Applications of Image Motion Estimation I

Mosaicing

Princeton University
COS 429 Lecture

Oct. 23, 2007

Harpreet S. Sawhney
hsawhney@sarnoff.com

vision technologies LAB

SARNOFF Corporation

Visual Motion Estimation : Recapitulation

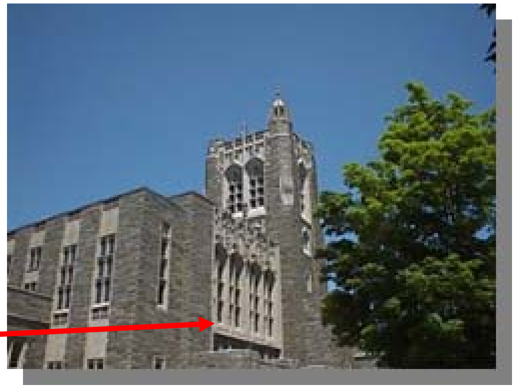
Plan

- Explain optical flow equations
- Show inclusion of multiple constraints for solution
- Another way to solve is to use global parametric models

Brightness Constancy Assumption

$\mathbf{I_1(p')}$

$\mathbf{p - u(p)}$



$\mathbf{I_2(p)}$

$\mathbf{p}$

Model image transformation as :

$$I_2(p) = I_1(p - u(p)) = I_1(p')$$

Brightness Constancy

Reference Coordinate

Optical Flow

Corresponding Coordinate

How do we solve for the flow ?

$$I_2(p) = I_1(p - u(p)) = I_1(p')$$

Use Taylor Series Expansion

$$I_2(p) = I_1(p) - \nabla I_1^T u(p) + O(2)$$

Image Gradient

Convert constraint into an objective function

$$E_{SSD}(u) = \sum_{p \in R} (\nabla I_1^T u(p) + \delta I(p))^2$$

$$I_2(p) - I_1(p)$$

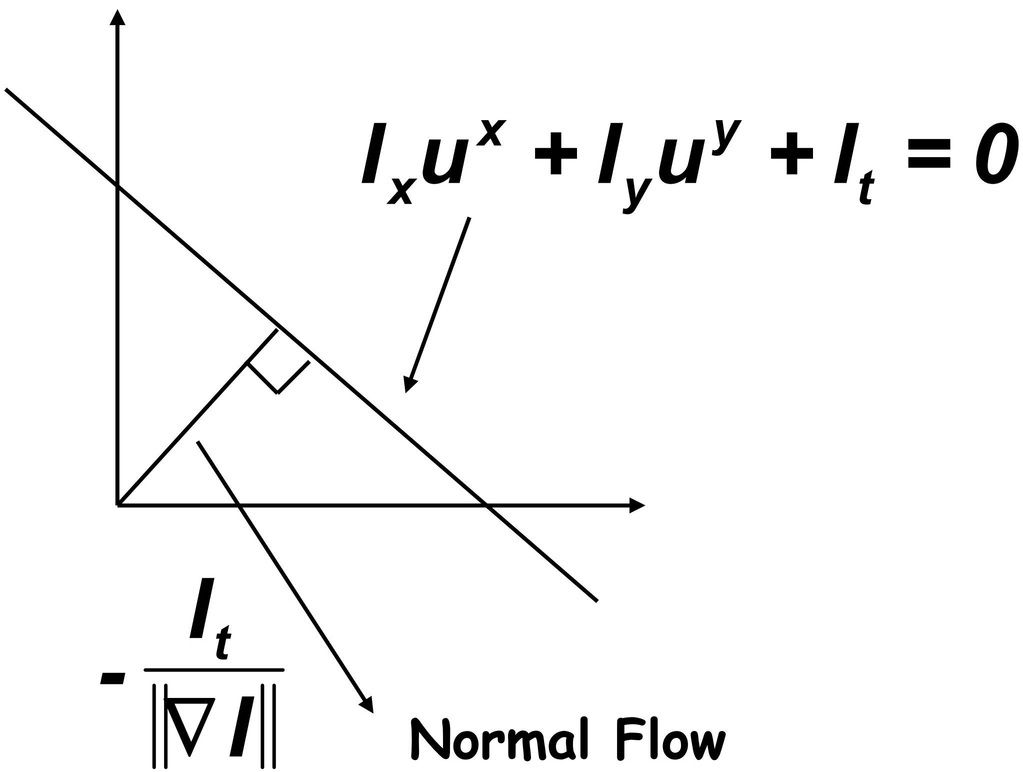
Optical Flow Constraint Equation

At a Single Pixel

$$I_2(p) = I_1(p) - \nabla I_1^T u(p) + O(2)$$

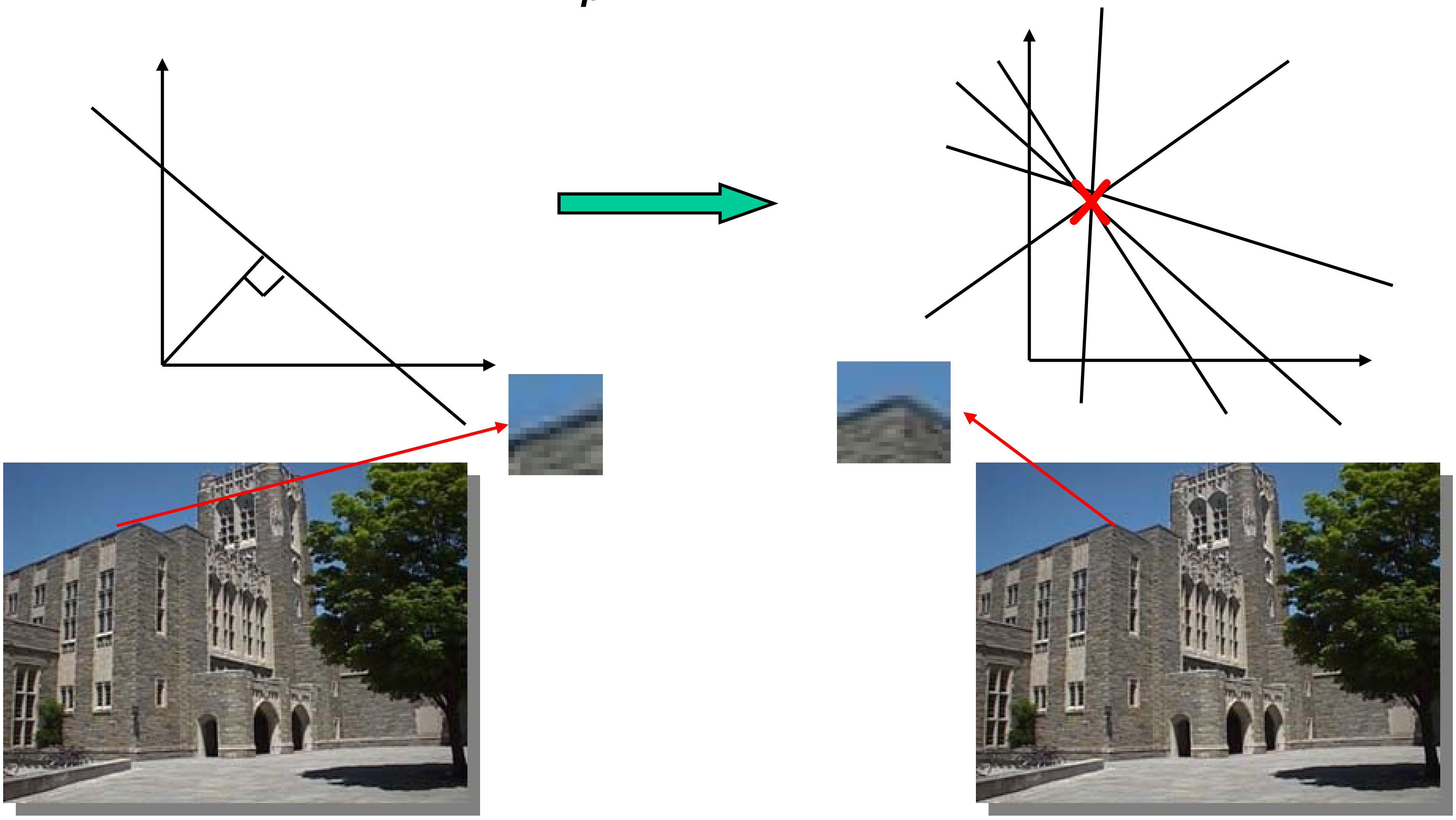
Leads to

$$\nabla I_1^T u(p) + \delta I(p) \approx 0$$



Multiple Constraints in a Region

$$E_{SSD}(u) = \sum_{p \in R} (\nabla I_1^T u(p) + \delta I(p))^2$$



Solution

$$E_{SSD}(u) = \sum_{p \in R} (\nabla I_1^T u(p) + \delta I(p))^2$$

$$\frac{\partial E_{SSD}(u)}{u} = 0$$

$$\sum_{p \in R} \nabla I (\nabla I_1^T u(p) + \delta I(p)) = 0$$

$$[\sum_{p \in R} \nabla I \nabla I_1^T] u = \sum_{p \in R} -\nabla I \delta I$$

$$Au = b$$

Solution

$$Au = b$$

$$A = \begin{bmatrix} \sum_R I_x^2 & \sum_R I_x I_y \\ \sum_R I_x I_y & \sum_R I_y^2 \end{bmatrix} \qquad b = \begin{bmatrix} \sum_R -I_x \delta I \\ \sum_R -I_y \delta I \end{bmatrix}$$

Observations:

- A is a sum of outer products of the gradient vector
- A is positive semi-definite
- A is non-singular if two or more linearly independent gradients are available
- Singular value decomposition of A can be used to compute a solution for u

Another way to provide unique solution

Global Parametric Models

$$E_{SSD}(u) = \sum_{p \in R} (\nabla I_1^T u(p) + \delta I(p))^2$$

- u(p) is described using an affine transformation valid within the whole region R

$$u(p) = Hp + t \qquad H = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \qquad t = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

$$u(p) = \begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \end{bmatrix} [h_{11} \quad h_{12} \quad h_{21} \quad h_{22} \quad t_1 \quad t_2]^T \qquad u(p) = B(p)\beta$$

$$E_{SSD}(u) = \sum_{p \in R} (\nabla I_1^T B(p)\beta + \delta I(p))^2$$

$$\frac{\partial E_{SSD}(u)}{u} = 0 \qquad [\sum_{p \in R} B(p)^T \nabla I \nabla I_1^T B(p)]\beta = \sum_{p \in R} - B(p)^T \nabla I \delta I$$

$$A\beta = b$$

Affine Motion

Good approximation for :

- Small motions
- Small Camera rotations
- Narrow field of view camera
- When depth variation in the scene is small compared to the average depth and small motion
- Affine camera images a planar scene

Affine Motion

- Affine camera: $p = s\begin{bmatrix} X \\ Y \end{bmatrix}$ $\quad p' = s'\begin{bmatrix} X' \\ Y' \end{bmatrix}$
- 3D Motion: $P' = RP + T$

$$p' = s'\begin{bmatrix} r_1^T P + T_x \\ r_2^T P + T_y \end{bmatrix} = s' R_{22}^T p + s'\begin{bmatrix} r_{13} \\ r_{23} \end{bmatrix} Z + s' T_{xy}$$

- A 3D Plane: $Z = \alpha X + \beta Y + \eta = \frac{1}{s}[\alpha \quad \beta] p + \eta$

$$p' = s' R_{22}^T p + s'\begin{bmatrix} r_{13} \\ r_{23} \end{bmatrix} \frac{1}{s}[\alpha \quad \beta] p + \eta + s' T_{xy}$$

$$u(p) = p' - p = Hp + t$$

Two More Ingredients for Success

- **Iterative solution through image warping**
  - Linearization of the BCE is valid only when u(p) is small
  - Warping brings the second image “closer” to the reference
- **Coarse-to-fine motion estimation for estimating a wider range of image displacements**
  - Coarse levels provide a convex function with unique local minima
  - Finer levels track the minima for a globally optimum solution

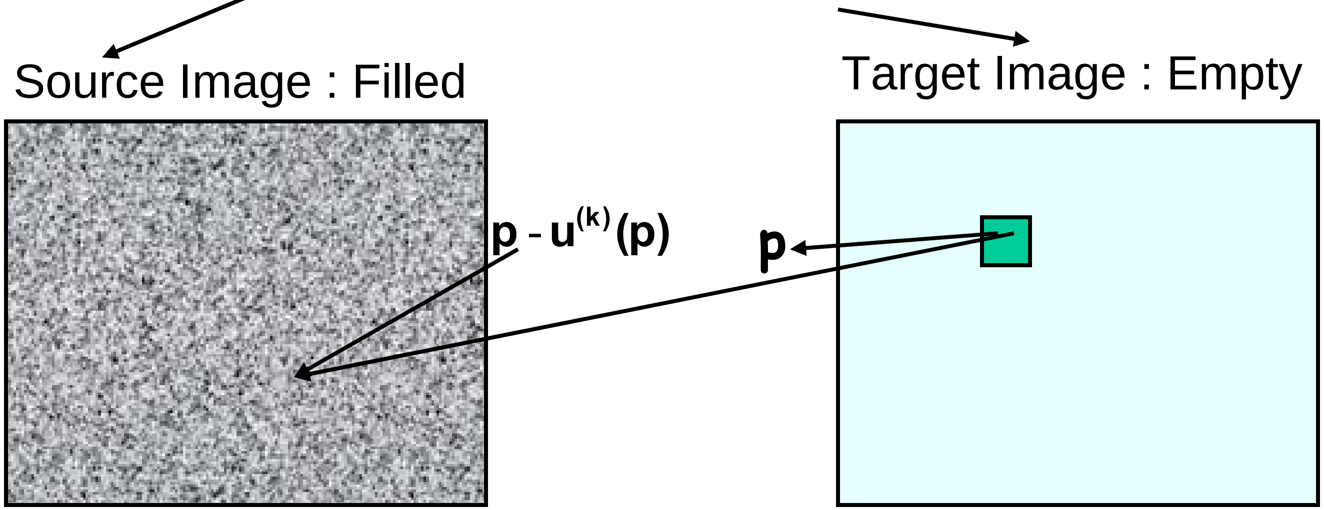
Image Warping

$$I_2(p) = I_1(p - u(p))$$

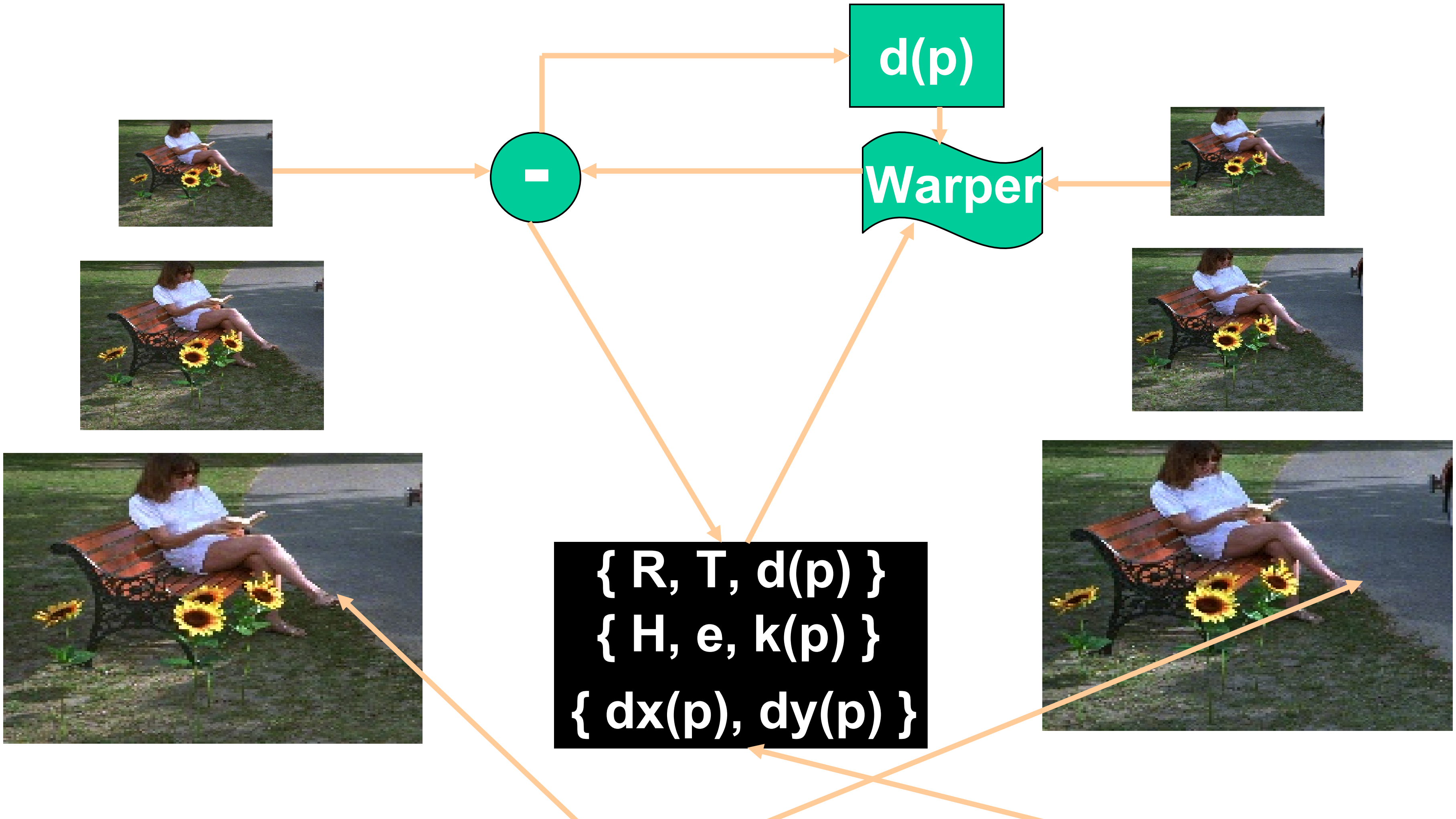
- Express u(p) as: $u(p) = u^{(k)}(p) + \delta u(p)$

$$I_2(p) = I_1(p - u^{(k)}(p) - \delta u(p)) \approx I_1^w(p - \delta u(p))$$

$$I_1(p - u^{(k)}(p)) \approx I_1^w(p)$$



Coarse-to-fine Image Alignment : A Primer



$$\min_{\Theta} \sum_{p} (I_1(p) - I_2(p + u(p;\Theta)))^2$$

Mosaics In Art

…combine individual chips to create a big picture...

