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…
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

\[ y = f \frac{Y}{Z} \quad \text{p} \approx f \text{P} \]
Camera Rotation (Pan)

\[ y' = f \frac{Y'}{Z'} \]

\[ p' \approx fP' \]

\[ P' = R'P \]

\[ p' \approx R'p \]
Camera Rotation (Pan)

\[ y'' = f \frac{Y''}{Z''} \]

\[ p'' \approx fP'' \]

\[ P'' = R''P \]

\[ p'' \approx R''p \]
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

\[ y = f \frac{Y}{Z} \quad \text{p} \approx fP \]
Camera Translation (Ty)

\[ y' = f \frac{Y'}{Z'} \]

\[ p' \approx fP' \]

\[ P' = P + T' \]
Translational Displacement

\[ y' = f \frac{Y'}{Z'} \]
\[ y' = f \frac{Y + Ty}{Z} \]
\[ y' - y = f \frac{Ty}{Z} \]

\[ y' = f \frac{Y}{Z + Tz} \]
\[ y' - y = -y' \frac{Tz}{Z} \]

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

In general, discrete correspondences are related through a transformation.

Images may be separated by time, space, sensor types.
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]

Model image transformation as:

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

Images separated by time, space, sensor types

Brightness Constancy
Formulation of Direct Model-based Image Alignment

Images separated by time, space, sensor types

Reference Coordinate System

Model image transformation as:

\[ I_2(p) = I_1(p - u(p; \Theta)) \]
Formulation of Direct Model-based Image Alignment

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
Formulation of Direct Model-based Image Alignment

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

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

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

$$\min_{\Theta} \sum_i \rho(r_i; \sigma),$$

$$r_i = l_2(p_i) - l_1(p_i - u(p_i; \Theta))$$

What all can be varied?

Filtered Image Representations (to account for Illumination changes, Multi-modalities)
Formulation of Direct Model-based Image Alignment

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

$$\min_{\Theta} \sum_{i} \rho(r_i; \sigma),$$

$$r_i = l_2(p_i) - l_1(p_i - u(p_i; \Theta))$$

What all can be varied?

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

Model Parameters
Formulation of Direct Model-based Image Alignment

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

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

What all can be varied?

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

Model Parameters

Measuring mismatches (SSD, Correlations)
Formulation of Direct Model-based Image Alignment

\[ I_1(p') \]

\[ p - u(p) \]

\[ I_2(p) \]

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

\[ \min_{\Theta} \sum_i \rho(r_i; \sigma), \quad \text{subject to} \quad r_i = I_2(p_i) - I_1(p_i - u(p_i; \Theta)) \]

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

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

\[
\min_{\Theta} \sum_{i} \rho(r_i; \sigma),
\]

\[
r_i = l_2(p_i) - l_1(p_i - u(p_i; \Theta))
\]

What all can be varied?

- Optimization Function
- Measuring mismatches (SSD, Correlations)
- Model Parameters
- Filtered Image Representations (to account for Illumination changes, Multi-modalities)
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]

\[ \min_{\Theta} \sum_{i} \rho(r_i; \sigma), \quad r_i = l_2(p_i) - l_1(p_i - u(p_i; \Theta)) \]

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

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

- \( w_i \) is a weight associated with each measurement.
An Iterative Solution of Model Parameters

\[
\min_{\Theta} \sum_i \rho(r_i; \sigma), \quad r_i = l_2(p_i) - l_1(p_i - u(p_i; \Theta))
\]

- In particular for Sum-of-Square Differences: \( \rho_{SSD} = \frac{r^2}{2\sigma^2} \)

- We obtain the standard normal equations:

\[
\sum_i \sum_i \frac{\partial r_i}{\partial \theta_k} \frac{\partial r_i}{\partial \theta_l} \partial \theta_l = - \sum_i r_i \frac{\partial r_i}{\partial \theta_k} \quad \forall k
\]

- Other functions can be used for robust M-estimation…
How does this work for images? (1)

\[
\min_\Theta \sum_i \frac{1}{2} r_i^2, \quad r_i = l_2(p_i) - l_1(p_i - u(p_i; \Theta))
\]

- Let there be a 2D projective transformation between the two images:
  \[ p'_i \approx P p_i \]

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

- First, warp \( l_1(p'_i) \) towards \( l_2(p_i) \)
How does this work for images? (2)

\[
\min_{\Theta} \sum_{i} \frac{1}{2} r_i^2, \\
r_i = l_2(p_i) - l_1(p_i - u(p_i; \Theta))
\]

\[
l_w^1(p_w) = l_1(p') = l_1(P^{(k)}p_w)
\]
How does this work for images? (3)

$$\min \sum_i \frac{1}{2} r_i^2,$$

$$r_i = I_2(p_i) - I_1(p_i - u(p_i; \Theta))$$

$$I^w_1(p^w) = I_1(p') = I_1(P^{(k)}p^w)$$

**$I^w_1(p^w)$**: Represents image 1 warped towards the reference image 2, using the current set of parameters.
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

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

\[ D = \begin{pmatrix}
    d11 & d12 & d13 \\
    d21 & d22 & d23 \\
    d31 & d32 & 0
\end{pmatrix} \]
How does this work for images? (5)

