Corresponding Points Tracking of Aerial Sequence Images

Sukhee Ochirbat* · Sungwoong Shin** · Hwanhee Yoo***

Abstract

The goal of this study is to evaluate the KLT (Kanade-Lucas-Tomasi) for extracting and tracking the features using various data acquired from UAV. Sequences of images were collected for Jangsu-Gun area to perform the analysis. Four data sets were subjected to extract and track the features using the parameters of the KLT. From the results of the experiment, more than 90 percent of the features extracted from the first frame could successfully track through the next frame when the shift between frames is small. But when the frame to frame motion is large in non-consecutive frames, KLT tracker is failed to track the corresponding points. Future research will be focused on feature tracking of sequence frames with large shift and rotation.

Keywords : Feature tracking, Video sequences, Kanade-Lucas-Tomasi, UAV camera, corresponding points

1. Introduction

Tracking is to find and describe the relative position change of the object according to recorded video frames (Hao wu et al., 2007) and it also a most fundamental process in extracting motion information from an image sequence for frame registration (Lucas, etc., 1981). Harris, Forstner corner detector and cross correlation have been widely used for extracting features (Harris, 1998) where as the cross correlation method takes the advantages. However, in these methods features can get translation between the two frames (Zhilong, 2000). Lucas and Kanade proposed a feature-tracking algorithm (KLT) that uses gradient descent method to iteratively align frame intensity patches in successive frames (Lucas, 1981) and later it is enhance to affine transformation estimator that is used to verify tracking by finding a transformation between the current and original frame patches (shi and Thomas, 1994). General algorithm of KLT (Figure 1) for extracting and tracking the feature is given below.

2. Extraction and Tracking of KLT Features

The Kanade-Lucas-Tomasi feature tracker (KLT) was developed by Lucas Kanda and Tomasi and Kanade(1981, 1991) for tracking feature between consecutive frames. Several authors extended the KLT algorithm for various purposes (Shi, 1994; Orchibat 2007). General advantages of KLT were

1) to extract good features,
Figure 1. The workflow for feature extracting and tracking in KLT

2) to track the features in subsequent frames,
3) Replacement of features and
4) features that are tracked, can be stored and recalled

The library structure of the KLT tracker and its main functions are presented in Table 1.

<table>
<thead>
<tr>
<th>Library names</th>
<th>Member functions in the library</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 base.h</td>
<td>basic user defined types</td>
</tr>
<tr>
<td>2 convolve.h</td>
<td>_KLTToFloatImage: Given a pointer to image data and copy data to a float image</td>
</tr>
<tr>
<td></td>
<td>_KLTComputeGradients: to compute the image gradient</td>
</tr>
<tr>
<td></td>
<td>_KLTComputeSmoothedImage: to make image smoothing</td>
</tr>
<tr>
<td>3 error.h</td>
<td>KLTError(char *fmt, ...); to print errors messages</td>
</tr>
<tr>
<td></td>
<td>KLTWarning(char *fmt, ...); to print warning messages</td>
</tr>
<tr>
<td>4 klt.h</td>
<td>KLTSelectGoodFeatures: extract good features from image</td>
</tr>
<tr>
<td></td>
<td>KLTTrackFeatures: track features through next frame</td>
</tr>
<tr>
<td></td>
<td>KLTReplaceLostFeatures: replace the lost feature while tracking</td>
</tr>
<tr>
<td></td>
<td>KLTChangeTCPyramid: change the subsampling of pyramid</td>
</tr>
<tr>
<td></td>
<td>KLTStoreFeatureList/ KLTWriteFeatureTable: store features in the table</td>
</tr>
<tr>
<td></td>
<td>KLTExtractFeatureList/KLTReadFeatureTable: read feature information from table</td>
</tr>
<tr>
<td>5 klt_util.h</td>
<td>_KLTCreateFloatImage: create the instance image</td>
</tr>
<tr>
<td></td>
<td>_KLTWriteFloatImageToPGM: save the instance image into PGM file</td>
</tr>
<tr>
<td>6 pnmio.h</td>
<td>pgmReadFile: load the image</td>
</tr>
<tr>
<td></td>
<td>pgmWriteFile: write the float image into PGM</td>
</tr>
<tr>
<td></td>
<td>ppmWriteFileRGB: save the instance image into PPM file</td>
</tr>
<tr>
<td>7 pyramid.h</td>
<td>_KLTCreatePyramid: _KLTComputePyramid: create and compute pyramid image of original image</td>
</tr>
</tbody>
</table>

Table 1. Member functions of KLT tracker

To detect good features following steps are followed in the KLT.

