

The task adaptation method for determining the optical flow problem of interactive objects recognition in real time

R A Bagutdinov, A A Zaharova

Tomsk Polytechnic University, The Institute of Cybernetics, 84/3, Sovetskaya Str.,
Tomsk, 634034, Russia

E-mail: bagutdinov@tpu.ru

Abstract. In this paper, we solved the problem of determining the optical flow as a stationary image and a real-time video stream. Unlike conventional methods of calculation, the problem of the module integrated vision determines the optical flow in Unity3D interactive environment to obtain real time data. The carried out testing and approbation of data presented an image processing algorithm, an experiment, the results obtained by the optical flow.

1. Introduction

The obtained image can be captured in real time; thus, it makes it possible to obtain useful exhaustive information: the movement of objects in the scene and the camera motion parameters. Light Change (brightness) of the image was observed in the movement of the camera relative to the depicted objects called an optical flow [14].

The optical flow provides information even if there is deformation of the image of the observed surface. The motion of objects in front of the camera during shooting or camera movement relatively the fixed stage leads to corresponding changes in the image. These changes can be used to restore the relative motion in the image frames on the basis of changes in time. Ideally, the optical flow corresponds to the field of motion (vector field), with the proviso that when shooting, illumination of objects does not change [13].

The relevance of the work consists in adapting the method for determining the optical flow to the problem of recognition of objects in the interactive mode in real time.

In this paper, to accelerate computation and optimization of image processing, we used the EmguCV library. The program is written on the basis of the C # language in the Unity3D environment.

In general, the term “Emgu CV” means a cross-platform "shell" for the OpenCV image processing libraries (the main library for working with technical vision). Emgu CV is also called a machine vision library.

EmguCV supports a variety of programming languages: C #, Visual, C ++, IronPython and several other languages. It is used to solve various problems associated with 2D and 3D graphics, identifying persons and objects in the photo and video.

Project Emgu CV can be compiled in Mono Development in the Unity3D medium, provided that work can be adapted under Linux, Windows, Mac OS X and popular mobile platforms, such as Android, iOS.



OpenCV, developed by Intel, is a library of programming functions. It uses a real-time computer vision. Developers are free to use the library in accordance with the license, an open-source. It is written in optimized C and C++ languages and contains more than 500 functions that cover different areas of computer vision. Moreover, the library may use multicore processors to retain computational efficiency.

2. Methods

Sequences of images in the stream allow estimating the velocity or movement of the image changes, such as a discrete shift of these images. Methods are based on the optical flow calculated motion between two frames, taken at two points in the previous time frame (T1) and the subsequent frame for them (T2), for each pixel [28].

Studies of the optical flow is widely carried out in various fields of image processing, including video compression and motion analysis. Optical Flow algorithms determine not only the motion of the vector field, but also the use of the optical flow in the analysis of 3D-scenes, as well as the movement of objects and the observer relative to the scene [26]

The problems of determining the optical flow are used in robotics when recognizing objects, object tracking, determining movement and robot navigation [23, 25]. In addition, the optical flow is used to study the structure of objects [7], since the definition of the movement and the creation of the scene surrounding the depths of the environment are applied in the areas of simulation of human vision.

In this problem, the formation of the optical flow consists in determining the magnitude and direction of the displacement of individual image points in two frames selected from the resulting video sequence from a connected camera in real time [1, 2].

The following actions are performed in the beginning:

1. Determine the presence of the camera on the computer, when the camera is not found, an error message appears.
2. If the camera is found:
 - 2.1. Sets story frames (array retention)
 - 2.2. Runs a frame capture process (process frame)
 - 2.3. Capture a frame and display it on the screen (Figure 2)
 - 2.4. Masking on Image 1 (Figure 3)
 - 2.5. Output Figure 3 to the screen.
 - 2.6. Figure 3 Improved color.
 - 2.7. Record Figure 3 in history.
3. Comparison of the last two frames in the story.
 - 3.1. Setting the threshold to define the motion detection area
 - 3.2. Getting the sequence of movement of components (Determination of the optical flow)
 - 3.3. Implementation of the actions of all movement components
 - deviation of the components that have a small area (Less than the predetermined threshold);
 - finding a movement angle in a certain area;
 - deviation of the field, which contains too little movement (angles less than the threshold);
 - rendering of the motion area.
4. Display the obtained result of the comparison with the painted areas of movement (Figure 4).
5. Return to step 1.

