Whitepaper

Chimera™: The NVIDIA Computational Photography Architecture

Delivering stunning camera capabilities to the Tegra 4 Family of SoCs.

Table of Contents

Executive Summary

The world's first camera phone was launched commercially in late 1999 and over the last decade the number of camera phones and smartphones with high resolution cameras has increased exponentially. Smartphones provide the convenience of spur-of-the-moment photography and ease of sharing due to their form factor and connectivity features. Recent market studies forecast that the global camera phone market volume is expected to exceed 1.6 billion units in 2013[1].

Unlike digital SLR and point-and-shoot cameras, smartphone cameras are constrained by thin form factors and limited battery sizes. Therefore they cannot use bulky optical lenses, large image sensors, or additional DSP chips to improve picture quality. And to date, smartphones have been unable to provide very responsive performance for advanced camera features. A number of camera phones now include "good enough" camera features, but high end smartphone manufacturers are differentiating by using high resolution image sensors, built-in LED flash, and features such as smile detection, panoramic photography, and High Dynamic Range (HDR) imaging.

Most of these advanced software-based features rely on the limited processing capabilities of the Image Signal Processor (ISP) within a SoC (System-on-a-Chip), and post processing capabilities of the SoC's CPU core(s). While an effective architecture for limited camera effects, such a design does not support truly impressive computational camera effects.

The **NVIDIA® Tegra® 4** Family of mobile processors includes the **NVIDIA Chimera™ Computational Photography Architecture** which fuses together the performance of its **custom GPU cores**, **quad core ARM CPU** complex, and advanced **ISP core** to deliver significantly higher compute performance for advanced camera features such as always-on HDR photography, HDR video, video stabilization, object tracking (tap-to-track), and other compute-intensive image processing applications. In addition, the software stack for Chimera architecture includes a set of well-defined APIs and plugins that enable OEM and third party application developers to easily harness the processing power of the engine to develop unique and differentiated camera applications.

Why HDR imaging?

High Dynamic Range imaging is a set of techniques that allow the capture of a wider range of light intensities from a scene that has both brightly lit areas, and areas that are not well lit or in the shadows. The dynamic light range of real-world scenes typically exceeds a 10,000:1 ratio between the brightest areas and the darkest areas. Capturing the entire dynamic range presented by real-world scenes is challenging and is greatly dependent on the size, sensitivity, and signal quality of the digital camera's CMOS sensor, and the size of each pixel on the sensor. Even the highest-end digital SLR cameras that have large full-frame sensors are able to capture only a limited dynamic range up to 4000:1. Most smartphone cameras use much

[1] http://www.giiresearch.com/report/iii248735-outlook-mobile-phone-cmos-image-sensor-market.html

smaller and lower quality image sensors and are only able to capture a fraction of the dynamic range presented by real-world scenes.

Consider the images in Figure 1 that have a very large dynamic light range with a brightly lit background and a foreground subject that is not well lit. Due to the limited dynamic range of its image sensor, today's smartphone camera is only able to capture either the light range of the dimly lit areas (the image on the left), or the light range in the brightly lit areas (the image on the right) of the scene in a single shot. Increasing the exposure time of the camera captures the details of the subject in the shadow, but it also washes out the areas that are brightly lit, resulting in significant loss of detail. On the other hand, when the exposure is set to capture the details in the brightly lit areas of the scene, the foreground subject is completely under-exposed.

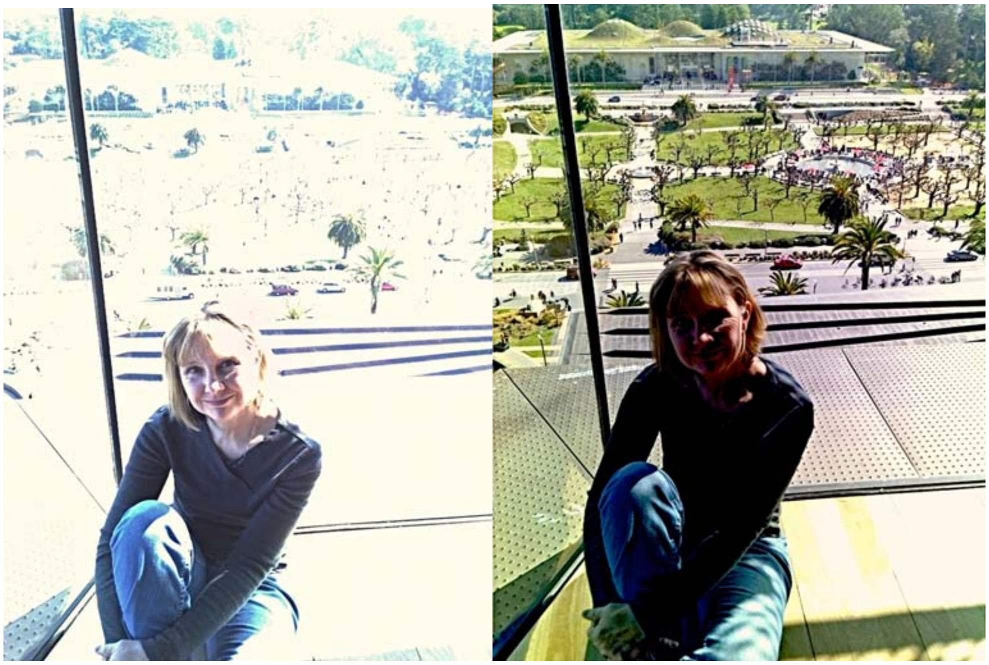

Figure 1 the challenge of capturing high contrast scenes – background washed out (l), or foreground dark (r)

