Panorama Stitching Method Using Sensor Fusion

Aleksei Goncharov[a], Sergei Bykovskii[a]

[a]ITMO University, 49 Kronverksky Pr., St. Petersburg, 197101, Russia

Abstract

A commonly used solution for stitching a set of images into a panorama is to use computer vision algorithms. The greatest computational complexity in these algorithms present by the methods of image analysis, specifically, the methods for finding key points. Now there are many methods for finding key points, suitable for various conditions and shooting parameters of the initial set of frames. By choosing the correct method, you can avoid stitching defects and get the final image faster. This article introduces a method that allows you to consider the initial set of images and select a suitable algorithm for finding key points by using various data from sensors. This method allows obtaining final panoramic images without significant defects, as well as better performance relative to the compared methods for finding key points. The developed method, using the PASSAT dataset as an example, made it possible to obtain a final panoramic image of about 1.33 Mb in size in 16 seconds, regardless of the number of frames used (11, 8 or 6) with an angular displacement of (25/35/45) degrees, respectively.

Keywords

Computer vision, Embedded systems, Cyber-Physical systems, Panoramic photography, Sensor fusion

1. Introduction

Panoramic images are widely used as a means of information support for various control systems. Creation of panoramic images can be performed using specialized hardware, but this is associated with high financial costs and requires serious professional photography skills. As an alternative to specialized devices, you can use ordinary cameras, turning the camera in the desired direction and taking a sequence of frames, and then using computer vision methods to stitch the original frames and get a panoramic image. Computer vision methods allow obtaining panoramic images in the general case according to the following algorithm [2]:

1. Detection of special points of images and their comparison;
2. Construction of a projective transformation for aligning images and transferring them to a common plane;
3. Stitching images aligned relative to each other.

To construct a feature description of an image, it is necessary to select the characteristic parts of the image, for example, corners, edges, regions corresponding to extrema of intensity, etc. Algorithms that highlight such features (key points) should be invariant to various transformations: displacement, rotation, zoom and illumination of the original image, as well as the

Proceedings of the 12th Majorov International Conference on Software Engineering and Computer Systems, December 10-11, 2020, Online & Saint Petersburg, Russia

goncharov.aleshka@gmail.com (A. Goncharov); sergei_bykovskii@itmo.ru (S. Bykovskii)
0000-0001-8742-0961 (A. Goncharov); 0000-0003-4163-9743 (S. Bykovskii)

position of the camera relative to the captured object (change in perspective). To search for interpreted information on the image, it is necessary to link to the local features of the image.

Different algorithms for selecting key points do not provide universal solutions for different images due to the specifics of determining local features. In this study, it is proposed to use various sensor data to select the most appropriate algorithm for searching for key points, depending on the scene in the image, the angular displacement between frames, illumination, and other parameters.

2. Related works

The technical literature is rich in new detection features and image description algorithms[1]. However, to this day, there is no ideal detector[4]. This is mainly due to the almost infinite number of possible computer vision applications (which may require one or more functions)[5], the discrepancy in the image conditions (zoom, viewpoint, lighting and contrast, image quality, compression, etc.)[2] and the possible scene[6]. The computational efficiency of such detectors becomes even more important when considered for real-time applications[3].

Three algorithms (SIFT, SURF, ORB) were studied in detail and the following conclusions were made:

1. ORB algorithm - the fastest algorithm, but with a lower percentage of matches among other algorithms.
2. The SIFT algorithm is the slowest, but at the same time it surpasses other algorithms in terms of percentage coincidence in most cases of frame distortion considered.
3. The SURF algorithm is close enough in percentage coincidence to the SIFT algorithm and is close in speed to the ORB algorithm.
4. It is important to note that the ORB algorithm finds key points mainly in the center of the image, while the SIFT and SURF algorithms are evenly distributed over the entire image.

3. Proposal

When creating panoramas, you can use various auxiliary data of the device from which the shooting was carried out: based on a timer, an encoder, a gyroscope, or other sensors. Smartphones are often used to capture panoramic images, the image below shows the various sensors found on most smartphones.

