Visual Odometry for Non-Overlapping Views Using Second-Order Cone Programming

Jae-Hak Kim 1, Richard Hartley 1, Jan-Michael Frahm 2 and Marc Pollefeys 2

1 The Australian National University / NICTA, 2 University of North Carolina at Chapel Hill

What is the motion of non-overlapping views?

- Non-overlapping view cameras are looking different directions, so they do not share view each other.
- Can we find a motion (R and t) with SCALE of them?
- Visual Odometry Problem

Related Work

- The Pless equation for generalized camera [Pless]
- Linear solution for non-overlapping views [Marc]

We present a method to solve this problem using Second-Order Cone Programming

Problem

- Given m calibrated cameras F_i, i = 1...m and (m ≥ 2)
  \[ F_i = [I | c_i] \]
  where c_i is the centre of the i-th camera.
- Estimate motion M of the system
  \[ M = \begin{bmatrix} R & -Rt \\ 0^T & 1 \end{bmatrix} \]
  where R is a rotation, and t is a translation of the set of cameras.
- The i-th camera matrix is \( F'_i \) = \([R^T | t - c_i] = R^T[I | -R(c_i-t)]\)

Algebraic Derivation (or skip "Geometric Derivation")

An essential matrix is
\[ E_i = R^T [c_i + R(t-c_i)] \cdot I \]
\[ = [R [c_i + (t-c_i)] \cdot R^T] \]
Considering that the decomposition of \( E_i \) is \( E_i = R_i (v_i) \cdot X \),
\[ t = \lambda \cdot R \cdot v_i + c_i - R \cdot c_i \]

Geometric Derivation

- Each essential matrix \( E_i \) constrains the final position of the first camera to lie along a line. These lines are not all the same, in fact unless \( R = I \), they are all different. The problem now comes down to finding the values of \( \lambda \) and \( c_i' \) such that for all \( i \):
  \[ c_i' = c_i + R(c_i - c_i') + \lambda v_i \] for \( i = 1 \ldots m \)
  Having found \( c_i' \), we can get t from the equation \( c_i' = R(c_i - t) \)

A Triangulation Program

- Denoting \( c_i + R(c_i - c_i') \) by \( C_i \), the point \( c_i' \) must lie at the intersection of the lines \( C_i + \lambda \cdot v_i \). In the presence of noise, these lines will not meet, so we need find a good approximation to \( c_i' \).

  The problem of estimating the best \( c_i' \) is identical with the triangulation problem studied in \( L_\infty \) norm.

  To formulation the triangulation problem, instead of \( c_i' \), we write \( X \) as the final position of the first camera where all translations derived from each essential matrix meet together.

  We have \( m \) cones for \( m \) cameras, each one aligned with one of the translation directions. The desired point \( X \) lies in the overlap of all these cones, and, finding this overlap region gives the solution we need in order to get the motion of cameras. Our original motion estimation problem is formulated as:

  \[ \min \ \max \ (||X - C_i|| \cdot v_i)\]

  This problem can be solved as a Second-Order Cone Programming using a bisection algorithm.

Experiments

- Point Grey’s Ladybug™ camera unit consists of six 1024x768 CCS color sensors with small overlap of their field of view is used for real experiments.

  - 139 frames of image are captured by each camera. Feature tracking is performed by the KLT tracker, and the lens distortion is corrected by LadyBug SDK library.

  - Rotations at key frames 0, 30, 57 and 80, and translations

<table>
<thead>
<tr>
<th>Rotation</th>
<th>True rotation</th>
<th>Estimated rotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 0)</td>
<td>85.15</td>
<td>(0.008647, -0.015547, 0.999842)</td>
</tr>
<tr>
<td>(0, 0)</td>
<td>157.0</td>
<td>(-0.022212, -0.008558, 0.999717)</td>
</tr>
<tr>
<td>(0, 0)</td>
<td>134.0</td>
<td>(0.024939, -0.005637, -0.999697)</td>
</tr>
</tbody>
</table>

- Paths (Front and Side view)

  - Paths (Top view and Comparison with Ground Truth)

http://www.nicta.com.au

{Jae-Hak.Kim,Richard.Hartley}@anu.edu.au  {jmf,marc}@cs.unc.edu