Recapitulation

- Problem of motion estimation
- Parametric models of motion
- Direct methods for image motion estimation
- Camera models & parametric motion
- Image & video mosaicing as an application
Plan

- Motivate Image-based Modeling & Rendering (IBMR)
  - Change in viewpoint, IMAX app
- Parameterize motion & structure for video
  - Euclidean case
  - Direct Estimation
- Plane+Parallax
  - Formulation
  - Direct Estimation
- IMAX app.
- Tweening app.
- Model-to-video pose estimation
- Video Flashlights
Graphics Vs. Vision
Graphics Vs. Vision

Vision
Modeling Complexity vs. Realism Dilemma

Can we model and render this scene authentically?

A very hard vision and graphics problem!
Can we model less and fill-in details with more and more images?
Plenoptic Function

Describes rays of light at every point in space from every direction, at every wavelength, and at each time instant !!!

\[ f(X, Y, Z, \theta, \varphi, \lambda, t) \]
Pure IBR
Use Images to Generate Novel Views

• Mosaics and Panoramas:
  – Projective, Cylindrical, Spherical

• Concentric Mosaics:
  – Restricted change in viewpoint
An Illustrative Example
Concentric Mosaics
An Illustrative Example
Concentric Mosaics
An Illustrative Example
Concentric Mosaics
An Illustrative Example
Concentric Mosaics
An Illustrative Example
Concentric Mosaics
An Illustrative Example
Concentric Mosaics
An Illustrative Example
Concentric Mosaics
An Illustrative Example
Concentric Mosaics
New View Generation
New View Generation
New View Generation
New View Generation

Details:

• New view rays will not be exactly original pixel locations
• Need to interpolate rays
• Since these rays come from cameras that have different centers of projection, interpolation ideally requires depth estimates
Local Geometry for IBR

- "Extreme" IBR works only with images

- Need a dense collection to cover even a restricted collection of generatable viewpoints

- Extend the scope of IBR by including local geometry modeling using parallax/depth
Image Motion : 3D Parameterization

\[ \dot{\mathbf{P}} = \Omega \times \mathbf{P} + \mathbf{T} \]

\[ \mathbf{P}' - \mathbf{P} \approx \Omega \times \mathbf{P} + \mathbf{T} \]

\[ \mathbf{P}' \approx \mathbf{P} + \Omega \times \mathbf{P} + \mathbf{T} \]

\[ \mathbf{P}' \approx [I + \Omega_x] \mathbf{P} + \mathbf{T} \]

\[
\Omega_x = \begin{pmatrix}
0 & -\omega_z & \omega_y \\
\omega_z & 0 & -\omega_x \\
-\omega_y & \omega_x & 0
\end{pmatrix}
\]
Image Motion: 3D Parameterization

\[
x' = \frac{X - \omega_z Y + \omega_y Z + T_x}{-\omega_y X + \omega_x Y + Z + T_z}
\]

\[
x' = \frac{x - \omega_z y + \omega_y y + \frac{T_x}{Z}}{-\omega_y x + \omega_x y + 1 + \frac{T_z}{Z}}
\]

\[
y' = \frac{\omega_z X + Y - \omega_x Z + T_y}{-\omega_y X + \omega_x Y + Z + T_z}
\]

\[
y' = \frac{\omega_z x + y - \omega_x + \frac{T_y}{Z}}{-\omega_y x + \omega_x y + 1 + \frac{T_z}{Z}}
\]

Using small motion approximation

\[
x' - x = \omega_y - \omega_z y - \omega_x xy + \omega_y y^2 + \frac{T_x - T_z x}{Z}
\]

Rotational Component

\[
y' - y = -\omega_x + \omega_z x + \omega_y xy - \omega_x y^2 + \frac{T_y - T_z y}{Z}
\]

Translational Component
Observations

• For a perspective camera:
  – Rotational image displacements are depth independent
  – Translational displacements contain depth information
• Each displacement vector provides two equations
  – There are 6 motion parameters common to all vectors
  – Depth parameter is typically unique to each vector
• Scale ambiguity between depth and translation

\[
x' - x = \frac{T_x - T_z x}{Z} \\
y' - y = \frac{T_y - T_z y}{Z}
\]