Current HDR Imaging Techniques

Photographers work around the limited dynamic range of cameras by capturing multiple images of the scenes at different exposure levels such that each image captures a portion of the dynamic range presented in the scene. These individual images at different exposure settings are then merged together in photo editing software such as Adobe Photoshop that use complex software algorithms to ensure that the merged HDR image includes most of the dynamic range present in the original scene.

This technique is widely used and its popularity has grown tremendously since the advent of digital cameras and affordable photo editing software tools.

Similarly HDR photos on smartphones are created by taking a high exposure shot and low exposure shot and merging the two shots using software algorithms that are processed by the CPU. The camera architecture of current high end smartphones that is used for advanced camera effects like HDR photos is shown in Figure 2.

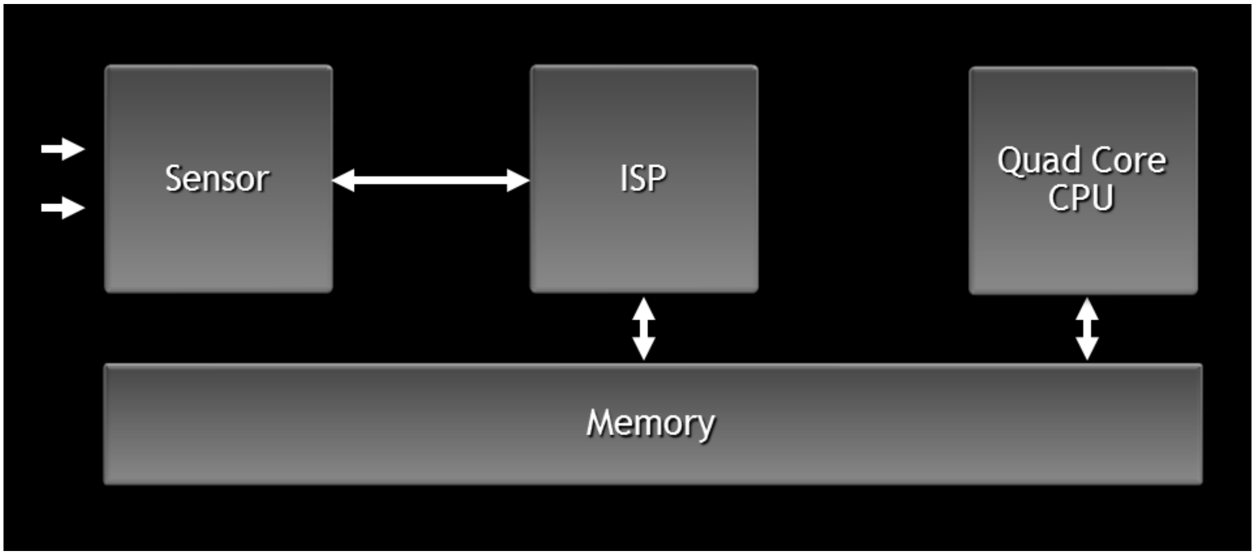

Figure 2 Camera Architecture in current high end mobile devices

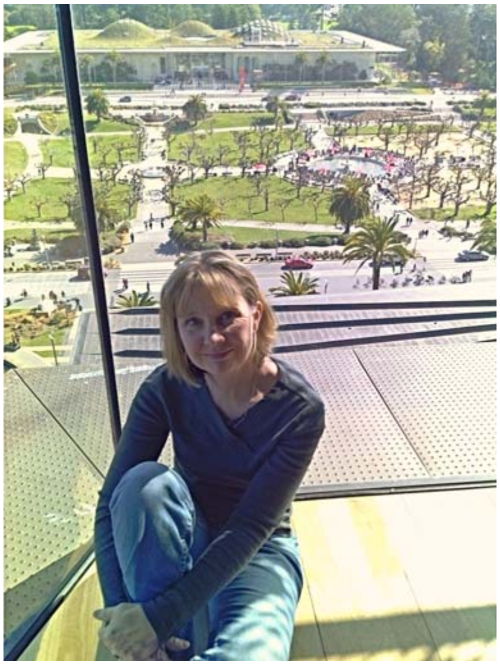

Figure 3 HDR image that captures the dynamic range of bright and dark regions of the scene

Even though HDR images captured on smartphones may offer acceptable image quality as shown in in Figure 3, the time taken to capture the two images at different exposure setting, processing the images through the ISP, and then merge the images using the CPU takes more than two seconds. This not only results in a poor user experience, but also makes it almost impossible to take good quality HDR photos when subjects (such as people, kids, pets) or background objects (such as clouds, waves, waterfalls, traffic) in the frame are in motion.

