Comparative analysis of methods for keypoint detection in images with different illumination level

Aleksei Makarov^{1,}, Marina Bolsunovskaya¹ and Olga Zhigunova²*

¹Peter the Great St.Petersburg Polytechnic University, Polytechnicheskaya, 29, St. Petersburg, 195251, Russia

²Tyumen Industrial University, Volodarskogo str., 38, Tyumen, 625000, Russia

Abstract. This article presents a comparative analysis of methods for keypoint detection that is a part of the research on the development of a surround camera system for large vehicles. Since the night time is the most dangerous for driving and the most difficult for image stitching, particular attention will be given to keypoint detection and image stitching in low light conditions. A comparative analysis of methods for keypoint detection has been made, a relevant technique has been developed and a series of experiments has been conducted to detect keypoints using the SURF, MSER, BRISK, Harris, FAST, and MinEigen methods. During the research, a search for identical keypoints for a pair of images, an analysis of their number and different methods of image stitching at different illumination levels were carried out. The results of the experiments are shown in graphs and tables.

1 Introduction

This article presents a comparative analysis of methods for keypoint detection that is a part of the research on development of a surround camera system for large vehicles. Recently, the issue of road safety management automation has been paid increased attention.

Driving is associated with uncontrolled risk of low-speed collisions and automobile-pedestrian accidents because of "blind spots". More than 7,000 people die and about 100,000 become injured in road accidents involving trucks in Europe annually [1]. The study carried out by the European Commission and the International Road Transport Union (IRU) representatives revealed that "blind spots" cause about 75% of traffic accidents involving trucks [2].

The main objective of the present study is the development of a surround camera system for large vehicles. It is supposed to result in a software-hardware complex consisting of four wide-angle fish-eye cameras, a hardware unit, a monitor, an interface and software that implements the developed algorithms for video signals processing.

Receiving images from the four cameras the complex will display a single real-time image of a vehicle and its environment from a bird's eye view on the monitor.

* Corresponding author: lyohamakov@yandex.ru

The process of receiving such an image consists of two stages:

1. Obtaining an orthographic top view by means of four cameras. (This stage is described in articles [3] and [4])

2. Stitching the images obtained during the first stage into a single image.

In this article, approaches to the implementation of the second stage will be discussed.

Due to the fact that the night time is the most dangerous for driving and the most difficult for image stitching, special attention will be given to the possibility of keypoint detection and image stitching in low light conditions.

2 Review of existing methods

This section provides a brief description and a comparative analysis of the existing methods for keypoint detection. The most commonly used tools are SURF and SIFT.

2.1 SURF (speeded up robust features)

The purpose of SURF (speeded up robust features) method [5] is twofold: detecting keypoints and creating their descriptors (that are descriptive elements invariant to scale and rotation change). Moreover, the search for keypoints itself also must be invariant, i.e. the rotated object of the scene must have the same set of keypoints as does the sample.

The method searches for keypoints using the Hessian matrix. The determinant of the Hessian matrix reaches an extremum at the points of the maximum change in the gradient of the image intensity. The determinant of bivariate function is defined as follows: [6]

$$H(f(x, y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}, \quad \det(H) = \frac{\partial^2 f}{\partial x^2} \frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2, \quad (1)$$

where H is the Hessian matrix, $f(x, y)$ is the change in the gradient of the image intensity function. The standard version of SURF is several times faster than SIFT, and, according to its authors, it is more invariant to various image transformations than SIFT. [5]

2.2 SIFT (scale-invariant feature transform)

In the SIFT (scale-invariant feature transform) method [7], the keypoints of objects first get extracted from the set of reference images and then stored in a database. The object is recognized in the new image by the individual comparison of each feature of the new image with the database and finding suitable candidates based on the Euclidean distance of their feature vectors. From the full set of matches the subsets of keypoints are identified. These subsets match the object and its location, scale and orientation in the new image to filter out the good matches. The definition of successive clusters is performed quickly by using efficient hash table implementation of the generalized Hough transform. Each cluster of three or more objects, which are consistent with the object and its position, is then subjected to further detailed examination of the model, and subsequently unnecessary objects are discarded. Finally, the probability that a certain set of features indicates the presence of an object, taking into account the accuracy of the correspondence and the number of possible false coincidences, is calculated.

