Image Mosaic Using FAST Corner Detection

S.MAHAMMADI NIGAR
M.TECH, Department Of ECE,
MITS, MADANAPALLE

MAHESH
ASSO. Prof., Dept. Of ECE,
MITS, MADANAPALLE

Abstract-- The main concept behind the image mosaic is image registration. The image mosaic can be defined as the automatic alignment or registration of multiple images into larger aggregates with two simultaneous images having some similarities between them. Image registration is an important part of the image processing and computer vision. On the basis of analyzing two types of image registration, an automated image registration method was put forward to dealing with image registration with similar transformation. SUSAN and Harris corner detection algorithms which are both based on intensity, were compared in stability, noise immunity and complexity quantification ally via stability factor $\eta$, anti-noise factor $\rho$ and the runtime of each algorithm. FAST (Features from Accelerated Segment Test). This detector uses the intensities of surrounding pixels to detect features. It shows how machine learning can be used to improve even further the speed of the detector. The FAST corner detector is considered an appealing feature detector due to its efficiency and fast performance and less intermediate memory requirements.

Index terms: FEATURE DETECTOR, CORNER DETECTOR, INTEREST POINT, THERESHOULD.

INTRODUCTION

Corners are important local features in images. Generally speaking, they are the points that have high curvature and lie in the junction of different brightness regions of images. In a variety of image features, corners are not affected by illumination and have the property of rotational invariance. They are only about 0.05% in the whole pixels. Without losing image data information,[1] extracting corners can minimize the processing data. It is an approach used within computer vision systems to extract certain kinds of features and infer the contents of an image. Corner detection is frequently used in motion detection, image registration, video tracking, image mosaicing, panorama stitching, 3D modeling and object recognition. Corner detection overlaps with the topic of detection. In this paper there was a brief discussion about different types of corner detectors like Moravec, SUSAN, and Harris. FAST is a special Detector which offers considerably higher performance than the other tested feature detectors. It consumes only a fraction of the time available during video processing, and on low power hardware, it is the only one of the detectors tested which is capable of video rate.

Corners as Interest Points

Many applications require relating two or more images in order to extract information from them. For example, if two successive frames in a video sequence taken from a moving camera can be related, it is possible to extract information regarding the depth of objects in the environment and the speed of the camera. [2] The brute force method of comparing every pixel in the two images is computationally prohibitive for the majority of applications. Intuitively, one can image relating two images by matching only locations in the image that are in some way interesting. Such points are referred to as interest points and are located using an interest point detector. Finding a relationship between images is then performed using only these points. This drastically reduces the required computation time.

Many different interest point detectors have been proposed with a wide range of definitions for what points in an image are interesting. Some detectors find points of high local symmetry; others find areas of highly varying texture, while others locate corner points. Corner points are interesting as they are formed from two or more edges and edges usually define the boundary between two different objects or parts of the same object.

Applications of Corner Detectors

The use of interest points (and thus corner detectors) to find corresponding points across multiple images is a key step in many image processing and computer vision applications.

Some of the most notable examples are:

- Stereo matching
- Image registration (of particular importance in medical imaging)
- stitching of panoramic photographs
object detection/recognition
Motion tracking
Robot navigation

Figure 1.1: Part of a systems for aligning boxes on an assembly line

Fig 1.1 shows part of a hypothetical system to illustrate how a corner detector [11] might be used in an automated assembly line. This assembly line fills triangle gift boxes with four different chocolates. However, the boxes must be positioned properly on the conveyor belt to ensure the chocolates are packed properly into the boxes. An overhead camera is used to capture a picture of each box as it passes under it and a computer compares it to a stored image of a properly aligned box. By finding the corners of each image, how much the box needs to be rotated can easily be computed.

Requirements of a Corner Detector

It is desirable for a corner detector to satisfy a number of criteria:

1. All "true corners" should be detected.
2. No "false corners" should be detected.
3. Corner points should be well localized.
4. Detector should have a high repeatability rate (good stability).
5. Detector should be robust with respect to noise.
6. Detector should be computationally efficient.

