Real-time Algorithm for SIFT Based on Distributed Shared Memory Architecture with Homogeneous Multi-core DSP

Liu Xin , Chen Wenjie, Ma Tao, Xu Lishuang

***Abstract*—When Multi-DSP parallel architecture transfers to distributed memory way from shared memory way, its parallelism with fine-grained become weak,and it`s difficult to offer SIFT`s complex computing and satisfy the need of real-time. In the paper, a parallel algorithm, based on distributed shared memory architecture with homogeneous Multi-core DSP, referring to the DSM architecture model of parallel processing machines is presented. Firstly, the master processor separates the task into several small tasks by exploiting the coarse-grained parallelism inherent; then, through a high-speed network for data-exchange, each small task transfers to a related subsystem, based on a homogeneous multi-core DSP; finally, the DSP partitions the small task across multiple cores by exploiting the fined-grained parallelism. The experimental result shows that, comparing to the traditional way, the proposed algorithm increases the speedup, calculates 45 frames on 640x480 images, and achieves the real-time application.**

I. Introduction

SIFT (Scale Invariant Feature Transform), the feature matching algorithm, is a domestic and international hot point recently in matching image research realm. It owes a robust matching ability, and is applicable to shift, revolve, even affine etc, taken place between two pictures. But its complex calculation makes it difficult to carry out video panorama synthesize in real-time circumstances actually, and then it need to be circulated through the high speed processor.

Due to the high digital signal processing speed, DSP has been widely used in video image processing area. Nowadays, video processing system tends to be designed into multi-DSP parallel structure, so that it could bear increasing amount of calculation.

Although the research for the SIFT parallel-algorithm have been tested in parallel machines, but the system of multi-DSP processing based on traditional single-core DSP, will cost much more time in the communication between the DSPs. In order to reduce the time cost caused by frequent communications, we need to decompose the original task in coarse-grained method to reduce the degree of parallelism, and increase DSP nodes to accelerate computation to achieve the ability of real-time.

With homogeneous multi-core DSP coming out in recent years, the communication among DSPs has been replaced by faster communication inter cores. This improvement has provide sufficient hardware convenience for introducing the SIFT parallel algorithm to DSP in practical application. This paper focuses mainly on Distributed Shared Memory (DSM) parallel computing model, which is introduced to digital video processing system based on multi-DSP with homogeneous multi-cores DSP. Besides, we have improved the SIFT parallel algorithm to reduce calculation time and realize real-time video panorama synthesis.

II. Design of Multi-DSP Processing System for SIFT Parallel Algorithm

A. Problems of Multi-DSP Parallel Structure

Generally, parallel architecture of multi-DSP can be classified into two classes, which are shared bus and point-to-point connection[1]. Moreover, the parallel processing system of multi-DSP can be regarded as a small parallel machine, and comparing to the architecture model of the parallel machine, it can be also classified into two categories: shared memory and distributed memory.

Usually, in the shared memory structure, all external buses (the address, data and interview control bus) of total DSPs are linked together directly, and each memory internal chip, IOP registers, and external memory address and peripherals both hanging to bus, could all be interviewed by each DSP. It is also a kind of shared memory structure, as figure 1(a) shows. Because all address spaces is mapping, and the data keep on renewing facing to all processors, it supplies a hardware condition for handled the depth in the task to scoop out fine-grained in video processing.

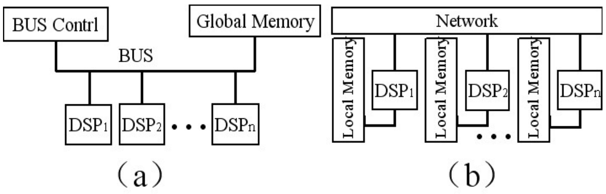

Fig.1. Shared and distributed memory multi-DSP structure

Manuscript received May 10, 2011.

