Vision-based Landing of an Unmanned Air Vehicle

O. Shakernia, R. Vidal, S. Sastry
Department of EECS, UC Berkeley
Applications of Vision-based Control

- Predator
- Global Hawk
- UCAV X-45
- SR/71
- Fire Scout
Goal: Autonomous landing on a ship deck

Challenges
- Hostile environments
  - Ground effect
  - Pitching deck
  - High winds, etc

Why vision?
- Passive sensor
- Observes relative motion
Simulation: Vision in the loop
Vision-Based Landing of a UAV

- Motion estimation algorithms
  - Linear, nonlinear, multiple-view
  - Error: 5cm translation, 4° rotation
- Real-time vision system
  - Customized software
  - Off-the-shelf hardware
- Vision in Control Loop
- Landing on stationary deck
- Tracking of pitching deck
Vision-based Motion Estimation

Pinhole Camera

\[ \lambda^j x^j = [R \ T] q^j \]
Pose Estimation: Linear Optimization

- Pinhole Camera:  \( \lambda_i x_i = [R \ T] q_i \)
- Epipolar Constraint:  \( 0 = \hat{x}_i[R \ T] q_i \)
- Planar constraint:  \( 0 = e_T^T q_i \quad \forall i \)

\[
0 = \hat{x}_i \begin{bmatrix} r_1 & r_2 & T \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ 1 \end{bmatrix} \Rightarrow G \begin{bmatrix} r_1 \\ r_2 \\ T \end{bmatrix} = 0
\]

- More than 4 feature points  \( \Rightarrow \text{rank}(G) = 8 \)
- Solve linearly for  \( [r_1^T \ r_2^T \ T^T]^T \in \mathbb{R}^9 \)
- Project  \( [r_1 \ r_2 \ 0] \in \mathbb{R}^{3\times3} \) onto \( SO(3) \) to recover \( R \).
Pose Estimation: Nonlinear Refinement

- Objective: minimize error

\[ G_i = \hat{x}_i[R \ T]q_i \]

\[ G = [G_1^T \ \ldots \ \ G_n^T]^T \]

- Parameterize rotation by Euler angles

\[ R = R_z(\psi)R_y(\theta)R_x(\phi) \]

\[ \beta = [\psi \ \theta \ \phi \ \ Tx \ \ Ty \ \ Tz]^T \in \mathbb{R}^6 \]

- Minimize by Newton-Raphson iteration

\[ \beta_{n+1} = \beta_n - k_n(D_\beta G|_{\beta_n})^+G(q, x, \beta_n) \]

- Initialize with linear algorithm
Multiple-View Motion Estimation

Pinhole Camera
\[ \lambda_i \mathbf{x}_i = [R_i \quad T_i] \mathbf{q} \]

- \[ R_1 = I_{3 \times 3}, \quad T_1 = 0 \]
- \[ \lambda_i \mathbf{x}_i = \lambda_1 R_i \mathbf{x}_1 + T_i \]
- \[ 0 = [\hat{x}_i R_i \mathbf{x}_1 \quad \hat{x}_i T_i] \begin{bmatrix} \lambda_1 \\ 1 \end{bmatrix} \]
- \[ 0 = \lambda_1 \pi^1 \mathbf{x}_1 + \pi^2 \]

Multiple View Matrix
\[ M = \begin{bmatrix} \hat{x}_2 R_2 \mathbf{x}_1 & \hat{x}_2 T_2 \\ \vdots & \vdots \\ \hat{x}_m R_m \mathbf{x}_1 & \hat{x}_m T_m \\ \pi^1 \mathbf{x}_1 & \pi^2 \end{bmatrix} \]

Rank deficiency constraint
\[ M [\lambda_1, 1]^T = 0 \quad \Rightarrow \quad \text{rank}(M) \leq 1 \]
Multiple-View Motion Estimation

- \( n \) points in \( m \) views for \( j = 1, \ldots, n \):
  \[
  \alpha^j \begin{bmatrix}
  \hat{x}_2^j T_2 \\
  \vdots \\
  \hat{x}_m^j T_m \\
  \pi^2
  \end{bmatrix} + \begin{bmatrix}
  \hat{x}_2^j R_2 x_1^j \\
  \vdots \\
  \hat{x}_m^j R_m x_1^j \\
  \pi^1 x_1^j
  \end{bmatrix} = 0.
  \]

