Mobile handsets have evolved from the bulky, heavy, voice-only communications systems of a decade or two ago to today’s ultra-compact devices equipped for MP3 playback, still- and video-image capture, high speed data transmission (rivaling DSL services), and even GPS and television services. The evolving lifestyle of consumers has driven the integration of many functions into a small-form-factor unit. All consumer devices (digital still cameras, video cameras, MP3 devices) that support video and imaging functions are getting smaller by the day and trends are similar for non-consumer markets in which video and image processing are required. These trends present challenges for systems designers in the development of specific system features as well as in the packing and integration of those features into a system.

System designers need to accomplish real-time video and image processing in a small-form-factor device that consumes minimal power. High quality video and imaging performance requires sophisticated, built-in algorithms, however. Each of these requirements tends to drive the solution in the opposite direction, which presents integrated-circuit designers with a challenging task. Solutions developed for imaging markets thus tend to be balanced toward one of the parameters more than the other (either moderate cost, moderate performance systems with lower power requirements or high cost, high performance systems with increased power requirements). We have built an application-specific integrated circuit (ASIC) for the mobile imaging market that executes sophisticated image-processing algorithms at high pixel throughput.

Understanding the Requirements

Consider the design of an image processor targeted for mobile-handset-camera applications. Let’s begin by looking at the system requirements. A typical digital-image/video-capture system will need optics; an image sensor (CCD or complementary metal-oxide semiconductor (CMOS)); analog front-end circuitry that includes sample and hold, programmable gain amplifiers, and analog-to-digital converters; timing and clock generation circuits; RAW data; image-processing algorithms; and possibly a still- or video-compression engine (see sidebar). The basic image-processing algorithms perform several steps to convert the raw image data (Bayer-formatted color-filter data captured directly from the image sensor) into YUV or YCbCr color-space data, suitable for the still- or video-compression engine.

On the system level, each block can be software controlled based on the overall characteristics of the pixel data in the system. The central processing unit (CPU) is integrated with
the device to operate at a system level, giving the designer flexibility in controlling camera operation. In addition, system-level control functions are required for the support of pulse-width modulators, general-purpose input/output devices, and inter-integrated circuit (I2C) and/or serial peripheral interfaces to control other system-level mechanics.

The goal of the color-processing system is to process raw image pixel data from the sensor into a full-color image with a high degree of color accuracy. Optional image enhancement can improve the subjective quality of images. One of the biggest concerns for the video/imaging system designer is the sophistication of the image-processing algorithms for delivering print-quality images. High quality image processing generally requires multistage complex algorithms, and as a result puts a high demand on the processing capability of the system if high data throughput is to be maintained. Complexity and data throughput are the biggest reasons for adopting a hardware approach with a digital-signal-processor (DSP)-based architecture or a field-programmable gate array (FPGA)-based architecture for camera designs. High volume image-processing applications are compatible with standardized DSPs; for niche applications, designers tend to use FPGAs in their designs.

Color-processing algorithms have to be implemented using ASICs or FPGAs whenever pixel throughput is a key requirement. The ASIC approach provides a more cost-effective solution. All the memory required during color processing should be implemented as pipeline memory and the color-processing algorithm hardware should be tightly coupled to this memory within the internal data paths. This approach permits high throughput for applications demanding real-time video performance.

Depending on system requirements, color-processing algorithms can produce print-quality still images or crisp, vivid video. Histogram and statistical data are collected on every frame using multiple zones during image capture while the image data flows through the various stages of the image-processing pipeline in the ASIC. This data is analyzed and used for implementing the camera-system algorithms, including auto-exposure (AE), auto-focus (AF), and auto-white balance (AWB). These system-level algorithms control the image-processor blocks, the sensors themselves, and the motors and the mechanics of the camera system to obtain output images with the desired characteristics, such as proper brightness, sharpness, and accurate color.

For base-level operation, video and image processors need to support a range of image-processing algorithms, including bad-pixel correction, lens-shading correction, gamma correction, color image interpolation, and color correction. The processors also need to support sharpening, smoothing, and adaptive-light processing, as well as AF, AE, and AWB. In addition to the above baseline performance needs, handset original equipment manufacturers (OEMs) demand advanced image-processing algorithms such as electronic image stabilization and red-eye detection.

All imaging devices face the problem of imaging scenes with simultaneous extremes of lighting (high intensity in one region of the image and low intensity in another). Any AE algorithm will try to adapt to one area or the other, depending on the implementation (area versus spot-metered). Adaptive light algorithms can improve images captured under such extreme conditions (see figure 1).

In addition to the typical image-processing algorithms discussed above, a number of advanced features provide substantial additional value to a handset camera. Electronic

**Glossary**

**Adaptive-light Processing (ALP):** An image processing technique in which the actual pixel processing is varied according to the amount of light in the source image. ALP accommodates scenes with extreme lighting conditions (very bright areas and low light areas).

