Visual Odometry System Using Multiple Stereo Cameras and Inertial Measurement Unit

Taragay Oskiper, Zhiwei Zhu, Supun Samarasekera, Rakesh Kumar

Kevin Kwan
CSCE 643
Outline

• Introduction
• Pose Transfer Mechanism
• Multicamera Visual Odometry Algorithm
• Inertial Measurement Unit Integration
• Results
• Conclusion
Introduction

- Localization in GPS denied environments
  - Wide variety of potential applications, from navigation to tracking military units
- Increased interest in inertial based passive sensors coupled with cameras
Previous Work

• Video streams from 1 or 2 cameras
• Landmark based 3D motion tracking systems

• These systems are not robust enough for autonomous use over large distances and time periods.
Their Approach

- Forward and rear facing pair of cameras
- Inertial Measurement Unit

Figure 1. Front and back views of the backpack system and sample images captured as the person enters a room.
Outline

• Introduction
• Pose Transfer Mechanism
• Multicamera Visual Odometry Algorithm
• Inertial Measurement Unit Integration
• Results
• Conclusion
Pose Transfer Mechanism

• Visual odometry is applied to each pair individually to estimate the left camera pose.

• Calibration will determine the extrinsic relation between the left and right cameras and the relation between the front and rear left camera.
Pose Transfer

• The visual odometry algorithm will tell us the pose of both pairs of cameras between $t_k$ and $t_{k+1}$.

  - $X_1(t_{k+1}) = R_1(t_k,t_{k+1})X_1(t_k) + T_1(t_k,t_{k+1})$

• $R(t_k,t_{k+1})$: rotation matrix

• $T_1(t_k,t_{k+1})$: translation vector

• $X_1(t_k)$: camera point expressed in the camera coordinate frame at $t_k$. 
Pose Transfer

• Transformation can also be expressed as
  \[ X_1(t_{k+1}) = P_1(t_k, t_{k+1}) X_1(t_k) \]

• where \( P_1 \) is the pose for the left camera of the front pair at time \( t_{k+1} \) at the coordinate frame of the left camera at time \( t_k \)
Pose Transfer

- \[ X_2(t) = P_{12}X_1(t) \]
- \( P_{12} \) describes the relation of the pose of the left camera of the back pair relative to the left camera of the front pair
Pose Transfer

• Then, \( P_2(t_k, t_{k+1}) = P_{12} P_1(t_k, t_{k+1}) P_{12}^{-1} \)

\[
P_2(t_k, t_{k+1}) = P_{12} P_1(t_k, t_{k+1}) P_{12}^{-1}
\]

• Camera poses in the front pair’s left camera coordinate frame can be transferred to the back pair’s left camera coordinate frame
Figure 2. Rig with two stereo pairs moving in 3-D space between time instants $t_k$ and $t_{k+1}$. Left and right cameras are denoted by letters 'L' and 'R' in each pair. In the multicamera visual odometry algorithm, given $P_{12}$, the fixed relative pose between the two left cameras of the front and back pairs, and $P_1(t_k, t_{k+1})$, the relative pose of the left camera of the front pair between two time instants, we can determine $P_2(t_k, t_{k+1})$ and vice versa. On the other hand, for the automatic calibration procedure, given many pairs of highly accurate poses $P_1(t_k, t_{k+1})$ and $P_2(t_k, t_{k+1})$ for $0 \leq k \leq N$, one can determine the $P_{12}$ which makes these pose pairs agree the most according to some criteria.
Outline

• Introduction
• Pose Transfer Mechanism
• Multicamera Visual Odometry Algorithm
• Inertial Measurement Unit Integration
• Results
• Conclusion
Multicamera Visual Odometry Algorithm

- Detect and match Harris corner based feature points in each pair
- Normalize using intrinsic calibration parameters in each camera
- Compute 3D locations based on these feature points
Multicamera Visual Odometry Algorithm

- Match feature points from old and new image pairs
- Compute pose of left camera in each pair using a robust resection method based on RANSAC
- Minimize the robust cost function of the reprojection errors in the left and right images
Cost Function

\[ c_j(P^j_k) = \sum_{i=1}^{K_j} \rho(x^l_i - h(P^j_k X^j_i)) + \rho(x^r_i - h(P^{s_j} P^j_k X^j_i)) \]

