Visual Motion Estimation

Problems & Techniques

Princeton University
COS 429 Lecture

Oct. 2, 2008

Harpreet S. Sawhney
hsawhney@sarnoff.com

Outline

1. Visual motion in the Real World
2. The visual motion estimation problem
3. Problem formulation: Estimation through model-based alignment
4. Coarse-to-fine direct estimation of model parameters
5. Progressive complexity and robust model estimation
6. Multi-modal alignment
7. Direct estimation of parallax/depth/optical flow
8. Glimpses of some applications

Types of Visual Motion

in the

Real World

Simple Camera Motion : Pan & Tilt

Camera Does Not Change Location

Apparent Motion : Pan & Tilt

Camera Moves a Lot

Independent Object Motion

Objects are the Focus
Camera is more or less steady

Independent Object Motion
with
Camera Pan

Most common scenario
for
capturing performances

General Camera Motion

Large changes
in
camera location & orientation

Visual Motion due to Environmental Effects

Every pixel may have its own motion

The Works!

General Camera & Object Motions

Why is Analysis and Estimation of Visual Motion Important?

Visual Motion Estimation as a means of extracting Information Content in Dynamic Imagery

...extract information behind pixel data...

Foreground
Vs.
Background

Information Content in Dynamic Imagery

...extract information behind pixel data...

Foreground Vs. Background

Extended Scene Geometry

Information Content in Dynamic Imagery

...extract information behind pixel data...

Foreground
Vs.
Background

Temporal
Persistence

Extended Scene
Geometry

Layers & Mosaics

Segment,Track,Fingerprint
Moving Objects

Layers with 2D/3D
Scene Models

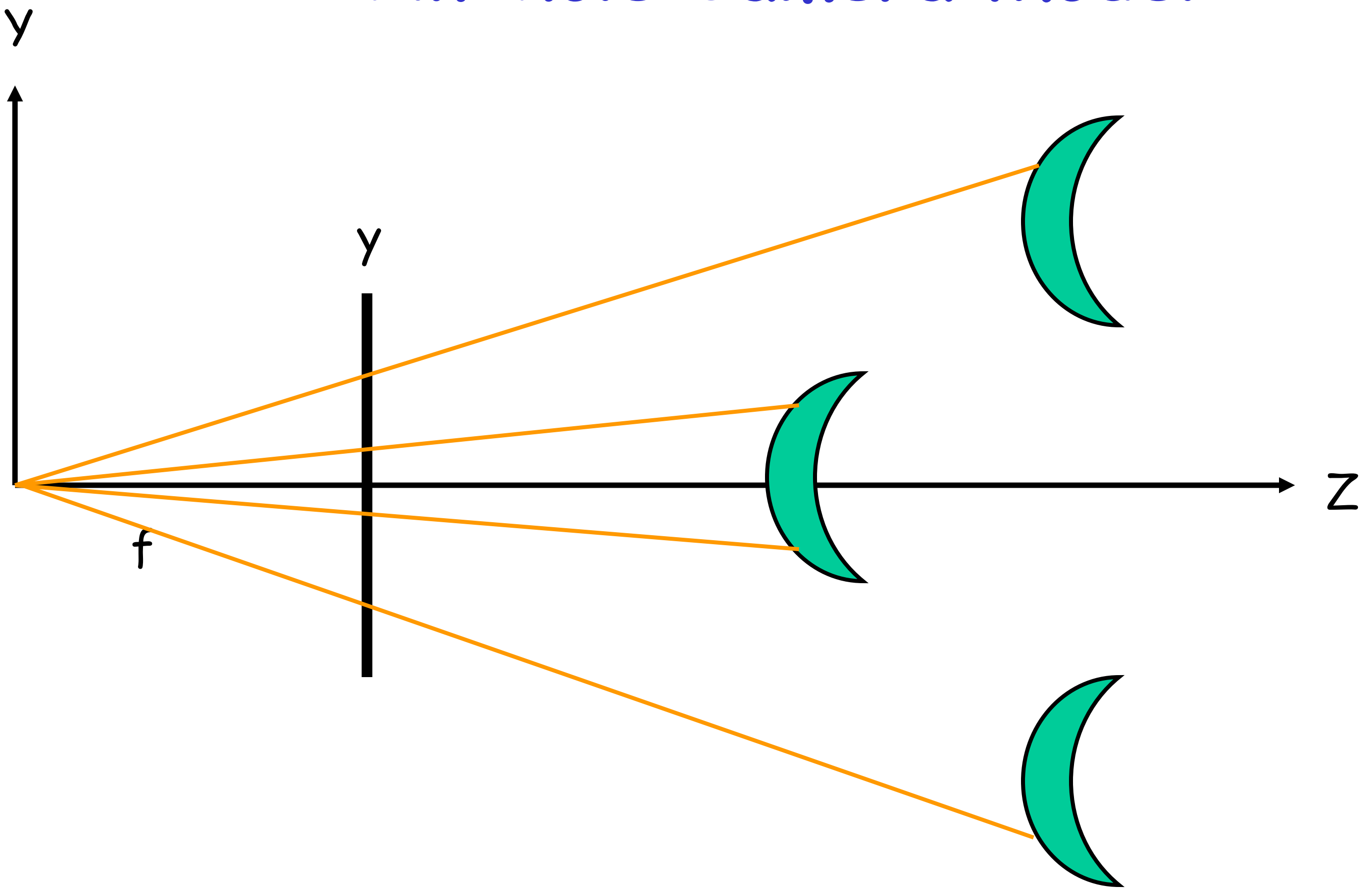
Layered, Motion, Structure & Appearance Analysis provides
Compact Representation for Manipulation & Recognition of Scene Content

An Example

A Panning Camera

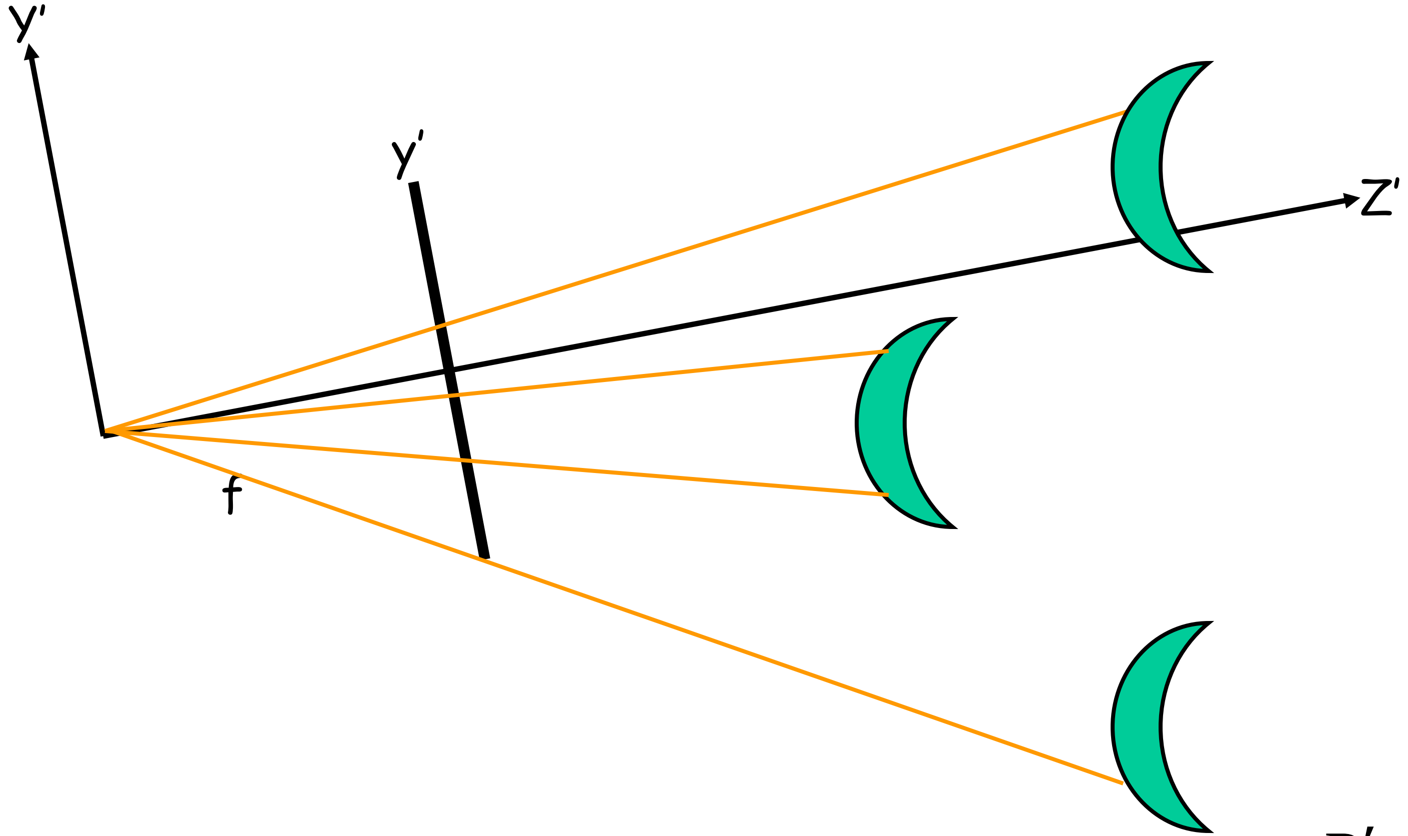
- Pin-hole camera model
- Pure rotation of the camera
- Multiple images related through a 2D projective transformation: also called a homography
- In the special case for camera pan, with small frame-to-frame rotation, and small field of view, the frames are related through a pure image translation

Pin-hole Camera Model



$$y = f\,\frac{Y}{Z} \qquad p \approx fP$$

Camera Rotation (Pan)



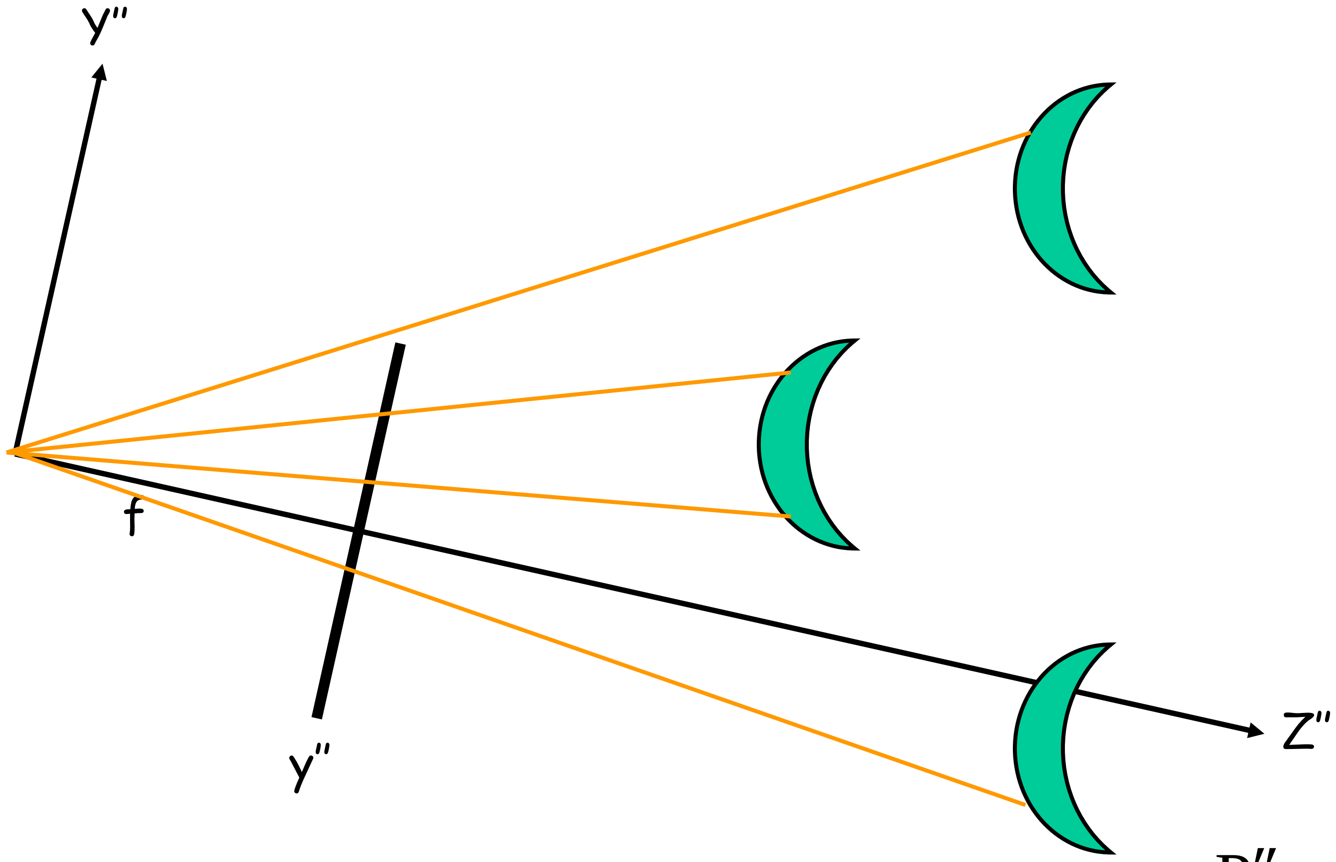
$$P' = R'P$$

$$y' = f\,\frac{Y'}{Z'}$$

$$p' \approx fP'$$

$$p' \approx R'p$$

Camera Rotation (Pan)



