A Video Deblurring Optimization Algorithm Based on Motion Detection

Yinan Zhang[1], Jing He[2], Jie Yuan[3]

Abstract Although the performance of image acquisition devices has been improved dramatically in recent years, especially in the resolution and clarity, defocusing and motion blur are still big problems. Upgrading the devices with the better hardware is one way to solve the problem, but the costs will usually increase disproportionately comparing with what we get. The appropriate image restoration algorithm could improve the clarity and the recognition rate of images significantly. However, the huge computation of those image restoration algorithms makes them unpractical. A new image restoration algorithm based on video target detection and an accelerating method using the Graphic Processing Units (GPU) parallel computing architecture are proposed, which makes it efficient enough to handle 720p high-definition (HD) video processing in real time and makes sure that only the interested blurred regions get restored and other parts of the image will not be impacted.

Keywords Video image restoration · GPU · Background extraction

1 Introduction

Recently many fields have raised demands for video quality, such as monitoring, traffic, security and so on. In these fields, videos often appear as the clues or the evidence. However, because image acquisition devices used are general collection equipments, the acquired video tends to appear image blur problems. For example, the object is not at the position of the focal plane or the object moves too fast and so on. When the related staff checks the video, he will find that the image of the object is not ideal. Nowadays, some professional softwares could restore degraded

[1] Y. Zhang
School of Information Science and Engineering, Jinling College, Nanjing University

[2] J. He
School of Information Science and Engineering, Jinling College, Nanjing University

[3] J. Yuan(✉)
School of Electronic Science and Engineering, Nanjing University
correspondent author e-mail: yuanjie@nju.edu.cn

© 2013. The authors - Published by Atlantis Press

images, but the softwares often require video frames with slow speed and without real-time processing. A scheme to restore blurred and degraded video image in real time is proposed, which allows people to obtain video images with relatively high quality while using general image acquisition devices. This scheme could not only save the cost of image acquisition devices, but also upgrade on the basis of the existing systems which provides good backward compatibility.

2 CUDA GPU Parallel Architecture

CUDA, compute unified device architecture, is the C language interface to program on graphics processing unit (GPU) without the knowledge of graphics application program interface (API) and GPU programming details. Run the instructions and access the storage spaces of GPU devices to benefit from GPU parallel computing. While general purpose graphics processing unit (GPGPU) programming is only for particular processors. And even if it is a standard API, it could not transplant between different GPU. In CUDA parallel architecture, GPU is used to compute as the coprocessor of CPU. Some functions in which data are parallel independent and computation is intensive are suitable to run in GPU. CUDA parallel architecture uses C language and other high-level language derived from C to develop and CUDA realizes abstraction on the specific hardware to make the same program run on different GPU. CUDA parallel computing architecture is a cross-platform which can be realized on many operating systems like Windows/Linux and so on, and it is a full set of development environment with the software and hardware. Because of its universal development method and powerful parallel ability, CUDA parallel computing architecture has received extensive attention, widely used in the scientific computing filed, such as biologic science, multimedia codecs, image processing and so on[1-4].

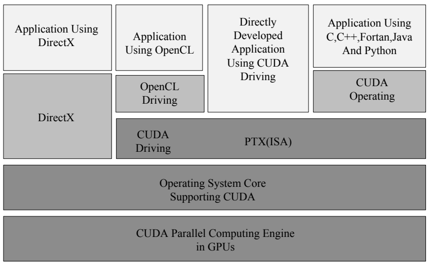
Application Using DirectX
Application Using OpenCL
Directly Developed Application Using CUDA Driving
Application Using C,C++,Fortan,Java And Python
OpenCL Driving
CUDA Operating
DirectX
CUDA Driving
PTX(ISA)
Operating System Core Supporting CUDA
CUDA Parallel Computing Engine in GPUs

Fig. 1 Composition of CUDA parallel architecture

CUDA parallel architecture is shown in Fig. 1. Components of CUDA are on the bottom including GPU supporting CUDA parallel computing, the operating system core supporting CUDA, the user mode driver and PTX instruction set for parallel computing. At the software level, CUDA provides complete software development environment including the compiler, the debugger, the performance assessment tools and the basic libraries. Support standard C language programming, which could support other high-level languages like Fortran, Java and Python. Two kinds of programming interfaces provided by CUDA are device-level programming interface and language integrated programming interface[5].