Liu Xin is with School of Automation, Beijing Institute of Technology and Key Laboratory of Complex Systems Intelligent Control and Decision, Ministry of Education, Beijing, 100081China(e-mail: liuxin0123@ foxmail.com).
Chen Wenjie is with School of Automation, Beijing Institute of Technology and Key Laboratory of Complex Systems Intelligent Control and Decision, Ministry of Education, Beijing, 100081China(e-mail: chenwenjie@bit.edu.cn).
Ma Tao is with School of Automation, Beijing Institute of Technology and Key Laboratory of Complex Systems Intelligent Control and Decision, Ministry of Education, Beijing, 100081 China (corresponding author to provide phone: 86-10-68948971; fax: 86-10-68918232; e-mail: matao.bit@gamil.com)
Xu Lishuang is with School of Automation, Beijing Institute of Technology and Key Laboratory of Complex Systems Intelligent Control and Decision, Ministry of Education, Beijing, 100081China(e-mail: yq_xls@126.com).

978-1-4577-0816-9/11/$26.00 ©2011 IEEE

The drawback of shared memory structure is that it depends overly on the bus and produces waiting list. It cannot be applied in frequently data exchange occasions, with a poor expand performance. When DSPs increase, one DSP accesses memory resources through bus at a certain moment, the other DSPs have to wait, emerging the limitation of shared memory structure.

As figure 1(b) shows, in distributed memory structure, each DSP has its own ID and point-to-point communicates with the other DSPs in the net. The pretty expand performance is the most advantage of distribute memory structure, and the system scale is free from restriction in principle. The network topology can be designed vividly as line type, star type, annulus type, and tree type, or super-cube type etc.

Because of no unified memory map addresses between DSPs and no correlation between instructions and low coupling effect, it is not suitable for development of fine-grained communication with the long communication time.

B. Design of DSM Structure

As figure 2 shows, Symmetric Multi-Processing (SMP) and Massively Parallel Processing (MPP) are two kinds of structures belonging to massively parallel machine structure.

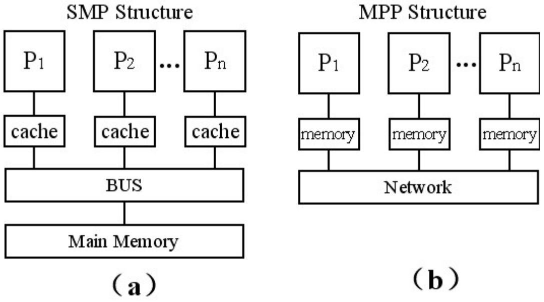

Fig.2. SMP and MPP structure

Figure 2(a) depicts the structure of SMP. The processors of SMP are connected with shared memory，and each processor has the same opportunities to interview shared memory, I/O devices, and the OS server, as run in shared memory multi-DSP system. Meanwhile, the SMP also has a poor expansibility.

Figure 2(b) depicts the structure of MPP. As the distributed memory multi-DSP system, MPP is constructed by many loose coupling processing unit, and CPU in each unit has its own private resources, such as bus, memory, hard drive and so on. Besides, the structure of MPP can be extended unlimitedly in principle. Similarly, when communication is frequent, MPP must transmit information in different units. As a result, the calculation efficiency is lower than SMP. Hence, SMP is not suitable for mining fine-grained in parallel algorithm.

The integrated model of combining the SMP with the MPP, called distributed shared memory (DSM), is adopted in massively parallel machine, which can maintain the advantages of the architecture of the SMP system, as well as can alleviate the non extendable problems of system scale happened in traditional SMP system. As figure 3 shows.

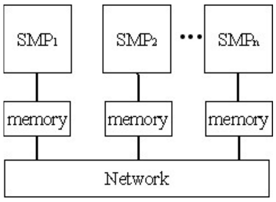

Fig.3. DSM structure

On one hand DSM cluster operates as MPP, but on the other hand local memory forms a shared memory space in each processing nodes. For users, the system hardware and software provides a single address programming space. Such design not only increases the parallel efficiency of multi-processors node, maintaining the SMP's advantages, but also unites the advantages of MPP, what makes the system expands scale to hundreds of thousands of processors. The concept of homogeneous multi-core DSP means to integrate several similar DSP cores in an IC. As figure 4(a) shows a three-core DSP TMS320C6474, the product of Texas Instruments.