Part of the Byzantine mosaic floor that has been preserved in the Church of Multiplication in Tabkha (near the Sea of Galilee).

www.rtlsoft.com/mmmosaic

Image Mosaics

- Chips are images.
- May or may not be captured from known locations of the camera.

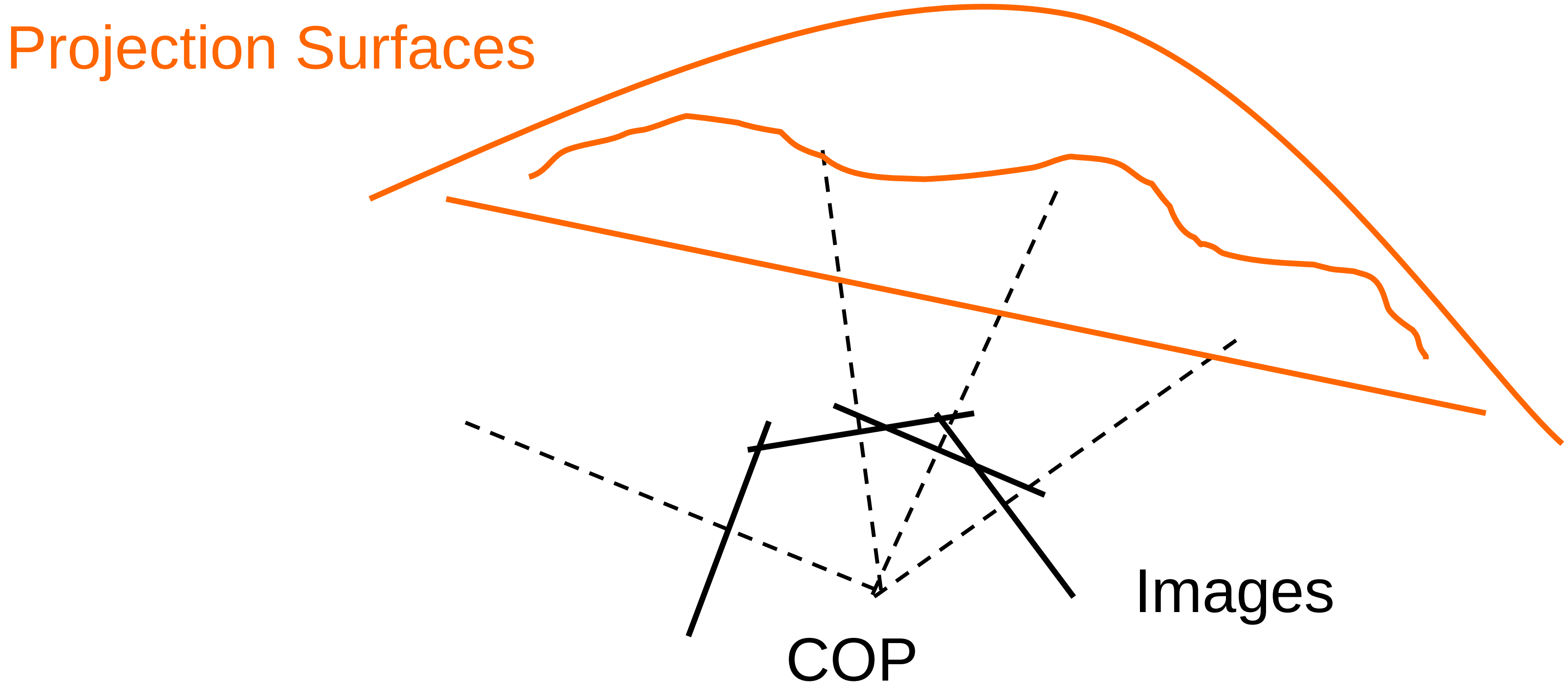
OUTPUT IS A SEAMLESS MOSAIC

VIDEOBRUSH IN ACTION

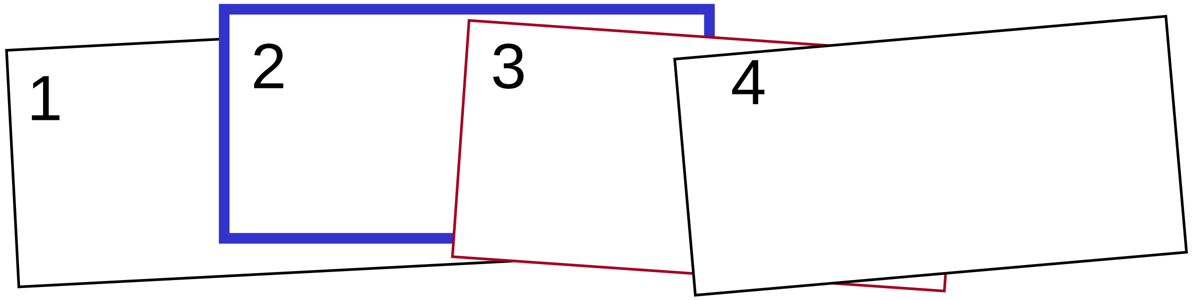
WHAT MAKES MOSAICING POSSIBBLE

…the simplest geometry…

Single Center of Projection for all Images



Planar Mosaic Construction

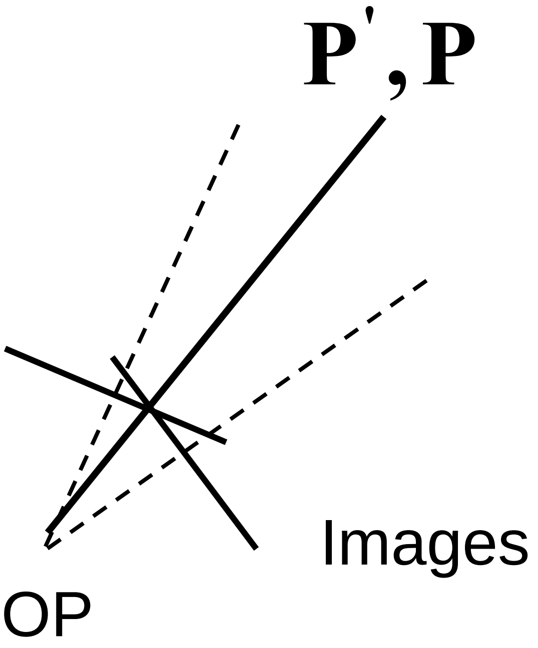


- Align Pairwise: 1:2, 2:3, 3:4, ...
- Select a Reference Frame
- Align all Images to the Reference Frame
- Combine into a Single Mosaic

Virtual Camera (Pan)
Image Surface - Plane
Projection - Perspective

Key Problem

What Is the Mapping From Image Rays to the Mosaic Coordinates ?



Rotations/Homographies
Plane Projective Transformations

$$\mathbf{P}' = \mathbf{R}\mathbf{P}$$

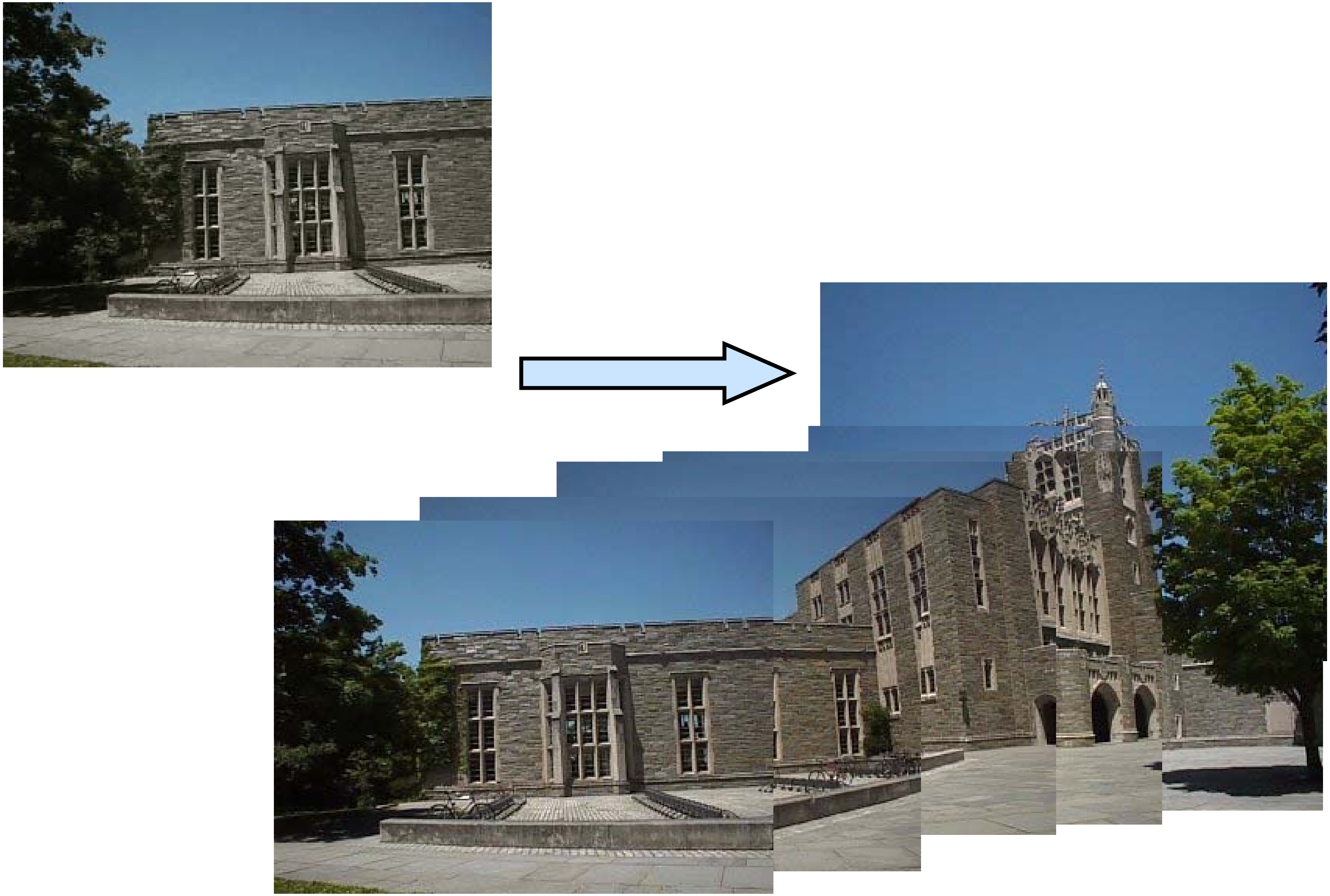
$$\mathbf{p}_c' \approx \mathbf{R}\mathbf{p}_c$$

$$\mathbf{K}'\mathbf{p}' \approx \mathbf{R}\mathbf{K}\mathbf{p}$$

$$\mathbf{p}' \approx \mathbf{K}'^{-1}\mathbf{R}\mathbf{K}\mathbf{p}$$

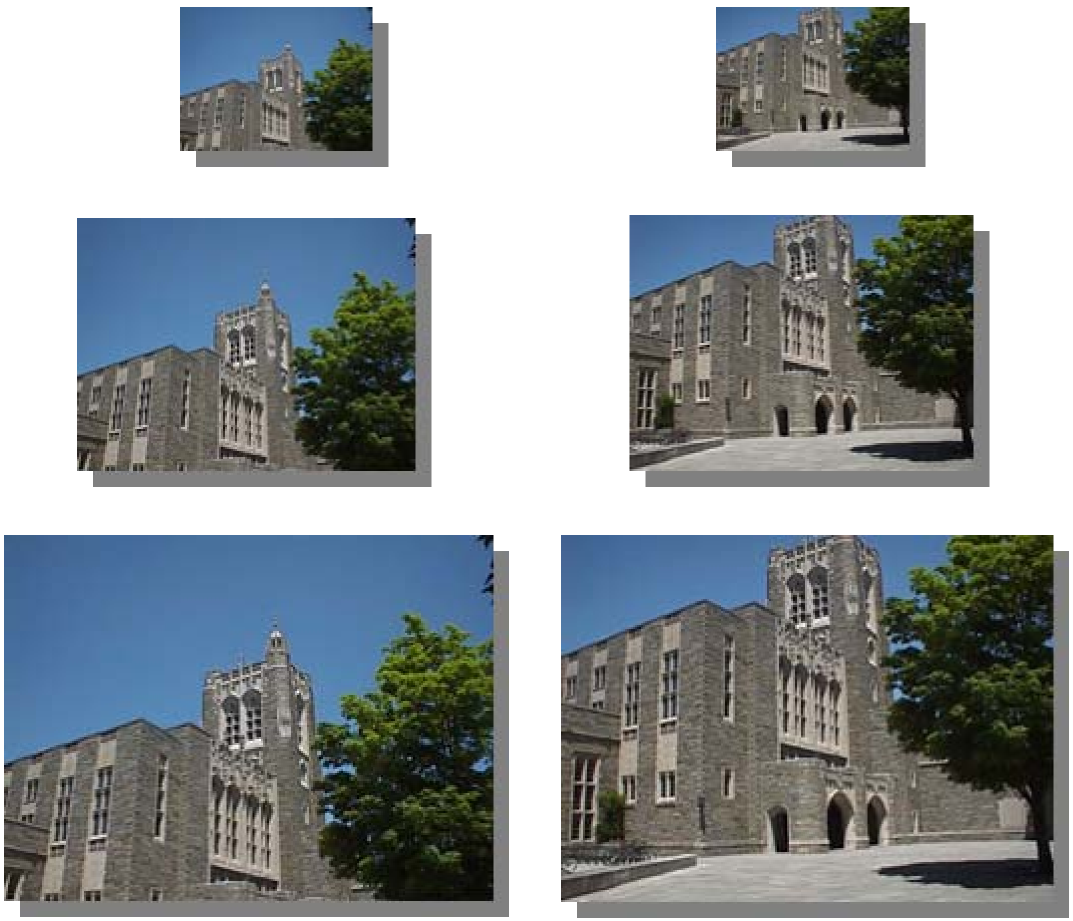
$$\mathbf{p}' \approx \mathbf{H}_\infty\mathbf{p}$$

IMAGE ALIGNMENT IS A BASIC REQUIREMENT



PYRAMID BASED COARSE-TO-FINE ALIGNMENT

… a core technology ...



- Coarse levels reduce search.
- Models of image motion reduce modeling complexity.
- Image warping allows model estimation without discrete feature extraction.
- Model parameters are estimated using iterative non-linear optimization.
- Coarse level parameters guide optimization at finer levels.

