韓國地形空間情報學會誌 第16卷 第4號 2008年 12月 pp. 11-16

연구논문

# Corresponding Points Tracking of Aerial Sequence Images

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

要 旨

본 연구의 목적은 KLT추적자를 사용하여 무인헬기에서 취득된 비디오 동영상과 같은 연속된 영상 프레임간의 공액점을 추적하기 위한 추적자의 실행능력을 평가하는데 있다. 연구 대상지역으로서 장수군을 선정하여 무인헬기의 카메라에서 취득된 연속영상을 이용하였으며, KLT 매개변수를 이용한 특징점 추출과 추적을 분석하기 위해서 4가지 경우를 고려한 영상자료셀이 사용되었다. 그 결과 영상간의 이동과 회전이 매우 작은 영상들간의 공액점 추적은 약 90%이상이 성공적으로 추적되었다. 그러나 이동 및 회전량이 큰 영상간의 추적에서는 잘못 추적된 공액점이 포함되고 있어서 이를 보완하기 위한 연구가 추가적으로 수행되어야 할 것으로 판단된다.

**핵심용어 :** 특징점 추적, 비디오 동영상, KLT, 무인헬기카메라, 공액점

**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,

2008년 11월 6일 접수, 2008년 11월 27일 채택

* Graduate Student (BK21), Dept. of Urban Engineering, Gyeongsang National University (khos88@yahoo.com)
** Senior Member of Engineering Staff, Spatial Information Research Team, ETRI (sshin@etri.re.kr)
*** Corresponding Author, Professor (BK21/ERDI), Dept. of Urban Engineering, Gyeongsang National University (hhyoo@gnu.ac.kr)

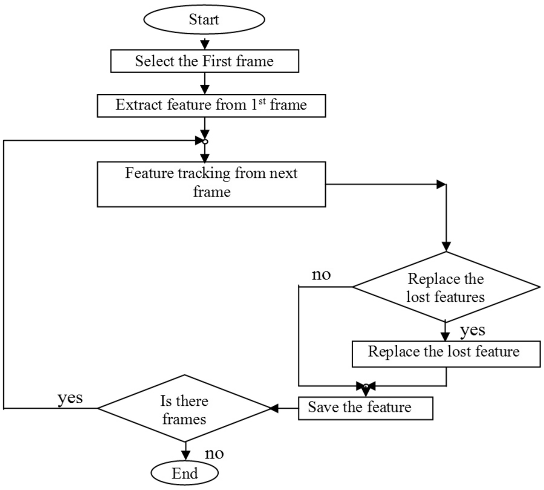



**Figure 1.** The workflow for feature extracting and tracking in KLT

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.

These library functions for feature detection, which identifies the N best features in a frame and stores them in a feature list in descending order of goodness.

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.

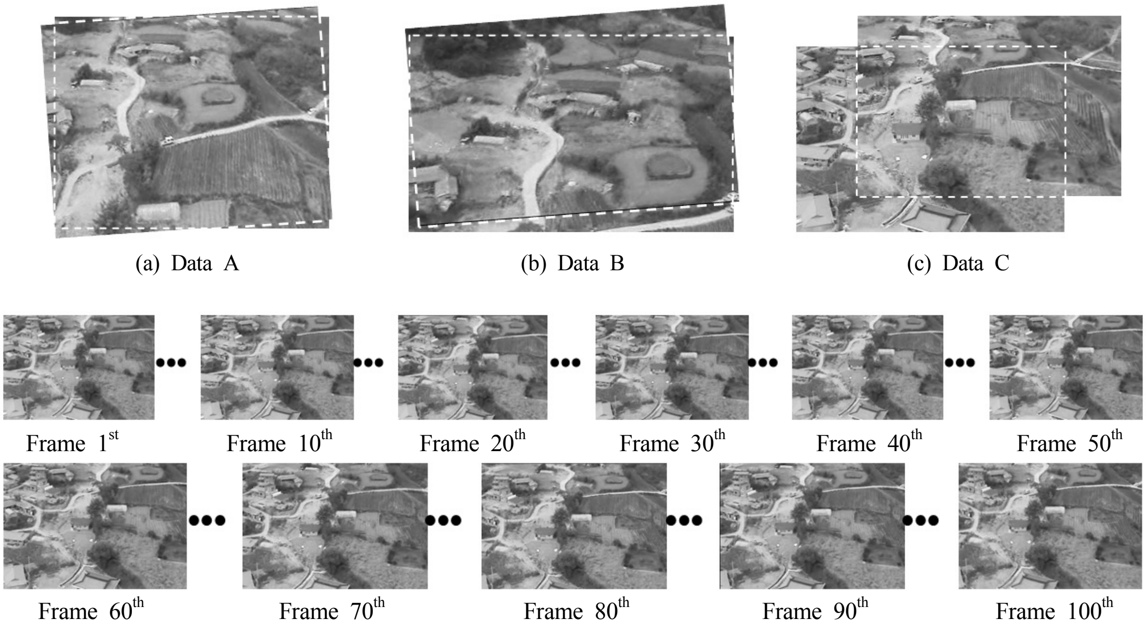
Based on the aforementioned reason our goal of research is to evaluate the KLT for extracting and tracking features using various data sets.

