Pipeline Processing Illustrated

Making High-Quality Real-Time Image Processing Possible

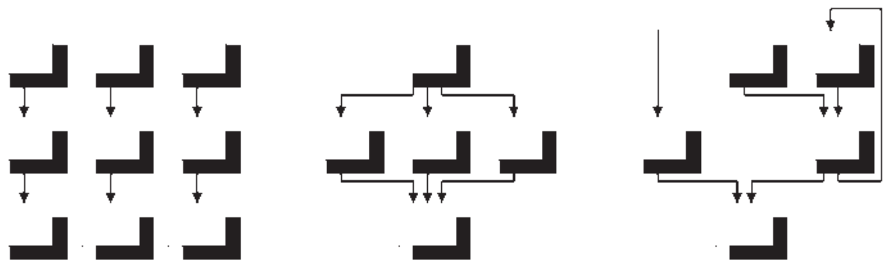
Today's powerful image processing technology enables the automation of a wide range of inspection and visualization tasks in many commercial markets. The processing of large and often complex images requires tremendous computational horsepower to quickly and efficiently handle the number of tasks associated with image processing, including:

- Acquisition of data from a wide variety of sensors with different formats, input rates, resolutions, and synchronization parameters at various frequencies, including infrared, x-ray, and visible light.
- Processing of a large volume of continuous data while performing complex algorithms that may vary from frame-to-frame.
- Storage and retrieval of image files in a variety of resolutions and formats.
- Display of large images at real-time rates in various formats, including gray scale, true color, pseudo color, and stereo.
- Analysis functions including real time feature extraction and object (blob) recognition capabilities.

The greatest challenge in image processing has been to develop systems that can handle all of these requirements with flexibility and precision, *in real-time*. To that end, Datacube has been a pioneer in the development of pipeline processing — a software architecture that provides exceptionally fast manipulation of large images where others have failed.

Pipeline Processing Surpasses Industry Standard DSP Platforms

In many image processing applications, the time in which data must be processed is fixed, and the success of the image processing system is dependent upon its ability to deliver results within the given time limits. Pipeline processing offers the advantage of determinism—the time it will take the pipeline processor to complete the operations is known in advance, before system development even begins. With general purpose processors, the time it will take to



Like an assembly line, data in a pipeline processing system passes through a series of specialized computational elements that are connected sequentially. In addition, these elements can be reconfigured on the fly to accommodate countless numbers of application-specific processing needs.

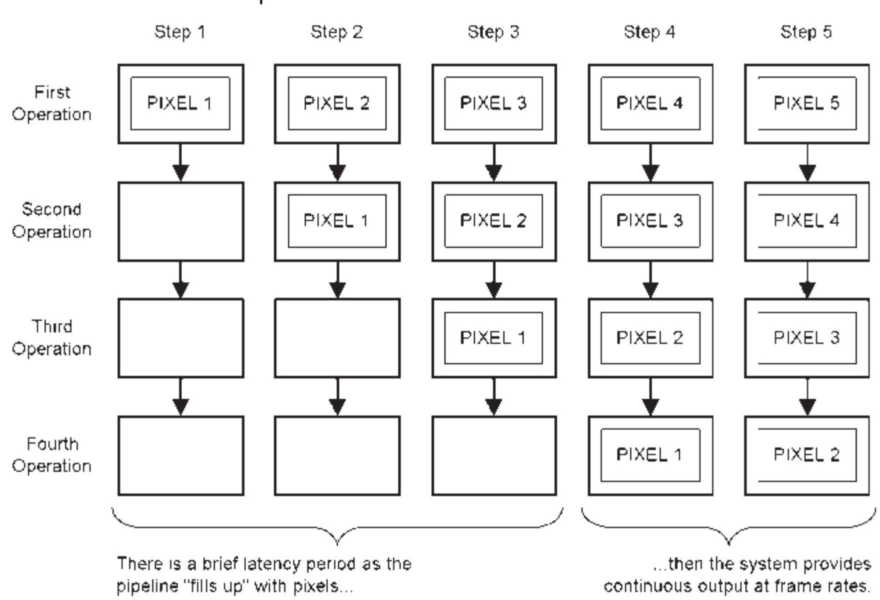
perform operations can be estimated, but will not be known precisely until after the system development has been completed.

For example, a given imaging algorithm may include four separate operations. Imagine that on a DSP-based system, these operations can typically be completed in 4, 7, 8, and 6 ms, respectively. Logically then, the total processing time could be calculated to be 25 ms, the sum of the time required for each of the individual operations. However, within a DSP-based architecture, the presence of some operations in an algorithm effects the processing time of others. Therefore, the actual time it takes for any application to be executed on a DSP-based system can not be calculated until the program has been completed and compiled, and could significantly more than 25 ms.

On a pipeline image processing system, even if each of the four operations took 8 ms, the total processing time would only be 8 ms after an initial latency period, because pipeline systems process images through an assembly line-like pipe with operations performed concurrently.

Here's how it works: Each step in the illustration below represents another tick of the system clock (in a 40 MHz pipeline, that equals 25 ns). Pixels enter the system one at a time, and with each tick of the clock (every 25 ns), they move from one operation to the next. The individual pixels follow each other, resulting in a steady stream of output as soon as the first pixel completes its journey through the entire pipeline.

