

Point Features Detector of Brightness Anomalies in Monochrome Images of a Real-time Video Sequence

Iakov Korovin¹, Maxim Khisamutdinov² and Donat Ivanov^{1,*}

¹Southern Federal University, Rostov region, Taganrog, Chekhov str., 2, GSP-284., 347900, Russia

²"Neuronetwork technologies" Limited, Rostov region, Taganrog, Socialisticheskaya str., 150-g., 347909, Russia

Abstract—Previously, the authors proposed an algorithm for detecting point objects in an image, applying the Harris corner detector. However, this algorithm has one drawback, which is associated with high computational complexity and the slight ability to use this approach in real-time video processing tasks. For these purposes, we propose a new approach of searching point objects on a complex background, which can also be adapted for large-scale objects search. The basis of the new approach is the analysis of all contrasting spots (blobs) in the image, as well as the trajectory analysis of frame-by-frame processing of the video sequence. The novelty of the method lies in the combination of the approach of analyzing local features descriptors of the image applying graph algorithms and extrapolating the values of the camera offset by the least squares method.

Keywords—Harris corner method; blobs detector; image stabilization; optical flow; Lucas-Kanade method

I. INTRODUCTION

The problem of highlighting brightness anomalies of point features in monochrome images is in demand in various fields of human activity: security systems, microbiology, astronomy, etc. Existing detectors [1,2] have some shortcomings, namely, high computational complexity, necessity of the patterns database use, and inability to be adopted for large-scale objects detection. We suggest that generally the process of processing a video sequence can be divided into two main stages:

- video sequence stabilization;
- highlighting point features.

Let us consider in detail these stages, and their adaptation for real-time systems application.

II. VIDEO SEQUENCE STABILIZATION FOR CONTAMINATED DATA

Previously, the authors had suggested a simple algorithm of camera motion analysis – video sequence stabilization with using of two adjacent images of an input video stream [1]. Using these images we supposed to calculate the items of the optical flow vector opt_flow with the help of the Lucas-Kanade method. The approach was based on analysis of image point features descriptors and data mining methods.

However, a situation, when not all images of the sequence have point features suitable for the Lucas-Kanade method, is possible in real conditions. This owes to the video camera functioning principles: in the moment of focus automatic adjustment or explosion the image can be overexposed or unfocused. As a result, the key points cannot be detected. In such situation, optical flow failure and video sequence stabilization break occur (Figure 1).

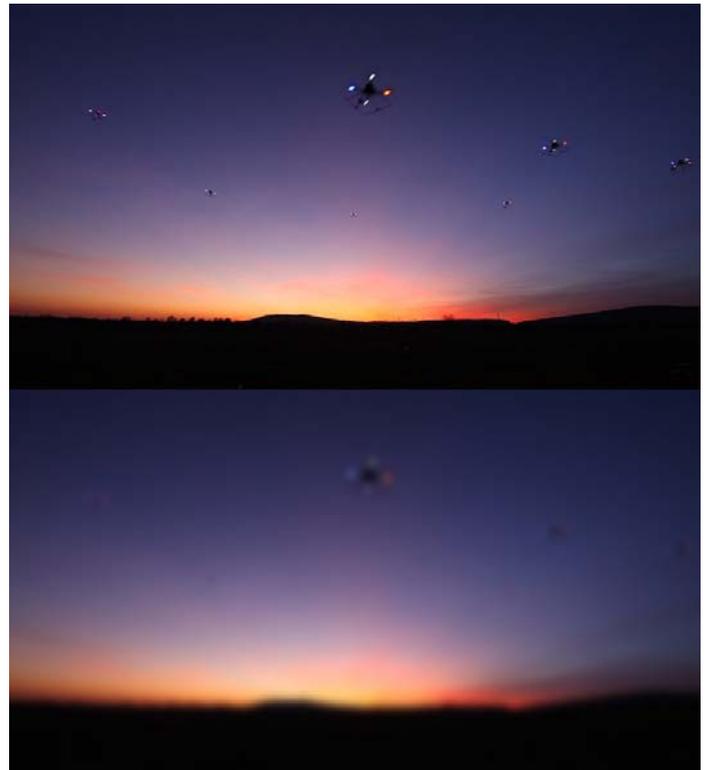


FIGURE I. ADJACENT IMAGES OF OPTICAL FLOW FAILURE.

