FPGA IMPLEMENTATION FOR REAL TIME CHROMA-KEY EFFECT USING ADAPTIVE FILTER

[1]ANAMIKA TIWARI, [2]GEETHA LAKSHMI BASKER, [3]P. KALAIARASI

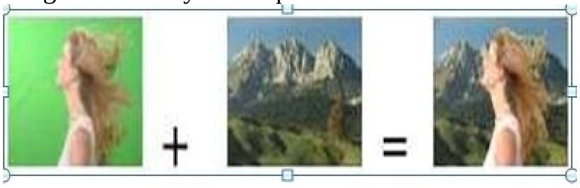
[1,2] ECE-Final Year, [3]M.Tech Assistant Professor Vel Tech High Tech Engineering College

Abstract— Chroma-key is a robust and important technique for processing image or video that is widely used in cinema films, magazine covers, video game industries as well as television programs such as weather forecast, live talk show, etc. This paper presents study of Chroma-key method and proposes a hardware architecture for implementing Chroma-key effect in real-time. Based on APC-OMS algorithm, we propose an improved method namely Adaptive Filter, which is more robust on segmentation and more appropriate for hardware design thanks to usage of small buffer. A VLSI architecture for the proposed method is implemented on the Altera cyclone II FPGA board. Experimental results show that the proposed design can perform Chroma-key effect with pleasing quality in real-time.

Keywords— APC algorithm, Coarse and Fine filter Chroma-key effect, K-means algorithm, OMS algorithm.

I. INTRODUCTION

Chroma-key is a technique using which a foreground object is extracted from a foreground frame and is combined with a background frame in order to create a new composite frame for special effects. The foreground frame has two parts consisting of foreground and background objects, in which the background object is a solid color, usually green or blue, due to which this chroma key technique is also known as green screen or blue screen technique. And the foreground object is an object placed before the screen. Fig. 1 shows an example of a foreground frame with green screen of background and a background frame. The frames are then processed by using Chroma-key technique.



(a) (b) (c)

Figure 1. Illustration of the Chroma-key effect. (a) Foreground frame, (b) background frame, (c) composite frame.

Chroma-key effect is often used in films, television programs, and in the weather forecasting, especially. The weather reporter generally stands in front of a green or blue screen. Meanwhile, the green or blue background color will be replaced with a weather map when weather report is telecasting. In general, the Chroma-key tools are used widely in creative fields and film industry. It is used to provide special effects or save a big amount of money in film industry while shooting the sections of a movie demanding foreign locations that can be performed within the studio to provide the effect of being in exotic locations abroad , while being in the city.

The Chroma-key effect can be done in post-production or in real-time. In post-production, one is used almost by software and in real-time, one is commonly used as a Chroma-key device to connect to or include in a camera. The Chroma-key effect includes a part of science and a part of art. Thus, in order to achieve the best result, techniques in studio and in processing Chroma-key effect by software or device should be combined together.

This paper introduces a Chroma-key core designed and implemented on FPGA to perform the Chroma-key effect in real-time.

This paper consists of six sections. Section I gives introduction to Chroma-key effect. Section II presents the existing models which can perform Chroma-key effect on Matlab software. In Section III an improved method is proposed which can be implemented on hardware. Then a proposed VLSI architecture for Chroma-key effect in real-time is described in Section IV. Section V shows the experimental results obtained from the implementation of the design on Altera Cyclone II FPGA board. The final section, the conclusion of this paper is shown in section VI.

II. EXISTING CHROMA-KEY METHODS

A common chroma-key method is to find a mask representing for the foreground region extracted from foreground frame, in which the pixels belong to foreground object will be equal to 1, and ones belong to background object will be equal to 0. Therefore, the mask will show up two colors, white and black. The values that are equal to 1 are white, and ones that are equal to 0 are black.

After the mask has been obtained, the Chroma-key effect is performed on foreground and background frames with the mask as below equation:

$$C = Fg.*mask + Bg.*(\sim mask) \qquad (1)$$

where C is composite frame, mask is the mask of foreground frame, Fg and Bg are value of all pixels in the RGB color space of the foreground and background frame, respectively.

Some of the existing methods that can perform the Chroma-key effect on MATLAB are referred and presented as follows:

A. Fixed Key Method

This method is demonstrated by Thilina Sammera. He proposed a simple method to find out a mask of the foreground frame. However, the use of the method to the Chroma-key process has to need a key value first. A manipulation scans overall values of the red (256 values), green (256 values), blue (256 values) components in each pixel of foreground frame to reach the key value. It takes many times to do that, so he has built a GUI (Graphic User Interface) to help finding the key value easily.