The detection of all true corners with no false corners is application (interpretation) dependent since there is no well defined definition of a grayscale corner. However, in many images the corners are intuitively clear and such images can be used to evaluate the performance of different corner detectors.

Localization refers to how accurately the position of a corner is found. This is critical in applications requiring the precise alignment of multiple images (for example, in registration of medical images).

Fig 1.2 the reported position of the corner is illustrated with a red circle. The corner detector on the right has good localization whereas the corner detector on the left has poor localization. Although good localization is desirable for all applications, it is not critical for all applications (for example, an object detection algorithm may simply require the approximate location of all the object's corners).

Figure 1.2: Illustration of good and poor localization, respectively

Figure 1.2: Illustration of good and poor localization, respectively

Think of a robot wandering down a hallway with a single camera. Many approaches in mobile robotics require analyzing the frames captured by the camera in order to interpret the robot's environment. A first step in many of these approaches requires finding corresponding points between frames. The two consecutive frames will be similar, but may differ due to slight geometric, illumination, or viewpoint transformations. [6] A corner detector that is robust against these transformations is said to have a high repeatability rate. The repeatability rate is the percentage of the total number of corner points which are repeated between two images. Figure 1.3 shows a corner detector that fails to detect one of the triangle's corners after it is rotated, so has a repeatability rate of 2/3 for this transformation.
Corner Detection Algorithms

The some of the corner detection algorithms which are historically significant, widely used, or well suited for a particular application i.e. real-time. In addition, all these detectors can be considered interest point corner detectors as they assign a measure of cornerness to all pixels in an image. The majority of corner detectors fall into this interest point category.

- Moravec (1977)
- SUSAN
- Harris/Plessey (1988)

These algorithms follow the same general steps for detecting corners. Fig 1.5 shows a flowchart of these steps.

1. Apply Corner Operator: This step takes as input the image and typically a few parameters required by the corner operator. For each pixel in the input image, the corner operator is applied to obtain a cornerness measure for this pixel. The cornerness measure is simply a number indicating the degree to which the corner operator believes this pixel is a corner.

2. Threshold Cornerness Map: Interest point corner detectors define corners as local maximum in the cornerness map. However, at this point the cornerness map will contain many local maximum that have a relatively small cornerness measure and are not true corners. To avoid reporting these points as corners, the cornerness map is typically thresholded. All values in the cornerness map below the threshold are set to zero. Choosing the threshold is application dependent and often requires trial and error experimentation. The threshold must be set high enough to remove local maximum that are not true corners, but low enough to retain local maximum at true corners.

3. Non-maximal Suppression: The thresholded cornerness map contains only nonzero values around...
the local maximums that need to be marked as corner points. To locate the local maximas, non-maximal suppression is applied. For each point in the thresholded cornerness map, non-maximal suppression sets the cornerness measure for this point to zero if its cornerness measure is not larger than the cornerness measure of all points within a certain distance. After nonmaximal suppression is applied, the corners are simply the non-zero points remaining in the cornerness map. The results of applying each of these steps is illustrated in Fig. 1.6.

Figure 1.6

Moravec Operator:
The Moravec operator is considered a corner detector since it defines interest points as points where there is a large intensity variation in every direction. This is the case at corners. However, Moravec was not specifically interested in finding corners, just distinct regions in an image that could be used to register consecutive image [3] frames. Moravec proposed measuring the intensity variation by placing a small square window, centered at P, and then shifting this window by one pixel in each of the eight principle directions. The intensity variation for a given shift is calculated by taking the sum of squares of intensity differences of corresponding pixels in these two windows. Intensity variation at P is the minimum intensity variation calculated over the eight principle directions.

Figure 1.7: Different cases for the Moravec operator

The Moravec operator can be used to give a measure of cornerness to each pixel in the image. This measure is the minimum intensity value found over the eight shift directions. Applying the Moravec operator to each pixel in an image creates a cornerness map.