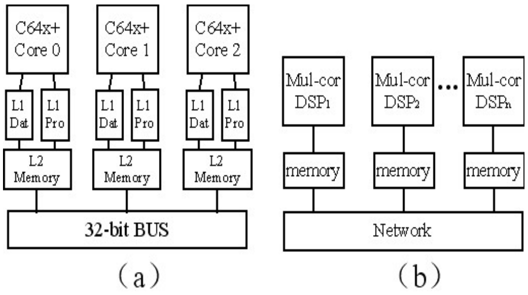

Fig.4. C6474 core and DSM multi-DSP structure

Compared with the SMP structure, homogeneous multi-core DSP is not only similar with SMP structure, but also fully in accordance with the SMP mode. In fact, the emergence of homogeneous multi-core DSP is not an accident, but a penetration of the massively parallel machine concept of parallel processing performance to the embedded system. The introduction of multi-core processor technology is a valid method of improving the parallel performance of the processor. The actual performance of the processor is measured by the total amount of instructions which the processor can handle per clock cycle; thereby the number of processor executable instructions per clock cycle will double by increasing a core.

Modeled on the DSM program of massively parallel machine， we embed homogeneous multi-core DSP in distributed memory multi-DSP structure as a separate processing node, to constitute DSM multi-DSP structure, as shown in Figure 4(b) above.

It can work in accordance with DSM mode, but also have the ability to mining fine-grained deeply from shared memory structure and the excellent expansibility from distributed memory structure.

But we need to consider programming problems of these two structures, in order to maintain memory consistency, try to avoid communication between the different DSP nodes; and we also need to consider how to divide parallelism between the DSP interior and different DSP nodes.

III. Design of SIFT Parallel Algorithm Design

The SIFT algorithm is divided into 4 parts:
(1)Detect extreme in scale space;
(2)Accurate stable keypoint localization;
(3)Assign orientation;
(4)Calculate keypoint descriptor[2].
As figure 5 shows the SIFT original algorithm flow.

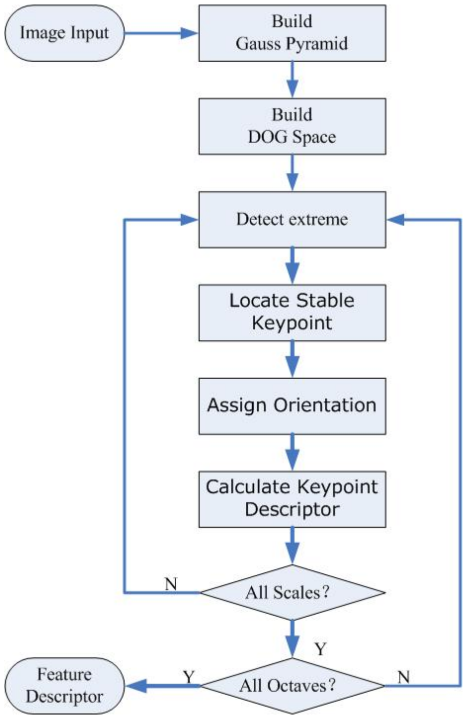

Fig.5. SIFT original algorithm flow

As the same as parallel machine, the design of SIFT parallel algorithm based on Multi-DSP System utilizes Task/Channel Model. In the model, video and image processing is divided into a series of sub-tasks, which send messages to each DSP by channels, then each DSP interacts with others, and the parallel computing is done. According to four step of parallel algorithm design process-dividing, communication, gathering and mapping- proposed by Ian Foster, the SIFT parallel algorithm design process of video processing system based on multi-DSP SIFT is improved as follows:

A. *Coarse-grained*

(1)Coarse-grained division: Extract modules which cost most time in the original algorithm, and divide them into sub-tasks as number of DSPs as possible.

