Using computer vision in the gameplay of educational computer games

Using computer vision in the gameplay of educational computer games

G Chursin and M Semenov

National Research Tomsk Polytechnic University, Lenin Avenue, 30, Tomsk, Russia, 634050

E-mail: sme@tpu.ru

Abstract. This article discusses the use of computer vision elements in an educational game developed with the Unity3D game engine. The game is aimed at teaching and developing the skills of students in mathematics, physics, programming. For the development of educational games, it is necessary to completely change the approach to their development. We assume that the introduction of modern technologies, namely computer vision and microcontrollers, will attract students to the study of technology. This paper examines the possibility of introducing computer vision into the gameplay in real time. We propose to use image segmentation based on the oriented gradient histogram functions. We achieve 98 % accuracy using support vector machines for classification

1. Introduction

To date, there is practically no development in the field of educational computer games. The game development cost is very expensive and most never recover their development costs. Many educational games have been developed by teachers with limited funding and technical skills for this reason didactic games are often primitiveness, simplicity and monotony in a gameplay. Unfortunately, existing games do not encourage students to learn new things [1]. In order to generate interest in the study of Mathematics, Physics, or Computer Science, games must demonstrate the applicability of these sciences in the real world [2, 3].

Since the inception of the first games, gaming peripherals have been an essential part of the gaming industry. These include keyboards, computer mice, game wheels, gamepads, joysticks. These are all physical devices that must be held in hand and used in the gameplay. Attempts to get rid of gaming peripherals lead to the introduction of computer vision into games, as the only possible analogue to gaming devices today. The first successful attempts at introducing object detection into games was the release of the Kinect [4, 5]. This controller allowed a player to play games on the Xbox 360 with full body gestures. The problem with Kinect-based games is that they are developed specifically for the concrete device. In addition, for a full-fledged game, a player must have a free space of 2 m approximately. That is, such games is difficult to integrate to school curriculum in a classroom.

Computer vision is successfully used in various spheres of life. But there was no full-fledged introduction into the gaming industry. First, the current level of development of computer vision does not allow using it in dynamic scenes. Recognition errors and low computation speed lead to problems in the gameplay. Secondly, the use of computer vision increases the load on the

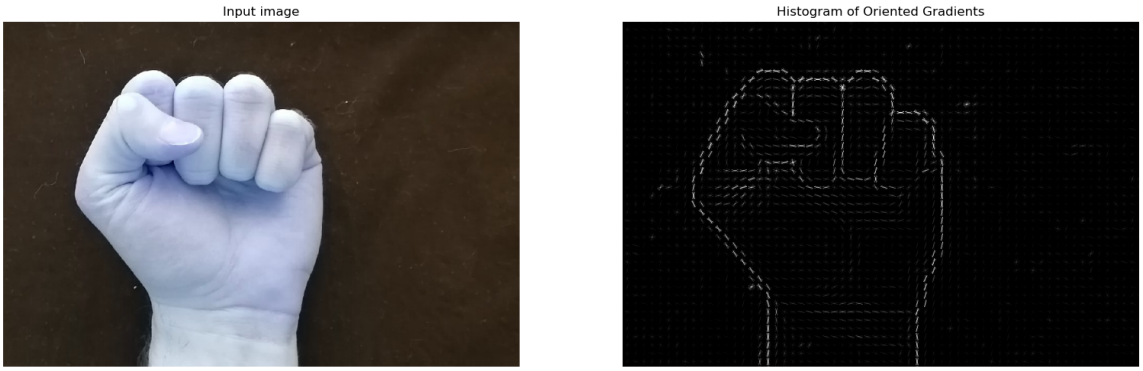

Figure 1. Image of a fist (left) and a histogram of oriented gradients (right).

Figure 2. Image of a palm (left) and a histogram of oriented gradients (right).

central processor and video card. Combined with dynamic game scenes, this will reduce the computer performance.

