視覚情報処理論　（学環）
Visual Information Processing
コンピュータビジョン　（情・電子情報）
Computer Vision
三次元画像処理特論　（情・ｺﾝﾋﾟｭｰﾀ科学）
Three-Dimensional Image Processing

2013/12/4（水）16:30-18:00

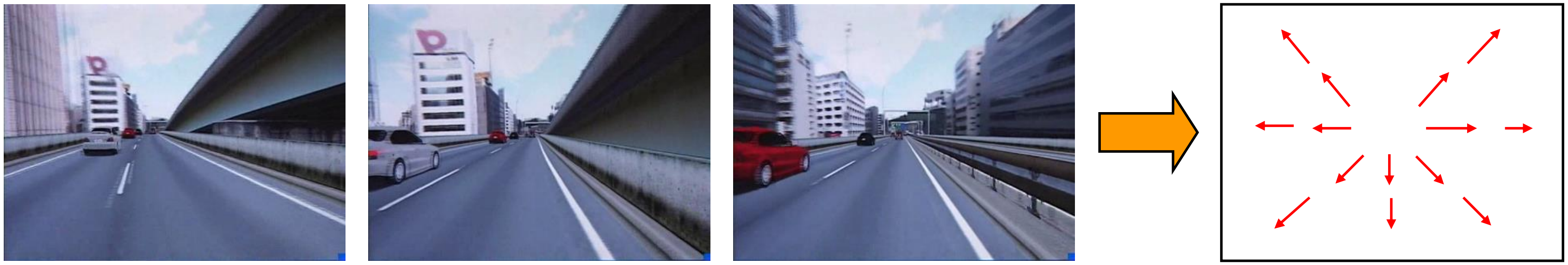
池内 克史（大学院情報学環 教授）

代理:
小野 晋太郎
（生産技術研究所 特任准教授、博士（情報理工学））

東京大学
THE UNIVERSITY OF TOKYO

Introduction

- How to obtain a motion field
  - Optical flow
  - Apparent motion of the brightness pattern
  - 2D problem
- How to characterize and what information can be obtained from a motion field
  - Structure from motion
  - 3D understanding from 2D



Time-varying Image Processing

- Introduction
- Basic technologies
  - Background subtraction
  - Optical flow
  - Structure from Motion (SfM)
  - Space-time Image Analysis
- Applied technologies
  - Introducing recent research cases

This time

Next time

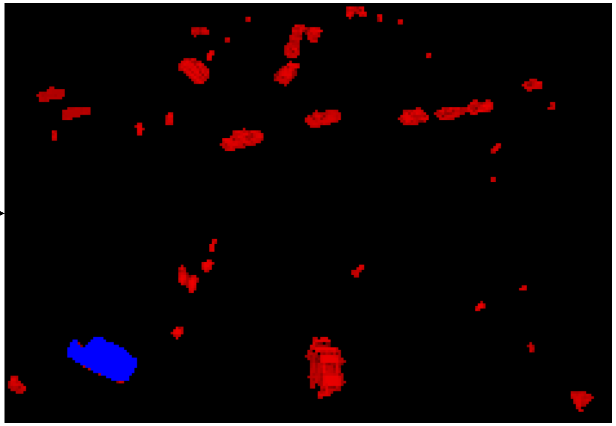
Motion understanding #1
動き解析・動画像処理　第1話

東京大学
THE UNIVERSITY OF TOKYO

Background Subtraction

Background Subtraction (Simplest Model)

Input image



Foreground image

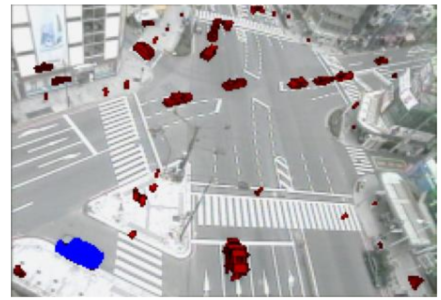
Background image

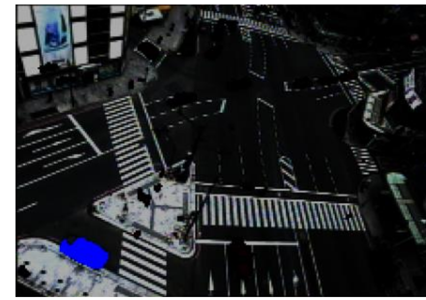
$$|I - I_{BG}| > \text{Threshold ?}$$

Using appropriate color space (RGB, HSV, YCbCr, …)

Problems in the Simplest Model

- Sensitive to lighting change
  - Sunlight change
  - Turning on/off lamp
  - Camera's auto exposure
- Same threshold for all pixels
- Objects moving periodically are identified as foreground
  - Leaves of trees
  - Signal lights, …





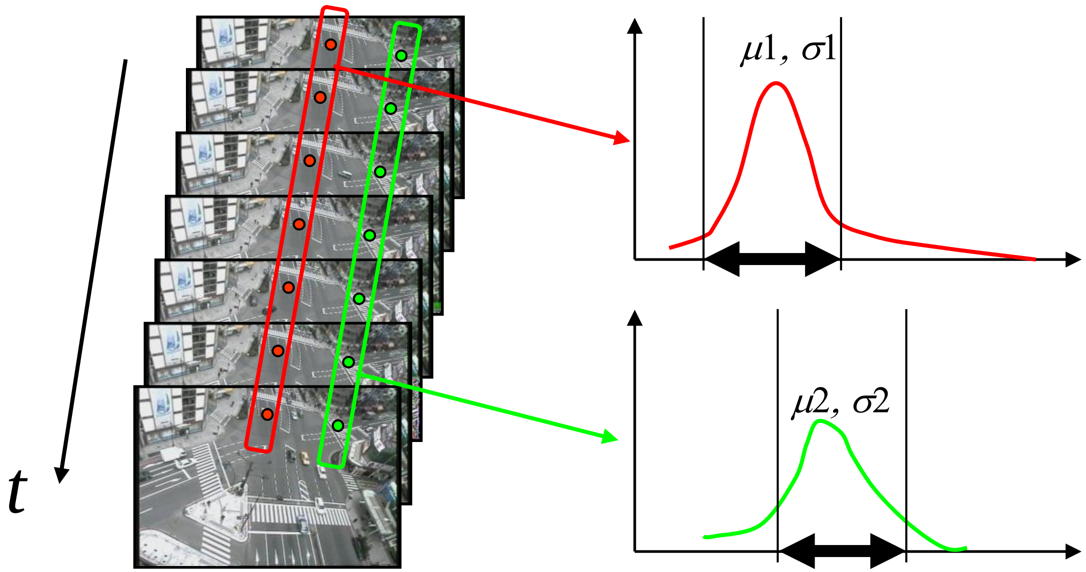
Intensity variance in background
Adaptive threshold

Normal Distribution Model in Background

Prior data

Intensity histogram (per-pixel)

Per-pixel threshold



$$|I - \mu| > k\,\sigma \quad ?$$

Another problem:
Still object appeared after is identified as foreground forever

Dynamic background update

Dynamic Background Update

- Potential Background (in the near future)
  - Objects identified as foreground for long duration
  - Non-moving objects

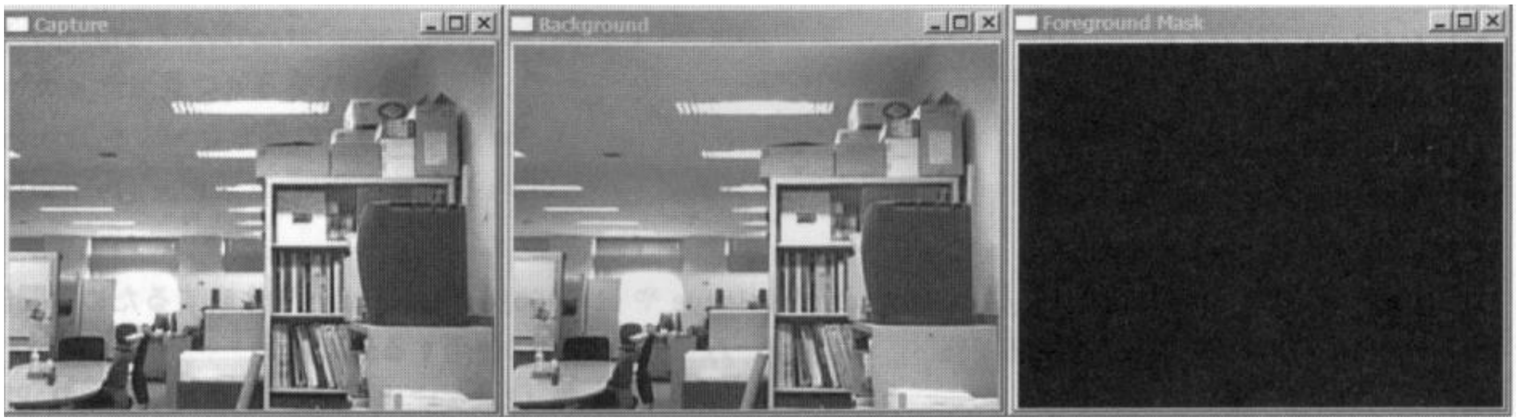
- Update process (Example)
  - Foreground → Slightly mixed to Background
  - Potential Background → Replace current Background

Dynamic Background Update (Example)

[OpenCV Programming Book]

Input **BG** **FG**

t



Initial State

Working as the ordinary background subtraction

Dynamic Background Update (Example)

[OpenCV Programming Book]

Input **BG** **FG**

t

The poster added has been identified as FG for long, and is not moving…

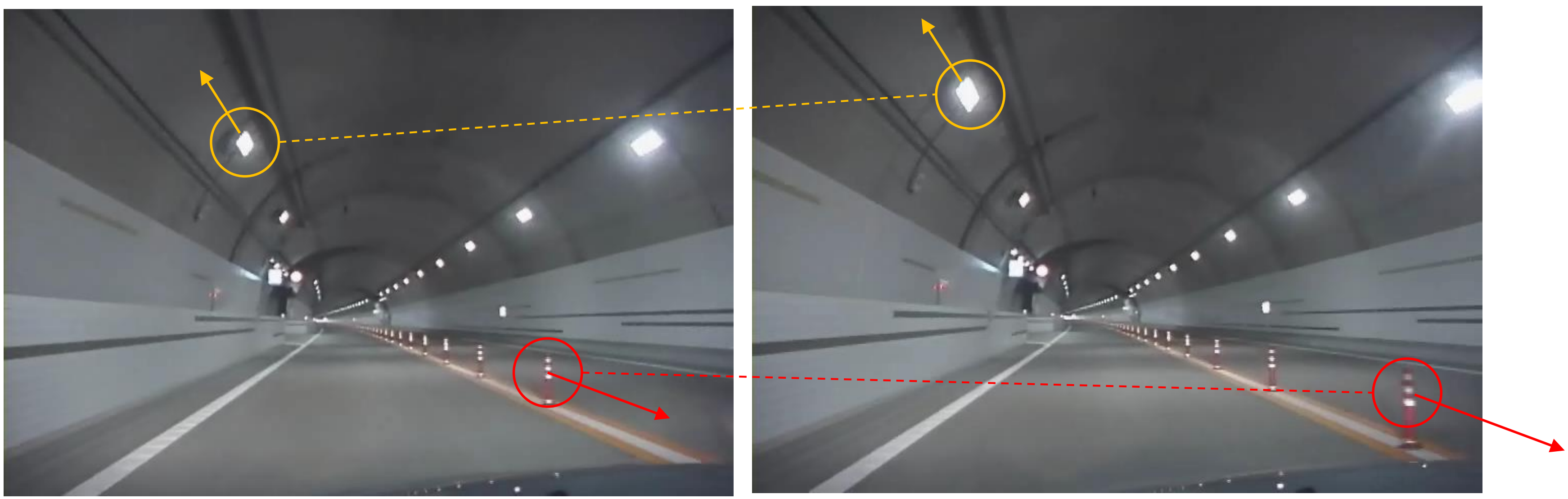
BG is updated

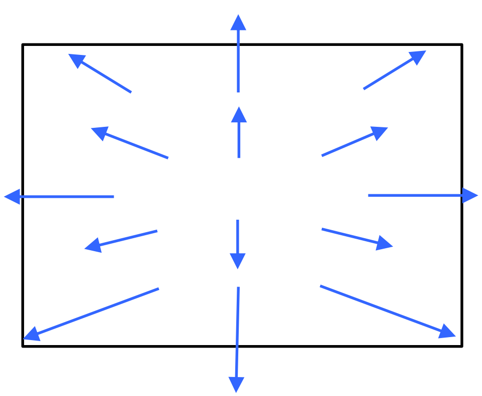
Optical Flow

東京大学
THE UNIVERSITY OF TOKYO

Optical Flow Example

Time t Time $t+\Delta t$

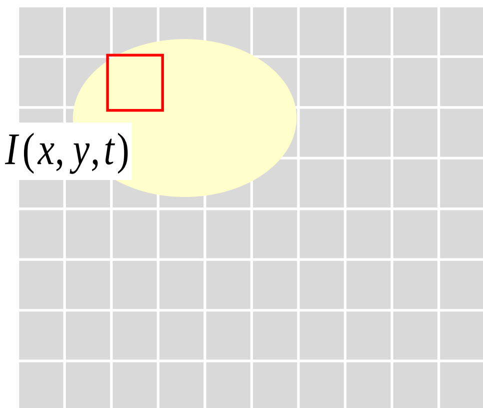




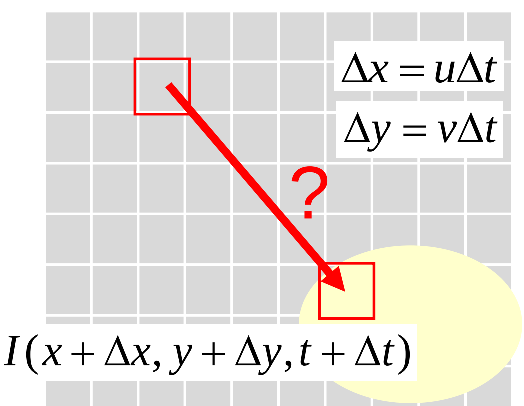
Solve motion field
For each pixel

Optical Flow Constraint Equation

Time t



Time $t+\Delta t$



Brightness conservation

$$I(x,y,t) = I(x+\Delta x, y+\Delta y, t+\Delta t)$$

Taylor expansion

$$I(x+\Delta x, y+\Delta y, t+\Delta t)$$
$$= I(x,y,t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t$$
$$= I(x,y,t) + I_x u\Delta t + I_y v\Delta t + I_t \Delta t$$

Optical flow constraint equation

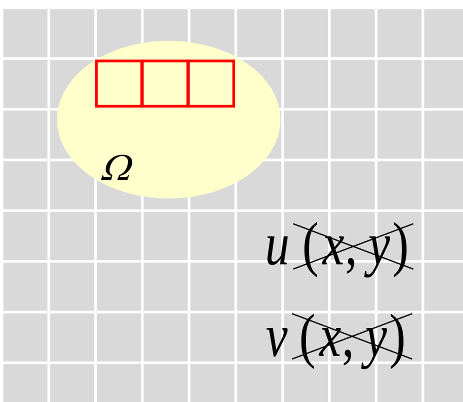
$$I_x u + I_y v + I_t = 0$$

For each pixel (x, y),
Two unknown variables $u(x,y), v(x,y)$
With one constraint equation