- \( j \) is either the front or back pair of cameras
- \( K_j \) is the number of feature points
- \( x_i \): coordinates of a specific feature point \( i \)
- \( X_i \): 3D position in homogeneous coordinates
- \( p^{s_j} \): pose of the right camera
- \( h \): converts from homogeneous to inhomogeneous coordinates
- \( \rho \): the robust cost function
Visual Odometry algorithm

- All steps are computed for each pair of cameras independently.
- To calculate the winning pose out of the 2 candidates: $d(P_k^1, P_k^2) = \begin{cases} 1 & \text{if } c_1(P_k^1) < c_2(P_k^2) \\ 2 & \text{otherwise} \end{cases}$
Preemptive RANSAC

• Each camera generates 500 pose hypotheses based on 3 random point correspondences
• Evaluate each hypotheses using a 100 point data set from each camera
• Obtain cumulative score for each hypothesis.
• Discard the least scoring half from each camera and reevaluate the remaining half on another 100 points.
• Repeat until there is only 1 hypothesis left for each camera
Outline

• Introduction
• Pose Transfer Mechanism
• Multicamera Visual Odometry Algorithm
• Inertial Measurement Unit Integration
• Results
• Conclusion
Inertial Measurement Unit

- What if both front and back pairs provide bad poses?
- Integrate visual odometry with (MEMS) Microelectromechanical Systems based IMU using extended Kalman filter framework.
Measurement Model

\[ v_{vo}^k = v_k + n_{v,k} \]
\[ \omega_{vo}^k = \omega_k + n_{w,k} \]
\[ \omega_{imu}^k = \omega_k + b_k + n_{w,k}^{imu} \]
\[ a_{imu}^k = R(q_k)g + n_{a,k}^{imu} \]

- \( v_{vo} \): translational velocity from visual odometry
- \( \omega_{vo} \): angular velocity
- \( \omega_{imu} \): gyro outputs from the IMU
- \( a_{vo} \): accelerometer output
Inertial Measurement Unit

• Compute the difference in velocities in all three rotational axes and compare to a threshold.
• If difference is less than the threshold, remove the gyro observations from the measurement model.
Inertial Measurement Unit

• Goal: rely on visual odometry for the majority of the trip.
• Only use IMU in those brief instances where visual odometry is unreliable.
Figure 3. Flow diagram for multicamera visual odometry and IMU integration in our system.
Outline

• Introduction
• Pose Transfer Mechanism
• Multicamera Visual Odometry Algorithm
• Inertial Measurement Unit Integration
• Results
• Conclusion
Results

• 4 digital cameras
• Gray-scale images at 640x480 resolution
• 530 meter outdoor and indoor sequence

Figure 6. Trajectory obtained by our system from a 530 meter long outdoor/indoor sequence overlaid in 'blue' on a map. Loop closure error is 3.9 meters.
Results

- Auto calibration:
  - 2411 pose estimates
  - $T = [0.3345 \ 0.667 \ -0.1138]$
    - $T$ is the translation vector
  - Roll, pitch, yaw = -0.9267, -30.2715, -179.6150 degrees
  - Average error: 3.29 millimeters

Figure 4. Histogram of the sequence alignment errors $\|\epsilon_k\|$, $0 \leq k \leq 2392$, based on the pose solution $P_{12}$ obtained by the auto-calibration algorithm.
Results

- Blue: IMU
- Green: filter output
- Red: visual odometry
Results: Each Pair By Itself

Figure 7. Visual odometry trajectories obtained by the individual stereo pairs alone. The regions where gross errors, ”breaks”, occur are highlighted and some thumbnail images corresponding to those moments are shown.
Results: Indoor and Stairs

Figure 8. Trajectory obtained by our system from a 264 meter long indoor sequence that includes staircase climbing. Loop closure error is 2.1 meters.
Outline

- Introduction
- Pose Transfer Mechanism
- Multicamera Visual Odometry Algorithm
- Inertial Measurement Unit Integration
- Results
- Conclusion
Conclusion

• Their results showed that their implementation can work over long distances and time periods.

• Future work
  • Continue integrating IMU with visual odometry
  • Miniaturizing configuration
  • Helmet mounted cameras