3 Image Restoration Method

The image restoration method studies how to improve given degraded images. This method tries to decrease or remove image blur and noise interference using prior knowledge of degradation to rebuild or restore the degraded image. First model image blur, then use reverse process to restore the original image[6].

Common linear image restoration methods include direct inverse filtering, constrained least-squares filtering[7] and wiener filter realized by CUDA parallel architecture in this paper[8-9].Wiener filtering was first proposed by N.Wiener in 1942, which is a stationary linear filter of least mean square error to be used to deal with the image degraded by the additive noise and the fuzzy function.

The wiener filter could get the least mean square error of the restoration image and the original image. However, the power spectrum functions of the original image and the noise should be got while it's difficult to get these two parameters. In this paper, the power spectrum of the autocorrelation approximate image is used, then assume that the noise in the image is white noise. Therefore, the power spectrum function of the noise is a constant value which is the result of the noise variance multiplied by the size of the image. The noise variance could be obtained by the variance of a flat zone in the image acquisition process.

4 Ring Effect and Elimination

When the image restoration algorithm is used to restore the degraded image, generally the restored image has some stripped noises called ring effect. There are two reasons. First, frequency operation is carried out on the images. In fact, Discrete-time Fourier transform (DTFT) is used. While the computation of DTFT is completed by discrete Fourier transform (DFT). Converted data are regarded as the period truncation of continuous periodic signals by DFT. For the image, it is equivalent to two-dimensional extension copy of the image. While in fact, the image is not periodic and ring effect is generated. Second, the restoration filter is used to restore the full image, but not all parts of the image have been degraded. For example, in the image, an object generates motion blur because of movement, while the background is clear. If motion blur is also carried out on the background, the noise is generated in the direction of motion. The methods to eliminate ring effect are cyclic boundary method and optimized window method.

The core thought of cyclic boundary method is to change the non-cyclic image to the cyclic image. Since one of the reasons of ring effect is that DTFT is calculated by DFT and DFT assumes that calculated signal is periodic, if the image is changed into the periodic image, ring effect will be eliminated. When the image is mirrored at up-down and left-right directions, the image is changed into the periodic image. The period of mirrored image is extended to be continuous, while the calculation amount is increased four times. This method is only suitable for the occasions which do not need high calculation speed.

DFT is carried out on the image directly which is equivalent to add the rectangular window on the signal. From the knowledge of digital signal processing, we know that adding the rectangular window on the signal will extend high frequency component of the signal, which is the reason of ring effect. Therefore, if a non-rectangular smooth window is added on the image, high frequency extension will be smaller. The optimized window method is to add a non-rectangular window on the image to reduce ring effect. Although above cyclic boundary method could solve the problem of ring effect, the computation amount is large. Therefore, the optimized window method is proposed by Limetal. First, the optimized window method divides the image to nine areas, shown in Fig. 2.

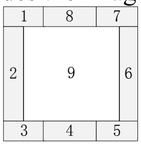

Fig. 2 Image area of optimized window method

In these nine areas, there are one in the ninth area which means original image data are fully used. The areas of 1 to 8 are determined by following equations.

$$\begin{aligned}
&H_{1,8,7} \in [0, H_{psf} - 2] \\
&H_{2,6} \in [(H_{psf} - 1), (H_{img} - H_{psf})] \\
&H_{3,4,5} \in [(H_{img} - H_{psf} + 1), (H_{img} - 1)] \\
&V_{1,2,3} \in [0, V_{psf} - 2] \\
&V_{4,8} \in [(V_{psf} - 1), (V_{img} - V_{psf})] \\
&V_{5,6,7} \in [(V_{img} - V_{psf} + 1), (V_{img} - 1)]
\end{aligned} \qquad (1)$$

H and V mean the ranges of horizontal and vertical coordinates. The subscript number means the image area. The subscript *psf* means the size of the point spread function. The subscript *img* means the size of the original image.

