Implementation of a Fire Detection Algorithm on TMS320DM642 DSP using MATLAB/Simulink*

Bingsong He, Xueping Zhao, Zhiguo Zhou⊠ and Zheyi Fan

School of Information and Electronics, Beijing Institute of Technology, Beijing, China
zhiguozhou@bit.edu.cn

Abstract - This paper introduces a new approach "model-based design" to develop and implement a fire detection algorithm using the TMS320DM642 DSP and MATLAB/Simulink. The fire detection algorithm model is constructed in Simulink in the form of graphical blocks. C code is automatically generated from the block-diagrams with the Real-Time Workshop (RTW). And an executable file, produced from the C code through the TI DSP c-compiler, is downloaded into the target DSP board and implemented through the DSP. This design methodology makes an easy way for the implementation of complex and high level algorithm in a digital signal processor (DSP).Experimental results demonstrate that the proposed approach not only has a good detection performance, but also simplifies the developing process effectively.

Index Terms - Model-Based; DSP; Fire Detection

1. Introduction

Nowadays, there are many requests for using advanced algorithms in the DSP applications. Such algorithms are usually very complex. However, the programming of the DSP board is possible only in the C and the assembler programming languages. For that reason a lot of work is needed for the realization of the control algorithms on the DSP board. Using the text programming languages, with the increasing complexity of the algorithm, the likelihood of errors in the code drastically increases. There are also numerous requests for shortening development time, increasing quality, reliability and safety of the final product. Therefore the need for new methods and tools that enable faster development and implementation of complex algorithms increases.

MATLAB/Simulink is one of the most widely used tools for DSP system simulation and design. A majority of the control algorithms are previously simulated and tested in the MATLAB/Simulink [1]. Real-Time Workshop (RTW) is the Simulink add-on software that enables automatic C or ADA code generation from the Simulink model [2]. To avoid here stated problems using the C code programming, we decide to take advantage of the MATLAB/Simulink and the RTW to develop and implement the fire detection algorithm.

As shown in Fig. 1, the desired aim is graphically presented. The fire detection algorithm is designed and simulated in the Simulink and after successful simulation the Simulink model is converted with the RTW into the C code.

The paper is organized as follows: In section 2 software and hardware requirements about the system are stated while in section 3 the fire detection algorithm is discussed and the corresponding models are described. In section 4 the simulation and hardware implementation results are presented respectively. Finally, section 5 is devoted to the conclusions.

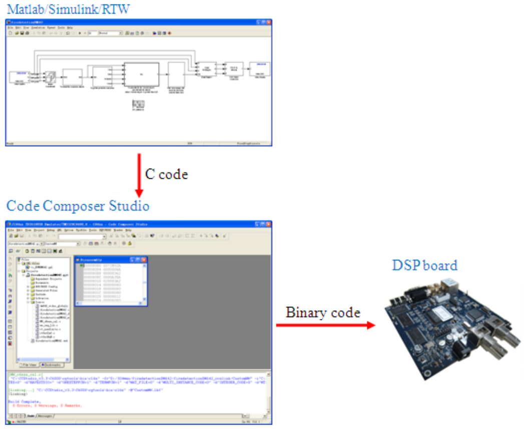

Fig. 1 Programming of the DSP board with the Simulink

2. Hardware and Software Requirements

The tools used for implementation is MATLAB R2009a with Embedded Target for TI C6000. The Code Composer Studio (CCS) v3.3 platform is used to interface DM642 DSP hardware. The entire simulation work is done using Simulink in MATLAB.

The TMS320DM642 (DM642) is a fixed point video-oriented DSP [3]. The CPU is an VLIW processor with a performance of up to 4800 MIPS @ 600 MHz.DM642 has three configurable video port peripherals (VP0, VP1, and VP2). VP0 and VP1 are used for capture ports and VP2 is used as a display port.