To solve the problem, we suggest a new approach, which registers camera motion according to N images of the video sequence. The method provides filtering of images with no key features owing to extrapolation of the camera offset by the least square method [3,4].

The new algorithm is based on the algorithm, suggested by the authors on the previous step [1]. The main stages of the new algorithm proposed are depicted on Figure. 2.

Input data for video sequence stabilization is N adjacent images of the input video sequence – $[\text{frame}_0 \dots \text{frame}_N]$. On initial stage of the algorithm the vectors O_x , O_y are initialized. They will contain results of camera offsets on each image. Then, all input images are processed, beginning with the second one. If it is possible to detect the key features for the current adjacent images $\text{frame}[i]$ and $\text{frame}[i-1]$, the algorithm, developed on the previous step for detection of the offset between two images by the Lucas-Kanade method [1], is started. The obtained results are stored in to the vectors O_x , O_y . In other case, the accumulated vectors of camera offset are extrapolated by the least square method [4]. The obtained results are written into the vectors O_x , O_y .

So, processing all N images of the video sequence, we obtain offset vectors O_x , O_y , which contain the images offsets relative to the first image of the video sequence.

Novelty of the method consists in combination of the approach to analysis of descriptors of the image local features by graph algorithms with extrapolation of camera offsets by the least square method.

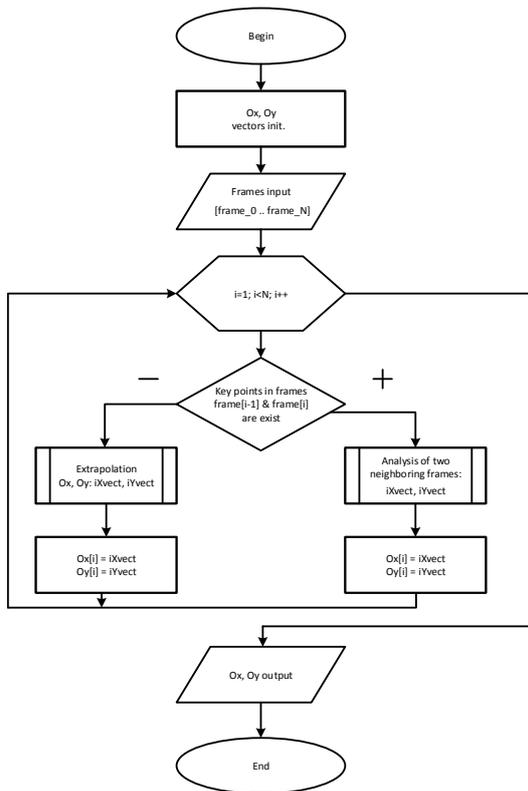


FIGURE II. THE ALGORITHM OF CAMERA MOTION REGISTRATION BY EXTRAPOLATION

III. SEARCH AND DETECTION OF POINT OBJECTS ON A COMPLEX BACKGROUND

Previously, the authors proposed an algorithm for detecting point objects in an image using a Harris corner detector.

However, this algorithm has one drawback, which is associated with high computational complexity and the inability to use this approach in real-time video processing tasks. For these purposes, the authors propose a new approach to the problem of searching for point objects on a complex background, which can also be adapted to search for large-scale objects.

The basis of the new approach is the analysis of all contrasting spots (blobs [5–9]) in the image, as well as the trajectory analysis of frame-by-frame processing of the video sequence [10]. The detection of blobs is usually the first step to more complex tasks, such as determining local deformations in images [6] or extracting scalable invariant points of interest [7–9]. The detection of blobs is often performed by calculating the local extrema of some normalized derivatives of the linear scale representation of the image [6].

Let us review a simplified flow chart of objects search on a complex background. Input data for objects search on a complex background is N adjacent images of the input video sequence $[\text{frame}_0 \dots \text{frame}_N]$. On initial stage of the algorithm the vector $vObj$ is initialized. It will contain results of found objects. Then, from the very first, all input images are processed. To do it, blobs are detected in each image $\text{frame}[i]$, and then their centers and sizes are analyzed. If there is an intersection of an object and the existing objects of the array $vObj$, the parameters of the object are updated. If there is no intersection, the new object is added in to the vector $vObj$ with the detected parameters of the blob.