ITERATIVE SOLUTION OF THE ALIGNMENT MODEL

Assume that at the *k*th iteration, $\mathbf{P}^{(k)}$, is available

$$\mathbf{I}^{w}(\mathbf{p}^{w}) = \mathbf{I}'(\mathbf{p}') = \mathbf{I}'(\mathbf{P}^{(k)}\mathbf{p}^{w})$$

model the residual transformation between the coordinate systems, $\mathbf{p}^{w}$ and $\mathbf{p}$, as:

$$\mathbf{p}^{w} \approx [\mathbf{I}+\mathbf{D}]\mathbf{p}$$

$$\mathbf{I}^{w}(\mathbf{p}^{w}(\mathbf{p};\mathbf{D})) \approx \mathbf{I}^{w}(\mathbf{p}^{w}(\mathbf{p};0)) + \nabla \mathbf{I}^{w^{T}} \frac{\partial \mathbf{p}^{w}}{\partial \mathbf{D}}\Big|_{\mathbf{D}=0} \mathbf{D} = \mathbf{I}(\mathbf{p})$$

$$\frac{\partial \mathbf{p}^{w}}{\partial \mathbf{D}}\Big|_{\mathbf{D}=0} \qquad \mathbf{p}^{w} = \begin{bmatrix} \dfrac{(1+d_{11})x + d_{12}y + d_{13}}{d_{31}x + d_{32}y + 1} \\ \dfrac{d_{21}x + (1+d_{22})y + d_{23}}{d_{31}x + d_{32}y + 1} \end{bmatrix} \therefore \frac{\partial \mathbf{p}^{w}}{\partial \mathbf{D}}\Big|_{\mathbf{D}=0} = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x^{2} & -xy \\ 0 & 0 & 0 & x & y & 1 & -xy & -y^{2} \end{bmatrix}$$

$$\mathbf{P}^{(k+1)} \approx \mathbf{P}^{(k)}[\mathbf{I}+\mathbf{D}]$$

ITERATIVE REWEIGHTED SUM OF SQUARES

- Given a solution $\boldsymbol{\Theta}^{(m)}$ at the mth iteration, find $\boldsymbol{\delta\Theta}$ by solving :

$$\sum_{l}\sum_{i}\frac{\dot{\rho}(r_i)}{r_i}\frac{\partial r_i}{\partial \theta_k}\frac{\partial r_i}{\partial \theta_l}\partial\theta_l = -\frac{\dot{\rho}(r_i)}{r_i}r_i\frac{\partial r_i}{\partial \theta_k} \qquad \forall k$$

w_i

- w_i acts as a soft outlier rejecter :

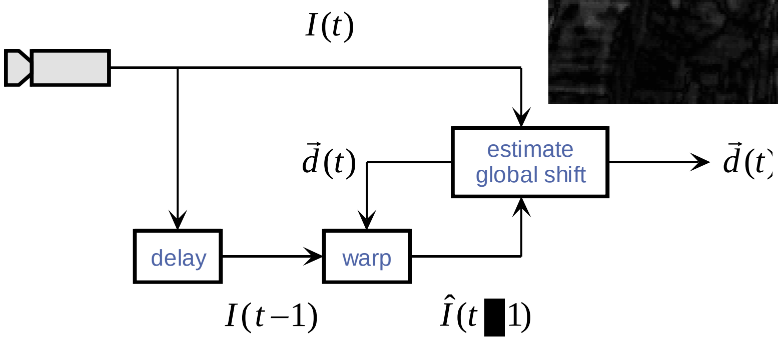
$$\frac{\dot{\rho}_{SS}(r)}{r} = \frac{1}{\sigma^2} \qquad \frac{\dot{\rho}_{GM}(r)}{r} = \frac{2\sigma^2}{(\sigma^2 + r^2)^2}$$

PROGRESSIVE MODEL COMPLEXITY

…combining real-time capture with accurate alignment...

- Provide user feedback by coarsely aligning incoming frames with a low order model
  - *robust matching that covers a wide search range*
  - *achieve about 6-8 frames a sec. on a Pentium 200*
- Use the coarse alignment parameters to seed the fine alignment
  - *increase model complexity from similarity, to affine, to projective parameters*
  - *coarse-to-fine alignment for wide range of motions and managing computational complexity*

COARSE-TO-FINE ALIGNMENT

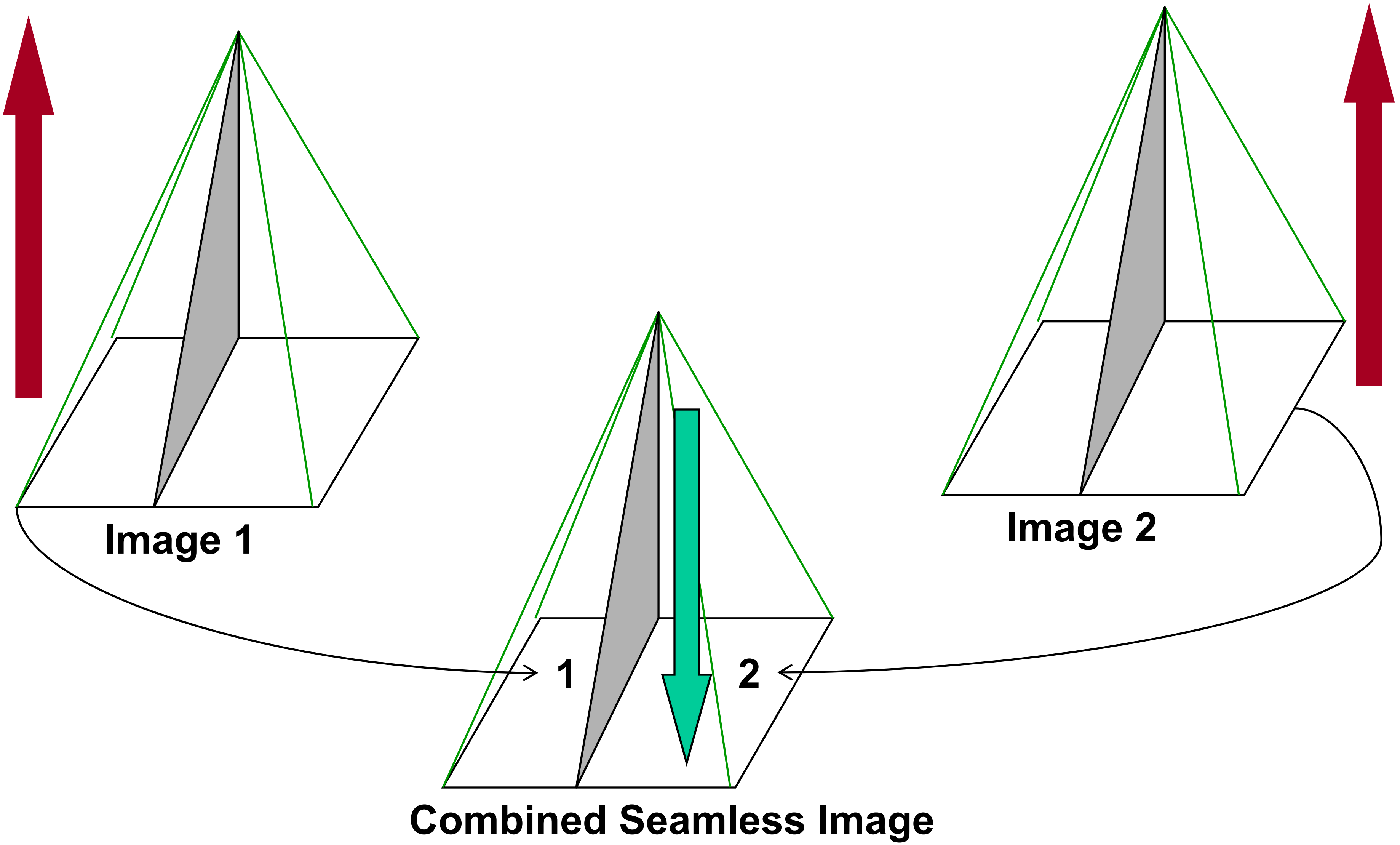


VIDEO MOSAIC EXAMPLE

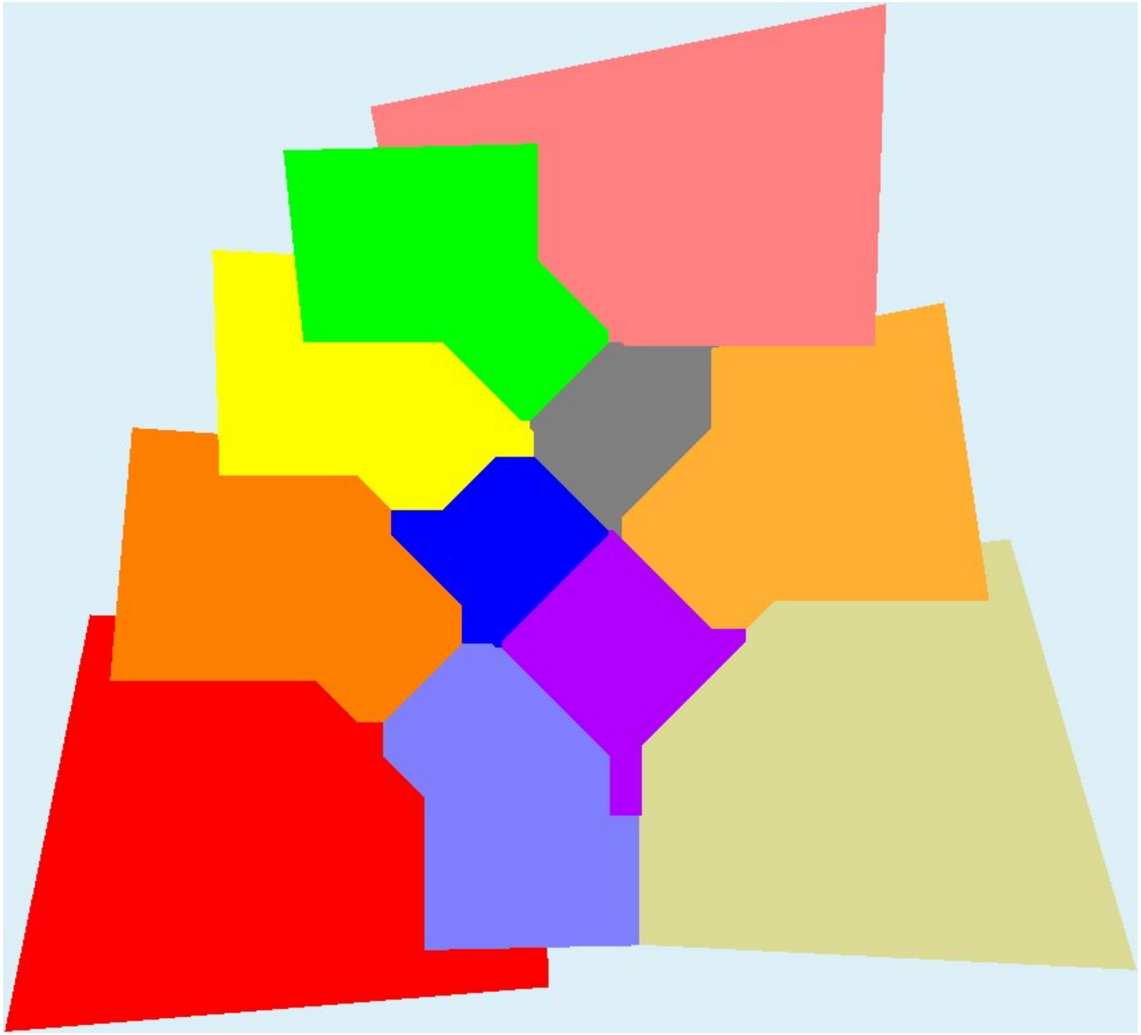
Princeton Chapel Video Sequence
54 frames

UNBLENDED CHAPEL MOSAIC

Image Merging withLaplacian Pyramids



VORONOI TESSELATIONS W/ L1 NORM



BLENDED CHAPEL MOSAIC

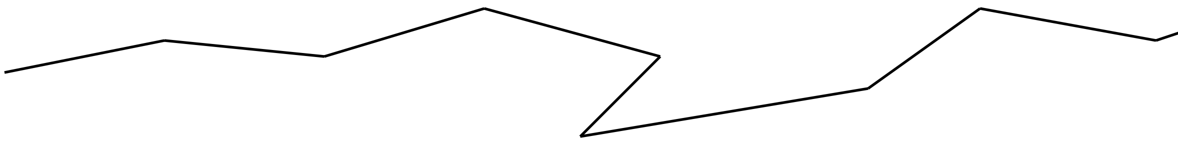
Image Matching Sidebar

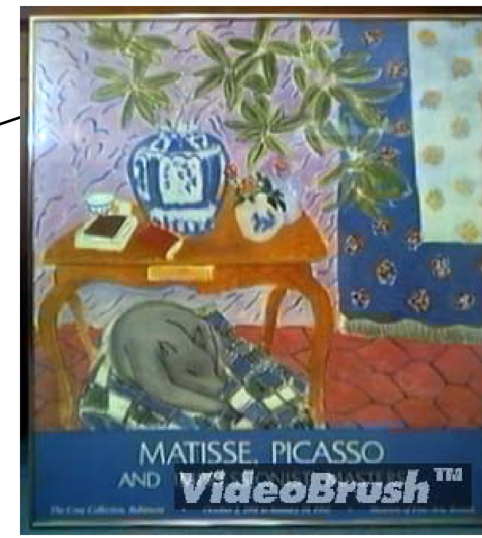
Discrete Features

Demo

1D vs. 2D SCANNING

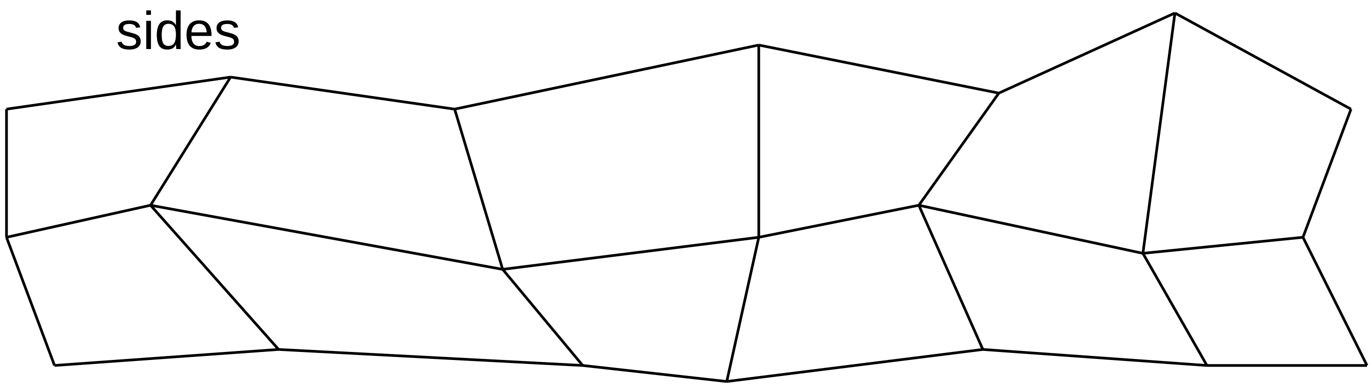
- 1D : The topology of frames is a ribbon or a string.
  Frames overlap only with their temporal neighbors.