For general control of the initial set of images, control of overlapping between frames and offsets along the axes, angular displacements, you can modify the basic algorithm based on the OpenCV library as shown in the block diagram below. As can be seen from the proposed block diagram, before starting the algorithm, it is planned to analyze the position of the camera between images to warn the user about the uselessness of processing this set of frames. With further stitching of images, it is proposed to estimate the displacement between frames, and, accordingly, the total area of overlap between frames. This approach will allow full control of the original images for the suitability of stitching into a general panoramic image, as well as control between individual frames, stopping the algorithm when the general overlaps between images are lost, and at the output of the algorithm, the user will not be provided with a full panorama, but correct in terms of image integrity. Figure 1 below shows the minimum required equipment for using the developed method and briefly shows the algorithm.

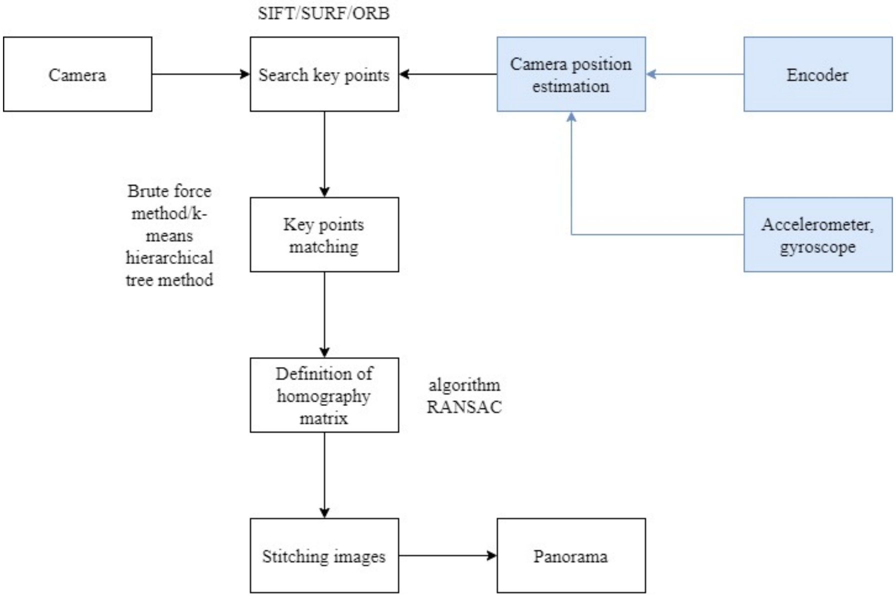


Figure 1: Proposed algorithm with a set of sensors and a camera required for use

Figure 2 shows the developed algorithm based on OpenCV in detail. Flowchart shows in detail the stages of creating a panoramic image - possible algorithms for finding key points, methods for comparing the found key points, as well as the necessary information from the sensors and their influence on the algorithm. It is important to note the potential for expanding this developed algorithm for use with other algorithms for finding key points.

4. Evaluation

To test the developed method, use the PASSTA datasets (ie, image sets) of Linkoping University. These image sets have a few functions:

1. Images were taken from a camera mounted on a tripod.
2. Between each subsequent image, the camera rotates around the vertical axis through the optical center.
3. Small enough.

The dataset includes three sets of images. Sets of images were used for the experiment:

1. Blue Dining Room: Contains 72 images captured with the Canon DS50, perspective lenses with a poor resolution of 1280 x 1920 pixels under lighting. The panoramic head

