ROLLING SHUTTER CAMERA TRACKING AND VIDEO RECTIFICATION USING VISUAL AND INERTIAL MEASUREMENTS

Chao Jia and Brian Evans
The University of Texas at Austin

Oct 6 2012
Overview

- Rolling shutter effects
- Previous work using IMUs
- Sensor fusion
  - Feature point correspondence in rolling shutter camera
  - EKF-based sensor fusion
  - Outlier detection using 1-point RANSAC
- Experiments and comparison
Rolling Shutter Effects

• Handheld cameras – fast motion
• CMOS image sensors:
  • Rows in sensor array are exposed sequentially from top to bottom

\[ t(u, i) = t_i + t_r \times \frac{u_y}{h} \]

• Rolling shutter effects:
Rectify Rolling Shutter Effects

• Pose estimation for each row needed
• Pure rotational model
  • Main cause of pose difference between rows
  • Fast rectification
    \[ u \sim KR(t(u, j))x \quad u' \sim KR((t_j)R^T(t(u, j))K^{-1}u\]
• Previous work on pose estimation
  • [Karpenko 2011]
  • Integrate gyro readings (100Hz on Nexus S)
  • Use SLERP for exposure time between samples
  • Unknown bias and noise; Fast changing motion
  
  • [Hanning 2011]
  • EKF-based estimation with accelerometer readings as measurements
  • acceleration = f (pose, gravity)
  • Good loop closing property
  • Gravity is not the only source of acceleration
Proposed Method

• Gyro readings integrated with visual measurements

• Feature point correspondences provide accurate geometric clue
  • Structure from Motion; Simultaneous Localization and Mapping (SLAM)
  • How to relate matched features with high-frequency camera poses in rolling shutter model
  • How to effectively detect outliers

• Research platform
  • Android smartphones (Google Nexus S)
  • Use app “data logger” to record video and gyro readings (with timestamp) at the same time
    • How to synchronize sensor measurements
Gyro and Feature Point Correspondence

- Gyro returns measurements with higher sampling rate

\[
\begin{align*}
\mathbf{R}(t(u', i + 1))\mathbf{R}^T(t(u, i)) &= \prod_{n=k+1}^{k+4} \Delta\mathbf{R}(\omega_n \Delta t_n) \\
\Delta\mathbf{R}(\omega_n \Delta t_n) &= \exp(\skew(\omega_n) \Delta t_n)
\end{align*}
\]
EKF-based Estimation

- State vector: two groups of angular velocities
  \[ \mathbf{x}_i = [\omega(i-1, 1), \ldots, \omega(i-1, N_{i-1}), \omega(i, 1), \ldots, \omega(i, N_i)]^T \]

  - Why two groups?
    - The visual measurements depend on both the group of the current frame and the group of the previous frame
  - Why angular velocity instead of rotation representation (unit quaternion)
    - Equivalent for relative rotation estimation
    - No SLERP needed (simple Jacobian in EKF)

- Probabilistic Graphical Model

\[ \begin{align*}
  &y_1 \\
  &y_2 \\
  &y_3 \\
  &x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow \ldots \\
  &z_2 \\
  &z_3
\end{align*} \]
Dynamic Motion Model (State Prediction)

- Group cloning in prediction

\[
x_i = \begin{bmatrix} x_{i,1} \\ x_{i,2} \end{bmatrix} = \begin{bmatrix} x_{i-1,2} \\ y_i \end{bmatrix} + \begin{bmatrix} 0 \\ w_i \end{bmatrix}
\]

- Linear model

\[
A_i = \left. \frac{\partial f}{\partial x} \right|_{x_{i-1}} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}, \quad W_i = \left. \frac{\partial f}{\partial w} \right|_{w_i} = \begin{bmatrix} 0 \\ I \end{bmatrix}
\]
Measurement Model (State Update)

- Use only feature points in current frame as the measurements; their matching points in previous frame are used as parameters

\[
\mathbf{u}_{i,j} = g \left( K \Delta R K^{-1} \begin{bmatrix} \mathbf{u}_{i-1,j} \\ \mathbf{v}_{i-1,j} \end{bmatrix} \right) + \mathbf{v}_{i,j}
\]