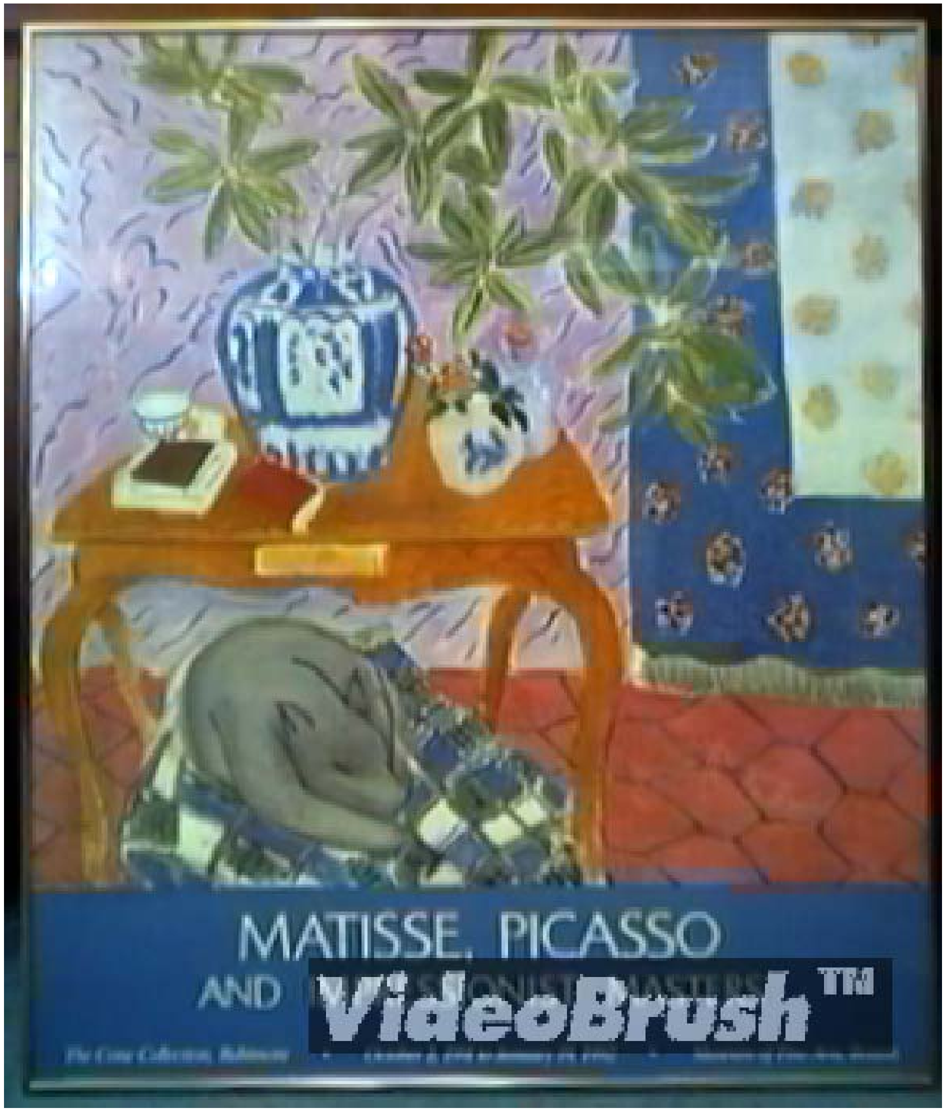


(A 300x332 mosaic captured by mosaicing a 1D sequence of 6 frames)

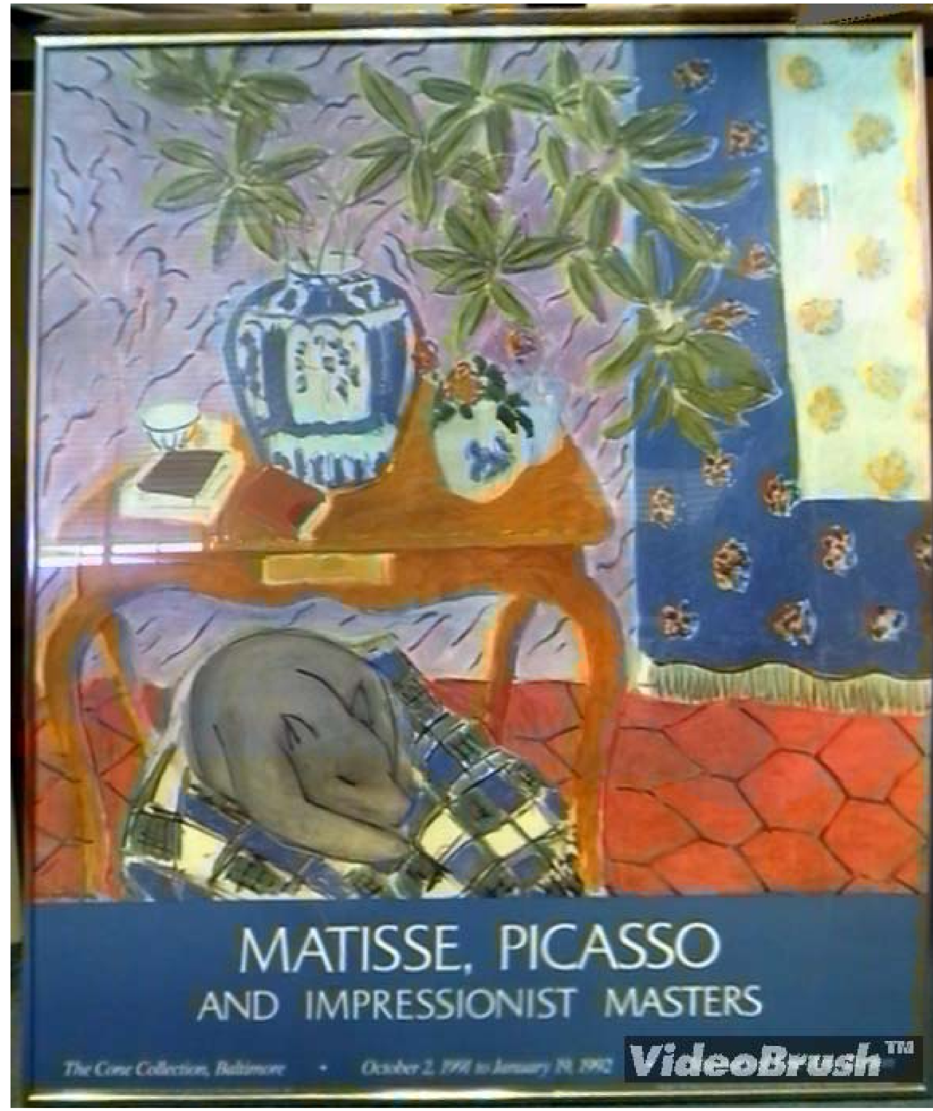
- 2D : The topology of frames is a 2D graph
  Frames overlap with neighbors on many sides



1D vs. 2D SCANNING

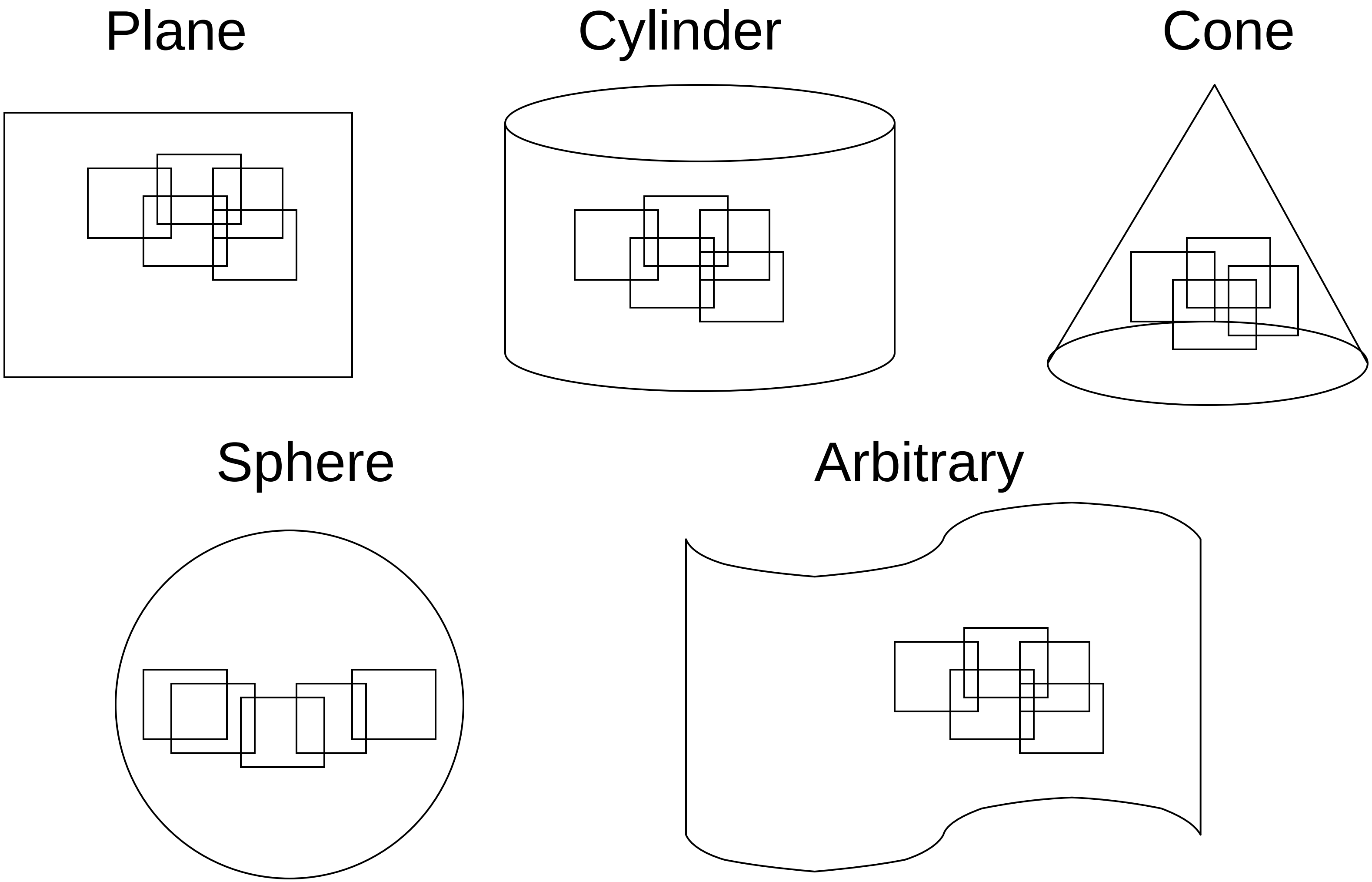


The 1D scan scaled by 2 to 600x692



A 2D scanned mosaic of size 600x692

CHOICE OF 1D/2D MANIFOLD



1D SCANNING

... handling camera tilt and wrap around ...

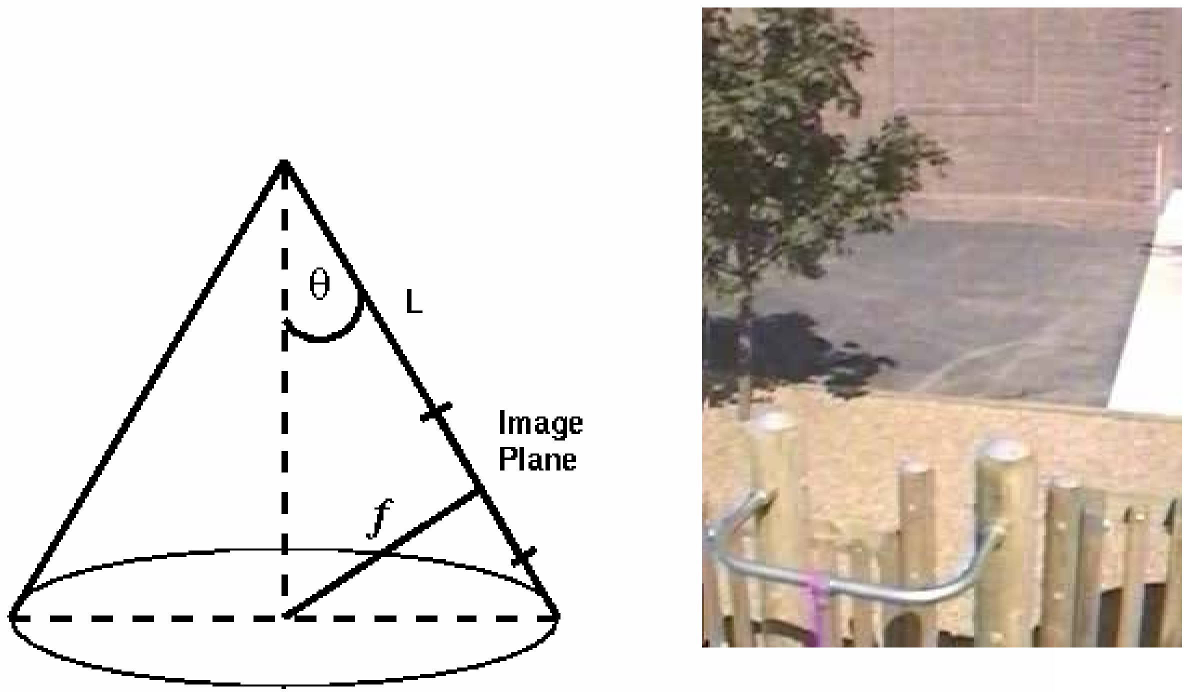


Figure 1: 1D scanning with the optical axis tilted by θ resulting in the cone geometry for the mosaic.

DEVELOPING THE CONE INTO A RECTANGULAR PLANAR MOSAIC

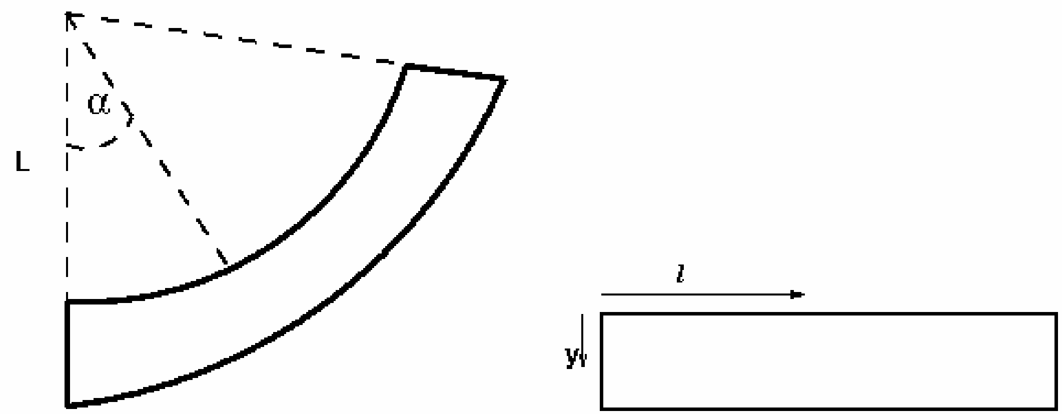


Figure 2: **Left**: The developed cone mosaic resulting in a curved mosaic on the plane. **Right**: The rectified mosaic with a rectilinear coordinate system whose mapping to the curved mosaic is given in the text.

$$\begin{bmatrix} l \\ y \end{bmatrix} \rightarrow y \begin{bmatrix} \sin\alpha \\ \cos\alpha \end{bmatrix} + \begin{bmatrix} L\sin\alpha \\ L(\cos\alpha - 1) \end{bmatrix}$$

where $\alpha = \frac{l}{L}$, and l, L, y are as shown.

THE “DESMILEY” ALGORITHM

- Compute 2D rotation and translation between successive frames
- Compute L by intersecting central lines of each frame
- Fill each pixel [l   y] in the rectified planar mosaic by mapping it to the appropriate video frame

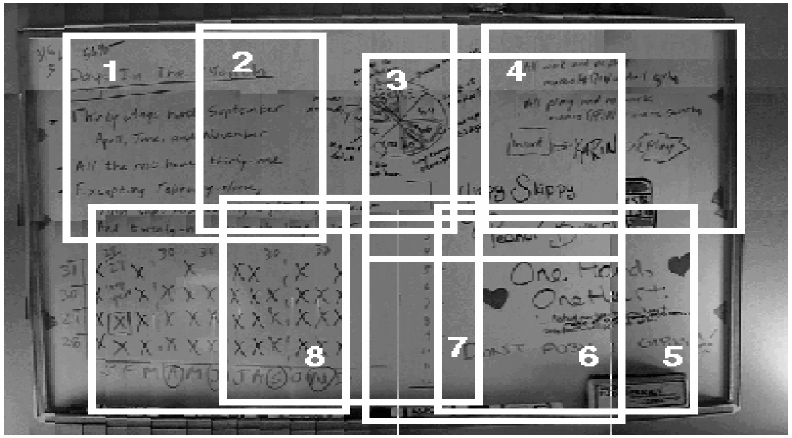
2D MOSAICING THROUGH TOPOLOGY INFERENCE
&
LOCAL TO GLOBAL ALIGNMENT

… automatic solution to two key problems ...

