Performance Analysis of Parallel Video Feature Extractions on Multi-Core Systems

Yurong Chen, Eric Li, Jianguo Li, Qi Zhang, Lei Hou and Yimin Zhang

Intel China Research Center,
8/F, Raycom Infotech Park A, NO.2, KeXueYuan South Road, Zhong Guan Cun,
HaiDian District, Beijing 100080, China,
yurong.chen@intel.com

Extended Abstract

With the explosive increase in video data, automatic video management (search/retrieval) is becoming a mass market application, and Content-Based Video Information Retrieval (CBVIR) is one of the best solutions. Most CBVIR systems are based on low-level feature extractions guided by the MPEG-7 standard for high-level semantic concept indexing. It is well known that CBVIR is a very compute-intensive task, and the low-level visual feature extractions are the most time-consuming components in CBVIR. Nowadays, with the multi-core processor becoming mainstream, CBVIR can be accelerated by fully utilizing the computing power of available multi-core processors.

A set of representative visual feature extractions have been optimized and parallelized to accelerate CBVIR on multi-core systems in our previous study. In this paper, we conduct a detailed performance analysis of these parallel applications using three video inputs with different format, MPEG1, MPEG2 and HDTV on a dual-socket, quad-core system. The analysis helps us identify possible causes of bottlenecks, and we suggest avenues for scalability improvement to make those applications more powerful in real-time performance.

Six parallel video feature extraction applications are studied in this paper. These applications are used to parallel extract important visual features such as the banded color correlogram, multi-resolution simultaneous autoregressive models (MRSAR, a kind of texture feature), Gabor textures, shape context, SIFT (scale-invariant feature transform) features and the optical flow feature in video on shared-memory multiprocessor and multi-core systems. They are highly optimized and parallelized to harness the computing power of multi-core systems.

We evaluate the performance of these applications on an 8-core system, which is a dual-socket, quad-core machine, with two Intel Core 2 quad processors running at 2.33GHz. Each socket has four cores, and each core is equipped with a 32KB L1 data cache and a 32KB L1 instruction cache. The two cores on one chip share a 4MB L2 unified cache. The maximum FSB bandwidth is 21GB/s. For software configuration, on both platforms, we use Intel C/C++ Compiler Version 9.1 to compile the six programs under Linux Kernel 2.6.22smp with full compiler optimization.

In this study, 50 video key frames are first selected from HDTV movies as the HDTV video input. Then we produce the MPEG1 and MPEG2 video inputs by

downsampling these 50 key frames from the HDTV format (1920x1080 pixels) to the MPEG1 (352x240 pixels) and MPEG2 (720x480 pixels) formats, respectively. As a result, these three video inputs have almost the same visual content just with different resolution.

As shown in Figure 1, most applications are far slower than real-time, i.e., 30 frames per second (FPS), in the serial performance on an 8-core system. In particular, the speed of four applications is more than 20x slower than real-time when using the HDTV video input. The result reveals these applications are really compute-intensive components in CBVIR which should be accelerated by parallel processing on multi-core systems.

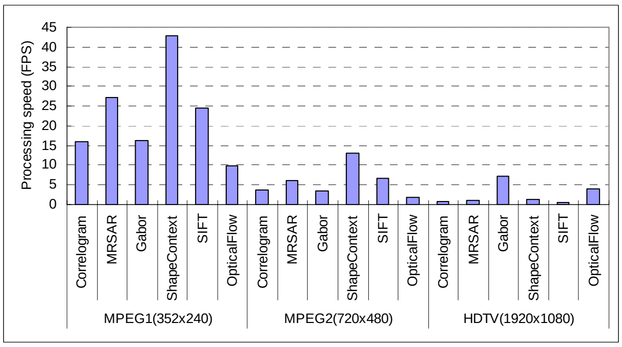

Fig. 1. Sequential processing speed of the six video feature extraction applications

These parallel applications scale very well as the number of threads increases, as shown in Figure 2. Four of them exhibit almost linear speedups and one (SIFT) achieve quite respectable speedup with different video inputs. Only Gabor's scaling performance incrementally becomes worse when the resolution of the video input increases. That is, most of these CBVIR workloads can efficiently use the computational power provided by multi-core processors.

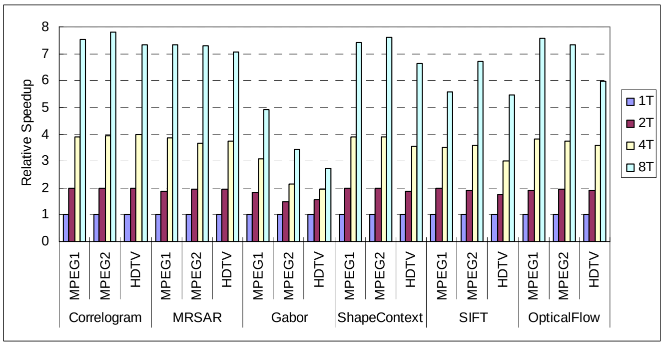

Fig. 2. Scalability of the six video feature extraction applications on an 8-core system

To fully understand the scaling limiting factors on an 8-core system, we characterize the parallel performance from the high-level general parallel overheads, e.g., synchronization penalties, load imbalance, and sequential regions, to the detailed memory hierarchy behavior, e.g., cache miss rates and FSB bandwidth.