SUSAN Corner Detection Algorithm
SUSAN (Smallest Univalue Segment Assimilating nucleus) corner detector [4] is realized by a circular mask. If the brightness of each pixel within a mask is compared with the brightness of that mask's nucleus then an area of the mask can be defined which has the same (or similar) brightness as the nucleus.

Figure 1.8

Harris/Plessey Operator
Harris is one of the most widely used corner detection algorithms based on intensity, and it has a good performance on its stability and robustness. According to Harris detection algorithm,[5] corner point is determined with the variation of gray value in a small window whose size is determined by the
actual situation. A point is not detected unless the gray value changes in both x and y direction. The gray value of the edge area changes only in x or y direction, but not both, and the value of smooth area will not be changed in either x nor y direction. Change of intensity for the shift \( [u,v] \):

\[
E(u,v) = \sum_{x,y} w(x,y)(x+u, y+v) - I(x, y) \]

**Features from Accelerated Segment Test (FAST)**

The FAST detector, introduced by Rosten and Drummond [7] in builds on the SUSAN detector. SUSAN computes the fraction of pixels within a neighborhood which have similar intensity to the center pixel. This idea is taken further by FAST, which compares pixels only on a circle of fixed radius around the point. The test criterion operates by considering a circle of 16 pixels around the corner candidate. Initially pixels 1 and 2 are compared with a threshold, then 3 and 4 as well as the remaining ones at the end. The pixels are classified into dark, similar, and brighter subsets. The ID3 algorithm from is used to select the pixels which yield the most information about whether the candidate pixel is a corner. This is measured by the entropy of the positive and negative corner classification responses based on this pixel. The process is applied recursively on all three subsets and terminates when the entropy of a subset is zero. The decision tree resulting from this partitioning is then converted into C-code, creating a long string of nested if-then-else statements which is compiled and used as a corner detector. Finally non-maxima suppression is applied on the sum of the absolute difference between the pixels in the circle and the center pixel. This results in a very efficient detector which is up to 30 times faster. A point is classified as a corner if one can find a sufficiently large set of pixels on a circle of fixed radius around the point such that these pixels are all significantly brighter (resp. darker) than the central point. The segment test criterion operates by considering a circle of sixteen pixels around the corner candidate \( p \). The original detector classifies \( p \) as a corner Machine learning for high-speed corner detection, if there exists a set of \( n \) contiguous pixels in the circle which are all brighter than the intensity of the candidate pixel \( I_p \) plus a threshold \( t \), or all darker than \( I_p - t \), as illustrated in Figure 1. \( n \) was chosen to be twelve because it admits a high-speed test [10] which can be used to exclude a very large number of non-corners. This detector in itself exhibits high performance, but there are several weaknesses:

1. The high-speed test does not generalize well for \( n < 12 \).
2. The choice and ordering of the fast test pixels contains implicit assumptions about the distribution of feature appearance.
3. Knowledge from the first 4 tests is discarded.
4. Multiple features are detected adjacent to one another.

**Figure 1.9**

FAST in general offers considerably higher performance than the other tested feature detectors. Importantly, it is able to generate an efficient detector for \( n = 9 \), which (as will be shown in Section 3) is the most reliable of the FAST detectors. On modern hardware, FAST consumes only a fraction of the time available during video processing, and on low power hardware, it is the only one of the detectors tested which is capable of video rate processing at all.

**Conclusion**

This paper proposes IMAGE mosaic method. The method extracts feature descriptor by FAST follow by SURF, and implements the matching feature points by BBF. Comparing to the previous methods like Harris, SUSAN, SIFT, this method is very fast in detecting corners, weather you take for finding the feature points, or matching the feature points between the image pairs, or the image illumination variations or the rotation variation. The time taken for mosaicking the 12 images are approximately 8 sec, which is better than its precedent Scale invariant feature transform (SIFT), which is taking more than 20s. This gives good results for the image rotation and image brightness variations. Especially FAST is preferable for video processing.

**ACKNOWLEDGMENT**

I got success in completing this work by the extreme guidance of my guide and project coordinator, I thank both of them.
REFERENCES


