent stitcher.

Panoramic Image Stitcher

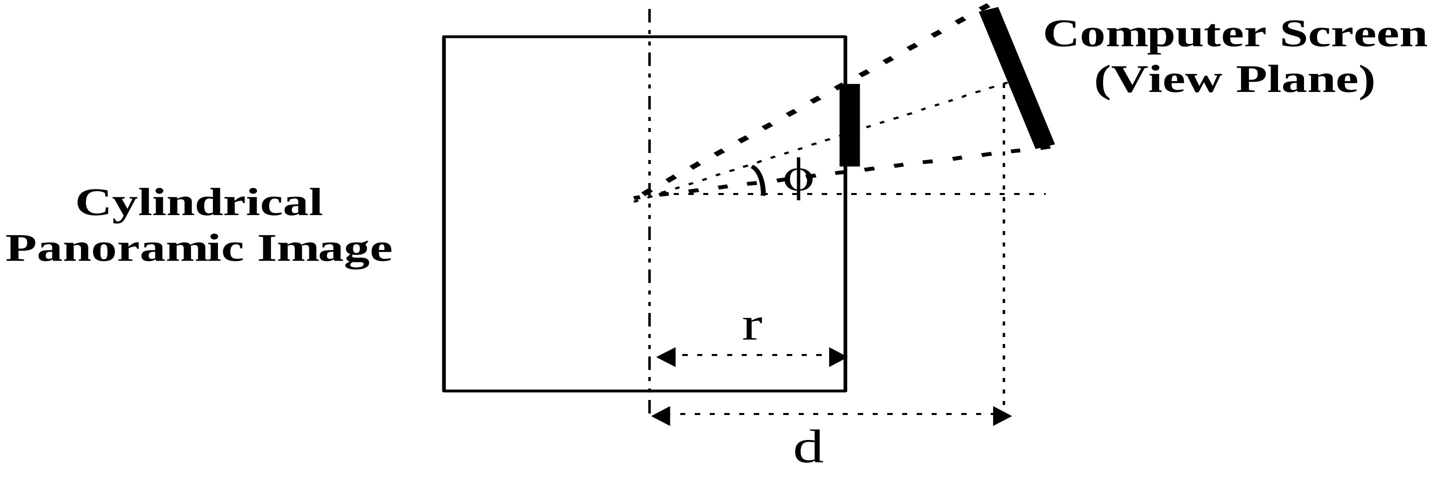
□ Dewarping Method

- 相關之技術領域
  - Texture Mapping



Panoramic Image Stitcher

- Dewarping Method
  - 環場影像虛擬實境



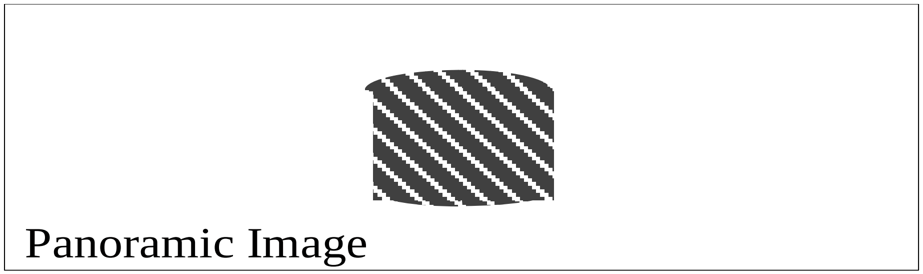
固定觀測點於幾何中心 ＋可動式投影平面

Panoramic Image Stitcher

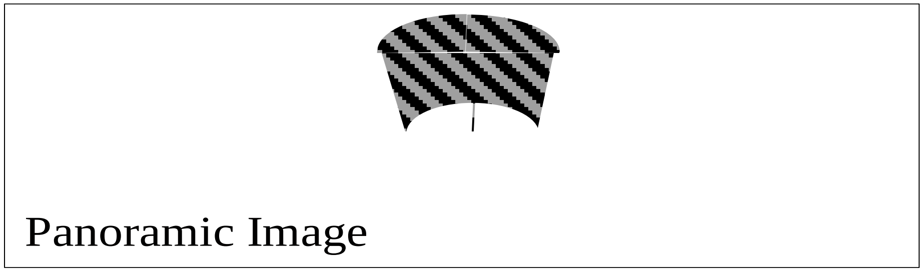
柱狀環場影像範例

Panoramic Image Stitcher

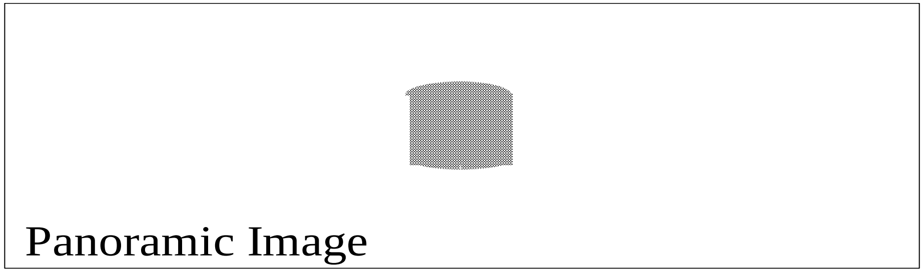
影像反扭曲(Image Dewarping)