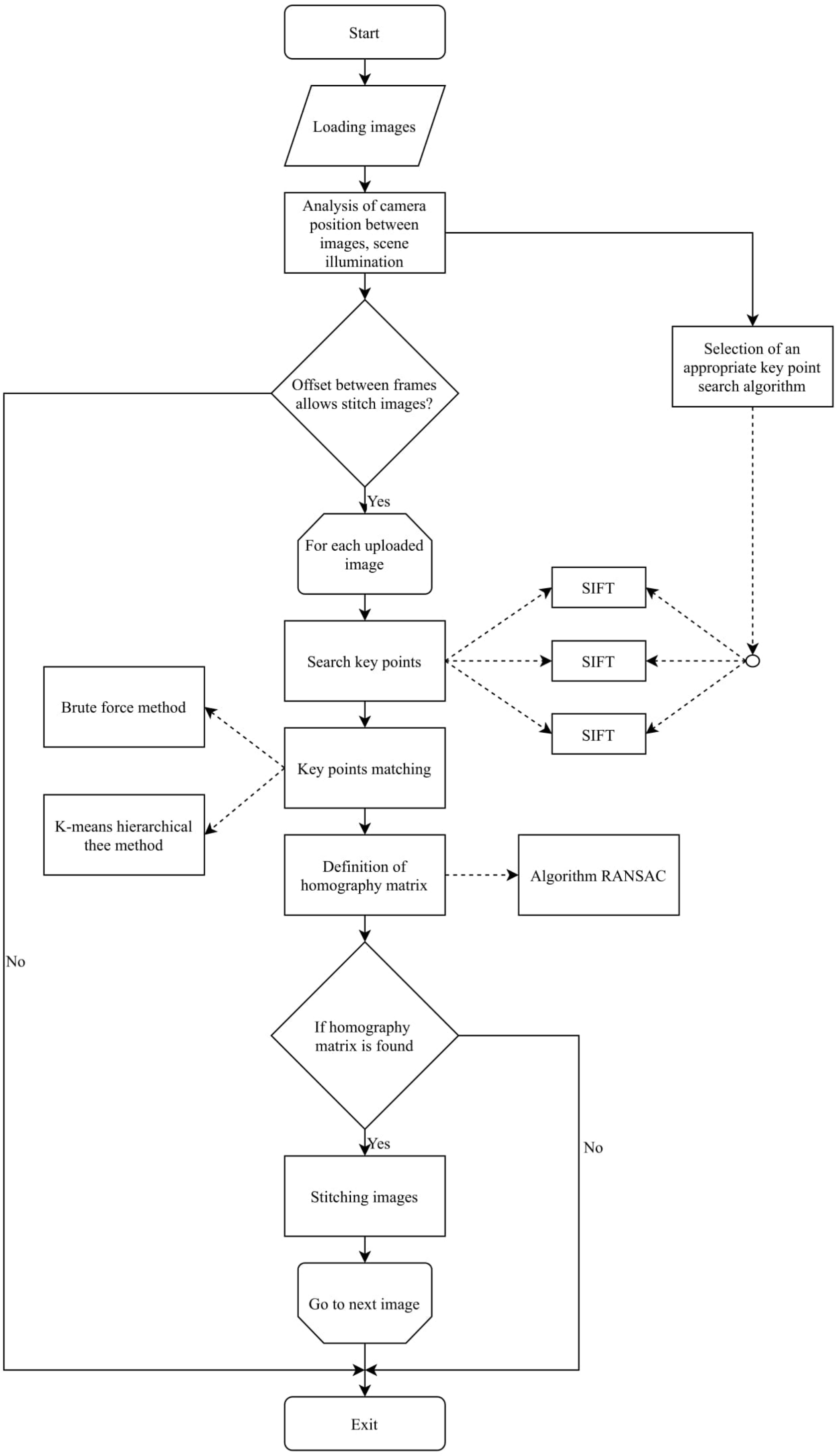

Start
Loading images
Analysis of camera position between images, scene illumination
Selection of an appropriate key point search algorithm
Offset between frames allows stitch images?
Yes
No
For each uploaded image
SIFT
SIFT
SIFT
Search key points
Brute force method
Key points matching
K-means hierarchical thee method
Definition of homography matrix
Algorithm RANSAC
If homography matrix is found
Yes
No
Stitching images
Go to next image
Exit

Figure 2: Proposed algorithm based on OpenCV algorithm

was used to rotate approximately 5 degrees around the vertical axis around the optical center of the camera.

2. Dining Room: Consists of 72 images captured with Canon DS70 Samyang 2.8 / 10mm wide-angle lenses (about 105 degrees), with a resolution of 5740x3780 pixels. The panoramic head was used to rotate approximately 5 degrees of the vertical axis around the optical center of the camera.