So, during processing of all N images of the video sequence, we will obtain a vector, which contains parameters of the objects $vObj$ which belong to the input video sequence. The algorithm is shown below (Figure 3).

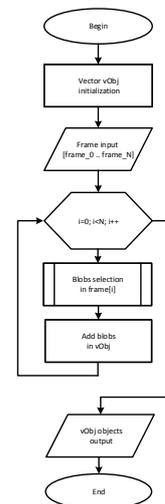


FIGURE III. OBJECTS SEARCH ON A COMPLEX BACKGROUND BASED ON A BLOB PARADIGM

Let us make a detailed review an algorithm of blobs detection, which is the key one in the suggested methods of objects search on a complex background (Figure 4).

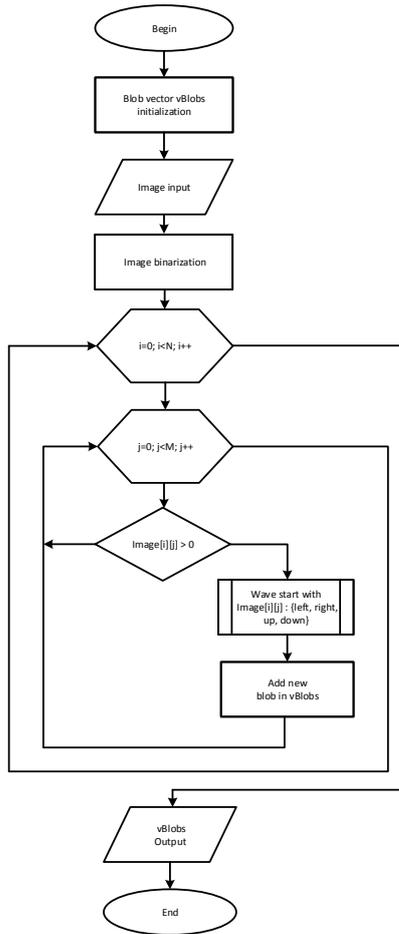


FIGURE IV. THE ALGORITHM OF BLOBS DETECTION ON A MONOCHROME IMAGE

Input data for the algorithm of blobs detection is a single image of the video sequence Image. The example of the image is given in Figure 5.



FIGURE V. AN EXAMPLE OF AN INPUT IMAGE FOR BLOBS SEARCH

Actually, it represents itself a 2D array of 8-bit data of N x M size. The first step of the algorithm is binarization of the array (see Figure. 6).



FIGURE VI. THE RESULT OF THE INITIAL IMAGE BINARIZATION

The next step is detection of locally connected areas or the searched blobs. To search locally connected areas, we use a wave algorithm for each non-zero item of the array Image. The wave is spreading and the items are being zeroised. Four boundaries {left, right, up, down} are updated. When wave spreading is over, then parameters of a new blob are saved into a result list of blobs on the image vBlobs. Figure 7 shows the result of blobs selection on the input image.

