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

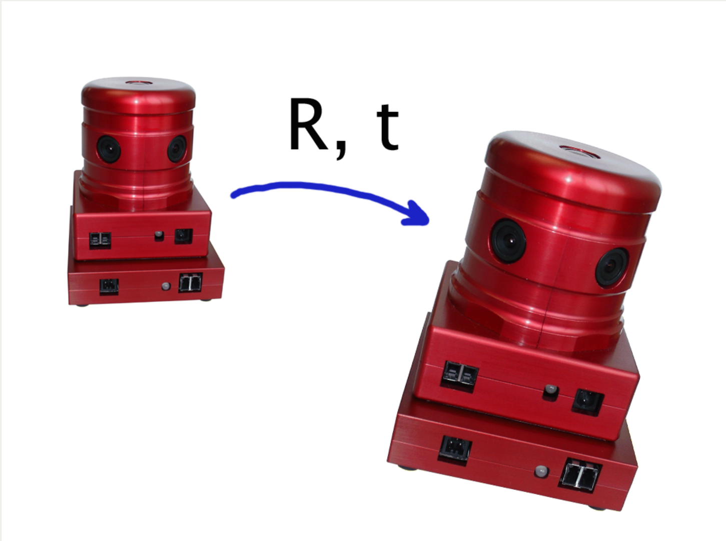
Jae-Hak Kim [1], Richard Hartley [1], Jan-Michael Frahm [2] and Marc Pollefeys [2]

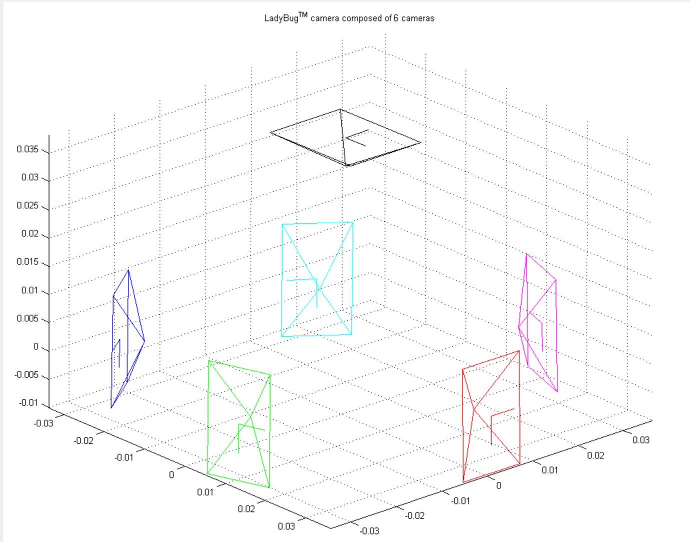
[1] The Australian National University / NICTA, [2] University of North Carolina at Chapel Hill

ACCV, Tokyo, Japan, 22nd Nov 2007

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 ($\mathtt{R}$ and $\mathbf{t}$ 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 $\mathtt{P}_i$, $i = 1...m$ and $(m \geq 2)$

$$\mathtt{P}_i = [\mathtt{I} \mid \mathbf{c}_i]$$

where $\mathbf{c}_i$ is the centre of the i-th camera.

- Estimate motion $\mathtt{M}$ of the system

$$\mathtt{M} = \begin{bmatrix} \mathtt{R} & -\mathtt{R}\mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

where $\mathtt{R}$ is a rotation, and $\mathbf{t}$ is a translation of the set of cameras.

- The i-th camera matrix is $\mathtt{P}'_i = [\mathtt{R}^\top \mid \mathbf{t} - \mathbf{c}_i] = \mathtt{R}^\top[\mathtt{I} \mid -\mathtt{R}(\mathbf{c}_i - \mathbf{t})]$

Algebraic Derivation (or skip to "Geometric Derivation")