Figures 3 and 6 below show the results of the developed application at various angular displacements between frames on the proposed datasets. For comparison, the results of work with the same initial data of the basic algorithm are presented in Figures 4, 5 for the “LunchRoomBlue” set, in Figures 7, 8 for the “LunchRoom” set. Figures 9 show the defect and distortions used in panoramas when the algorithm operates in the angular values that are limiting for the search for key points, with violation of the spacing. Figure 10 shows for comparison the work of the developed method and OpenCV tools when using various methods for finding key points, the arising defects in the final images are separately marked.

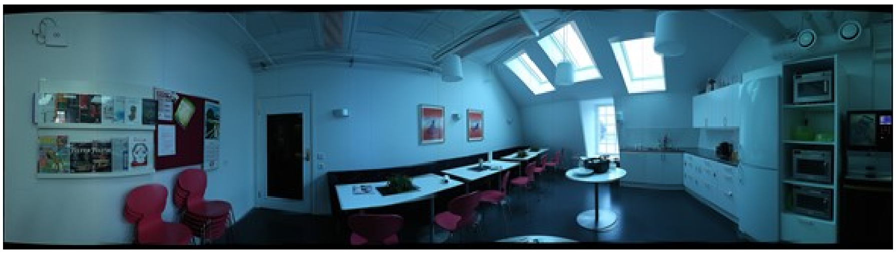


Figure 3: Panoramic image obtained from 8 frames of the "LunchRoomBlue" image set with an angular displacement of 35 degrees between frames using the developed algorithm.

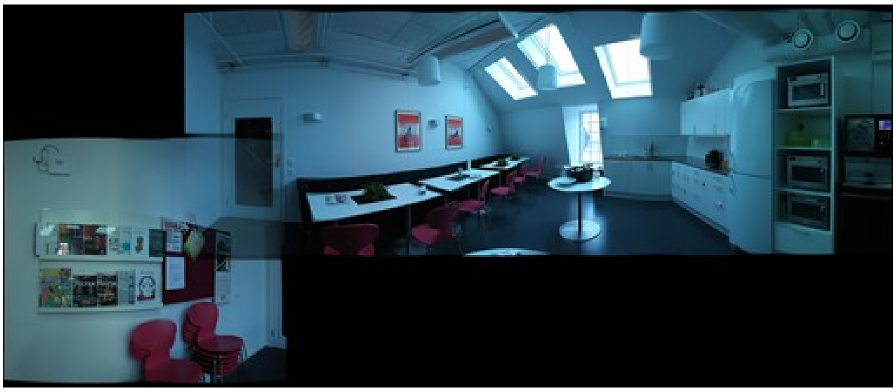


Figure 4: Panoramic image obtained from 8 frames of the "LunchRoomBlue" image set with an angular offset of 35 degrees between frames using the OpenCV algorithm (SURF).

Figure 11 shows a comparison diagram with different input data for the developed method and the standard method library OpenCV using different methods for finding key points. Defects in the final images are indicated separately. Table 1 below shows the results obtained with various input data and methods used.

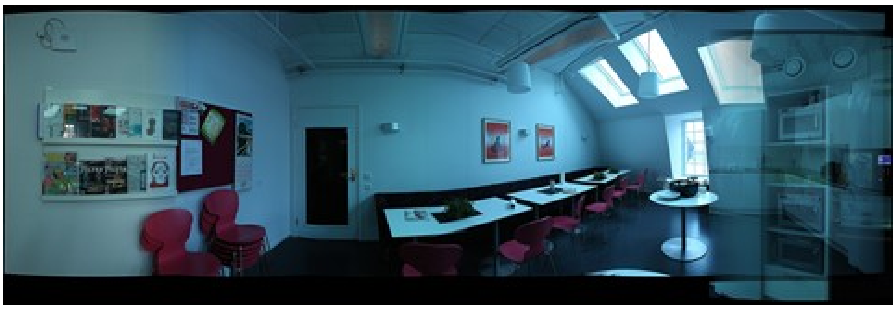


Figure 5: Panoramic image obtained from 8 frames of the "LunchRoomBlue" image set with an angular offset of 35 degrees between frames using the OpenCV algorithm (ORB).