Consider the image on the left in Figure 4 that was taken with HDR disabled. The subject in the foreground is well lit, the subjects in the background that are in motion are sharp, but the brightly lit sky and the buildings in the background are completely washed out. The same scene captured with HDR mode enabled shows that the dynamic range of light from the foreground subject, the sky, and building are correctly captured. But the HDR image of the background subjects that were in motion show “motion artifacts” (in this case the ghosting on his face), thus ruining the image.

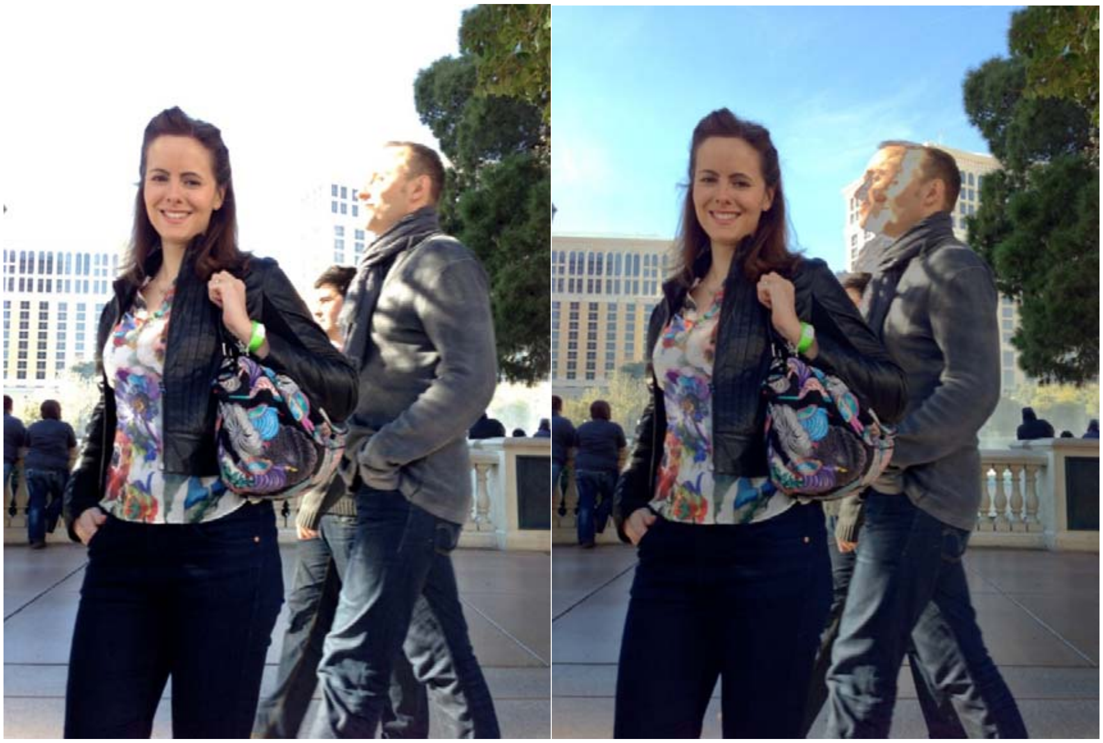

Figure 4 Current Camera architectures are too slow to capture HDR images of subjects in motion

Due to the finite time delay between the low and high exposure shots, the location of objects that are in motion in the scene will not be the same in the captured images from the two shots. Thus when these two images are merged, the resulting HDR image will appear blurry and unusable as seen in the image on the right in Figure 4.

Since smartphones are most often used to shoot candid shots of people or scenes in motion, HDR imaging based on current camera architectures delivers a poor experience and thus is rarely used. In addition, the total time taken to capture, merge, and create the HDR image takes between two to five seconds on current high-end smartphones, also rendering HDR mode useless for other HDR enabled features such as burst HDR, strobe motion HDR, and HDR panoramas.