平視



抬頭



鏡頭拉近

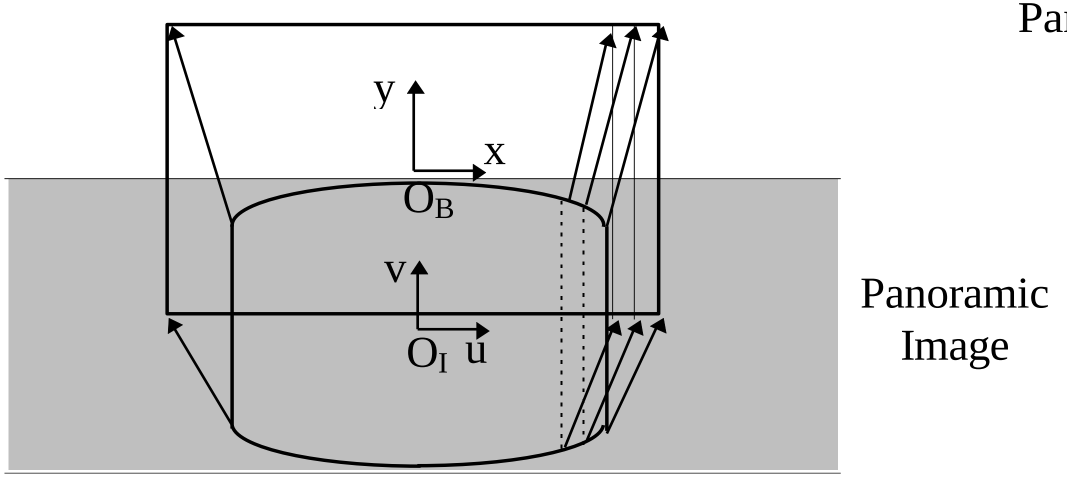
查表法之使用

- 記錄反扭曲區域內之各**垂直掃瞄線**的起點位置及終點位置
- 表中資料在抬頭/ 低頭，以及拉近/拉遠鏡頭時必須重算

Panoramic Image Stitcher

投影關係 （一）

環場影像與投影緩衝區：



Panoramic Image ↔ Projection Buffer

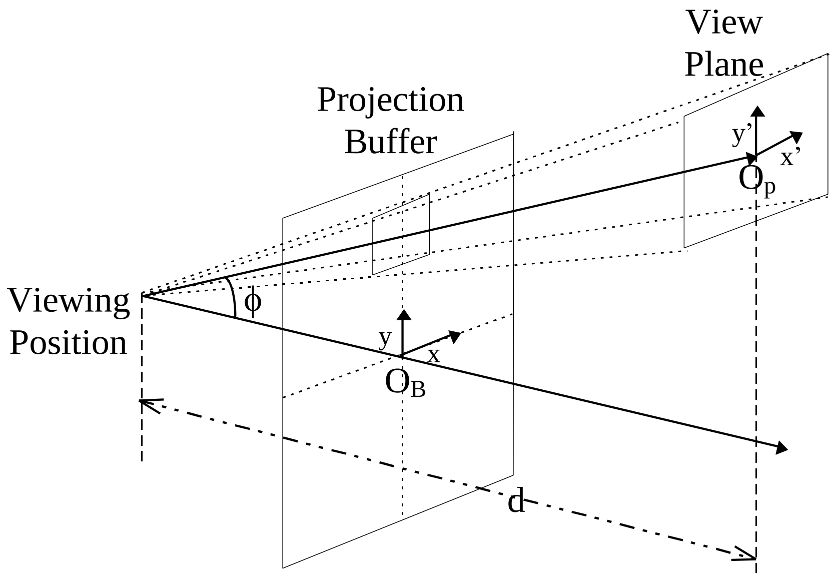
$$u = r\tan^{-1}(x/r),$$

$$v = yr/\sqrt{x^2 + r^2}.$$

Panoramic Image Stitcher

投影關係 （二）

投影緩衝區與成相平面：



Projection Buffer (x', y') ↔ View Plane (x', y')

