
This version is available at https://strathprints.strath.ac.uk/57125/

Strathprints is designed to allow users to access the research output of the University of Strathclyde. Unless otherwise explicitly stated on the manuscript, Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Please check the manuscript for details of any other licences that may have been applied. You may not engage in further distribution of the material for any profitmaking activities or any commercial gain. You may freely distribute both the url (https://strathprints.strath.ac.uk/) and the content of this paper for research or private study, educational, or not-for-profit purposes without prior permission or charge.

Any correspondence concerning this service should be sent to the Strathprints administrator: strathprints@strath.ac.uk
Visual Pose Estimation and Identification for Satellite Rendezvous Operations

Mark A. Post and Xiu T. Yan

Space Mechatronic Systems Technology Laboratory
Dept. of Design, Manufacture & Engineering Management
University of Strathclyde, 75 Montrose St. Glasgow, U.K.

For the Sixth China-Scotland SIPRA workshop
“Recent Advances in Signal and Image Processing”
University of Stirling, Scotland, UK.
31 May 2015 16:35, Lomond Room, Stirling Court Hotel
Overview

1) Introduction

2) Triangulation & Reconstruction

3) Correspondence Recognition

4) CubeSat Identification Results

5) Conclusions & Future Directions
Introduction
Visual Pose Estimation & Rendezvous

- Automated rendezvous & docking with a target
- Small satellite (CubeSat or inspection robot)
- Close range, slow inertial movements assumed
- Monocular visual method
  - Sensing without specialized Radar or Lidar hardware

YUSend Nanosatellite (Credit: York University)

SPHERES with VERTIGO vision system (Credit: MIT Space Systems Laboratory)

NASA Mini-AERCam (Credit: NASA)
Steps for Visual Identification

1) Approach
   – Recognize that “something” is there

2) Track
   – Follow the object to identify relative motion

3) Observe
   – Build up additional information on the object

4) Identify
   – Match the object with a model to determine pose
Feature Tracking & Pose Estimation

- Detect visible features from a sequence of 2-D images
- Build up a feature cloud of the scene in 3-D over many images
- Recognize the scene or a part of the scene from a model
- Estimate the pose of what is recognized for rendezvous

Multiple Images → Approach & Identify Features

Track Features & Triangulate → Cloud Creation

Observe to build a more complete feature cloud

Comparison To Model → Identify Target & Target Pose from model

SMeSTech
Space Mechatronic Systems Technology

SIPRA 2015
Triangulation & Reconstruction
Multiple-View Geometry (SfM)

Approach and Localize

Ideally <1s per frame!

Tracking and Identification

Estimated Camera Poses

Point Cloud of Target Object

SMeSTech
Space Mechatronic Systems Technology

SIPRA 2015
Feature Detection

Features are based on a patch \( p \) and many kinds are available:

- SIFT (patented)
- SURF (patented)
- ORB (Oriented BRIEF)

We use ORB (Rublee et al, 2011), with orientation “steering” from

\[
F = R_f \begin{pmatrix} a_1 & \cdots & a_n \\ b_1 & \cdots & b_n \end{pmatrix}
\]

ORB algorithm uses FAST corners by intensity centroid to speed matches

\[
C = \left( \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad \text{where} \quad m_{pq} = \sum_{x,y} x^p y^q I(x,y)
\]

and BRIEF keypoint descriptors (Calonder et al, 2010) described from intensity \( p(a) \) at \( a \):

\[
\tau(p; a, b) = \begin{cases} 
1 & : p(a) < p(b) \\
0 & : p(a) \geq p(b)
\end{cases}
\]

\[
f_n(p) = \sum_{1 \leq i \leq n} 2^{i-1} \tau(p; a_i, b_i)
\]

\[
g_n(p, \theta) = f_n(p) \lor (a_i, b_i) \in F
\]
• Feature points are matched between successive images with FLANN (Muja & Lowe, 2009)

\[ M_g = M_f(a)|d_a < d_{max}/2 \]

• Fundamental matrix \( F \) found by least-squares or RANSAC

\[ a_i^T Fa_i = 0, \quad i = 1, \ldots, n \]