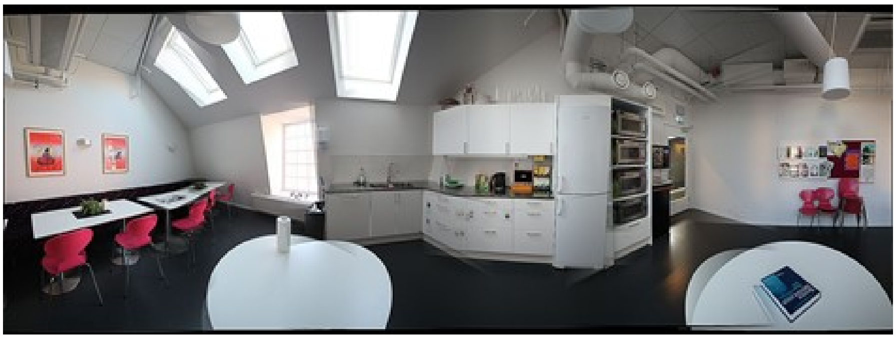


Figure 6: Panoramic image obtained from 6 frames of the "LunchRoomBlue" image set with an angular displacement of 45 degrees between frames using the developed algorithm.

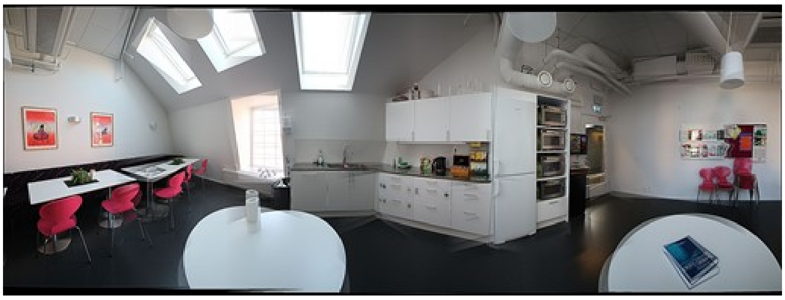


Figure 7: Panoramic image obtained from 6 frames of the "LunchRoomBlue" image set with an angular displacement of 45 degrees between frames using the OpenCV algorithm (SURF).

The operating time of the developed method was estimated for various sets of initial images. Based on the data obtained, the following conclusions can be drawn:

1. The developed method, regardless of the displacement between frames, creates a panoramic image in approximately the same time
2. With an angular displacement between frames up to 45 degrees for light scenes and up to 40 for dark scenes, the developed method does not create obvious defects in the final image.

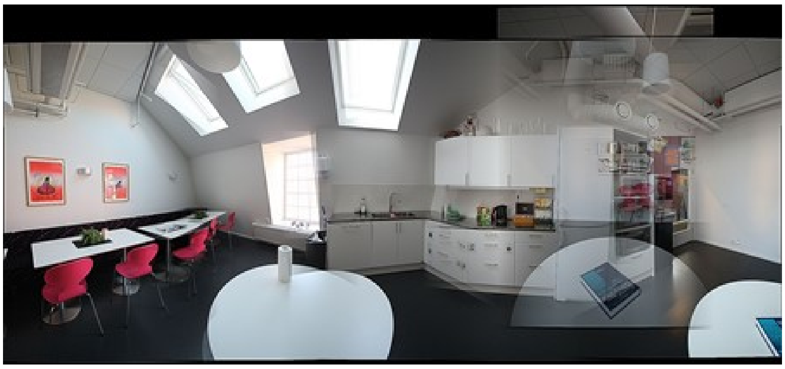
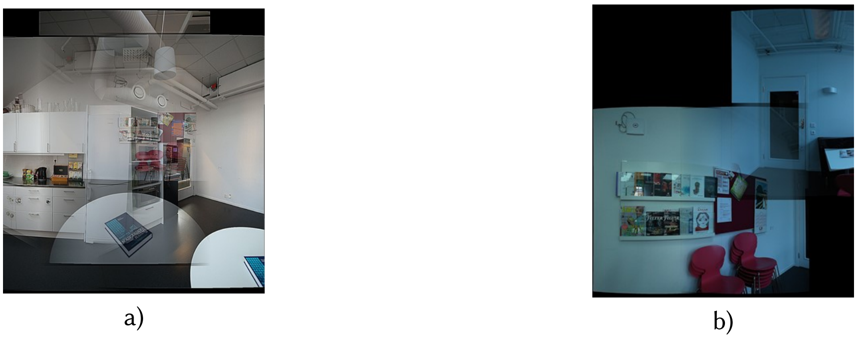


