An FPGA-cluster-accelerated Match Engine for Content-based Image Retrieval

Chen Liang, Chenlu Wu, Xuegong Zhou, Wei Cao*, Shengye Wang, Lingli Wang

State Key Laboratory of ASIC and System, Fudan University, Shanghai 201203, P. R. China
*Email: caow@fudan.edu.cn

***Abstract*—In this paper, a high-performance match engine for content-based image retrieval is proposed. Highly customized floating-point(FP) units are designed, to provide the dynamic range and precision of standard FP units, but with considerably less area than standard FP units. Match calculation arrays with various architectures and scales are designed and evaluated. An CBIR system is built on a 12-FPGA cluster. Inter-FPGA connections are based on standard 10-Gigabyte ethernet. The whole FPGA cluster can compare a query image against 150 million library images within 10 seconds, basing on detailed local features. Compared with the Intel Xeon 5650 server based solution, our implementation is 11.35 times faster and 34.81 times more power efficient.**

Keywords—CBIR; FPGA cluster; high performance computing

I. Introduction

Nowadays the scale of medical and scientific media databases are growing exponentially. Thus content-based media search tools with high speed and high usability becomes ever more important[1].

A typical content-based image retrieval (CBIR) system requires off-line indexing and training to build a media feature database and on-line matching to search the media feature database and retrieve the corresponding images.

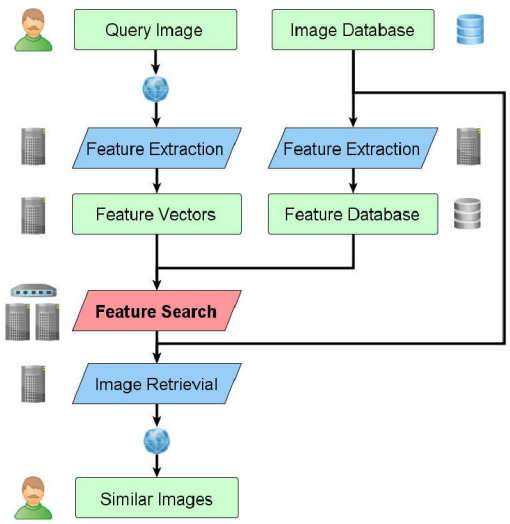

Fig. 1. A typical content-based image retrieval (CBIR) system

In this paper, we focus on the FPGA acceleration for the feature search. Because the speed of feature search calculation really determines the search delay of a query image, the scale of the media database to use, and the number of feature descriptors to include when indexing each image.

A high-performance match engine implemented on a high-end Virtex-6 FPGA cluster is proposed in this work. Each FPGA node features a finely tuned match calculation array with highly customized floating-point units.

This paper is outlined as follows: Section II describes the SURF-based match algorithm. Section III covers the proposed CBIR system. Section IV evaluates the whole system based on performance, power efficiency, and quality of service (QOS). Section V concludes this paper.

II. Match Algorithm for SURF-based CBIR

The SURF[2] algorithm provides better feature quality for each image than traditional approaches[3]. In this paper, FPGA acceleration for the matching process of SURF features is addressed.

After the SURF-based feature extraction, an image frame is represented by a sequence of feature point descriptors, and each descriptor is a vector with 64 dimensions.

Two steps are required to judge whether an query image matches a library image:

1) Check each of the feature points in the query image, to see if they match the library image.

2) Check whether the number of matched feature points in the query image is larger than a threshold.

To check if a query feature point matches a library image, we look for the 2 nearest neighbors of the query feature point among the feature points of the library image.

$$\left(\sum_{i=1}^{n}\left((a_i-b_i)\times(a_i-b_i)\right)\right)^{\frac{1}{2}} \qquad (1)$$

The distance is measured by Euclid distance in the 64 dimension descriptor space as shown in (1). Suppose the first nearest distance is FND and the second nearest distance is SND, feature point match occurs when *" FND/SND < 0.65 "*. Practically, we can removed the square root operator from the

978-1-4799-2198-0/13/$31.00 ©2013 IEEE

Euclid distance calculation and the feature point match formula is changed to (2).

$$FND^2 < 0.42 * SND^2 \quad (2)$$

If the number of matched feature points in the query image is larger than the threshold, an image match occurs.