(\alpha T, \alpha Z) \text{ Give the same flow}
Observations

\[ x' - x = \frac{T_x - T_z x}{Z} \]

\[ y' - y = \frac{T_y - T_z y}{Z} \]

Focus of Expansion/Contraction

\[ \left( \frac{T_x}{T_z}, \frac{T_y}{T_z} \right) \]
Quasi-Parametric Motion Model

\[ x' - x = \omega_y - \omega_z y - \omega_x x y + \omega_y x^2 + \frac{T_x - T_z x}{Z} \]

\[ y' - y = -\omega_x + \omega_z x + \omega_y x y - \omega_x y^2 + \frac{T_y - T_z y}{Z} \]

Global Parametric Component

Local Parametric Component

For rigid small motion scenario:

- 6 global model parameters
- \((\omega_x, \omega_y, \omega_z)\) Rotations \((T_x, T_y, T_z)\) Translations
- One local parameter per pixel: Depth \(Z\)
Direct Model Estimation

\[ E_{SSD}(u) = \sum_{p \in \mathbb{R}} \left( \nabla I_i^T \delta u(p) + \delta I(p) \right)^2 \]

Recall:

\[ \delta u(p) = u(p) - u^{(k)}(p) \]

\[ u(p) = \begin{pmatrix} -xy & 1 + x^2 & -y \omega_x \\ -y(1 + y^2) & xy & x \omega_y \end{pmatrix}(\omega) + \begin{pmatrix} 1 & 1 & 0 & -x \\ 0 & 1 & y \\ \omega_z \\ T_x \\ T_y \\ T_z \end{pmatrix} \]

\[ u(p) = A\Omega + \tilde{Z}KT \]

Observations: The error function is quadratic in inverse depth

Given the depth at each pixel, it is quadratic in rotation and translation parameters

Therefore, can be iteratively solved using depth and motion optimizations
Solution

Simplifying assumption: Although depth varies from pixel-to-pixel we assume, that within a small window, (say 5x5) around each pixel, depth is constant

Local Optimization Step: Eliminate depth analytically

\[ E_{\text{local}}(u) = \sum_{p \in R_{\text{local}}} (\nabla l^T_1 \delta u(p) + \delta l(p))^2 \]

\[ \frac{\partial E_{\text{local}}}{\partial \tilde{Z}} = 0 \quad \frac{\partial E_{\text{local}}}{\partial \tilde{Z}} = 2 \sum_{p \in R_{\text{local}}} \nabla l^T_1 K T (\nabla l^T_1 \delta u(p) + \delta l(p)) \]

\[ \delta u(p) = A(\Omega - \Omega^{(k)}) + (\tilde{Z} K T - \tilde{Z}^{(k)} K T^{(k)}) \]

\[ \tilde{Z}_{\text{opt}} = \frac{\sum_{p \in R_{\text{local}}} \nabla l^T_1 K T (\nabla l^T_1 A(\Omega - \Omega_0) - \nabla l^T_1 \tilde{Z}^{(k)} K T^{(k)} + \delta l(p))}{\sum_{p \in R_{\text{local}}} (\nabla l^T_1 K T)^2} \]
Solution

Global Optimization Step: Substitute the analytical solution for $Z$ in the optimization function and solve iteratively for rotation and translation.

\[ E_{\text{global}}(u) = \sum_{p \in R_{\text{global}}} (\nabla I_1^T A(\Omega - \Omega_0) + \nabla I_1^T \tilde{Z}_{\text{min}} KT - \nabla I_1^T \tilde{Z}^{(k)} KT^{(k)} + \delta I(p))^2 \]

- This optimization function is quadratic in rotations
- So solve for rotations using closed form
- Solve for translations analytically
- Employ the older trick of warp and solve
Global Constraints on Local Motion

Unambiguous Local Estimate

3D Motion Constraint

Brightness Constancy
Layered Representation of 3D Geometry

• Dense representation of geometry
  – Planar layers capture sharp depth boundaries
  – Global matching constraints handle visibility