## 3. Test Data Sets

To evaluate KLT, the set of data were collected for

**Table 1.** Member functions of KLT tracker

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




(d) Data D

**Figure 2.** Data set used in the experiment

**Table 2.** Feature tracking in Data A using KLT

| Data A | | | | |
|---|---|---|---|---|
| Cases | Minimum distance(pixels) | Mask size(pixels) | Searching mask size(pixels) | The number of tracked features |
| A-1 | 20 | 7×7 | 35 | 96/76 |
| A-2 | 15 | 11×11 | 35 | 100/85 |
| A-3 | 10 | 15×15 | 35 | 100/88 |
| A-4 | 5 | 21×21 | 35 | 100/86 |

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-




Case A-1 (Left frame: 96Pts) (Right frame: 76Pts) Case A-2 (Left frame : 100Pts) (Right frame : 85Pts)




Case A-3 (Left frame : 100Pts) (Right frame : 88Pts) Case A-4 (Left frame : 100Pts) (Right frame : 86Pts)

**Figure 3.** Extracted features distribution of Data A

**Table 3.** Feature tracking in Data B using KLT

| Data B | | | | |
|---|---|---|---|---|
| | Minimum distance (Pixels) | Mask size (pixels) | Searching mask size (pixels) | The number of tracked features |
| B-1 | 10 | 7×7 | 35 | 100/81 |
| B-2 | 10 | 9×9 | 35 | 100/79 |
| B-3 | 10 | 11×11 | 35 | 100/65 |
| B-4 | 10 | 15×15 | 35 | 100/71 |
| B-5 | 10 | 21×21 | 35 | 100/53 |

tance 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.

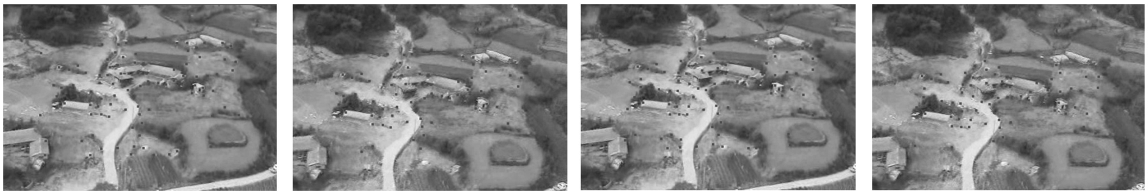
### 4.3 Result of Data C

In Data C pair of images having large shift and rotation was used for analysis. Here the number of features tracked was poor and it was less that 10. The quality of the features tracked was also poor. Figure 5 shows the pyramidal presentation of a pair of images in Data C.

General case KLT, uses the pyramidal implementation to handle the large motion data, in Data C, KLT can create only two levels of images, from which the analysis couldn't be proceeded.

### 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

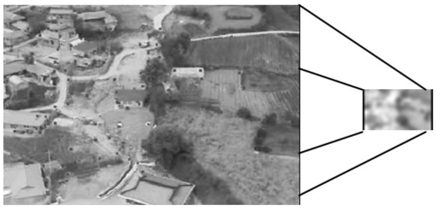


Case B-1 (Left frame: 100Pts) (Right frame: 81Pts) Case B-2 (Left frame : 100Pts) (Right frame : 79Pts)

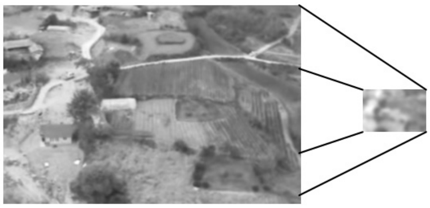
Case B-3 (Left frame : 100Pts) (Right frame : 65Pts) Case B-4 (Left frame : 100Pts) (Right frame : 71Pts)