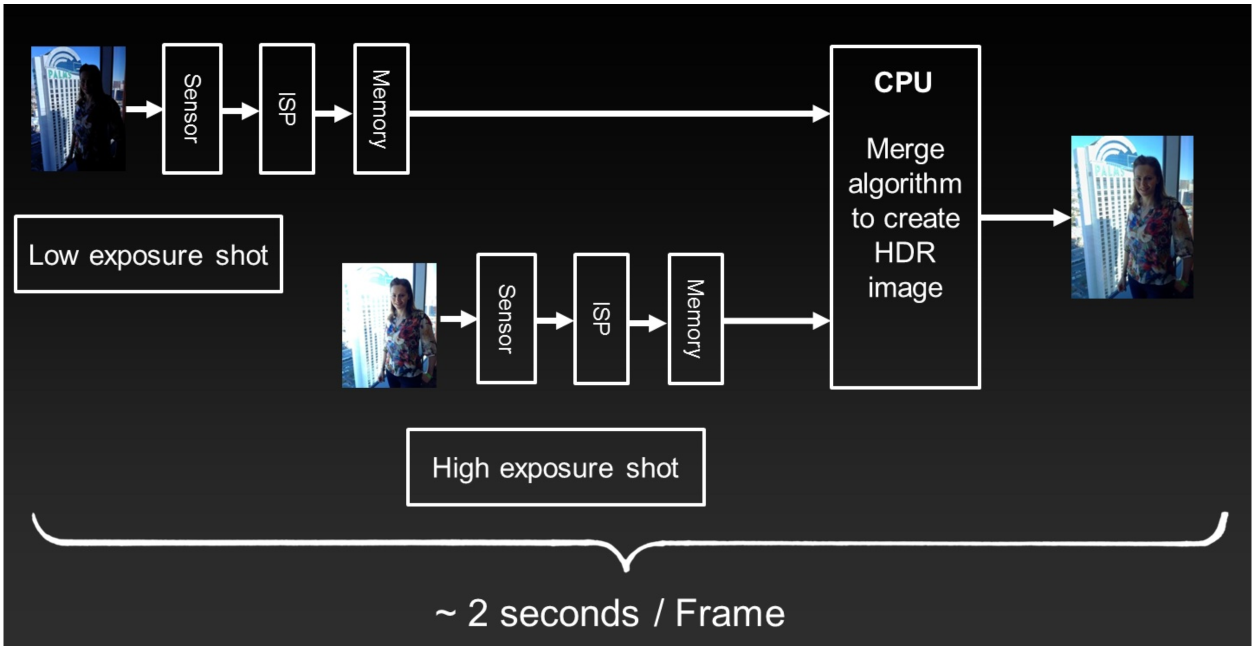

Figure 5 Current smartphones take around two seconds to create HDR photo

Chimera: The NVIDIA Computational Photography Architecture

The NVIDIA® Tegra® 4 Family of mobile processors includes NVIDIA Chimera™ architecture, a proprietary architecture (shown in Fig 6) developed based on NVIDIA's extensive experience in using the processing power of the GPU for compute-intensive applications. The core of this architecture is the NVIDIA Computational Photography Engine that harnesses the parallel processing performance of the custom GPU cores and the quad core ARM CPU to run advanced image processing algorithms at speeds that are orders of magnitude faster than current smartphone camera architectures.

Conventional smartphone camera architectures process data serially by passing it from the image sensor, through the ISP core, CPU, and memory subsystem leading to longer processing times and limited processing capabilities. Chimera architecture virtualizes the image processing pipeline across the CPU, GPU, and ISP cores and delivers significantly higher compute resources and shorter processing times for advanced imaging features such as NVIDIA "one shot" HDR, real-time HDR video, Burst HDR capture, Strobe Motion imaging, and Object Tracking.

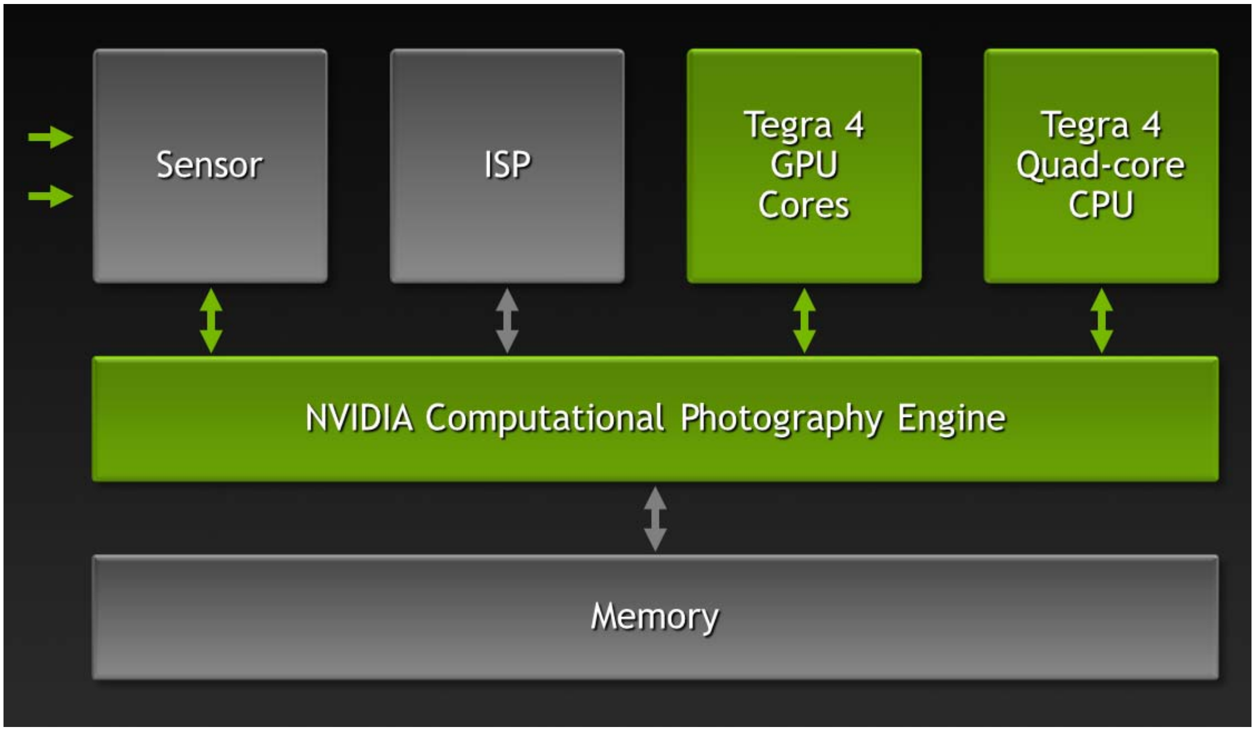

Figure 6 Chimera: The NVIDIA Computational Photography Architecture

NVIDIA "One Shot" HDR Imaging

The Computational Engine employs algorithms to construct the final HDR image by harnessing the massive processing power of the GPU, quad core CPU and the ISP cores.

