Real-Time Object Detection in Video Streams on Low Performance Embedded Systems

Karpov Alexey*
Moscow State University,
Department of Mechanics and Mathematics

Abstract

Object detection on images is an important task of computer vision. Images are often extracted from continuous video streams, since many real-life applications of object detection are various control systems. This article presents the way to utilize continuity of a video stream in order to speed up object detection by Viola-Jones-like [1] [2] algorithms.

Keywords: *object detection, face detection, embedded system.*

1. INTRODUCTION

If one looks closely at the real-file applications of object detection, such as various surveillance and control systems, human-computer interaction systems, internet conferencing, one can see that those applications require analysis of a continuous video stream. Moreover the analysis has to be done in a real-time, hence object detection also has to be done in a real-time. At the same time computers used for such tasks have to be compact, power efficient, able to work in a harsh environment, therefore usually compromising its speed. The algorithm proposed in the article is required to work in a real-time (more than 1 frames per second performance) on a TMS320 based embedded system.

2. THE ALGORITHM

The algorithms allowing (with modifications) to achieve acceptable real-time performance on the embedded system are Viola-Jones type boosted cascade algorithms [1] [2]. However these methods do not take into account that object detection is performed in a video stream. The following modifications are proposed so as to increase the performance of the base object detection algorithm.

2.1 Related Methods

The problem of object detection in a video stream is not new. There are comprehensive taxonomies [3] and papers [4] [5] which give object detection algorithms using movement detection and segmentation. Therefore such algorithms try to detect objects through evaluation and analysis of the movement regions and object tracking. These algorithms are conceptually different than the algorithm presented in this article, since the object detection itself is done using rather effective boosted cascade algorithm. Furthermore, not all of the algorithms can be implemented for a real-time environment while maintaining satisfactory detection quality. Other algorithms [6] [7] attempt to use probabilistic approach for object detection in a video stream. Such algorithms try to achieve better detection quality other than processing speed. However, recognizing the quality of object detection of Viola-Jones based algorithms as satisfactory, this article concentrates on increasing of object detection speed in a video stream.

The idea behind modification is that consecutive frames in a video stream generally tend to have regions that differ insignificantly. Hence, the object detection should be performed only in changing regions or, more precisely, the object detection should not be performed in regions where has been no changes. The algorithm tries to find only one object in a frame of a video stream (the extension for multiple objects is not difficult to implement).

2.2 Frame Differencing

The methods described in [3] [4] [5] perform analysis of each pixel in a video stream to determine if it has sufficiently changed. Thus such analysis requires significant amount of time. In order to quickly yet reliably estimate region of variance of two consecutive frames, the following method is proposed. One of the algorithms for image segmentation described in [8] uses column and row sums of the image to extract foreground. The technique is called amplitude projection segmentation. If one applies the technique to the difference of two frames, one gets the estimation of variance region for the frames. That is to say, let I_{n-1}, I_n – two consecutive frames from a video stream, then

$$HSumDiff(i) = \left| \sum_{j=1}^{width} I_{n-1}(i,j) - \sum_{j=1}^{width} I_n(i,j) \right|,$$

$$VSumDiff(j) = \left| \sum_{i=1}^{height} I_{n-1}(i,j) - \sum_{i=1}^{height} I_n(i,j) \right|.$$

Let $Hthr$ and $Vthr$ are the thresholds to control sensitivity of variances in rows and columns, respectively. Then

$$imin = \min_i \{i \mid HSumDiff(i) > Hthr\},$$

$$imax = \max_i \{i \mid HSumDiff(i) > Hthr\},$$

$$jmin = \min_j \{j \mid VSumDiff(j) > Vthr\},$$

$$jmax = \max_j \{j \mid VSumDiff(j) > Vthr\}.$$

Finally the region of variance is a rectangular area with upper left vertex at $(imin, jmin)$, width of $jmax - jmin$ and height of $imax - imin$. It should be noted that the sums $HSumDiff(i)$ and $VSumDiff(j)$ can be easily calculated from data of the base object detection algorithm as difference of the integral sums. Also the algorithm does not require to store the previous frame, but just the sums.