The main point in the keypoint detection is building a Gaussian pyramid and Differences of Gaussian (DoG). A Gaussian (or an image blurred by Gaussian filter) is an image:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \quad (2)$$

where L is the value of the Gaussian at the point with the (x, y) coordinates, and σ is the blur radius. G is the Gaussian kernel, I is the value of the original image, $*$ is the convolution. [8]

There are other methods based on the methods described above that expand their capabilities.

2.3 BRISK (Binary Robust Invariant Scalable Keypoints)

In BRISK (Binary Robust Invariant Scalable Keypoints) [9], the keypoint detection is based on scale: interest points are identified by both the size of the image and the scale using the significance criterion. To improve the efficiency of calculations, keypoints are found in the octaves of the image pyramid, as well as in intermediate layers. The location and scale of each keypoint can be obtained in a continuous area using a quadratic function. For the keypoint descriptor, this method uses a sample selection consisting of one point lying on appropriately scaled concentric circles that is applied in the vicinity of each keypoint to obtain shades of grey values: by processing local intensity gradients and determining the specific direction of the function. An oriented sample of BRISK selection is used to obtain paired brightness comparison results that are collected in the BRISK binary descriptor.

2.4 MSER (maximally stable extremal regions)

The advantages of the MSER (maximally stable extremal regions) method [10] are:

- Invariance to affine transformation of image intensities.
- Covariance to the transformation (continuous) of the neighbourhood $T: D \rightarrow D$ in the image area.
- Stability: only regions with almost the same support in the range of threshold values are selected.
- Multi-scale search without any smoothing reveals both small and large structure.
- Repeatability: the detection of MSER points in the scale pyramid increases the repeatability, and the number of correspondences between scales changes.
- Distinctness: the set of all extreme regions can be listed in the worst case $O(n)$, where n is the number of pixels in the image.

2.5 FREAK (Fast Retina Keypoint)

The FREAK (Fast Retina Keypoint) method [11] uses a keypoint descriptor similar to the structure of human visual system, particularly retina. A cascade of binary strings is calculated by means of effective comparison of the image intensities on the template of the retina sample. The authors claim that FREAK points are usually calculated faster with a lower memory load, and also more reliable than SIFT, SURF or BRISK. They suggest treating corners as keypoints. Harris corners are used in this descriptor. A grid is used to select retina of circular shape, characterized by a higher density of points near the center. It is suggested to imitate fast versioned eye movements performed by jumps during the visual search ("saccadic search"), by parsing the descriptor into several steps. The method begins with a search using the first 16 bytes of the FREAK descriptor representing raw information. If the distance is less than the threshold, the comparison with the following bytes continues for analyze more accurate information. As a result, a cascade of comparisons is performed, which accelerates one more step of matching.

2.6 KAZE Features

The KAZE Features method [12] uses a multiscale 2D detector and a description algorithm in nonlinear scale spaces. Previous approaches detect and describe features at different scale levels by constructing or approximating a Gaussian scale spatial image. However, Gaussian blur does not take into account the natural boundaries of objects and smoothes both details and noise in the same way, reducing the accuracy of localization and distinctiveness. In contrast, in this method 2D functions are detected and described in nonlinear scale space by means of nonlinear diffusion filtering. Thus, it becomes possible to make the blur locally adaptive to image data, reducing noise, but preserving the boundaries of objects, obtaining excellent localization accuracy and distinctiveness. A nonlinear scale space is constructed using the effective methods of additive operator splitting (AOS) and diffusion of variable conductivity.

2.7 Harris

The Harris method [14] uses discrete features of an image. According to the authors, in order to provide clear tracing of image features, the image features should be discrete, rather than forming a continuum, for example, a texture or extreme pixels. However, the lack of points connectivity is the main limitation of this method when obtaining descriptors of a higher level, such as surfaces and objects. In the method, the Morawetz angle detectors are used that view the local window on the image and determine the average image intensity changes resulting from the window shift to a small number of pixels in different directions.

2.7 FAST

The FAST method [13] is used when applications with real-time frame rates use feature points and require a high-speed feature detector. Feature detectors such as SIFT (DoG), Harris and SUSAN are good methods that provide high-quality functions, but they are too demanding for computational resources to be used in real-time applications of any complexity. The authors show that machine learning can be used to obtain a feature detector that can fully process real-time video using less than 7% of the available processing time. When compared with the previously described methods, it should be noted that neither the Harris detector (120%) nor the SIFT (300%) can operate at full frame rate.