Figure 8: Panoramic image obtained from 6 frames of the "LunchRoomBlue" image set with an angular displacement of 45 degrees between frames using the OpenCV algorithm (ORB).



a)

b)

Figure 9: Panoramic image defects and distortions: a) a set of images "LunchRoom" obtained from 6 frames with an angular displacement of 45 degrees between frames using the OpenCV algorithm (ORB) , b) a set of images "LunchRoomBlue" obtained from 8 frames with an angular displacement of 35 degrees between frames using the OpenCV algorithm (SURF).

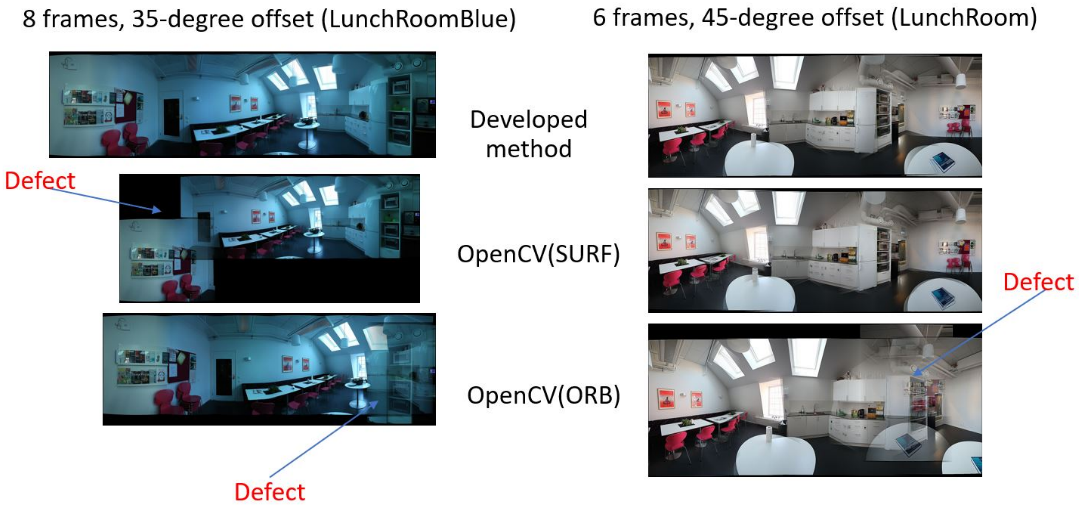

8 frames, 35-degree offset (LunchRoomBlue)
6 frames, 45-degree offset (LunchRoom)
Developed method
Defect
OpenCV(SURF)
Defect
OpenCV(ORB)
Defect

Figure 10: Method test on PASSAT dataset.