• Alignment based estimation of structure
  – Correspondences are not a pre-requisite for structure estimation
  – 3D constrained depth/parallax maps are sharper

• Independent motion detection with 3D constraints
  – Shape and motion constraints used to detect independent motions
LOCAL RANGE ESTIMATION HANDLES ARBITRARY SHAPED OBJECTS

Two Views

Automatically computed Shape/Depth

Synthetic video rendered from original new views
Multi-baseline depth estimation

Depth maps

New view rendering

Global matching method

Accurate boundaries

Thin structures
Depth vs. Parallax

• Euclidean depth estimation requires calibrated cameras:
  – External Calibration : Camera R & T for each position of the camera
  – Internal Calibration : Focal length, center, skew

• IBR can be accomplished usually without calibration:
  – Compute parallax instead of depth
  – A special practical version : plane + parallax
Direct Estimation of Parallax/Depth
**Plane + Parallax 3D Model**

[Sawhney ’94, Kumar et al.’94, Shashua & Navab’94]

- Given a reference view, a reference plane, points in any other view are related by:
  1. a plane projective transformation
  2. the epipole, and
  3. the “relative depth/parallax”

That is,

\[ p' \approx A_{\pi} p + \kappa t' \]

\[ \kappa = d_P / (Zd_{\pi}) \]
Plane+Parallax for Small Displacements

\[
\begin{align*}
\mathbf{u}(\mathbf{p}) &= \begin{pmatrix} -xy & 1 + x^2 & -y \\
- (1 + y^2) & xy & x \end{pmatrix} \mathbf{\omega} + \begin{pmatrix} 1 & 1 & 0 & -x \\
0 & 1 & y \end{pmatrix} \begin{pmatrix} T_x \\
T_y \end{pmatrix} \\
\mathbf{u}(\mathbf{p}) &= \mathbf{A} \mathbf{\Omega} + \tilde{\mathbf{Z}} \mathbf{K} \mathbf{T}
\end{align*}
\]

Plane equation in image coordinates:

\[
\tilde{\mathbf{Z}} = a\mathbf{x} + b\mathbf{y} + \mathbf{c}
\]

\[
\mathbf{u}_{\text{plane}}(\mathbf{p}) = \mathbf{A} \mathbf{\Omega} + (a\mathbf{x} + b\mathbf{y} + \mathbf{c}) \mathbf{K} \mathbf{T}
\]

\[
\mathbf{u}(\mathbf{p}) = \mathbf{u}_{\text{plane}}(\mathbf{p}) + (\tilde{\mathbf{Z}} - \tilde{\mathbf{Z}}_{\text{plane}}) \mathbf{K} \mathbf{T}
\]

\[
\mathbf{u}(\mathbf{p}) = \mathbf{u}_{\text{plane}}(\mathbf{p}) - \frac{\delta \mathbf{Z}}{\mathbf{Z}} \mathbf{Z}_{\text{plane}} \mathbf{K} \mathbf{T} = \mathbf{u}_{\text{plane}}(\mathbf{p}) - \frac{\mathbf{H}}{d_\pi} \mathbf{Z}_{\text{plane}} \mathbf{K} \mathbf{T}
\]
**Shape Estimation through 3D Constrained Image Alignment**

- Parameterize the image correlation constraint with plane+parallax:
  \[ I_2(p) = I_1(p - u(p; A_{\pi}, t', \kappa(p))) \]

- Given an estimate of \( A^{(m)}_{\pi}, t'^{(m)}, \kappa^{(m)}(p) \)
  \( I_1 \) is warped towards \( I_2 \) using the plane+parallax model:
  \[ I^w(p) = I_1(p - u^{(m)}(p)) \]

- With \( \delta u(p) = u(p) - u^{(m)}(p) \)
  compute the increment in the unknown parameters:
  \[ \min_{A^{(m)}_\pi, t', \kappa(p)} \sum_p \rho(I(p) - I^w(p) + \nabla I^T(p) \delta u(p)) \]

\( \rho \) is a robust error function as before.
Plane+parallax Demo

Input Sequence

Plane Alignment

Shape

Parallax Alignment
A Novel Real-world Application of IBR

IMAX 3D Movie Synthesis