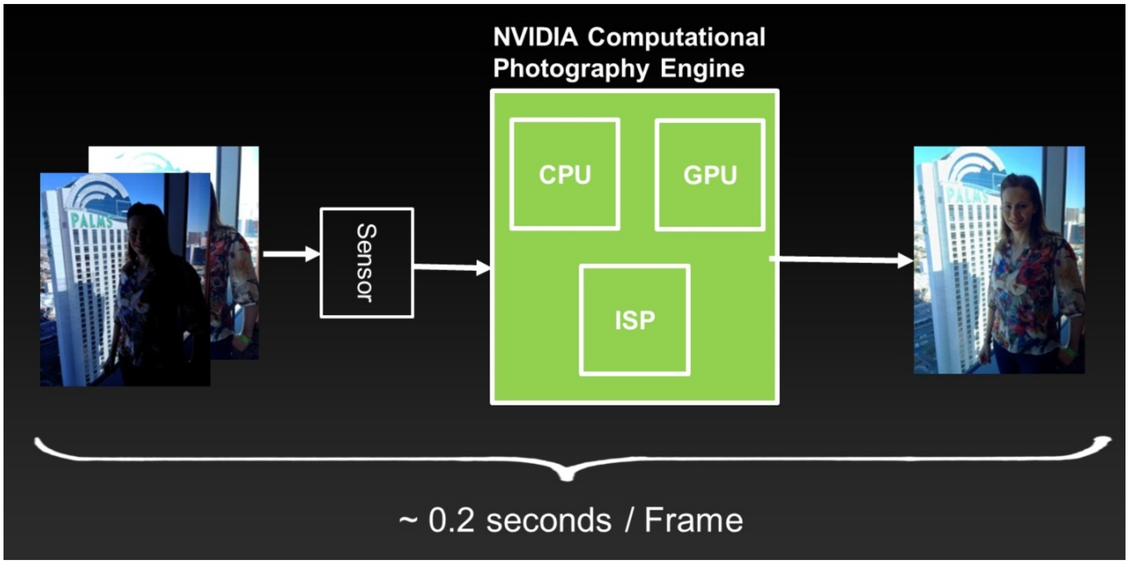

Figure 7 Up to ten times faster HDR imaging using Chimera on Tegra 4

Combining the latest advances in image sensor technology and the compute horsepower of NVIDIA® Chimera™ architecture, [NVIDIA Tegra® 4 processor enables "one shot" HDR imaging that is up to ten times faster than HDR imaging capabilities of current high-end smartphones. NVIDIA Tegra 4-powered smartphones and tablets [will be the very first mobile devices able to capture real-time HDR videos at 1080p 30 fps and deliver advanced HDR features such as burst HDR mode and flash-enabled HDR modes. HDR mode in smartphones no longer needs to be a special mode that is enabled only for some shots. The NVIDIA Computational Photography Engine in the NVIDIA Tegra 4 processor delivers the performance, speed, and convenience for the "Always On" HDR experience on mobile devices.