CPU- and DSP-based platforms must make up for their slower processing times by skipping frames which may result in missed information, or by adding processors which adds both expense and overhead to the total system.



Comparison of Popular Processor Speeds

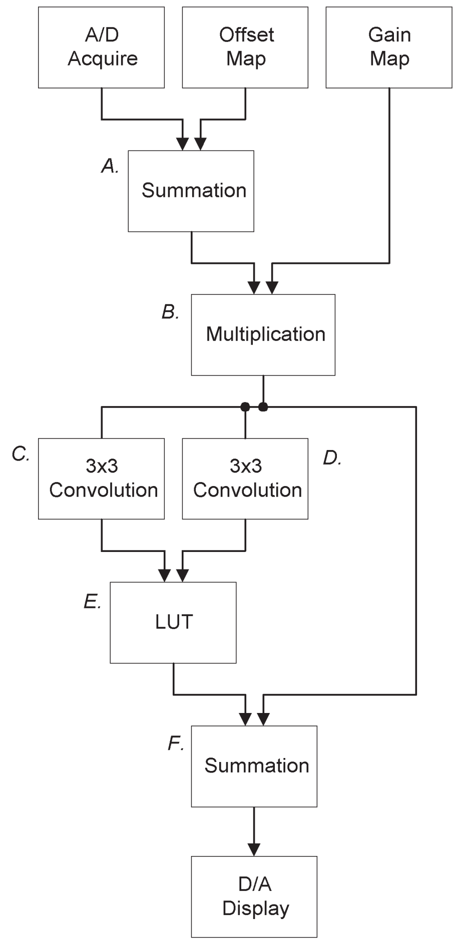
The application depicted by the highly simplified diagram to the right provides gain and offset correction, plus edge enhancement using a Sobel filter for real-time analog video. The benchmarks listed illustrate how much faster pipeline processing is than two of its most common alternatives, the Pentium CPU and C80 DSP. The times provided are for operations performed on 512x512x8-bit images. Note that while certain *individual* operations may be performed more quickly on a CPU- or DSP-based platform, the *total processing time* of a pipeline image processing system can't be beat.

	CPU*	DSP*	Pipeline @40Mhz*
A. Summation 1:	10 ms	3 ms	7 ms
B. Multiplication:	10 ms	3 ms	7 ms
C. Convolution 1:	48 ms	10 ms	7 ms
D. Convolution 2:	48 ms	10 ms	7 ms
E. Lookup Table:	7 ms	2 ms	7 ms
F. Summation 2:	10 ms	3 ms	7 ms
Total:	133 ms	31 ms	7 ms

Other Significant Benchmark Comparisons:

	CPU*	DSP*	Pipeline @40Mhz*
8x8 Convolution:	336 ms	70 ms	7 ms

* Times provided are rounded to the nearest ms.



So you can see, the image processing system based on Datacube pipeline processing technology far outperforms the CPU- and DSP-based options — both on key individual operations and more importantly, on overall system performance.

The Pipeline Processing Price/Performance Advantage

Image processing at frame rates requires the ability to handle a continuous stream of a very large volume of pixels. Pipeline processing does this more effectively than any other architecture on the market.

With Datacube's pipeline processors, a stream of video data is piped through a configurable series of specialized computational elements. The on-board processing elements are connected via very fast, high-bandwidth, non-blocking crosspoint switches to create pipelines. All of these elements operate simultaneously to produce a steady stream of processed image data. Multiple pipelines may be configured to work in parallel, providing even greater data throughput.

Even the most powerful general purpose CPUs and DSPs cannot begin to compare to the highly efficient nature of pipeline processing for many image processing tasks. The overall throughput of a system can be many thousands of times faster than traditional methods, with performances over 10,000 MIPS—approaching super computer power at a small fraction of the cost.

Additional Information

For more information about pipeline processing and the products mentioned in this document, please refer to the following literature available from Datacube:

ImageFlow Data Sheet
ImageFlow Technical Description
MaxPCI Data Sheet
MaxVideo 250 Data Sheet
Technical Training Data Sheet

(11/98) DS0099-1.1

	Pipeline Image Processing	***CPU/DSP Technology***
System Cost	Individual pipeline image processors can be implemented with low cost custom silicon devices.	It is not cost-effective to have multiple processors on a single board. Also, it may be difficult or sometimes impossible to divide tasks between multiple generalpurpose processors.
Processing Power	Systems are easy to expand by putting dozens – even hundreds of processors on the same board.	The more processes a CPU- or DSP-based system is asked to do,the slower it becomes.
System Throughput	Datacube pipeline processing guarantees processing rates up to 40 megapixels/second.	Throughput is variable at best, and it is difficult to predict performancein advance of implementation.
Development Environment	Datacube's ImageFlow software provides a time-tested, proven-reliable environment with easy-to-use development tools.	DSP-based development environments are dominated by expensive,exotic compilers, linkers and debuggers.

D A T A C U B E

Datacube, Inc. • 300 Rosewood Drive • Danvers, MA 01923 • Voice 978-777-4200 • Fax 978-777-3117 • http://www.datacube.com • info@datacube.com