- Inference of 2D neighborhood relations (topology) between frames
  - Input video just provides a temporal 1D ordering of frames
  - Need to infer 2D neighborhood relations so that local constraints may be setup between pairs of frames
- Globally consistent alignment and mosaic creation
  - Choose appropriate alignment model
  - Local constraints incorporated in a global optimization

PROBLEM FORMULATION

Given an arbitrary scan of a scene



Create a globally aligned mosaic by minimizing

$$\min_{\{\mathbf{P_i}\}} E = \sum_{ij \in G} E_{ij} + \sum_i E_i + \sigma^2 (\text{Area of the mosaic})$$

Like an MDL measure :

Create a compact appearance while being geometrically consistent

ERROR MEASURE

$$\min_{\{\mathbf{P_i}\}} E = \sum_{ij \in G} E_{ij} + \sum_{i} E_i + \sigma^2 (\text{Area of the mosaic})$$

where

$\mathbf{P_i}$: Reference-to-image mapping, $\mathbf{u_i} = \mathbf{P_i X}$

E_{ij} : Any measure of alignment error between neighbors i and j

G : Graph that represents the neighborhood relations

E_i : Frame to reference error term to allow for a priori criterion like least distortion transformation

ALGORITHMIC APPROACH

From a 1D ordered collection of frames

to

A Globally consistent set of alignment parameters

Iterate through

1. Graph Topology Determination

   Given: pose of all frames
   Establish neighborhood relations $\rightarrow$ min(Area of Mosaic)
   $\rightarrow$ Graph G

2. Local Pairwise Alignment

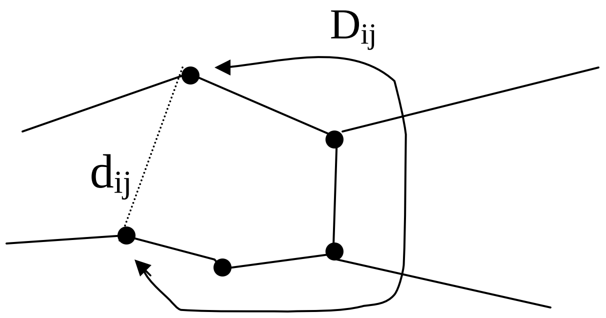
   Given: G
   Quality measure validates hypothesized arcs
   Provides pairwise constraints

3. Globally Consistent Alignment

   Given: pairwise constraints
   Compute reference-to-frame pose parameters $\rightarrow$ min $\sum E_{ij}$

GRAPH TOPOLOGY DETERMINATION

- Given: Current estimate of pose*
- Lay out each frame on the 2D manifold (plane, sphere, etc.)
- Hypothesize new neighbors based on
  - proximity
  - predictability of relative pose
  - non redundancy w.r.t. current G



- Specifically, try arc (i,j) if

  Normalized Euclidean dist d_{ij} << Path distance D_{ij}

- Validate hypothesis by local registration
- Add arc to G if good quality registration

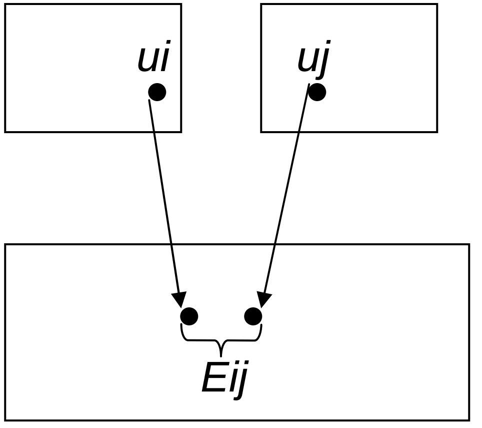
* Initialize using low order frame-to-frame mosaic algorithm on a plane

LOCAL COARSE & FINE ALIGNMENT

- Given: a frame pair to be registered
- Coarse alignment
  - Low order parametric model e.g. shift, or 2D R & T
  - Majority consensus among subimage estimates
- Fine alignment [Bergen, ECCV 92]
  - Coarse to fine over Laplacian pyramid
  - Progressive model complexity, up to projective
  - Incrementally adjust motion parameters to minimize SSD
- Quality measure
  - Normalized correlation helps reject invalid registrations

GLOBALLY CONSISTENT ALIGNMENT

- Given: arcs ij in graph G of neighbors
- The local alignment parameters, *Qij,* help establish feature correspondence between *i* and *j*
- If $u_{\mathbf{il}}$ and $u_{\mathbf{jl}}$ are corresponding points in frames i,j, then



$$E_{ij} = |\mathbf{P_i}^{-1}(u_{\mathbf{il}}) - \mathbf{P_j}^{-1}(u_{\mathbf{jl}})|^2$$

- Incrementally adjust poses $\mathbf{P_i}$ to minimize

$$\min_{\{\mathbf{P_i}\}} E = \sum_{ij \in G} E_{ij} + \sum_i E_i$$

SPECIFIC EXAMPLES : 1. PLANAR MOSAICS

- Mosaic to frame transformation model: $\mathbf{u} \approx \mathbf{P_i} \ \mathbf{X}$

- Local Registration
  - Coarse 2D translation & fine 2D projective alignment

- Topology : Neighborhood graph defined over a plane
  - Initial graph topology computed with the 2D T estimates
  - Iterative refinement using arcs based on projective alignment

- Global Alignment

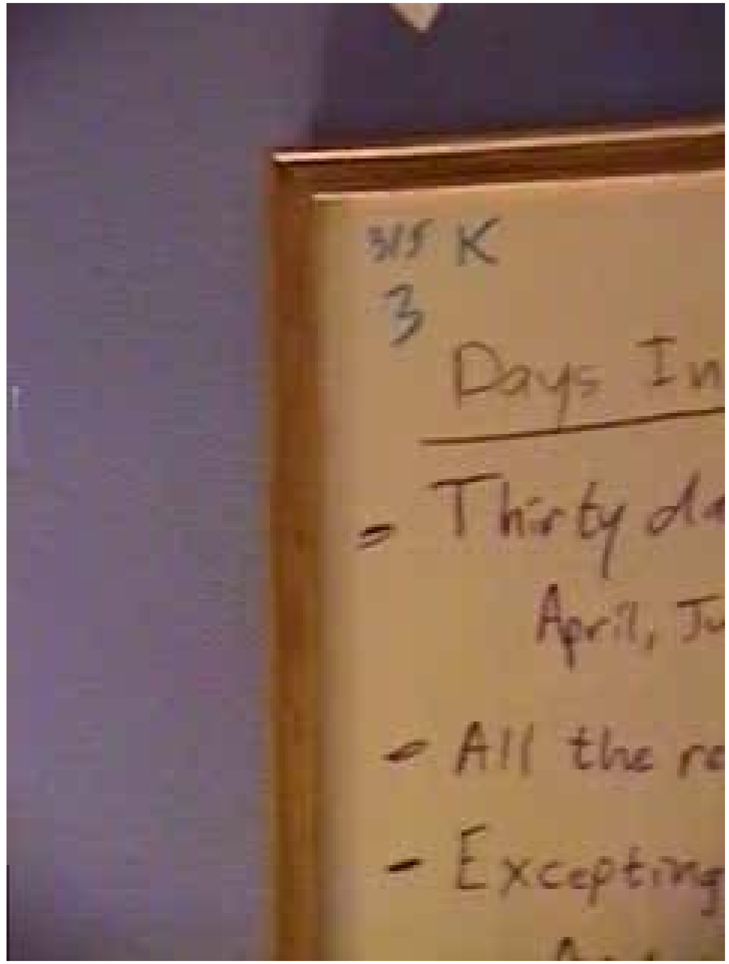
$$E_{ij} = \sum_{k} | \prod(\mathbf{A_i u_{ik}}) - \prod(\mathbf{A_j u_{jk}}) |^2$$

Pair Wise Alignment Error

$$E_i = \sum_{k=1}^{2} | (\prod(\mathbf{A_i}\alpha_\mathbf{k}) - \prod(\mathbf{A_j}\beta_\mathbf{k})) - (\alpha_\mathbf{k} - \beta_\mathbf{k}) |^2$$

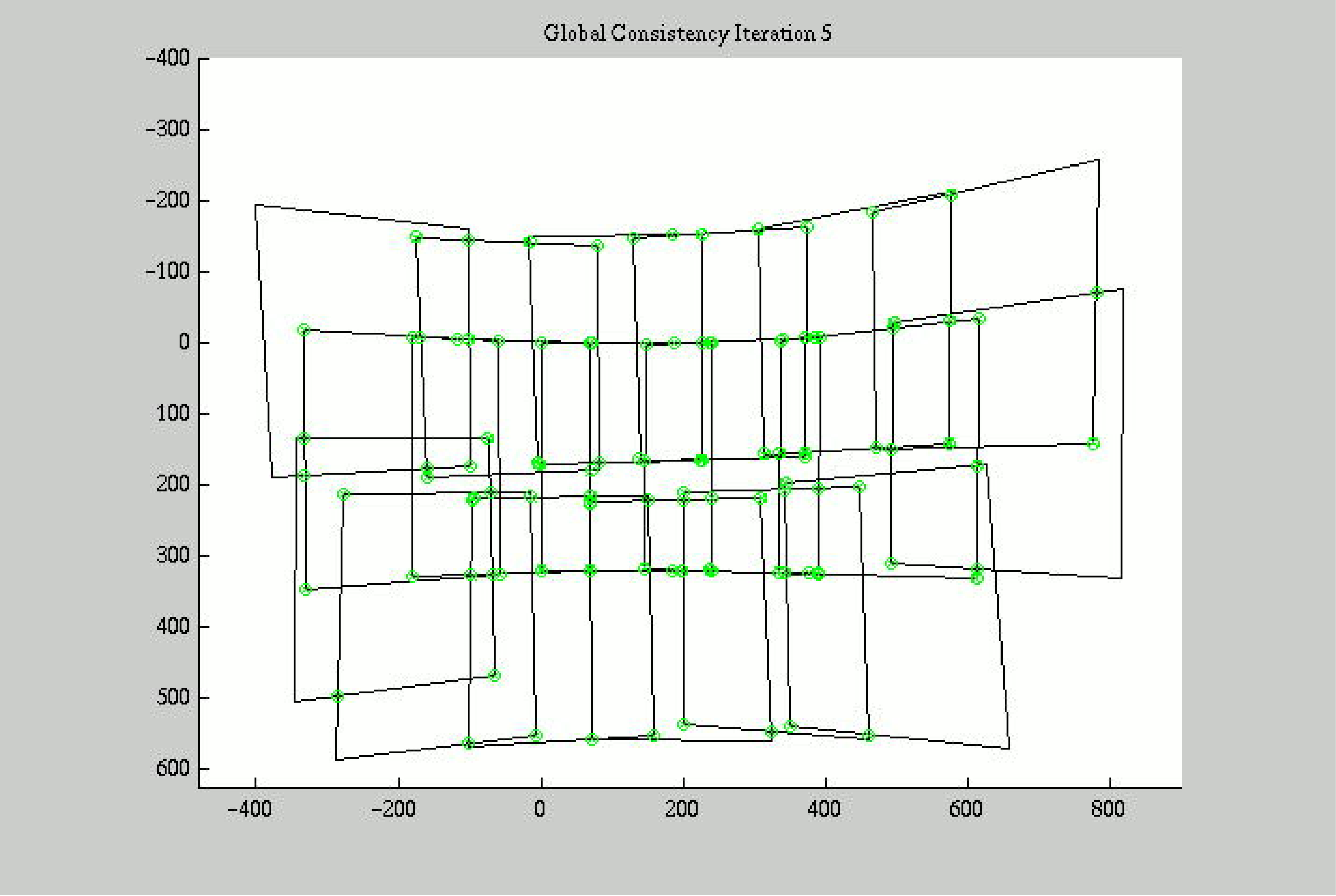
Minimum Distortion Error

PLANAR TOPOLOGY EVOLUTION

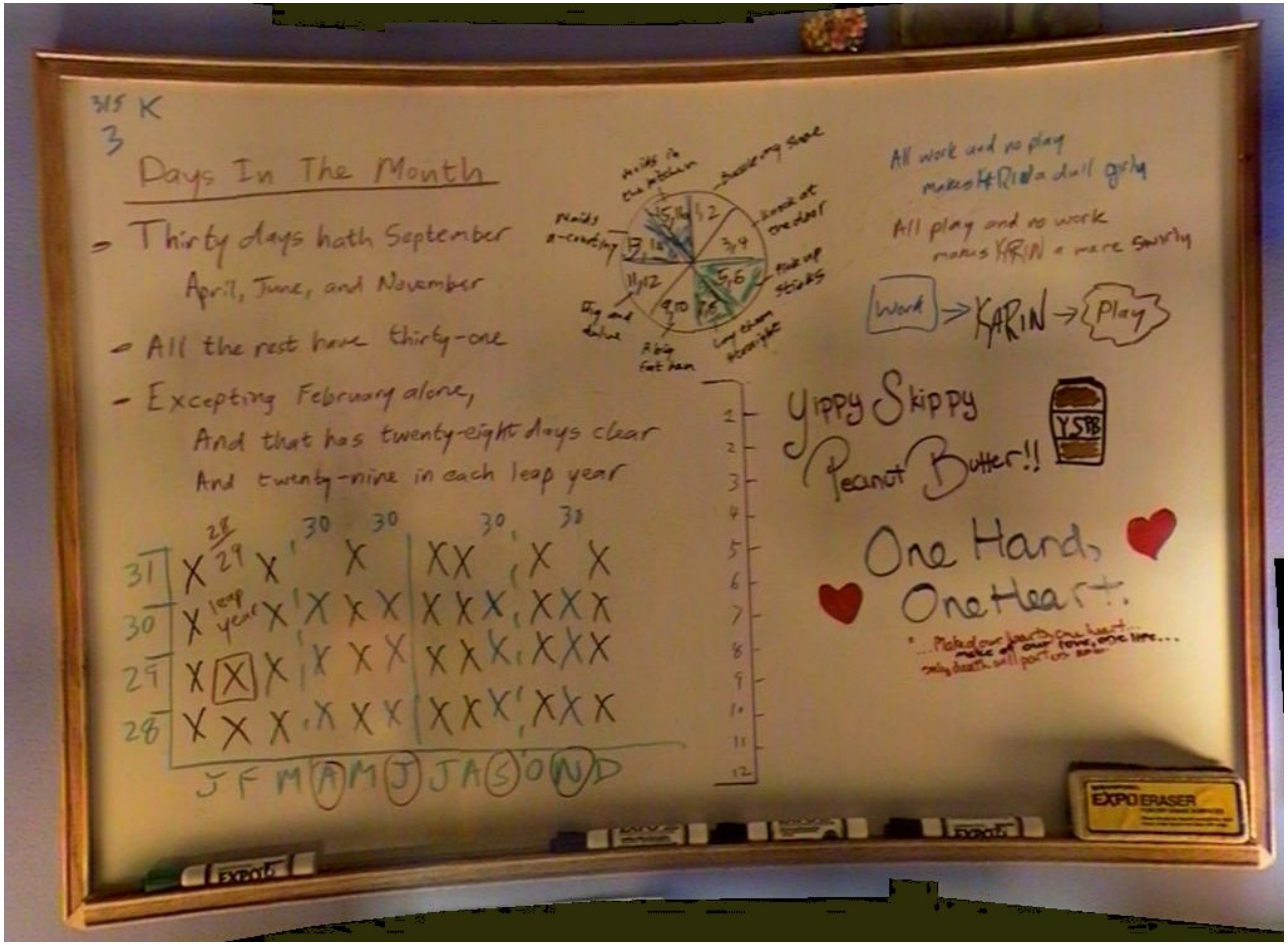


Whiteboard Video Sequence
75 frames

PLANAR TOPOLOGY EVOLUTION



FINAL MOSAIC



SPECIFIC EXAMPLES : 2. SPHERICAL MOSAICS

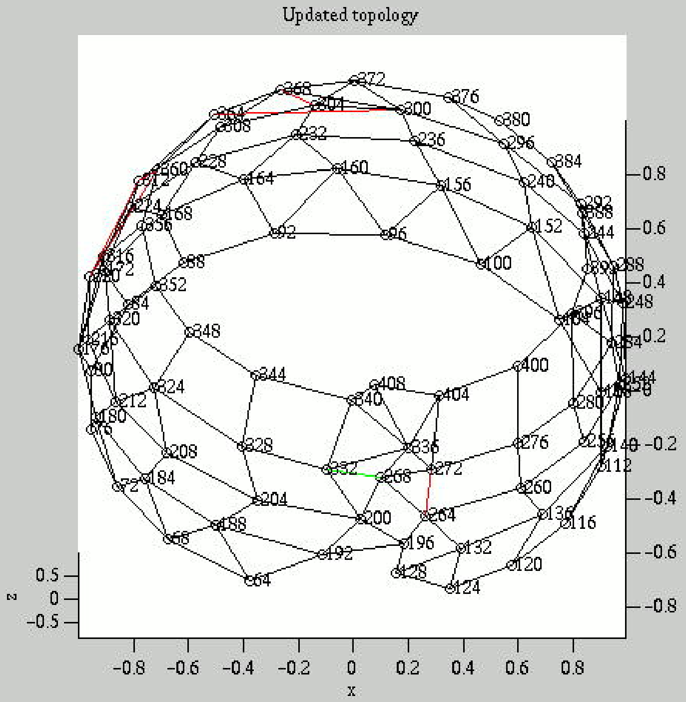
- Frame to mosaic transformation model: $\mathbf{u} \approx \mathbf{F}\mathbf{R_i}^T\mathbf{X}$
- Local Registration
  - Coarse 2D translation & fine 2D projective alignment
- Parameter Initialization
  - Compute **F** and **R**'s from the 2D projective matrices
- Topology :
  - Initial graph topology computed with the 2D R & T estimates on a plane
  - Subsequently the topology defined on a sphere
  - Iterative refinement using arcs based on alignment with **F** and **R**'s
- Global Alignment

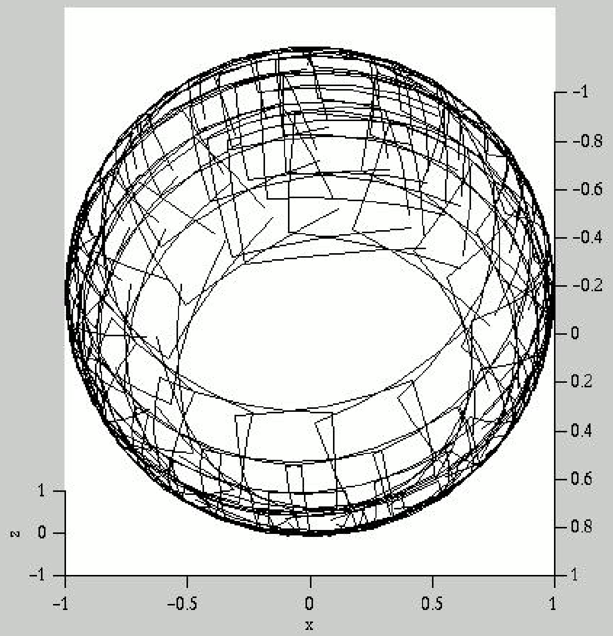
$$E_{ij} = \sum_k |\mathbf{R_i}\mathbf{F}^{-1}\mathbf{u_{ik}} - \mathbf{R_j}\mathbf{F}^{-1}\mathbf{u_{jk}}|^2$$