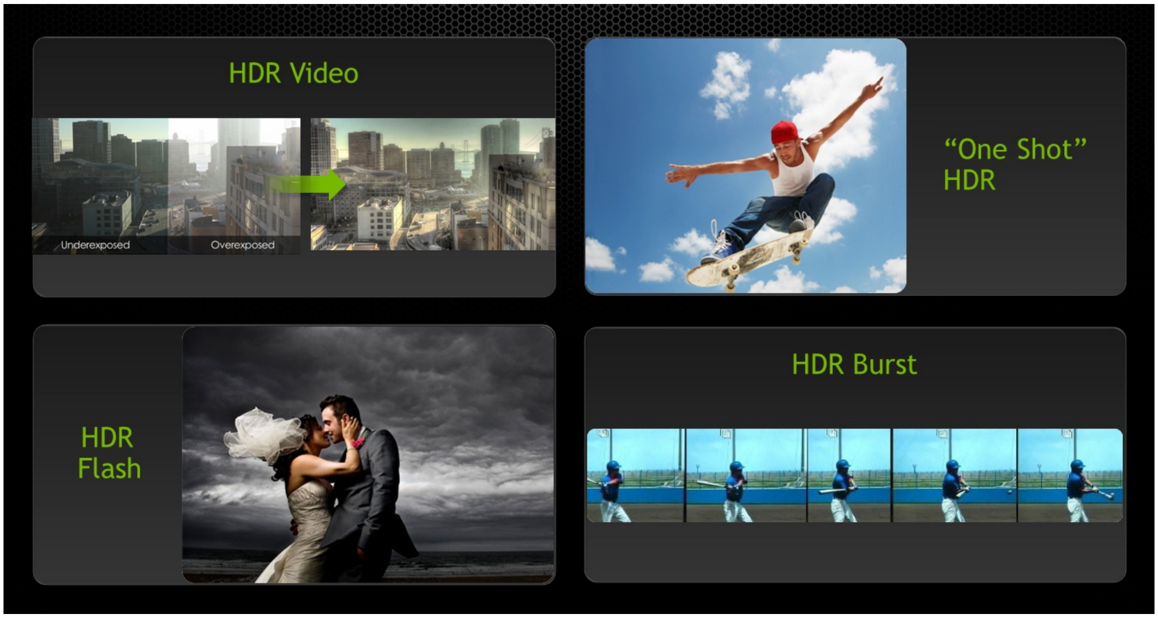

Figure 8 NVIDIA "Always On" HDR Experiences

A Programmable and Customizable architecture

The NVIDIA® Chimera™ architecture is easily programmable and includes well-defined APIs. OEMs and third party camera application vendors can develop plugins that interface with Chimera to implement custom camera features and custom algorithms for differentiated performance.

The camera application interfaces with the underlying hardware through the public Camera APIs to run basic camera functions. In addition, Chimera architecture is accessed through plugins developed by NVIDIA and/or partners to enable advanced features such as real-time HDR, video stabilization, 3D image reconstruction and image tracking.

Depending on the task request presented, Chimera architecture uses the appropriate hardware level APIs such as OpenGL for GPU, Neon for CPU, and core Camera calls to harness the GPU, CPU, and ISP cores to process the workload presented by the camera application.

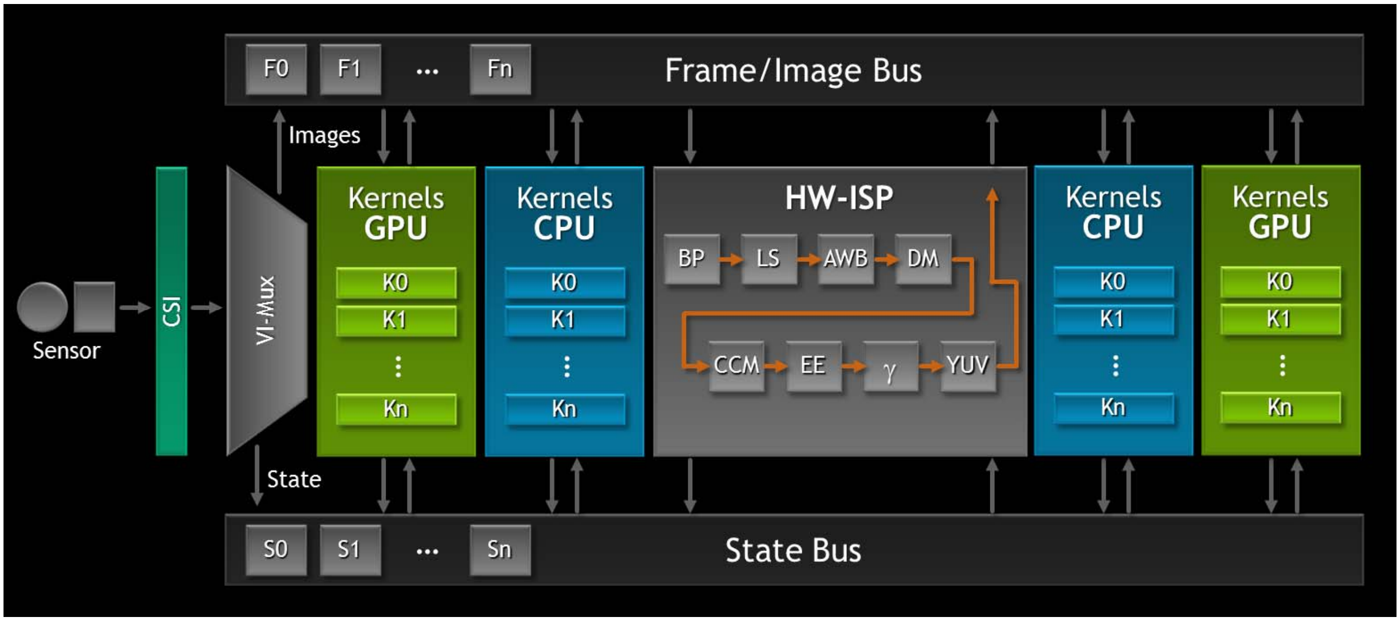

Figure 9 Chimera Process Flow

Chimera architecture is a software framework (Figure 7) that abstracts NVIDIA Tegra4®'s GPU, quad core CPU and ISP cores. The flow of data in the engine is through two information busses, one that carries image or frame data and another that carries state information associated with each frame. The framework ensures efficient propagation from one computational block to another so that programmers do not need to manage this process themselves.

The second key element is the notion of a series of computational elements chained together in a specific order: GPU, CPU, ISP, CPU, GPU. An application developer may decide to choose all or part of the computational chain. The GPU and CPU computational element run "native" code, called kernels (K0… Kn in the diagram) provided by the application developer.

For example, let's say you wanted to perform some type noise reduction on raw Bayer data coming from the camera sensor. Using the GPU you would create a kernel in the OpenGL-ES fragment shader language. In this example you get called with per pixel access to the Bayer data in a specific format so that you can easily write your shader without having to worry about how the data arrived or even where it goes.