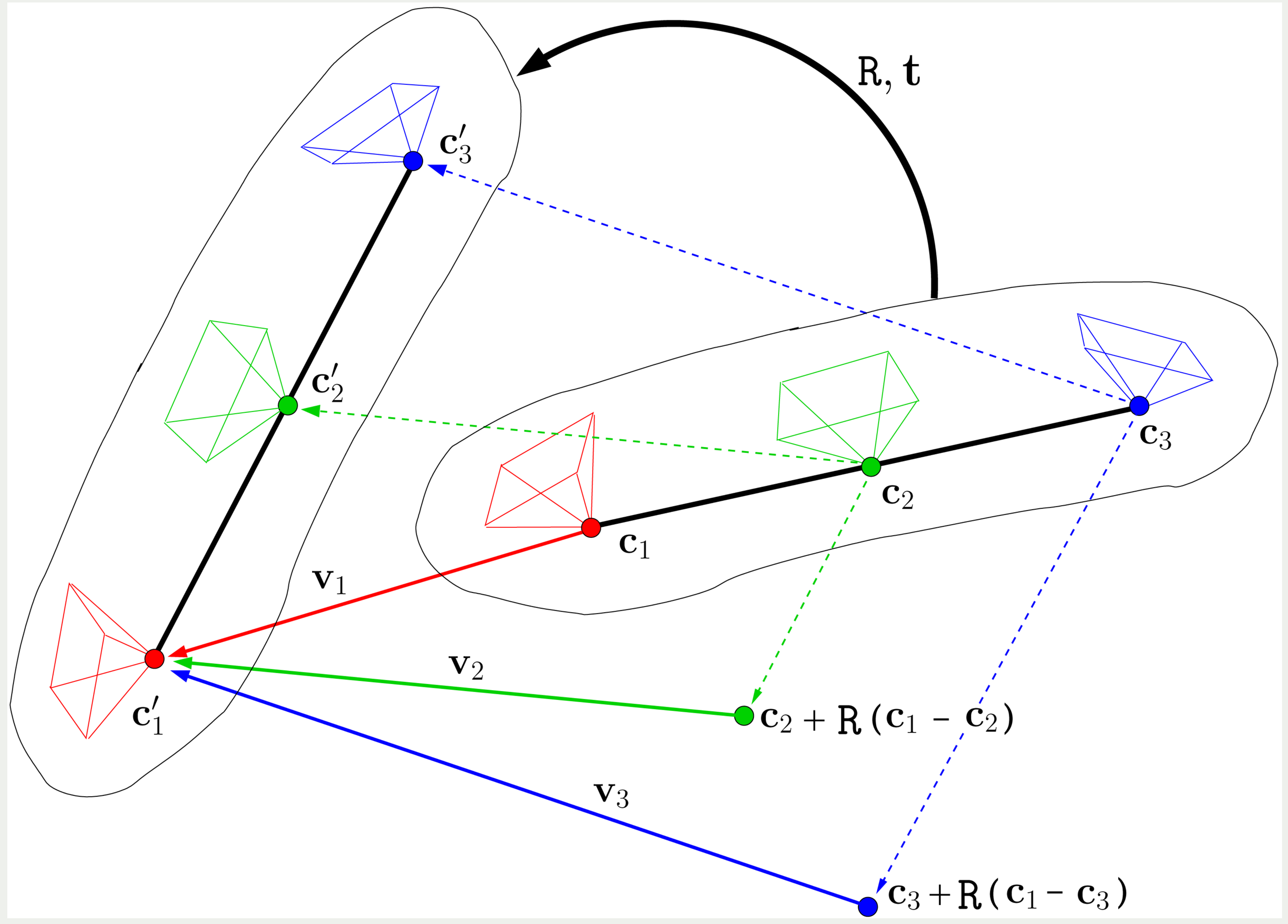
An essential matrix is

$$\begin{aligned}\mathtt{E}_i &= \mathtt{R}^\top[\mathbf{c}_i + \mathtt{R}(\mathbf{t} - \mathbf{c}_i)]_\times \mathtt{I} \\ &= [\mathtt{R}^\top \mathbf{c}_i + (\mathbf{t} - \mathbf{c}_i)]_\times \mathtt{R}^\top\end{aligned}$$

Considering that the decomposition of $\mathtt{E}_i$ is $\mathtt{E}_i = \mathtt{R}_i[\mathbf{v}_i]_\times$,

$$\mathbf{t} = \lambda_i \mathtt{R}^\top \mathbf{v}_i + \mathbf{c}_i - \mathtt{R}^\top \mathbf{c}_i$$

Geometric Derivation



- Each essential matrix $\mathtt{E}_i$ constrains the final position of the first camera to lie along a line. These lines are not all the same, in fact unless $\mathtt{R} = \mathtt{I}$, they are all different. The problem now comes down to finding the values of λ_i and $\mathbf{c}'_1$ such that for all i:

$$\mathbf{c}'_1 = \mathbf{c}_i + \mathtt{R}(\mathbf{c}_1 - \mathbf{c}_i) + \lambda_i \mathbf{v}_i \quad \text{for i=1 ... n}$$

Having found $\mathbf{c}'_1$, we can get $\mathbf{t}$ from the equation $\mathbf{c}'_1 = \mathbf{R}(\mathbf{c}_1 - \mathbf{t})$