At the same time, educational computer games can include simple puzzles [6] that do not require dynamic scenes with high computational complexity. This means that it allows us to use computer vision in the gameplay. Using a regular webcam will make it easier to move from gaming mechanics that use a keyboard and mouse to using computer vision. Such approach will effectually allow to engage students into the gameplay and to improve performance in the specific subject taught. Therefore, the aim of the work is to introduce the computer vision into the education game development pipeline.

2. Experimental part

For computers, recognizing well-defined gestures (for instance, sign language, referee signals) is challenging and has traditionally required thousands of training examples to learn [7].

As an experiment, consider the following game situation. The appearance of an *open palm* in a camera is accompanied by the creation of an object on the game scene, and vise versa the appearance of a *clenched fist*, leads to the removal of a random object from the scene (Figures 1, 2, left side). Game classification is a binary classification problem, where we have to classify an

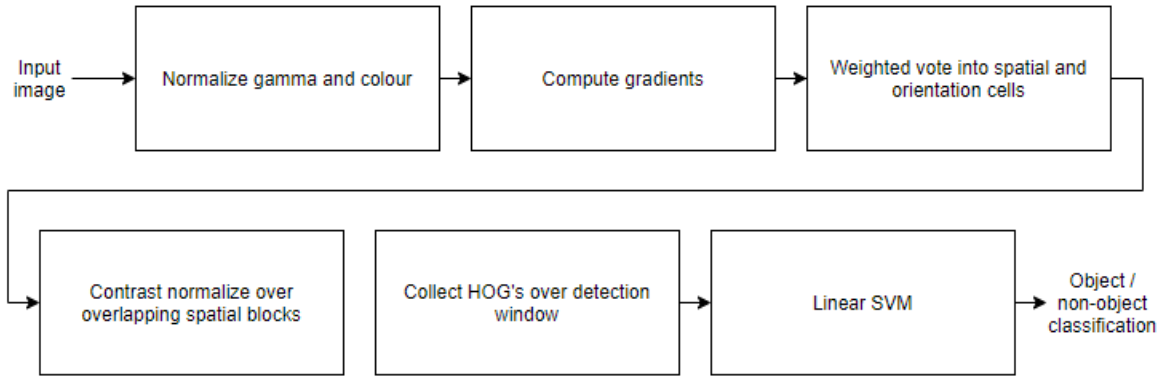

Figure 3. Binary classification scheme in a game situation.

image into one out of the two possible classes: fist and palm.

We used a combination of histogram of oriented gradients (HOG) and support vector machine (SVM) to detect and classification an object in a game window, respectively. The histogram of directional gradients (Figures 1, 2, right side) is based on the assumption that the presence and shape of objects in the image can be described by the distribution of image intensity gradients. These descriptors are constructed by dividing the image into cells, and assigning to each cell a histogram of gradient directions for pixels within a cell, their combination is a *descriptor*. In order to increase the accuracy, the processed image, as a rule, is made black and white, and the local histograms are normalized in contrast with respect to the intensity measure calculated over a larger fragment of the image. Contrast normalization allows for greater lighting invariance of descriptors. The implementation of these descriptors is obtained by breaking the image under study into cells. In the cells, histograms h_k, $k = \{L_2, L_1, \sqrt{L_1}\}$ of the directed gradients of the interior points are calculated. These histograms are combined into one common histogram $h = f(h_1, h_2, \ldots, h_k)$, after which it is normalized to brightness [8]

$$h_{L_2} = \frac{h}{\sqrt{||h||_2^2 + \varepsilon^2}}, \qquad h_{L_1} = \frac{h}{||h||_1 + \varepsilon}, \qquad h_{\sqrt{L_1}} = \sqrt{h_{L_1}},$$

where $||\cdot||_k$ is k-norm, $\varepsilon > 0$ is a some small constant. To calculate the angles, θ and the gradients values, g at each point of the image, it is necessary to calculate two matrices D_x and D_y of the derivatives of the x and y axes, respectively [9, 10]:

$$\theta = \cot\left(\frac{D_y}{D_x}\right), \quad g = \sqrt{D_x^2 + D_y^2}.$$

The matrix D_x, D_y is formed by convolution of the image with the kernels: $[-1, 0, 1]$ and $[-1, 0, 1]^\top$. These kernels are used to filter the color and brightness components of the original image. After that, the HOG-descriptors are classified using a SVM which is based on the principle of changing the original vector space into a space with a higher dimension, as well as searching for the optimal hyperplane separating the classified features. The result of the work of the SVM-classifier is two images of the object with positive and negative weights of the support vectors, respectively. Positive weights mean that the features belong to the desired object, negative to the background [11]. The final scheme of proposed pre-possessing and classification steps is shown in Figure 3.