3. Model-Based Design Fire Detection Realization

To achieve automatic surveillance of fires, an intelligent fire detection method based on a 2-stage decision strategy of video processing is proposed. The first decision stage is to extract fire-colored pixels through 3 rules from visual images. In the second decision stage, dynamic features are adopted to distinguish other fire aliases and validate a real burning fire.

* This work is partially supported by National Natural Science Foundation of China (Grant #61001063) and Science Foundation of BIT (Grant #20120542017).

Furthermore, corresponding models that realize the algorithms are given respectively.

A. Potential Fire Region Detection with Chromatic Features

The color information is one of its important static characteristics of a flame, which can separate the flame from a large portion of other objects. In this paper, we adopt rule-based color detection approach proposed by Chen [4] as it is simple and effective. They used the RGB color space and HSI color space. In spite of various colors of fire flames, the initial flames frequently display red-to-yellow color. Thus for a given fire pixel, the value of the red channel is greater than that of the green channel, and the value of the green channel is greater than that of the blue channel. The first rule in detecting fire colors is defined as $R \geq G \geq B$. Furthermore, there should be a stronger R in the captured fire image due to the fact that R becomes the major component in a colorful image of fire. This is because that fire is also a light source and the video camera needs sufficient brightness during the night to capture the useful video sequences. Hence, the second rule adds a constraint that the value of R component should exceed a threshold R_T . However, lighting conditions in the background may adversely affect the saturation of flames or generate a fire alias, which leads to false detections. For this reason, when determining the pixel color, the saturation value of fire pixel should be larger than a specified threshold. The whole rules for extracting fire regions from an image are described as follows.

$$\begin{aligned} &R \geq R_T \\ &R \geq G \geq B \\ &S \geq (255 - R) \cdot S_T / R_T \end{aligned} \quad . \qquad (1)$$

Where R_T is the threshold of red component, ranging from 115 to 135 and S_T is the saturation threshold, ranging from 55 to 65. Both R_T and S_T are defined according to various experimental results. Pixels which are satisfied with all the three rules are regarded as fire pixels.

According to the basic principles above, blocks in Video and Image Processing Blockset, Signal Processing Blockset and Simulink are used to build a model of the algorithm. The block diagram is shown in Fig. 2. The function of the detecting rules is written as M code. An Embedded MATLAB Function block is used to include the M code in the model. The created MATLAB function executes for simulation and generates code for a Real-Time Workshop target.

B. Real Fire Region Validation with the Dynamic Feature

To improve the reliability of detection, dynamic features are usually adopted to distinguish fire aliases, including sudden movements of flames, changeable shapes, growing rate, and oscillation (or vibration) in the infrared response. In this paper, circular degree is utilized to check a real burning fire.

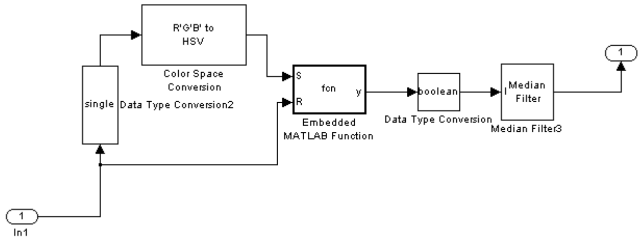

Fig. 2 Block diagram of potential fire region detection subsystem

The circular degree of an object can measure the degree of the regularity. Generally, the flame shape is complex and irregular while most sources of interference (such as lights, cigarette butts, etc.) have a high degree of shape regularity. Therefore for the regions detected by the chromatic features, the more irregular of a region, the greater its circular degree value is. Circular degree [5-6] is calculated as

$$C_k = \frac{P_k^2}{4\pi A_k} \qquad k = 1, 2, \cdots, n \; . \qquad (2)$$

C_k is the circular degree of the region k , P_k is the perimeter of the region k , A_k is the area of the region k , n is the number of regions in the image. There is a certain threshold value C_0 in this paper, the region is not the flame area while $C_k < C_0$, and it is flame area while $C_k \geq C_0$.