The entire process is schematically shown in Figure 1

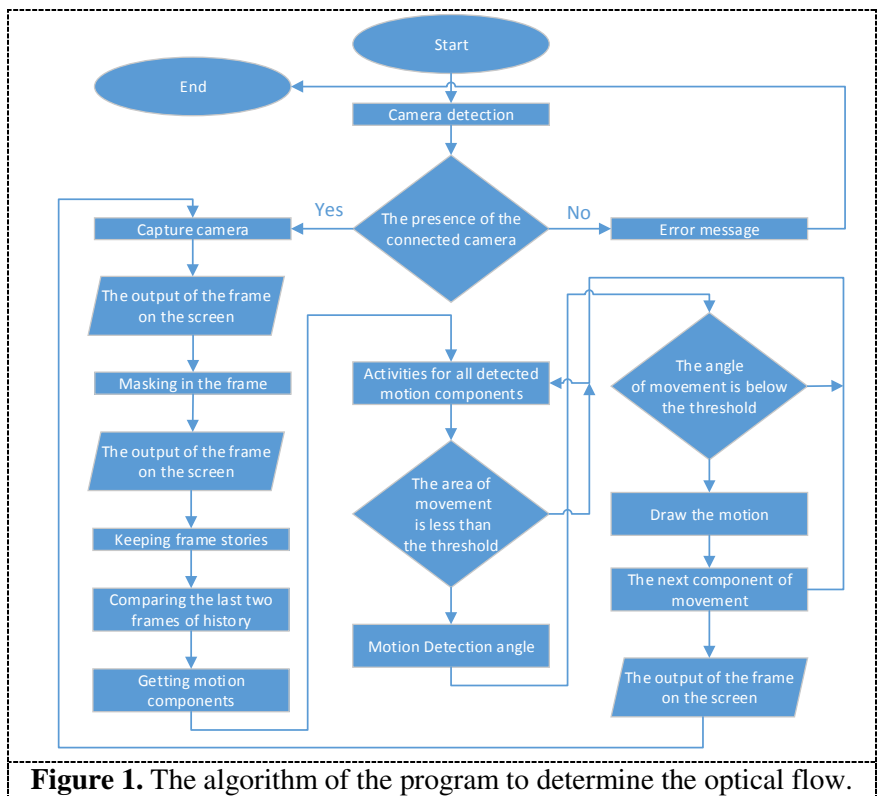


Figure 1. The algorithm of the program to determine the optical flow.

The implemented algorithm used the C# programming language based on Unity3D v.5.3.5 environment. (64 bit). For the calculations, a PC was used with the following specifications: Intel Core i5 3.20 GHz, RAM 8 GB, OS Windows 7 Pro 64bit. The test was carried out on the Sven IC-990HD camera with 1920x1080 resolution and a frame rate of up to 30fps / sec. To speed up the computation, the original image was converted to black and white. As a result of the work, the following drawings were obtained (Figure 2, 3, 4).



Figure 2. The results of the experiment: the original image



Figure 2. The results of the experiment: the original image

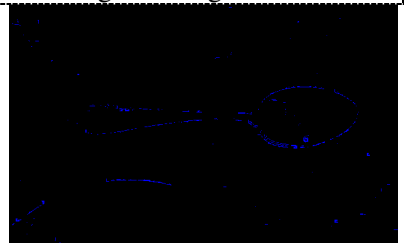


Figure 4. The results of the experiment: the optical flow

overlaid with the mask image

3. Conclusion

In this article, the problem of determining the motion parameters for the optical flow based on the C# language was considered. Unlike the traditional methods of calculation, the problem of the module integrated machine vision was used interactively to determine the optical flow in the Unity3D environment to obtain real-time data. The carried out testing and approbation of data, resulting in the advantage of working with this method, significantly increase the speed of image processing by an average of 4.86% in contrast to conventional methods. During the experiment, it was found that the accuracy of a sketch of the contours of objects depends on the illumination of the object, the better the lighting, the better the image quality.

For any movement relative to the camera and the object, we can determine the appropriate optical flow. The optical flow is dependent on many parameters, such as the brightness of the subject, and the camera features. The results of this work can be implemented in the design of robotic systems, used in various technical fields of application, including the analysis of images obtained during astronomical studies.

References

- [1] Bagutdinov R. A. 2016 *Cybernetics and programming* **5** 10-15.
- [2] Bagutdinov R. A. 2015 *Modern scientific research and innovation* **10** 1-3
- [3] Moshkin V. I., Petrov A. A., Titov V. S., Yakushenkov Y. G. 1990 *Technical vision of robots* (Moscow: Engineering) pp 270-272
- [4] Potapov A. A. 2008 *The latest image processing techniques* (Moscow: FIZMATLIT) p 496
- [5] Girshina C. B., Vatolin D. S., Lukin A. S., Putilin S. Y., Strelnikov K. N. 2008 *Software systems and tools* (Publishing house of the MSU Faculty of CM and Cybernetics) **9** 50-62
- [6] Fissenko V. T., Fissenko T. Y. 2008 *Mechanics and Optics* (ITMO University) p 192
- [7] D. A. Forsyth, Jean Ponce 2004 *Computer vision* (Moscow: Publishing House "Williams") 928
- [8] Horn B. K. N. 1989. *Robot Vision* (Moscow: Mir) p 487
- [9] Gruzman I. S., Kirichuk V. S. 2002 *NSTU* 352
- [10] Chen L., Huang X. 2015 *4th Inter. Conf. on Comp. Sci. and Netw. Tech* 452-456
- [11] Horn B. K. P., Schunck B. G. 1981 *Artificial Intelligence* **17** 185-203
- [12] Glyn W., Humphreys, Bruce Vicki 1995 *Visual Cognition: computational, experimental and neuropsychological perspectives* (Lawrence Erlbaum Associates Ltd) p 339
- [13] Glocker B., Komodakis N., Tziritas G., Navab N., Paragios N. 2008 *Medical Image Analysis Journal* **12(6)** 731-741
- [14] Jong-hyuk K., Brambley G. 2006 *Department of Engineering* (Australian National University Canberra) 32
- [15] Lucas B. and Kanade T. 1981 *DARPA IU Workshop* 121-130
- [16] Zingg S., Scaramuzza D., Weiss S., Siegwart R. 2004 *Autonomous Systems Lab ETH Zurich* 18
- [17] Messaoudi F., Simon G., Ksentini A. 2016 *Annual Workshop on Network and Systems Support for Games* 16-20
- [18] Stone S. H. 2001 *IEEE Transactions on Geoscience and Remote Sensing* **39 (10)** 2235-2242
- [19] Wang Yao 2003 *Motion Estimation for Video Coding Polytechnic* (University Brooklyn) p 43
- [20] Yang B., Zhang T., Duan K. 2015 *Int. Conference on VR and Visualization* 276-280