This method has proposed a simple equation to find out the mask of foreground frame. But if this one is used, finding a key value must be performed first by scanning over 256x256x256=16.777.216 values. It is hence very time-consuming and difficult in the finding key exactly; therefore, this method is not chosen to implement by hardware. The simulation result of this method is seen in Fig. 7(b) with key=[75 170 80] for green background object and key=[130 172 80] for blue background object.

B. K-Means Clustering Method

In order to find a mask for Chroma-key method, key-value based methods cannot provide pleasing result due to the noise of the solid color in background object. A clustering method must be used to find the mask for Chroma-key method. Since the number of cluster is predefined as two, K-Means, a simple and effective clustering algorithm is an appropriate choice.

In the Matlab software, a K-Means function guides us on how to classify color clustering. Here, the image is converted from RGB to L*a*b* color space, a* and b* components are used to find the mask of foreground, and then create composite frame. Chroma-key effect performed by K-Means clustering has fairly good result, as shown in Fig. 2(c).

Kardi Teknomo proposed a modified K-Means function for the Chroma-key effect which uses some iterations before the clustering has reached stability. The simulation result on Matlab is shown in Fig. 2(d). This method has improved the performance of clustering process. However, the number of iterations in each different foreground frames is large, about eight or nine iterations in the testing foreground frames. Thus, the required buffer for the algorithm is extremely large. Moreover, the number of iterations is not constant for every different foreground frames; hence, the size of buffer needed for hardware solution is not fixed.

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 2. Simulation results on Matlab software. (a) foreground frames, (b) composite frames of the fixed key method, (c) composite frames of the method using K-Means clustering function in Matlab software, (d) composite frames of the method using K-Means clustering function modified by KardiTeknomo, (e) and (f) composite frames after coarse filter and fine filter process of the Coarse and Fine Filter method in L*a*b* color space, (g) composite frames after coarse filter and fine filter process of the Coarse and Fine Filter method in YCbCr color space. (h) Composite frame of the proposed model based on APC-OMS algorithm.

C. Coarse and Fine Filter method

The third method is Coarse and Fine filter. It involves two iterations coarse and fine filter.
The main idea of this method is to improve the K-Means function modified by Kardi Teknomo. In the proposed method, the number of iterations is fixed on two, coarse and fine filter processes, and the iteration on each frame is replaced by on each line of frame. So, the size of buffer will be decreased considerably from a frame size to a line size of a frame in hardware solution. For example, a frame with 640x480 sizes, the size of buffer is just equal 640.

In addition, the clustering just done for all pixels on each line of a frame instead of all pixels of a frame will execute more accurately. A value that uses to set the permissible minimum deviation between foreground and background object in foreground frame is applied to the coarse filter process as input value for the setting two centroids differently in the first iteration to help this filter perform better. The fine filter updates values of the two centroids and applies clustering process based on Euclidean distances from those centroids.
The simulation result of this one on Matlab software shown in Fig. 2(e)(f).

III. PROPOSED CHROMA KEY METHOD

The required buffer for the K-Means Clustering method is very large, and it is a really hard constraint for real-time hardware implementation. A requirement of decreasing the buffer size is considered. On the other hand, in the weather forecasting programs or movies that use the Chroma-key effect will have that the first line of the composite frame always includes all the pixels of background frame.

The main concept of introducing this method is to reduce the buffer size, minimize the power dissipation, to reduce the hardware complexity and to increase the speed. The proposed system introduces an adaptive filter which works on APC-OMS algorithms feeded to it. Since it is a filter based application it involves memory based computations. These memory arrays are designed in the form of look up tables.

The APC-OMS algorithm introduced here helps in designing Chroma-key effect for low area and low power dissipation. APC stands for Antisymmetric Product Coding. Here two memories are having same address and the two different memories having the same symmetric address are curtailed to one and thus the LUT size is reduced. OMS stands for Odd-Multiple storage where only odd values are stored and even values are obtained by performing shifting operation. This is shown in fig. 3 and 4. By combining together the APC and OMS approach the LUT size is reduced to 50% i.e., $1/4^{th}$ of the conventional LUT size.
The advantage of using this method is small buffer size, high accuracy in clustering, implementing Chroma-key effect in real time, low power dissipation.