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

$$ r_i = l_2(p_i) - l_1^w(p_i^w - \delta p_i^w(p_i^w; D)) $$

$$ \approx l_2(p_i) - l^w(p_i^w(p_i; 0)) - \nabla l^w \frac{\partial p_i^w}{\partial d} |_{D=0} d $$

$$ p^w = \begin{bmatrix}
(1 + d_{11})x + d_{12}y + d_{13} \\
\frac{d_{31}x + d_{32}y + 1}{d_{21}x + (1 + d_{22})y + d_{23}} \\
d_{31}x + d_{32}y + 1
\end{bmatrix} $$

$$ \dddot{\frac{\partial p^w}{\partial D}} |_{D=0} = \begin{bmatrix}
x & y & 1 & 0 & 0 & 0 & -x^2 & -xy \\
0 & 0 & 0 & x & y & 1 & -xy & -y^2
\end{bmatrix} $$
How does this work for images? (6)

\[
\min \sum_i \frac{1}{2} r_i^2 , \\
\]

\[
\approx \nabla I_i w^T \nabla D p_i w \bigg|_{D=0} d - \delta I(p_i)
\]

\[
\sum_i \nabla D^T p_i w \nabla I_i w^T \nabla I_i w^T \nabla D p_i w d = \sum_i \nabla D^T p_i w \delta I(p_i)
\]

\[
Hd = g
\]

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

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.

\[ p' = P \, p^w \]

Source Image: Filled

Target Image: Empty

Bicubic Warp
Iterative Alignment: Result
How to handle Large Transformations?

[Burt, Adelson'81]

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

![Pyramid Processing](image)
Iterative Coarse-to-fine Model-based Image Alignment

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

- Warper
  - $d(p)$
  - $\{ R, T, d(p) \}$
  - $\{ H, e, k(p) \}$
  - $\{ dx(p), dy(p) \}$
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

\[ \min_{\Theta} \sum_{i} \rho(r_i; \sigma), \quad r_i = l_2(p_i) - l_1(p_i - u(p_i; \Theta)) \]

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

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

• \( W_i \) is a weight associated with each measurement. Can be varied to provide robustness to outliers.

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

\[
\rho_{SS} = \frac{r^2}{2\sigma^2} \quad \rho_{GM} = \frac{r^2/\sigma^2}{1 + r^2/\sigma^2}
\]

\[
\frac{\dot{\rho}_{SS}(r)}{r} = \frac{1}{\sigma^2} \quad \frac{\dot{\rho}_{GM}(r)}{r} = \frac{2\sigma^2}{(\sigma^2 + r^2)^2}
\]
Optimization Functions & their Corresponding Weight Plots

Geman-Mclure

\[ \frac{x^2}{1+x^2} \]

Sum-of-squares

\[ \frac{2}{(1+x^2)^2} \]
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

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

\[ \approx I_2(p_i^w) - I^w(p_i^w(p_i; 0)) - \nabla I^w \delta p \]

\[
\begin{bmatrix}
I_x^w \\
I_y^w
\end{bmatrix}
\begin{bmatrix}
\delta x \\
\delta y
\end{bmatrix}
\approx I_2(p_i^w) - I^w(p_i^w) = \delta I
\]

Gradient Direction
Flow Vector
Normal Flow Constraint

At a single pixel, brightness constraint:

\[ I_x u + I_y v + I_t = 0 \]
Computing Optical Flow: Discretization

• Look at some neighborhood $N$:

$$\sum_{(i,j) \in N} (\nabla I(i, j))^T v + I_t(i, j) = 0$$

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

$$A = \begin{bmatrix} \nabla I(i_1, j_1) \\ \nabla I(i_2, j_2) \\ \vdots \\ \nabla I(i_n, j_n) \end{bmatrix} \quad b = \begin{bmatrix} I_t(i_1, j_1) \\ I_t(i_2, j_2) \\ \vdots \\ I_t(i_n, j_n) \end{bmatrix}$$
Computing Optical Flow: Least Squares

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

\[ Av + b = 0 \]

\[ \Rightarrow (A^T A) v = -A^T b \]

\[ v = -(A^T A)^{-1} A^T b \]
Computing Optical Flow:
Stability

• Has a solution unless \( \mathbf{C} = \mathbf{A}^T \mathbf{A} \) is singular

\[
\mathbf{C} = \mathbf{A}^T \mathbf{A}
\]

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

\[
\mathbf{C} = \begin{bmatrix}
\sum_N I_x^2 & \sum_N I_x I_y \\
\sum_N I_x I_y & \sum_N I_y^2
\end{bmatrix}
\]
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
From brightness constancy, normal flow:

\[ \mathbf{v}_n = \frac{\nabla E^T \mathbf{v}}{\|\nabla E\|} = -\frac{E_t}{\|\nabla E\|} \]

Motion field for a Lambertian scene:

\[ E = \rho l^T n \quad \frac{dn}{dt} = \omega xn \quad \nabla E^T \mathbf{v} + E_t = \rho l^T (\omega xn) \]

\[ \therefore |\Delta \mathbf{v}| = \rho \left| \frac{l^T (\omega xn)}{\|\nabla E\|} \right| \]

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
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