(2)Coarse-grained communication: Definite the communicating mode between sub-tasks.

(3)Coarse-grained gathering: Combines the linked sub-tasks as many as possible.

(4)Coarse-grained mapping: Assign tasks to DSP nodes, and avoid communication between DSP nodes to maximize utilization of each DSP node.

Responding to the analysis from the original SIFT algorithm, the time-consuming module is down-sampling when to build Gauss pyramid, detecting extreme in scale space and calculating keypoint descriptor.

When building a gaussian pyramid, the image at the first level of octave 2 can be obtained by down-sampling the image at middle level of octave 1 with a scale factor of $k^2\sigma$, and the image at the first level of octave 3 can be obtained by down-sampling the image at middle level of octave 2. A gaussian pyramid conventionally contains 4 octaves, 5 levels for each octave, and besides, Gaussian filter is a typical operation which contains only multiplication and addition, so the construction of gaussian pyramid should be carried out sequentially to compress the calculation and maximum the utilization of DSP.

When building Gaussian pyramid, if other DSP cores are to detect extreme and calculate keypoint descriptor in scale space built well, the algorithm can save a lot of time.

As shown in Figure 6, in scale space, the down-sampling ratio is 1: 4: 16: 64. We can divide scale space into a series of data blocks of approximately equal size as ratio, and assign them to DSP nodes through high speed communication interface. Although assigning data blocks is a serial algorithm, it can reduce the parallel time less than the serial one, and make the serial one as bottleneck of the whole algorithm.

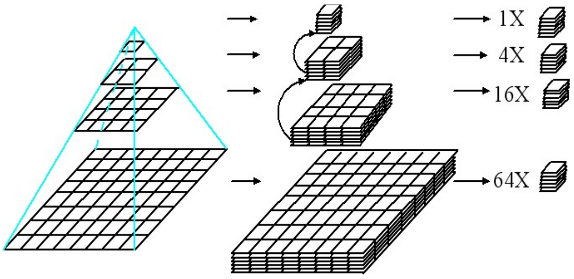

Fig.6. Data blocks division

B. *Fined-grained*

Because under the fine-grained parallel computing, the original task can be divided into section programs as tiny as possible to maximum parallelism. However, the maximum parallelism means the biggest overhead at the same time.

R represents the time for task calculating, C denotes the extra time consumption for task communicating and so on. Fine-grained need more communication frequency, as a result, the value of R/C is smaller due to the long time of communication. Though higher parallelism can be obtained, it will fail to make the system finish the calculation faster, and the system optimum performance is not obtained.

The solution is that, shorten correspondence distance and make calculation time to over correspondence time as far as possible. In unify memory space, each instruction could include parallel information abundantly, and map whole memory space into a big processing node. When a DSP will write a result into memory, another DSP can interview it directly. In this process, the handshake and correspondence between DSPs are more convenience than distributed memory structure, so the sub-tasks fine-grained algorithm flow chart is modified as follows:

(5)Fine-grained division: Divide the sub-tasks into small segment programs to acquire maximum parallelism, and extract time-consuming module.

(6)Fine-grained communication: Definite the communicating mode between small segment programs.

(7)Fine-grained gathering: Combines the small segment programs, which communicate frequently with each other, as many as possible.

(8)Fine-grained mapping: Assign small segment programs to DSP cores, and enrich assembly line. Cover communication time inter-core with computing time.

In the original SIFT algorithm flow process chart, the detection of extreme of next scale space will not start until the keypoint descriptor of previous one is generated, which is a huge waste of time. A modified flow process can be seen in figure 7. When a DSP core has located a stable feature point of a data block, it saves the feature point to the stack and then continues to detect the extreme point of next data block. While another DSP core loads the static feature point instantly, then determines the direction, and generates a key descriptor. Thus a piece of shared memory of two DSP cores is needed, which provides a faster inter-core communication.