III. The Proposed CBIR System

A. Overview of the Proposed CBIR System

Our CBIR solution consists of an Intel Xeon server, an ethernet switch, a disk array and many FPGA nodes.

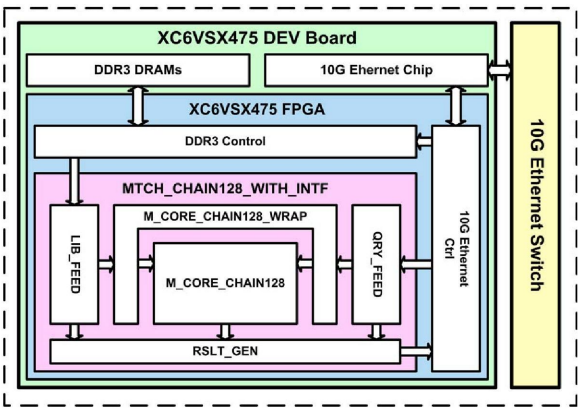

Fig. 2. Block diagram of the proposed FPGA node

An FPGA node consists of a Xilinx Virtex-6 XC6VSX475 FPGA, a 10-Gigabyte ethernet module and two independent DDR3 modules. The RTL design includes a match calculation array, a DDR3 controller and an ethernet controller. The whole design is controlled by state machines with no soft or hard CPU cores involved.

During the off-line training stage, a tara-byte scale media library composed of movies and pictures was indexed using SURF[2] local features to form a media feature library.

When our match engine powers on, each FPGA node loads a unique part of the feature library. The more nodes we use, the larger our feature library can be. Currently, each node can handle a 16GB feature library segment.

When the query images come in from CBIR system users, the Xeon server extracts SURF features from the query images and broadcasts the query feature packets to each FPGA node through standard 10-Gigabyte ethernet.

When a query feature packet is received from the server, the FPGA node loads the query vector into its match calculation units and begins to traverse the 16G feature library segment stored in its DDR3 memory. Each query frame is compared against each library frame through a sequence of distance calculations. The matched frame number pairs are buffered and sent back as result packets.

When the Xeon server has collected all the matched frame number pairs from each of the FPGA nodes, it retrieves the corresponding images or video frames from the disk array and sends them back to the CBIR system users.

Thank to the robustness of SURF[2] local features, our CBIR system can retrieve more useful results than previous systems[3][4]. For example, rotated versions of the query image and larger images containing similar objects can be retrieved. To some extent, we offer a more versatile "similarity" definition in our implementation.

B. Designing Customized Floating-point Units

Several customized floating-point units are designed here. Among them the most notably ones are the fused floating-point *(a-b)*(a-b)* unit, the 4-operand floating-point positive-number adder, and the continuous floating-point accumulator.

1) The fused floating-point *(a-b)*(a-b)* unit. In this design, floating-point substraction and squaring are pipelined and some redundant rounding logic are discarded.

2) The 4-operand floating-point positive-number adder. The total delay of a 4-operand floating-point addition is reduced from 8 cycles to 6 cycles by using a fused 4-operand floating-point adder. Since Euclid distance calculation only includes positive number addition, the LOD[5] logic is removed, and thus more area is saved.

3) The continuous floating-point accumulator. In this work, we have designed a customized floating-point accumulator, in order to do accumulations efficiently and continuously, unlike previous works[6].

Our continuous FP accumulator is 5-stage pipelined. Instead of feeding back FP sum from stage 5 output to stage 1 input, we have implemented an internal feedback path. Consequently, our accumulator can add one new FP input to the current FP sum every cycle without stall or interleaving.