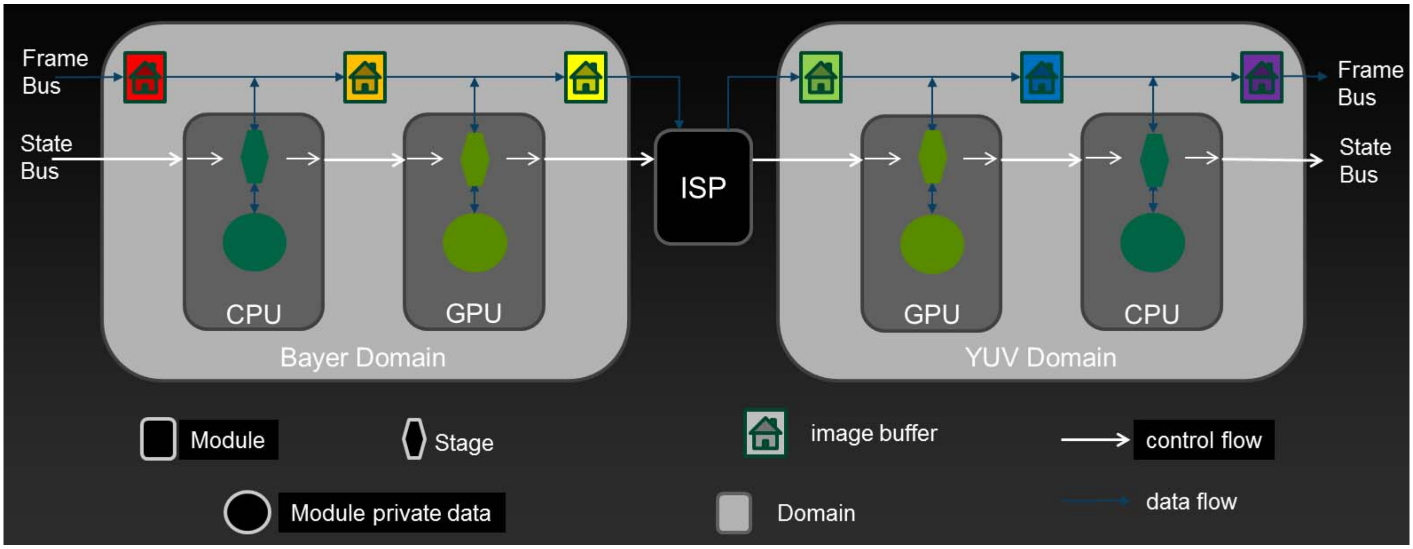

Figure 10 A deeper look at the Chimera Programming Model

Figure 8 presents a deeper view of the programming model of the NVIDIA® Chimera™ architecture. Each kernel (seen in figure 7) is embedded into a "module" as a stage in the module. The reason we called it a stage will become apparent. The module also holds local state information called "module private data". This is intermediate data used by the stage during computation. Module stages get state data coming from the "state bus" associated with the incoming frame. The frame state data and the frame are transitory since every new frame will contain new state data, such as exposure value (EV) or frame size.

The stage runs to completion and takes as long as the programmer decides. The module also contains debugging and timing information necessary to tune and perfect modules. Given the pipeline nature of the Chimera architecture, each stage is running on separate frames in parallel allowing the developer to get the most out of the architecture.

Now let's look at how this programming model is used to implement the "Always On" HDR feature.

In this case only the GPU and CPU are used once, in the beginning of the pipe and at the end respectively. This illustrates the flexibility of the architecture allowing the developer to pick and choose how much processing they want to use. To implement "Always On" HDR, the programmer would use the GPU to perform HDR processing, conditioning and reconstructing Bayer data before sending it to the ISP HW for "normal" ISP processing. The ISP outputs a YUV image that is passed onto the tone mapper that runs on the CPU accelerated using C++ and ARM Neon instructions. Again all the programmer had to do was set up the "modules" and write the kernel stages.