A Triangulation Program

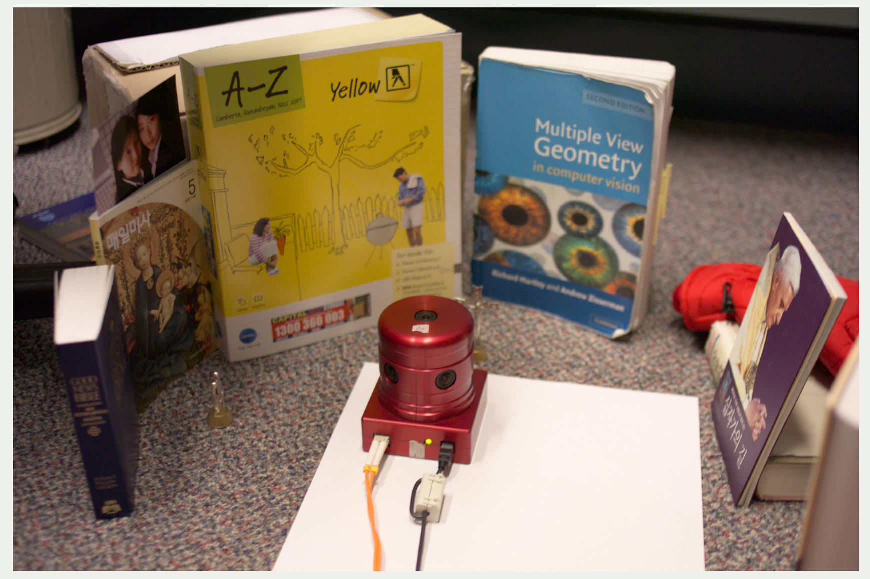
- Denoting $\mathbf{c}_i + \mathbf{R}(\mathbf{c}_1 - \mathbf{c}_i)$ by $\mathbf{C}_i$, the point $\mathbf{c}'_1$ must lie at the intersection of the lines $\mathbf{C}_i + \lambda_i \mathbf{v}_i$. In the presence of noise, these lines will not meet, so we need find a good approximation to $\mathbf{c}'_1$.
- ***The problem of estimating the best $\mathbf{c}'_1$ is idential with the triangulation problem studied in L_∞ norm.***
- To formulation the triangulation problem, instead of $\mathbf{c}'_1$, we write $\mathbf{X}$ 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 one aligned with one of the translation directions. The desired point $\mathbf{X}$ 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:

$$\min_{\mathbf{X}} \max_i \frac{\|(\mathbf{X} - \mathbf{C}_i) \times \mathbf{v}_i\|}{(\mathbf{X} - \mathbf{C}_i)^\top \mathbf{v}_i}$$

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.