Solution 1: [Lucas&Kanade 1984] Same Motion in Local Region

A local region Ω moves in the mass

Sampling points (1, 2, 3, …) inside the region have the same u, v



Simultaneous equation

$$I_x^1 u + I_y^1 v = -I_t^1$$

$$I_x^2 u + I_y^2 v = -I_t^2$$

$$I_x^3 u + I_y^3 v = -I_t^3$$

$$\vdots$$

Solved as a (weighted) least squares method

Solution 1: [Lucas&Kanade 1984] Same Motion in Local Region

Or else,

$$I_x u + I_y v + I_t = 0$$ for all (x, y) inside local region Ω

$$e = \iint_{\Omega} \{I_x(x,y)u + I_y(x,y)v + I_t(x,y)\}^2 dxdy \rightarrow \min$$

$$\frac{\partial e}{\partial u} = 0 \qquad u\iint I_x^2 dxdy + v\iint I_x I_y dxdy = -\iint I_t I_x dxdy$$

$$\frac{\partial e}{\partial v} = 0 \qquad u\iint I_x I_y dxdy + v\iint I_y^2 dxdy = -\iint I_t I_y dxdy$$

Limitation

- Defining the mass region to be small
  - Solution (u, v) becomes unstable
- Defining the mass region to be large
  - The assumption "Region move in the mass" will be fail
- These are trade-off's

Solution 2: [Horn & Schunck 1981] Motion Smoothness Constraint

Optical flow equation: $I_x u + I_y v + I_t = 0$

Smoothness constraint: Neighboring pixels have similar motions

$$u_x^2 + u_y^2 + v_x^2 + v_y^2 \rightarrow \min$$

$$e = \iint \left\{ (I_x u + I_y v + I_t)^2 + \lambda (u_x^2 + u_y^2 + v_x^2 + v_y^2) \right\} dx\,dy \rightarrow \min$$

Solution 2: [Horn & Schunck 1981] Motion Smoothness Constraint

$$\begin{cases} \dfrac{\partial e}{\partial u} = 0 \\ \dfrac{\partial e}{\partial v} = 0 \end{cases} \Rightarrow \begin{cases} u(x,y) = \bar{u}(x,y) - I_x \dfrac{I_x \bar{u}(x,y) + I_y \bar{v}(x,y) + I_t}{4\lambda + I_x^2 + I_y^2} \\ v(x,y) = \bar{v}(x,y) - I_y \dfrac{I_x \bar{u}(x,y) + I_y \bar{v}(x,y) + I_t}{4\lambda + I_x^2 + I_y^2} \end{cases}$$

$$\bar{u}(x,y) = \frac{1}{4}\{u(x+1,y) + u(x-1,y) + u(x,y+1) + u(x,y-1)\}$$

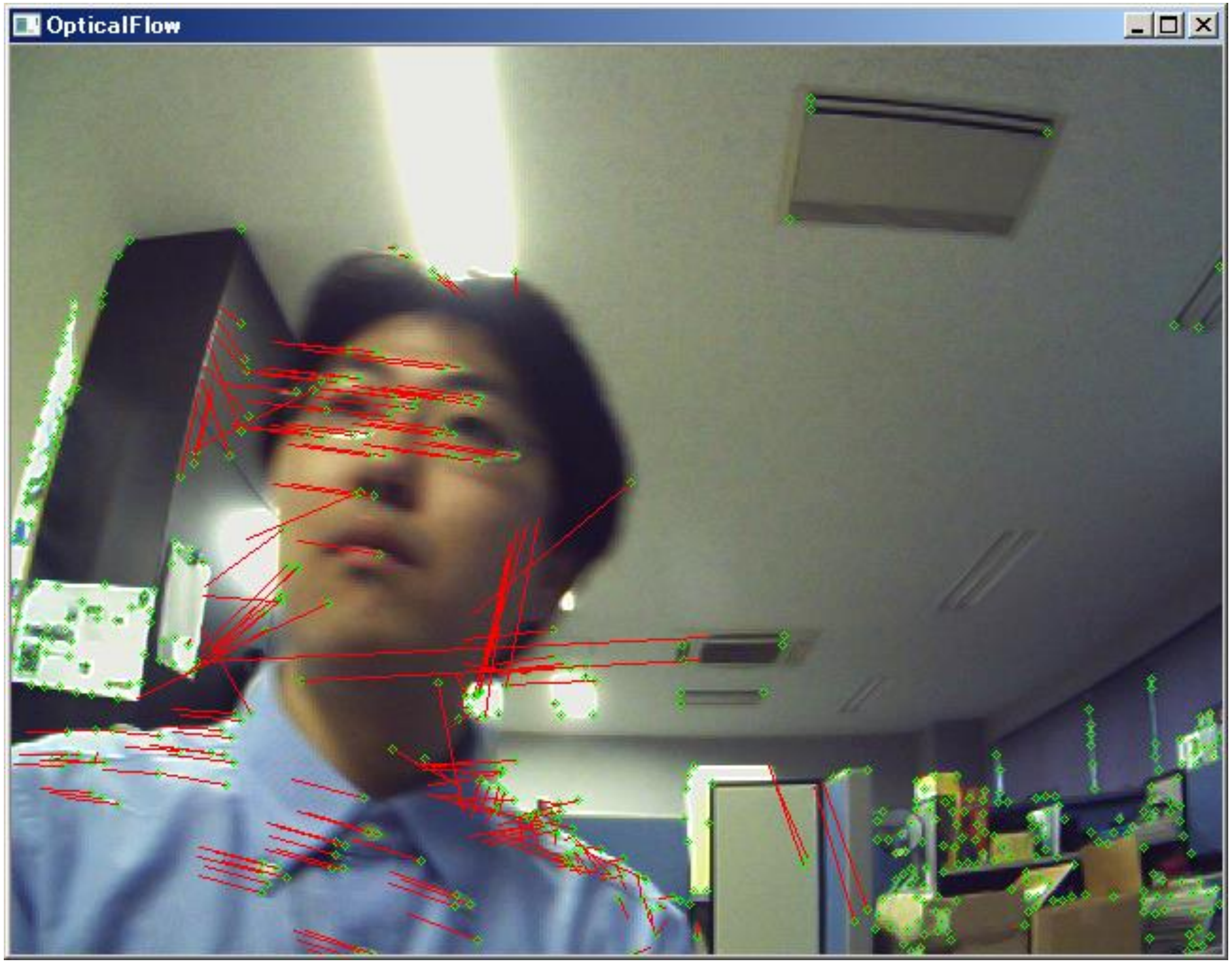
$$\bar{v}(x,y) = \frac{1}{4}\{v(x+1,y) + v(x-1,y) + v(x,y+1) + v(x,y-1)\}$$

Solved by iterative calculus
("Relaxation method")

$$u^{(k+1)} = u^{(k)} - I \frac{I_x \bar{u}^{(k)} + I_y \bar{v}^{(k)} + I_t}{4\lambda + I_x^2 + I_y^2}$$

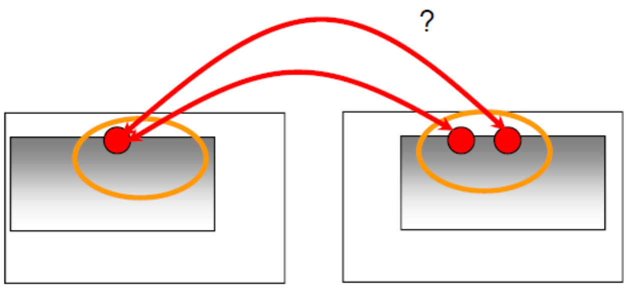
$$u_0 \rightarrow u_1 \rightarrow \cdots$$

Example



Limitation of the Optical Flow

- No solution in texture-less regions
- Large error in non-continuous region such as object boundary
- Difficulty in specifying unique correspondence (Aperture Problem)



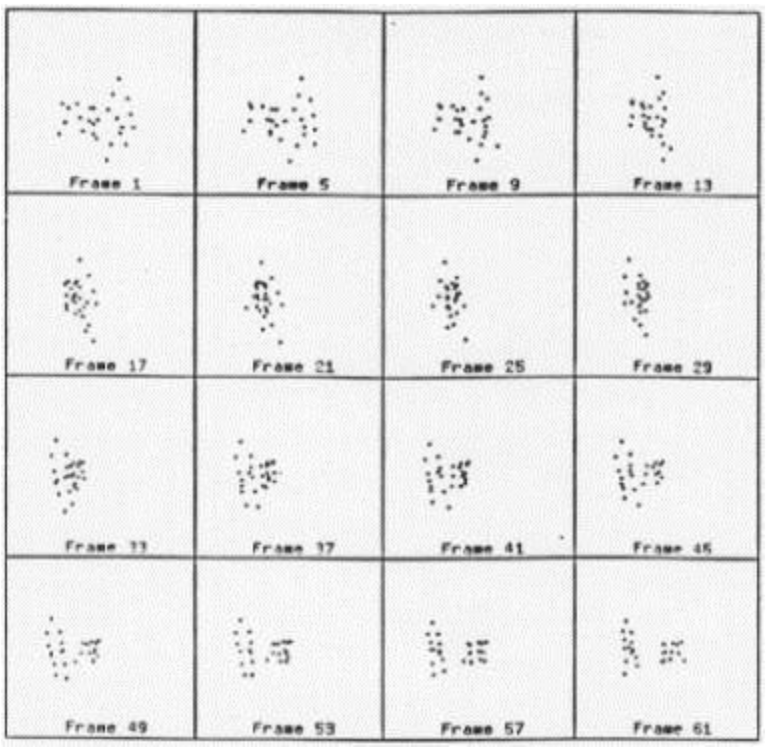
3D Reconstruction from Moving Images

東京大学
THE UNIVERSITY OF TOKYO

Is it possible to reconstruct 3D structure only from video?

- Some other knowledge:
  - When looking outside through a window of a train
    - Telegraph poles → rapidly pass
    - Mt. Fuji → can be seen during long time
  - When looking at two poles; one is near, the other is far
    - How do they appear in position, if the camera moves
    - When camera *pan …?*
    - When camera *transition …?*

Johannson’s experiment



- Put LED on each joint of a human body and observe them in the dark room.
- While the human is still, an observer cannot recognize what the pattern is.
- Immediately after the human begins to move, a sequence gives not only a compelling perception of motion of a 3D body, but allows recognition of the sequence as depicting a walking person, and a description of the type of motion.

"Structure from Motion" (SfM)

Recovering 3D Scene

Captured in a Video

- Obtain 3D structure information from 2D image sequence

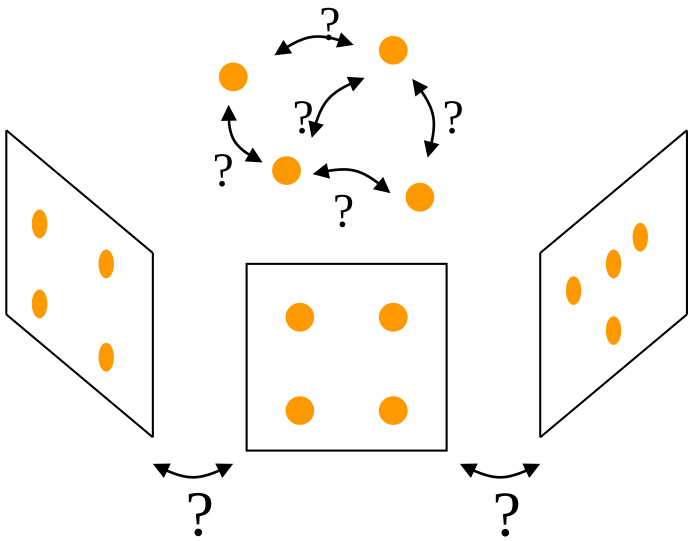
  → Similar to stereo vision, however,

  AT THE SAME TIME,

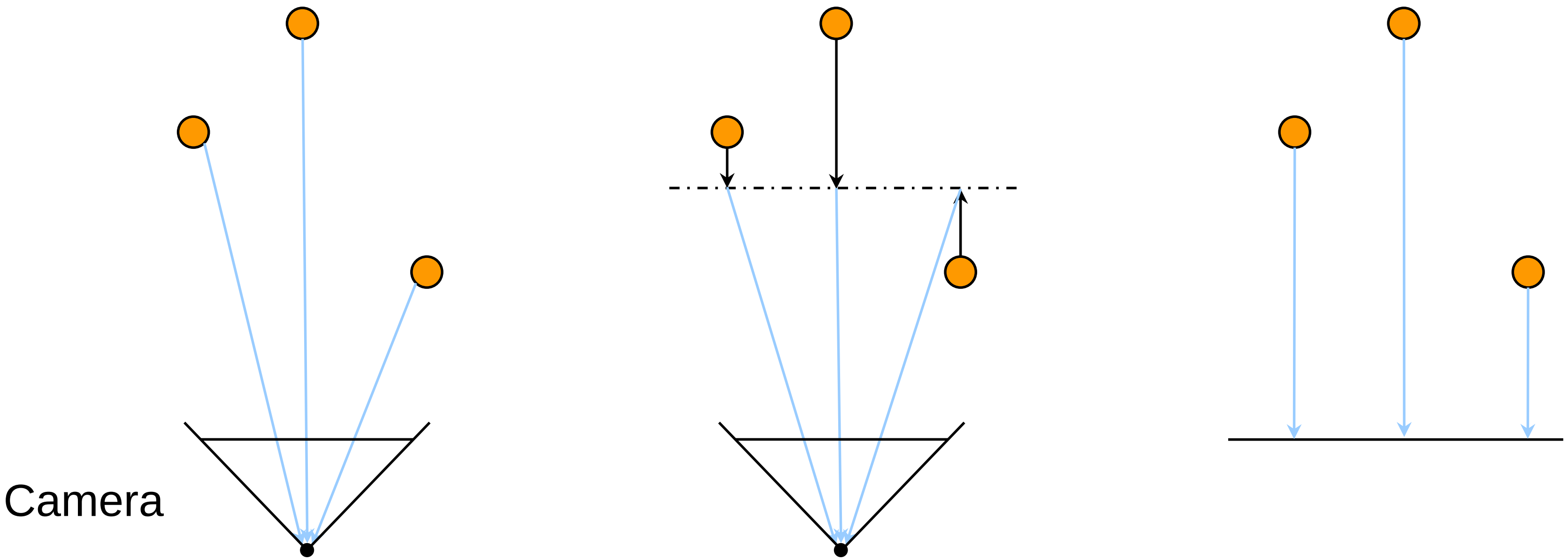
- Obtain camera's 3D motion (position and posture) from 2D image sequence

"Structure from Motion" (SfM)

- Input:
  - (More than) 3 *orthographic* or *weak-perspective* cameras
  - (More than) 4 non-coplanar points in a rigid configuration on each images
- Output:
  - 3D position of the points
  - 3D pose/position of the cameras