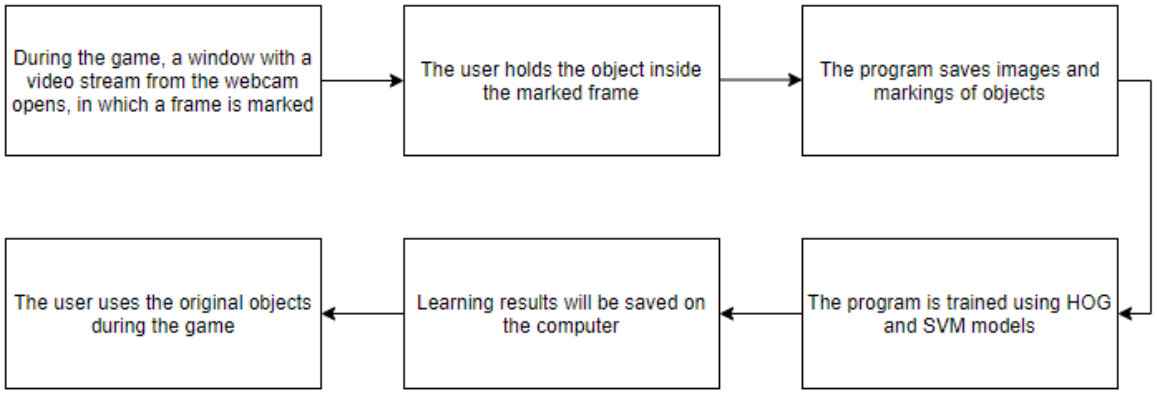

Figure 4. Scheme of a player interaction with a game situation.

3. Results and Discussion

3.1. Procedure

In the considered game situation, the training of the HOG-SVM model is possible in the game process, which suggests that the SVM is more suitable than deep learning, since this method requires less initial data and less time for training. Also it is be noted, in educational games, the background, lighting, distance, remain constant. This suggests that the original set can be simplified. In this approach, the video stream from the webcam is accompanied by a sliding window, in which it is necessary to hold the object under study. The scheme of a player interaction with the considered game situation is shown in Figure 4.

3.2. Dataset

Using a sliding window, two datasets for the palm and fist images were prepared, respectively. The first dataset included images with a monochrome background and the palm and fist were the only objects in the frame. While the second dataset in addition to target objects also included non-target objects, like a face, chair, microphone, room, dynamic objects in the frame. In addition, depending on the movement of the palm, the color shade of the object changed. The second dataset allows us to create a more natural environment in which the detector will operate. Each dataset includes 151 webcam screenshots.

3.3. Results and Discussion

Model training and testing takes place in two stages. At the first stage, training is performed on 80 % of the initial data. After training, testing is performed separately on the used data and 20 % of the remaining data. According to the results, the detection accuracy of the first set is 98%, the detection accuracy of objects in new images is 96%. At the second stage, training takes place on 100 % prepared initial data. Retesting on images that participated in the training showed an accuracy of 97 %. Classification performance parameters is presented in Table 1 where TP (true positive) indicates that the classifier correctly attributed the object to the class in question, TN (true negative) indicates that the classifier correctly asserts that the object does not belong to the class in question, FP (false positive) indicates that the classifier has incorrectly assigned the object to the class in question, and FN (false negative) indicates that the classifier incorrectly asserts that the object does not belong to the class in question.