Case B-5 (Left frame : 100Pts) (Right frame : 53Pts)

**Figure 4.** Extracted features distribution of Data B



Original image (1st level)



Original image (2nd level)

**Figure 5.** Pyramid representation of Data C.

**Table 4.** Feature Tracking in Sequence of images in Data D using KLT

| Frame ID | 1 | 5 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tracked features | 100 | 96 | 100 | 100 | 96 | 100 | 99 | 96 | 100 | 72 | 97 | 100 |
| Replaced feature | 0 | 4 | 0 | 0 | 4 | 0 | 1 | 4 | 0 | 28 | 3 | 0 |

the bad features and selects the good features and subsequencial mode enhances the tracking features. Table 4 and Figure 6 show the number the features tracked and replacement feature through out from the 1st farme to 100th frame. This shows that KLT in the video sequences track maximum number of features that is 90% under certain condition such as the small shift and rotation.

## 5. Conclusion

In this paper we have evaluated the KLT for extracting and tracking corresponding features . It reveals that the

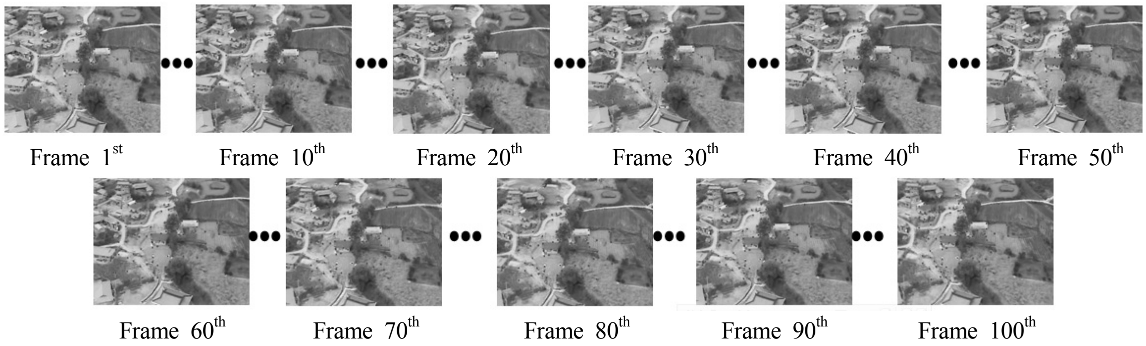



**Figure 6.** Featured tracked in a sequence of images 1st -100th frame by KLT

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 set 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 trough 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

1. Ochirbat Sukhee, 박상언, 신성웅, 유환희, 2008, "KLT특징점 검출 및 추적에 의한 비디오영상등록", *한국지형공간정보학회지*, 16권, 2호, pp. 49-56.
2. Ochirbat Sukhee, 조은래, 김의명, 유환희, 2008, "Registration of UAV Overlapped Image", *NSDI 2008 공동 추계학술대회 논문집*, pp. 245-246.
3. Hao Wu, Rama Chellappa, Aswin Sankaranarayanan, S. Kevin Zhou, 2007, "Robust Visual Tracking Using the Time-Reversibility Constraint", *IEEE International Conference on Computer Vision* (ICCV).
4. Harris Chris and Stephens Mike, 1988, "Combined Corner and Edge Detector". *Proc. Of The Fourth Alvey Vision Conference, Manchester*, pp. 147-151.
5. Lucas D. and Kanade T. 1981, "An Iterative Image Registration Technique with an Application to Stereo Vision", *International Joint Conference on Artificial Intelligence*, pp. 674-679.
6. Ochirbat Sukhee, Park, Sangeon, Shin, Sungwoong, Yoo, Hwanhee, 2007, "Application of KLT Feature Tracker for Video Sequences Registration", *Proceeding of Korean Society of Surveying, Geodesy, Photogrammetry and Cartography*, pp. 407-410,.
7. Shi J. and Tomasi C., 1994, "Good Feature to Track", *IEEE Conference on CVPR*, pp. 593-600.
8. Tomasi C. and Kanade T., 1991, "Detection and Tracking of Point Features", *Carnegie Mellon University, Technical Report* CMU-CS-91-132, Pittsburgh, PA, USA.