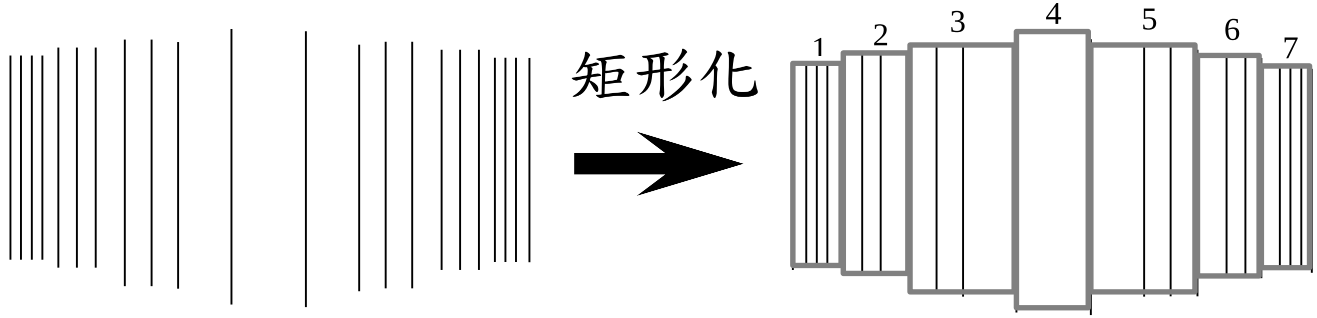
$$x = \frac{r}{d} x',$$

$$y = \frac{r}{d}\left(y' + d \tan\phi\right)$$

Panoramic Image Stitcher

查表法改進

- 使用矩形化反扭曲技術



優點：

（１）可充分利用圖形顯示卡之硬體加速器（如Blitter）
（２）可大量降低資料表之大小

Panoramic Image Stitcher

三維環場影像

1) Regular stereo

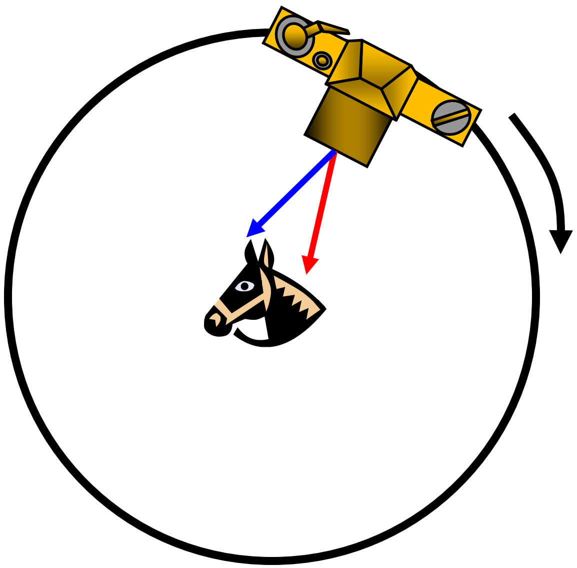
2) Generalized / non-classical camera

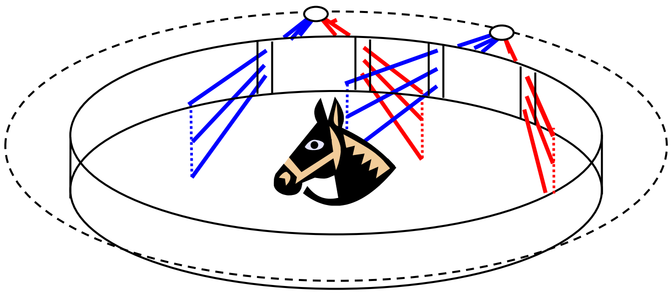
3) Stereo with generalized cameras (an example)

4) When can we compute stereo from generalized cameras?

沿著一個圓相機往內拍(Object Video)

Inward-facing camera, moving around an object

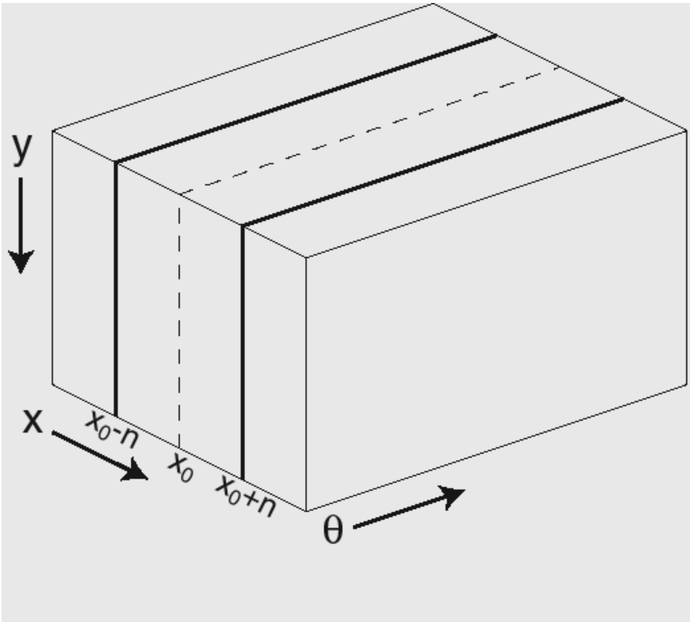




thanks to Steve Seitz, University of Washington

Images for both eyes

- Input: video sequence
- Output: 2 symmetric cuts of 3D video array



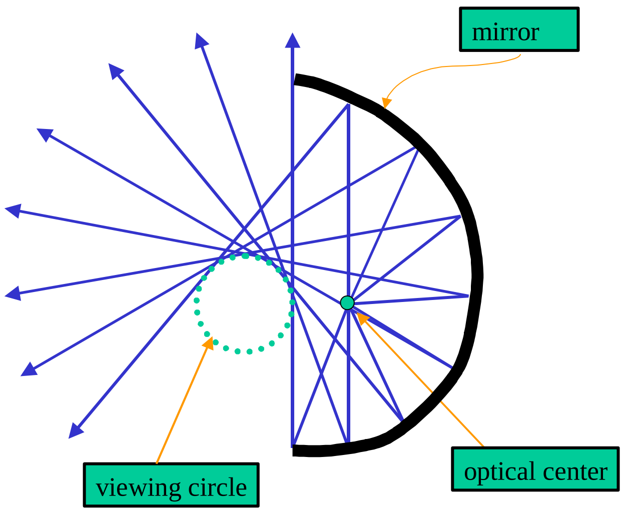
thanks to Steve Seitz, University of Washington

Results: red-blue stereo image

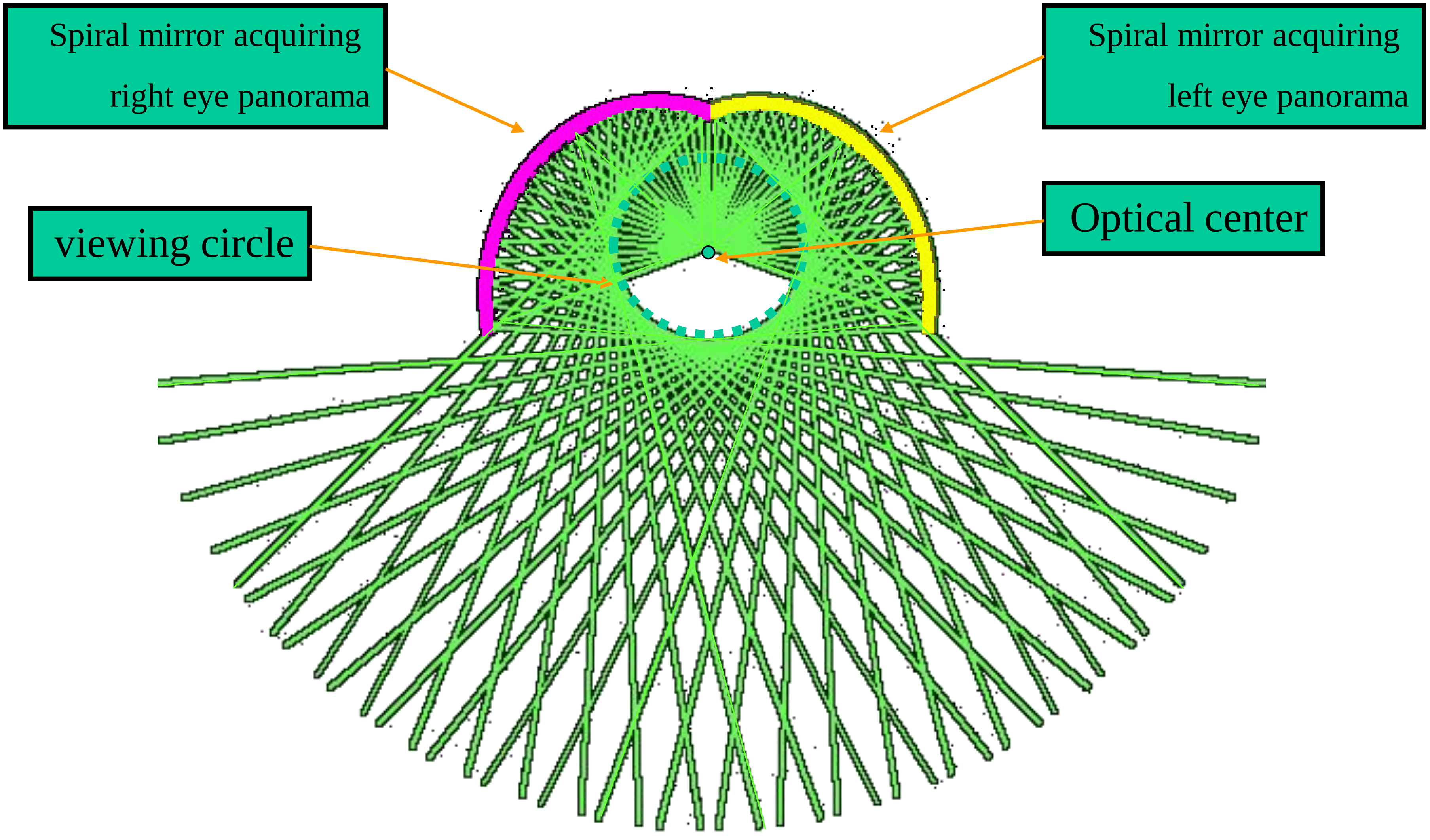
thanks to Steve Seitz, University of Washington

OmniStereo with Mirrors

Dynamic Scenes



Stereo Image



Stereo Image