According to the authors of [13], the conclusion that the high-speed detector has limited application, if the created features are not suitable for subsequent processing, is obvious. In particular, the same scene, viewed from two different positions, should use features that correspond to the same real 3D locations. Therefore, the second important contribution of the authors is the angle comparison detectors, based on this criterion, applied to 3D scenes.

2.8 MinEigen

The MinEigen [15] method selects characteristics and controls features during the tracking. The choice specifically maximizes the quality of tracking and is therefore optimal in design, unlike more special texturing measures. Monitoring is inexpensive and reliable in the computational domain and allows distinguishing good and bad attributes based on the degree of dissimilarity that uses movement as a basic model of image change.

The presented descriptions of the features and functionality of the methods for detecting keypoints do not allow us to make an informed choice of a method for forming a single image based on the orthographic top view obtained from four cameras at different times of day, so the main task of the experimental study was to obtain data for performing comparative analysis methods for detecting keypoints at different illumination levels.

3 Methods of conducting experiments

For the experiments, the MATLAB modeling environment tool was used. For the repeatability of the proposed experiments and the possibility of verifying the obtained results in the program implementation of the methods, the built-in functions for detecting keypoints were used (Table 1) [16]. To define common keypoints on two images, the function “matchFeatures” was used.

Table 1. Keypoint detection search features.

Features	Type of keypoints
detectFASTFeatures	Angles. Single-scale detection.
detectMinEigenFeatures	
detectHarrisFeatures	
detectBRISKFeatures	Angles. Multiscale detection.
detectSURFFeatures	Blobs. Multiscale detection.
detectMSERFeatures	Areas of homogeneous intensity. Multiscale detection.

The sources of the initial data for experimental studies were several specialized image bases (Matlab Stitching Example [17], Adobe Panoramas Dataset [18], Panoramic Image Database [19], IIAA Panorama dataset [20]) containing 445 sets of panoramas each consisting of 5 to 32 images (Figures 1 and 2) that later were stitched together into a single image according to the keypoints obtained by different methods.

Fig. 1. Example of an initial set of images (Matlab Stitching Example).

Fig. 2. Example of an initial set of images (cvc01passadis-cyl-pano10 from IIIA Panorama dataset).

For the experiments, the images were artificially darkened, using the built-in function "imadjust" of the MATLAB tools [21]. The whole range of image brightness values (0; 255) was converted to a range (0; 150), gamma correction factor gamma in this case changed consistently from 1 to 6 in 0.1 increments, which allowed obtaining 50 modified images for each set (Fig. 3).

Fig. 3. Set of images after sequential darkening (from left to right - the original image, gamma = 2, gamma = 4).

In accordance with Table 1, each of the sets was processed in MATLAB to obtain a single image from them. Figure 4 shows the images obtained when stitching on keypoints obtained by the SURF method.

Fig. 4. The results of stitching sets of images with different gamma values by the SURF method.

Conducting a visual comparative analysis of the effectiveness of processing by different methods (Fig. 5) on the final image of the panorama is rather difficult. To perform an evaluation of the effectiveness of methods, it is suggested that the results are to be compared at an earlier stage of the algorithms, namely at the stage of determining common keypoints in two images. A metric for comparing methods is the average estimate of the number of common keypoints in the images for each set. The results of calculating the average number of common keypoints in the images for each set for different methods are presented in the next section of the article.

Fig. 5. Stitching of images with the detection of keypoints by different methods: from above - by SURF method, from below - by MSER method.

4 Results

In the course of experimental studies-the quality of image stitching was determined by the number of common keypoints between a pair of analyzed images.

The graph (Fig. 6) shows the dependence of the number of common keypoints on the illumination level.

Fig. 6. Dependence of the number of common keypoints on the illumination level of the image.

The above graph shows that the SURF method has the best results in both good and low light, including when shooting at night.

Another important parameter by which we compared the methods of detecting keypoints is the invariance to the degree of illumination, calculated by the formula:

$$inv = \frac{min}{max} * 100\%, \tag{3}$$

where max is the maximum number of common keypoints between a pair of images, min is the minimum number of common keypoints between a pair of images.