APC WORDS FOR DIFFERENT INPUT VALUES FOR $L = 5$

Input, X	Product values	Input ,X	Product values	Address $x_3'x_2'x_1'x_0'$	APC words
00001	A	11111	31A	1111	15A
00010	2A	11110	30A	1110	14A
00011	3A	11101	29A	1101	13A
00100	4A	11101	28A	1100	12A
00101	5A	11100	27A	1011	11A
00110	6A	11011	26A	1010	10A
00111	7A	11010	25A	1001	9A
01000	8A	11001	24A	1000	8A
01001	9A	11000	23A	0111	7A
01010	10A	10111	22A	0110	6A
01011	11A	10110	21A	0101	5A
01100	12A	10101	20A	0100	4A
01101	13A	10100	19A	0011	3A
01110	14A	10010	18A	0010	2A
01111	15A	10001	17A	0001	A
10000	16A	10000	16A	0000	0

Fig. 3 Shows the APC algorithm

OMS-BASED DESIGN OF THE LUT OF APC WORDS FOR $L = 5$

Input X' $x_3'x_2'x_1'x_0'$	Product value	# of shifts	Shifted input, X''	Stored APC word	Address $d_3d_2d_1d_0$
0001	A	0	0001	P0=A	0000
0010	2×A	1			
0100	4×A	2			
1000	8×A	3			
0011	3A	0	0011	P1=3A	0001
0110	2×3A	1			
1100	4×3A	2			
0101	5A	0	0101	P2=5A	0010
1010	2×5A	1			
0111	7A	0	0111	P3=7A	0011
1110	2×7A	1			
1001	9A	0	1001	P4=9A	0100
1011	11A	0	1011	P5=11A	0101
1101	13A	0	1101	P6=13A	0110
1111	15A	0	1111	P7=15A	0111

Fig. 4 Shows the OMS algorithm

IV. HARDWARE IMPLEMENTATION OF PROPOSED METHOD

This section presents hardware implementation to perform the Chroma-key effect in real-time on a FPGA chip by using Adaptive Filter method.

A. VLSI Architecture

In this section we are going to see the VLSI architecture for implementing Chroma-key effect in real time.

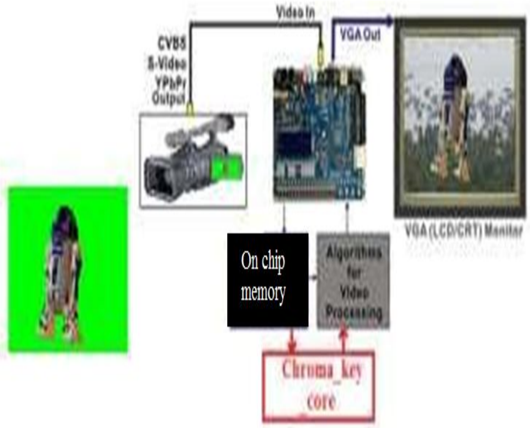


Fig. 5 A proposed system for implementing Chroma-key effect in real-time.

Fig. 5 illustrates a proposed system to perform the Chroma-key effect on the Altera DE2 board. The input video signal is retrieved from a camera, and then it is transferred to the Altera DE2 board to perform the Chroma-key effect by a Chroma-key chip. The Altera board consists of an On-chip memory and the APC-OMS algorithms installed in it which helps in processing the videos or images. After the Chroma-key process, the output video signal is displayed on a VGA (LCD/CRT) monitor.

B. Block diagram of APC-OMS algorithm

This section gives the overall block diagram of the APC-OMS algorithm . The block diagram consists of four main units the address generator, LUT, barrel shifter and the Add/Sub unit.

Here the memory address is given as input to the address generator. The address generator generates address to each memory unit feeded as input and stores the corresponding output in LUT. The LUT gives all these address bits to the barrel shifter. The barrel shifter checks for the parity. If the bit is odd it stores it as it is, and if it is even it performs the shifting operation. Now the operated bits are given to the final unit that is add/sub unit which performs the arithmetic operations addition and subtraction respectively. After the add/sub unit we finally get the desired 32 bit address as output.

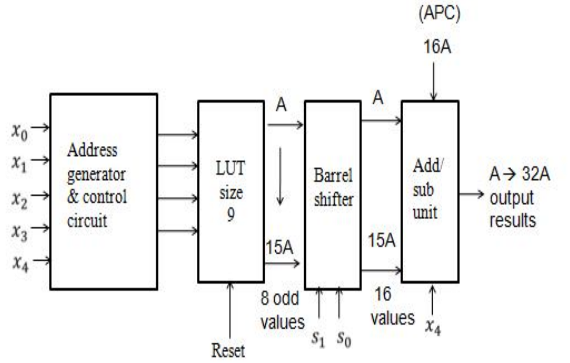


Fig. 6 Block diagram of APC-OMS algorithm

C. RTL Schematic of the overall module

The figure below shows the schematic diagram in terms of RTL for the various units or modules used in the implementation of Chroma-key effect using APC-OMS algorithm.