Fig. 3 shows the block diagram of the validation subsystem. The flame suspicious regions segmented from the background of the image in the first decision stage are passed to the Blob Analysis block in order to obtain the perimeter for each region and the corresponding region area. The Blob Analysis block in Simulink is very similar to the "regionprops" function in MATLAB. They both measure a set of properties for each connected object in an image file. The properties include area, centroid, bounding box, major and minor axis, orientation and so on. In this project, the area, bounding box, perimeter and count measurement are utilized. Thus the Blob Analysis block has four output values. Area represents the number of pixels in labeled regions. BBox is a 4-by-N matrix of bounding box coordinates, where N is the number of blobs. Perimeter is a vector containing an estimate of the perimeter length, in pixels, for each blob. Count represents the actual number of labeled regions in each image. The M code performing calculation of C_k and deciding the real fire region is included in an Embedded MATLAB Function block.

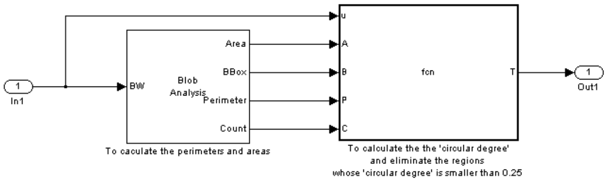
1
In1
BW
Blob Analysis
Area
BBox
Perimeter
Count
To caculate the perimeters and areas
u
A
B
P
C
fcn
T
To calculate the the 'circular degree' and eliminate the regions whose 'circular degree' is smaller than 0.25
1
Out1

Fig. 3 Block diagram of the validation subsystem

C. *Complete Models*

The subsystems shown in Fig. 2 and Fig. 3 are merged together and designed as a single fire detection system. The complete graphic model of the proposed algorithm in MATLAB/Simulink is shown in Fig. 4. The subsystem in Fig. 2 is masked in a subsystem block, followed by the subsystem in Fig. 3. After the 2-stage processing, the image is passed to the third block from right to determine the zero-based shape coordinates of the fire region. The Draw Shapes block draws rectangles on images to indicate fire. In the simulation stage, a From Multimedia File block reads video frames from a multimedia file and imports them into the Simulink model. The original video and the processed video are displayed by two To Video Display blocks.

To transform the Simulink model to a DSP implementable model that can use the RTW and TC6 to target a DSP, the following parts of the Simulink model are modified.

- The From Multimedia File block and the last To Video Display block are replaced by the Video Capture block and the Video Display block in DM642 EVM Board Support Library respectively. Another unused To Video Display block is deleted.
- A target preference block is placed in the implementation model.
- A Color Space Conversion block is added to convert color information between color spaces. And a Matrix Concatenate block is inserted to concatenate input signals of same data type to create contiguous output signal.

After these modifications, the implementation model is shown in Fig. 5. The code can be generated from the modified model if the "Build" button is pressed, and the code will be loaded to the CCS as a CCS project automatically, then it will be built in CCS, and a executable file (*.out) will be generated if no errors appeared.

4. RESULTS

A. *Simulation Results*

The simulation results are shown as follows. Fig. 6 shows the original input video. There is a burning fire and also a candle flame in the image. As we can see from Fig. 7, after the first decision stage fire-colored pixels are extracted, including the candle flame which is not expected to be. While after the second-stage processing, as shown in Fig. 8, the interference, i.e. the candle flame is eliminated and the real burning fire is indicated by a rectangle.

B. *Hardware Implementation Results*

We implement the model in Fig. 5 on DM642 DSP board. The original video and the detecting result are shown in Fig. 9 and Fig. 10 respectively. From our observation, fire can be identified and indicated by a black rectangle within a video sequence successfully. And interference like candle flame is not detected as fire.