Table 2 shows the invariance values for all methods.

Table 2. Values of the invariance of the methods of detecting keypoints to the degree of illumination.

Method	Invariance
SURF	9,85 %
MSER	10,71 %
BRISK	34,62 %
Harris	6,88 %
FAST	34,62 %
MinEigen	8,99 %

In accordance with the obtained results, it can be concluded that the SURF method is not invariant to illumination change, and, therefore, the quality of the stitching with the tool will differ significantly at light and dark times of the day.

The BRISK and FAST methods showed equally high invariance to lighting, that is, it had the least significant effect on the quality of the search for keypoints and, accordingly, the stitching. A significant disadvantage of the BRISK method is a very small number of common points found for a pair of images. Another interesting conclusion that can be drawn from the above graph is that the MinEigen and MSER methods significantly exceed FAST at high illumination level, at low illumination level FAST shows slightly better results than competitors' that indicates that FAST is not only invariant, but in general works better in low light.

5 Conclusions and further research

Thus, we conducted the experiments on detection keypoints on the images using the SURF, MSER, BRISK, Harris, FAST and MinEigen methods. The search for common keypoints for a pair of images was performed, as well as the analysis of their number. The images were stitched by different methods at different illumination levels. The results showed that from the methods we studied-SURF is the best in quality of the stitching obtained, and the FAST method is the most invariant to the change in illumination. The next stage of the research will be comparing these methods by their speed, which is necessary for real-time stitching. Then, based on this and the next research, it will be possible to draw conclusions about what methods, in what combination and under what conditions should be applied. The research is planned to result in a new method that would be a complex of the studied methods and have a high stitching quality, illumination invariance and rapidity to allow real-time stitching.

References

1. URL: <http://www.transportal.by/obsujdaem/pochemu-gruzovye-avtomobili-popadayut-v-avarii.html>
2. URL: <http://www.gazu.ru/safety/bdd/10225/>
3. A.S. Makarov, M.V. Bolsunovskaya, Supplementary Proceedings of the Fifth International Conference on Analysis of Images, Social Networks and Texts (AIST-SUP 2016) **1710**, 183-192 (2016)
4. A.S. Makarov, M.V. Bolsunovskaya, Proceedings of the 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), IEEE, 697-699 (2017)
5. H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, *Features S.U.R.* (ETH Zurich, BIWI, 2008)
6. Application of the SURF method in control and management systems based on biometric technologies, <https://habrahabr.ru/post/152679/>
7. D.G. Lowe, The proceedings of the seventh IEEE international conference on, IEEE **2**, 1150-1157 (1999)
8. The construction of SIFT descriptors and the task of mapping images, <https://habrahabr.ru/post/106302/>
9. S. Leutenegger, M. Chli, R.Y. Siegwart, 2011 IEEE International Conference on, IEEE, 2548-2555 (2011)
10. J. Matas, Image and vision computing **22-10**, 761-767 (2004)
11. A. Alahi, R. Ortiz, P. Vandergheynst, Computer vision and pattern recognition (CVPR), 2012 IEEE conference on, IEEE, 510-517 (2012)
12. P.F. Alcantarilla, A. Bartoli, A.J. Davison, European Conference on Computer Vision, Springer, 214-227 (2012)
13. C. Harris, M. Stephens, Alvey vision conference **15-50**, 10.5244 (1988)
14. E. Rosten, T. Drummond, Computer Vision - ECCV 2006, 430-443 (2006)
15. J. Shi, Computer Vision and Pattern Recognition, Proceedings CVPR'94, 1994 IEEE Computer Society Conference, 593-600 (1994)
16. Point Feature Types, <https://www.mathworks.com/help/vision/ug/point-feature-types.html>
17. URL: <https://www.mathworks.com/help/vision/examples/feature-basedpanoramic-image-stitching.html>

18. URL: <https://sourceforge.net/adobe/adobedatasets/panoramas/home/Home/>
19. R. Möller, A. Vardy, S. Kreft, S. Ruwisch, *Autonomous Robots*, **23(3)**, 231-245 (2007)
20. A. Ramisa, A. Tapus, R. Lopez de Mantaras, R. Toledo, 2008 IEEE International Conference on Robotics and Automation, 538-543 (2008)
21. URL: <http://matlab.exponenta.ru/imageprocess/book3/10/imadjust.php>