We profile them with the Intel Thread Profiler to see their general parallel limiting factors. From Figure 3, we can see that the parallel region dominates in the execution time breakdown even for the two applications with correspondingly poor scaling performance, which suggests these CBVIR workloads expose good parallel performance metrics. Figure 3 also shows that, some workloads, especially SIFT, suffer a lot from load imbalance when the number of threads increases to four and eight, which leads to the poor speedup of SIFT. If we assume the parallel region can scale perfectly, Gabor and SIFT should achieve theoretical speedups of 7.8 and 6.7 at least, respectively, on eight cores. The theoretical speedups are much higher than the practical results shown in Figure 2. Therefore, we believe the scalability of our workloads is also limited by some other factors.

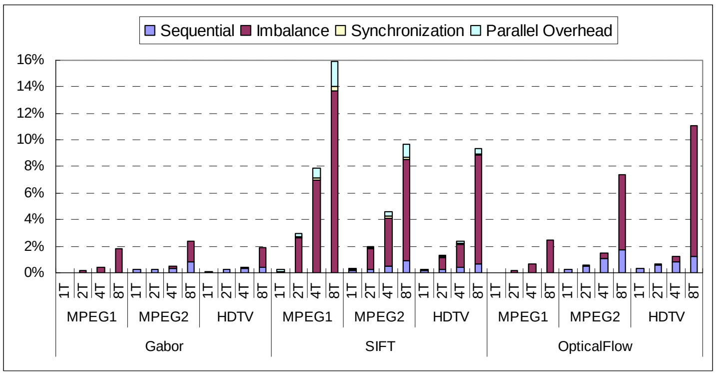

Fig. 3. Percentage of sequential regions, imbalance, synchronization and parallel overhead in three parallel video feature extraction applications on an 8-core system

Besides the general scalability performance factors, the memory subsystem also plays an important role in identifying the scaling performance bottlenecks. For further assurance, we get the memory-hierarchy micro-architectural statistics and memory bandwidth utilization with the Intel VTune Performance Analyzer. Generally speaking, memory bandwidth is a key factor that may potentially limit the speedup on multi-core systems. Figure 4 shows how the average FSB bandwidth utilization varies with the number of threads. The bandwidth usages of all workloads except Gabor with the MPEG2 and HDTV inputs are far below the saturated FSB bandwidth capacity (~7GB/s) supported by the system. This seems to indicate bus bandwidth does not limit the scalability of most of these workloads on an 8-core system. However, the scalability is limited by the instantaneous bandwidth usage for some workloads, such as Gabor and SIFT. We perform interval sampling of the memory subsystem behavior over time. Figure 5 shows a representative phase of the bandwidth usage over time for these two workloads on eight cores with the MPEG1 video input. Several modules in these two workloads have higher bandwidth requirements than the saturated bandwidth provided by the system.

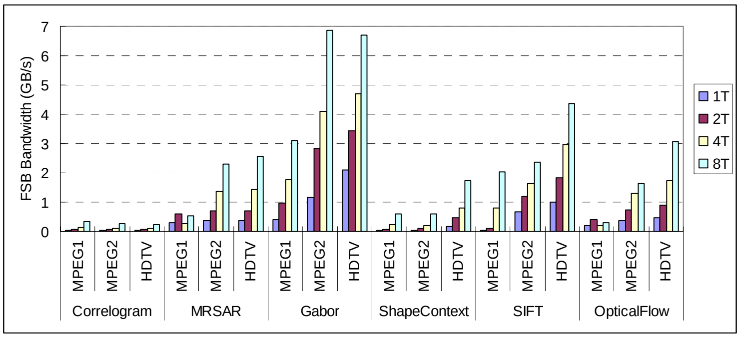

Fig. 4. Average FSB bandwidth utilization vs. number of threads for the six video feature extraction applications on an 8-core system

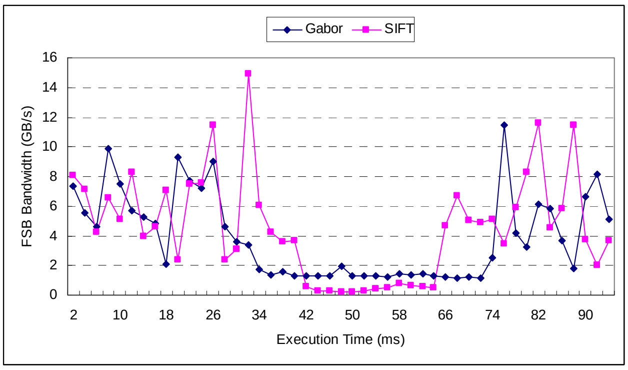

Fig. 5. Bandwidth usages over time for eight-threaded Gabor and SIFT workloads with the MPEG1 video input on an 8-core system

By this study, we can see most of the parallel CBVIR applications have very good scaling performance, which can be expected to scale well on even more cores/processors. Our analysis shows there are no obvious parallel limiting factors for performance scalability, e.g., very low synchronization and small sequential regions even with up to 8 cores. The main scalability limiting factors for SIFT and Gabor are load imbalance and the amount of available system bandwidth. The two parallel applications would perform better on multi-core systems with a higher memory bandwidth. Furthermore, we may need to conduct more fine-grained-level parallelization for the SIFT application to reduce the load imbalance and get maximum performance on future large-scale multi-core systems.