SPHERICAL MOSAICS

Sarnoff Library Video
Captures almost the complete sphere
with 380 frames

SPHERICAL TOPOLOGY EVOLUTION



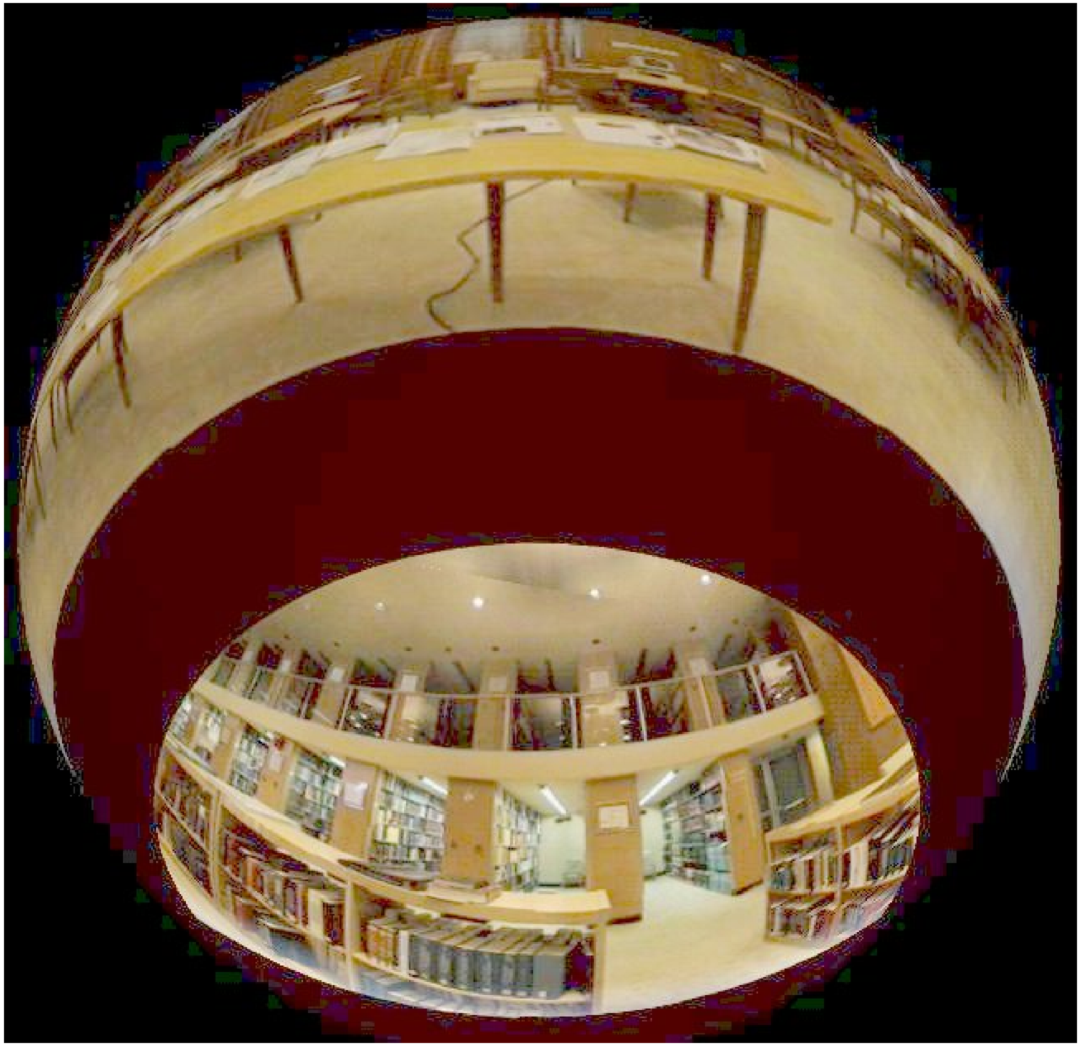


SPHERICAL MOSAIC

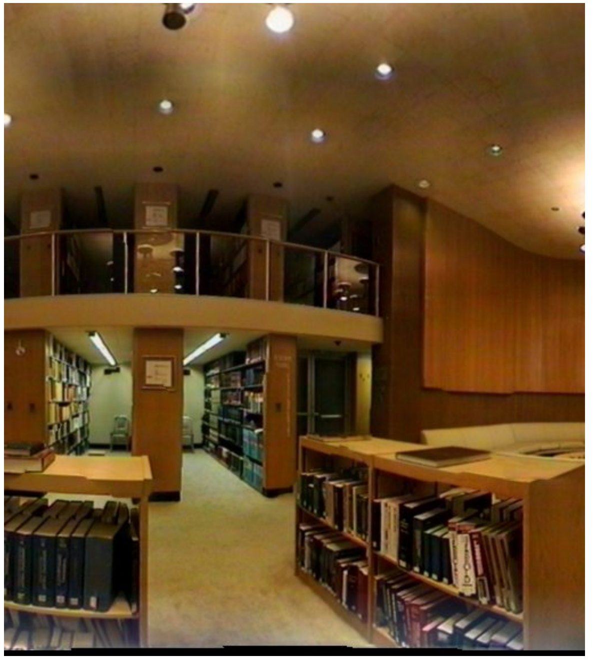
Sarnoff Library

SPHERICAL MOSAIC

Sarnoff Library



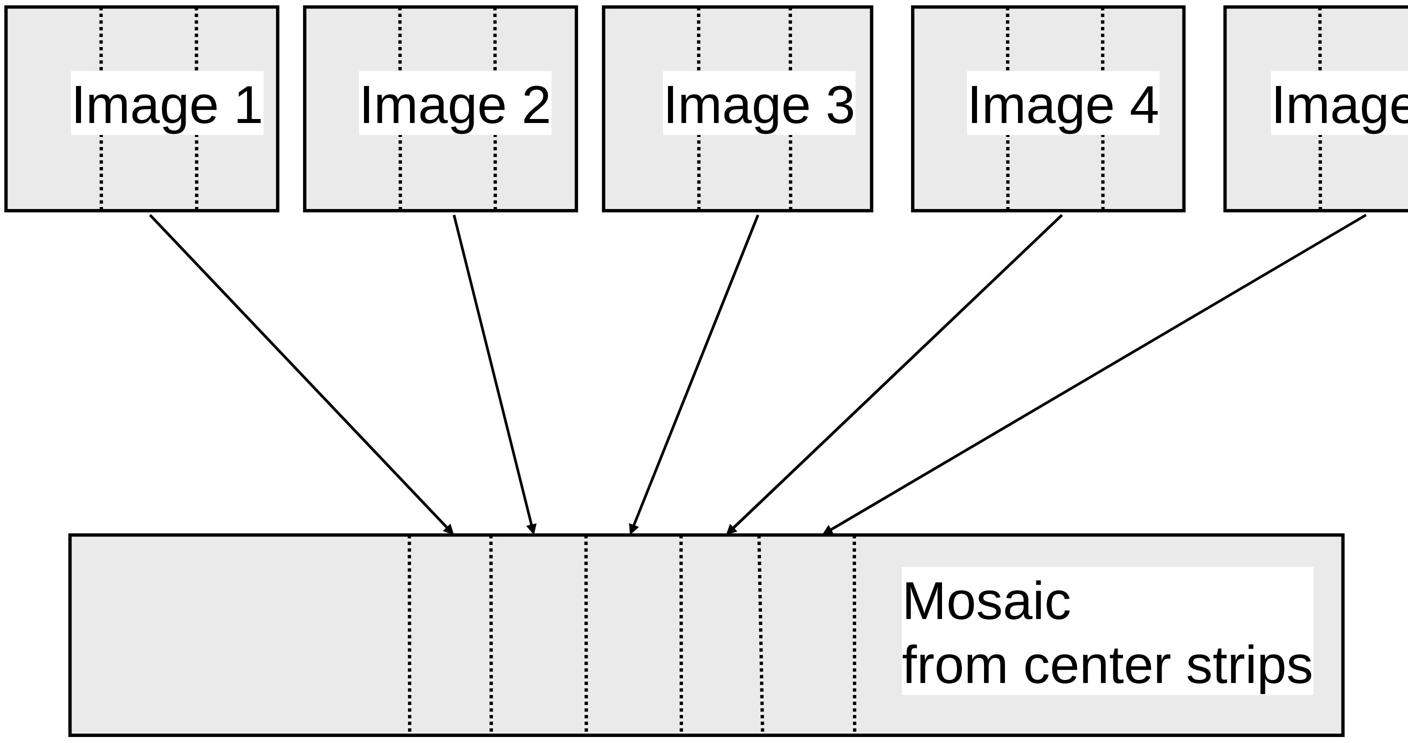
NEW SYNTHESIZED VIEWS



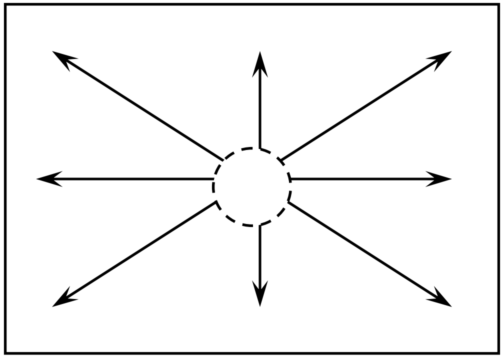
FINAL MOSAIC

Princeton University Courtyard

Mosaicing from Strips

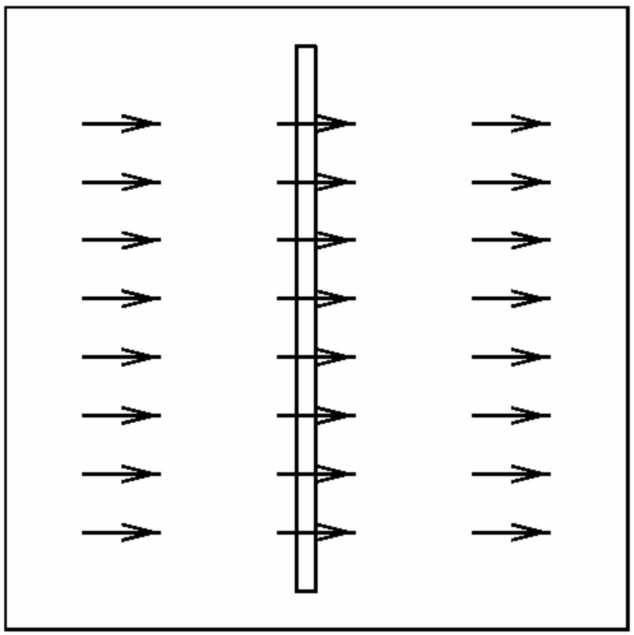


Problem: Forward Translation

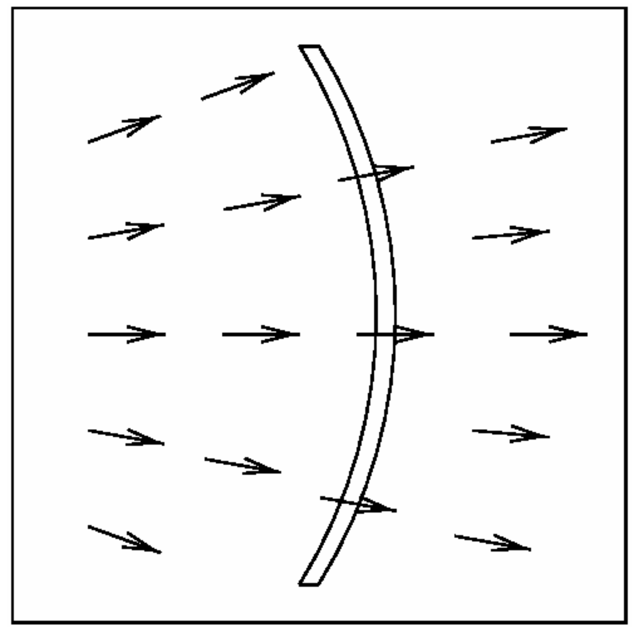


General Camera Motion

- Strip Perpendicular to Optical Flow
- Cut/Paste Strip (warp to make Optical Flow parallel)

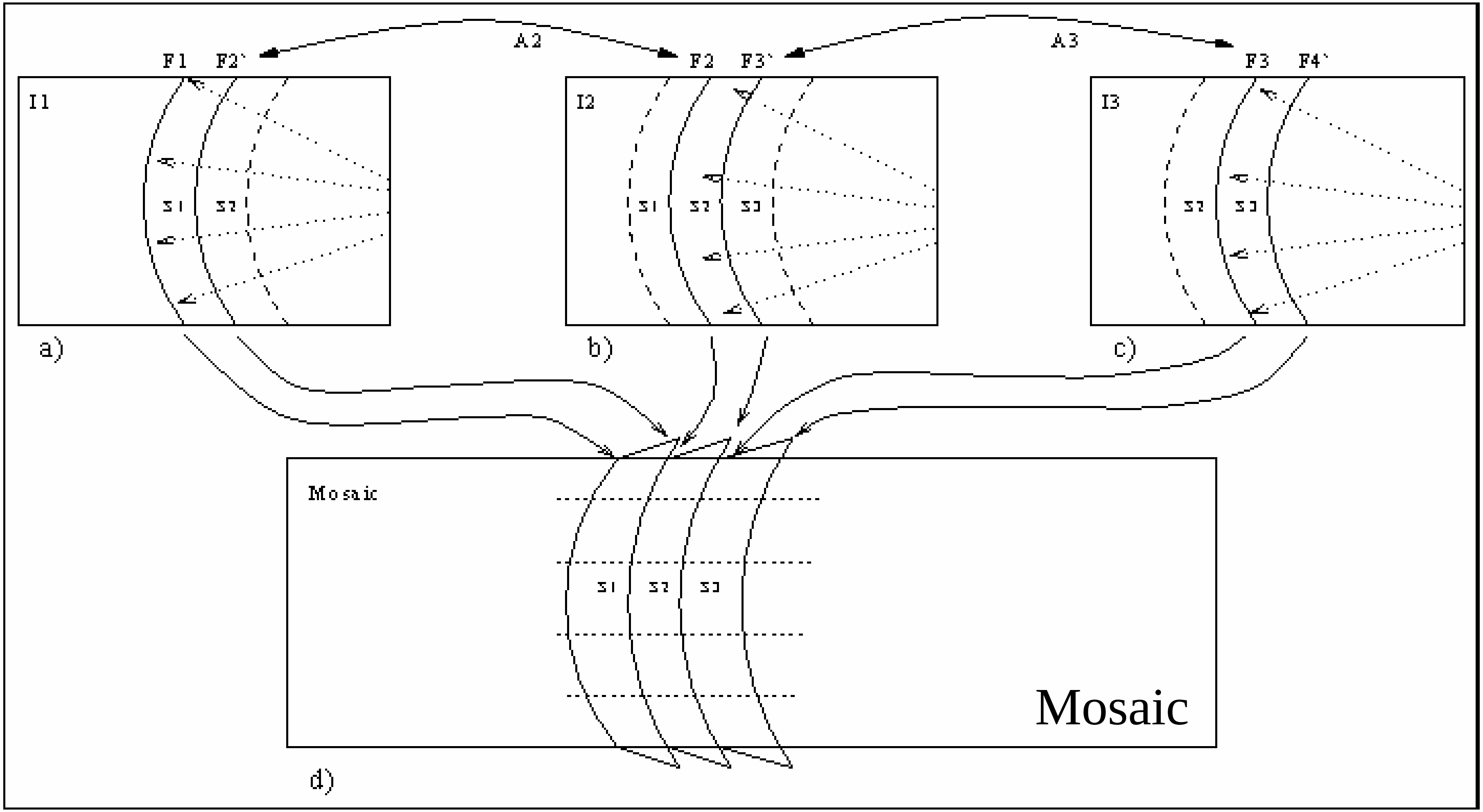


Parallel Flow:
Straight Strip



Radial Flow (FOE):
Circular Strip

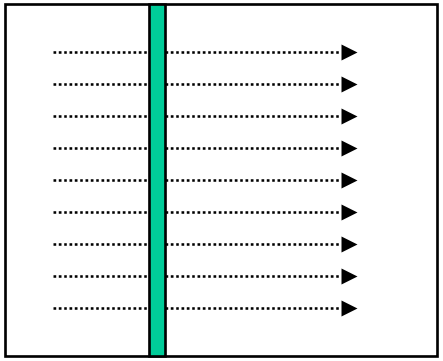
Mosaic Construction



Simple Cases

Horizontal Translation

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} a \\ 0 \end{pmatrix}$$

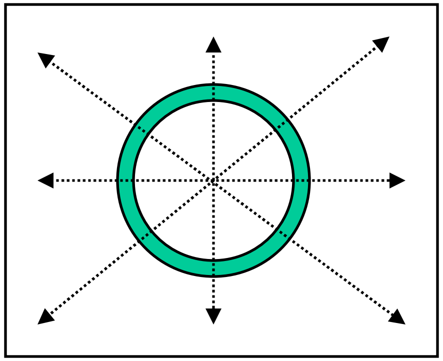


$$ax + M = 0$$

(M determines displacement)

