Interesting Video Frames Capturing on Digital Video Development Platform

D. Nethaji

Applied Electronics, Velammal EngineeringCollege, Chennai, INDIA, dnethajianandhavalli@gmail.com

Interesting Video Frames Capturing on Digital Video Development Platform

Mary Joans, Mrs. S. J. Grace Shoba & D.Nethaji*

Applied Electronics, Velammal Engineering College, Chennai, INDIA

Email: dnethajianandhavalli@gmail.com

Abstract - Video surveillance has been a popular security tool for years. Video surveillance systems produce huge amounts of data for storage and display. Long-term human monitoring of the acquired video is impractical and in-effective. This paper presents a novel solution for real-time cases that identify and record only "interesting" video frames containing motion. In addition to traditional methods for compressing individual video images, we could identify and record only "interesting" video images, such as those images with significant amounts of motion in the field of view. The model would be built in simulink, one of tools in matlab and incorporated with davinci code processor, a video processor. That could significantly help reduce the data rates for surveillance-specific applications.

Keywords— *Code Composer Studio (CCS), MATLAB, Digital video development hardware.*

I. INTRODUCTION

A video surveillance system covering a large office building or a busy airport can apply hundreds and even thousands of cameras.

Abnormal motion detection is the key to effective and economical video surveillance. The detection of an abnormal motion can trigger video transmission and recording, and can be used to attract the attention of a human observer to a particular video channel. The problem is characterized by three related challenges. One is the reliability requirement, meaning that irregular events should be consistently detected, while the false-alarm rate should be sufficiently low. The second is effective characterization of normal motion, allowing discrimination between normal and abnormal activity. Third, abnormal motion detection should be accomplished using the limited computational power available at or near the camera.

This paper presents a novel solution to capture the frame that conation motion scheme relying on Sum of absolute differences (SAD). This is both the most obvious and most simple algorithm of all: The two consecutive frames are compared pixel by pixel, summing up the absolute values of the differences of each two corresponding pixels. The result is a positive number that is used as the score. SAD reacts very sensitively to even minor changes within a scene: fast movements of the camera, explosions or the simple switching on of a light in a previously dark scene result in false hits. On the other hand, SAD hardly reacts to soft cuts at all. Yet, SAD is used often to produce a basic set of "possible hits" as it detects all visible hard cuts with utmost probability. Motion vectors on pixel data reduces the input data rate by about two orders of magnitude, and allows real-time operation on limited computational platforms.

Previous works that rely on segmentation, grouping have human motion, such as periodicity, gait or gestures, can be found in [1, 12, 19, 11]. In [11], principal component analysis of the macro block motion vectors was used to match the detected activity in a video stream to known human activities (walking, running, kicking), and for selective access of details from the uncompressed domain. Novelty or activity detection in video using pixel-level motion analysis has been reported by [8, 5].been reported in [7, 2, 13, 20, 16, 14, 6, 10].

Unlike most previous methods for video analysis, the suggested approach completely avoids segmentation, etc. In the next section, the implementation of SAD is presented. Section III proposed the SAD in simulink model. Implementing SAD with Davinci code processor is focused in section IV. Conclusion and future work is given in section V.

II. IMPLEMENTATION OF SAD

To perform the appropriate video surveillance, initially the frame analyses need to done. In which, the changes in the pixels between the consecutive frames need to be identified. In the proposed method, the sum of absolute differences (SAD) method is used, which a popular technique for motion detection in video is processing. The SAD block finds the similarity between two input images by performing the sum of absolute differences. The greater the similarity between the two matrices, the smaller the SAD values that result. Suppose input matrix I has dimensions (Mi, Ni) and the input

matrix Template has dimensions (Mt, Nt). The equation for the two-dimensional discrete SAD is,

$$X(\varphi,\kappa)=\sum\sum \alpha\beta\sigma(I(\mu+\varphi, \nu+\kappa)-T(\mu,\nu)) \text{ ————— (1)}$$

Where

$$0\leq\varphi\leq M\iota-M\tau+1 \text{ ———————————— (2)}$$

$$0\leq\square\leq N\iota-N\tau+1\text{———————————— (3)}$$

Sum of Absolute Differences (SAD) is one of the simplest of the similarity measures which is calculated by subtracting pixels within a square neighborhood between the reference image and the target image followed by the aggregation of absolute differences within the square window, and optimization with the winner-take-all (WTA) strategy [1].