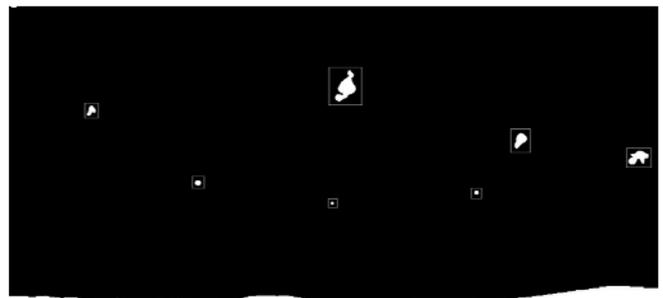


FIGURE VII. THE RESULT OF BLOBS SELECTION ON THE INITIAL IMAGE

Further on, it is necessary, using the obtained information about the objects vObj, to detect point objects. Since the size of point objects is small (it may achieve 2x2 pixels), pattern usage is low. For this, the authors suggest to use the angular size of the object. It is possible to obtain the angular size, using the pixel size and the camera parameters (angular size of a pixel in arc degrees), and besides, the average speed of the object moving arc degrees. For this, it is necessary to prepare a table TableObj of the known objects with description of their possible angular speeds and sizes. The table is a vector; every item of the vector is a tuple which contains 6 values (min_spd is a minimum speed of the object, max_spd is a maximum speed of the object, min_W is a minimum width of the object, max_W is a maximum width of the object, min_H is a minimum height of the object, max_H is the maximum height of the object). So, comparing the parameters of the found objects vObj with the values from the TableObj table, it is possible to make clear detection of point objects. Figure 8

shows the result of objects detection on a single image of the video sequence.

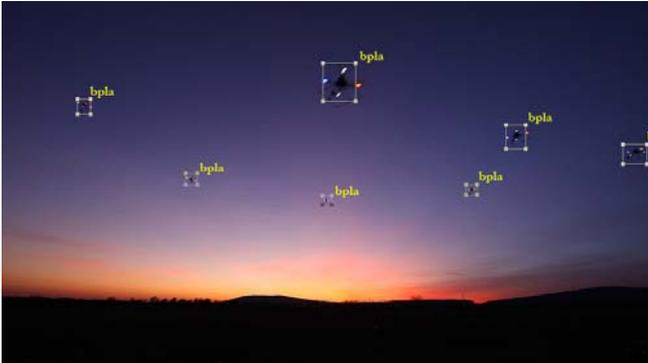


FIGURE VIII. THE FINAL RESULT OF POINT OBJECTS DETECTION

IV. CONCLUSION

In the paper we have proposed a method that with high probability can successfully handle point objects. Due to the blobs paradigm, the method successfully works with both static and moving objects. The method is applicable in real-time systems through the use of algorithms of low computational complexity. The approach is resistant to optical flow failures due to extrapolation of the accumulated values of the camera displacement vectors using least squares methods. As it was mentioned above, we have proposed a new approach to the problem of finding point objects on a complex background can be adapted to search for large-scale objects. The essence of adaptation is the necessity to modify the blob search algorithm, namely to increase the wave propagation step. If a wave with step 1 is applied for point objects, i.e. extends only to neighboring pixels, then to highlight large-scale objects, the wave step must be increased, in accordance with the characteristics of the video source. The authors found that it is enough to use the step of wave 3 to confidently highlight large-scale objects. With a smaller step, a large-scale object can be divided into several disconnected blobs.

ACKNOWLEDGMENT

The research is conducted with the support of RFBR project 17-29-03407.

REFERENCES

- [1] Korovin I., Khisamutdinov M., Ivanov D. A Basic Algorithm of a Target Environment Analyzer // Proceedings of the 2nd International Conference on Advances in Artificial Intelligence. 2018. P. 7–11.
- [2] Korovin I., Khisamutdinov M., Schaefer G. Efficient Noise-free Image Acquisition of Moving Objects in a Complex Background using CUDA // Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV). 2016. P. 331.
- [3] Pratt W.K. Digital image processing: PIKS Scientific inside. Wiley-interscience Hoboken, New Jersey, 2007. Vol. 4.
- [4] Gonzalez R.C. Digital image processing. Prentice hall, 2016.
- [5] Danker A.J., Rosenfeld A. Blob detection by relaxation // IEEE Trans. Pattern Anal. Mach. Intell. IEEE, 1981. № 1. P. 79–92.
- [6] Lindeberg T. Feature Detection with Automatic Scale Selection // Int. J. Comput. Vis. 1998.
- [7] Bay H. et al. Speeded-up robust features (SURF) // Comput. Vis. image Underst. Elsevier, 2008. Vol. 110, № 3. P. 346–359.
- [8] Bay H., Tuytelaars T., Van Gool L. SURF: Speeded up robust features // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2006.
- [9] Mikolajczyk K., Schmid C. Scale & affine invariant interest point detectors // Int. J. Comput. Vis. 2004.
- [10] Ming A., Ma H. A blob detector in color images // Proceedings of the 6th ACM International Conference on Image and Video Retrieval, CIVR 2007. 2007.