RTT TECHNOLOGY TOPIC
February 2005

Camera Phone Design

In this month's Hot Topic, we look at some of the present compromises implicit in designing low cost camera phones and how various correction algorithms can be applied to help produce an acceptable user experience. Specifically, we want to consider some of the hardware and software requirements of a video phone capable of supporting a two way (duplex) voice and video call.

The image processing chain

When considering the imaging processing chain, it is logical to start with the image.

Our eyes see the world in terms of brightness (luminance also known as luma), colour (chrominance also known as chroma) and the shapes and patterns of objects within our field of vision. Our eyes have a remarkable dynamic range and can tolerate and process visual information in anything between direct sunlight (+100,000 lux) to a moonlit night (fractions of a lux).

There are various tricks that help us increase this dynamic range, for instance we just see things in black and white at very low light levels. We also have auto white colour balancing as part of our natural visual tool set. White has a green tinge in daylight, a yellow tinge under fluorescent light and an orange tinge under incandescent light, but the tinge disappears courtesy of our natural image processing chain.

These dynamic range and colour capture and colour correction capabilities have to be recreated in artificial image processing systems.

In addition, our natural image processing system is incredibly efficient at sorting out the entropy in the viewed image (the useful information) and redundant information that can be effectively ignored and discarded. This process is also adaptive. If you are hurtling down a hill on a mountain bike, your brain will just be processing the visual information relevant to the task in hand (staying on the bike and not hitting any trees in the process).

These compression capabilities and adaptive pattern recognition capabilities have to be recreated in artificial image processing systems.

We are also naturally adept at pattern recognition, recognising a face in a crowd for example.

These pattern recognition capabilities have to be recreated in artificial image processing systems.

Each component in an artificial image processing chain has a required function and a

'wanted effect'. Unfortunately most components also produce 'unwanted effects' which introduce impairments. We can however go some way towards cancelling out or at least concealing these effects. For example;

The Lens

The wanted effect in a lens is optical quality, the ability to focus an image on the sensor array with minimum distortion. Optional functions include a good depth of field, focal length, angle of view, and optical zoom. Unfortunately, because of cost, size and weight constraints, most lenses used in camera phones are far from perfect. The unwanted effects that result include vignetting (not enough light getting to the edge of the sensor array causing edge/corner shading) and lateral chromatic aberration caused by rays of light being sent obliquely across the colour filter array (see the section on the Colour Filter Array below).

The effects can be cancelled out or reduced by using **anti-vignetting/shading correction algorithms, anti blur algorithms** and **colour correction.**

The Sensor Array

A sensor array is an array of photosensitive cells, the discrete parts of which are described as pixels. The photosensitive cells are more or less identical to solar cells in that their job is to turn incoming photons from the lens into countable electrons. The efficiency with which they accomplish this task is described as Quantum Efficiency (QE) measured in bits per lux per second. **Bigger pixels collect more photons** but take up more space and cost more. **More pixels** on a sensor array **increase resolution** but take up more space and cost more.

A sensor array is either a Charge Coupled Device (CCD) or CMOS device.

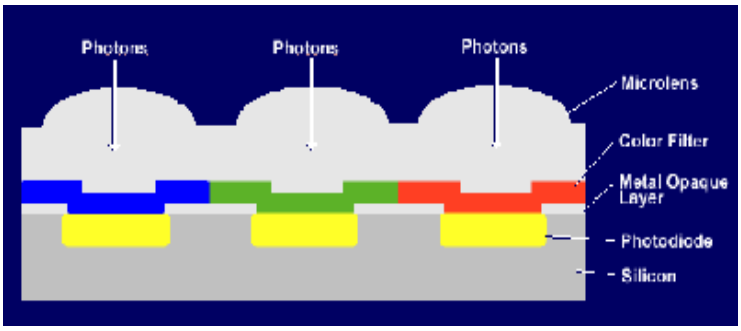
In a **Charge Coupled Device**, the charge is transported across the chip and read at the corner of the array.

In a **CMOS device**, each pixel has either 3 or 4 transistors, which amplify and move the charge across a wired backplane. Because each pixel on a CMOS sensor has several transistors located next to it, some of the incoming photons hit the transistors rather than the photodiode so sensitivity is lower than a CCD device The transistors also produce noise.

This noise (and the effects of transistor mismatch) can however be reduced by a process of double sampling and a combination of **noise cancellation and noise reduction algorithms.**

CCD sensor arrays produce better images particularly at higher (multi-megapixel) resolutions.