*karpov@fit.com.ru

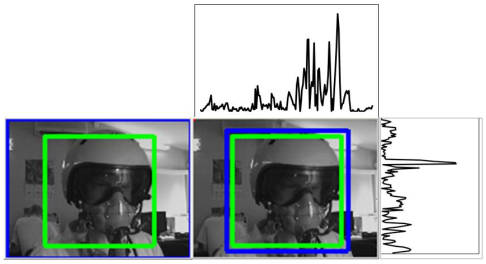

Figure 1: *Amplitude projections differencies example*

2.3 Region of Interest Computation

The object detection is performed in the region of interest (ROI) – specific rectangular area. Below is given the algorithm for determining of the region of interest. To begin with, let the *active region* (AR) is an area of frame, where the object may occur. At first the active region is the whole frame. Consider the following:

- Calculate region of variance (ROV)
- Consider the following cases
    1. If the object has been detected on the previous frame (in the rectangular area $ROBJ$), then
        (a) if ROV is too small (practically less than half of minimum possible object size), return $ROBJ$
        (b) if $ROV \bigcap ROBJ$=Ø, then $AR = AR \bigcup ROV$, return $ROBJ$
        (c) if $ROV \bigcap ROBJ \neq$ Ø, then
            i. $ROI = ROBJ \bigcup ROV \bigcup AR$,
            ii. $ROBJ = \mathrm{DetectObject}(ROI)$,
            iii. if object is found then $AR = ROBJ$, return $ROBJ$ else $AR = ROI$, return $NOOBJFOUND$
    2. If the object has not been detected on the previous frame, then
        (a) if ROV is too small, then return $NOOBJFOUND$
        (b) if ROV is big enough, then
            i. $ROI = ROV \bigcup AR$,
            ii. $ROBJ = \mathrm{DetectObject}(ROI)$
            iii. if the object is found then $AR = OBJ$, else $AR = ROI$. Return $ROBJ$.

The result of operation of joining ($\bigcup$) two rectangular areas is a bounding box area of these rectangular areas.

2.4 Results

The proposed algorithm was tested on video streams with different frame rates (see tables 1, 2 for results). As one should expect, the more frames per second has a video stream, the more increase of processing speed is achieved by using the modified algorithm. It is important to note, that the quality of object detection is almost the same. The algorithm can be implemented as a real-time algorithm, since one can derive formulae showing upper bound for processing time for given parameters for the base algorithm and ROV calculation complexity is very low and depends only on height and width of a frame.

Video frame rate	Original algorithm	Modified algorithm
∼1 fps	18.5	21.9
4 fps	73	42.5
10 fps	430	225

Table 1: *Processing time (seconds) for the sequences*

Video frame rate	Original algorithm	Modified algorithm
∼1 fps	0.76 / 0.025	0.74 / 0.03
4 fps	0.93 / 0.001	0.9 / 0.001
10 fps	0.83 / 0.003	0.84 / 0.01

Table 2: *Performance comparison (hit rate/false alarms rate)*

3. FUTURE WORK AND CONCLUSION

The article presents method for object detection in video streams in a real-time. The method is implemented on the embedded systems based on TMS320 CPU. The method is part ASKPM (railroad safety) system which is put on approve test on Russian Railways locomotive ChS2K type. The method is also part of BASBP (aircraft safety) system which is put on approve test on planes produced by the Russian Aircraft Corporation MiG, Sukhoi design bureau and Yakovlev design bureau.

In the future, various improvements are planned for the algorithm, such as more precise region of variance determination by using more localized features, deeper feature history.

4. REFERENCES

[1] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” 2001.

[2] "Rainer Lienhart et al, “Empirical analysis of detection cascades of boosted classifiers for rapid object detection.,” in *"DAGM-Symposium"*, 2003, pp. "297–304".

[3] Jacinto C. Nascimento and Jorge S. Marques, “Performance evaluation of object detection algorithms for video surveillance,” *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 761–774, 2006.

[4] Rita Cucchiara et al, “Detecting moving objects, ghosts, and shadows in video streams,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1337–1342, 2003.

[5] Robert T. Collins et al, “A system for video surveillance and monitoring,” 2000.

[6] Jacek Czyz, “Object detection in video via particle filters,” *Pattern Recognition, International Conference on*, vol. 1, pp. 820–823, 2006.

[7] Kenji Okuma et al, “A boosted particle filter: Multitarget detection and tracking,” in *In ECCV*, 2004, pp. 28–39.

[8] William K. Pratt, *Digital Image Processing: PIKS Inside*, John Wiley & Sons, Inc., New York, NY, USA, 2001.