As we expected, on the second case, the detection accuracy has decreased. The accuracy in training results on 80 % of the data and testing on the remaining 20 % was found to be

Table 1. Classification performance parameters

Parameter	Formula	Dataset 1	Dataset 2
Precision	TP/(TP+FP)	0.99	0.97
Recall/Sensitivity	TP/(TP+FN)	0.97	0.92
Accuracy	(TP+TN)/(TP+TN+FP+FN)	0.98	0.93

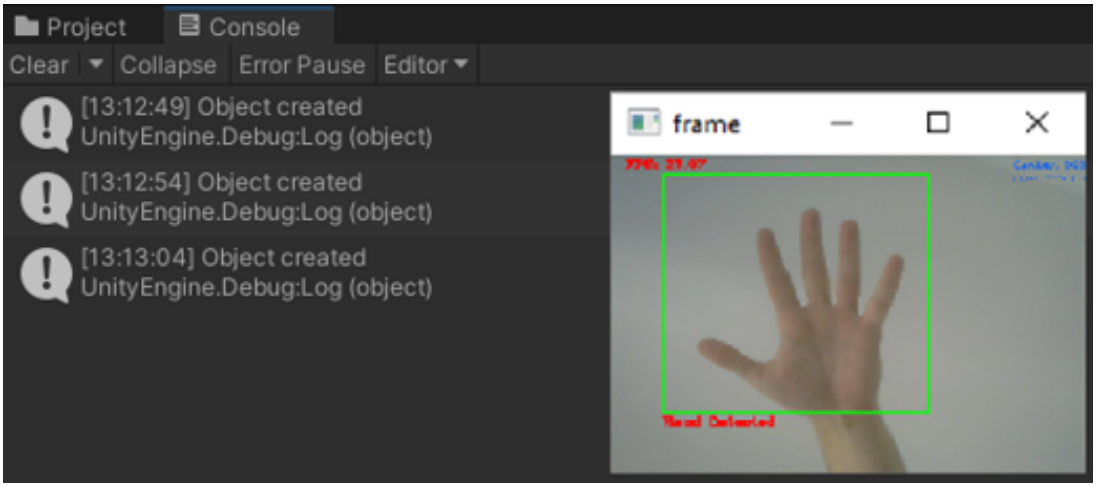

Figure 5. Creating an object with the open palm.

93 %. Real-time testing of the detector shows that one of the key facts in the detection accuracy when training on a small sample is the light entering the object. The detector poorly detects an object without lighting or too intense light (for example, when exposed to sunlight). One of the problems is low-quality webcams, which react strongly to sudden changes in light. Original images are cropped to 160 × 144 pixels. In this case, the size of the cells is 16 × 16 pixels. Accordingly, the image consists of 39 horizontal blocks and 35 vertical blocks. Each block forms a 36 × 1 vector of elements. So the length of the vector is:

$$h = 35 \times 39 \times 36 = 49,140.$$

It is worth noting that in used datasets, lighting, object appearance, and object distance do not change. Therefore, the detection accuracy was high. To achieve a better level of detection, it is necessary to prepare more initial data, with different conditions (background, lighting). Finally, the resulting detector was used in Unity3D software to create (Figure 5) and remove objects (Figure 6) from the playing field [12]. In Figures 5, 6, the logs confirm the completion of creation and deletion of an object, respectively.

When using lightly loaded game scenes, the performance drops do not affect the gameplay. For the test we used a AMD Ryzen 3 1200 processor. During the detector's operation, the processor load ranges from 10 % to 14 %, the amount of consumed RAM is about 54 MB. When using the detector, the number of video frames drops from 60 to 29 frames/s.

4. Conclusion

The introduction of modern technologies in educational computer games may interest students in learning new technologies and working in scientific fields. The proposed implementation of computer vision allows us to quickly and efficiently connect programs for detecting objects to the game process. This approach has several advantages: quick creation of the initial sample