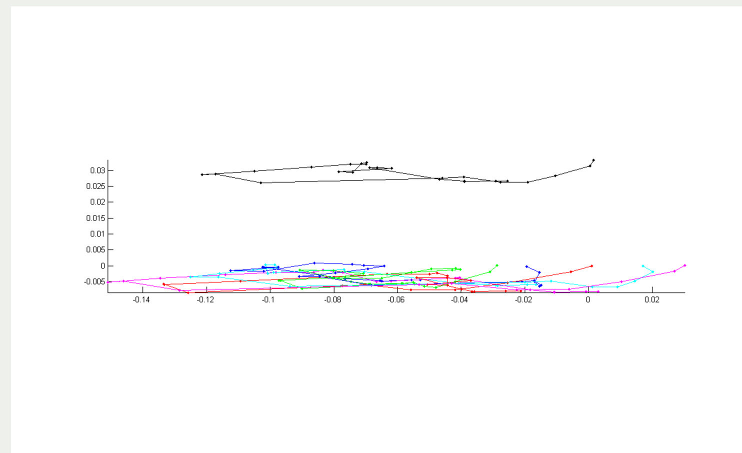


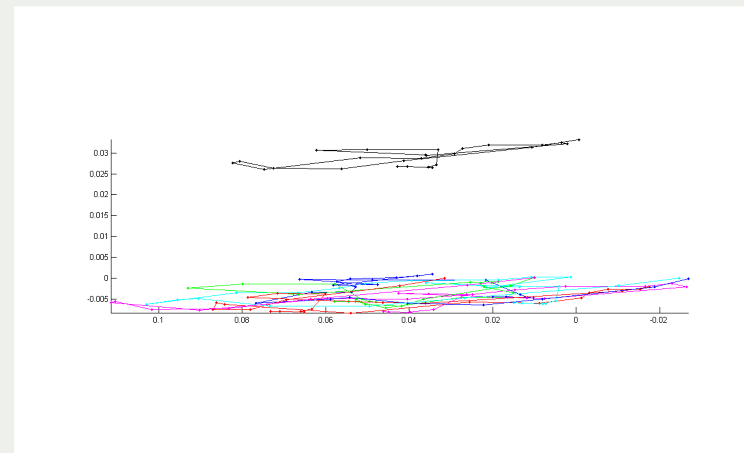
- 139 frames of image are captured by each camera. Feature tracking is performed by the KLT tracker, and the lens distortion is corrected by LadyBug SDK library.
- Rotations at key frames 0, 30, 57 and 80, and translations

Rotation pair	True rotation Axis	True rotation Angle	Estimated rotation Axis	Estimated rotation Angle
(R_0, R_1)	[0 0 -1]	85.5	[-0.008647 -0.015547 0.999842]	85.15
(R_0, R_2)	[0 0 -1]	157.0	[-0.022212 -0.008558 0.999717]	156.18
(R_0, R_3)	[0 0 -1]	134.0	[0.024939 -0.005637 -0.999673]	134.95

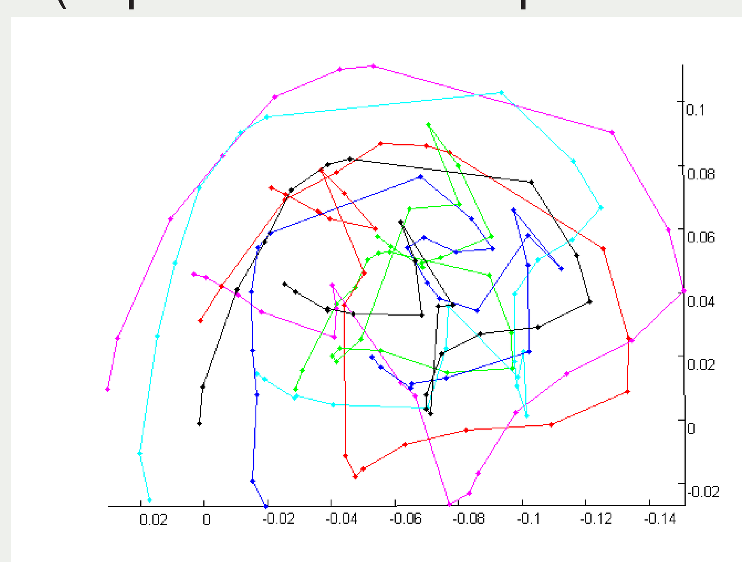
Translation pair	Scale ratio True value	Scale ratio Estimated value	Angles True value	Angles Estimated value
$(\mathbf{t}_{01}, \mathbf{t}_{02})$	0.6757	0.7424	28.5	28.04
$(\mathbf{t}_{01}, \mathbf{t}_{03})$	0.4386	1.3406	42.5	84.01

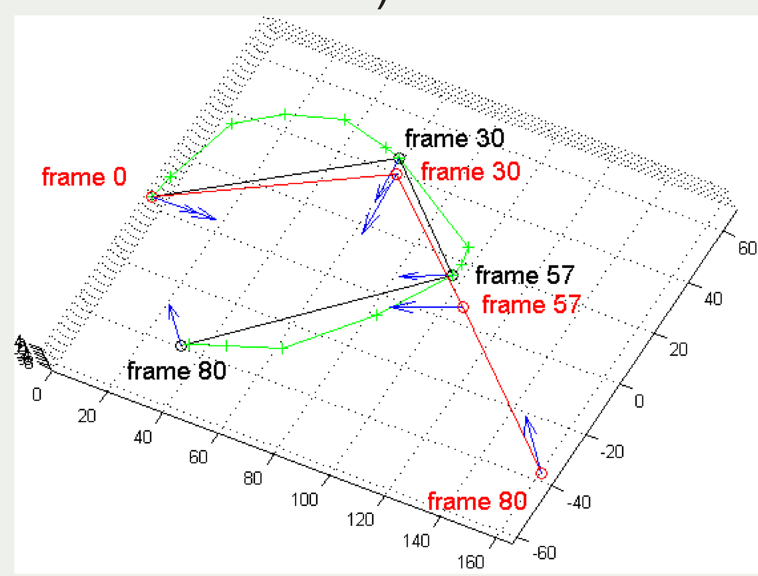
- Paths (Front and Side view)





- Path (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