C. Tuning the Match Array

Two architectures are studied. Both of them are tightly coupled with the system bandwidth.

1) The chain architecture. The key concept of the chain architecture is to process each feature point in the query image with a separate match core node. The whole match core is a long chain of small-scale match core nodes.

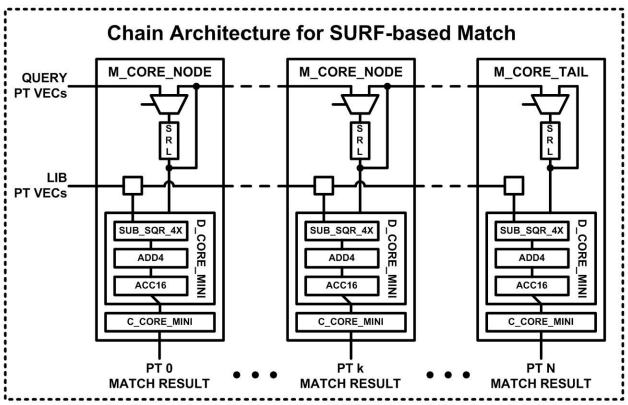

Fig. 3. Chain Architecture for SURF-based Feature Match

2) The tree architecture. The tree architecture adopted large-scale distance cores implemented with 64-operand adder trees, so as to simplify the control logic and save area.

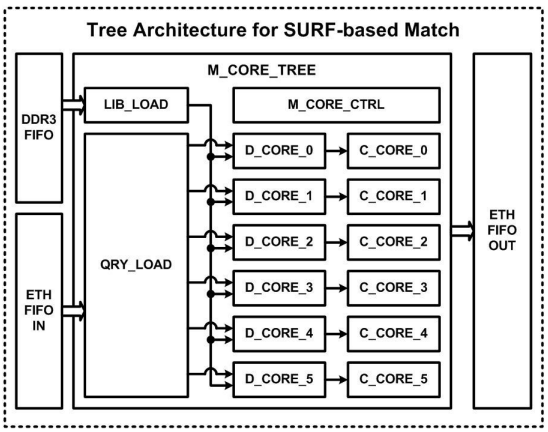

Fig. 4. Tree Architecture for SURF-based Feature Match

D. *Lower-power considerations*

Firstly, in the chain-based match cores, the enable signals of DFFs in the match nodes are dynamically controlled according to the current work load.

Secondly, in the top-level CBIR system, we implement fixed length feature query packets with 0xffffffff paddings. 0xffffffff stands for overflow in IEEE 754 standard FP format. An 0xffffffff (overflow) input will cause FP units to overflow, and output an 0xffffffff (overflow). This technique naturally limits the toggle rate of the unused calculation units, saving a lot of power.

IV. Experimental Results

In this paper, we evaluates our design on three levels: the match core level, the FPGA-node level, and the whole CBIR system level. Performance, power consumption and system usability are measured.

A. *Performance evaluation of different match cores*

Four match core designs are studied. They are the 96-point chain core, the 128-point chain core, the 96-point tree core and the 128-point tree core. All the designs are implemented on Xilinx Virtex-6 XC6VSX475 FPGA with ISE 13.4.

TABLE I. PERFORMANCE OF DIFFERENT MATCH CORES

	96-pt chain	*128-pt chain*	*96-pt tree*	*128-pt tree*
DFF	139581	185980	120133	159047
LUT	136177 (45%)	182106 (61%)	132221 (44%)	157213 (52%)
DSP	384	512	384	512
Fmax	204.625 MHz	154.154 MHz	204.248 MHz	169.722 MHz
Ftest	200MHz	150MHz	200MHz	166MHz
Dist calc per sec.	1.2G times	1.2G times	1.2G times	1.3G times

In Table IV, the 128-point tree design beats the 128-point chain design. In large FPGA designs, interconnection plays a key role in the final performance. Smaller and simpler designs usually run faster.