• Essential matrix \( E \) is \( F \) with calibration:

\[ E = K^T FK \]

• Rotation \( R \) and translation \( t \) matrices from SVD of \( E \) (Hartley & Zisserman, 2004)

– 4 Combinations of factorizations:

\[ R = UW^T V^T \quad R = UWV^T \]
\[ t = U(0, 0, 1)^T \quad t = -U(0, 0, 1)^T \]

• Least-Squares triangulation finds 3D points by iterative solution

• Locate camera (PnP solution)

• Bundle Adjustment (optional)
Image Choice for Triangulation

Features Tracked Forward Between Closely-Spaced Images

Triangulation Performed Back Between Widely-Spaced Images

- Txform camera: $C_w(t) = [R_w(t - 1)R(t)|T(t) + T_w(t - 1)]R(t)$
- Txform points: $x' = (R_w(t - 1)R(t))^T x + (T(t) + T_w(t - 1))R_w(t - 1)$
Correspondence & Recognition
Correspondence Grouping

• For matching, the normals $N$ of the point cloud are obtained

• A set of keypoints are chosen & given 3D SHOT descriptors $D$ (Signature of Histograms of OrienTations: Salti, Tombari, Stefano, 2014)

• Cosine function with $N$: $\cos(\theta) = f(N_p, N_q)$

• As dot product: $f(N_p, N_q) = N_p \cdot N_q$

• FLANN search again used to find corresponding keypoints between Scene & Model
Correspondence Grouping

- BOrder Aware Repeatable Directions (BOARD) algorithm used to calculate local reference frames for each descriptor.
- Clustering is performed by pre-computed Hough voting (Tombari and Stefano, 2010)
  - Model (offline): \( V_{i,L}^M = [L_{i,x}^M, L_{i,y}^M, L_{i,z}^M] \cdot (C^M - F_i^M) \)
  - Scene (online): \( V_{i,G}^S = [L_{j,x}^S, L_{j,y}^S, L_{j,z}^S] \cdot V_{i,L}^M + F_j^S \)
- Estimated pose has the largest number of correspondence votes.

Matched Possible Poses of Model

Model (pre-loaded and high resolution)

Scene (current, sparse and noisy)
CubeSat Identification

Results
Testing - CubeSat Image Sequences

• Monocular resolution of 640x480 (VGA)
• Rotation and translation
• No background features (assumed to be filtered)
• 1U and 3U CubeSat engineering models
• Slow capture movement, one direction
Sequential Triangulation

Final Target Cloud:
Relative Target Motion
Pose Estimation Accuracy

RMS Error X: 7mm  Y: 8mm  Z: 7mm

RMS Error X: 0.14rad  Y: 0.11rad  Z: 0.19rad
Correspondence: Dense Scene

6524 Model Points, 5584 Scene Points (from 220 images)

Test 1: Descriptor Radius 0.05, Cluster Size 0.1: 167 points, 63 matches

Test 2: Descriptor Radius 0.1, Cluster Size 0.5: 632 points, 594 matches
Correspondence: Sparse Scene

6524 Model Points, 1816 scene points (from 32 images)

Test 3: Descriptor radius 0.05, cluster size 0.1: 77 points, 28 matches

Test 4: Descriptor radius 0.1, cluster size 0.5: 77 points, 70 matches
### Timing

Time taken in seconds, for 667MHz ARM-Cortex A9

Point Cloud Generation (mean time for one pose estimate)

<table>
<thead>
<tr>
<th>Test</th>
<th>Feature Detection</th>
<th>Feature Matching</th>
<th>Feature Selection</th>
<th>Fundamental Matrix</th>
<th>Essential Matrix</th>
<th>Triangulation</th>
<th>PnP RANSAC</th>
<th>Ego-Motion</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>0.12</td>
<td>0.058</td>
<td>0.015</td>
<td>0.083</td>
<td>0.0017</td>
<td>0.038</td>
<td>0.0033</td>
<td>0.0005</td>
<td>0.32</td>
</tr>
<tr>
<td>3-4</td>
<td>0.12</td>
<td>0.061</td>
<td>0.010</td>
<td>0.048</td>
<td>0.0014</td>
<td>0.025</td>
<td>0.0026</td>
<td>0.0004</td>
<td>0.27</td>
</tr>
</tbody>
</table>