- Final measurement equation for state update
  - Closed form Jacobian matrix
    using chain rule

\[
\mathbf{z}_i = \begin{bmatrix} \mathbf{u}_{i,1} \\ \mathbf{u}_{i,2} \\ \vdots \\ \mathbf{u}_{i,M} \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}_i, \mathbf{u}_{i-1,1} - \mathbf{v}_{i-1,1}) + \mathbf{v}_{i,1} \\ h_2(\mathbf{x}_i, \mathbf{u}_{i-1,2} - \mathbf{v}_{i-1,2}) + \mathbf{v}_{i,2} \\ \vdots \\ h_M(\mathbf{x}_i, \mathbf{u}_{i-1,M} - \mathbf{v}_{i-1,M}) + \mathbf{v}_{i,M} \end{bmatrix}
\]
Measurement Model (State Update)

- State prediction of current stage is **correlated** with the noise of the observation (measurements) in previous stage

- Solution: augment the state vector with the measurement noise

\[ u_{i,j} = g \left( K_\Delta R K^{-1} \begin{bmatrix} u_{i-1,j} - v_{i-1,j} \\ 1 \end{bmatrix} \right) + v_{i,j} \]
Outliers Removal

• For global shutter model
  • Epipolar constraint / Homography fitting

• 1-point RANSAC in EKF
  • In standard RANSAC, each hypothesis need the minimum number of points necessary to estimate the parameters
  • For EKF, we have got a prior distribution of the parameters (state vector) by prediction
  • The minimum number of points to estimate the parameters can be reduced to one (in the rolling shutter case we choose three)
Outlier Removal

- Track features using the state prediction result, refine by KLT
- \text{inliers} = [ ]
- for \( i = 1 \) to \( N_{\text{hyp}} \)
  - randomly choose 3 matches
  - update the states through EKF filtering
  - computer the re-projection error (innovation) and choose \text{current_inliers}
  - if \( \text{num}_{\text{current_inliers}} > \text{max}_{\text{num}} \)
    - \text{inliers} = \text{current_inliers}
    - \text{max}_{\text{num}} = \text{num}_{\text{current_inliers}}
  - end
- Use \text{inliers} to compute the EKF update

75ms/frame in Matlab implementation on 2.3GHz Intel i5 processor
20 features tracked
Sensor Synchronization and Calibration

- Parameters
  - Rolling shutter speed (actual exposure time)
  - Intrinsic parameters of the camera
  - Delay between timestamps of IMUs and video
- Batch optimization [Karpenko 2011]
  - Camera intrinsic parameters initialized by Zhang’s self-calibration
  - Get relative rotation from gyro readings
  - Optimize over all matching points, minimize the average re-projection error
  - Solve by Levenberg-Marquardt algorithm

\[
J = \sum_{i} \sum_{u,v} \|u - KR(t(u, i + 1))R^T(t(v, i))K^{-1}v\|^2
\]

convergence of synchronization & calibration
Average Re-projection Error per Point

- RANSAC EKF vs. Integrating gyro readings
Rotation Estimation Accuracy

- **Zero-angle test**
  - Start with cell phone on a flat surface
  - Rotate cell phone at will, then put it back on the surface and stay still for several seconds
  - Repeat for ten times
  - Ground truth available naturally for pitch and roll

![Graph showing rotation estimation accuracy with x, y, and z axes marked](image)
Rotation Estimation Accuracy

- Using raw gyro readings (with bias)
Rotation Estimation Accuracy

- Using raw gyro readings (with bias)
Rotation Estimation Accuracy

- Using unbiased gyro readings
Rotation Estimation Accuracy

- Using unbiased gyro readings

![Roll angle estimation error (rad)](chart)
Rotation Estimation Accuracy

- Using unbiased gyro readings
Rotation Estimation Accuracy

- Using unbiased gyro readings

![Pitch angle estimation error (rad)](chart.png)
Rolling Shutter Rectification
Rolling Shutter Rectification

gyro

gyro + cam
Rolling Shutter Rectification

gyro

gyro + cam
Rolling Shutter Rectification

gyro

gyro + cam
Rolling Shutter Rectification

• Complete failure using gyro + accel
Numerical Comparison

- No ground truth → no-reference method
- Vanishing point check
  - Lines detected manually
  - Find vanishing point by least-square

Average Euclidean distance from the lines to the vanishing point (in pixel)