Development of Real-Time Object Tracking algorithm for UAVS

DEVSHREE KUMAR, (SCIENTIST, MAV UNIT, NAL, BENGULURU)

SUVARNA AGARWAL (M.TECH STUDENT)
Object tracking is one of the most sought out problems in computer vision these days.

The algorithm needs to be robust enough so that it is not restricted to a particular object.

For object tracking basically four type of algorithms are used. Either individually or in combination with each other.

- Feature based tracking
- Template matching (Pattern matching)
- Color detection and tracking (Histogram matching)
- Edge based detection
• In this project, feature based tracking is taken into account and SURF (Speeded-Up Robust Features) algorithm is used.

• SURF is a feature matching algorithm and MATLAB provides various commands to use this algorithm.

• The three main steps involved in the algorithm are as follows and also the commands provided by MATLAB to perform the function:
  ➢ Detection \( \text{detectSURFFeatures} \)
  ➢ Description \( \text{extractFeatures} \)
  ➢ Matching \( \text{matchFeatures} \)

• SURF is scale invariant, rotation invariant and also translation invariant.
For detection,
SURF uses integral form of the image and Hessian matrix as detector.

\[ I_e(x) = \sum_{i=0}^{i_{\text{max}}} \sum_{j=0}^{j_{\text{max}}} I(i,j) \]

\[ H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{yx}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \]

For description,
SURF uses Haar wavelets in both X and Y directions to assign the orientation to a detected point. Using this help in making the feature rotational invariant.

For matching,
Surf uses the previously extracted descriptors and generate matching pairs in both images.

Affine transformation,
After matching, to calculate the centroid and bounding box positions, a geometrical transformation, affine, is used. Using affine transformation the object image is transformed according to the scene image and hence centroid and bounding box positions are calculated.

\[
\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = A \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + B
\]

Here, A and B are defined according to the translation, rotational and scalar effects.
PROCESS FOR OBJECT TRACKING IN VIDEOS USING SURF ALGORITHM

1. LOAD THE VIDEO/CONNECT THE CAMERAS
2. EXTRACT A FRAME/CAPTURE A FRAME
3. SELECT THE OBJECT OF INTEREST
4. DETECT/EXTRACT THE OBJECT FEATURES USING SURF
5. PASS A FRAME OF THE VIDEO

- DRAW BOUNDING BOX AND CENTROID IN THE FRAME
- MATCH THE FEATURES OF OBJECT AND THE FRAME
- DETECT/EXTRACT FEATURES OF THE FRAME USING SURF
- YES
- NO
- SKIP THE FRAME
Firstly, the object to be tracked needs to be selected.

Here, user can select the object from the video stream itself.

For further processing the image format is converted from RGB to grayscale.

Using Computer vision system toolbox in MATLAB, the SURF features in the object image are detected and descriptors are assigned.
For the next step the video stream is converted into frame sequence. Again, for processing the RGB format is converted into grayscale. Using same computer vision system toolbox in MATLAB, the SURF features in each frame is detected and descriptors are assigned. The number of features vary with each frame and not getting enough features can lead to certain errors.
Finally, the matching is done between the object features and each frame.

The matching is done in two stages. First, matching of all the features then removal of outliers.

For the frames not having enough features, are skipped and next frame is taken for processing.

The location of the bounding box and centroid is then calculated and plotted on the video frame.
GUI IMPLEMENTATION OF SURF ALGORITHM (MATLAB)

1. Browse the object image
2. Extract the frames of the video
3. Load a frame of the video
4. Detect features in both object image and frame
5. Match all the features detected
6. Remove outliers
7. Show the detected object using bounding box and centroid
CASE: 1 STATIC AND NON-DEFORMATIVE OBJECT DETECTION AND TRACKING (NON-REALTIME VIDEO)

SELECTED OBJECT

VIDEO FRAMES

DETECTED OBJECT IN VARIOUS FRAMES

Bounding box from SURF algorithm

Bounding box from other algorithm
CASE: 2 MOVING AND NON-DEFORMATIVE OBJECT DETECTION AND TRACKING (NON-REALTIME VIDEO)

SELECTED OBJECT

VIDEO FRAMES

DETECTED OBJECT IN VARIOUS FRAMES
CASE:3 MOVING AND DEFORMATIVE OBJECT DETECTION AND TRACKING (NON-REALTIME VIDEO)

SELECTED OBJECT

VIDEO FRAMES

DETECTED OBJECT IN VARIOUS FRAMES
CASE: 4 REAL-TIME OBJECT DETECTION AND TRACKING

SELECTED OBJECT

PARTIAL OCCLUSION HANDLED WHILE PERFORMING OBJECT TRACKING

DETECTED OBJECT IN VARIOUS FRAMES
CASE:5 REAL-TIME FACE DETECTION AND TRACKING

SELECTED FACE

DETECTED FACE IN VARIOUS FRAMES
## Run-time of the Tracking Function Used in Various Videos for Object Tracking

<table>
<thead>
<tr>
<th>CASE</th>
<th>Number of Frames</th>
<th>Number of Matched Frames</th>
<th>Outlier Features of Scene</th>
<th>Outlier Features of Object</th>
<th>Inlier Matched Points</th>
<th>Function Time for Single Frame (S)</th>
<th>Total Function Time (S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>41</td>
<td>22345</td>
<td>540</td>
<td>272</td>
<td>0.092</td>
<td>3.464</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>100</td>
<td>49152</td>
<td>8000</td>
<td>1905</td>
<td>0.099</td>
<td>6.688</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>85</td>
<td>39241</td>
<td>3312</td>
<td>613</td>
<td>0.096</td>
<td>10.693</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>79</td>
<td>59109</td>
<td>28576</td>
<td>2269</td>
<td>0.136</td>
<td>12.608</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>70</td>
<td>31856</td>
<td>4680</td>
<td>717</td>
<td>0.113</td>
<td>6.511</td>
</tr>
</tbody>
</table>
CONVERSION OF USER-DEFINED MATLAB FUNCTION IN .DLL USING MATLAB COMPILER

2 .dlls are created which are then added to Visual C# as follows

- Adding References in Visual C#
  - MWArray.dll
  - *.dll
- Adding MATLAB libraries in Visual C#
  - MathWorks.MATLAB.NET.Array
  - MathWorks.MATLAB.NET.Utility
  - MATLAB created function
MATLAB .dll IMPLEMENTATION IN VISUAL C#

Object Selection

Windows Form in Visual C#

Object Image

Wrapper class for MATLAB function

MWNumeric Array

Class Name

Scene Image

Scene Features (x,y)
newBoxPolygon Points
New Centroid Points

Windows Form in Visual C#

Object Detection

Images

Work-flow of MATLAB implementation in Visual C#

Object detection using SURF in Visual C#

MATLAB function importing time

Detected SURF feature points
ONGOING WORK

START TRACKING

GIMBAL CORRECTION

MANUAL

JOYSTICK

CALCULATE DISPLACEMENT

Packets
Lx, Ly,Rx, Ry

ARDUINO BOARD

GENRATE PWM

Signals

SEND TO SERVO

Move Camera

BRING OBJECT IN CENTRE

AUTOMATIC

CALCULATE CENTRE OF FARMEm AND THE BOUNDING BOX

CALCULATE OFFSET ERROR

Packets
Herror, Verror

ARDUINO BOARD

GENRATE PWM

Signals

SEND TO SERVO

Move Camera

BRING OBJECT IN CENTRE