CMOS devices cost less, use less power and their ability to address individual pixels, individual rows of pixels and individual columns of pixels allows for more flexibility both in terms of image processing, pattern matching and image recognition.



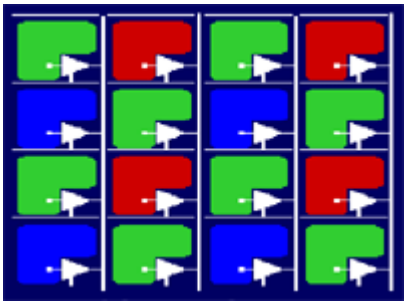
The diagram shows the sensor array.
In this example a small microlens is used over each pixel, light then passes through the colour filter array (see below) and then to the photodiode.

With thanks to Micron (Imaging Technology Overview)

The Colour Filter Array

Colour images are a mix of red (400 nm), green (510nm) and blue (700nm). A pixel will have a red, blue or green filter and the readings from 3 pixels (a red, blue and green pixel) are combined together to produce the chroma information. This is why colour sensors are three times less sensitive than black and white sensors. This part of the component chain is known as the Colour Filter Array.

Human eyes are less sensitive to green than they are to red or blue. The colour filters are therefore arranged in a Bayer pattern, originally invented and patented by Kodak. The Bayer 'checkerboard ' has red and green in one row and blue and green in the next.



Bayer pattern

The Bayer pattern means that **interpolation algorithms** have to be used. The unwanted effects of interpolation include optical cross talk (rays of light taking an oblique route through a red or blue pixel before landing on a green pixel) and blooming caused by charge overflows into neighbouring pixels. Anti aliasing filters reduce these effects but the filters blur the image and obscure detail. Using **edge-sharpening algorithms** (known as high frequency component gain algorithms) can put this detail back but these algorithms amplify noise in the flat regions of the image. This can be avoided by using **edge recognition algorithms.**

The sensor array and colour filter array provide the information needed for the DSP and microcontroller to perform a number of system functions including auto exposure, stray light compensation, auto focus, auto white balance, and gamma balancing (correcting for the non linear relationship between pixel value and displayed intensity on devices such as TV monitors). These are 'wanted effects' or required functions.

The unwanted effects introduced by the DSP are due to the fact that is has to digitise a complex combination of waveforms which have large variations in signal amplitude (wide dynamic range).

When the input bandwidth exceeds the sampling rate, the DSP will produce aliasing effects (difference frequencies/false frequencies).

False low frequencies caused by ADC undersampling create Moire effects (the distortion that you see when someone wears a check jacket on television), coarse quantisation causes fringing, ringing and shadowing. Some of these effects ripple up and down the processing chain. For example, a series of flash guns go off at a press conference and put the DSP into compression, prediction errors fill the output buffer and trigger heavy requantisation which produces tiling effects. If the system drops chroma coefficients in a desperate attempt to recover, then colour disappears (not that you will really notice by this stage). These are sometimes described as excess compression effects.

This brings us to the final stage of the processing chain, the JPEG and MPEG encoder.

The JPEG encoder / decoder

Entropy and redundancy are separated in still images by dividing the image into eight by eight pixel **macroblocks.** Each macroblock is coded in terms of it's luma and chroma content divided into what are termed as Y, U and V planes in which Y is the luminance and U and V are colour difference channels. Colour difference is a more efficient way of describing the RGB content of each discrete part of the image. The JPEG compression algorithms are based on a Discrete Cosine Transform (DCT) that separates out the luminance and chroma information in the spatial and frequency domain. These are expressed as DCT coefficients.

Compression can then be achieved by comparing one macroblock with other adjacent macroblocks and in effect just coding the difference from block to block.

This is something of an oversimplification. There is actually a stage described as **quantisation** which removes perceptually insignificant data (and reduces the number of bits per DCT coefficient) and **AC/DC prediction** which uses filters to predict co efficient values from one or more adjacent blocks but otherwise that's JPEG in a nutshell. There is also a version of JPEG known as motion JPEG - a succession of JPEG images used for video streaming.

The MPEG encoder

Generally though, moving images, i.e. video, are coded using MPEG (usually MPEG 4).

JPEG is based on the principle of macroblock to macroblock comparison. MPEG adds in image to image/frame to frame comparison and motion estimation.