III. SAD IN SIMULINK

The above described sum of absolute differences has been constructed in Simulink, one of tool in matlab. With Simulink, building models has been dragged and dropping blocks from the library browser onto the graphical editor and connecting them with lines that establish mathematical relationships between the blocks. Thus simulink would provide the easier way of building the dynamical architecture, which in turn can be incorporated with any external target too. It has ability of generating the code of ANSI c/c++. Thus the SAD is been subsystems in the model, which incorporate the functions like Generating data for motion energy and threshold display and Comparing Motion Energy against Threshold.

As explained in above context, SAD would perform the operation of comparison of pixels, which indirectly mean the result that they would generate the data for motion energy and threshold display. This has been get done by matrix Concatenate block, which is available in simulink library. The Concatenate block concatenates the signals at its inputs to create an output signal whose elements reside in contiguous locations in memory. This block operates in either vector or multidimensional array concatenation mode, depending on the setting of its Mode parameter. In either case, the block concatenates the inputs from the top to bottom, or left to right, input ports.

Similarly the desired motion energy found in previous context would be compared with the threshold point by means of relational comparator. The relational operator is also one of the tools which available in the simulink library, which perform comparing two inputs using the Relational operator parameter that you specify. The first input corresponds to the top input port and the second input to the bottom input port by default. Thus the desired result of capturing the motion energy and comparing with the threshold point is shown in figure 1.
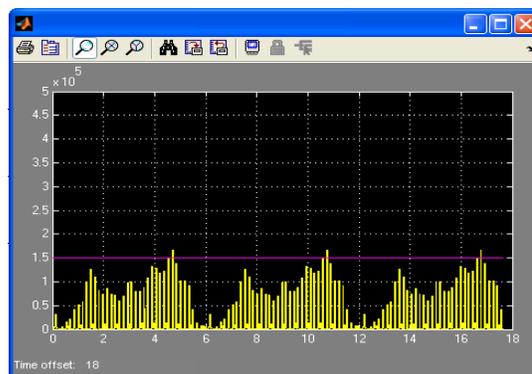

Fig1. Implies motion energy (yellow lines) and threshold point (magenta line).

In this scope display, yellow represents the motion estimate from each video frame, and magenta is the threshold level. In the Motion Threshold scope here, we see that we identified three frames with motion above the threshold that were captured and recorded.

IV. SAD WITH DM6437

In order to improve the results of the video surveillance associated with SAD algorithm described above, we use davinci code processor; it's an image video development platform from Texas instrumentation, which is shown in figure 2.
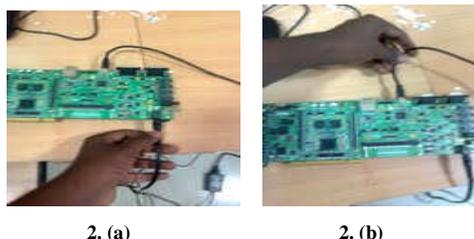

| 2. (a) | 2. (b) |

Fig2. DM6437 with supplied voltage and USB connection

The processes here involve the model of SAD algorithm, drawn in simulink, shown in figure 3.
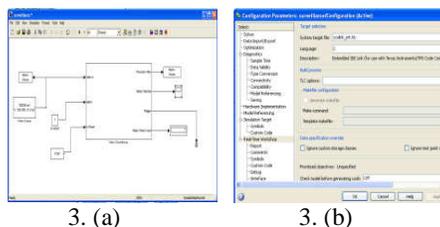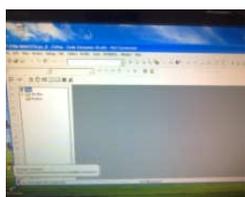

| 3. (a) | 3. (b) |

Fig3. Simulink model and its connection to CCS

The code can be generated through real time workshop via target language compiler in simulink. The desired code is then viewed in code composer studio software tool, shown in figure 4.

4.(a)

4. (b)   4. (c)

Fig4. CCS before and after connecting to processor

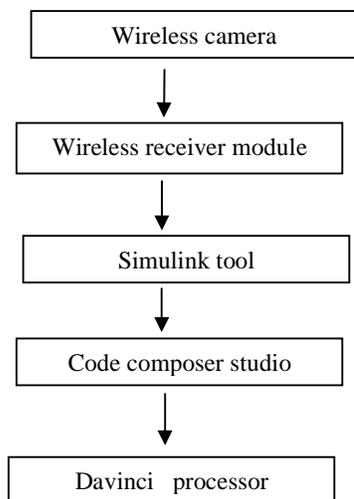The objective of code composer studio is to inject the code onto davinci code processor.

Fig 5. Flow of mechanism between matlab and Davinci

Thus the generated code from the simulink is seen through code composer studio is shown as figure 6.
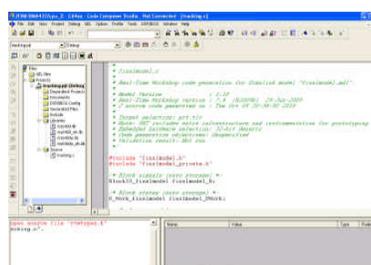

Fig6. Generated code in code composer studio

And thereby in turn, it would generate the machine code for the davinci processor. Thus the achieved code is feed to davinci through USB JTAG. Thus as we run the processor, the desired result can be seen in matlab too, which is shown in figure 7.

7 . (a)   7.(b)

Fig7. Final output of interesting frames capturing

V. CONCLUSION AND FUTURE WORK

The proposed method is well-suited for modern video-surveillance architectures, Continuous-scene monitoring applications, such as ATM booths, parking lots or traffic monitoring systems, generate large volumes of data. Recording and archiving such volumes of data is a real problem, and one way to solve this is to reduce the size of the data stream right at the source. Thus the proposed method can be enhanced with multiple applications by means of interacting with advanced video processor.

REFERENCES

[1] J.K. Aggarwal and Q. Cai, Human motion analysis: a review, CVIU, Vol. 73, No. 3, pp. 428-440, 1999.

[2] C. Beleznai, B. Fruhst¨uck¨ and H. Bischof, Tracking mul-tiple humans using fast mean shift mode seeking, Int. Workshop on Visual Surveillance, pp. 25-32, 2005.

[3] V. Bhaskaran and K. Konstantinides, Image and Video Compression Standards: Algorithms and Architectures, Kluwer, 1997.

[4] O. Boiman and M. Irani, Detecting irregularities in im-ages and in video, ICCV,2005.

[5] R. Hamid, Y. Huang and I. Essa, ARGMode - Activity recognition using graphical models, CVPR, Vol. 4, pp. 38-44, 2003.

[6] Haritaoglu, D. Harwood and L.S. Davis, W4: Real-time surveillance of people and their activities, PAMI, Vol. 22, No. 8, pp. 809-830.

[7] R.S. Gaborski, V.S. Vaingankar, V.S. Chaoji, A.M. Tere-desai and A. Tentler, VENUS: A system for novelty detection in video streams with Event detection and analysis from video streams, PAMI, Vol. 23, No. 8, pp. 873-889, 2001.

[8] B. Ozer, W. Wolf and A.N. Akansu, Human activity detection in MPEG sequences, Workshop on Human Motion pp. 61, 2000.

[9] R. Polana and R. Nelson, Detection and recognition of periodic non-rigid motion, IJCV, Vol. 23, No. 3, pp. 261-282, 1997.

[10] S. Rao and P.S. Sastry, Abnormal activity detection in video sequences using learned probability densities, TENCON, 2003, Vol. 1, pp. 369-372.

[11] C. Rao, A. Yilmaz and M. Shah, View-invariant representation and recognition of actions, IJCV, Vol. 50, pp. 203-226, 2002.

[12] J. Sherrah and S. Gong, VIGOUR: A system for recognition of multiple people and their activities, ICPR, pp. 179-182, 2000.

[13] C. Stauffer and W.E.L. Grimson, Learning patterns of activity using realtime surveillance , PAMI, Vol. 22, No. 8, 747-757, 2000.

[14] A. Veeraraghavan, A.R. Chowdhury and R. Chellappa, Role of shape and kinematics in human movement analysis, CVPR, Vol. I, pp. 730-737, 2004.

[15] T. Xiang and S. Gong, Beyond tracking: modelling activity and Understanding behavior, IJCV, Vol. 67, No. 1, 21-51, 2006.

[16] Y. Yacoob and M.J. Black, Parameterized modeling and recognition of activities, ICCV,1998.

[17] H. Zhong, J. Shi and M. Visontai, Detecting unusual activity in video, CVPR, Vol. 2, pp. 819-826, 2004.

❑❑❑