However, the chain architecture has a relatively flexible structure, which helps placement and route. When DDR3 and Ethernet controllers are connected, the chain-based designs would eventually beat the tree-based designs.

B. *Power simulation of different match cores*

In order to verify our low power techniques, we did some power analyzes based on the post-place-and-route simulation results, using Xilinx XPower tools.

The 96-point tree match core and chain match core were analyzed with 4 levels of work load: 24 points per image, 48 points per image, 72 points per image and 96 points per image.

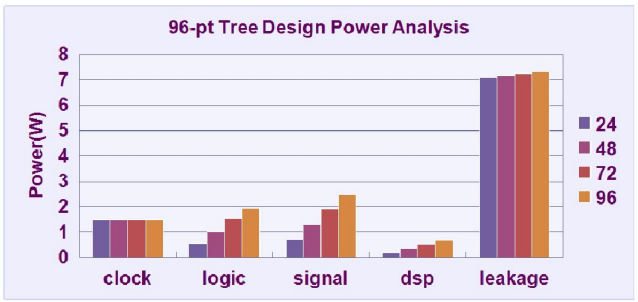

Fig. 5. Power analysis of the 96-pt tree design (under different work load)

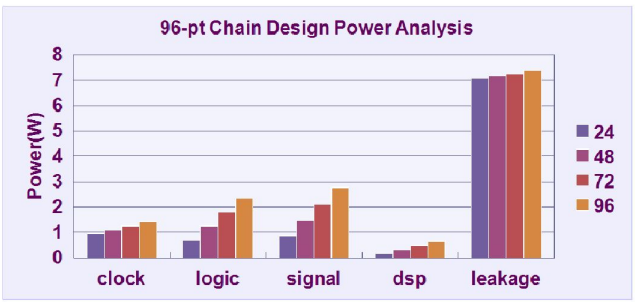

Fig. 6. Power analysis of the 96-pt chain design (under different work load)

We observed remarkable clock power difference in Fig. 6, but not in Fig. 5. Because only the chain-based core employed the flip-flop enable technique.

As shown in both Fig. 5 and Fig. 6, power consumption varies significantly with the current workload. The special query packet padding format limited the toggle rate of the unused FP calculation units, which in turn saved a lot of power.

C. *Performance of the FPGA node design*

Two FPGA node designs are implemented. One is based on a 128-point chain match core. The other is based on a 96-point tree match core.

The chain based FPGA node design can handle a query image with 128 feature points. Its match core operates at 122MHz and calculates 128 Euclid distances every 16 cycles.

So it features 976M 64-dimension distance calculations per second.

The tree based FPGA node design can handle a query image with 96 feature points. Its match core also operates at 122MHz and calculates 96 Euclid distances every 16 cycles. So it features 732M 64-dimension distance calculations per second.

We've also implemented our match algorithm on an Intel Xeon 5650 server with 6 cores and 12 threads. The program performs best with 24-thread parallelism, and achieves 86M 64-dimension distance calculations per second.

So our chain based FPGA node design actually accelerates the match process by 11.35 times against an 6-core Intel Xeon server.

D. *Power Efficiency of the FPGA node design*

Power meters were connected on the power supply lines of the Xeon server and FPGA nodes. The calculation devices, the memory devices and the cooling devices had all been taken into account, when power consumption was measured.

A 4-FPGA cluster with the 128-point chain design consumes 205W power when doing feature match operation at top speed. An Xeon server consumes 148W power when doing feature match operation at top speed.