The computational formula of the optimized window is shown as follows.

$$\begin{bmatrix}
\sum_{p=0}^{i}\sum_{q=0}^{k} h(p,q) & \sum_{p=0}^{i}\sum_{q=0}^{H_{psf}-1} h(p,q) & \sum_{p=0}^{i}\sum_{q=k+H_{psf}-H_{img}}^{H_{psf}-1} h(p,q) \\
\sum_{p=0}^{V_{psf}-1}\sum_{q=0}^{k} h(p,q) & 1 & \sum_{p=0}^{V_{psf}-1}\sum_{q=k+H_{psf}-H_{img}}^{H_{psf}-1} h(p,q) \\
\sum_{p=i+V_{psf}-V_{img}}^{V_{psf}-1}\sum_{q=0}^{k} h(p,q) & \sum_{p=i+V_{psf}-V_{img}}^{V_{psf}-1}\sum_{q=0}^{H_{psf}-1} h(p,q) & \sum_{p=i+V_{psf}-V_{img}}^{V_{psf}-1}\sum_{q=k+H_{psf}-H_{img}}^{H_{psf}-1} h(p,q)
\end{bmatrix} \qquad (2)$$

i and *k* are the positions of computation and *h( )* is the point spread function.

5 Motion Detection Method

In general video systems, the backgrounds are often clear, while the image quality of videos is disturbed by the foreground because of motion blur and so on. If the background and the foreground are uniformly processed, although objects in the foreground may be restored, generally other clear parts will be affected. Therefore, the video image restoration algorithm is proposed in which the motion tracking algorithm is used to realize separation of the foreground and the background, then the foreground is processed and the background is kept.

General methods of motion detection[10-11] include inter-frame difference method, optical flow method, background extraction method[12] and so on. In the background extraction method, first the background is extracted, then calculate the difference between the current frame image and the background image. The key of this method is the background extraction algorithm. Common background extraction methods include Surendra algorithm[13-14], multi-frame mean method, median method, Gaussian mixed model method and so on. In this paper, Surendra algorithm is used.

In general methods of motion detection, the detection results of the motion objects are binary images, but the results after binaryzation have noises or the effect is not ideal. To remove noises and achieve the ideal effect, morphological processing should be carried out on these binary images. That is to say, mathematical morphology is used to extract some image contents which have certain effect to express and describe the zone shape. Morphological processing methods used in this paper are expansion and corrosion.

6 Implementation and Performance Test

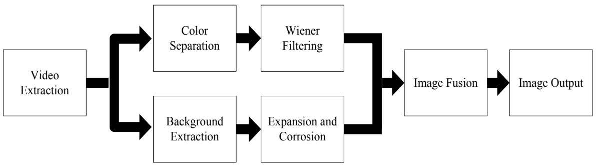

Fig. 3 Whole structure diagram of program

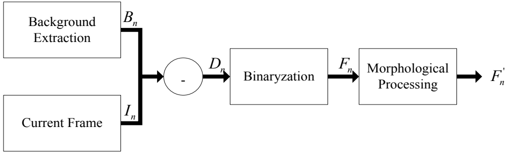

Fig. 4 Detection step of moving object

The whole structure diagram of program is shown in Fig. 3. Open Computer Vision (OpenCV) is used to read and output videos. The background extraction method is used to realize motion object detection, shown in Fig. 4.

To test the algorithm proposed in this paper, the videos with defocus and image degradation generated by motion blur are collected and the algorithm is applied. Video screenshot with image blur because of defocus is at the left of Fig. 5.

Restoration video screenshot used with the proposed algorithm is at the right of Fig. 5. In above two figures of Fig. 5, the camera sets focus on the book and the lens cap in the hand is not at the focal plane. Thus words on the lens cap in the left screenshot are very vague. After processing, words on the lens cap become clear and the object at the focal plane does not degrade with image restoration of lens cap. So do following two figures of Fig. 5. The cup at the focal plane does not degrade with restoration of books. Fig. 6 shows the screenshots of motion blur video and the restoration video. On the left, words on the handbag are unclear because of movement. After restoration, words on the handbag become recognizable. The image below Fig. 6 shows amplification of the handbag.