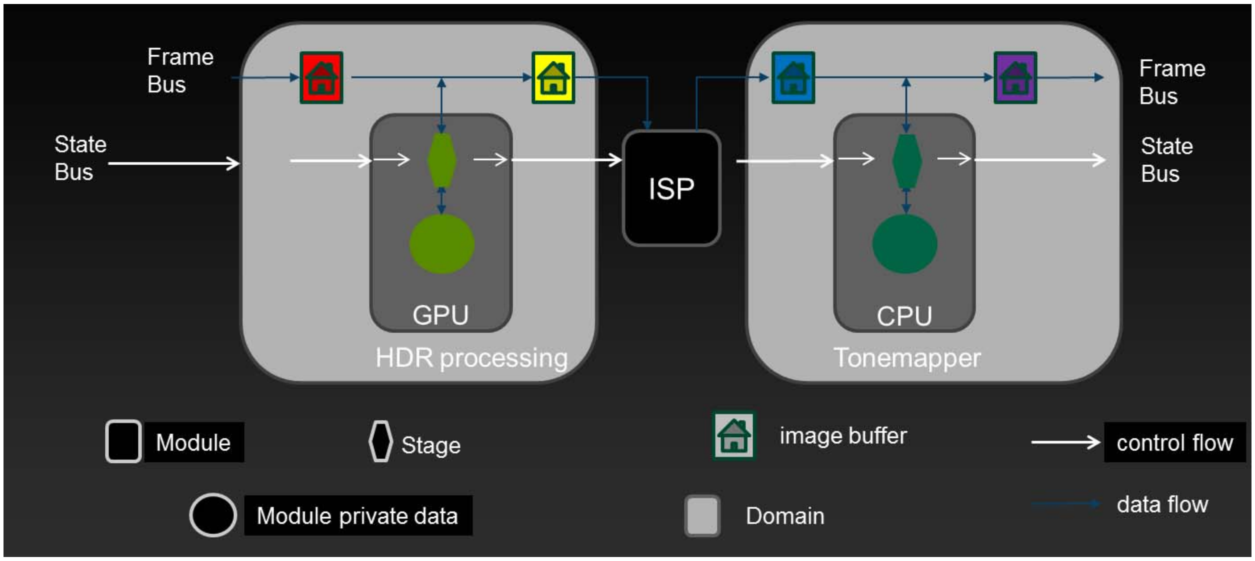

Figure 11 Chimera Process Flow for "Always On" HDR

In many situations, the programmer may want to stack kernels on top of one another each performing some element of processing. This is done by adding "stages" to any given module (Figure 10). Each module can contain zero or more stages. Within the module they are executed in the order provided to the module object. Since the module private data is shared stages communicate state between themselves using the module private data. The module can export its private data onto the state bus at behest of the module developer so that modules can share data from one processing module to the next.

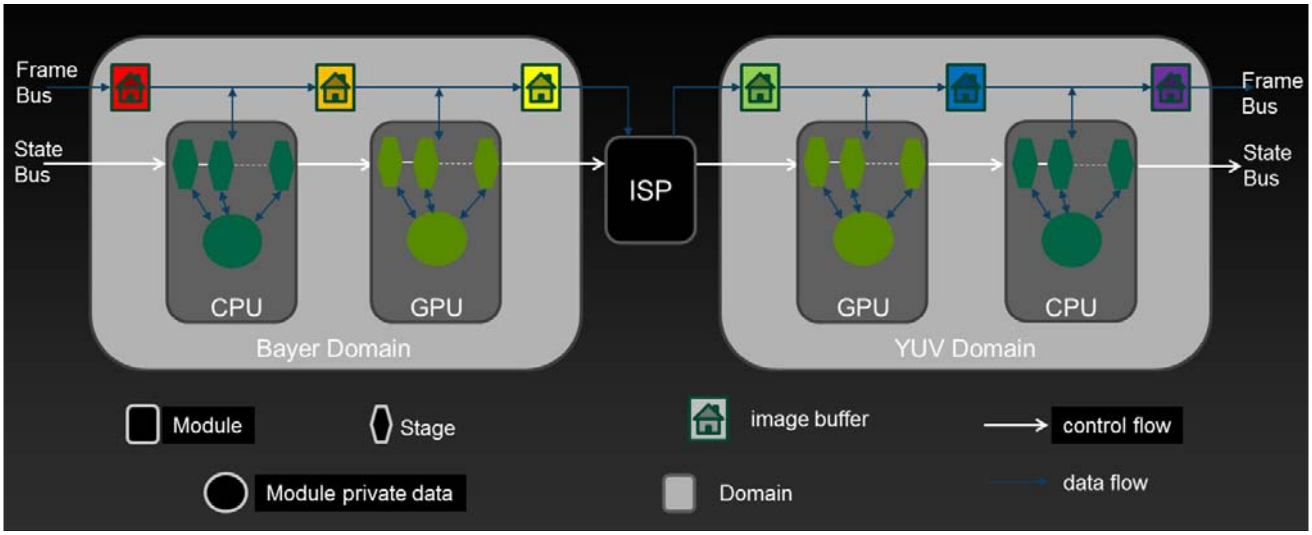

Figure 12 Multiple kernels can be stacked on top of each other by adding stages to the process flow

Conclusion

Capturing memories through taking photos or recording video continues to be a key activity on mobile devices with usage still on the rise. To put things into perspective, 10% of all photos ever taken, were taken in 2011 – all due to the convenience of personal devices and ease of capturing pictures. The next wave in smartphone camera use will be fueled by computational photography. This will result in better images, capturing what you intended based and what your eyes saw, thereby capturing forever that fleeting moment you'll never experience again.

The NVIDIA® Chimera™ architecture harnesses the processing power built into the GPU, CPU, and ISP, enabling new features and capabilities such as Always-on HDR for truly stunning photos.

This is just the first step in a wide range of new features and photographic capabilities on the verge of coming to the next generation of smartphones and mobile devices.

Document Revision History

Revision Number	Notes

Notice

ALL INFORMATION PROVIDED IN THIS WHITE PAPER, INCLUDING COMMENTARY, OPINION, NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, “MATERIALS”) ARE BEING PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. NVIDIA Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

Trademarks

NVIDIA, the NVIDIA logo, Chimera and Tegra, are trademarks or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright