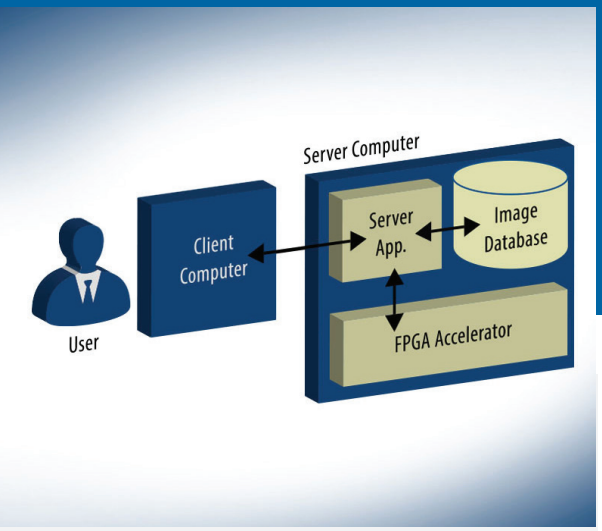


Boosting the Speed of Image Searches

By Hidetoshi Matsumura, senior researcher, Fujitsu Laboratories

FUJITSU

The Design Team:

Founded in 1968 as a wholly owned subsidiary of Fujitsu Limited, Fujitsu Laboratories Ltd. is one of the premier research centers in the world. With a global network of laboratories in Japan, China, the United States and Europe, the organization conducts a wide range of basic and applied research in the areas of Next-Generation Services, Computer Servers, Networks, Electronic Devices and Advanced Materials.

Challenge:

Finding other document pages containing images that partially match those on a particular document page is compute intensive and must be executed at human-interaction speeds. This speed can be achieved by using multiple servers, but at a significant cost in power and resources.

Solution:

Offloading the most compute-intensive algorithms in our search procedure to an FPGA significantly increases search speed without the massive costs of scaling out to multiple servers.

The Project

We developed a document search system with a function called partial image search. This function employs a document page as a query page, and performs an image-based search across a large document database. The query page and the pages of the documents in the database are compared in their appearances. The system does partial matching – pages that include a part of the query page are retrieved from the database and returned as search results. Minor changes such as resizing, cropping, and changing the text on an image are ignored.

Particularly in presentation materials, the minor changes mentioned are often performed when the images are reused. Hence, partial image search is useful for retrieving the former versions of the documents or the source of the image. It enables the efficient reuse of the document and allows more efficient production work.

Figure 1 shows some examples. The left column shows the key pages for which we wish to find similar-looking pages. The center and right columns show pages that would be identified as similar enough to call out as partial matches. They each contain an image that is also present on the key pages.

Figure 1. Examples of Partially Similar Images

ALTERA
now part of Intel

The Design Challenge

To realize this, we treat the query page and the pages of the documents in the database as images. Then we detect the distinguishing points in the images and exhaustively compare those points in the query image and database images. This results in a quantitative measure of similarity between the query image and each database image.

However this approach requires a huge amount of computational power. By using a single CPU, a search can take over one minute for running through upwards of 10,000 database images. To be practical, a search should be completed in a short enough time to be considered interactive.

Our first attempt to improve the system performance was by optimizing the software, which includes adopting Streaming SIMD Extensions (SSE). However, the search speed was still insufficient. Next, we attempted the scale-out approach using multiple servers. But achieving an adequate performance requires enormous power consumption and installation space. Finally, we decided to identify the intensive algorithms within the approach and accelerate them.

The Design Solution

Our approach for comparing images and scoring pages for similarity includes a number of stages. First, to retrieve resized images from the database, the query image is scaled into various sizes. All of them will be compared with the database images in subsequent steps of the algorithm. Next, feature extraction is performed for all the scaled query images and the database images. Feature extraction is a widely used method in computer vision. It is a kind of dimensional compaction, used to reduce the computational load of the matching, detection, and recognition processes. We compact the images by identifying distinguishing points in the images, called keypoints, which we then enumerate. This step is called keypoint detection. Then, a vector that compactly expresses the feature composed of the pixels around the keypoint is calculated. This step is called feature description.

We use various methods for keypoint detection and feature description. In our partial image search algorithm, Canny edge detection and Binary Robust Independent Elementary Features (BRIEF) are adopted as the keypoint detection and feature descriptor algorithms respectively. Through Canny edge detection, the edge and the corner pixels in the image are enumerated. The BRIEF descriptor is a 128 bit data element. A 48x48 pixel region is set around the keypoint, and 128 randomly selected point-pairs in this region are used for calculating the BRIEF descriptor. Each bit of the BRIEF descriptor is the sign bit of the subtraction result between the two pixels of a point-pair. Thus, the keypoint and BRIEF descriptor make up a very compact description of a recognizable feature in the image, obtained with minimum computation.

Finally, the matching process is performed between the query image and all the database images. For each keypoint in the database image, a keypoint in the query image that has the most similar BRIEF descriptor is selected. After that, the database image keypoints are classified according to the difference of the position in the image between the image keypoint and the most similar query keypoint, labeled as "voting" in Figure 2. If the number of a group member exceeds a predefined threshold, the region extracted from the group member is judged as a match to the corresponding region in the query image.

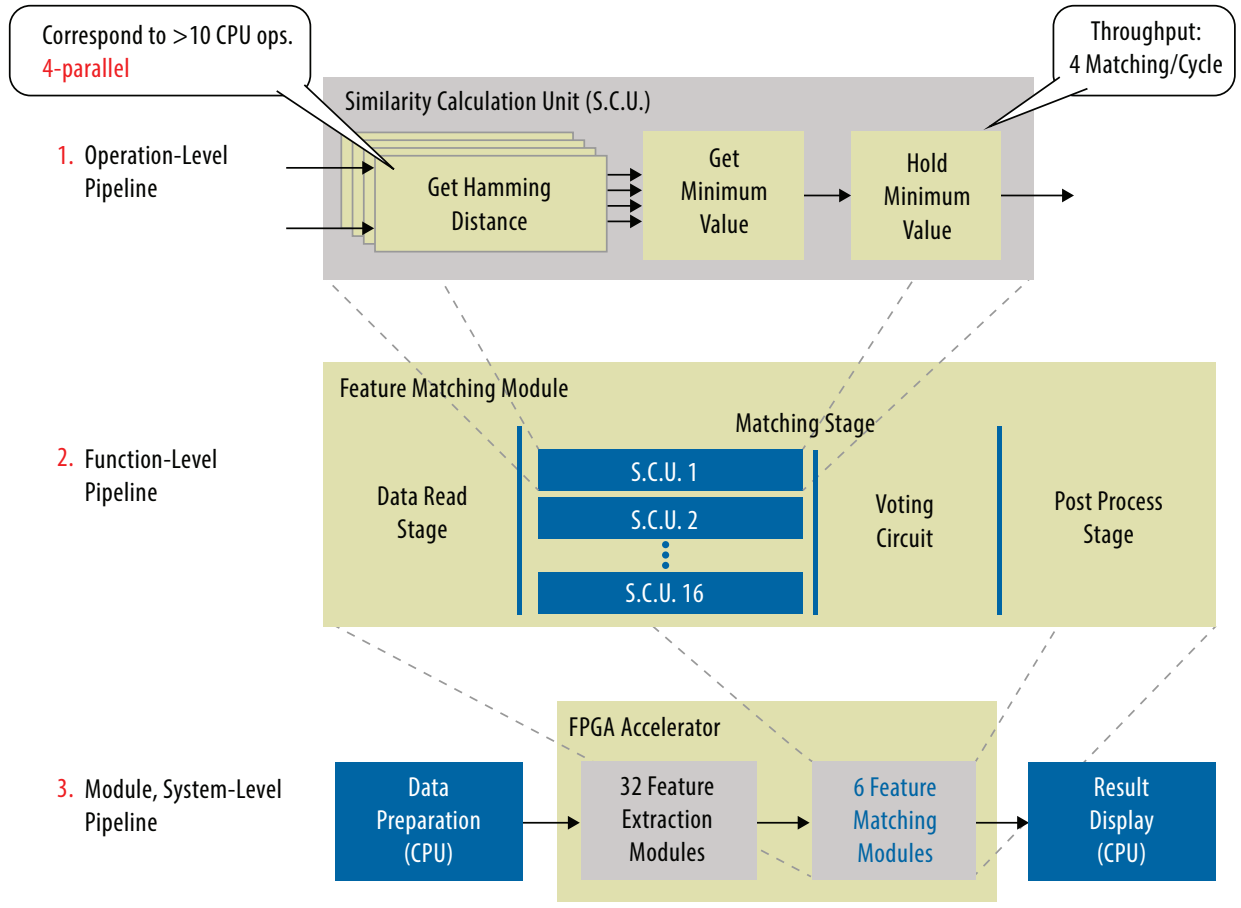
The similarity between two BRIEF descriptors is measured in Hamming distance. This simple operation in similarity calculation is one of the characteristics of the BRIEF descriptor. To reduce the computational load, a two-step matching scheme, which consists of coarse matching and fine matching, is adopted. The number of keypoints is reduced by limiting the position of the keypoint on the grid: a 6-pixel grid for coarse matching and 12-pixel grid for fine matching.

In spite of adopting this two-step scheme, the computational load is still huge, and the processing stage occupies most of the execution time in a CPU-only implementation. Hence, we implemented the Hamming distance calculations and some related operations on an FPGA accelerator. Feature description is also a heavy task and offloaded to the FPGA accelerator.

Now let us examine the FPGA accelerator architecture, shown in Figure 2. Lines 1 and 2 in Figure 2 illustrate the architecture of the feature-matching module. Hamming distance calculations and several related operations are integrated in the similarity calculation unit (SCU). The unit executes the operations for four keypoint pairs in parallel. The operations for one keypoint pair correspond to over ten CPU operations. The throughput of this calculation unit is four keypoint pairs per cycle. There are 16 instances of this unit in one feature-matching module, and 6 instances of feature-matching modules. As a result, the ideal performance of feature matching is 4 x 16 x 6 = 384 keypoint pairs per cycle.

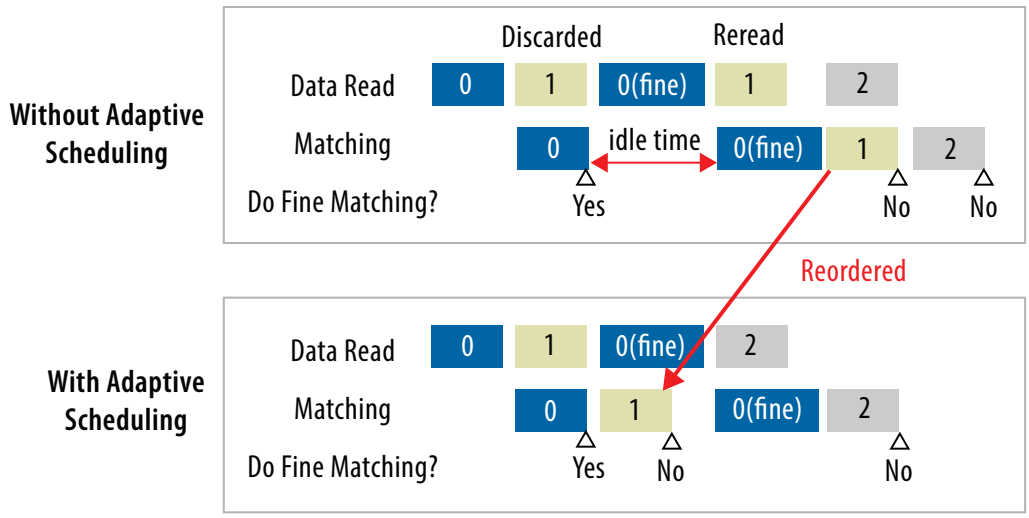
In addition to this significant parallelism, we use a multi-level pipeline structure, shown also in Figure 2. The SCUs employ an operation-level pipeline, while the feature-matching modules use a function-level pipeline. The feature extraction and matching modules are also pipelined, and so are the software and hardware on the entire FPGA accelerator.

Figure 2. Multi-Level Pipeline Structure



To hide the DRAM access latency, data read is executed ahead of matching . To remove the idle time of the processing units, the feature-matching module changes the task schedule adaptively (Figure 3). Without this, pre-fetched data would have to be discarded and idle time emerges when fine matching occurs.

Figure 3. Adaptive Scheduling vs. Non-Adaptive Scheduling



Theory of Operation

Figure 4 shows a high-level diagram of the document search system. It consists of client computers for the user interface and a server computer including the FPGA accelerator. They communicate through the Ethernet network. The user interface is implemented on the client computer. It sends the query request – a query image – to the server computer and receives the search results. The server computer can communicate with multiple client computers in order to share documents as a general file server. As described in the section above, the feature extraction and matching processes are offloaded to the FPGA. The remainder of the algorithm, which includes the communication with the client computer and the file I/Os are performed on the server computer's CPU.

Figure 4. System Overview of Document Search System

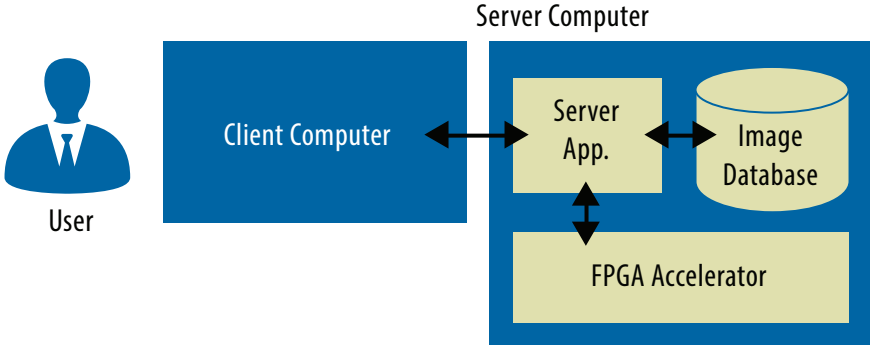
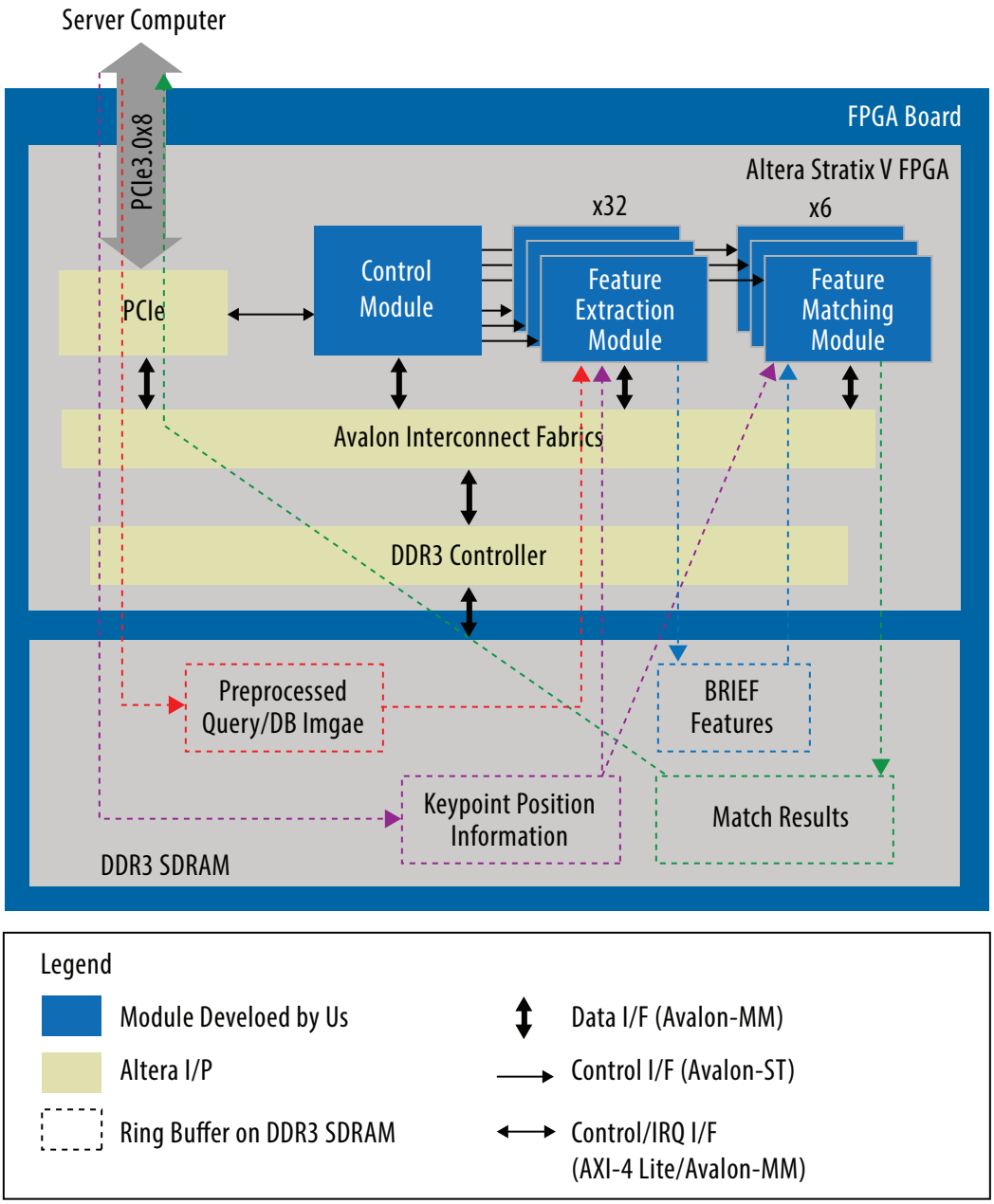


Figure 5 illustrates the overall implementation of the accelerator in the FPGA. PCI Express® 3.0 x8 is used for the communication between the server computer and FPGA accelerator. We developed three modules: the feature-extraction module, the feature-matching module, and a control module. All of them are connected to the Avalon® Streaming (Avalon-ST) bus for direct control connection, and they also have an Avalon Memory-Mapped (Avalon-MM) interface to access the data on the DDR3 SDRAM. Other modules in the FPGA accelerator are Altera's standard intellectual property (IP) cores, and the whole system is constructed by Altera's Qsys system integration tool. As a result, this system can be easily ported to a variety of Altera FPGAs. For this system, we instantiated 32 feature-extraction modules and 6 feature-matching modules.

Figure 5. FPGA Accelerator Architecture and the Data Flow



Results

Before the adoption of an FPGA accelerator, a search takes over one minute for running through upwards of 12,000 database images. The accelerator shortens this time to approximately one second – fast enough to be called "interactive". Compared with a CPU-only solution with the same performance, the power consumption and the installation space are 1/30 and 1/50 respectively.

Table 1. Performance Comparison

	Processing Speed (Database Images per Second)
CPU Only	200
FPGA- Accelerated	12,000

Additional Links:

- **Fujitsu Develops Technology for Instantaneous Searches of a Target Image from a Massive Volume of Images press release**

"This Design Solution describes an actual design that has been developed. However, it does not represent a supported product or reference design, and is not orderable from Altera. If you would like additional information, please contact Altera's authorized distributor".

Altera
101 Innovation Drive
San Jose, CA 95134
USA
Telephone: (408) 544 7000
www.altera.com

Altera European Headquarters
Holmers Farm Way
High Wycombe
Buckinghamshire
HP12 4XF
United Kingdom
Telephone: (44) 1 494 602 000

Altera European Trading Company Ltd.
Building 2100
Cork Airport Business Park,
Cork, Republic of Ireland
Telephone: +353 21 454 7500

Altera Japan Ltd.
Shinjuku i-Land Tower 32F
6-5-1, Nishi Shinjuku
Shinjuku-ku, Tokyo 163-1332
Japan
Telephone: (81) 3 3340 9480
www.altera.co.jp

Altera International Ltd.
Unit 11- 18, 9/F
Millennium City 1, Tower 1
388 Kwun Tong Road
Kwun Tong
Kowloon, Hong Kong
Telephone: (852) 2945 7000
www.altera.com.cn

Altera Technology Center
Plot 6, Bayan Lepas Technoplex
Medan Bayan Lepas
11900 Bayan Lepas
Penang, Malaysia
Telephone: 604 636 6100

now part of Intel

DS-1004-1.0