Camera Projection Model



Perspective

General model

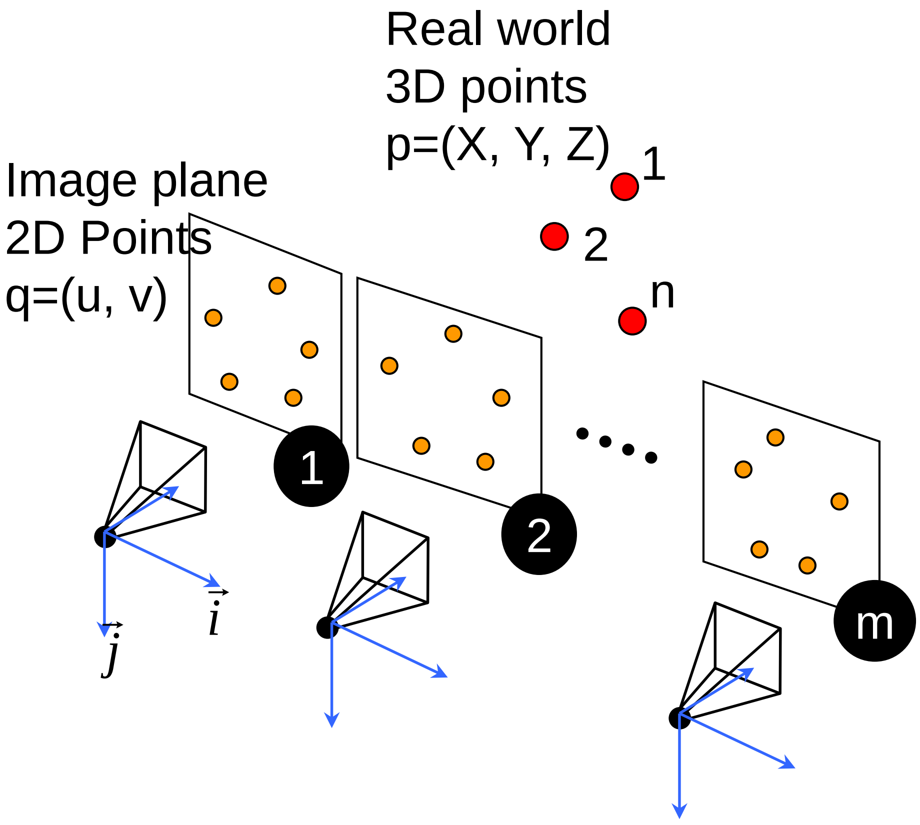
Weak-Perspective

Good if the scene depth is not varied

Orthographic (Parallel)

Good if the scene depth is very large

Basic Idea



Camera projection model (orthographic)

$$\begin{pmatrix} u \\ v \end{pmatrix} = \left(\text{Camera parameters} \right) \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \left(\text{Camera parameters} \right)$$

Want to know
motion (the camera parameters)
and structure (X, Y)
for all points and image frames

$$\sum \left| \begin{pmatrix} u \\ v \end{pmatrix} - \underbrace{\left(\left(\quad \right) \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \left(\quad \right) \right)}_{\text{Unknown}} \right|^2 \to \min$$

Simplification by variable transformation

- Real world origin: Centroid of 3D points
- Image plane origin: Centroid of 2D points

$$\sum \left| \begin{pmatrix} u' \\ v' \end{pmatrix} - \left(\left(\quad \right) \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} \right) \right|^2 \to \min$$

Hereafter, X' is described as X

SfM Theorem: Tomasi–Kanade Factorization

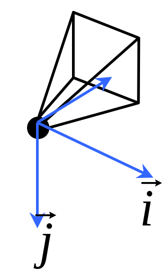
$$\sum_{\substack{\text{All points} \\ \text{All frames}}} \left| \begin{pmatrix} u \\ v \end{pmatrix} - \underbrace{\begin{pmatrix} & M & \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}}_{\text{Unknown}} \right|^2 \rightarrow \min$$

It can be minimized
if and only if we can find
the unknown M and X, Y, Z
that can decompose the W,
a set of the known u, v,
as follows:

Points

Frames

$$\begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ v_{11} & v_{12} & \cdots & v_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{m1} & u_{m2} & \cdots & u_{mn} \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{pmatrix} = \begin{pmatrix} i_{1x} & i_{1y} & i_{1z} \\ j_{1x} & j_{1y} & j_{1z} \\ \vdots & \vdots & \vdots \\ i_{mx} & i_{my} & i_{mz} \\ j_{mx} & j_{my} & j_{mz} \end{pmatrix} \begin{pmatrix} X_1 & X_2 & \cdots & X_n \\ Y_1 & Y_2 & \cdots & Y_n \\ Z_1 & Z_2 & \cdots & Z_n \end{pmatrix}$$



Observation Matrix W

Motion Matrix (camera pose) M

Shape Matrix S

Ideally it can be decomposed (Rank W = 3), but not because of observation noise

SfM Theorem: Tomasi–Kanade Factorization

It is known that as a computational technique, *Singular Value Decomposition (SVD)* can give the optimal approximation.

$$W = UDV^T \longrightarrow W' = U_{2m\times 3} D_{3\times 3} V_{3\times n}{}^T$$

$$D = \begin{pmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \sigma_3 & & \\ & & & \sigma_4 & \\ & & & & \ddots \end{pmatrix}$$

σ_1 Largest singular value

σ_2 2nd largest

σ_3 3rd largest

$\sigma_4, \ddots$ Quite small

$$D_{3\times 3} = \begin{pmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{pmatrix}$$

Skipping 4th largest or smaller singular values and vectors

W' is the nearest to W, with its rank 3.

SfM Theorem: Tomasi–Kanade Factorization

$$W = UDV^T \quad \rightarrow \quad \underbrace{U_{2m\times 3}}_{MA} D_{3\times 3} \underbrace{V_{3\times n}^{T}}_{A^{-1}S}$$

$$\begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ v_{11} & v_{12} & \cdots & v_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{m1} & u_{m2} & \cdots & u_{mn} \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{pmatrix} \Rightarrow \begin{pmatrix} i_{1x} & i_{1y} & i_{1z} \\ j_{1x} & j_{1y} & j_{1z} \\ \vdots & \vdots & \vdots \\ i_{mx} & i_{my} & i_{mz} \\ j_{mx} & j_{my} & j_{mz} \end{pmatrix} \begin{pmatrix} \\ 3\times 3 \\ \\ \end{pmatrix} \begin{pmatrix} X_1 & X_2 & \cdots & X_n \\ Y_1 & Y_2 & \cdots & Y_n \\ Z_1 & Z_2 & \cdots & Z_n \end{pmatrix}$$

A can be solved by the “metric constraint”, i.e.

$$\vec{i} \perp \vec{j}$$

$$|\vec{i}| = |\vec{j}| = 1$$

SfM in Perspective Projection

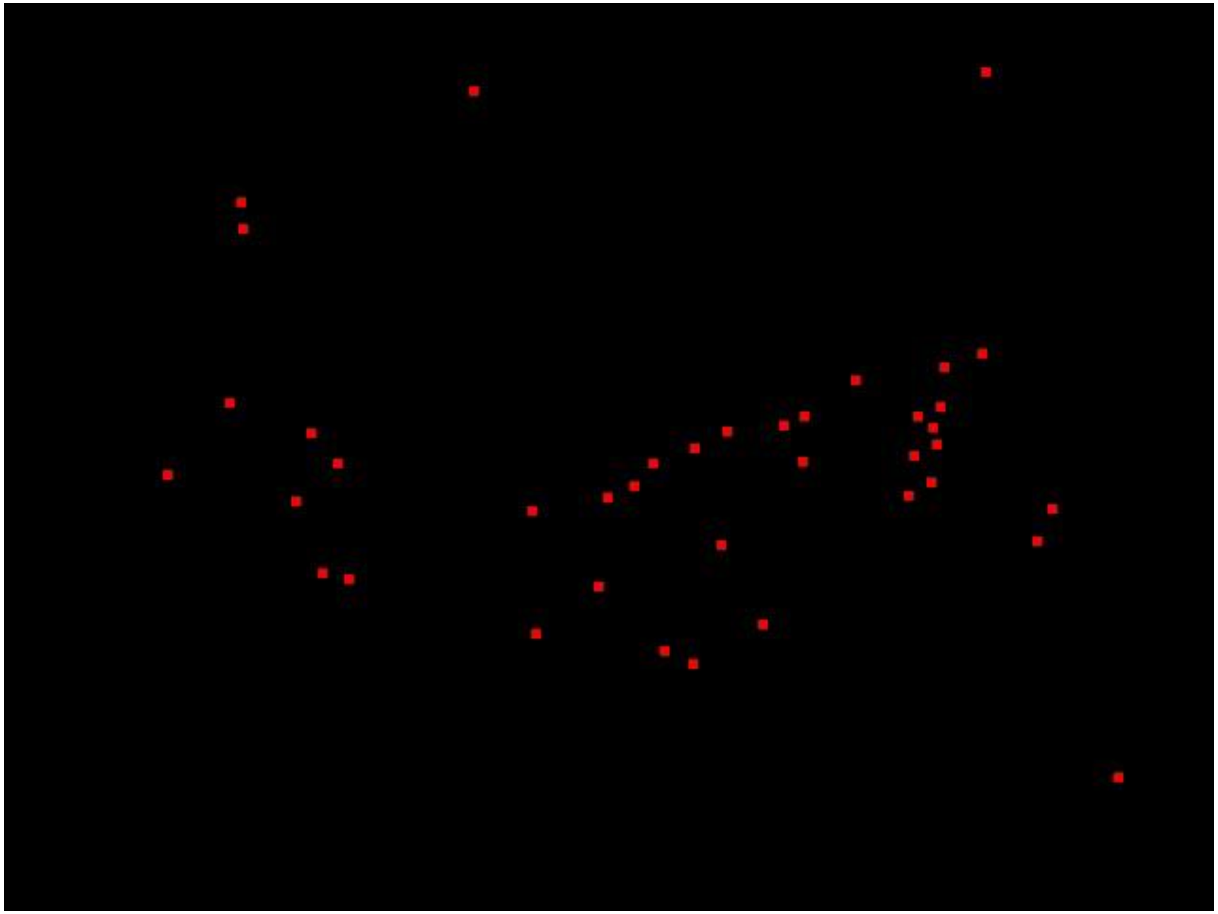
- Projection depth should be obtained
- Set initial value, and iteratively update it

1. Depth=1
2. Factorize
3. Structure and Motion are obtained
4. New projection depth
5. Back to 2 …
6.

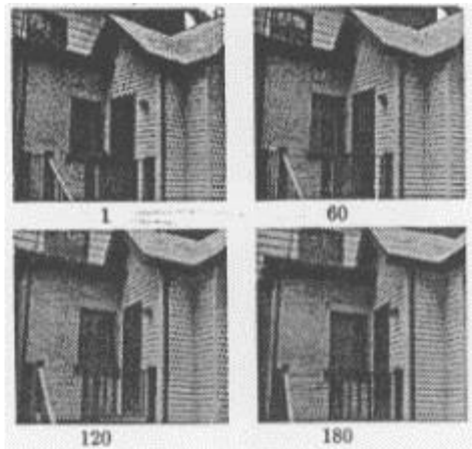
Input Video

Tracking Result

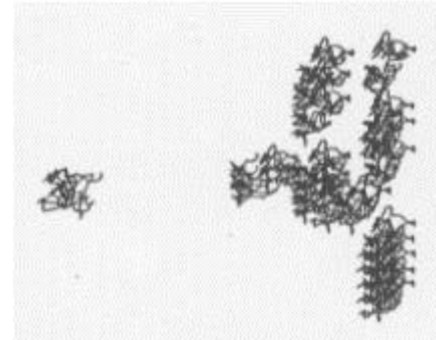
Tracking Result



Example

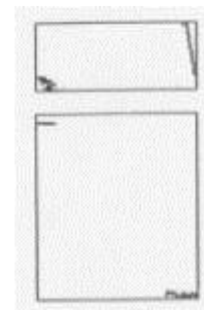


Four out of the 180 frames
of the real house image stream.

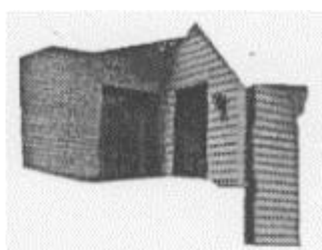


Tracks of 60 randomly selected
features from the real house stream.

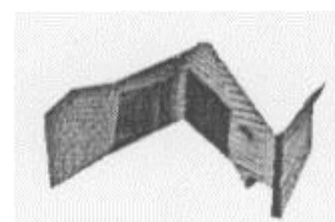
The features selected in the first frame
of the real house stream.



Top and side views of the i_f and j_f vectors
identifying the camera rotation for the real house
stream .



A front view of the three reconstructed walls,
with the original image intensities mapped onto
the resulting surface.



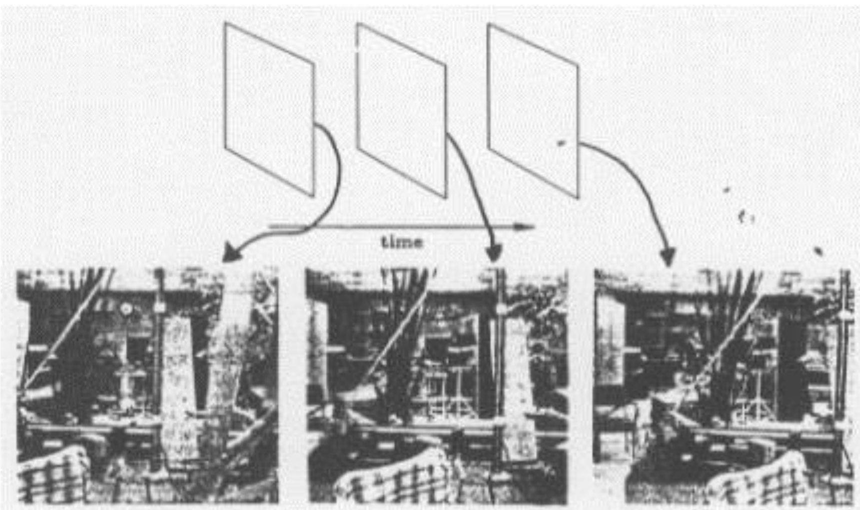
A view from above of the three reconstructed
walls, with image intensities mapped onto the
surface.