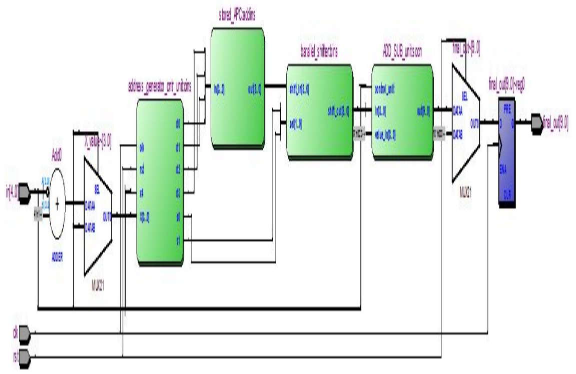


Fig. 7 RTL Schematic of APC-OMS algorithm

V. EXPERIMENTAL RESULTS

A demonstration system has been set up for testing the implementation of design in real-time on the Altera DE2 board as shown in Fig. 8.

Fig. 8 A demonstrated system for the Chroma-key effect in real-time.

This system consists of a camera used to record a video that its each frame is a foreground frame in Chroma-key process, a Altera DE2 board with the FPGA Cyclone-II chip, a monitor used to display the video result after the Chroma-key effect is implemented, a green or blue cardboard utilized to create the background object of the foreground frame, and a computer (laptop) used to download the Chroma-key program to the Altera DE2 board. Besides, a person also need to work as a foreground object of the foreground frame.

Table I shows the experimental results of various parameters for the four modules used in the APC-OMS algorithm.

TABULATION OF PARAMETERS RELATED TO FOUR MODULES

Module	Power(mW)	Frequency(MHz)	Area(LE's)
Stored APC	88.02	125.32	2
Barrel Shifter	88.20	131.12	4
Address generator	87.85	139.81	1
Add-Sub unit	88.55	145.52	1
Over APC-OMS module	87.38	215.00	34

Table I. Tabulation of parameters related to our four models

Now we have the result for simulation which was developed by running the progam codings of the various modules. The figure below shows the simulation output performed on the software MODELSIM 6.4a .here the codings are developed for the various blocks we are using in the APC-OMS algorithm proposed in this paper.

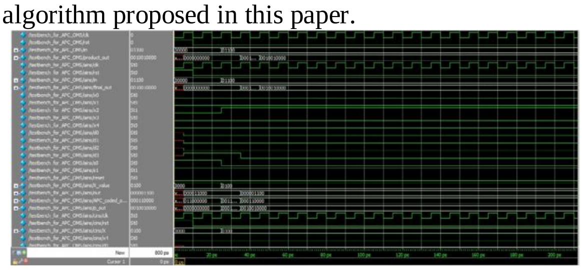


Fig 9 simulation output developed on the software MODELSIM 6.4a.

Thus we saw the experimental results developed, both on hardware as well as software which states that in general, the proposed design can perform the Chroma-key effect in real-time properly, and the experimental results are fairly good and satisfactory. They can be better and higher quality if the design is implemented in a professional studio or television system.

CONCLUSION

In this paper we proposed the adaptive filter method for studying the Chroma-key effect. We also have proposed a VLSI architectur for implementing Chroma-key effect in real time. The whole design process works mainly on two algorithms namely APC and OMS. The hardware design is developed on the Altera Cyclone II board and the programming is done using the software Modelsim 6.4a. Interfacing both the hardware and the softare together we implement the Chroma-key effect in real-time on the Altera Cyclone II board. A demonstration system is also set up to show the performance of the proposed design in real time. The experimental results show that the quality of the output composite frames by the proposed system is better than Fixed key, K-mean methods and Coarse and Fine filter.

The results of this paper are initial achievements in order to propose a Chroma-key IP core, which can be applied for the professional studios or television systems in the future.

REFERENCES

[1] Alvy Ray Smith and James F. Blinn (1996), "Blue Screen Matting", SIGGRAPH, pp. 259–268

[2] Changick Kim, Jungwoo Park, Jaeho Lee, and Jenq-Neng Hwang (2007), "Fast Extraction of Objects of Interest from Images with Low Depth of Field", ETRI Journal, 29(3)

[3] Christopher Schultz (2006), Digital Keying Methods, Technical report, University of Bremen, Germany.

[4] David Yamnitsky (2009), Real-Time Chroma Keying on the GPU, A White Paper by David Yamnitsky Boris FX.

[5] F. van den Bergh, V. Lalioti (1999), "Software Chroma Keying in an Immersive Virtual Environment", South African Computer Journal, pp. 155-162 .

[6] The IEEE paper reference number 06482383 titled "FPGA Implementationfor Real Time Chroma Key Using Coarse And Fine Filter".

[7] The IEEE reference paper titled "AN Effective Interaction Tool For Performance IN Virtual Studio".

★★★