Zoom

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} bx \\ by \end{pmatrix}$$



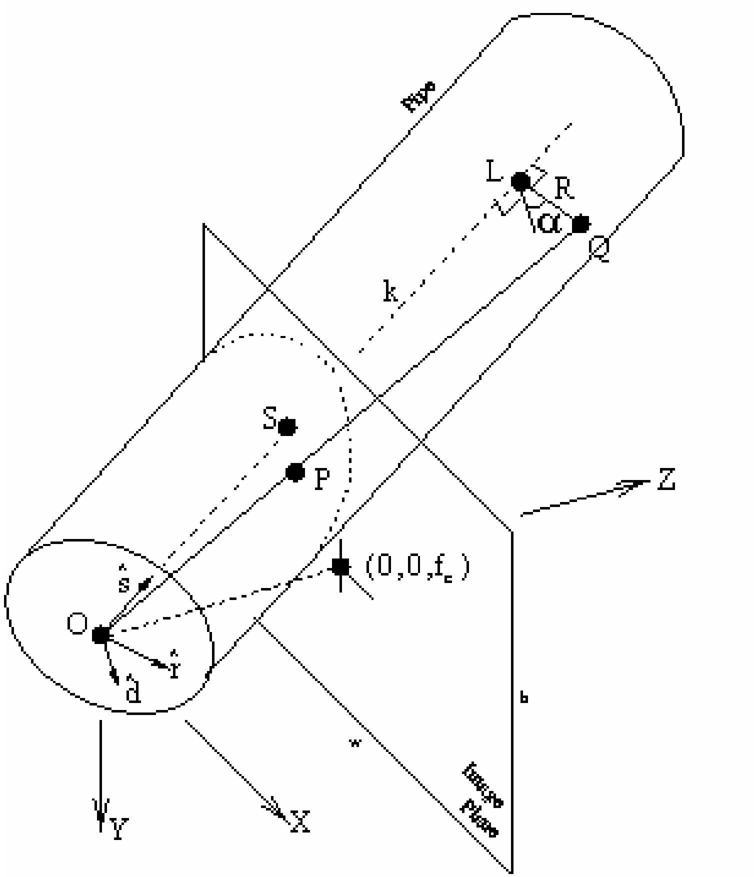
$$\frac{b}{2}(x^2 + y^2) + M = 0$$

(M determines radius)

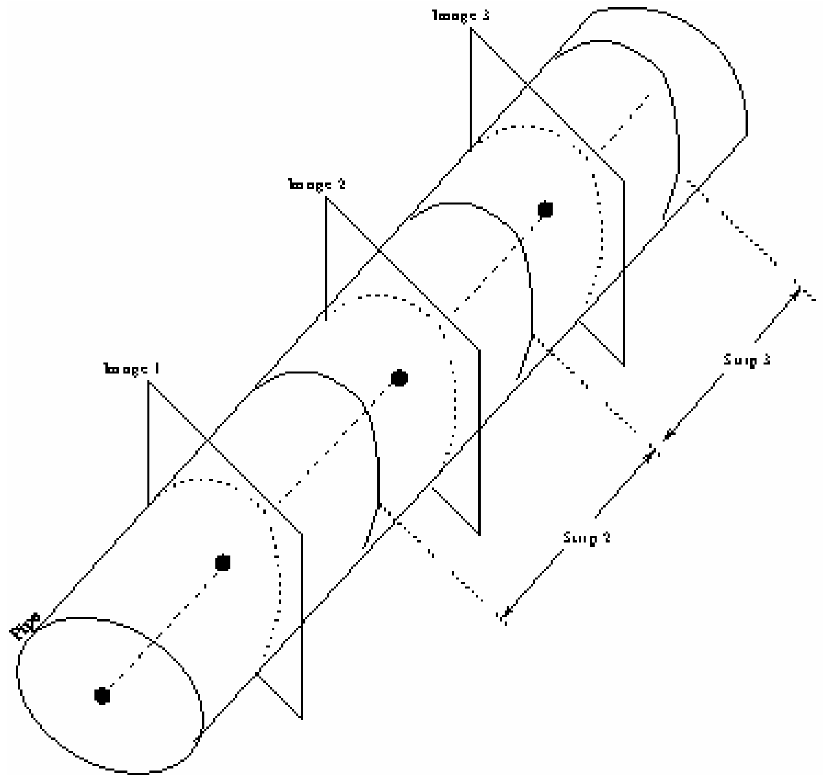
Manifold for Forward Motion

- Stationary (but rotating) Camera
  - Viewing Sphere
- Translating Camera
  - Sphere carves a “Pipe” in space

Pipe Projection

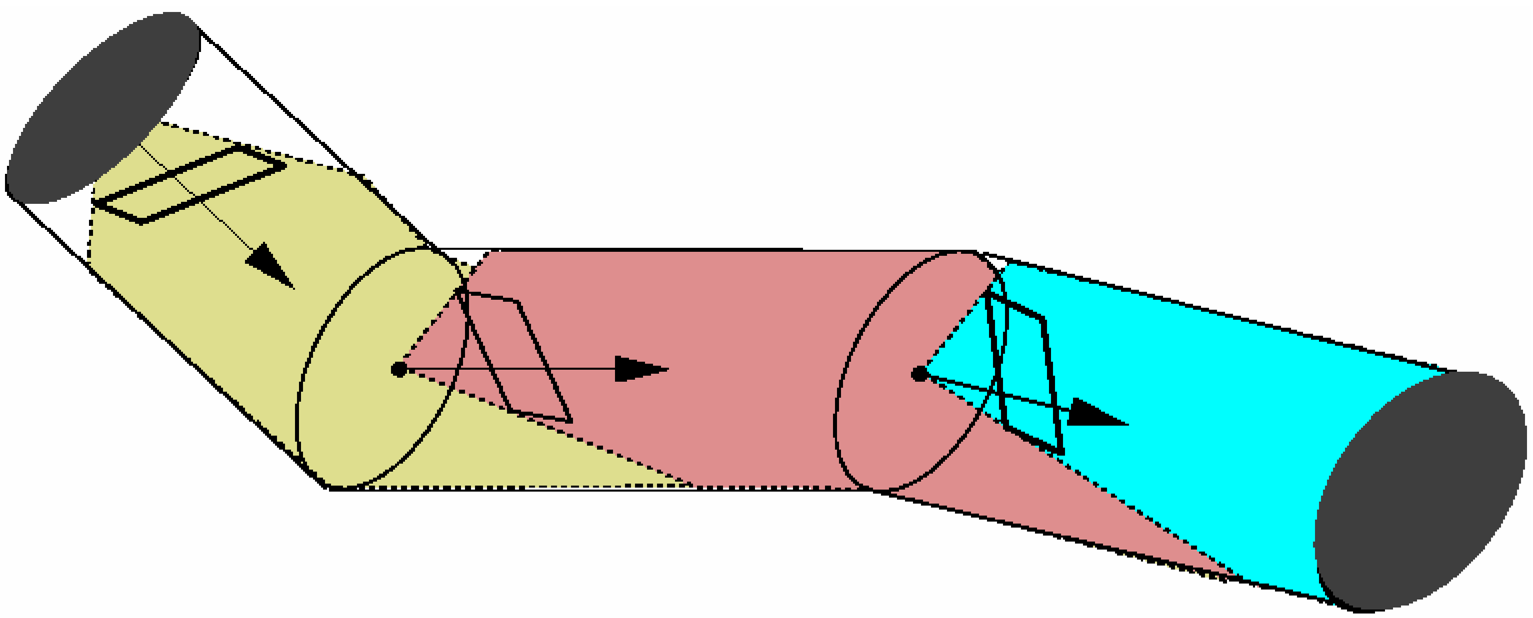


One Image



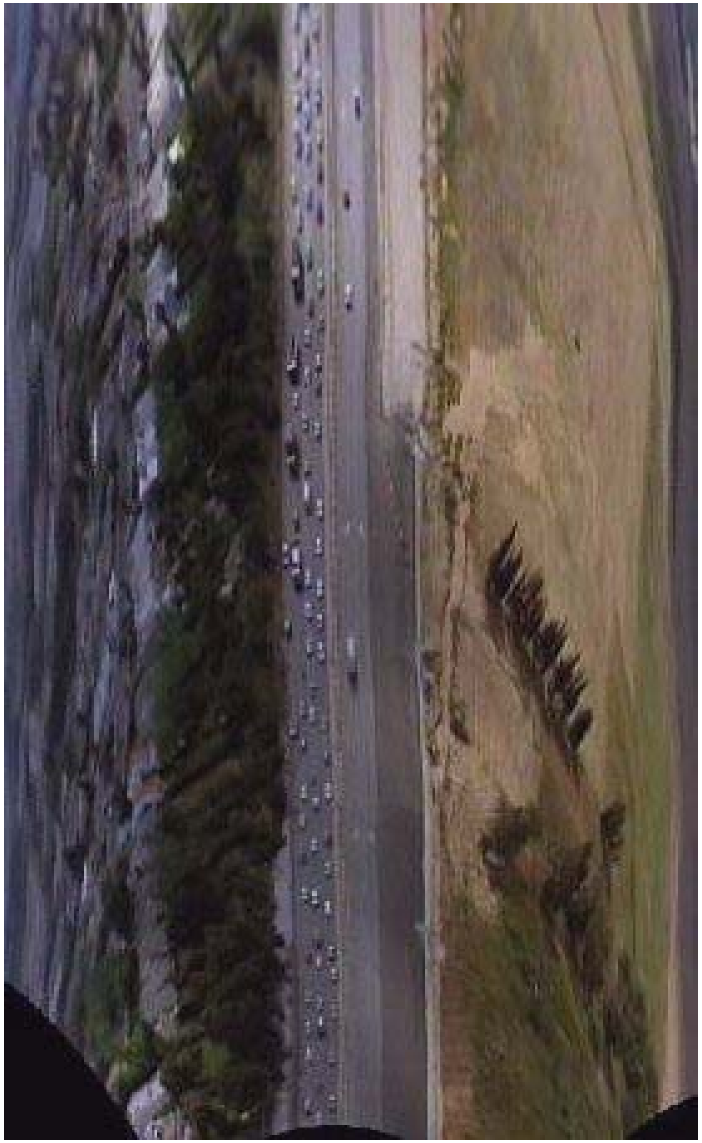
Sequence

Concatenation of Pipes



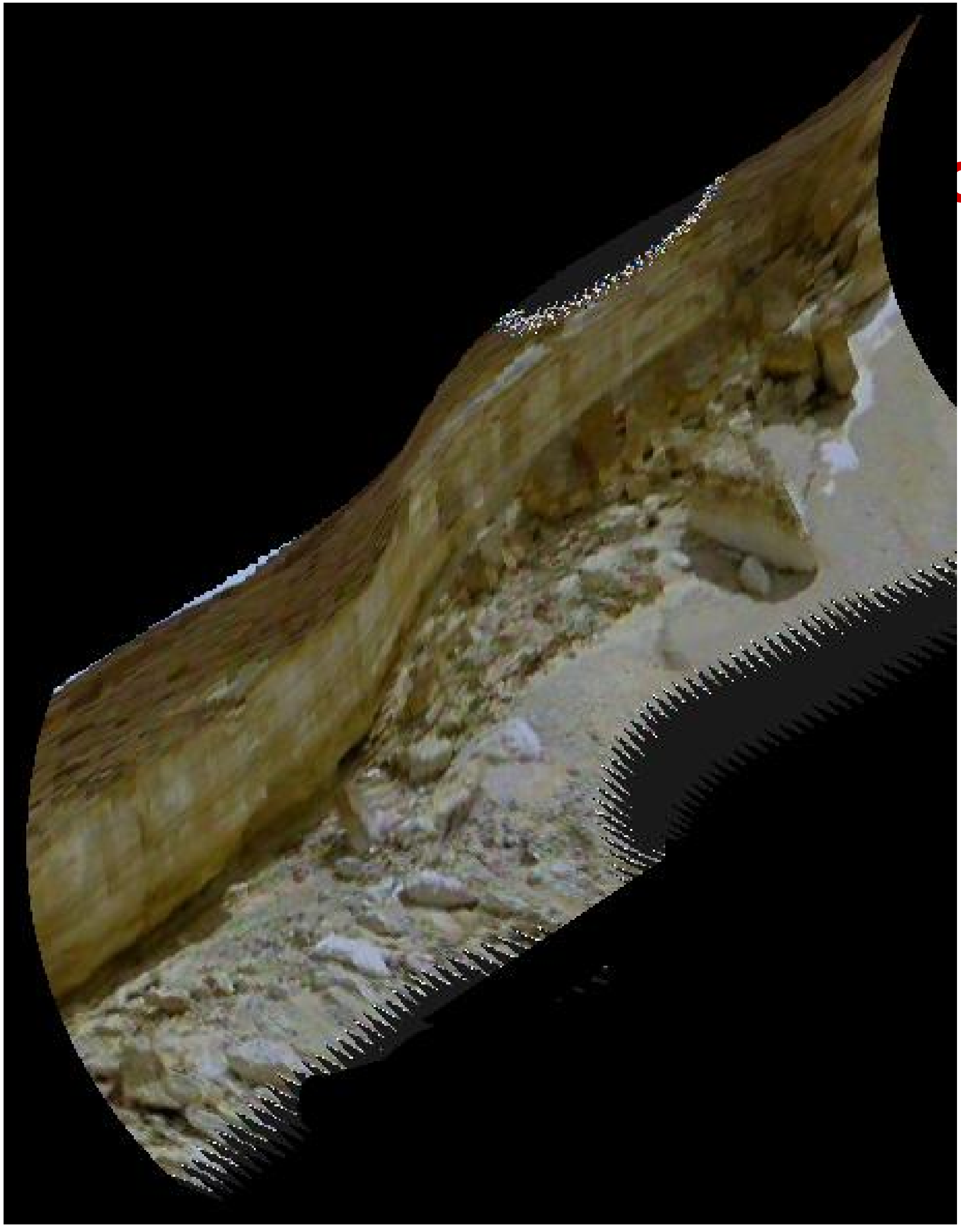
Forward Motion Mosaicing

Example: Forward Motion



Side View of Mosaic

orward Mosaicing II



Mosaic Construction

OmniStereo: Stereo in Full 360^{o}

*Two **Panoramas: One for Each Eye***

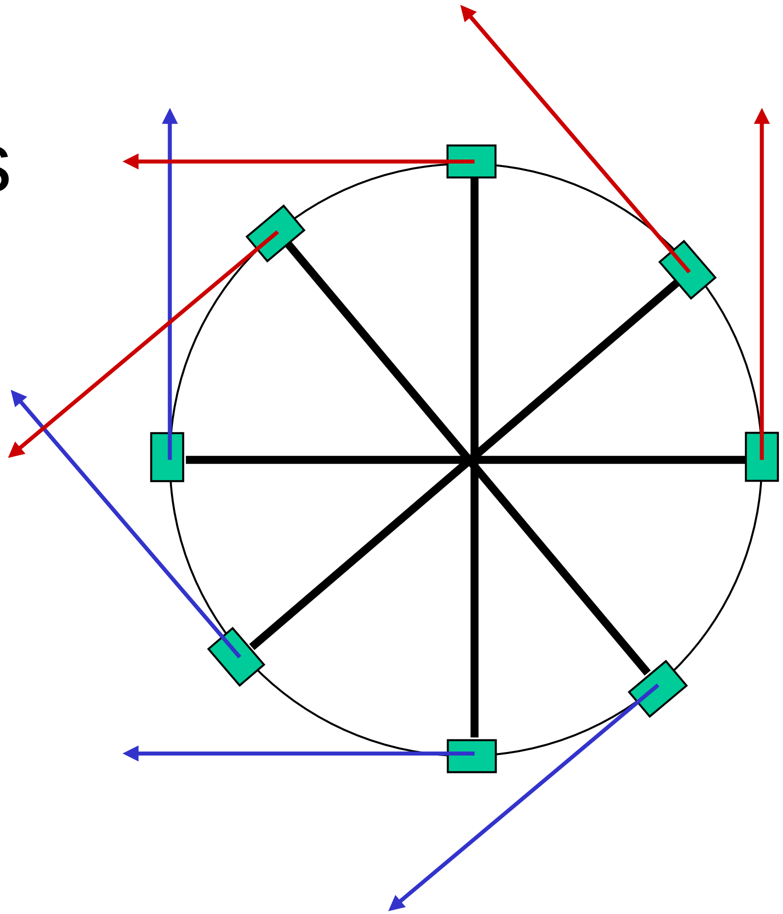
Each panorama can be mapped on a cylinder