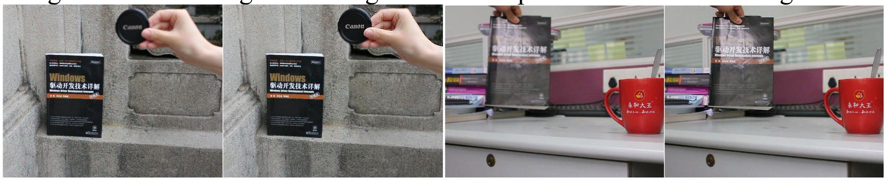

Fig. 5 Defocused video screenshot (Left) and restoration video screenshot (Right)

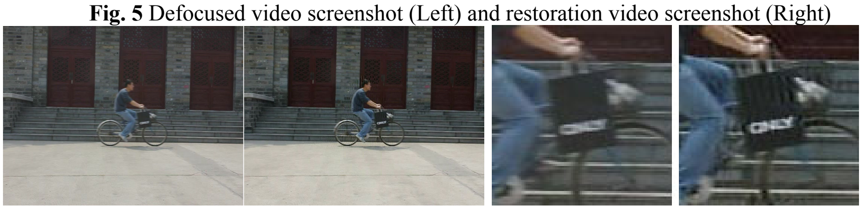

Fig. 6 Motion blur video screenshot (Left) and restoration video screenshot (Right)

Proposed algorithm in this paper could be used to process videos with different resolutions. The average time of every frame is shown in Table 1. For standard definition (SD) 480p and high definition (HD) 720p, NVIDIA GeForce GTX 460 GPU could be used to do real-time processing. Fig. 7 uses the curve to describe contents of Table 1 which shows great performance improvement brought by CUDA more intuitively. In Fig. 7, the horizontal coordinate is the image size and the vertical coordinate is processing time. The red star line is the processing speed of CPU. The blue block line is accelerated processing speed of NVS 300. The green cross line is accelerated processing speed of GTX 460. From this figure, with the improvement of resolution, the computation time by using CPU rises quickly while the computation time by using GPU parallel computing rises slightly. In this paper, two kinds of GPU are tested which are NVS 300 and GeForce GTX 460. NVS 300 has 64-bit bandwidth, 16 CUDA processor units, and the memory bandwidth is 12.6G/s. GeForce GTX 460 has 256-bit bandwidth, 336 CUDA processor units, and the memory bandwidth is 108.8G/s. For 1080p HD video, GeForce GTX 460 could only achieve the processing speed which is 16 frames per second. But the performance of this kind of GPU is ordinary with low price which is 700 yuan, fully accepted by general public and 720p real-time processing also meets general requirements. If we pursuit for real-time processing of 1080p HD videos, NVIDIA professional computing GPU Tesla series could be used to achieve faster computing speed. For example, Tesla C2075 processor has

384-bit bandwidth, 448 CUDA processor units, and the memory bandwidth is 144G/s. Moreover, it is dedicated for structures designed for computing, making performance more stable with better speed. Table 2 shows the performance difference of these types of GPU.

Table 1 Execution Time of Every Frame in Video Restoration Algorithm (ms)

Video Resolution Processor Type	320x240	640x480	1280x720	1920x1080
Intel CPU P8700	246	1529	4230	11148
NVS 300	35	97	240	500
GeForce GTX 460	≈0	16	31	62

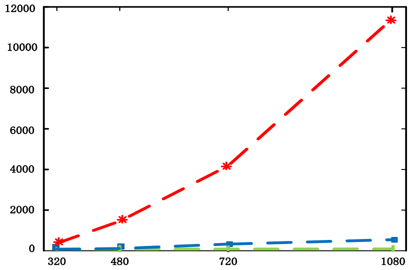

Fig. 7 Testing Speed of Different GPU and CPU

Table 2 Performance Contrast of Different GPU Processors

GPU Processor	Bit Wide（bit）	CUDA Processor Unit	Memory Speed(GB/s)
NVS 300	64	16	12.6
GeForce GTX 460	256	336	108.8
Tesla C2075	384	448	144