- Equivalent to finding \( \pi \in \mathbb{R}^4, \vec{R}_i \in \mathbb{R}^9 \) and \( \vec{T}_i \in \mathbb{R}^3 \), s.t.
  \[
  \begin{bmatrix}
  x_1^1 T \\
  \vdots \\
  x_n^T \\
  \alpha^1 \\
  \vdots \\
  \alpha^n
  \end{bmatrix} \pi^T = 0,
  \begin{bmatrix}
  \alpha^1 \hat{x}_i^1 \\
  \vdots \\
  \alpha^n \hat{x}_i^n \\
  \hat{x}_i^1 \times x_1^T \\
  \vdots \\
  \hat{x}_i^n \times x_1^T
  \end{bmatrix}
  \begin{bmatrix}
  \vec{T}_i \\
  \vec{R}_i
  \end{bmatrix} = 0
  \]

- Initialize \( R_2, T_2, \pi \) with two-view linear solution.

- Least squared solution:
  \[
  \alpha^j = \frac{- (\hat{x}_2^j T_2) \hat{x}_2^j R_2 x_1^j + \pi^2 \pi^1 x_1^j}{|| \hat{x}_2^j T_2 ||^2 + (\pi^2)^2}.
  \]

- Use \( \alpha^j \) to linearly solve for \( \pi, \vec{R}_i, \vec{T}_i \).

- Iterate until \( \alpha^j \) converge.
Real-time Vision System

- Ampro embedded Little Board PC
  - Pentium 233MHz running LINUX
  - 440 MB flashdisk HD robust to vibration
  - Runs motion estimation algorithm
  - Controls Pan/Tilt/Zoom camera

- Motion estimation algorithms
  - Written and optimized in C++ using LAPACK
  - Estimate relative position and orientation at 30 Hz
Hardware Configuration

On-board UAV

Vision System
- Vision Computer
  - Camera
  - WaveLAN to Ground
  - Frame Grabber
  - Vision Algorithm
  - RS232

Navigation System
- Navigation Computer
  - INS/GPS
  - WaveLAN to Ground
  - Control & Navigation
  - RS232
  - RS232
Feature Extraction

- Acquire Image
- Threshold Histogram
- Segmentation
- Target Detection
- Corner Detection
- Correspondence
Camera Control

- Pan/Tilt to keep features in image center
  - Prevent features from leaving field of view
  - Increased Field of View
  - Increased range of motion of UAV
### Comparing Vision with INS/GPS

<table>
<thead>
<tr>
<th>Onboard Camera View</th>
<th>X Translation (m)</th>
<th>X Rotation (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Onboard Camera View" /></td>
<td><img src="image" alt="X Translation Graph" /></td>
<td><img src="image" alt="X Rotation Graph" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Y Translation (m)</th>
<th>Y Rotation (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Y Translation Graph" /></td>
<td><img src="image" alt="Y Rotation Graph" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Thresholded Image</th>
<th>Features</th>
<th>Z Translation (m)</th>
<th>Z Rotation (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Thresholded Image" /></td>
<td><img src="image" alt="Features" /></td>
<td><img src="image" alt="Z Translation Graph" /></td>
<td><img src="image" alt="Z Rotation Graph" /></td>
</tr>
</tbody>
</table>
Motion Estimation in Real Flight Tests

Comparison of Motion Estimation Algorithms

Translation error (cm)

Rotation error (degrees)

ICRA 2004
Landing on Stationary Target
Tracking Pitching Target

Vision-Based Landing of a UAV

Omid Shakernia
Dept of EECS, UC Berkeley
http://robotics.eecs.berkeley.edu/~omids
Conclusions

Contributions
- Vision-based motion estimation (5cm accuracy)
- Real-time vision system in control loop
- Demonstrated proof of concept prototype: first vision-based UAV landing

Extensions
- Dynamic vision: Filtering motion estimates
- Symmetry-based motion estimation
- Fixed-wing UAVs: Vision-based landing on runways
- Modeling and prediction of ship deck motion
- Landing gear that grabs ship deck
- Unstructured environments: Recognizing good landing spots (grassy field, roof top etc)