1) to check the window size and correct if necessary,
2) to create a point list, which is a simplified version of a feature list,
3) to create temporary frames and compute gradient of frame in x and y direction,
4) to write internal frames
5) to compute track ability of each frame pixel as the minimum of the two eigenvalues of the Z matrix,
6) to sort the features and check minimum distance between features,
7) to enforce minimum distance between features.

Based on the aforementioned reason our goal of research is to evaluate the KLT for extracting and tracking features using various data sets.

3. Test Data Sets

To evaluate KLT, the set of data were collected for
Jangsu-Gun, area. Four pair of Data set “A”, “B”, “C”, “D” were used for the analysis. Frame size 350×240 pixels were selected from the video sequences. Data A and B, a pair of frames containing small shift and rotation; Data C, a pair of frames containing large shift and rotation; Data D, Sequence of images were used i.e., around 1-100 frames were used for the analysis (Figure 2).

4. Results Analysis of Feature tracking

Four data sets were used for evaluating the KLT for extracting and tracking the features. Above mentioned parameters of KLT were used for analyzing the data sets, out of 7 expect three all are constant, they are minimum distance, mask size and searching mask. The process of the feature extraction takes time for computation so that a few numbers of features are preferred and have been chosen default 100 features.

4.1 Result of Data A

Data A was a pair of images with small shift and rotation. About 100 features were chosen in the reference frame and consistency checking of feature was done using the affine model. In this case the minimum distance of the pixels and the mask size was altered and searching mask parameter was fixed and number of featured tracked is mentioned in table 2. In Case A-1, the extracted features were about 76 out of 96 features, when the minimum dis-
distance was 20 pixels. In the Case A-2 and A-3 condition good numbers of features were tracked.

4.2 Result of Data B

In Data B, the rotation and shift of the frames are comparatively greater than the Data A. In this case minimum distance and searching mask size parameter were kept as the constant where mask size were altered. Table 3 and Figure 4 shows the number of features extracted and tracked. In Case B-1 condition is said to the best in which the 81 features had been extracted and tracked out of 100 features.

In both the test case Data A and Data B the minimum distance, mask size and searching mask size were 10, 15, 35 condition gives the good numbers of tracked features. From these results we conclude that, this condition is good for tracking the features.

4.3 Result of Data C

In Data C pair of images having large shift and rotation was 20 pixels. In the Case A-2 and A-3 condition good numbers of features were tracked.

4.4 Result of Data D

In dataset around 1-100 frames of video sequences were subjected for tracking. In this minimum distance, mask size and searching mask size were 10,15,35. This condition was based on the analysis done in Data A and Data B. This condition had tracked the maximum number of features. In this case both the affine model, subsequencial mode was used during the tracking process, affine mode is remove
Table 4. Feature Tracking in Sequence of images in Data D using KLT

<table>
<thead>
<tr>
<th>Frame ID</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tracked features</td>
<td>100</td>
<td>96</td>
<td>100</td>
<td>100</td>
<td>96</td>
<td>100</td>
<td>99</td>
<td>96</td>
<td>100</td>
<td>72</td>
<td>97</td>
<td>100</td>
</tr>
<tr>
<td>Replaced features</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>28</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

5. Conclusion

In this paper we have evaluated the KLT for extracting and tracking corresponding features. It reveals that the
Data A and B, which has the small shift and rotation track the maximum number of features comparing to the Data C which had large shift and rotation. In Data A and B the accuracy of the number of features tracked was around 90 percent, which has been successfully tracking from the first frame to the successive frames. In Data C the number of featured tracked was less than 10, and the quality of feature was bad because of the large shift and rotation. In this general case KLT can built 4 level of image pyramid but in this only 2 level pyramid can be built because of the image size. In the Data D the sequence of images was subjected to track results that around 90% of features were tracked through out 1st-100th frame. With these results we conclude that KLT can track the features to the maximum when it has the small shift and rotation. Our future prospects are to enhance the KLT algorithm in tracking the features in the images having the large shift and rotation, and also to improve the performance speed and time when more than 100 features (default) subjected to track.

Acknowledgment

This research was supported by a grant(code 07KLSGC 03) from Cutting-edge Urban Development - Korean Land Spatialization Research Project funded by Ministry of Land, Transport and Maritime Affairs.

References