**Automatic White Balance (AWB):** Processing that ensures all colors in the scene will be represented faithfully.

**Color Image Interpolation:** Color resampling; an imaging method used to increase or decrease the number of pixels in a digital image.

**Gamma Correction:** A correction to the contrast of images and displays to account for the nonlinear response of the display or the capture device.

**Inter-integrated Circuit (I2C):** A serial computer bus used to connect low-speed peripherals in an embedded system or motherboard.

**RAW File Formats:** A RAW file contains the original image information as it comes off the sensor before in-camera processing.

**YCbcCr:** A family of color spaces used in video systems. “Y” is the luma component, and “Cb” and “Cr” are the chroma components.

**YUV:** A type of color encoding system. Again, “Y” stands for luma, which is brightness, or lightness. “U” and “V” provide color information and are “color-difference” signals. —PW. and M.B.
image stabilization, for example, allows consumers to capture video clips using a camera phone without shaky captures due to hand movement. Mainstream digital still and video cameras tend to implement these features using gyro that move either the optics or the image sensor to minimize motion artifacts even before the detector captures an image. Unfortunately, the small optics and ultra-small form factors of mobile handsets do not allow such elegant solutions. An electronic-image-stabilization (EIS) approach with an appropriate algorithm implemented in an ASIC can provide a cost-effective solution by minimizing frame-to-frame motion. EIS is typically implemented in hardware with software-programmable parameters per the choice of the system designer to deliver the desired response.

Lofty Goals

High end mobile handsets are being designed to replace consumer digital cameras. This not only drives the need for high quality image-processing algorithms, but also drives the need for solutions to imaging problems, such as red eye, caused by flash reflections from the retinas of subjects with dilated pupils. One design option is to include provisions for reducing the possibility of red-eye occurrences during image capture. A common approach used by hardware designers is to pulse high energy xenon flashes from digital cameras so that subjects’ pupils contract prior to image capture.

What can a camera processing system do if red eyes still appear in the raw image data? Image analysis and segmentation can help determine the presence of red eye in the captured image. Consumer digital still cameras use software-based red-eye detection and correction algorithms to fix the problem. Advanced solutions use hardware-based algorithms to address the detection and correction problem while maintaining high image data throughput.

Our solution incorporates such an approach, allowing the imaging system designer to actually detect all potential red-eye locations in the image while the pixels are flowing through the image-processing pipeline. The algorithm detects red-eye locations using physical and geometric characteristics of red eyes in the image data. This is accomplished by image segmentation using a combination of skin-tone detection and a search for certain features that would constitute a typical representation of red eye. These locations can be used to correct red-eye occurrences either while the image data is temporarily buffered in the local synchronous dynamic RAM (SDRAM) memory, or subsequently in the system (when image data is flowing through the pipeline, requiring 8 to 16 MB of SDRAM).

Another challenge facing camera designers is the use of CMOS image sensors for image capture in extreme low light conditions. This problem can be solved by enhancing the exposure of the image using the image-processing chip (see figure 2).

Cascading algorithms creates a parallel processing architecture and delivers the performance required. This approach balances cost, power, and performance. DSP architecture can deliver a lower-cost solution, but at the expense of performance; this typically translates to higher energy consumption. ASIC image processors are highly optimized to perform only a specific task, and hence they cannot be shared. Individual processors can be bypassed and turned off per the needs of the programmer or system designer.

Overall, the custom ASIC is designed in a pipeline approach wherein each imaging block processes the image data and passes the result to the next block. Timing of the entire chip is programmed so that image data goes through without creating bottlenecks in the system. The characteristics of each individual block (i.e., filter coefficients, gain parameters) can be controlled according to the designer’s needs. Furthermore, any block can be turned off under system-software control so that the entire system can operate at an optimal power level.

The new breed of consumer electronic devices demands a fresh look at the problem of devising imaging solutions that meet OEM requirements. Delivering such custom solutions can be difficult as no one solution addresses every need sufficiently. A modular approach provides design flexibility, high data throughput, and low power consumption. This solution brings the digital-still-camera class of image processing to small-form-factor, low-power, and cost-conscious consumer devices. Without sacrificing image quality, performance, or advanced features, the system designer obtains a very flexible product to implement a variety of still-image or video-processing applications.

System designers will encounter more and more optimal solutions, each solving the problem in his or her own way. To minimize development times and costs, the winning architecture may be the one that provides a scalable architecture for the consumer market.

Ping Wah Wong is the chief imaging scientist and a vice president, and Murty Bhavana is the vice president of marketing for Nethra Imaging Inc., Cupertino, CA. For questions, contact Wong at 408-257-5880; 408-257-0882 (fax); or ping@nethra.us.com.