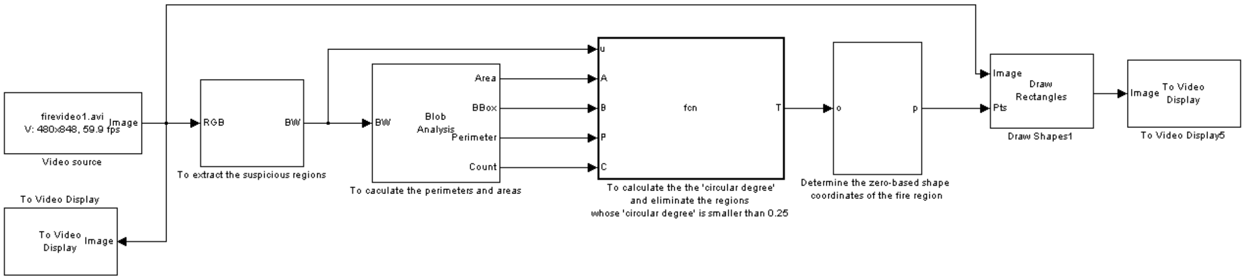
firevideo1.avi
V: 480x848, 59.9 fps
Image
Video source
To Video Display
To Video Display
Image
RGB
BW
To extract the suspicious regions
BW
Blob Analysis
Area
BBox
Perimeter
Count
To caculate the perimeters and areas
u
A
B
P
C
fcn
T
To calculate the the 'circular degree' and eliminate the regions whose 'circular degree' is smaller than 0.25
o
p
Determine the zero-based shape coordinates of the fire region
Image
Pts
Draw Rectangles
Draw Shapes1
Image
To Video Display
To Video Display5