Image comparison is based on I frames, P frames and B frames. **I frames** are

encoded as still images and are not dependent on any reference frames.
P frames depend on the previously displayed reference frame and **B frames** depend on previous and future reference frames. I frame coding is sometimes described as **intra coding** (literally, coding within), P and B frame coding is sometimes described as **inter coding** (literally, coding between).

In MPEG 1 and MPEG2, the macroblock sizes are fixed. In MPEG4, the macroblocks can be subdivided or sliced and become part of a far more complex encode/decode process involving shape coding, texture coding and motion estimation coding.

Motion estimation coding predicts the contents of each macroblock based on the motion relative to a reference frame. The motion vectors and the difference between the predicted and actual macroblock pixels are encoded.

The way this search and comparison algorithm is implemented by different vendors can make a substantially impact on the performance of the whole encode/decode process.

Search techniques include cross searching, step searching, diamond searching. Matching techniques include sum of absolute difference block matching, luma/chroma motion vector matching and methods based on deriving chrominance motion from luminance motion.

Shape coding is also known as object coding. The MPEG 4 standard describes how video objects can be described in terms of their shape. Video objects in an image stream make up a video object sequence (VOS) which is made up of video object layers (VOL). The video object layers grade the importance of the coded information and help to support **'graceful degradation' algorithms** sometimes also described as **enhancement layer encoding.**

Video object layers can be described within video object planes (VOP) that can be built into video object groupings (VOG).

Shapes can be opaque or transparent and 2D or 3D. They can also be described in terms of their 'texture' (**texture coding**), in which picture gradients (consisting of AC/DC coefficients) can be combined with **horizontal and vertical prediction algorithms** to reduce the overall code rate.

Most of the remaining algorithmic effort then gets focused on getting rid of the '**compression artefacts**' - the unwanted effects of compression. These include '**blocking**' in which the borders of each macroblock become visible in the reconstructed frame and '**ringing**' which creates distortion near the edge of image features.

Blocking is reduced by using a low pass filter. Deringing depends on being able to detect the edges of image features. A 2D filter is then applied to smooth out areas near the edges but with little or no filtering on the edge pixels (which would cause blurring).

Finally there are various **error resilience** techniques used to hide the fact that 'the

channel' (which for a camera phone includes the radio channel) is highly variable with high and **unevenly distributed errors and frame erasures**.

Error resilience depends on the use of resynchronisation and motion markers, the use of extender header codes, data partitioning and reversible length coding also known as forward decoding/backward decoding, also known as **bi-directional coding.**

Image processing software - processor and memory requirements

Each of the algorithms and filter functions highlighted above occupy processor clock cycles. Most of them also occupy memory space.

For JPEG, the DCT transform, quantisation, variable length coding and AC/DC prediction all have an impact on processor loading.

For MPEG, motion estimation, shape coding and texture coding add to the encoder/decoder overhead.

Quite often there are just not enough clock cycles to go round.

Image processing hardware

Which is why most image processing chains in mobile phones use hardware co-processors/hardware accelerators to meet power budget and latency constraints.

The power budget/latency budget issue has also prompted device vendors to suggest and sometimes implement a range of **new memory and DSP and microcontroller architectures** optimised for media processing/image processing applications.

This creates a number of complex decision issues when designers are trying to decide on which algorithms to use on which hardware platform.

Image processing, voice processing and audio processing

An additional complication is that it is not just the image processing chain that we need to deal with but also voice and audio. The inclusion of wideband high quality solid state microphones in high end cameras today is a sure sign that enhanced audio capture capabilities will be an inherent part of future camera phone products - the addition of **Hear What I Hear** (HWIH) to **See What I See** (SWIS) as part of the user experience.

Summary

Camera phone design and the integration of the image processing chain into low cost small form factor power limited devices is a fascinating but complex challenge which demands a very intimate relationship between product specification and hardware and software engineering.

About RTT Technology Topics

RTT Technology Topics reflect areas of research that we are presently working on.

We aim to introduce new terminology and new ideas to clarify present and future technology and business issues.

Do pass these Technology Topics on to your colleagues, encourage them to join our Push List and respond with comments.

Contact RTT

RTT, the Shosteck Group and The Mobile World are presently working on a number of research and forecasting projects in the cellular, two way radio, satellite and broadcasting industry.

If you would like more information on this work then please contact

geoff@rttonline.com

00 44 208 744 3163