Example

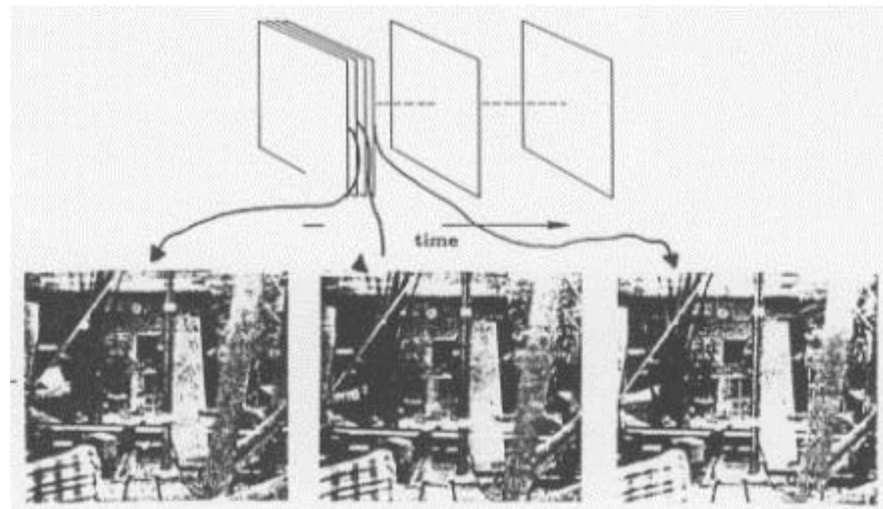
Space-Time Image

東京大学
THE UNIVERSITY OF TOKYO

More Images (moving camera) → Space-Time Image

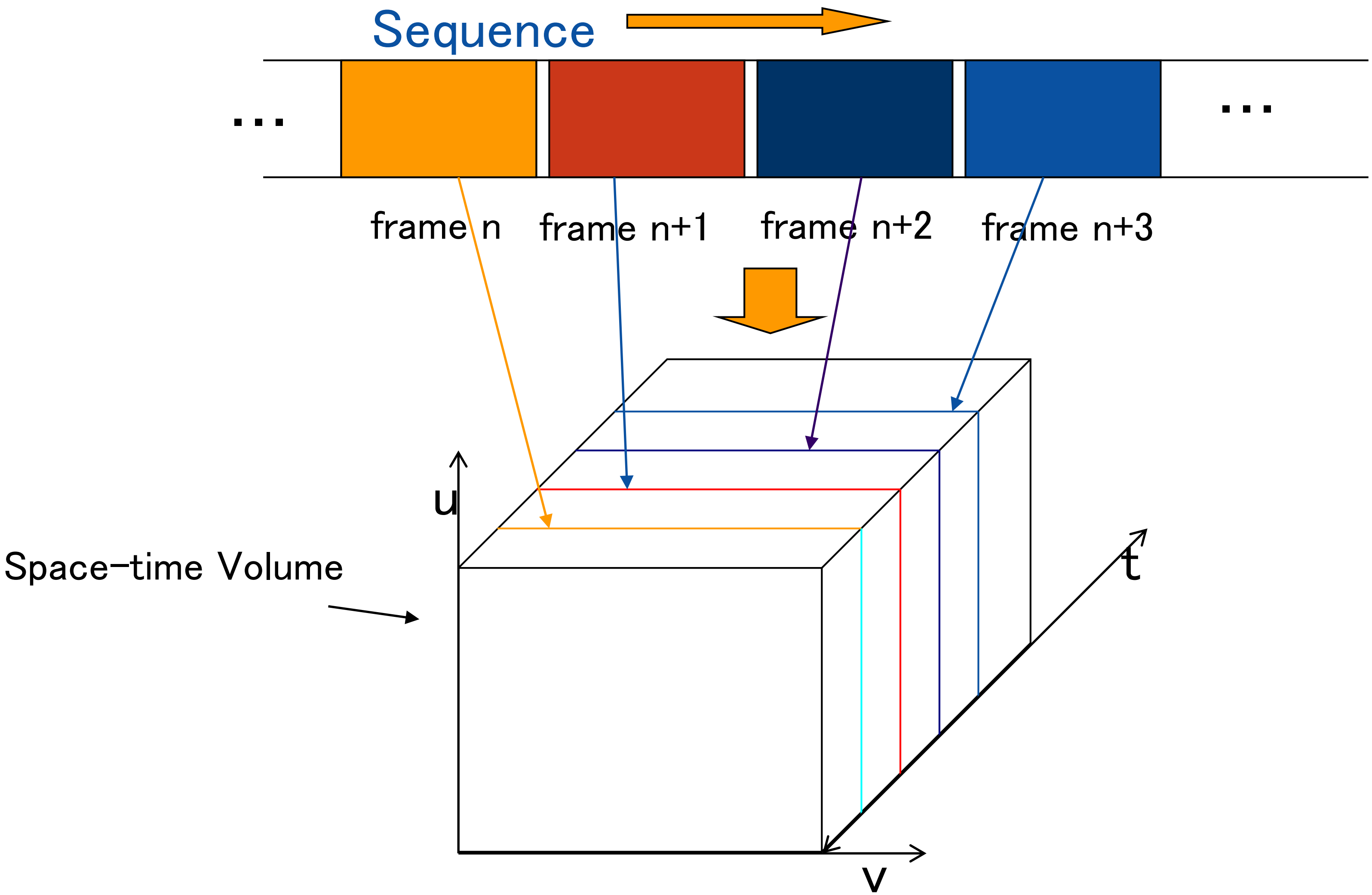


Typical image separation



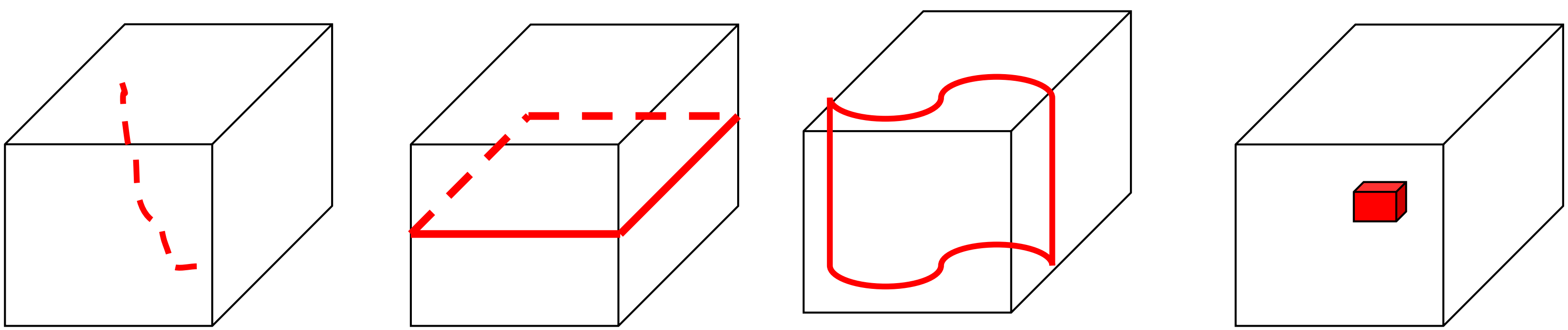
Close sampling image separation

Space-time Volume



Information from Space-Time Volume

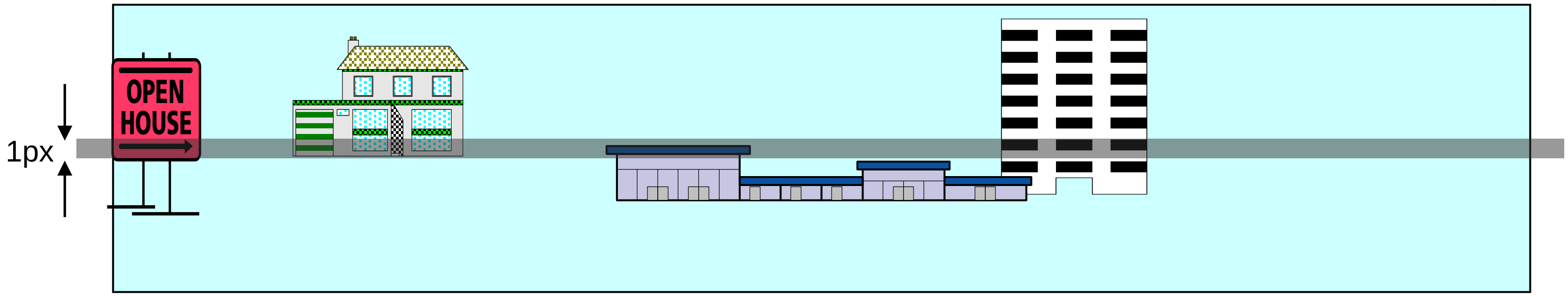
Use partial information

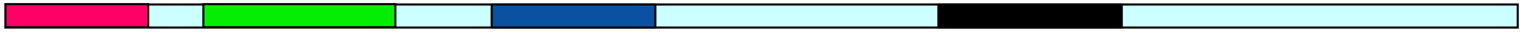


Trajectory Slice 3D-block/shape

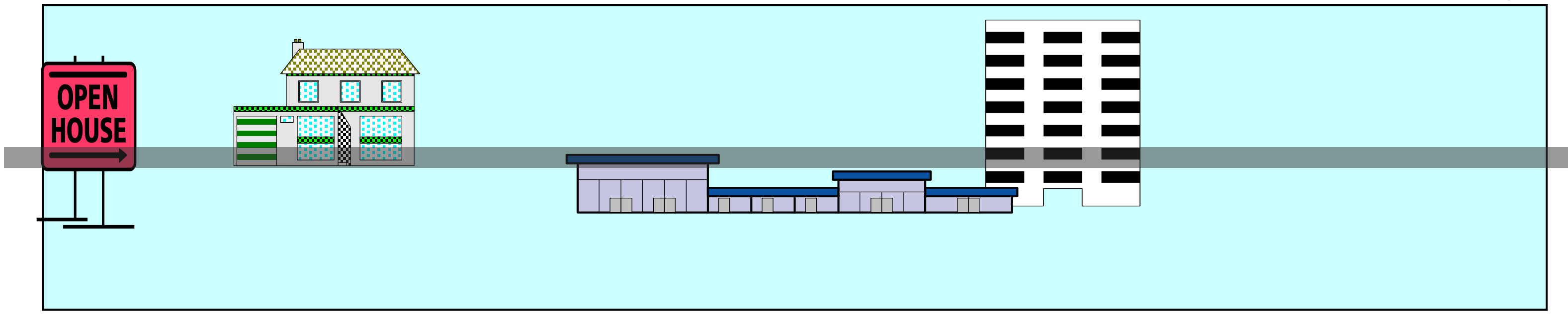
Example:
EPI (Epipolar Plane Image)

Moving camera: Initial position

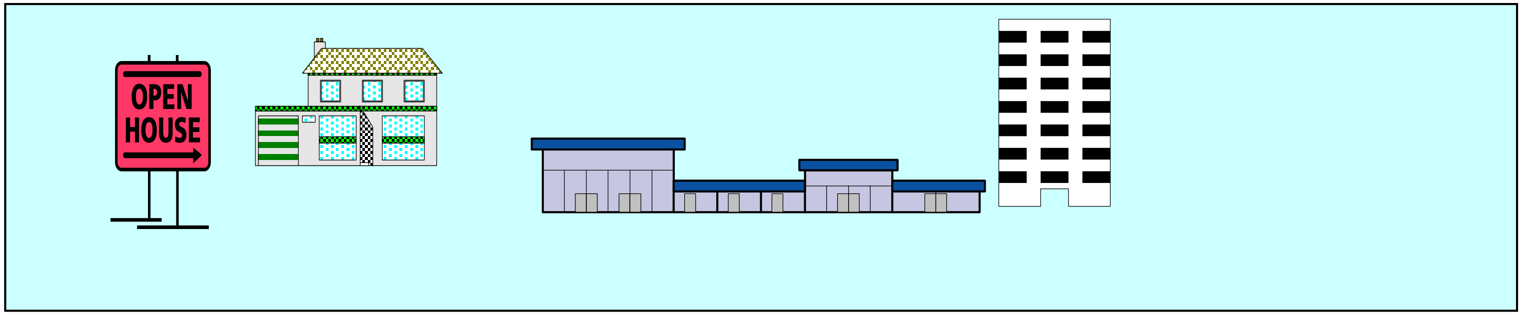


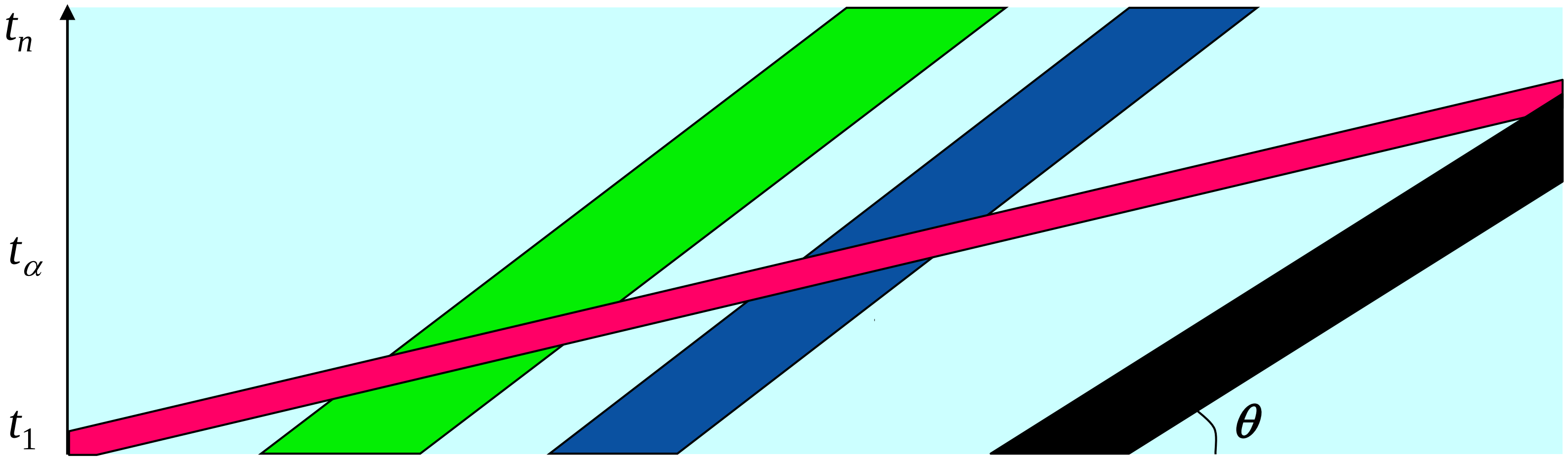


Moving camera: If the camera moves…

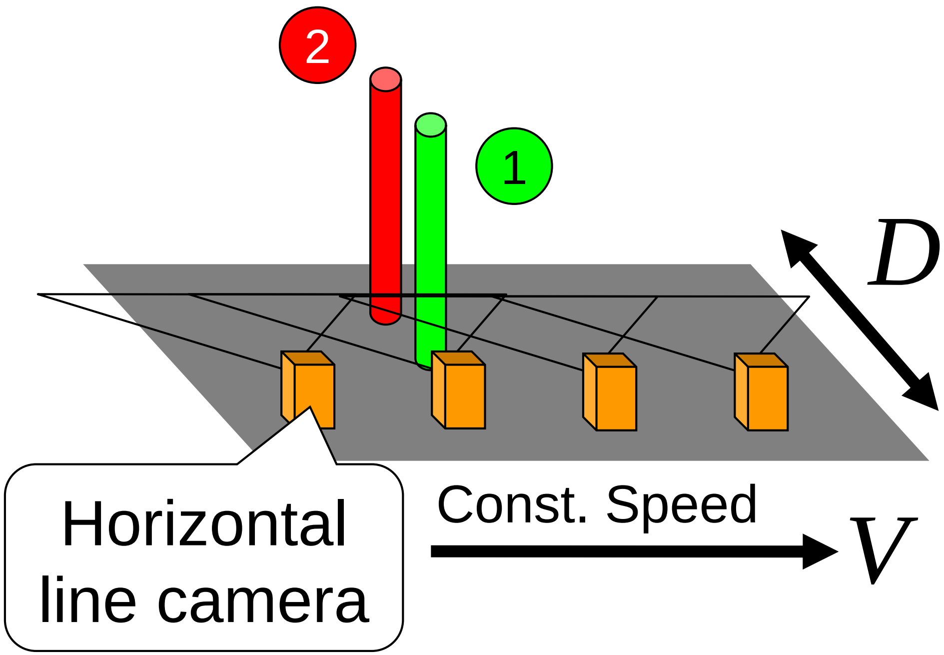


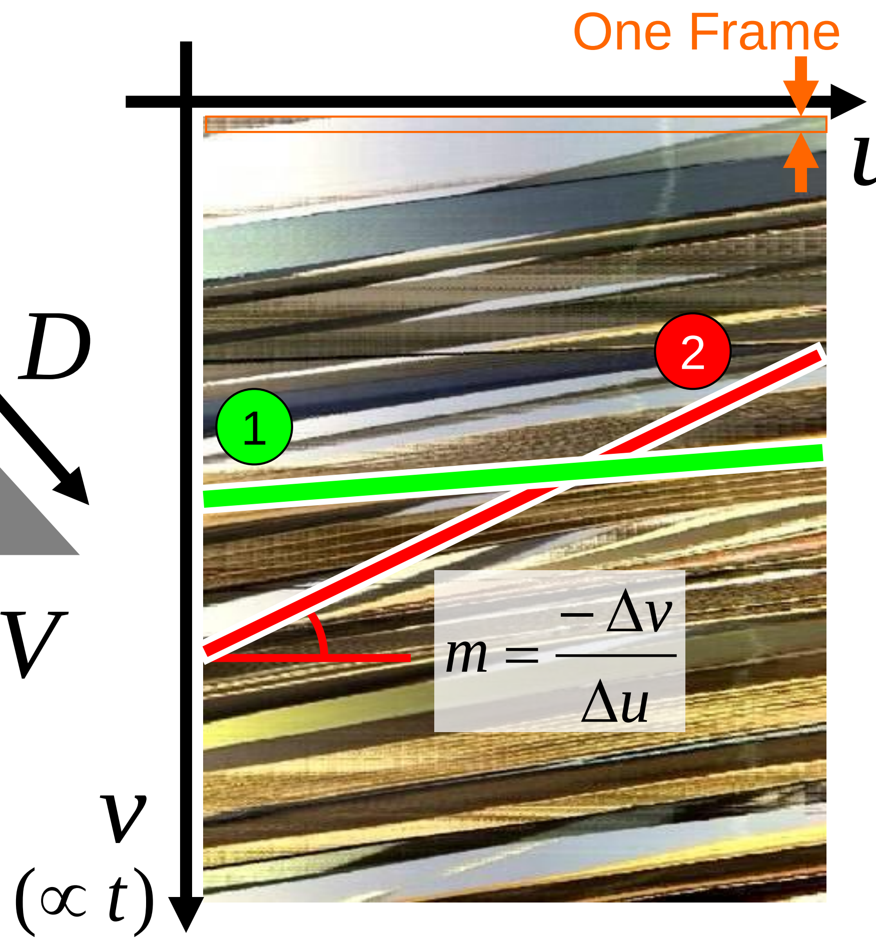
EPI (Epipolar Plane Image)





EPI (Epipolar Plane Image)





$$mV \propto D$$

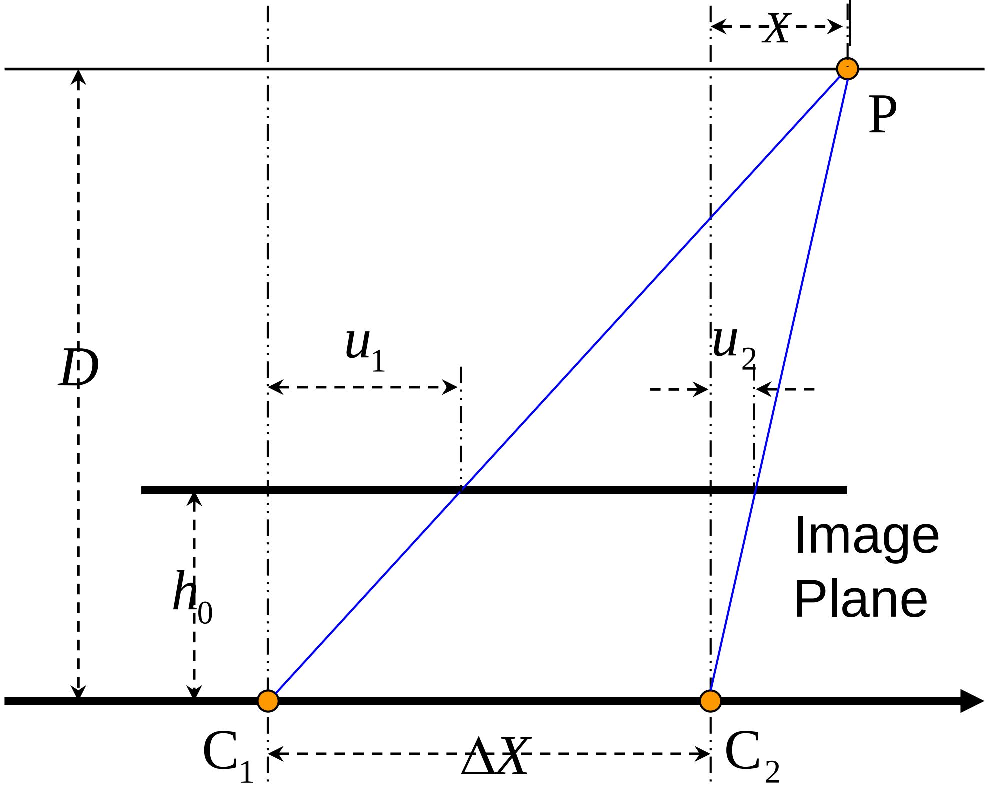
Lateral motion geometry

$$\Delta u = u_2 - u_1$$

$$= \frac{h_0 X}{D} - \frac{h_0(\Delta X + X)}{D}$$

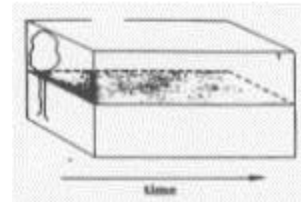
$$= -\frac{h_0}{D}\Delta X$$

$$\Delta v = F_0 \Delta t$$

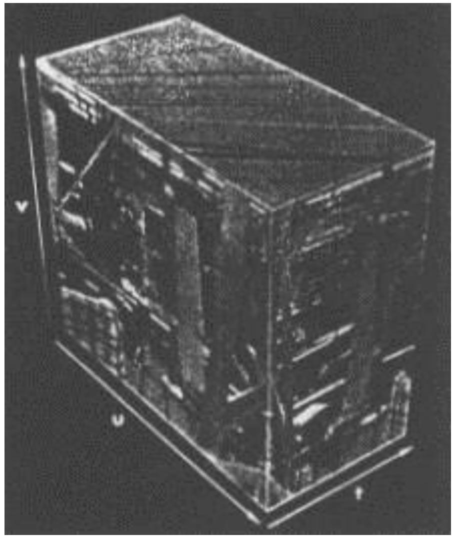


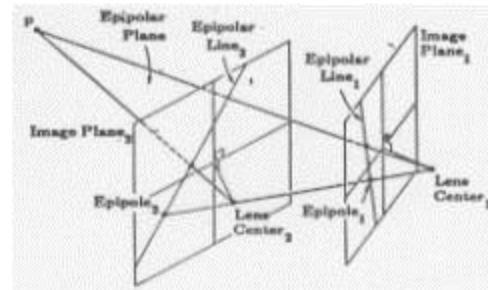
$$m \equiv \frac{-\Delta v}{\Delta u} = \frac{-F_0 \Delta t}{-\frac{h_0}{D}\Delta X} = -\frac{F_0}{h_0} \cdot \frac{D}{V} \propto \frac{D}{V}$$

Same as stereo vision
(Do you remember?)

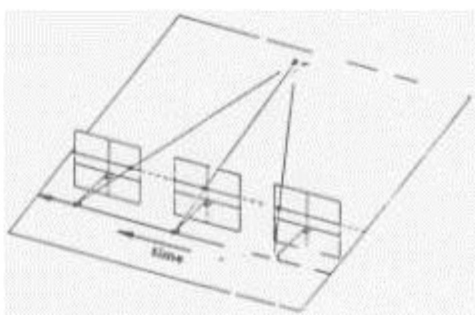


Slice of the solid of data.

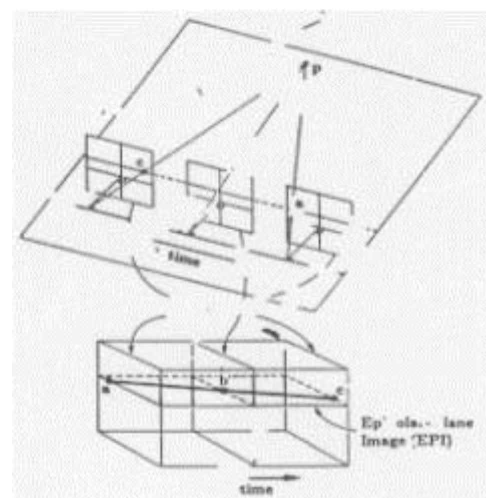

Spatio-temporal solid of data.



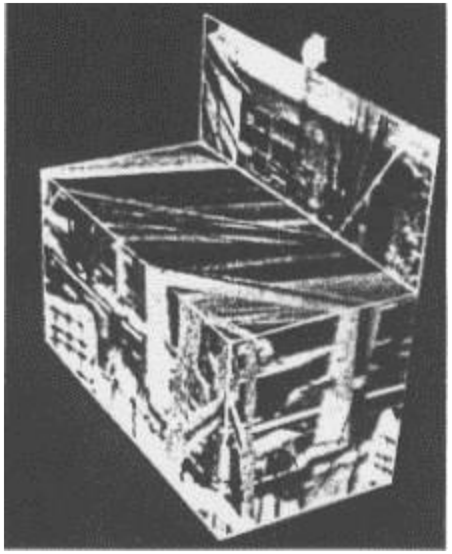
General stereo configuration.



Right-to-left motion.

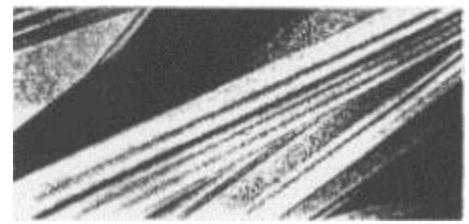


Sliced solid of data.



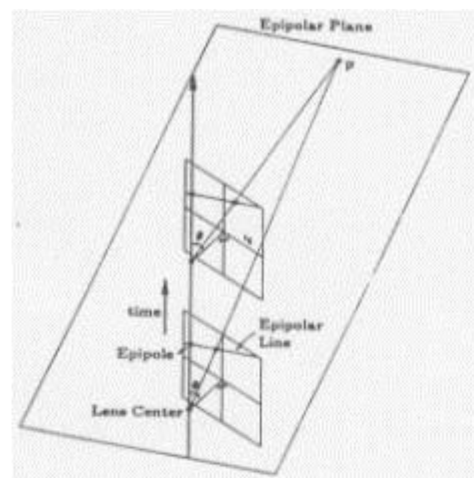
Right-to-left motion with solid.

Frontal view of the EPI.



A second EPI.

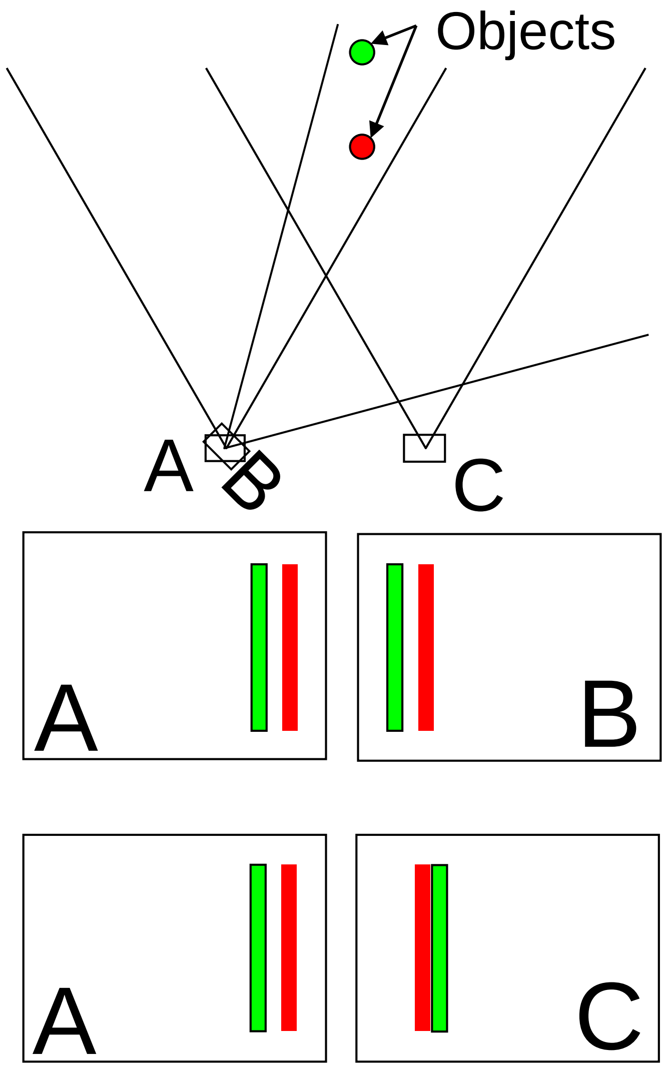
EPI from forward motion.

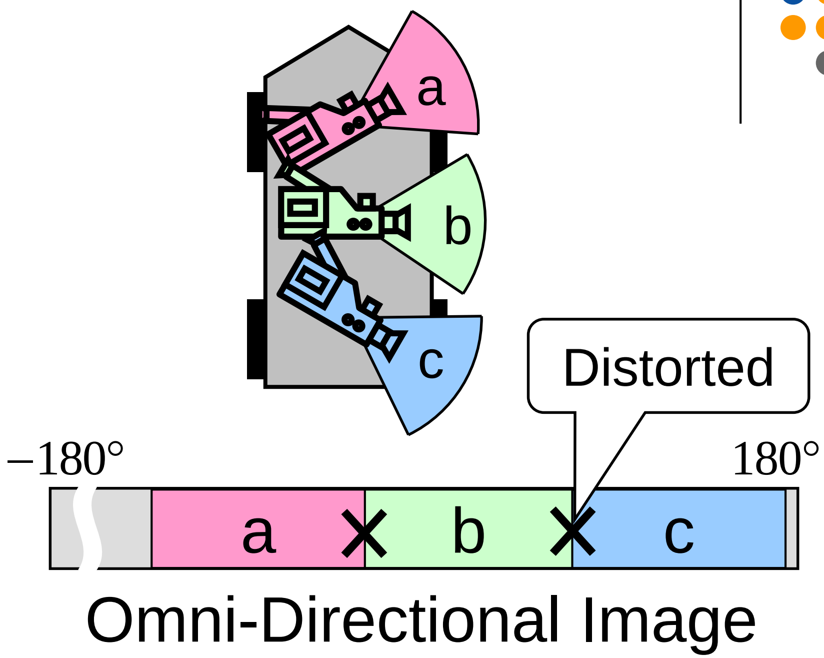


Forward motion.

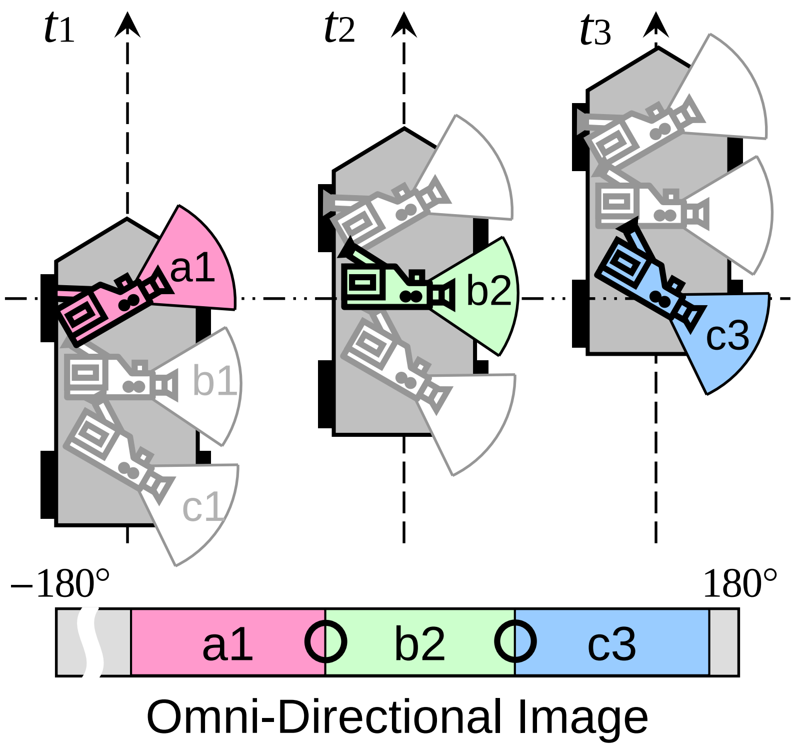
Applications

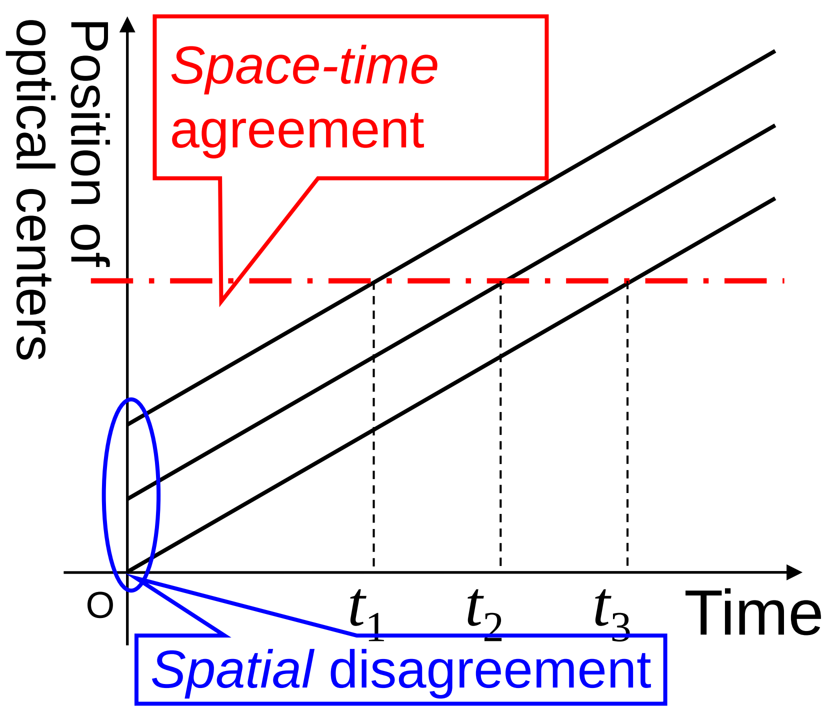
Camera center and distortion





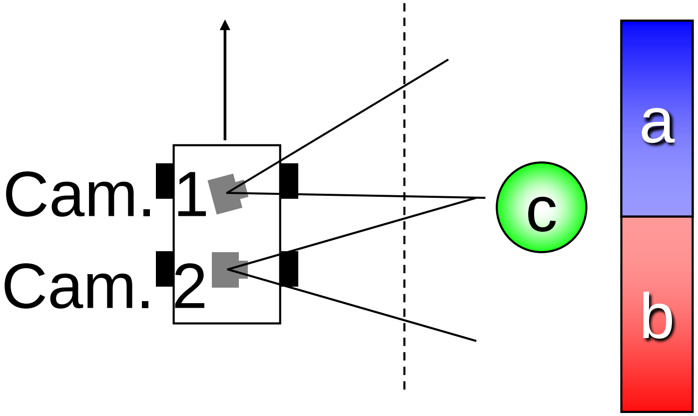
Spatio-temporal coincidence of camera optical center

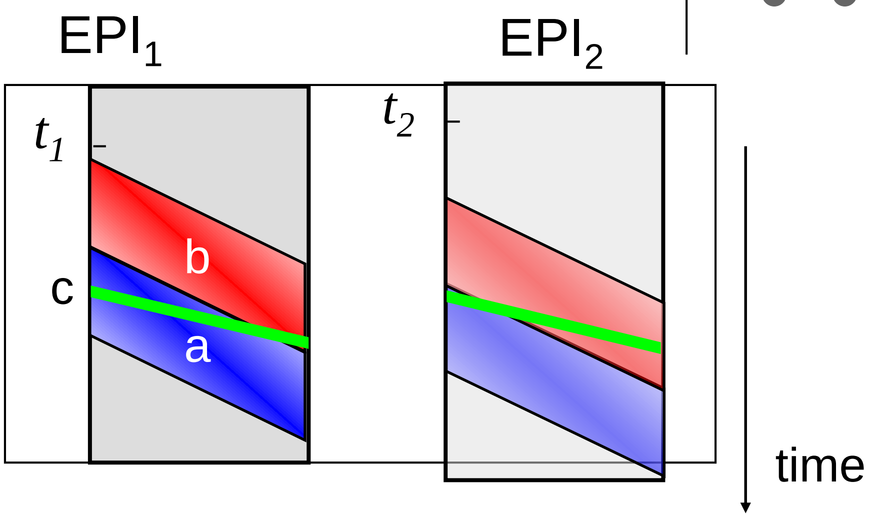


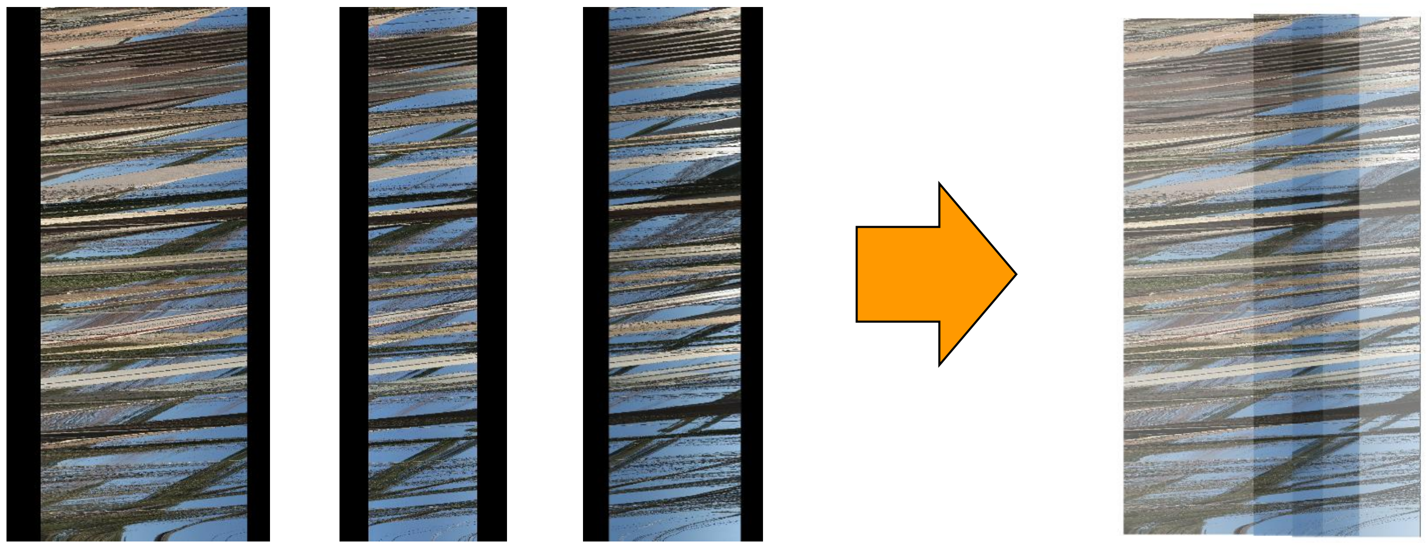


How to know t2, t3?

Temporal adjustment using EPI (Software-based camera sync.)



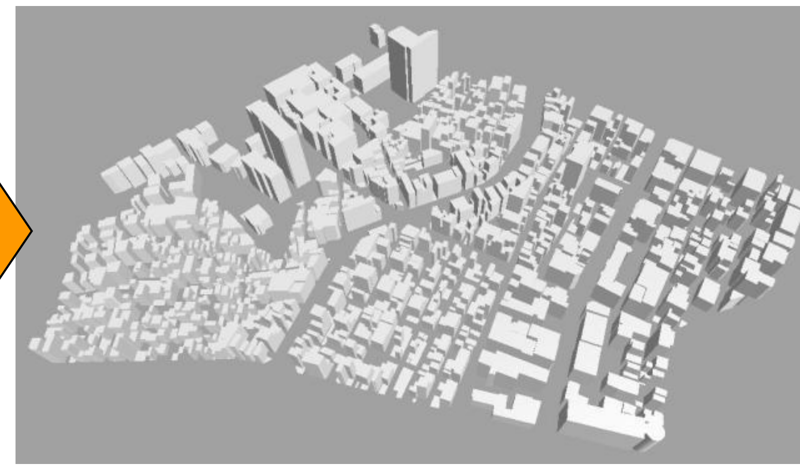




Result

Spacetime Feature Matching for Texturing (Wang 2008)

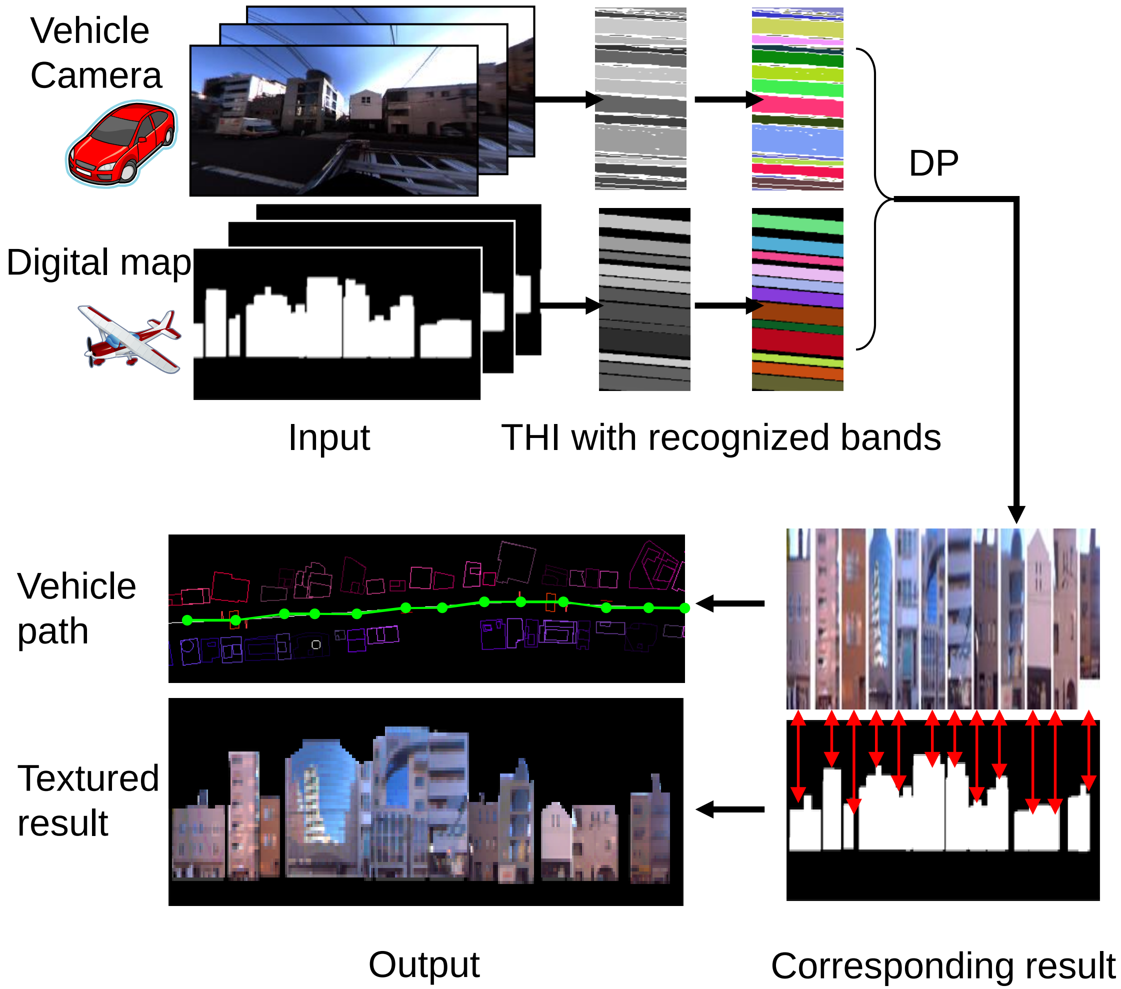
Ground-view image
(Vehicle survey, Local)



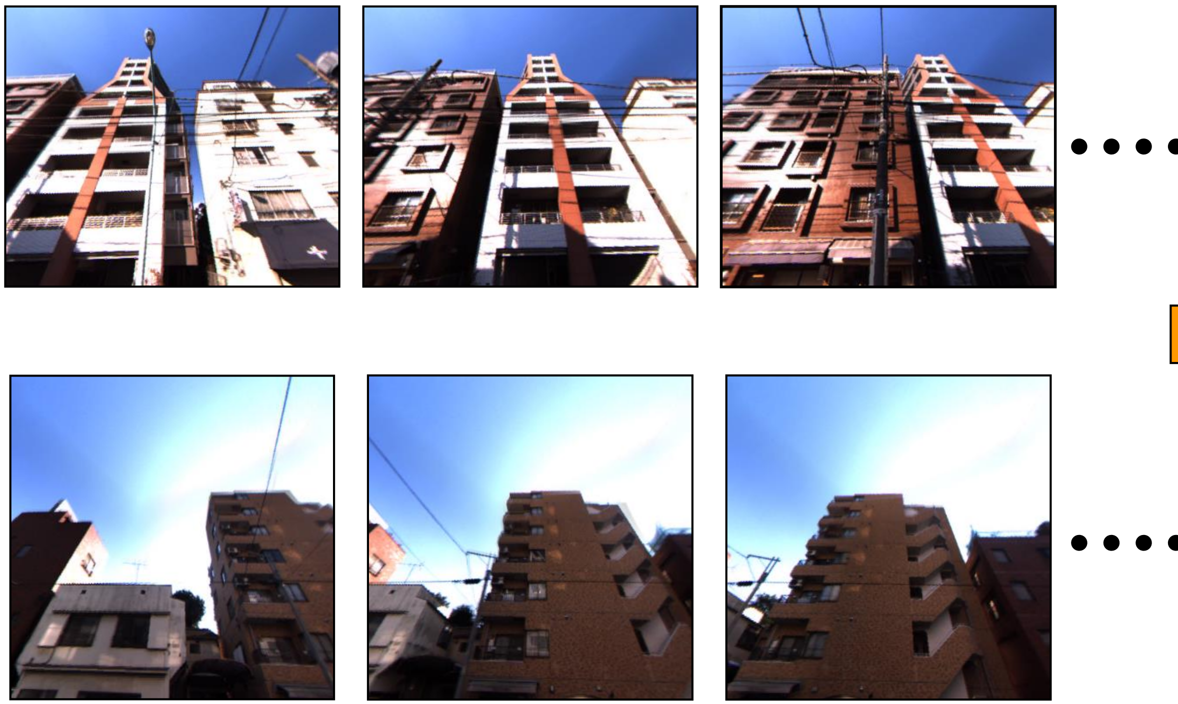
3D residential map
(Aerial survey, Global)

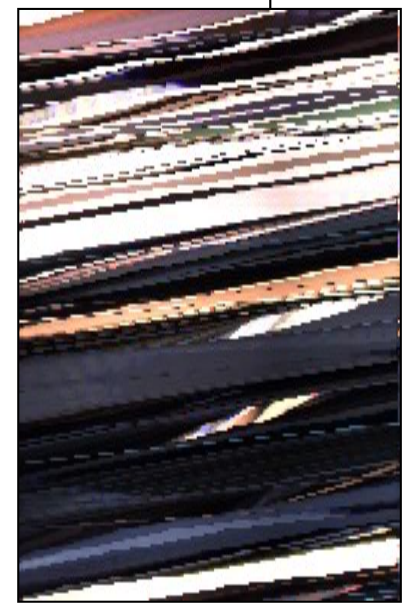
How can we get correspondence, and add a texture onto building walls?