Fig. 4 Simulation model of the fire detection system

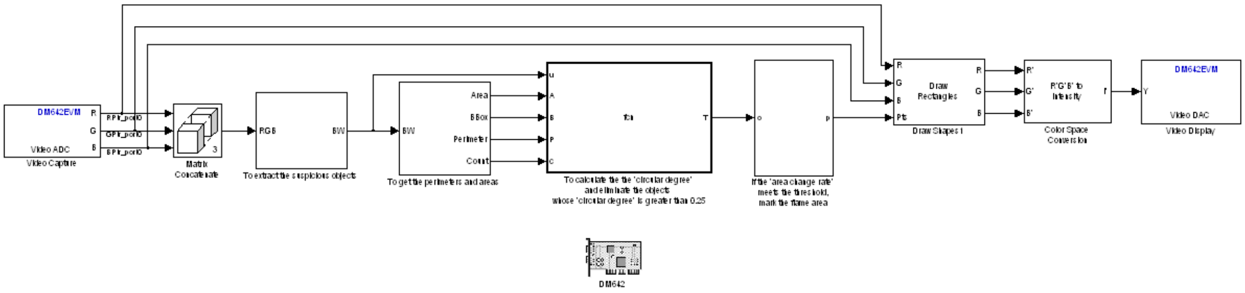

Fig. 5 Hardware implementation model of the fire detection system

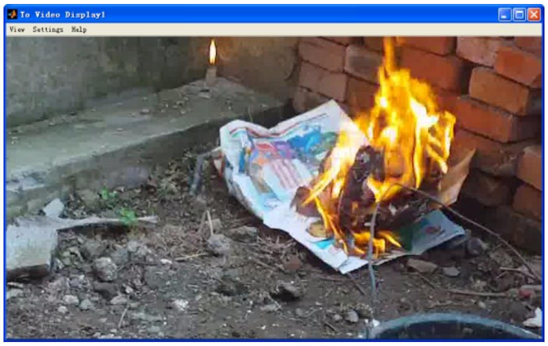

Fig. 6 Image of the original input video

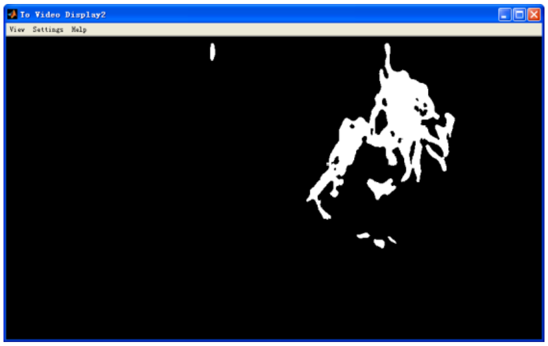

Fig. 7 Extracted fire-colored pixels after the first-stage processing

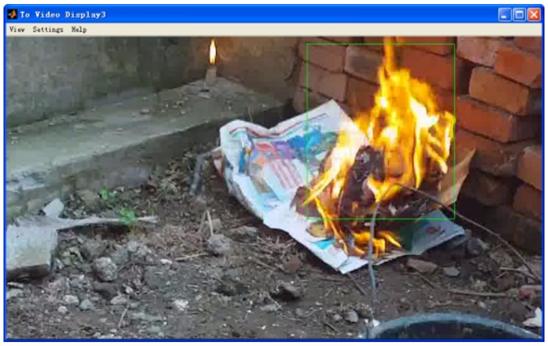

Fig. 8 Final fire region detection result

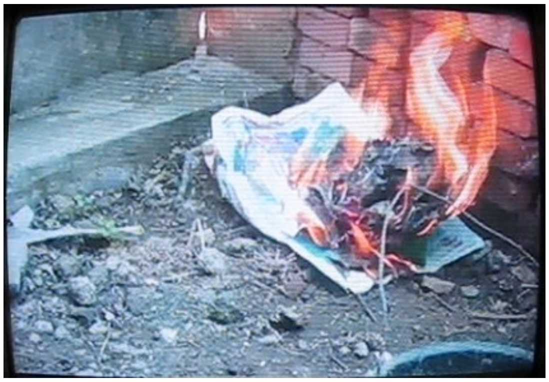

Fig. 9 Original input video displayed on hardware

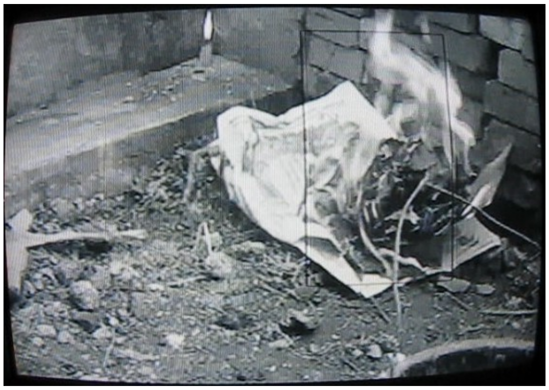

Fig. 10 Video of fire indicating by a box displayed on hardware

5. CONCLUSION

This paper proposed a model-based design for fire detection algorithm implemented on DSP. The fire detection algorithm was designed with Simulink in the form of block diagrams and the C code for this system was automatically generated from the block diagrams with the RTW. The simulation and experimental results are shown. They verify the validity of the model-based design. The model-based design can reduce design errors, cost and time. This provides a new idea for implementing fire detection algorithms on DSP.

6. REFERENCES

[1] The MathWorks, Inc.: *Using Simulink (sl_using.pdf)*, version 3, October 1999, www.mathworks.com.

[2] The MathWorks, Inc.: *Real-Time Workshop User's guide (rtw_ug.pdf)*, version 5, July 2002, www.mathworks.com.

[3] Texas Instruments, *TMS320DM642 Technical Overview*, September 2002.

[4] T.H. Chen, C.L. Kao, and S.M. Chang, "An intelligent real-time fire-detection method based on video processing," *in Proceedings of the IEEE 37th Annual (2003) International Carnahan Conference on Security Technology*, pp. 104-111, 2003.

[5] M. Li, "Method of early fire detection in large area based on double band Images," *Low Voltage Apparatus*, vol. 2, pp. 37-40, 2007.

[6] A.G. Wu, M. Li, and Y. Chen, "Research for image fire detection technology in large space," *Computer Measurement & Control*, vol. 14, pp. 869-871, 2006.