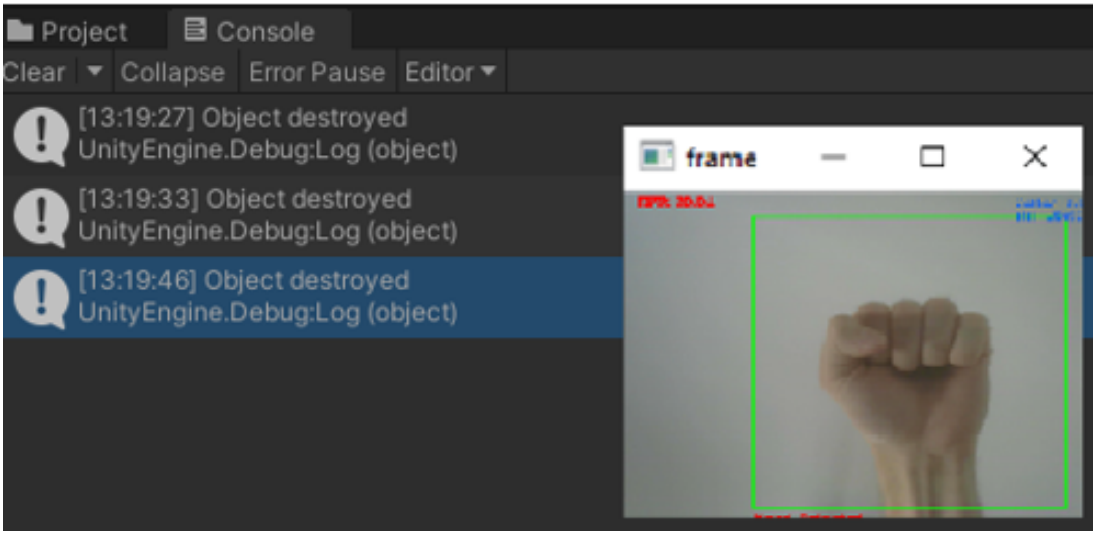

Figure 6. Removing an object with a clenched fist.

for training the model, fast model training, and low load on the central processor and low consumption of RAM.

The listed features make it possible to use computer vision and object detection in lightly loaded and non-dynamic scenes. Taken together, this helps to reduce the requirements for the playr's computer. In the process of work, a hand gesture classifier was used. Using the HOG feature extraction method, we obtained images described by a vector, the length of which is $h = 49,140$. At the same time, the recognition accuracy, using the SVM classifier, reaches 98 %.

In the future, it is planned to compare various methods for determining objects, including deep learning. It is important to identify not only the quality of the comparison, but also the speed and load on a computer.

References

[1] Wastiau P and Kearney C and Van den Berghe W 2009 How are digital games used in schools? Complete results of the study. European Schoolnet, EUN Partnership AISBL `http://games.eun.org/upload/gis-full_report_en.pdf`
[2] Fitri R I, Lavicza Z and Houghton T 2021 *Contemporary Educational Technology* **13** 299
[3] Manesis D 2020 *Digital Games in Primary Education* (Ionian University) chap 6, pp 543–434
[4] Homer B D, Kinzer C K, Plass J L, Letourneau S M, Hoffman D, Bromley M, Hayward E O, Turkay S and Kornak Y 2014 *Computers and Education* **74** 37–49
[5] Xu M, Zhai Y, Guo Y, Lv P, Li Y, Meng W and Zhou B 2019 *Journal of Visual Communication and Image Representation* **62** 394–401
[6] Semenov M, Colen Y S, Colen J and Pardala A 2020 *Journal Korean Soc. Math. Educ., Ser. D, Res. Math. Educ.* **23** 47–62
[7] Žemgulys J, Raudonis V, Maskeliūnas R and Damaševičius R 2018 *The Workshop on Computational Intelligence in Ambient Assisted Living Systems (CIAALS 2018)* vol 130 p 953–960
[8] Dalal N and Triggs B 2005 *In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (San Diego, United States) pp 886–893
[9] Freeman W and Roth M 1995 *Workshop on Automatic Face and Gesture Recognition* 296–301
[10] Naveenkumar M and Vadivel A 2015 *Proceedings of National Conference on Big Data and Cloud Computing* **15** 52–56
[11] Yadav R, Senthamilarasu V, Kutty K and Ugale S 2015 *International Journal of Computer Applications* **19** 10–16
[12] Chursin G and Semenov M 2020 *Journal of Physics: Conference Series* vol 1611 p 012059