Paradigm: A Rotating Stereo Pair of Slit Cameras

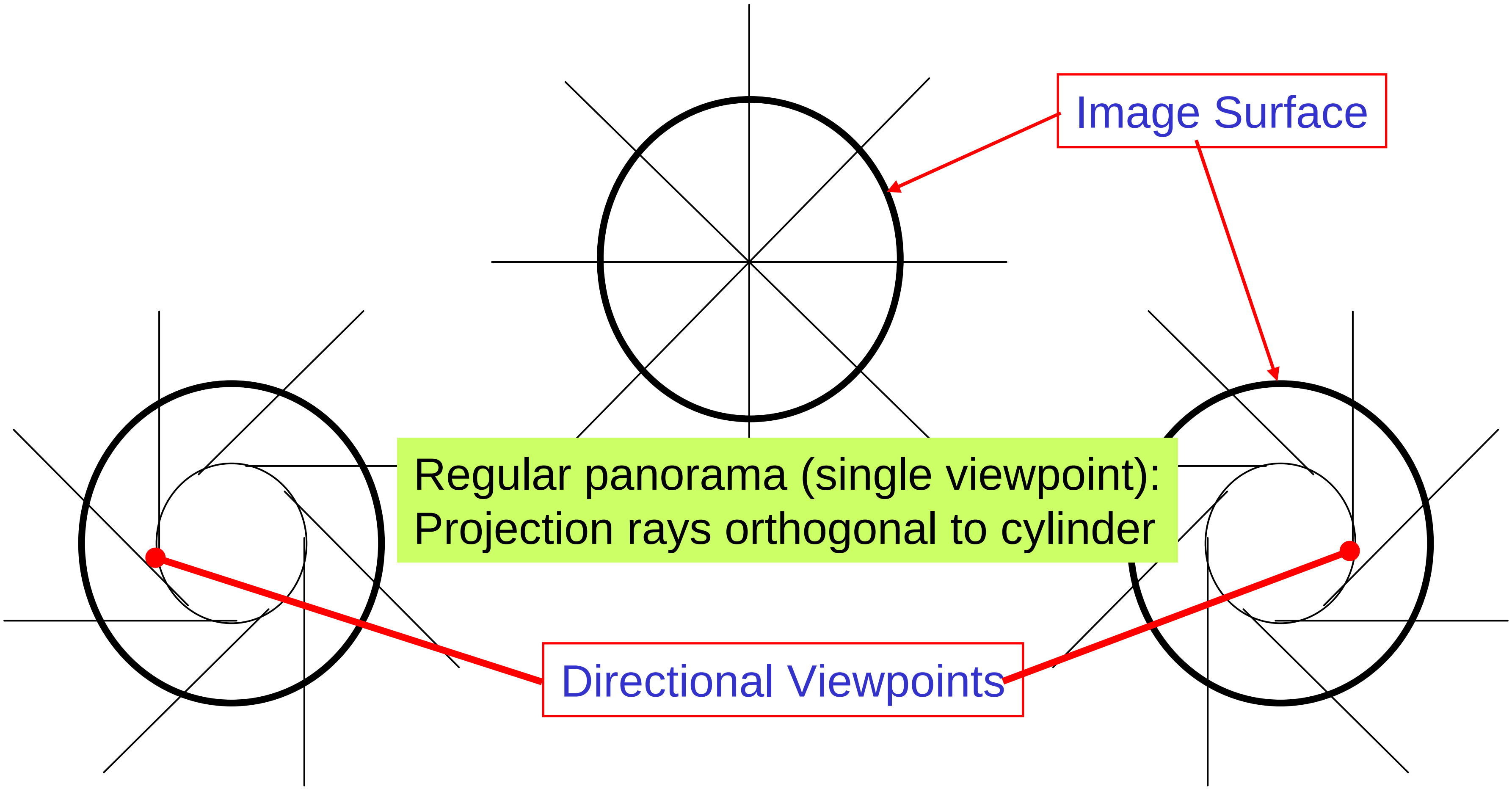
- Rays are tangent to ***viewing circle*** (Gives 360^{o} stereo)

- Image planes are radial

(Makes mosaicing difficult)



Panoramic Projections of Slit Cameras

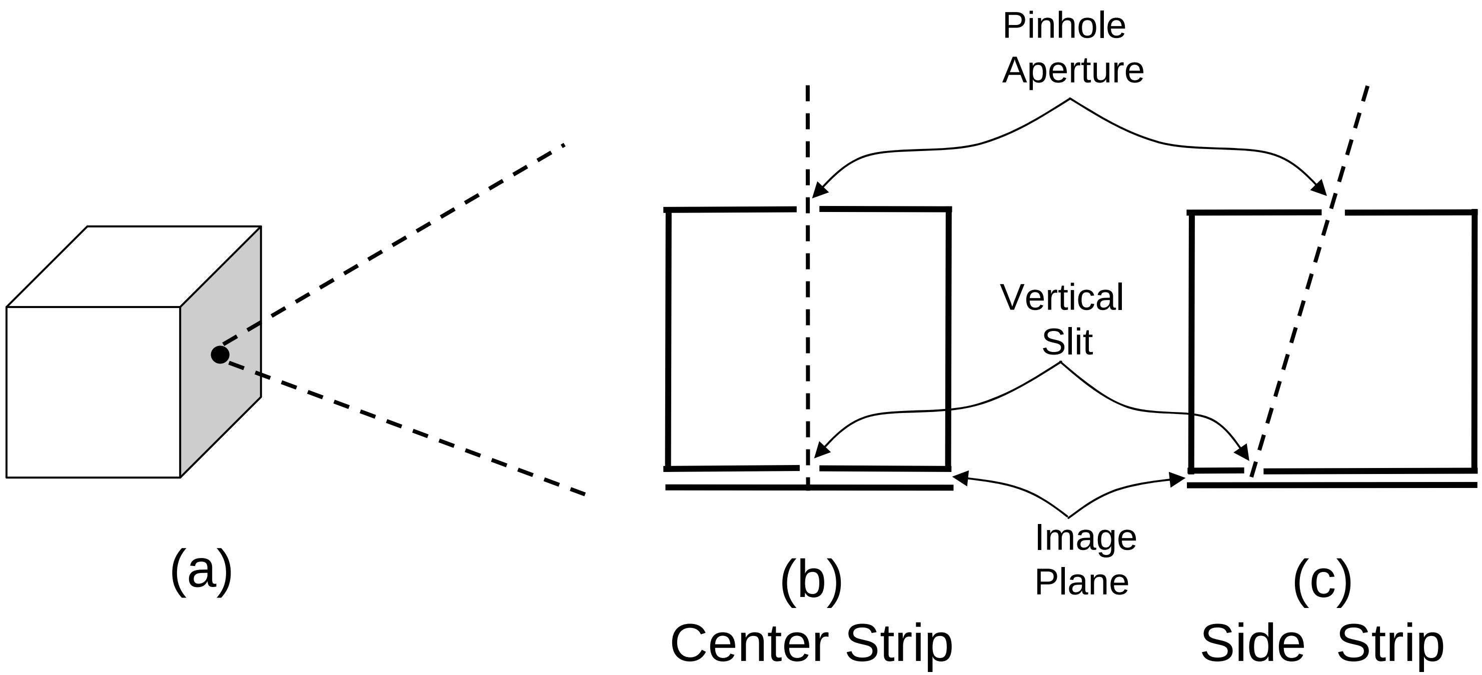


Left Eye panorama (viewing circle)
Projection rays tilted right

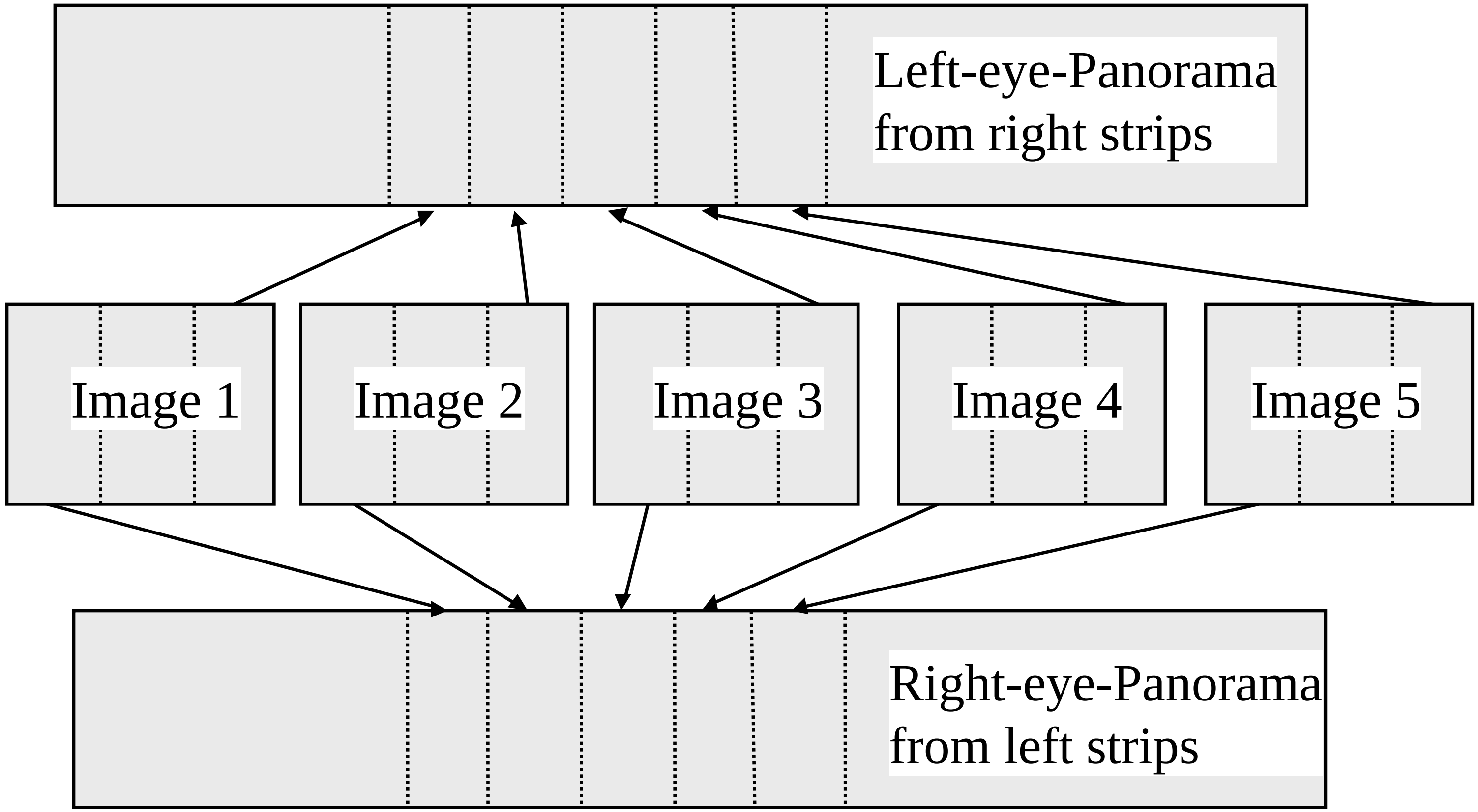
Right Eye panorama (viewing circle)
Projection rays tilted left

Slit Camera Model

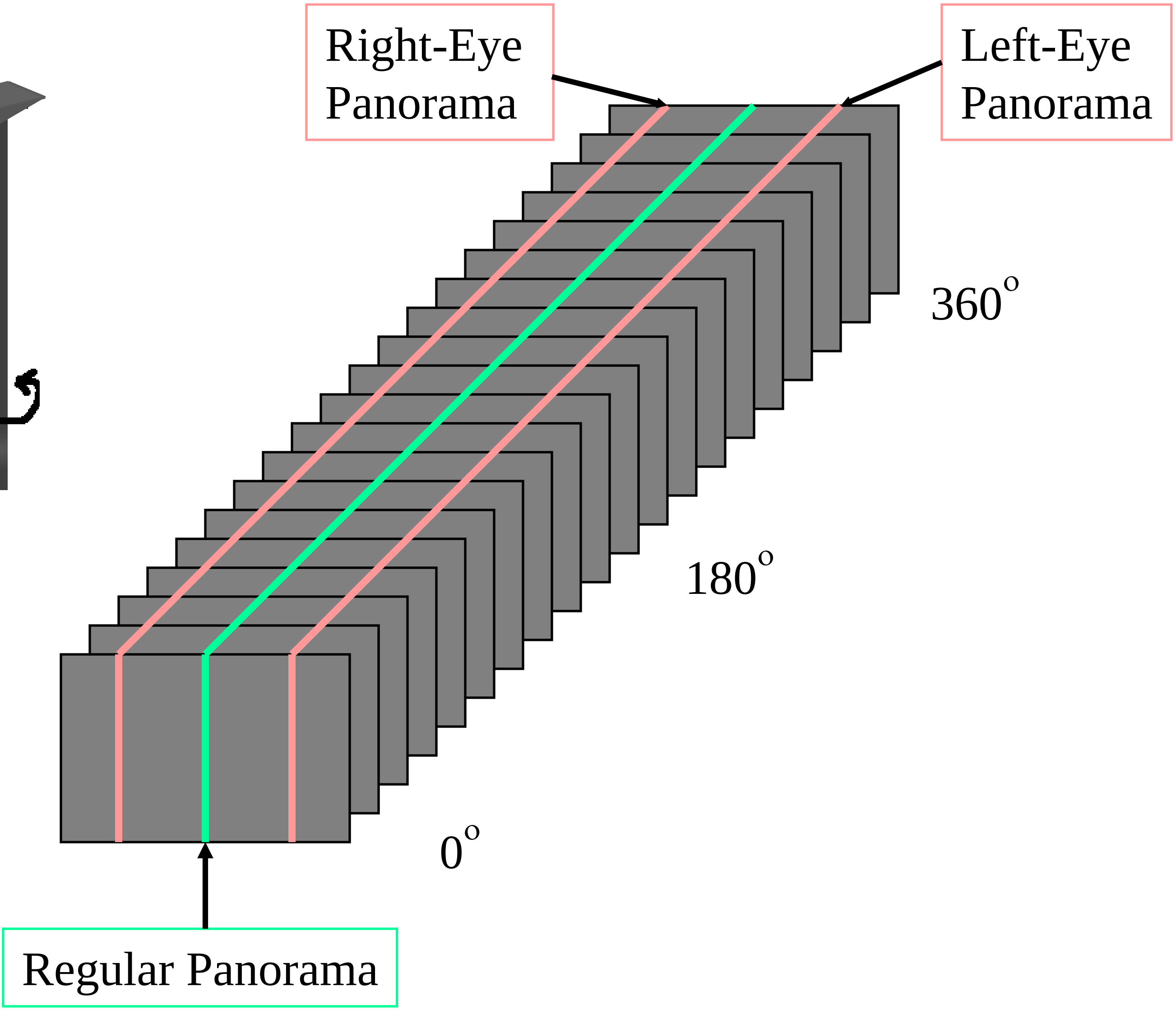
- Center Strip: Rays perpendicular to image plane
- Side Strip: Rays tilted from image plane



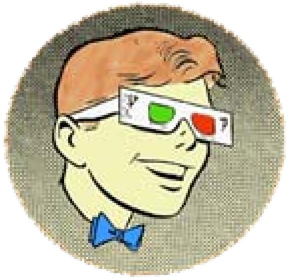
Stereo Panorama from Strips



MultiView Panoramas



Stereo Panorama from Video

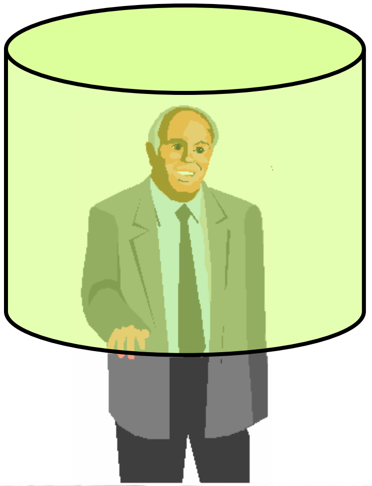


Stereo viewing with Red/Blue Glasses

Viewing Panoramic Stereo

Printed Cylindrical Surfaces

- Print panorama on a cylinder
- No computation needed!!!



The End