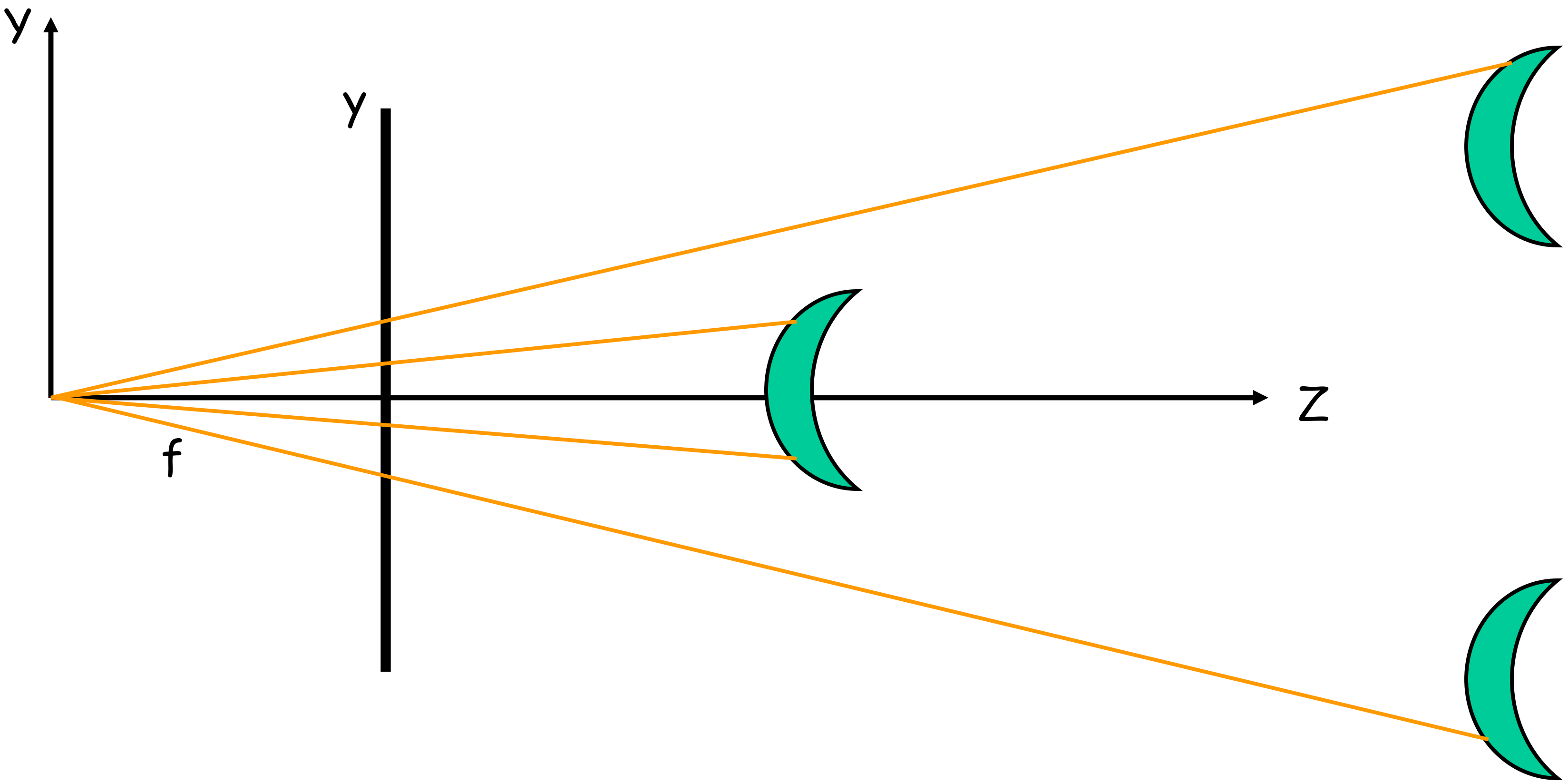
$$P'' = R''P$$

$$y'' = f\,\frac{Y''}{Z''} \qquad p'' \approx fP'' \qquad p'' \approx R''p$$

Image Motion
due to
Rotations
does not depend
on the
depth / structure of the scene

Verify the same for a 3D scene and 2D camera

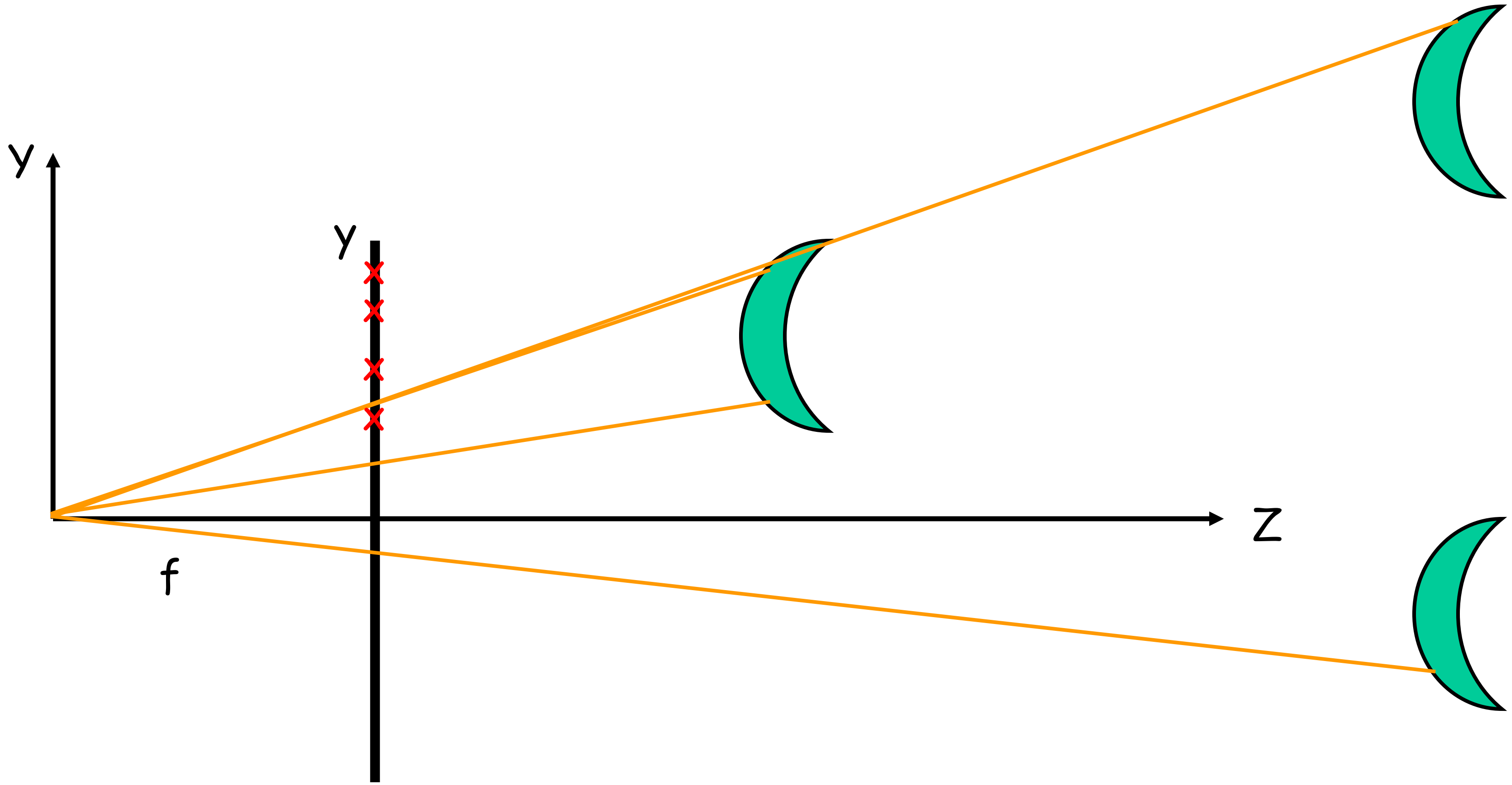
Pin-hole Camera Model



$$y = f\,\frac{Y}{Z}$$

$$p \approx fP$$

Camera Translation (Ty)



$$y' = f\,\frac{Y'}{Z'}$$

$$p' \approx fP'$$

$$P' = P + T'$$

Translational Displacement

$$y' = f\,\frac{Y'}{Z'}$$

$$y' = f\,\frac{Y + Ty}{Z}$$

$$y' - y = f\,\frac{Ty}{Z}$$

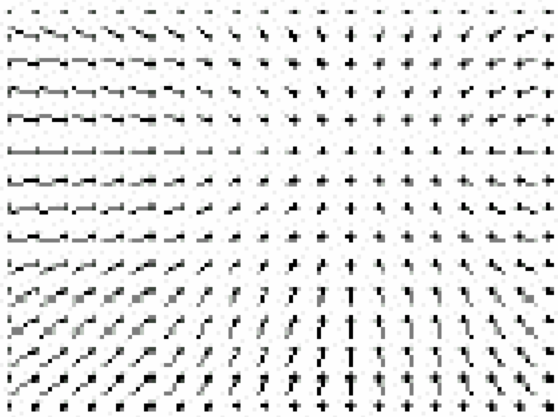
$$y' = f\,\frac{Y'}{Z'}$$

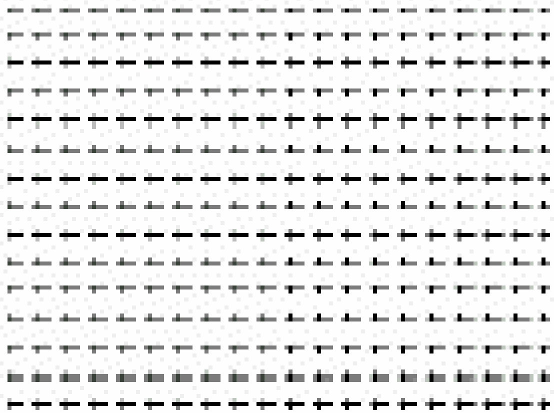
$$y' = f\,\frac{Y}{Z + Tz}$$

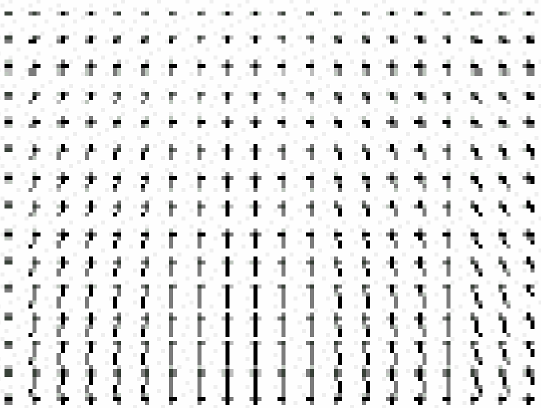
$$y' - y = -y'\,\frac{Tz}{Z}$$

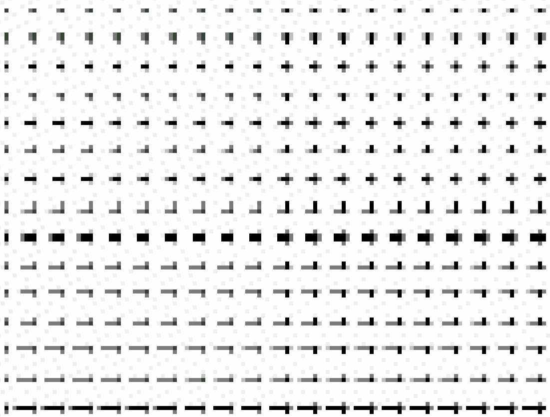
Image Motion due to Translation
is a function of
the depth of the scene

Cannonical Optic Flow Fields







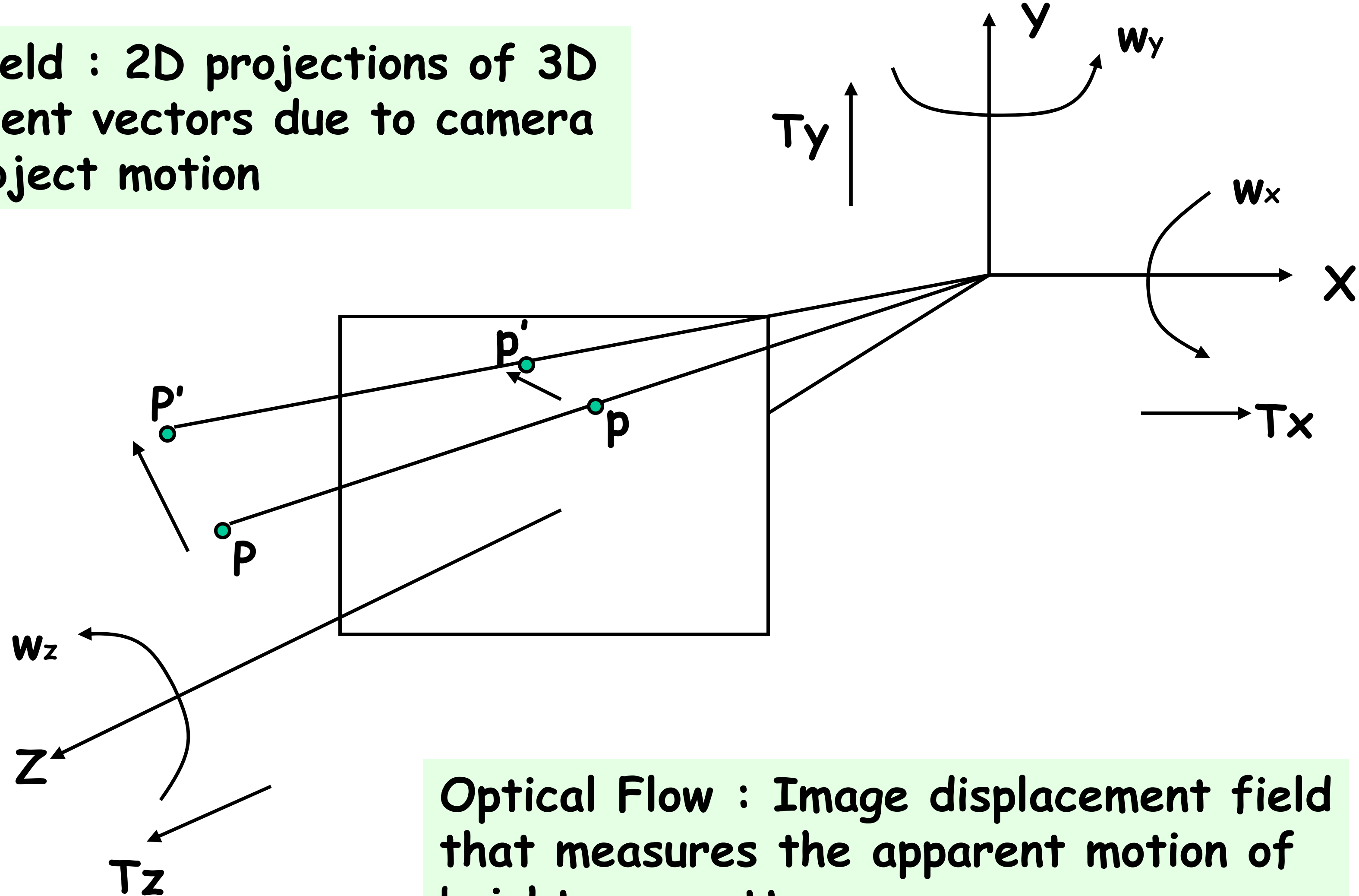


Sample Displacement Fields

Render scenes with various motions and plot the displacement fields

Motion Field vs. Optical Flow

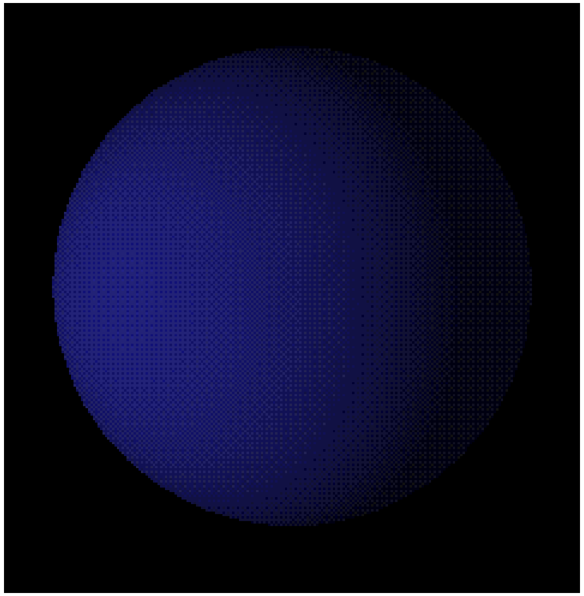
Motion Field : 2D projections of 3D displacement vectors due to camera and/or object motion



Optical Flow : Image displacement field that measures the apparent motion of brightness patterns

Motion Field vs. Optical Flow

Lambertian ball rotating in 3D



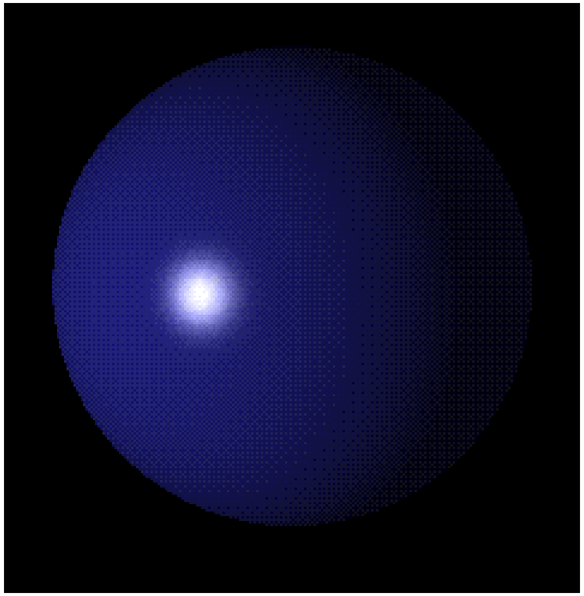
Motion Field ?