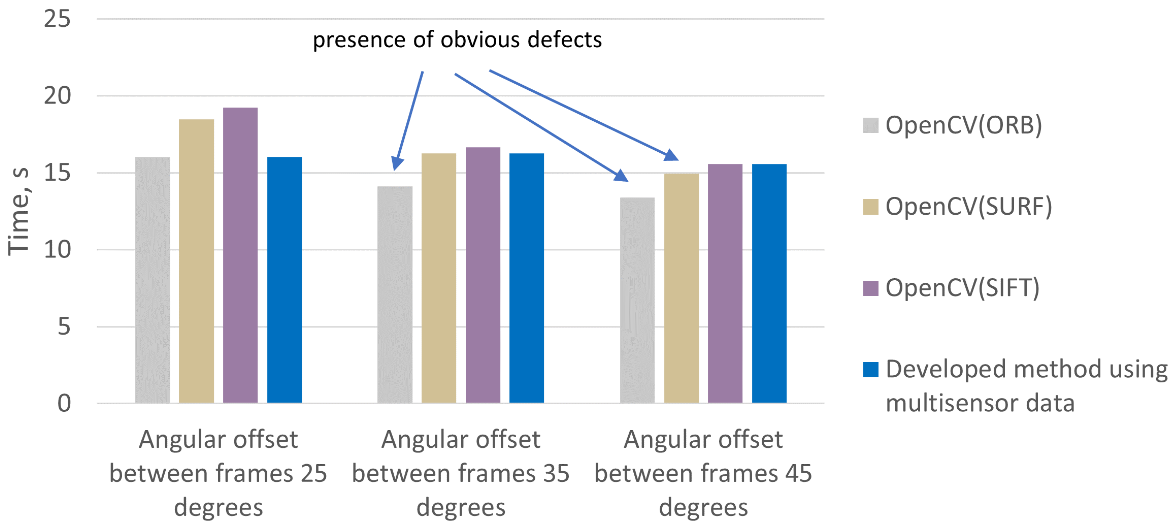


Figure 11: Testing the method on the PASSAT dataset (LunchRoom).

5. Conclusion

The analysis of existing methods for creating panoramic images carried out, methods for finding key points in images using computer vision are analyzed in detail. As a result of the analysis, it was concluded that the presence of different methods is dictated by the difference in applied problems and, accordingly, objects in the images that require the search for key points. A method was developed for creating panoramic images using multisensory data based on the OpenCV library in the Python programming language. To improve the quality of the created panoramic images by using the most suitable key point search algorithm for scenes on the original frames, as well as to control the mutual overlap between frames, shifts and displacements, it was proposed to add data from position sensors (gyroscope and accelerometer) to the algorithm. Choosing the optimal algorithm for finding key points also allows you to reduce the total running time of the algorithm without losing quality. It can be concluded that multisensor data is useful for creating panoramic images. At the same time, the developed method can be implemented in an embedded system due to a decrease in the operating time due to the use of an optimal algorithm for finding key points.

The developed method, using the PASSAT dataset as an example, made it possible to obtain a final panoramic image of about 1.33 Mb in size in 16 seconds, regardless of the number of frames used (11, 8 or 6) with an angular displacement of (25/35/45) degrees, respectively.

The developed method is adapted for expansion and use with other algorithms for finding key points, as well as the use of various sensors.

Table 1
Method test for PASSAT dataset(LunchRoom).

Panorama stitching method	Angular offset between frames, degrees	Number of frames	Panorama stitching time, seconds	Final image size, Mb	Presence of obvious defects
OpenCV (ORB)	25	11	16,04	1,27	No
OpenCV (SURF)	25	11	18,49	1,3	No
OpenCV (SIFT)	25	11	19,23	1,29	No
Developed method	25	11	16,04	1,27	No
OpenCV (ORB)	35	8	14,11	1,33	Yes
OpenCV (SURF)	35	8	16,27	1,38	No
OpenCV (SIFT)	35	8	16,65	1,32	No
Developed method	35	8	16,27	1,38	No
OpenCV (ORB)	45	6	13,41	1,35	Yes
OpenCV (SURF)	45	6	14,93	1,31	Yes
OpenCV (SIFT)	45	6	15,58	1,34	No
Developed method	45	6	15,58	1,34	No

References

[1] Y. Li, S. Wang, Q. Tian, and X. Ding, “A survey of recent advances in visual feature detection,” Neurocomputing, vol. 149, pp. 736–751, 2015

[2] Q. Liu, R. Li, H. Hu, and D. Gu, “Extracting semantic information from visual data: A survey,” Robotics, vol. 5, no. 1, p. 8, 2016

[3] E. Salahat, H. Saleh, B. Mohammad, M. Al-Qutayri, A. Sluzek, and M. Ismail, “Automated real-time video surveillance algorithms for soc implementation: A survey,” in Electronics,

Circuits, and Systems

[4] Tareen S. A. K., Saleem Z. A comparative analysis of sift, surf, kaze, akaze, orb, and brisk //2018 International conference on computing, mathematics and engineering technologies (iCoMET). – IEEE, 2018. – C. 1-10

[5] Karami E., Prasad S., Shehata M. Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images //arXiv preprint arXiv:1710.02726. – 2017

[6] Jayanthi N., Indu S. Comparison of image matching techniques //International Journal of Latest Trends in Engineering and Technology. – 2016. – T. 7. – №. 3. – C. 396-401