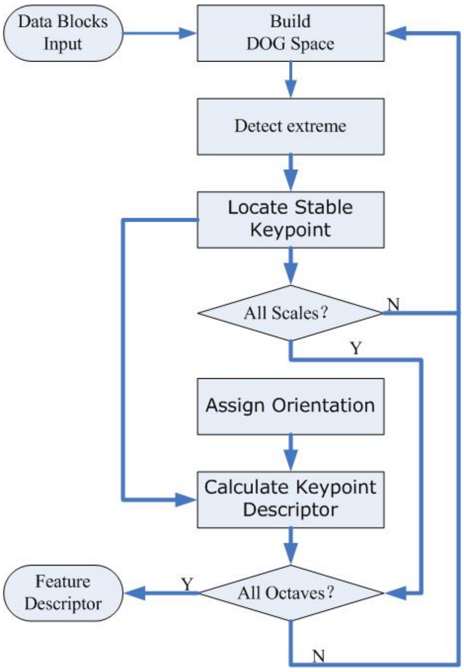

Fig.7. The modified algorithm flow

IV. Experiment Result

Two video processing systems are adopted, one is tri-core DSP TMS320C6474 based dual DSP hardware platform, of which the two DSPs communicate through high speed data exchange interface SRIO; and another is single-core DSP TMS320C6455 based four DSP process system, of which the four DSPs can not only communicate through SRIO(distributed memory structure) , but also share EMIF bus(shared memory structure). C6474 has a same DSP core C64X+ with C6455, and its frequency can reach 1GHz. The data to be processed is a piece of 45fps@640X480 video output by industry color digital camera.

Table.1. the performance of different algorithms

	Core Num	Freq	640X480	Real-Time
Original Algorithm	1	1GHz	12fps	NO
Parallel Algorithm	4	1GHz	45fps	YES

Table 1 experiment result express, compare original calculate way, SIFT after improving proceed together the calculate way calculation efficiency has obvious advantage, can satisfy solid sex request.

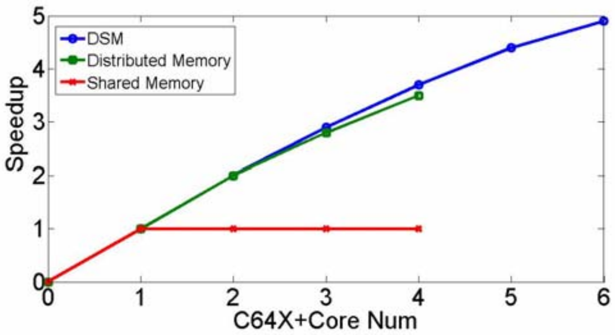

Fig.8. Speedup of different structure

From figure 8, the speedup of DSM increases faster with the increase of core number than distributing structure. However, if the core number continues growing, the accelerating ratio will fix to a stable value, and the difference between the two accelerating ratios of these two structures will be stable. Additionally, with a same increase in DSP number, the benefit of distributing structure is far below that of distributing sharing structure. Figure 8 also shows that the accelerating ratio of sharing structure is always tended to be 1. Because the time that to be processed nodes spent in waiting for free bus is a waste in the shared structure, so the best case is that only one DSP works and uses the bus solely.

V. Conclusion

During the parallel development of the DSM with homogeneous multi-core DSP, the real-time in practice is greatly improved. However, there are still a number of node waiting condition during the process of the implement. Therefore the application of some new parallel algorithm is required in the later researches to optimize assignment of caculation mostly and to improve the computing speed.

Reference

[1] W. Wei, et al., "Design methods of multi-DSP parallel processing system," 2009 WRI World Congress on Computer Science and Information Engineering (CSIE 2009), vol. vol.3, pp. 458-64, 2009.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, pp. 91-110, 2004.