Spacetime Feature Matching for Texturing (Wang 2008)



Problems in using EPI

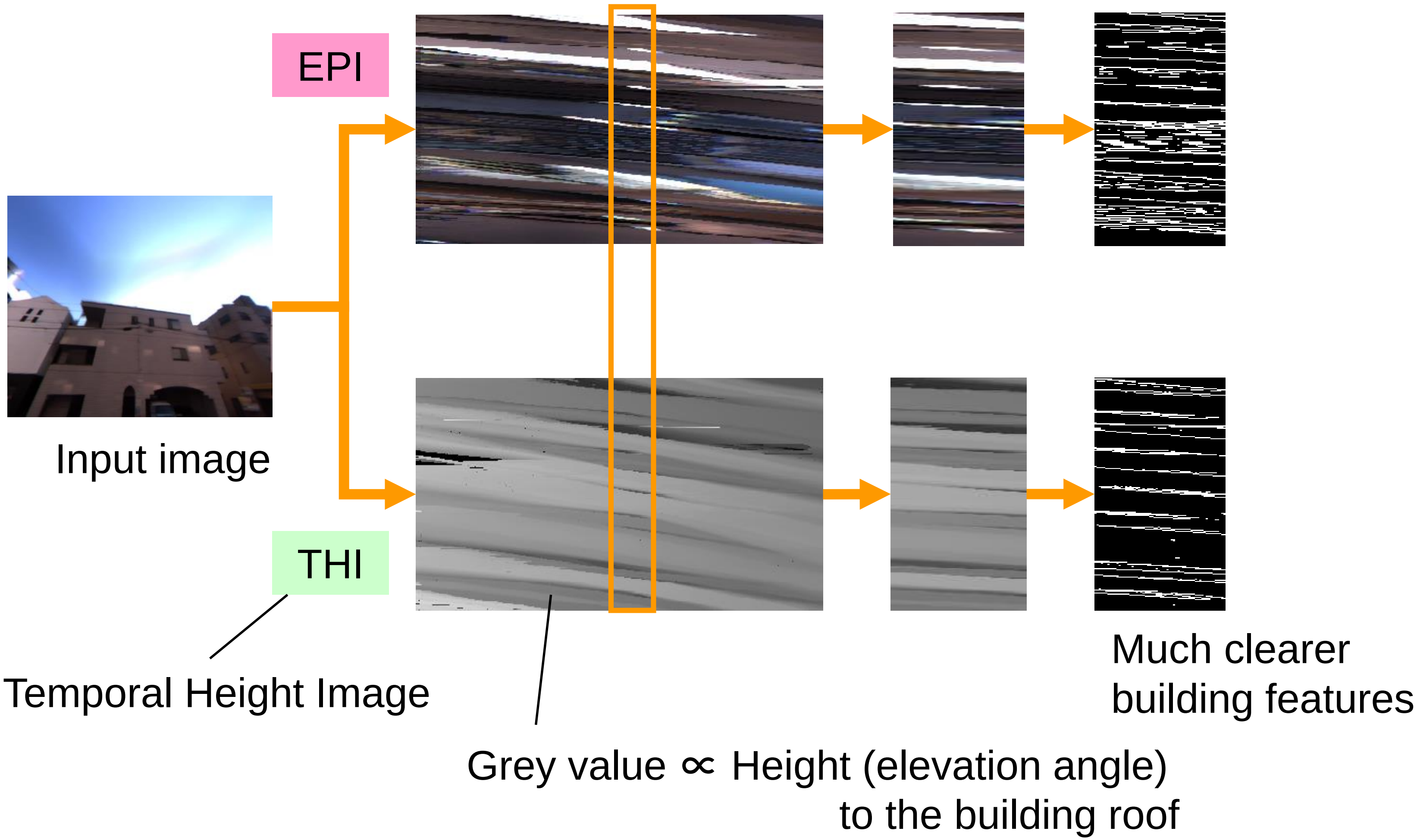




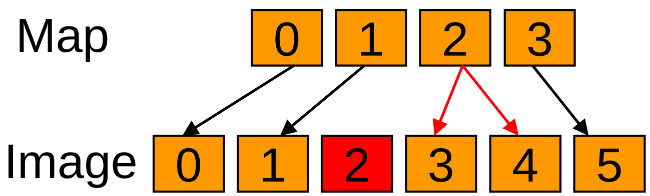
Real example of EPI

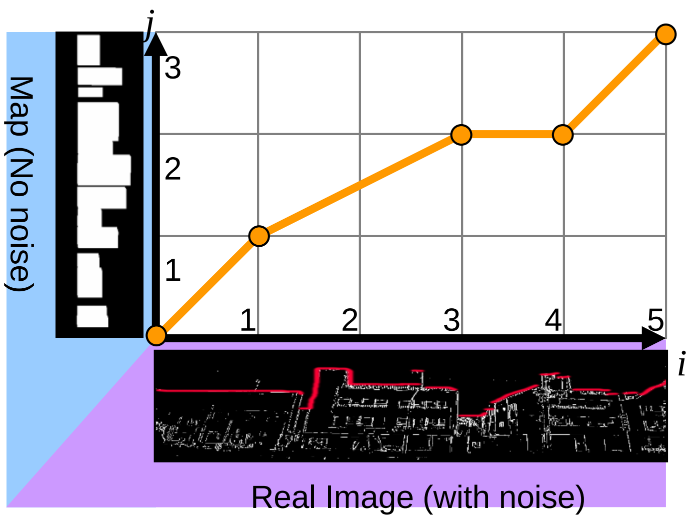
Textures inside building (windows, etc.) disturb to recognize the building features stably

Using Structural Information Instead of Color Information

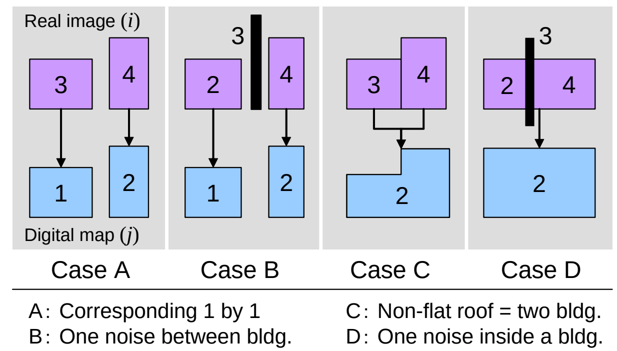


Building Matching between Map and Image using THI





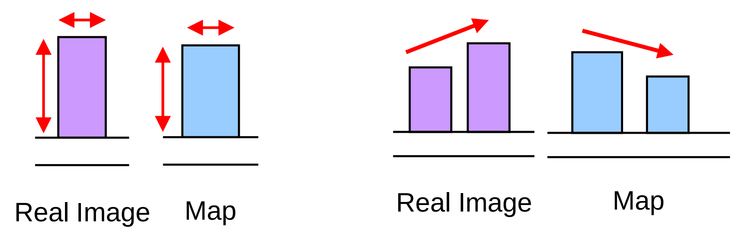
Matching Pattern



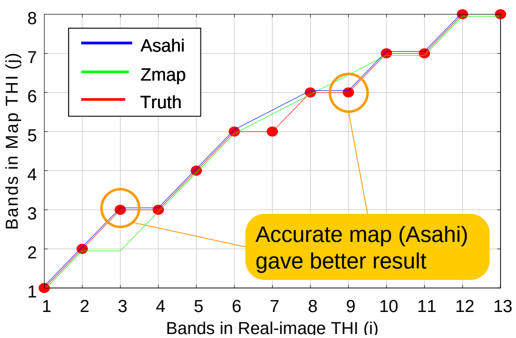
Matching Cost

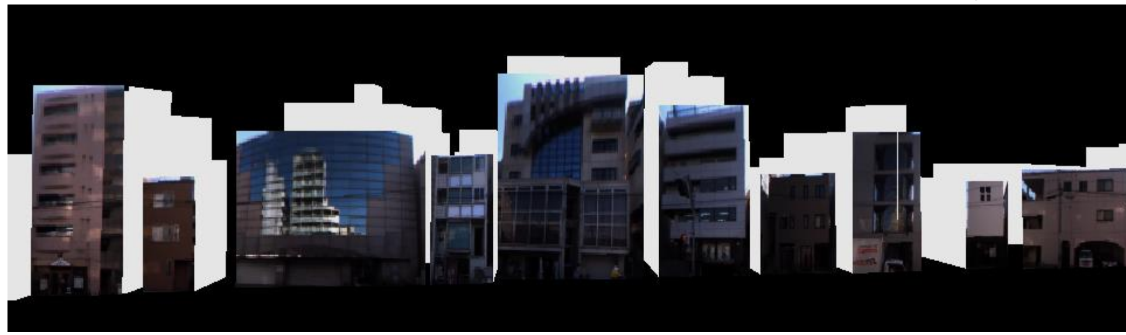
Aspect similarity

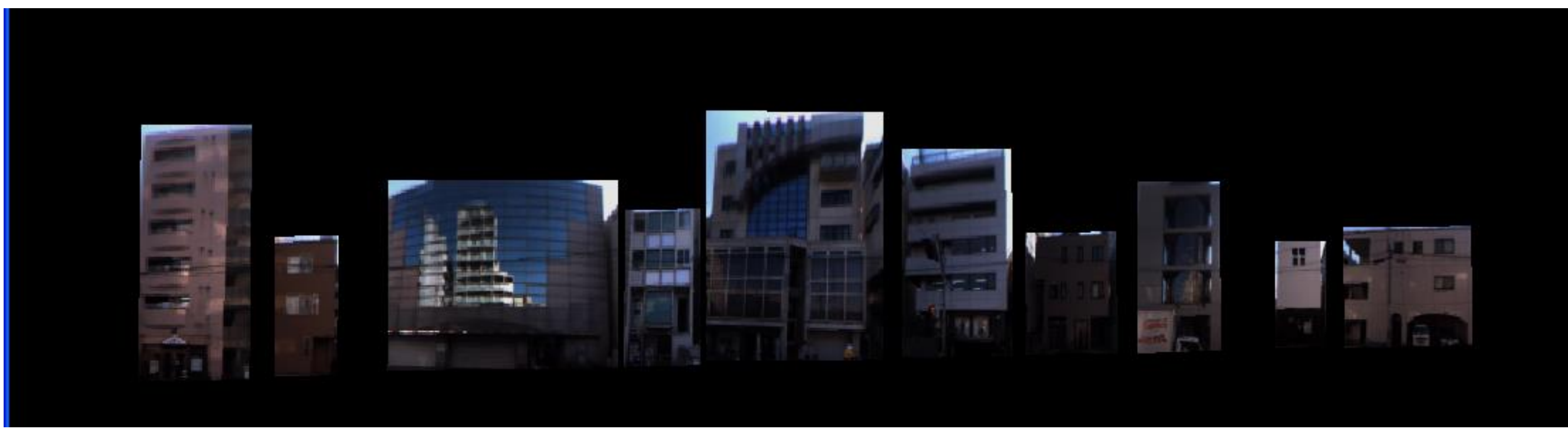
Height-transition similarity



Matching and Texturing Result

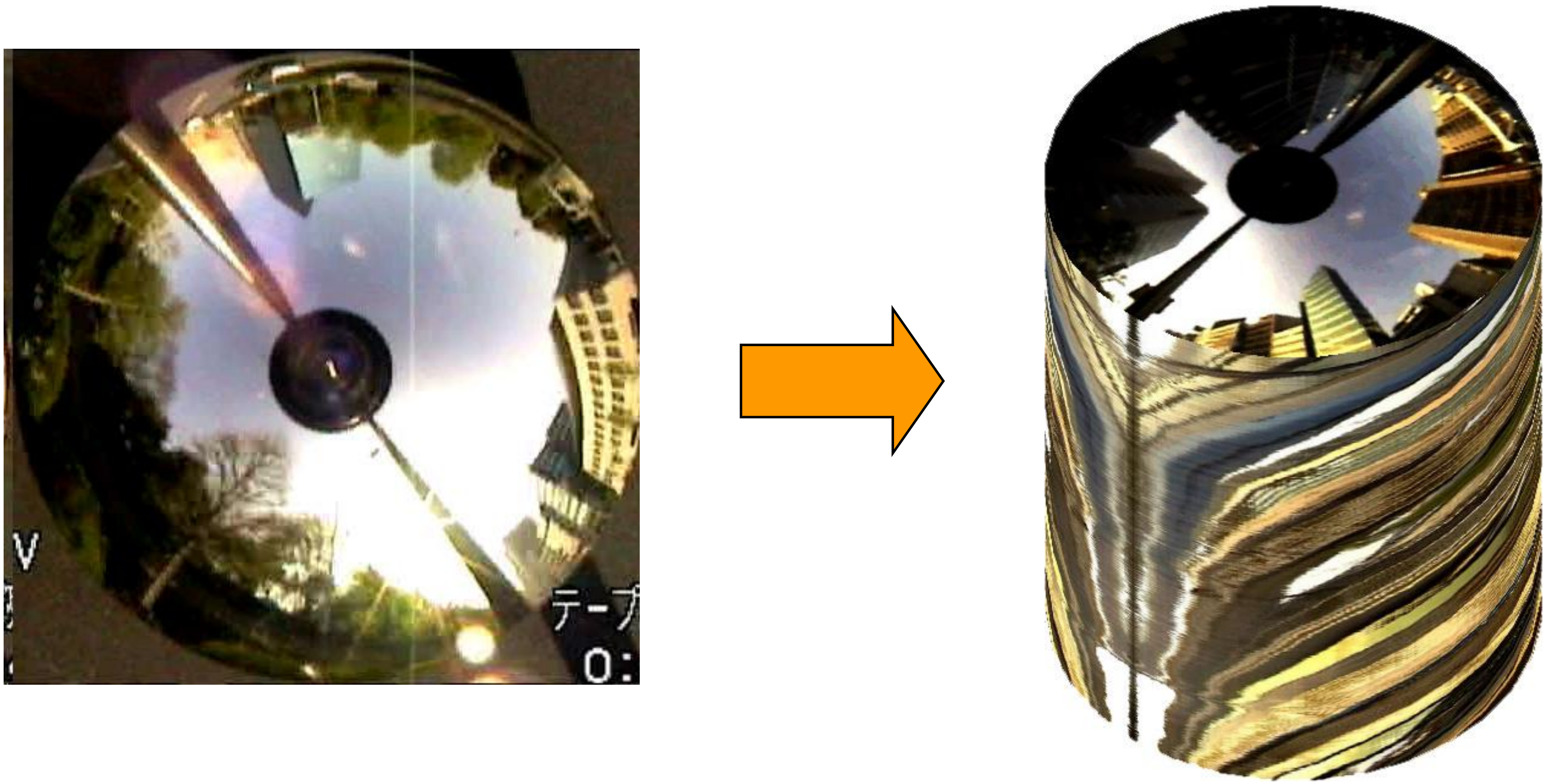




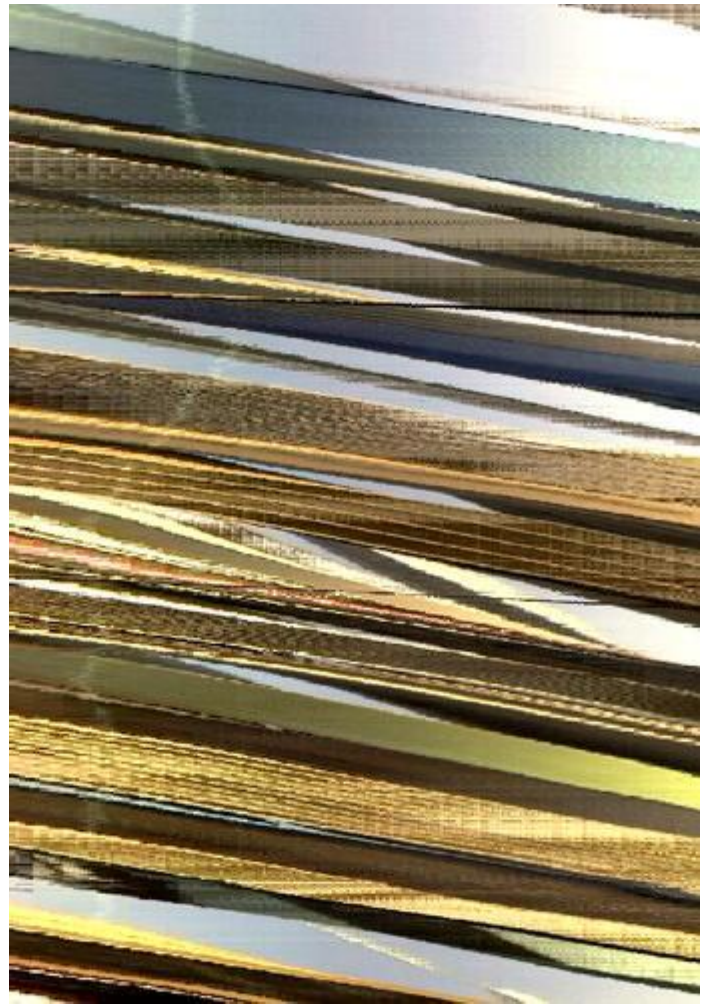


Omnidirectional Camera

Spatio-temporal volume of omni-directional image



Cross-section
(an elliptic curve)

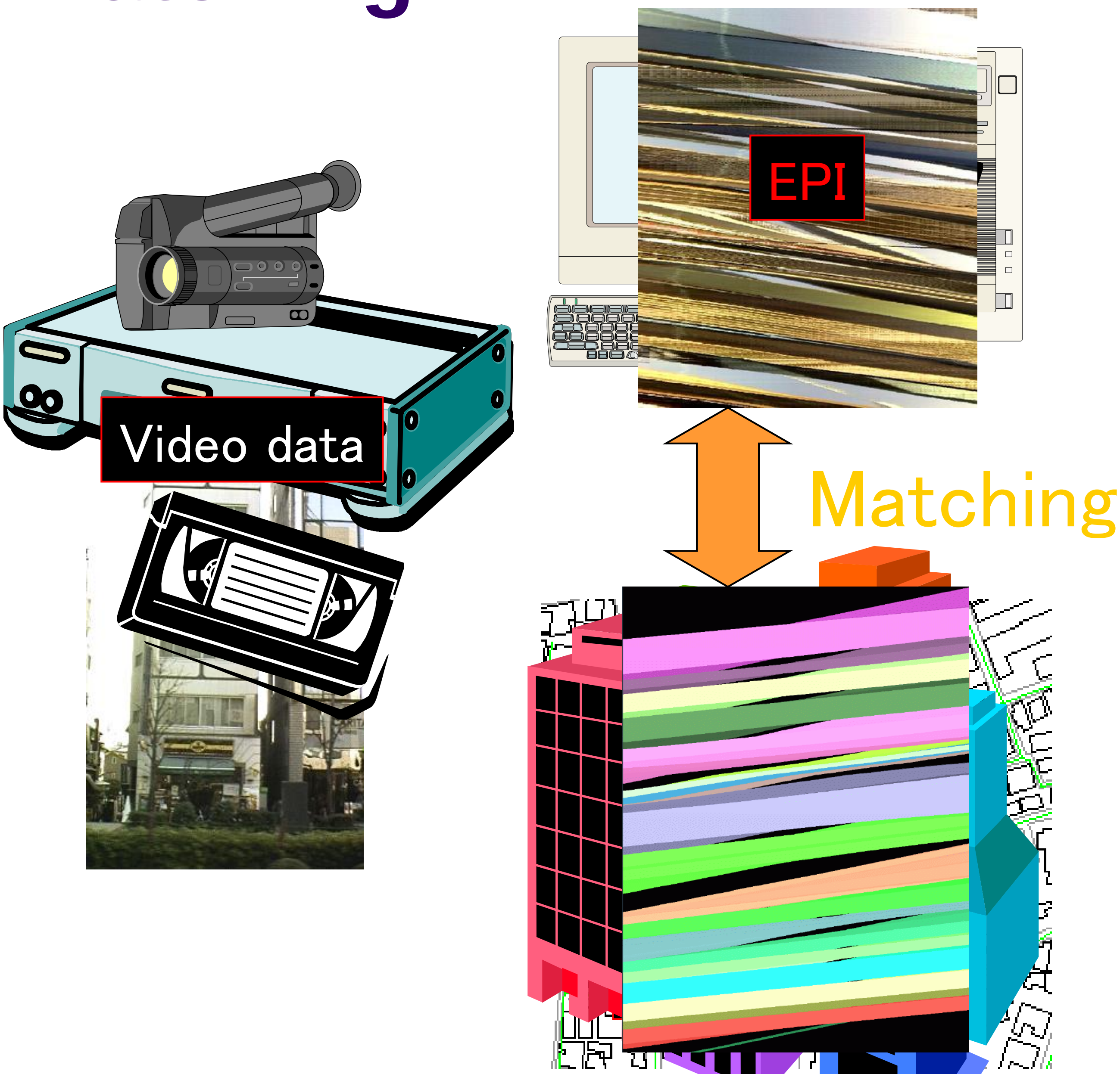


Depth Info

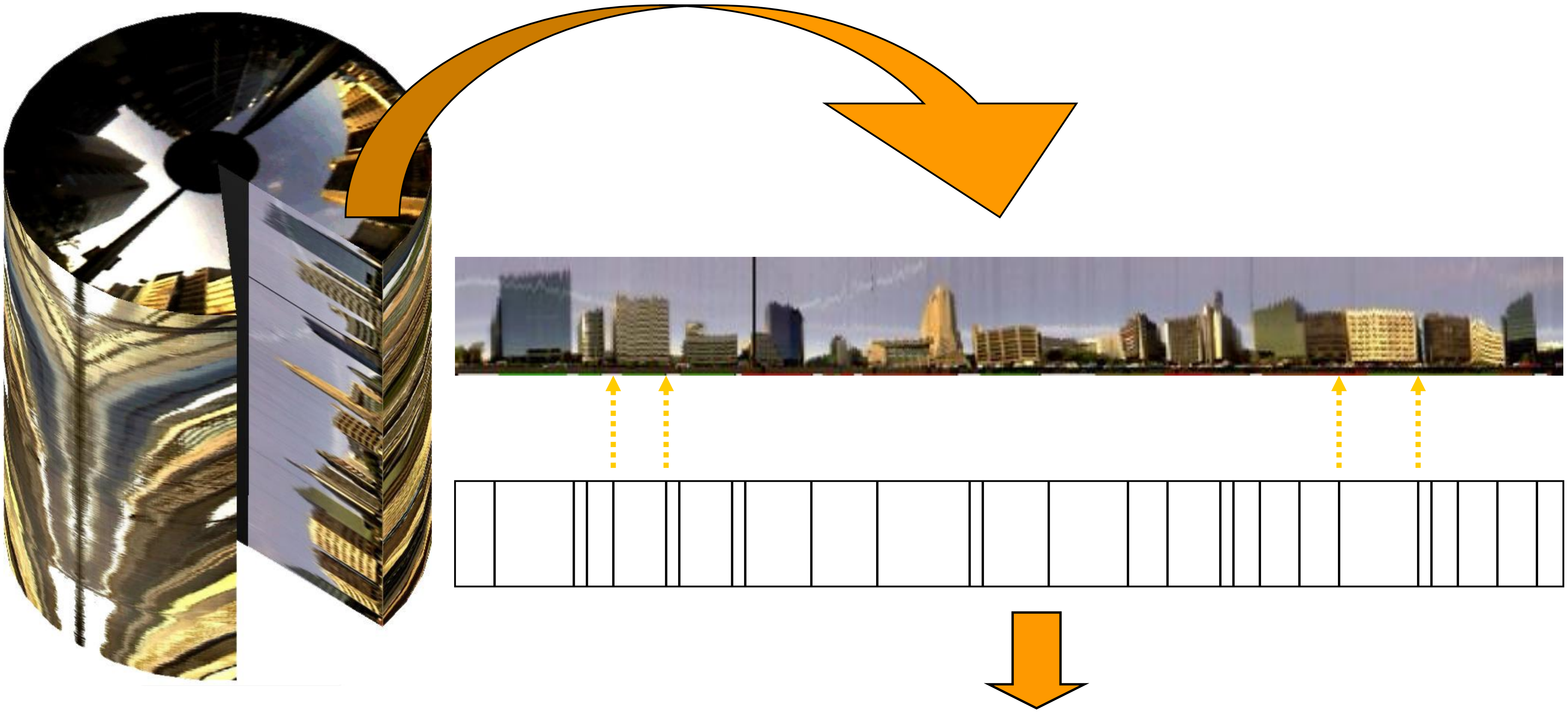
Digital residential map

Correspondence between map and image

EPI Matching



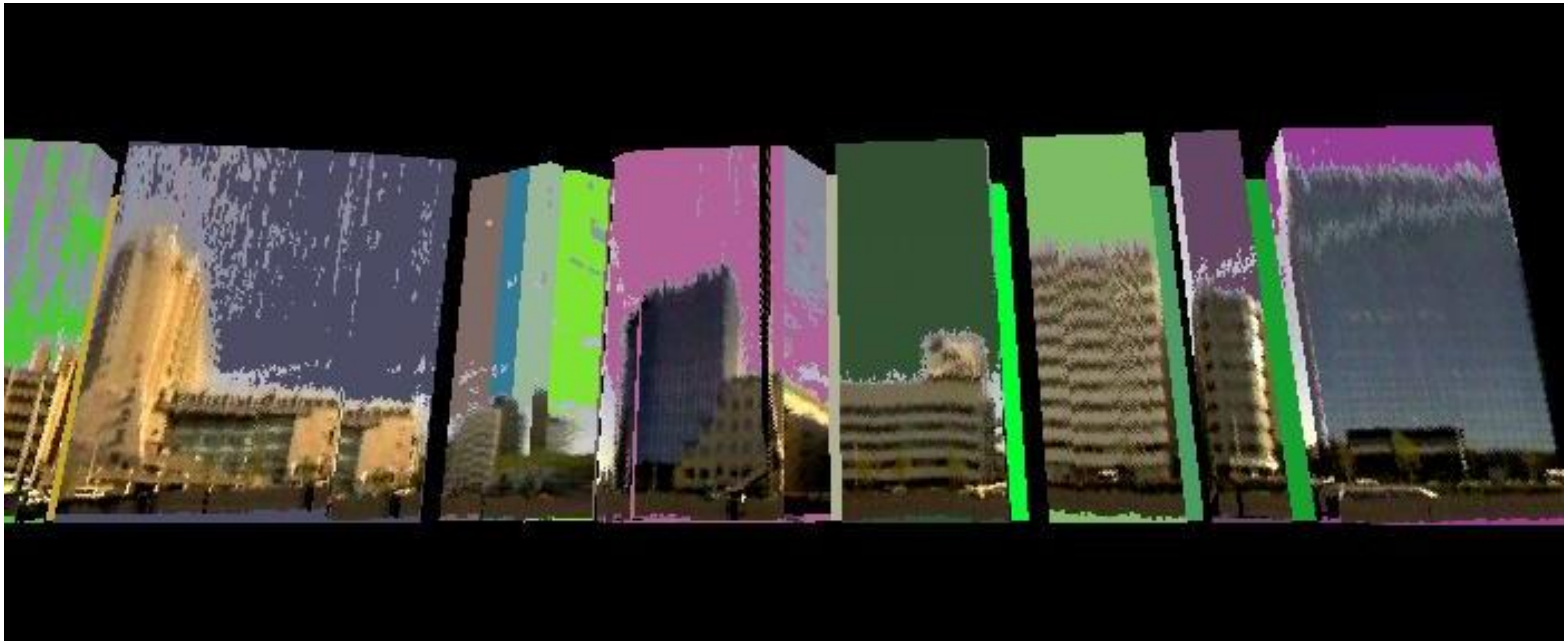
Cross-section (a radius line)



Panorama image

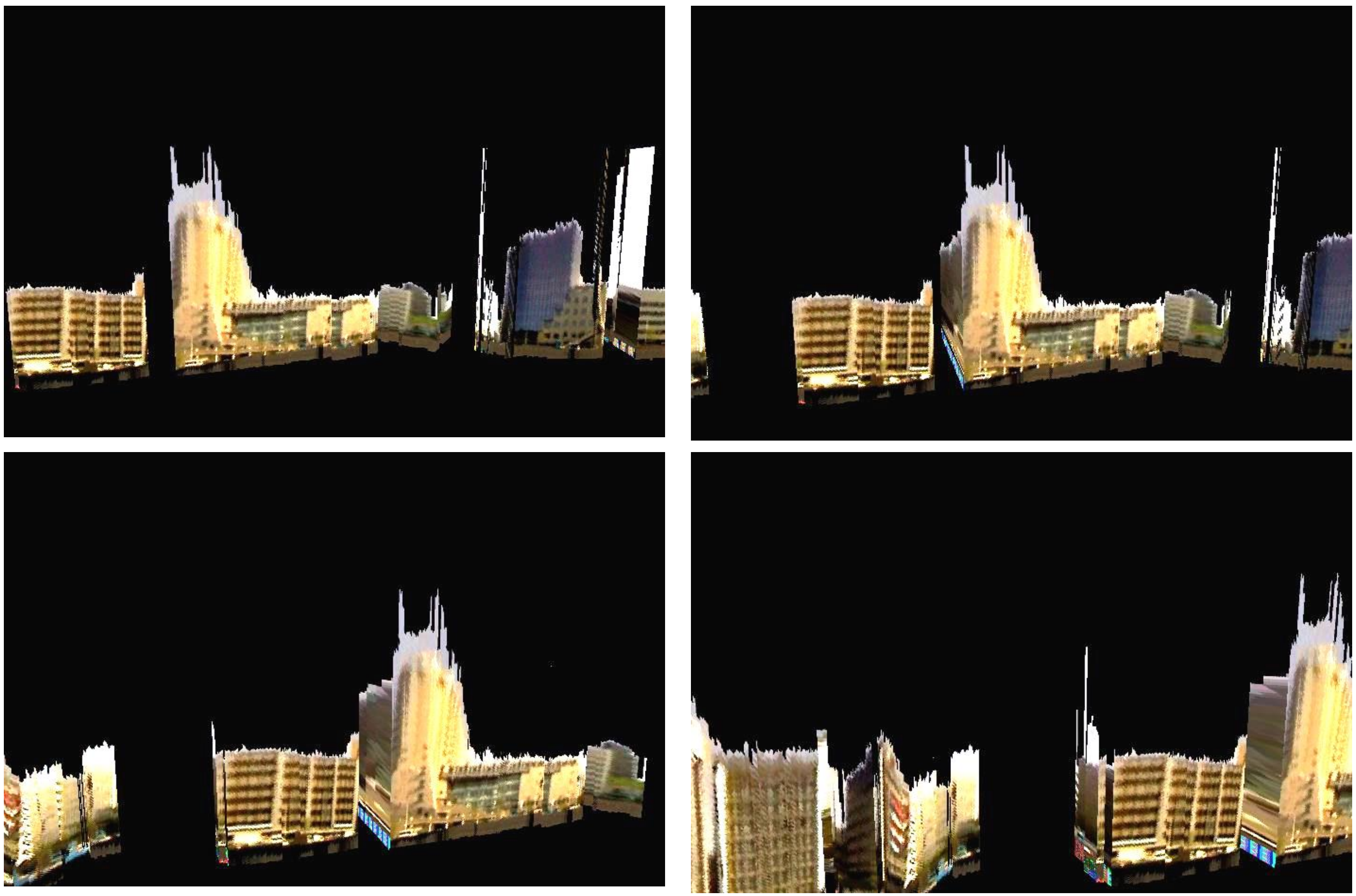
Texture Mapping

- Height info and texture



Texture Mapping

- Side faces



Summary

- Introduction
- Basic technologies
  - Background subtraction
  - Optical flow
  - Structure from Motion (SfM)
  - Space-time Image Analysis