Optical Flow ?

Courtesy : Michael Black @ Brown.edu
Image: http://www.evl.uic.edu/aej/488/

Motion Field vs. Optical Flow

Stationary Lambertian ball with a moving point light source



Motion Field ?

Optical Flow ?

Courtesy : Michael Black @ Brown.edu
Image : http://www.evl.uic.edu/aej/488/

A Hierarchy of Models

Taxonomy by Bergen,Anandan et al.'92

- Parametric motion models
  - 2D translation, affine, projective, 3D pose *[Bergen, Anandan, et.al.'92]*
- Piecewise parametric motion models
  - 2D parametric motion/structure layers *[Wang&Adelson'93, Ayer&Sawhney'95]*
- Quasi-parametric
  - 3D R, T & depth per pixel. *[Hanna&Okumoto'91]*
  - Plane+parallax *[Kumar et.al.'94, Sawhney'94]*
- Piecewise quasi-parametric motion models
  - 2D parametric layers + parallax per layer *[Baker et al.'98]*
- Non-parametric
  - Optic flow: 2D vector per pixel *[Lucas&Kanade'81, Bergen,Anandan et.al.'92]*

Sparse/Discrete Correspondences

&

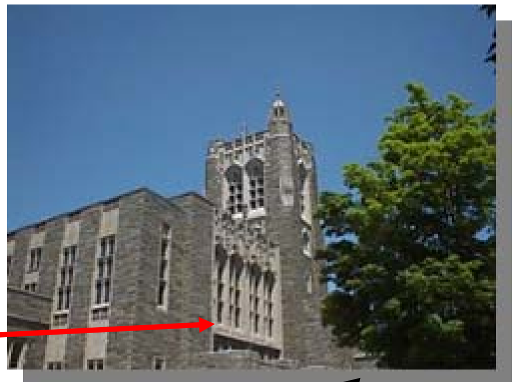
Dense Motion Estimation

Discrete Methods

Feature Correlation
&
RANSAC

Visual Motion through Discrete Correspondences

p'



p

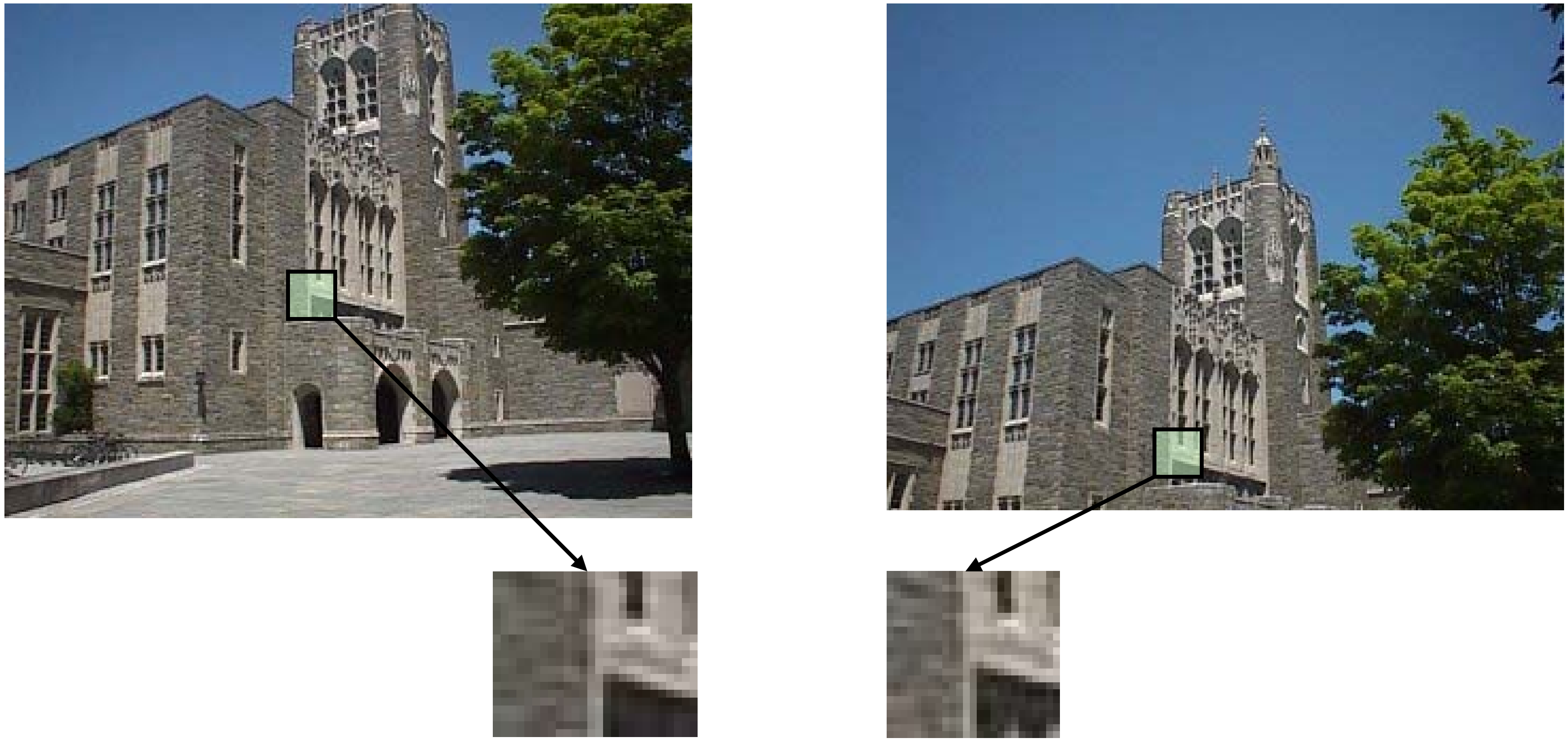
Images may be separated by time, space, sensor types

In general, discrete correspondences are related through a transformation

Discrete Methods

Feature Correlation
&
RANSAC

Discrete Correspondences



- Select corner-like points
- Match patches using Normalized Correlation
- Establish further matches using motion model

Direct Methods for Visual Motion Estimation

Employ Models of Motion
and
Estimate Visual Motion
through
Image Alignment

Characterizing Direct Methods

The What

- Visual interpretation/modeling involves spatio-temporal image representations directly
  - Not explicitly represented discrete features like corners, edges and lines etc.
- Spatio-temporal images are represented as outputs of symmetric or oriented filters.
- The output representations are typically dense, that is every pixel is explained,
  - Optical flow, depth maps.
  - Model parameters are also computed.

Direct Methods : The How

Alignment of spatio-temporal images is a means of obtaining :
Dense Representations, Parametric Models

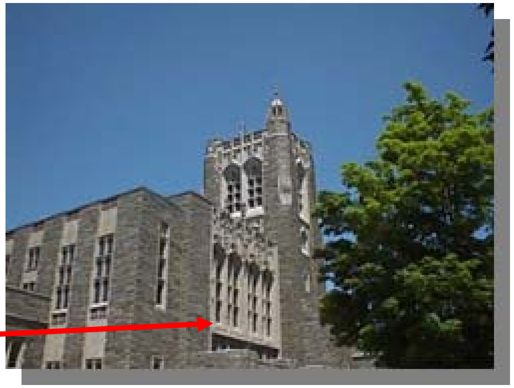
Direct Method based Alignment

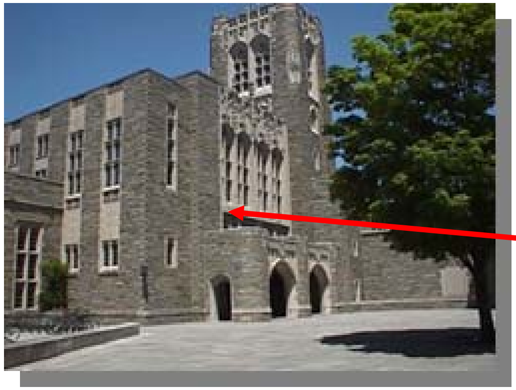
Formulation of Direct Model-based Image Alignment

[Bergen, Anandan et al.'92]

$\mathbf{I_1(p')}$

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





$\mathbf{I_2(p)}$

$\mathbf{p}$

Model image transformation as :

$$I_2(p) = I_1(p - u(p;\Theta)) = I_1(p')$$

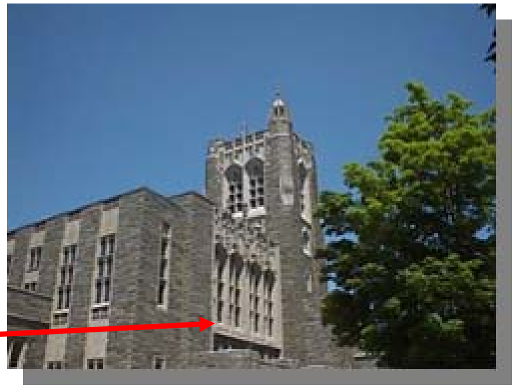
Brightness Constancy

Images separated by
time, space,
sensor types

Formulation of Direct Model-based Image Alignment

$\mathbf{I_1(p')}$

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



$\mathbf{I_2(p)}$

$\mathbf{p}$

Model image transformation as :

$$\mathbf{I_2(p) = I_1(p - u(p;\Theta))}$$

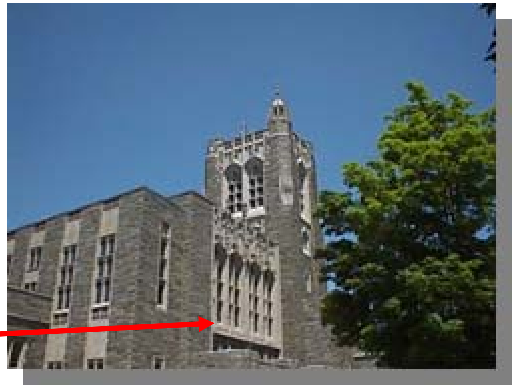
Images separated by
time, space,
sensor types

Reference Coordinate System

Formulation of Direct Model-based Image Alignment

$\mathbf{I_1(p')}$

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



$\mathbf{I_2(p)}$

$\mathbf{p}$

Model image transformation as :

$$\mathbf{I_2(p) = I_1(p - u(p;\Theta))}$$

Images separated by time, space, sensor types

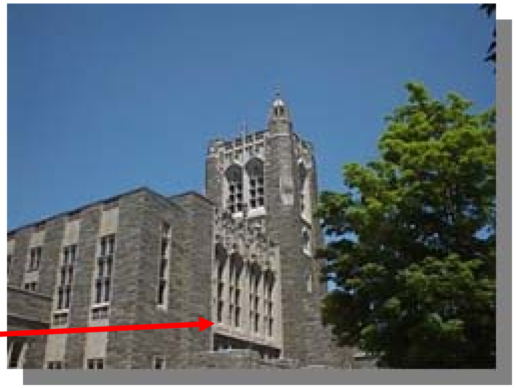
Reference Coordinate System

Generalized pixel Displacement

Formulation of Direct Model-based Image Alignment

$I_1(p')$

$p - u(p)$



$I_2(p)$

p

Model image transformation as :

$$I_2(p) = I_1(p - u(p;\Theta))$$

Images separated by time, space, sensor types

Reference Coordinate System

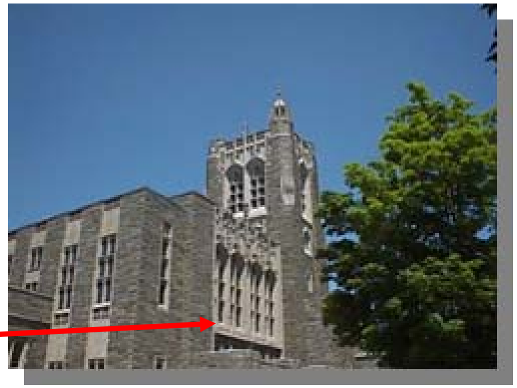
Generalized pixel Displacement

Model Parameters

Formulation of Direct Model-based Image Alignment

$\mathbf{I}_1(\mathbf{p}')$

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



$\mathbf{I}_2(\mathbf{p})$

$\mathbf{p}$

Model image transformation as :

$$\mathbf{I}_2(\mathbf{p}) = \mathbf{I}_1(\mathbf{p} - \mathbf{u}(\mathbf{p};\Theta))$$

Images separated by time, space, sensor types

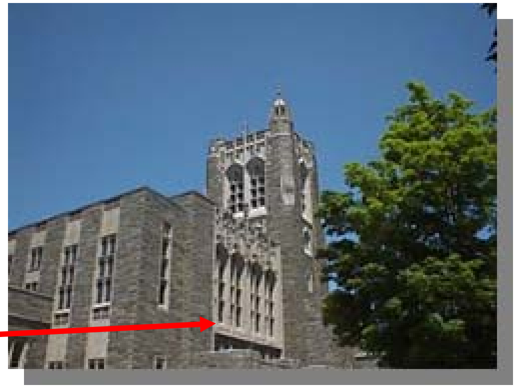
Reference Coordinate System

Generalized pixel Displacement

Model Parameters

Formulation of Direct Model-based Image Alignment

$\mathbf{I_1(p')}$



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

$\mathbf{I_2(p)}$

$\mathbf{p}$

Compute the unknown parameters and correspondences while aligning images using optimization :

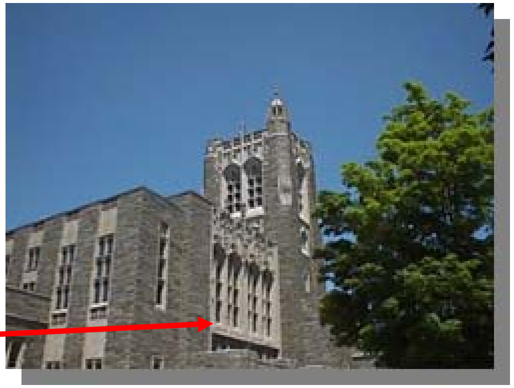
$$\min_{\Theta} \sum_i \rho(r_i;\sigma), \qquad r_i = I_2(p_i) - I_1(p_i - u(p_i;\Theta))$$

Filtered Image Representations (to account for Illumination changes, Multi-modalities)

What all can be varied ?

Formulation of Direct Model-based Image Alignment

$I_1(p')$



$I_2(p)$

$p - u(p)$

p

Compute the unknown parameters and correspondences while aligning images using optimization :

$$\min_{\Theta} \sum_i \rho(r_i;\sigma), \qquad r_i = I_2(p_i) - I_1(p_i - u(p_i;\Theta))$$

Model Parameters

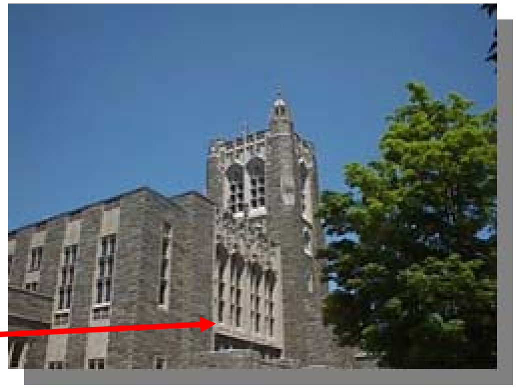
Filtered Image Representations (to account for Illumination changes, Multi-modalities)

What all can be varied ?

Formulation of Direct Model-based Image Alignment

$I_1(p')$

$p - u(p)$



$I_2(p)$

p

Compute the unknown parameters and correspondences while aligning images using optimization :

$$\min_{\Theta} \sum_{i} \rho(r_i;\sigma), \qquad r_i = I_2(p_i) - I_1(p_i - u(p_i;\Theta))$$

Measuring mismatches (SSD, Correlations)

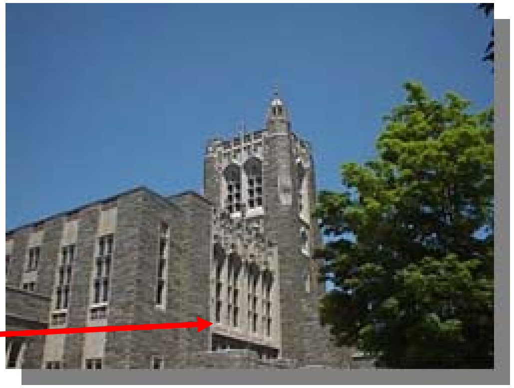
Model Parameters

Filtered Image Representations (to account for Illumination changes, Multi-modalities)

What all can be varied ?

Formulation of Direct Model-based Image Alignment

$I_1(p')$



$I_2(p)$

$p - u(p)$

p

Compute the unknown parameters and correspondences while aligning images using optimization :

$$\min_{\Theta} \sum_i \rho(r_i;\sigma), \qquad r_i = I_2(p_i) - I_1(p_i - u(p_i;\Theta))$$

Optimization Function

Measuring mismatches (SSD, Correlations)

Model Parameters

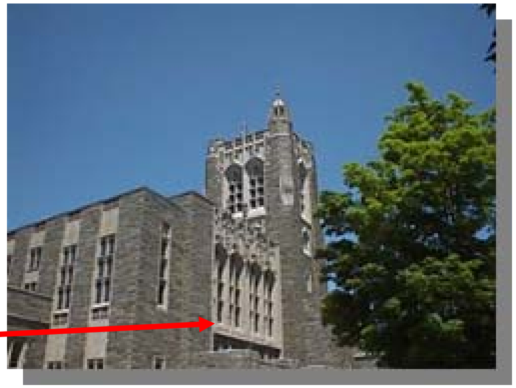
Filtered Image Representations (to account for Illumination changes, Multi-modalities)

What all can be varied ?

Formulation of Direct Model-based Image Alignment

$I_1(p')$

$p - u(p)$



$I_2(p)$

p

Compute the unknown parameters and correspondences while aligning images using optimization :

$$\min_{\Theta} \sum_{i} \rho(r_i;\sigma), \qquad r_i = I_2(p_i) - I_1(p_i - u(p_i;\Theta))$$

Optimization Function

Measuring mismatches (SSD, Correlations)

Model Parameters

Filtered Image Representations (to account for Illumination changes, Multi-modalities)

What all can be varied ?

A Hierarchy of Models

Taxonomy by Bergen,Anandan et al.'92

- Parametric motion models
  - 2D translation, affine, projective, 3D pose *[Bergen, Anandan, et.al.'92]*
- Piecewise parametric motion models
  - 2D parametric motion/structure layers *[Wang&Adelson'93, Ayer&Sawhney'95]*
- Quasi-parametric
  - 3D R, T & depth per pixel. *[Hanna&Okumoto'91]*
  - Plane+parallax *[Kumar et.al.'94, Sawhney'94]*
- Piecewise quasi-parametric motion models
  - 2D parametric layers + parallax per layer *[Baker et al.'98]*
- Non-parametric
  - Optic flow: 2D vector per pixel *[Lucas&Kanade'81, Bergen,Anandan et.al.'92]*

Plan : This Part

- First present the generic normal equations.
- Then specialize these for a projective transformation.
- Sidebar into backward image warping.
- SSD and M-estimators.

An Iterative Solution of Model Parameters

[Black&Anandan'94 Sawhney'95]

$$\min_{\Theta} \sum_{i} \rho(r_i;\sigma), \qquad r_i = I_2(p_i) - I_1(p_i - u(p_i;\Theta))$$

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

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

w_i

- w_i is a weight associated with each measurement.

An Iterative Solution of Model Parameters

$$\min_{\Theta} \sum_i \rho(r_i; \sigma), \qquad r_i = I_2(p_i) - I_1(p_i - u(p_i; \Theta))$$

- In particular for Sum-of-Square Differences : $\rho_{SSD} = \dfrac{r^2}{2\sigma^2}$

- We obtain the standard normal equations:

$$\sum_l \sum_i \frac{\partial r_i}{\partial \theta_k} \frac{\partial r_i}{\partial \theta_l} \partial \theta_l = -\sum_i r_i \frac{\partial r_i}{\partial \theta_k} \qquad \forall \mathbf{k}$$

- Other functions can be used for robust M-estimation…

How does this work for images ? (1)

$$\min_{\Theta} \sum_i \frac{1}{2} r_i^2, \qquad r_i = I_2(p_i) - I_1(\underbrace{p_i - u(p_i;\Theta)}_{p_i'})$$

- Let their be a 2D projective transformation between the two images:

$$\mathbf{p}_i' \approx \mathbf{P}\mathbf{p}_i$$

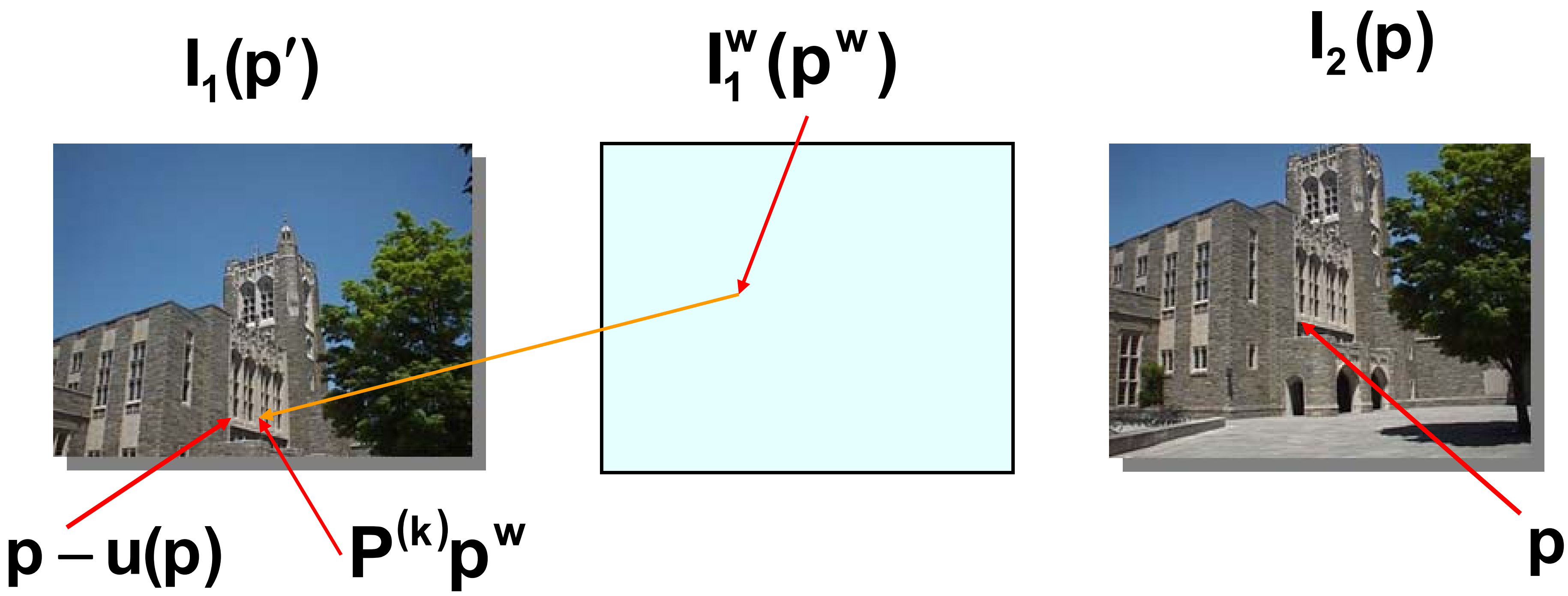
- Given an initial guess $\mathbf{P}^{(k)}$

- First, warp $I_1(p_i')$ towards $I_2(p_i)$

How does this work for images ? (2)

$$\min_{\Theta} \sum_i \frac{1}{2} r_i^2, \qquad r_i = I_2(p_i) - I_1(\underbrace{p_i - u(p_i;\Theta)}_{p_i'})$$

$$I_1^w(p^w) = I_1(p') = I_1(P^{(k)} p^w)$$



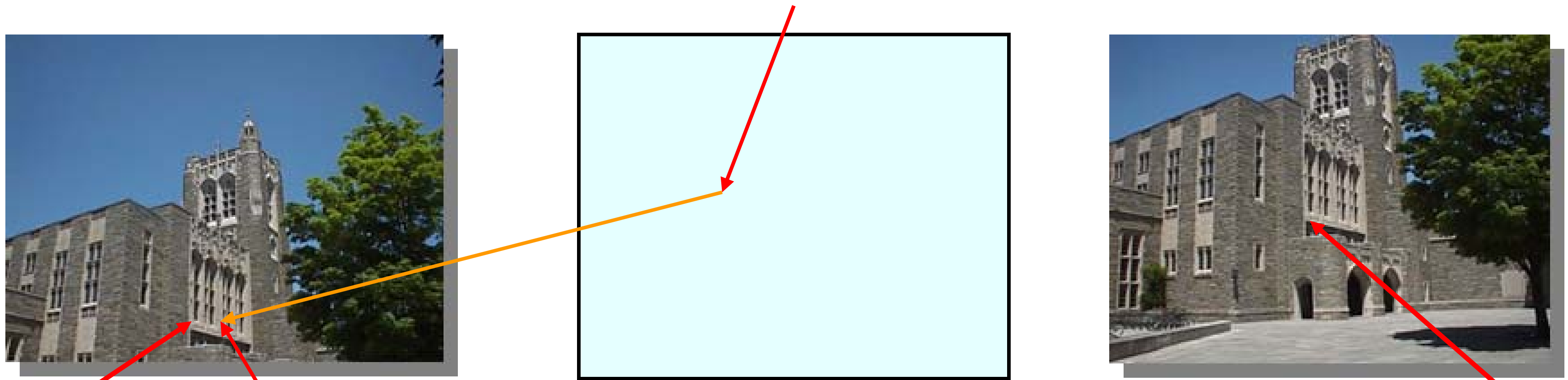
How does this work for images ? (3)

$$\min_{\Theta} \sum_i \frac{1}{2} r_i^2, \qquad r_i = I_2(p_i) - I_1(\underbrace{p_i - u(p_i;\Theta)}_{p_i'})$$

$$\mathbf{I}_1^{\mathbf{w}}(\mathbf{p}^{\mathbf{w}}) = \mathbf{I}_1(\mathbf{p}') = \mathbf{I}_1(\mathbf{P}^{(k)}\mathbf{p}^{\mathbf{w}})$$

$\mathbf{I}_1^{\mathbf{w}}(\mathbf{p}^{\mathbf{w}})$: Represents image 1 warped towards the reference image 2, Using the current set of parameters

$\mathbf{I}_1(\mathbf{p}')$ $\quad$ $\mathbf{I}_1^{\mathbf{w}}(\mathbf{p}^{\mathbf{w}})$ $\quad$ $\mathbf{I}_2(\mathbf{p})$



$\mathbf{p} - \mathbf{u}(\mathbf{p})$ $\quad$ $\mathbf{P}^{(k)}\mathbf{p}^{\mathbf{w}}$ $\quad$ $\mathbf{p}$

How does this work for images ? (4)

- The residual transformation between the warped image and the reference image is modeled as:

$$r_i = I_2(p_i) - I_1^w(p_i^w - \delta p_i^w(p_i^w; \delta\Theta))$$

Where $$p_i^w \approx [I + D] p_i$$

$$D = \begin{bmatrix} d11 & d12 & d13 \\ d21 & d22 & d23 \\ d31 & d32 & 0 \end{bmatrix}$$

How does this work for images ? (5)

- The residual transformation between the warped image and the reference image is modeled as:

$$r_i = I_2(p_i) - I_1^w(p_i^w - \delta p_i^w(p_i^w; D))$$

$$\approx I_2(p_i) - I^w(p_i^w(p_i; 0)) - \nabla I^{w^T} \frac{\partial p_i^w}{\partial d}\Big|_{D=0} d$$

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

How does this work for images ? (6)

$$\min_{\Theta} \sum_i \frac{1}{2} r_i^2 ,$$

$$r_i \approx \nabla \mathbf{I}_i^{\mathbf{w}^\mathsf{T}} \nabla_{\mathbf{D}} \mathbf{p}_i^{\mathbf{w}} \big|_{\mathbf{D}=0}\ \mathbf{d} - \boldsymbol{\delta} I(p_i)$$

$$\sum_i \nabla_{\mathbf{D}}{}^{\mathsf{T}} \mathbf{p}_i^{\mathbf{w}} \nabla \mathbf{I}_i^{\mathbf{w}^\mathsf{T}} \nabla \mathbf{I}_i^{\mathbf{w}^\mathsf{T}} \nabla_{\mathbf{D}} \mathbf{p}_i^{\mathbf{w}} \mathbf{d} = \sum_i \nabla_{\mathbf{D}}{}^{\mathsf{T}} \mathbf{p}_i^{\mathbf{w}} \boldsymbol{\delta} I(p_i)$$

$$\mathbf{Hd} = \mathbf{g}$$

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

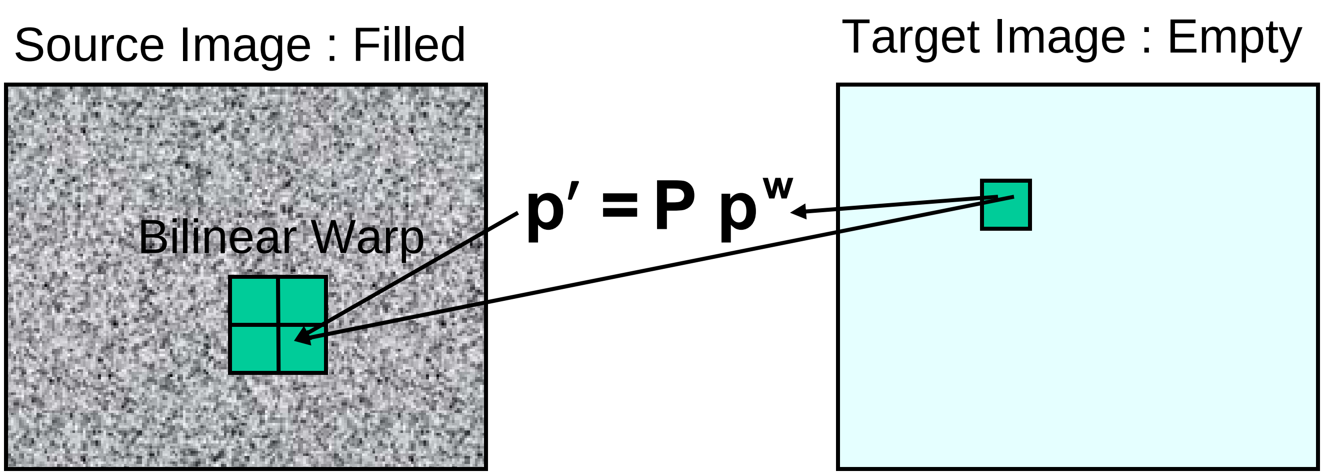
So now we can solve for the model parameters while aligning images iteratively using warping and Levenberg-Marquat style optimization

Sidebar : Backward Warping

- Note that we have used backward warping in the direct alignment of images.

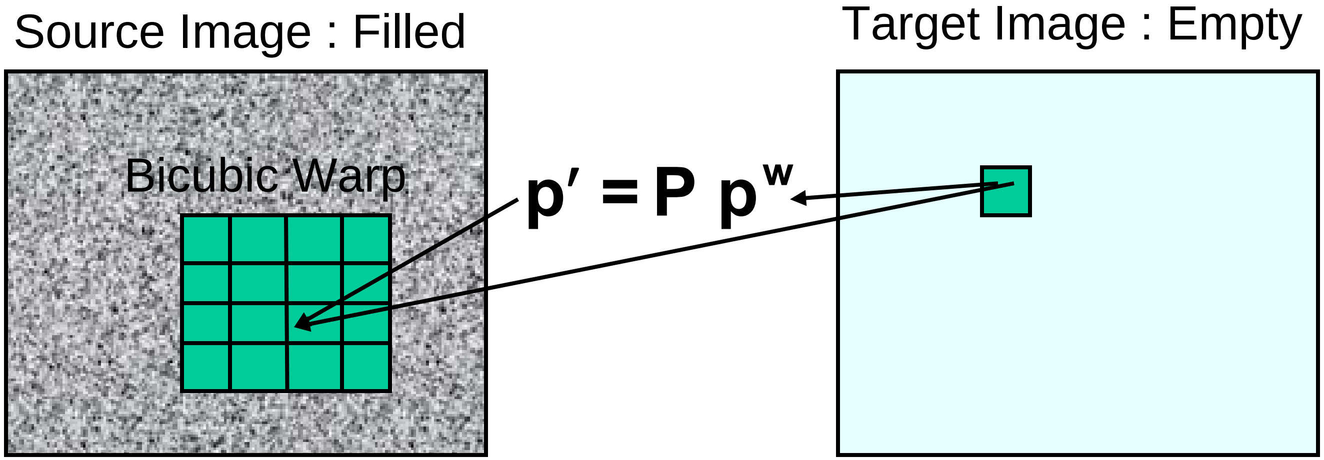
- Backward warping avoids holes.

- Image gradients are estimated in the warped coordinate system.

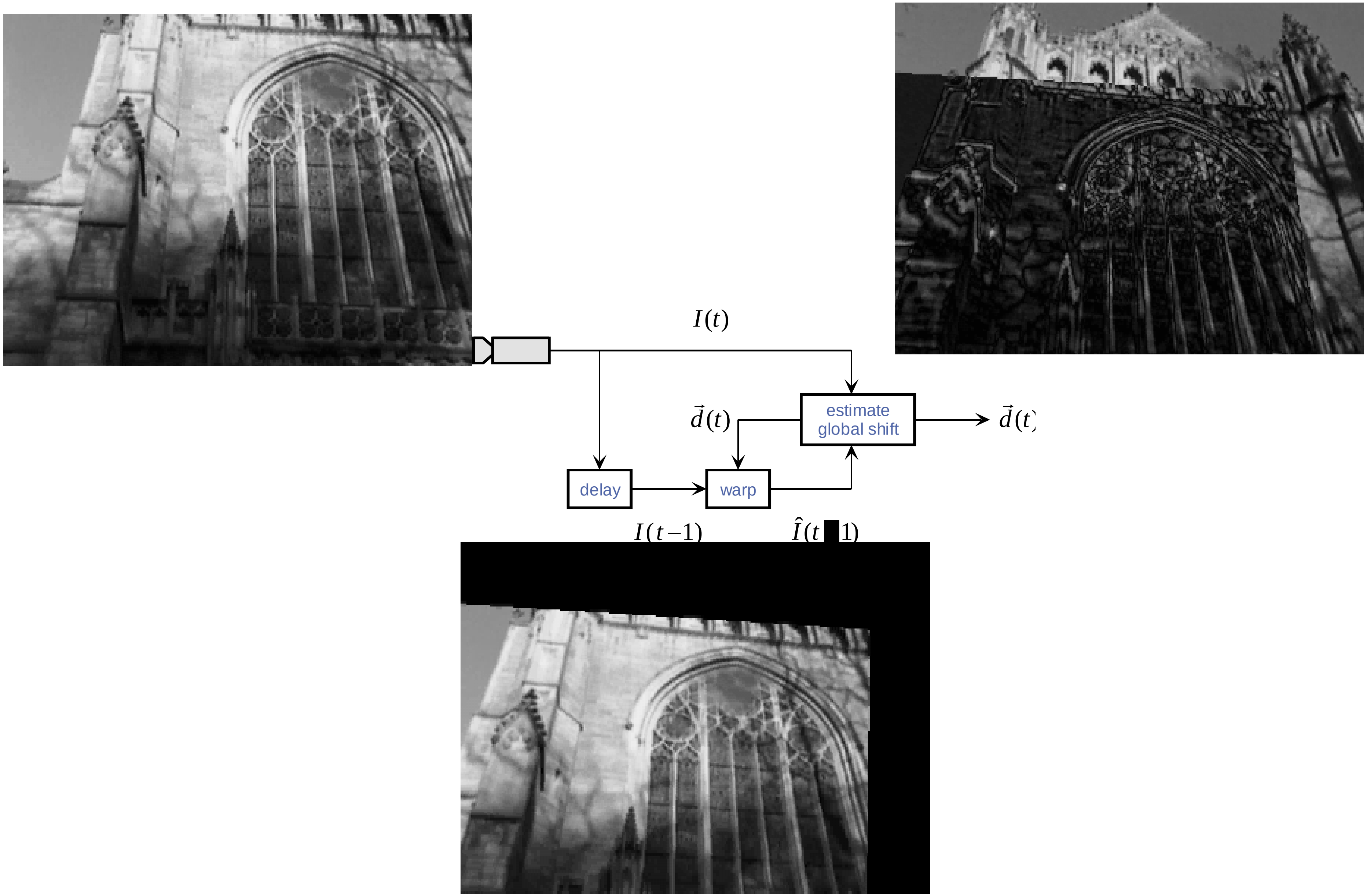


Sidebar : Backward Warping

- Note that we have used backward warping in the direct alignment of images.
- Backward warping avoids holes.
- Image gradients are estimated in the warped coordinate system.

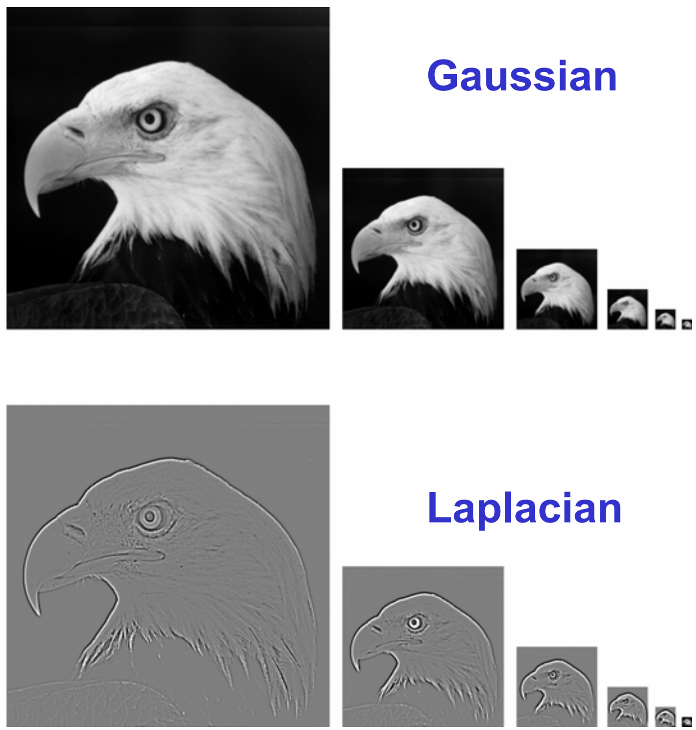


Iterative Alignment : Result



How to handle Large Transformations ?

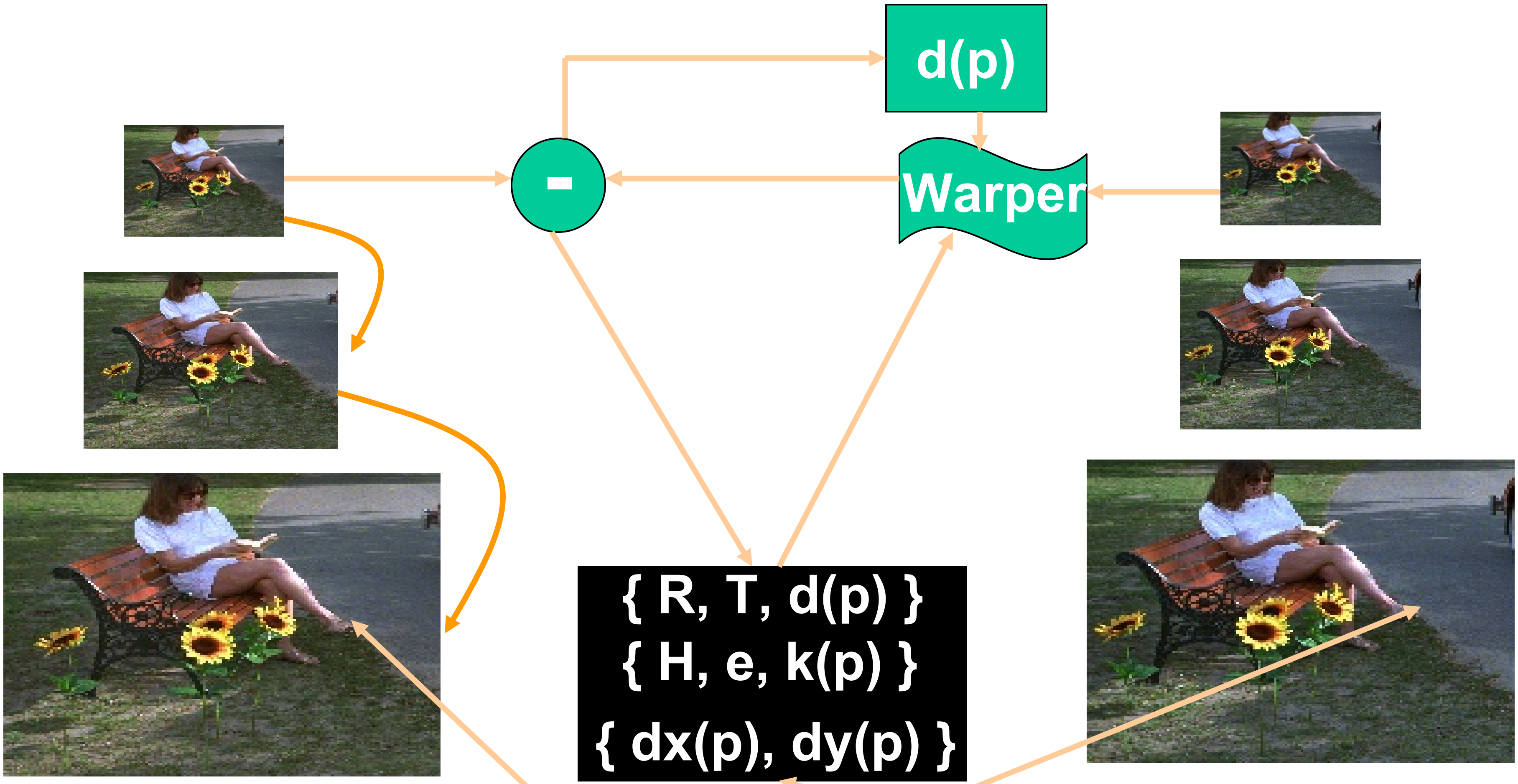
[Burt,Adelson'81]



Pyramid Processing

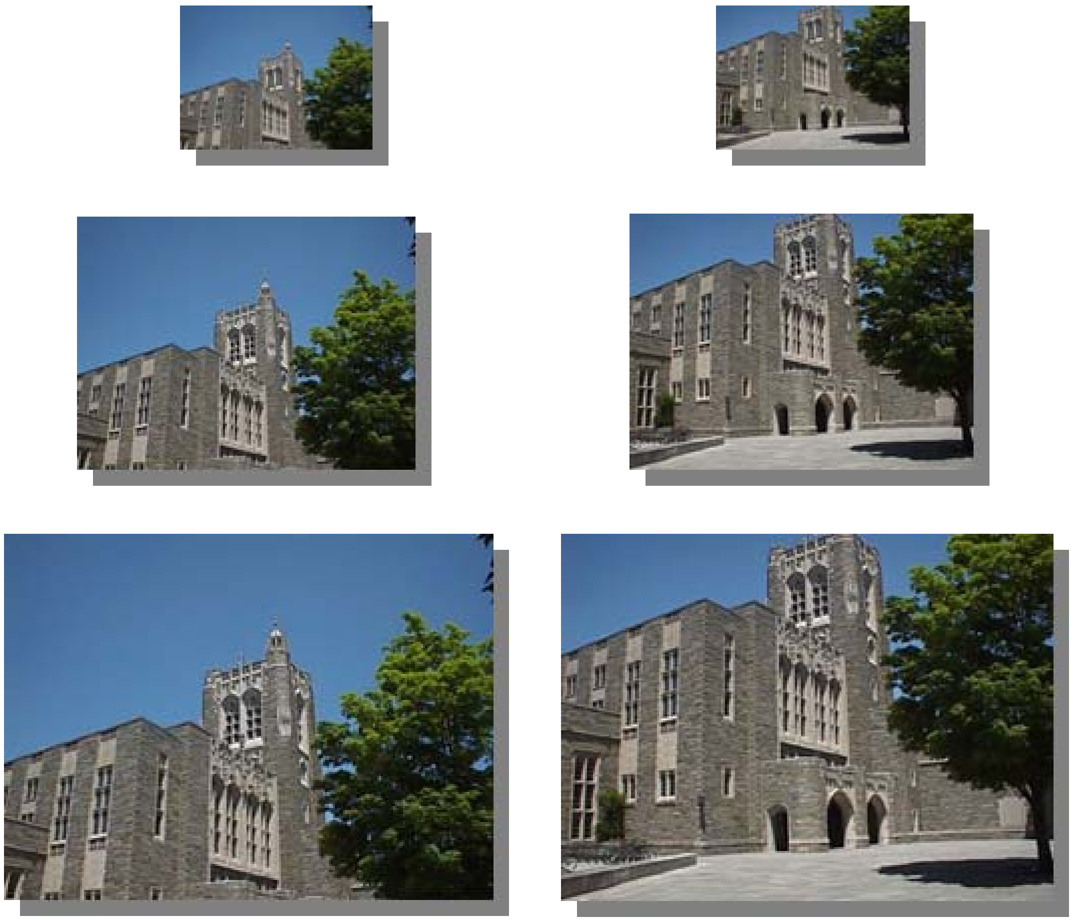
- A hierarchical framework for fast algorithms
- A wavelet representation for compression, enhancement, fusion
- A model of human vision

Iterative Coarse-to-fine Model-based Image Alignment



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

Pyramid-based Direct Image Alignment



- 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.

Application : Image/Video Mosaicing

- Direct frame-to-frame image alignment.
- Select frames to reduce the number of frames & overlap.
- Warp aligned images to a reference coordinate system.
- Create a single mosaic image.
- Assumes a parametric motion model.

Video Mosaic Example

VideoBrush'96

Princeton Chapel Video Sequence
54 frames

Unblended Chapel Mosaic

Image Mosaics

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

Output Mosaic

Handling Moving Objects in 2D Parametric Alignment & Mosaicing

Generalized M-Estimation

$$\min_{\Theta} \sum_i \rho(r_i;\sigma), \qquad r_i = I_2(p_i) - I_1(p_i - u(p_i;\Theta))$$

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

$$\sum_l \sum_i \frac{\dot{\rho}(r_i)}{r_i} \frac{\partial r_i}{\partial \theta_k} \frac{\partial r_i}{\partial \theta_l} \partial\theta_l = -\sum_i \frac{\dot{\rho}(r_i)}{r_i} r_i \frac{\partial r_i}{\partial \theta_k} \quad \forall k$$

w_i

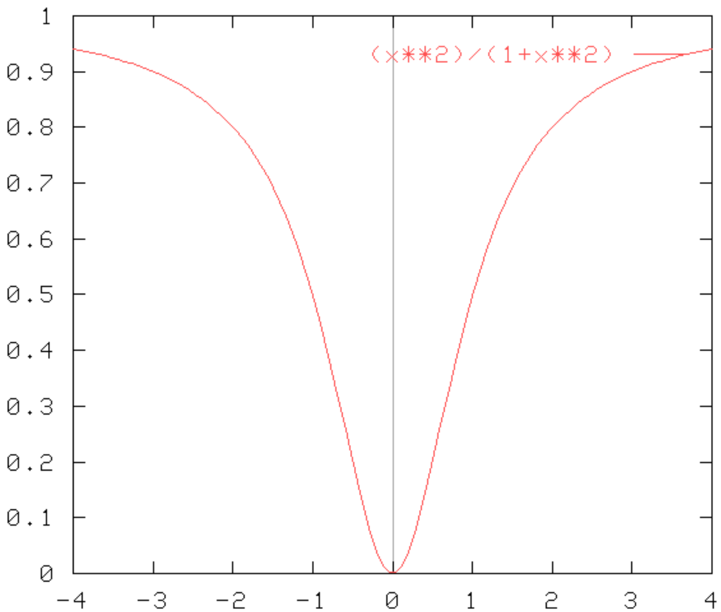
- w_i is a weight associated with each measurement. Can be varied to provide robustness to outliers.

Choices of the $\rho(r_i;\sigma)$ function:

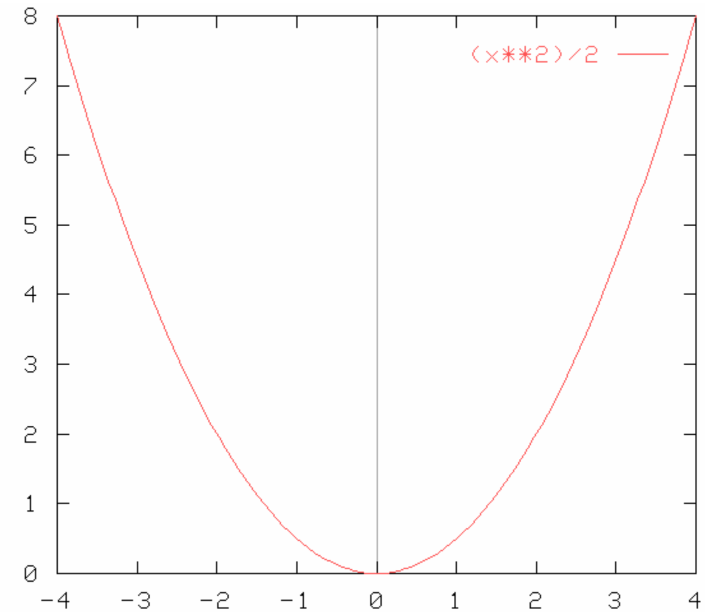
$$\rho_{SS} = \frac{r^2}{2\sigma^2} \qquad \rho_{GM} = \frac{r^2/\sigma^2}{1 + r^2/\sigma^2}$$

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

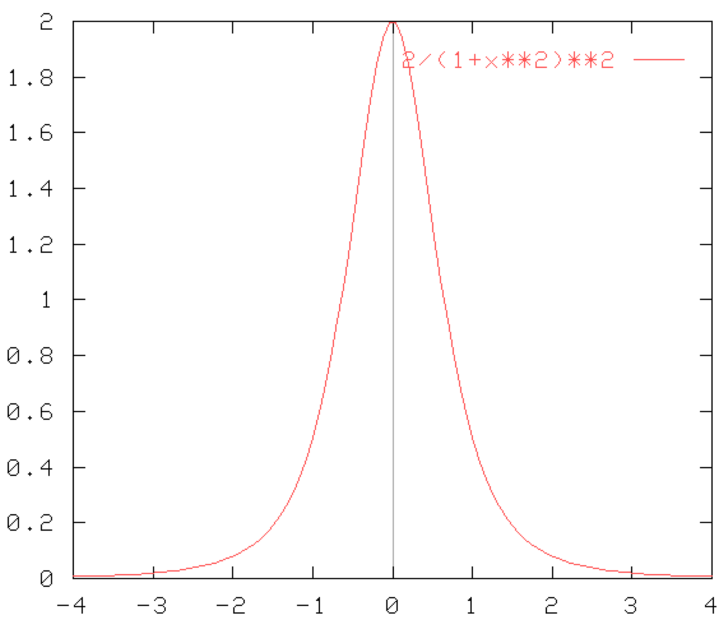
Optimization Functions & their Corresponding Weight Plots

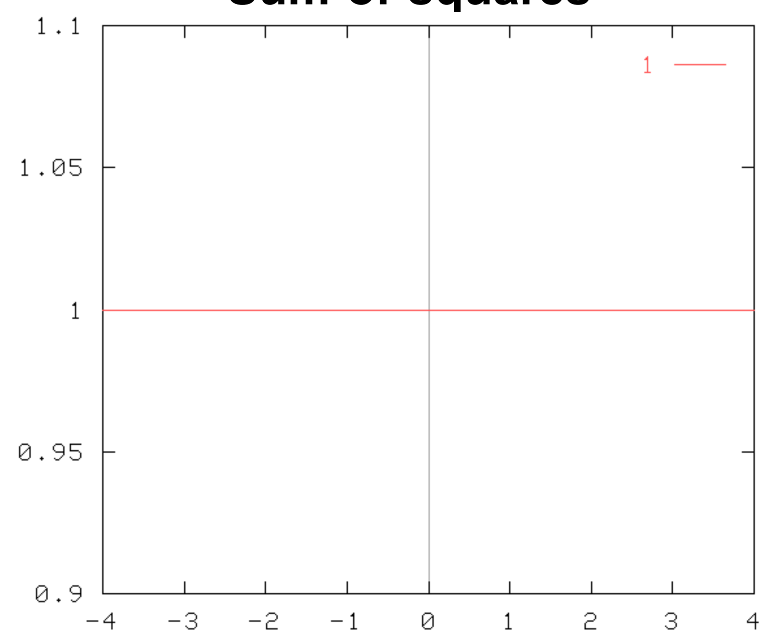


Geman-Mclure



Sum-of-squares





With Robust Functions Direct Alignment Works for Non-dominant Moving Objects Too

Original two frames

Background Alignment

Object Deletion with Layers

Original Video

Video Stream with deleted moving object

Optic Flow Estimation

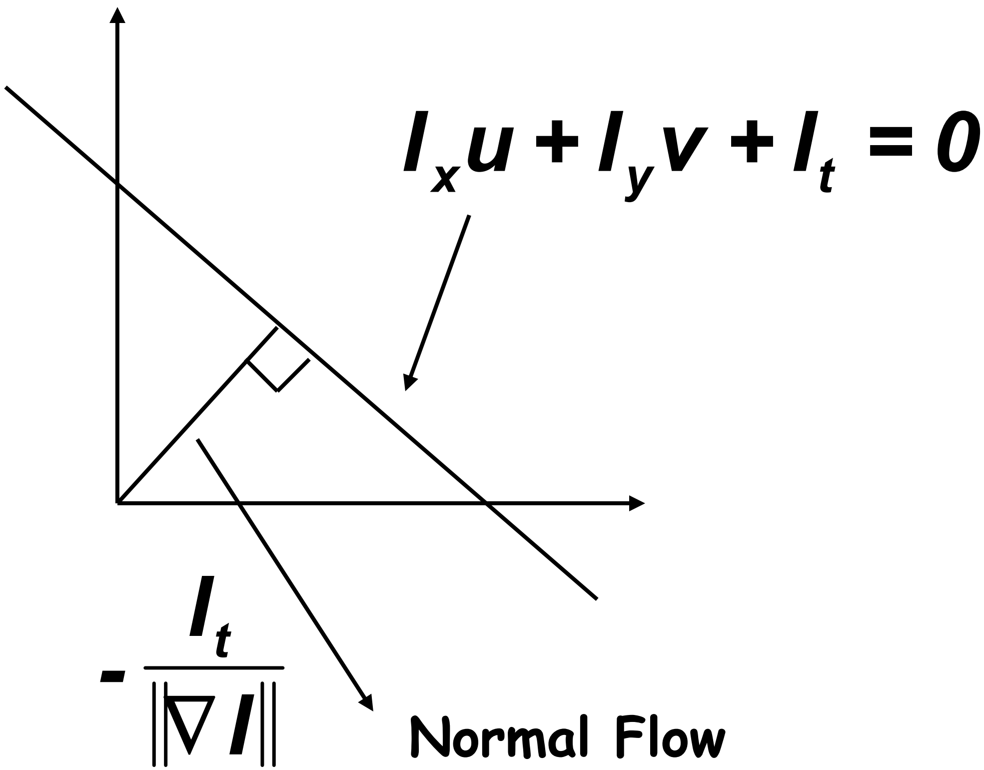
$$r_i = I_2(p_i) - I_1^{w}(p_i^{w} - \delta p_i^{w}(p_i^{w};D))$$

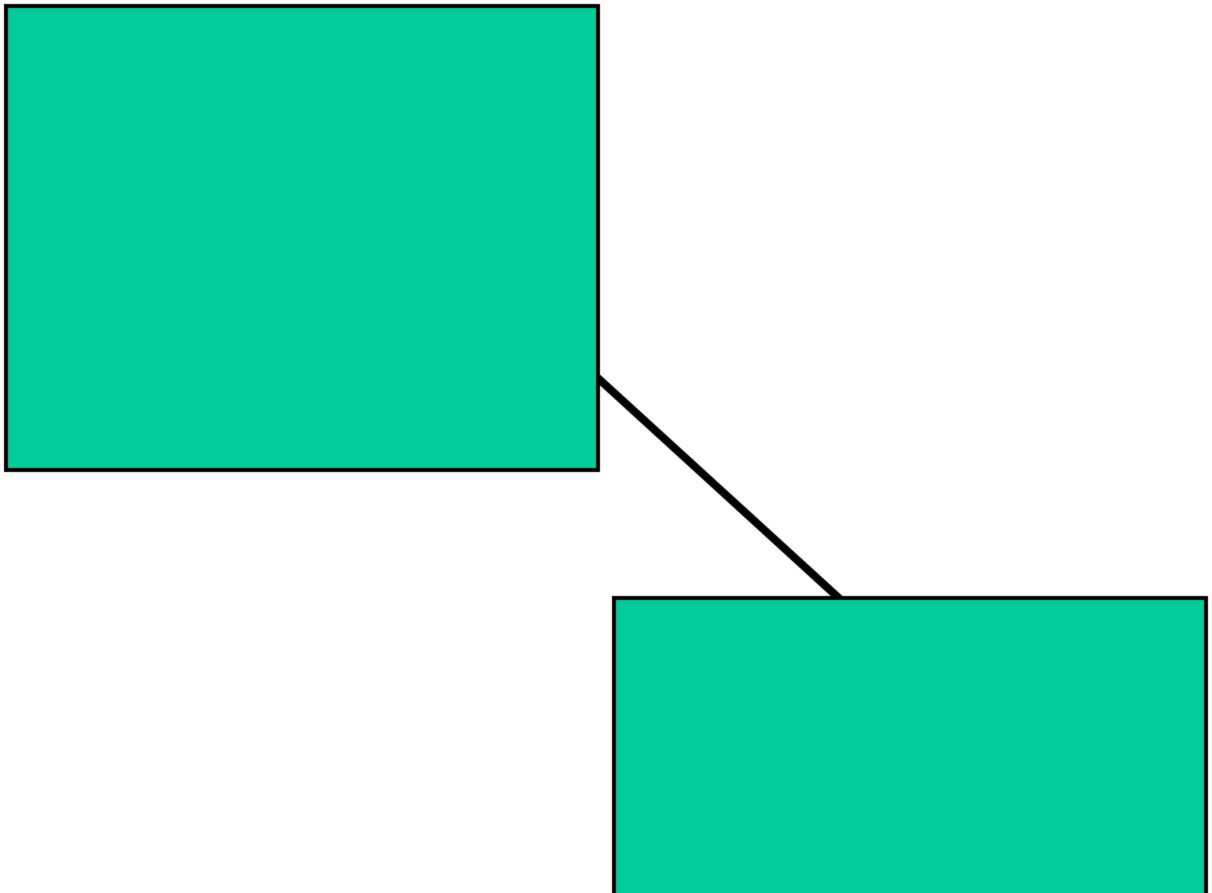
$$\approx I_2(p_i) - I^{w}(p_i^{w}(p_i;0)) - \nabla I^{w^T} \delta p$$

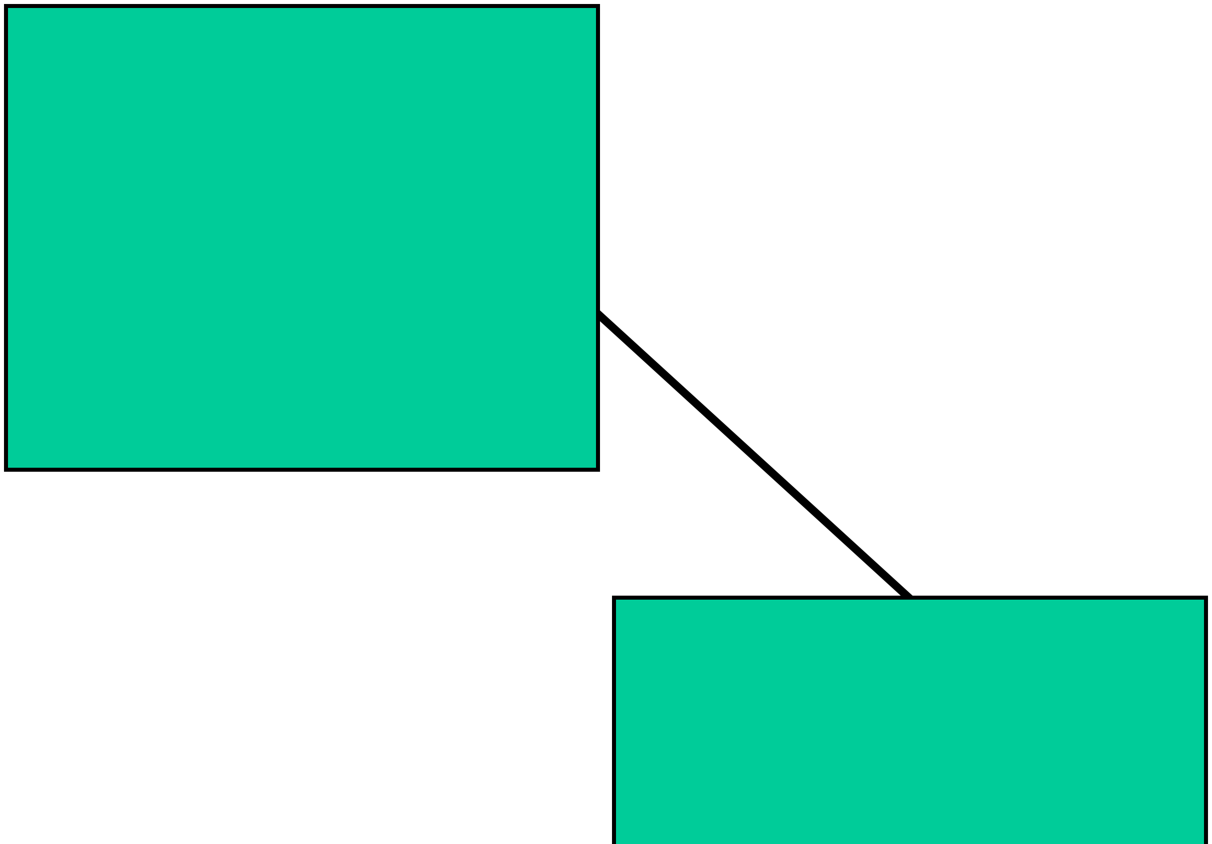
$$\underbrace{\begin{bmatrix} I_x^w & I_y^w \end{bmatrix}}_{\text{Gradient Direction}} \underbrace{\begin{matrix} \delta x \\ \delta y \end{matrix}}_{\text{Flow Vector}} \approx I_2(p_i) - I^{w}(p_i^{w}) = \delta I$$

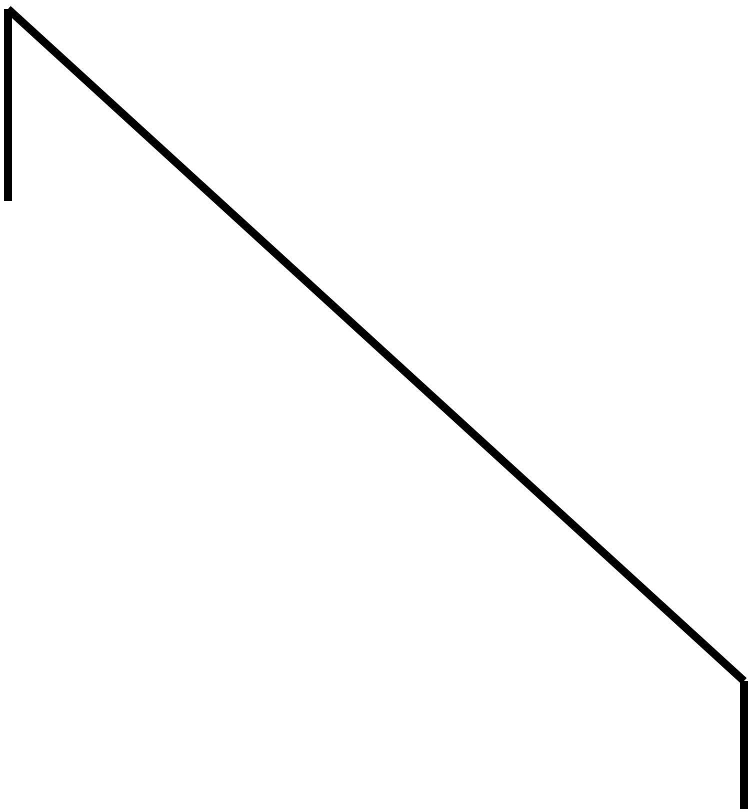
Normal Flow Constraint

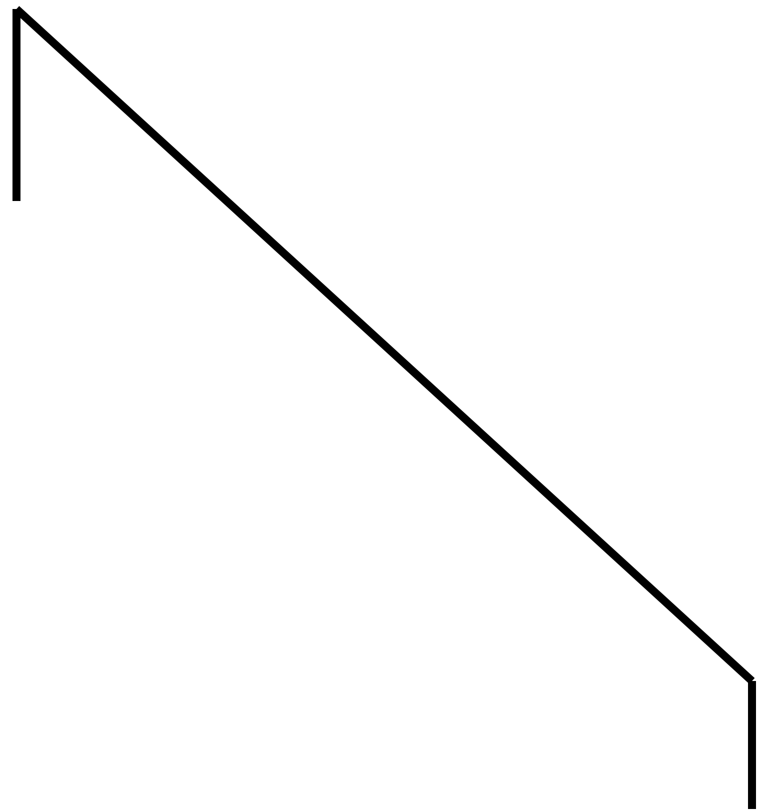
At a single pixel, brightness constraint:











Computing Optical Flow: Discretization

- Look at some neighborhood N:

$$\sum_{(i,j)\in \mathrm{N}} \left(\nabla I(i,j)\right)^{\mathrm{T}} \mathbf{v} + I_t(i,j) \overset{\text{want}}{=} 0$$

$$\mathbf{A}\mathbf{v} + \mathbf{b} \overset{\text{want}}{=} 0$$

$$\mathbf{A} = \begin{bmatrix} \nabla I(i_1, j_1) \\ \nabla I(i_2, j_2) \\ \vdots \\ \nabla I(i_n, j_n) \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} I_t(i_1, j_1) \\ I_t(i_2, j_2) \\ \vdots \\ I_t(i_n, j_n) \end{bmatrix}$$

Computing Optical Flow: Least Squares

- In general, overconstrained linear system
- Solve by least squares

$$\mathbf{A}\mathbf{v} + \mathbf{b} \overset{\text{want}}{=} 0$$

$$\Rightarrow (\mathbf{A}^{\mathrm{T}}\mathbf{A})\,\mathbf{v} = -\mathbf{A}^{\mathrm{T}}\mathbf{b}$$

$$\mathbf{v} = -(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}$$

Computing Optical Flow: Stability

- Has a solution unless $\mathbf{C} = \mathbf{A}^{\mathrm{T}}\mathbf{A}$ is singular

$$\mathbf{C} = \mathbf{A}^{\mathrm{T}}\mathbf{A}$$

$$\mathbf{C} = \begin{bmatrix} \nabla I(i_1, j_1) & \nabla I(i_2, j_2) & \cdots & \nabla I(i_n, j_n) \end{bmatrix} \begin{bmatrix} \nabla I(i_1, j_1) \\ \nabla I(i_2, j_2) \\ \vdots \\ \nabla I(i_n, j_n) \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} \sum_N I_x{}^2 & \sum_N I_x I_y \\ \sum_N I_x I_y & \sum_N I_y{}^2 \end{bmatrix}$$

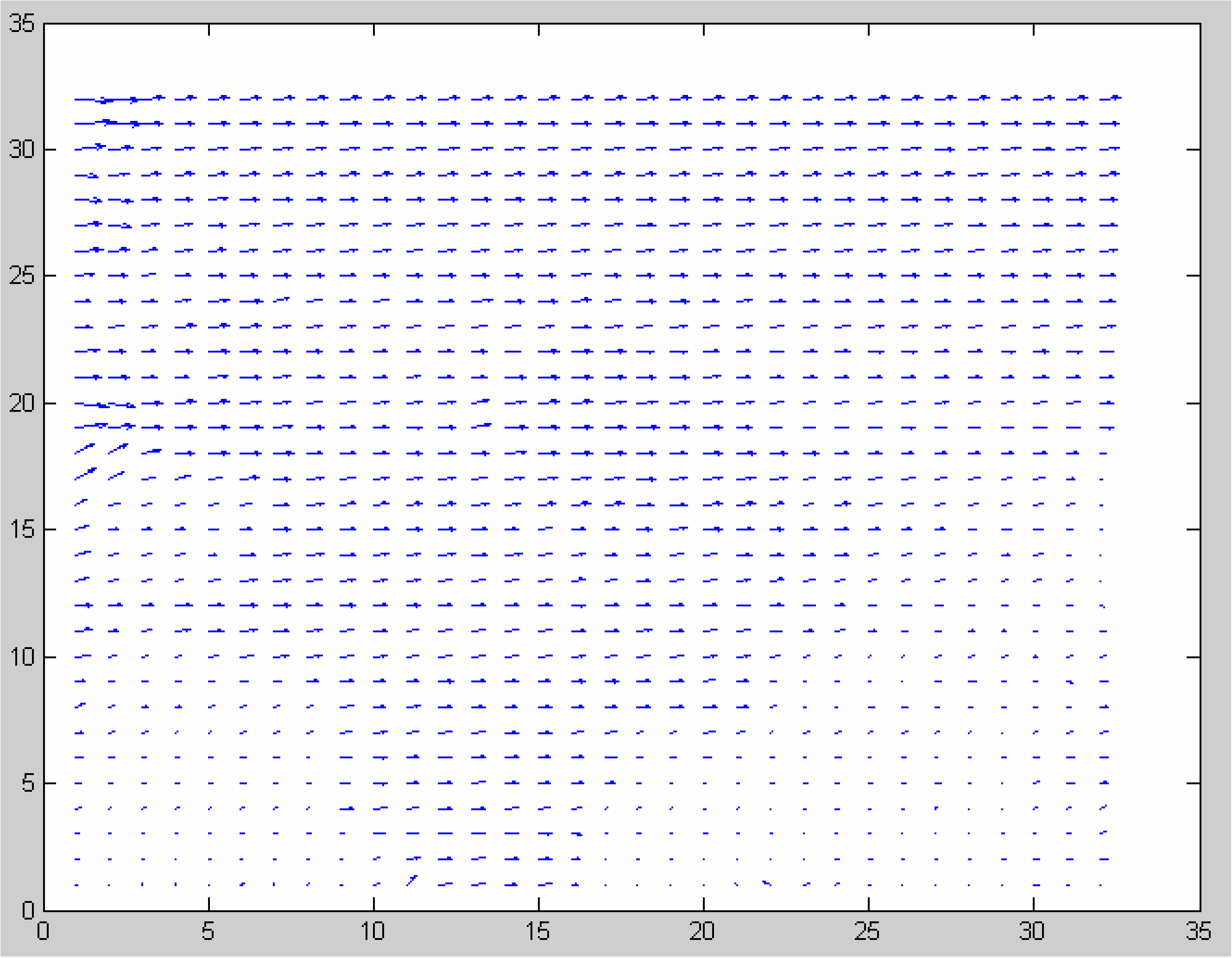
Computing Optical Flow: Stability

- Where have we encountered **C** before?
- Corner detector!
- **C** is singular if constant intensity or edge
- Use eigenvalues of **C**:
  - to evaluate stability of optical flow computation
  - to find good places to compute optical flow (finding good features to track)
  - [Shi-Tomasi]

Example of Flow Computation

Example of Flow Computation

Example of Flow Computation



But this in general is not the motion field

Motion Field = Optical Flow ?

From brightness constancy, normal flow: $v_n = \frac{(\nabla E^T)v}{\|\nabla E\|} = -\frac{E_t}{\|\nabla E\|}$

Motion field for a Lambertian scene:

$$E = \rho I^T n \qquad \frac{\mathrm{dn}}{\mathrm{dt}} = \omega x n \qquad \nabla E^T v + E_t = \rho I^T(\omega x n)$$

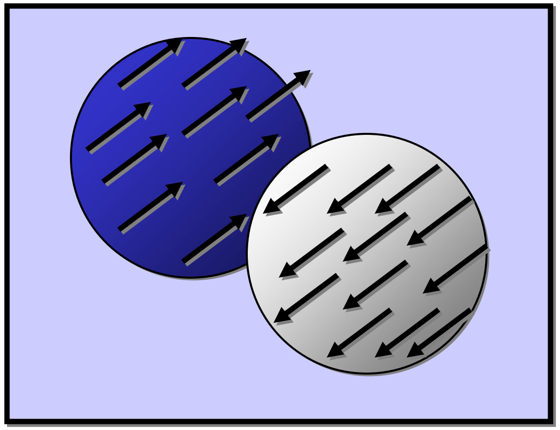
$$\therefore |\Delta v| = \rho \frac{|I^T(\omega x n)|}{\|\nabla E\|}$$

Points with high spatial gradient are the locations
At which the motion field can be best estimated
By brightness constancy (the optical flow)

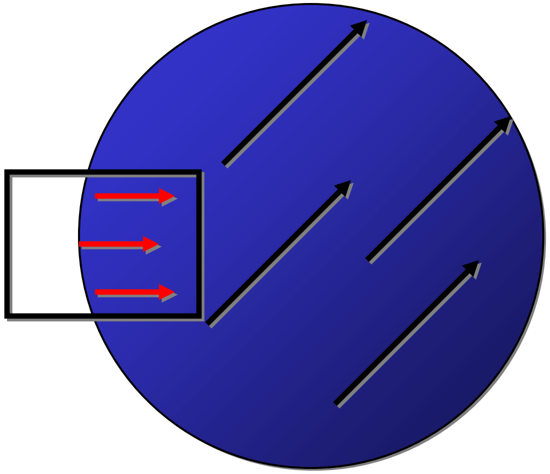
Motion Illusions
in
Human Vision

Aperture Problem

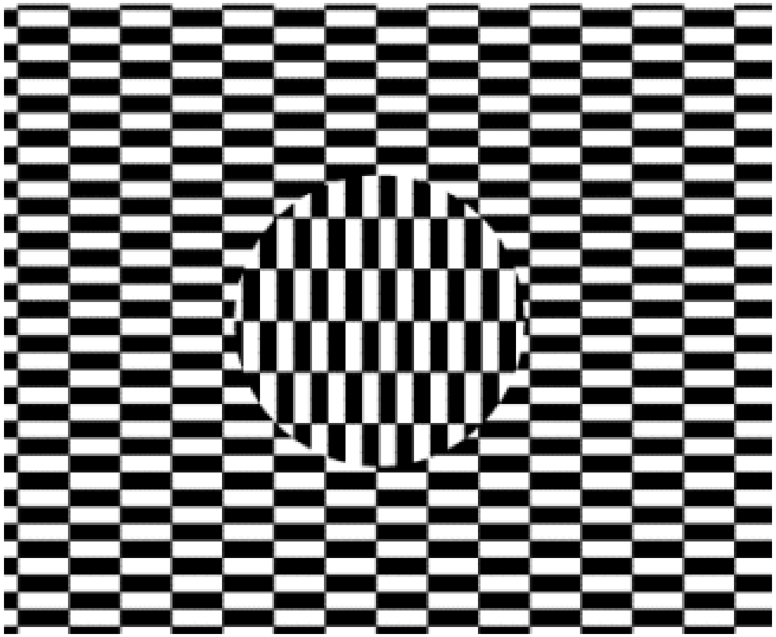
- Too big:
  confused by
  multiple motions



- Too small:
  only get motion
  perpendicular
  to edge



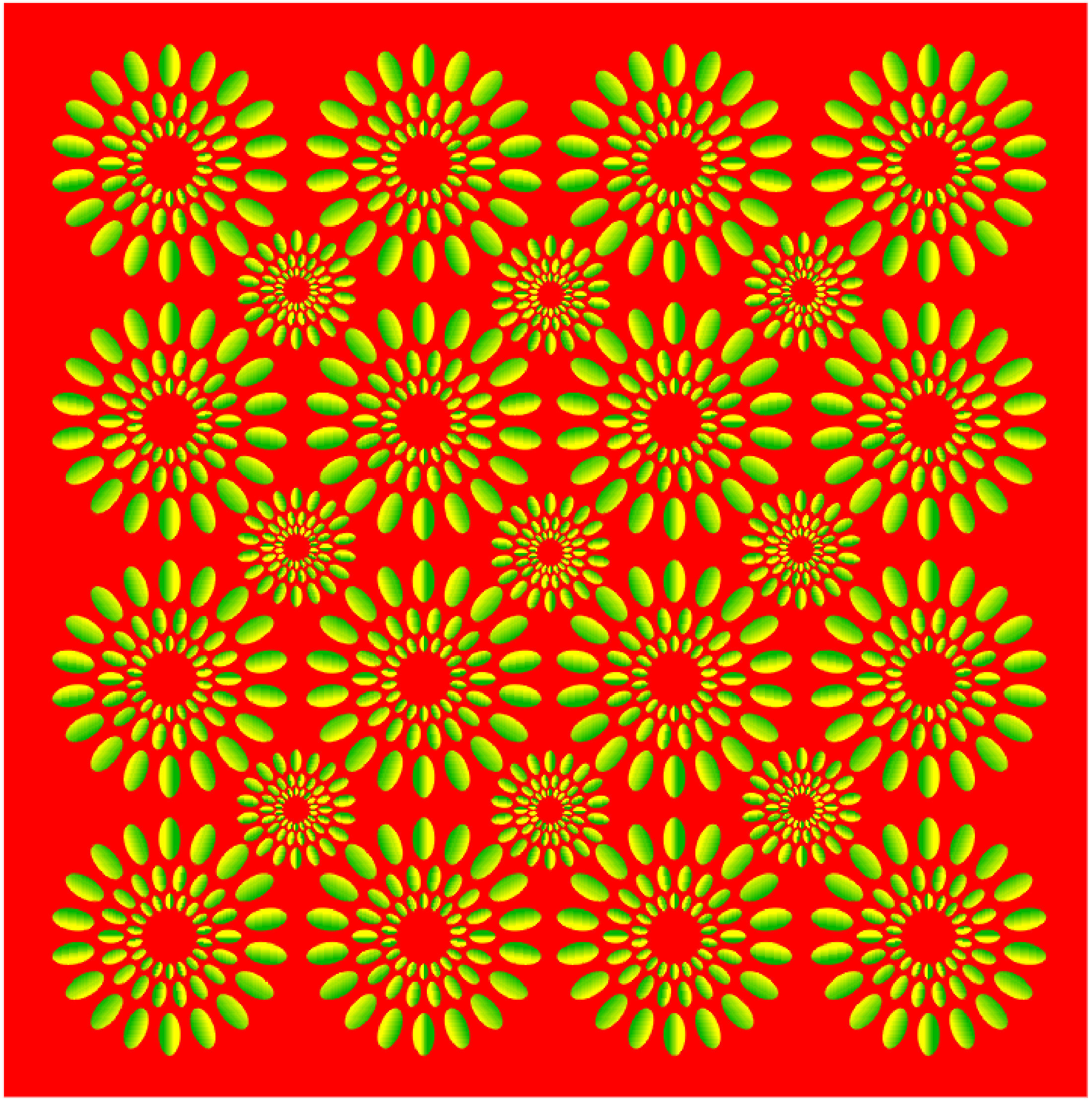
Ouchi Illusion



The Ouchi illusion, illustrated above, is an illusion named after its inventor, Japanese artist Hajime Ouchi. In this illusion, the central disk seems to float above the checkered background when moving the eyes around while viewing the figure. Scrolling the image horizontally or vertically give a much stronger effect. The illusion is caused by random eye movements, which are independent in the horizontal and vertical directions. However, the two types of patterns in the figure nearly eliminate the effect of the eye movements parallel to each type of pattern. Consequently, the neurons stimulated by the disk convey the signal that the disk jitters due to the horizontal component of the eye movements, while the neurons stimulated by the background convey the signal that movements are due to the independent vertical component. Since the two regions jitter independently, the brain interprets the regions as corresponding to separate independent objects (Olveczky *et al.* 2003).

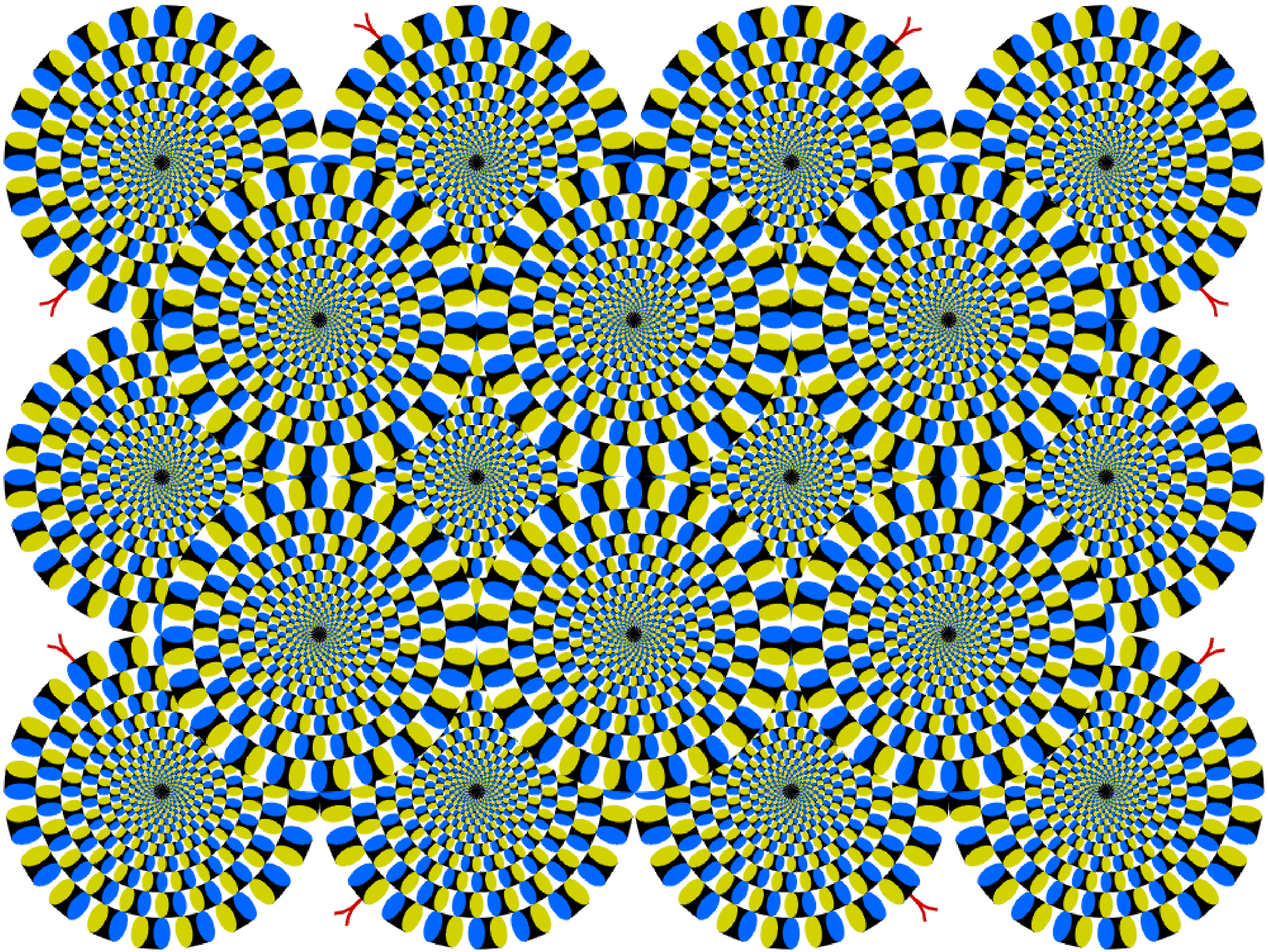
http://mathworld.wolfram.com/OuchiIllusion.html

Akisha Kitakao



http://www.ritsumei.ac.jp/~akitaoka/saishin-e.html

Rotating Snakes



The End