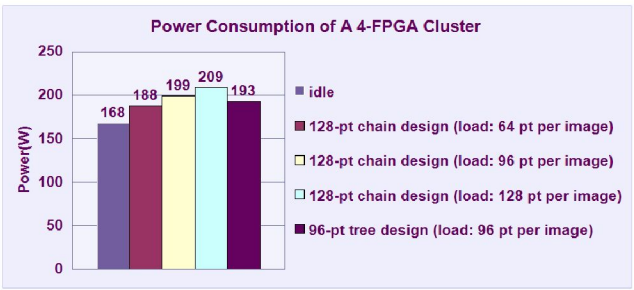

Fig. 7. Power Consumption of a 4-FPGA Cluster

With one Joule of energy, our FPGA node with the 128-point chain design can accomplish 18.679 times of 64-dimension distance calculation. By contrast, an Xeon server can only do 0.593 times of such distance calculation with the same energy.

So our chain based FPGA node design is actually 35 times more power efficient than an 6-core Intel Xeon server.

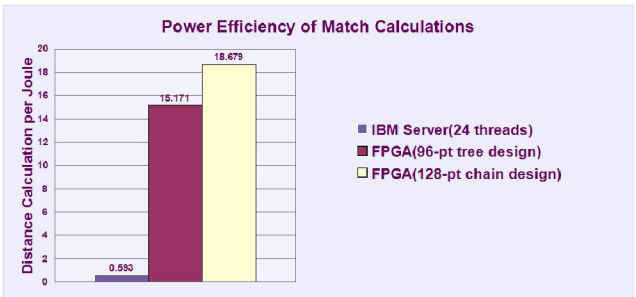

Fig. 8. Power Efficiency of Feature Match Operation

E. *System Usability and QoS*

The whole operation flow of our CBIR system has been described in Section Two. As mentioned before, our system is scalable. It can traverse a larger feature database within a given time, when more FPGA nodes are available to search the additional feature database in parallel.

Our CBIR system operates with 12 FPGA nodes working in parallel, and the total feature database size is about 192GB. In our design, we extract 50 feature points from each library image in average. And the 12-FPGA CBIR cluster holds an index for 150 million library images in its memory.

All the FPGA nodes work in parallel to serve the same query image search. It takes our chain based FPGA node design 8.1 seconds to compare the query image features against the 16GB on board memory. Taking into account feature extraction delay and network delay, the total query delay of our CBIR system is about 10 seconds.

V. Conclusion

In this paper, a high-performance feature matching engine for content-base image retrieval systems is proposed. An CBIR system is built with an Intel Xeon server and 12 FPGA nodes.

Our 12-FPGA cluster can traverse 150 million images indexed with high quality local features within 10 seconds and provide numerous result images with a more versatile "similarity" definition than previous CBIR solutions[3][4].

We hope to apply the proposed CBIR design to medical care and scientific research applications, where both detailed feature compare and short search delay are demanded.

Acknowledgment

This paper is supported by National Natural Science Foundation of China(61131001, 61171011).

References

[1] Y. Liu, D. Zhang, G. Lu and W. Ma, "A survey of content-based image retrieval with high-level semantics," in Pattern Recognition, vol. 40, no. 1, pp. 262–282, January 2007.

[2] H. Bay, T. Tuytelaars, and L. V. Gool, " Surf: speeded up robust features," European Conference on Computer Vision, pp. 404–417, May 2006.

[3] E. Castillo, C. Pedraza, J. Castillo, C. Camarero, J. L. Bosque, R. Menendez and J. I. Martinez, "SMILE scientific parallel multiprocessing based on low-cost reconfigurable hardware," in IEEE Symposium on Field-Programmable Custom Computing Machines, pp. 277-278, April 2008.

[4] A. Noumsi, S. Derrien and P. Quinton, "Acceleration of a content-based image-retrieval application on the RDISK cluster," in IEEE International Parallel and Distributed Processing Symposium, April 2006.

[5] V. G. Oklobdzija, "An algorithmic and novel design of a leading zero detector circuit: comparison with logic synthesis," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 2, no. 1, pp. 124-128, March 1994.

[6] M. R. Bodnar, J. R. Humphrey, P. F. Curt and D. W. Prather, "Floating-point accumulation circuit for matrix applications," in IEEE Symposium on Field-Programmable Custom Computing Machines, pp. 303-304, April 2006.