7 Conclusion

In video collection, because of image acquisition devices or environmental reasons, image degradation may occur like defocus and motion blur and so on. There is a solution to improve the performance of acquisition devices, but the cost is too high. In this paper, motion detection is used to detect the moving object and image restoration is carried out on the moving part of the image. This method not only ensures not to interfere with the clear part of videos, but also restores the fuzzy part of the image. According to the test, the algorithm which needs several seconds in CPU realized real-time processing of 720p HD video. Although this algorithm could realize real-time restoration of HD video, it still has some problems. First, robustness of the algorithm is not strong. Good result could not be restored in some complicated situations. This is not only because of wiener filtering characteristic, but also because of motion detection algorithm. Second, restoration parameters of the algorithm should be provided in advance. For defocus restoration, it is not a big problem in the case of fixed image acquisition devices. But for motion blur restoration, fixed restoration parameters do not meet the needs. Therefore, problems such as how to improve robustness of the algorithm and how to change restoration parameters dynamically need to be further studied in the future.

Acknowledgments This study is funded by National Natural Science Foundation of China (Grant Number: 61201425).

References

1. Ching-Lung Su, Po-Yu Chen, et al. Overview and Comparison of OpenGL and CUDA Technology for GPGPU[C]. IEEE Conference Publications: Circuits and Systems (APCCAS). pp. 448-451, (2012)
2. Al Mouhamed M., ul Hassan Khan A.. Exploration of Automatic Optimization for CUDA Programming[C]. IEEE Conference Publications: Parallel Distributed and Grid Computing (PDGC). pp. 55-60, (2012)
3. Garland M.. Parallel Computing with CUDA[C]. IEEE Conference Publications: Parallel & Distributed Processing (IPDPS). pp.1, (2010)
4. Yinghong Sun, Yuanman Tong. CUDA Based Fast Implementation of Very Large Matrix Computation[J]. IEEE Conference Publications: Parallel and Distributed Computing, Applications and Technologies (PDCAT). pp. 487-491, (2010)
5. N.P. Karunadasa, D.N. Ranasinghe. Accelerating High Performance Applications with CUDA and MPI[C]. International Conference on Industrial and Information Systems, pp. 331-336, (2009)
6. Aftab Khan, Hujun Yin. Quality Measures for Blind Image Restoration[C]. IEEE Conference Publications. pp. 456-459, (2012)
7. Yongning Li. Comparison between Several Constrained Least Squares Methods in Image Restoration[J]. Journal of Qinghai Normal University(Natural Science Edition), pp. 30-34, (2010)
8. Ho-Gun Ha, In-Su Jang, et al. Subpixel Shift Estimation in Noisy Image Using a Wiener Filtered Local Region[J]. IEEE Journals & Magazines, pp. 57(2): 623-630, (2011)
9. Xu Yang, Yilin Chang, et al. Adaptive Wiener Filter Based Chrominance Up-sampling Enhancement Method for Video Coding[J]. IEEE Journals & Magazines, pp. 57(4): 1851-1856, (2011)
10. Li Fu, Shuai Fang, Xinhe Xu. Computer Vision Method in Human Motion Detection[J]. Journal of China ordnance. pp. 3(1): 54-58, (2007)
11. Charley Paulus, Yujin Zhang. Spatially Adaptive Subsampling for Motion Detection[J]. Tsinghua science and technology. pp. 14(4): 423-433, (2009)
12. Ting Zhang, Xiangdong Zhao, et al. An Improved Real-time Background Extraction Algorithm Based on Jung's Method[J]. Microcomputer & Its Applications, pp. 31(9): 41-43, (2012)
13. Surendra Gupte, et al. Detection and Classification of Vehicle[J]. IEEE Transactions on Intelligent Transportation System, pp. 3(1): 37-47, (2002)
14. Zhiqi Yang. A New Algorithm of Background Image Extraction and Update in the Vehicle Detection System[J]. IEEE conference publications. pp. 5238-5240, (2011)