Correspondence Grouping (mean time for one correspondence)

<table>
<thead>
<tr>
<th>Test</th>
<th>Model Normals</th>
<th>Scene Normals</th>
<th>Model Sampling</th>
<th>Scene Sampling</th>
<th>Model Keypoints</th>
<th>Scene Keypoints</th>
<th>FLANN Search</th>
<th>Clustering</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.17</td>
<td>0.15</td>
<td>0.027</td>
<td>0.020</td>
<td>1.26</td>
<td>0.84</td>
<td>107.7</td>
<td>0.92</td>
<td>112.1</td>
</tr>
<tr>
<td>2</td>
<td>0.17</td>
<td>0.15</td>
<td>0.029</td>
<td>0.024</td>
<td>3.37</td>
<td>2.19</td>
<td>118.0</td>
<td>2.00</td>
<td>127.2</td>
</tr>
<tr>
<td>3</td>
<td>0.17</td>
<td>0.043</td>
<td>0.031</td>
<td>0.0083</td>
<td>3.31</td>
<td>0.37</td>
<td>42.5</td>
<td>0.63</td>
<td>48.4</td>
</tr>
<tr>
<td>4</td>
<td>0.17</td>
<td>0.041</td>
<td>0.031</td>
<td>0.0078</td>
<td>3.31</td>
<td>0.37</td>
<td>42.6</td>
<td>1.36</td>
<td>49.1</td>
</tr>
</tbody>
</table>
Correspondence: Accuracy

2042 model points, 1753 scene points (from 52 images)

Test 5: Descriptor Radius 2.0, Cluster Size 1.0

1% Translation Error, 2% Rotation Error

Test 6: Descriptor Radius 2.0, Cluster Size 0.1

7% Translation Error, 3% Rotation Error

Test 7: Descriptor Radius 0.2, Cluster Size 1.0

3% Translation Error, 4% Rotation Error
Correspondence: Partial Shadowing

2042 model points, variable scene points (from 52 images)

Test 8: Scene 25% in shadow: 1254 Scene Points

4% Translation Error, 9% Rotation Error

Test 9: Scene 50% in shadow: 989 Scene Points

8% Translation Error, 21% Rotation Error

Test 10: Scene 75% in shadow: 547 Scene Points

No Shape Correspondence Found
Discussion of Results

• Scene requires time to develop and process
  – Slower movement = more points = higher accuracy
  – Not every image used

• Can use two, three, or more cameras to increase accuracy (known baseline)

• Quality of results depends on image choice & parameters

• Increase descriptor sizes:
  – More keypoints used
  – Better accuracy
  – Longer processing time

• Increase cluster sizes:
  – More precise matching
  – Less choices for pose
  – Optimal value needed

• FLANN search takes 90% of current processing times
  – High-value candidate for hardware acceleration
Conclusions & Future Directions
Conclusions

• We have presented a method for close-range small satellite Visual Identification and Tracking
• Features implemented using OpenCV Libraries
• Correspondence using Point Cloud Library (PCL)
• Feature Detection and Point Cloud Generation takes time, and could be accelerated further
• Hardware acceleration for FLANN & keypoints may help

Critical factors for good results:
• Sharpness of image
  – good focusable optics
  – limited exposure time
• Consistency of exposure
  – Can automate to linearize image values
• Speed of processing
  – frequent frame updates essential
Future Work

• Improving Robustness
• Removal of background features from clouds
• Evaluation of sources of error and responses
• FPGA Acceleration
• Quality of optics
DSP-Based Vision System

- Board based on open designs of Surveyor SRV-1 and LeanXCam
- ADI Blackfin BF537 DSP provides optimized fixed-point processing
- Onboard processing for keypoints & FLANN
- OpenCV and uCLinux
  - fixed point code needed
  - efficient, but limited in resolution and fidelity

![Image of DSP-Based Vision System](image-url)
Thank You!

Any Questions?

Dense reconstruction courtesy of C. Wu's VSFM and Y. Furukawa's CMVS