Mobile 3D Vision – Algorithm & Platform Challenges

Kyle Rupnow[#], Yun Liang[#], Dongbo Min[#], Minh Do[*], Deming Chen[*]

[#]*Advanced Digital Sciences Center*
{k.rupnow,eric.liang,dongbo}@adsc.com.sg

[*]*University of Illinois*
{minhdo,dchen}@illinois.edu

I. Introduction

Rich media mobile devices have experienced significant growth in recent years − cell phones have become mobile computing platforms; personal entertainment devices have rich interactivity and high quality displays; and tablets and portable computers have increasing demands in display, interactivity complexity and computational demand. In all of these devices, there is unprecedented computational density, but, users continue to demand good battery life.

One key technology employed throughout these devices is high quality interactive displays. In the recent evolution of mobile devices, one consistent trend is improved resolution, contrast and interactivity of displays. These displays enable more richly interactive applications through higher quality display of content and more accurate (often touch based) interaction with display elements. Recently released devices such as the Nintendo 3DS and the LG Optimus 3D foretell the next major step in interactive displays.

Three dimensional displays offer the potential for even richer display and interaction with users through the use of depth to highlight display elements, and detection and use of depth for augmented reality. Mobile devices are an attractive option for introduction of interactive 3D displays – mobile devices, particularly cell phones, are typically single user which limits the complexity of producing the 3D illusion and interactivity.

It is a major challenge to produce 3D content in real-time with high quality and energy efficiency. Therefore, current mobile 3D devices primarily support display of pre-produced 3D content. However, future mobile devices will need to support live capture and real-time display of 3D media. Teleconferencing applications will include capture and display of 3D video; augmented reality applications will capture images, detect and highlight objects, and use the 3D display to highlight elements or more accurately correspond the screen view to the actual view; camera applications will allow dynamic removal of objects and capture of 3D video for upload and later playback. In all of these applications, the extra application complexity significantly increases computational demand, but increased computational performance cannot come at the cost of energy consumption. Even as we increase the capabilities of devices, we continue to demand equal or improved battery life.

In this paper, we discuss the algorithmic challenges of generating high quality depth map information in real-time on a mobile platform and the hardware platform challenges of making these algorithms sufficiently fast and energy efficient on the mobile platform.

II. Algorithm Challenges

Due to the constraints of mobile devices, there are many challenges in designing algorithms to incorporate real-time 3D video computation into mobile devices.

1) *Image sensor for mobile device*

The CMOS (Complementary metal oxide semiconductor) sensors typically used in mobile devices commonly have noise, edge blurring, and color distortion between multiple sensors, all of which adds complexity to algorithms in order to retain final image quality [1][2]. Furthermore, mobile devices also have camera movement, thus requiring additional computation for image stabilization.

2) *3D camera set-ups for mobile device*

There are two main camera setups that could be used for capturing 3D video on mobile devices. First, two (or more) color image sensors can be embedded on mobile devices, so the sensors are pre-calibrated in both photometric and geometric manners, and the device manufactured to rectify the sensors (therefore, corresponding points are always on the same horizontal line).

In addition, a single mobile device with single image sensor can be used in a static environment to generate depth information; normally called a Structure-from-Motion (SfM) framework [3]. Instead of capturing 3D video with multiple cameras, we obtain 3D video by taking temporally-neighboring images with a single (moving) camera when scene is static. There is significant prior work in SfM frameworks for scene modeling and augmented reality [3], but all require significant computational complexity.

3) *3D image analysis*

A key underlying technology for real-time 3D video applications is the stereo matching problem, which compares a set of two or more images taken from slightly different viewpoints (e.g. two spatially separated cameras in the same mobile device). Given camera calibration parameters, corresponding points between multiple images, estimated by stereo matching, are converted into depth value between camera and object. In general, stereo matching presents a challenging problem due to an image ambiguity [4]. Many algorithms have been proposed by using several cost aggregation methods [5] and global optimization techniques [6], but their performance is still far from practical due to noise, occlusion, lack of texture, real-time computation requirement, and memory consumption. Real-time stereo matching algorithms that are optimized to specific hardware environment have been proposed [7], but most of them are based on the workstation. However, mobile stereo matching must also consider power/energy consumption, memory use, and latency limitations. Considering these constraints, stereo matching algorithms should be re-designed to find the best trade-off among accuracy, complexity, and computing resource.

The depth accuracy, spatial/temporal resolution and disparity precision all affect algorithm complexity and memory use. Some applications for mobile 3D devices will require dense, high-quality depth maps, while other applications will only require coarsely segmented depth maps. The required quality is related to human perception, and is thus (at least partially) subjective. Thus, depth map algorithms must be developed to meet each application’s power, performance and display quality goals.

4) *3D Depth sensing technologies*

In addition to the color image based techniques, active depth sensors may be used in mobile environments. Recently, depth sensors such as Time-of-Flight (ToF) camera and Microsoft Kinect have been widely used in computer vision research fields. Since it provides a low resolution depth map at a high frame rate, it can be used in a dynamic environment without the time-consuming stereo matching. Computer vision algorithms can improve the resolution of the depth map with sufficient spatial/temporal information from video-rate depth video and corresponding color video [8].

5) *3D Depth video compression*

We may need to transmit the depth video, estimated by stereo matching or depth cameras, over the wireless channel, e.g. on the 3D immersive tele-conferencing. Recently, many algorithms have been proposed to compress the depth video in a compatible manner to conventional video coding such as H.264, but these algorithms focus on high bitrate coding for 3DTV. Thus, in the mobile 3D applications, the extremely low bit rate depth video coding should be addressed in an efficient manner.

III. Platform Challenges

In order for real-time 3D video to have sufficient quality, algorithms must retain high quality output, but without increased power/energy consumption. Meanwhile, there are platform-level challenges for achieving the same goal. There are many components to mobile 3D video applications including video encoding/decoding, stereo matching, image rendering, depth map enhancement and object recognition – each of which may have multiple program phases. In order to meet power, energy and performance goals, we must design a heterogeneous platform well suited to the workload applications, accurately model applications' behaviour, schedule applications' execution, and use both static and dynamic information to reduce power/energy consumption.

1) *Heterogeneous platforms.*

For each application and application phase, there are many potential trade-offs in performance, power and energy consumption, depending on how well-suited the computation resource is to the application's computation. CPUs perform well for complex control flow applications, GPUs perform well for highly data parallel floating point applications and FPGAs perform well with irregular operand sizes, customized modules, detailed pipelining, and interleaving of computation. Recently, there has been a proliferation of commercial heterogeneous platforms notably including NVidia, Intel, and Xilinx. Mobile 3D vision applications require strict real-time computation constraints and significant memory consumption. Thus, the platform must be designed with low-latency communication and efficient mechanisms to share memory between the heterogeneous resources, and the HW/SW co-design problem must carefully consider migration and communication overheads.

2) *Memory design.*

For embedded processor design, memory hierarchy plays a critical role in terms of power consumption. It is shown that up to 50% of a microprocessor's energy is consumed by memory hierarchy [9, 10]. Thus, careful memory optimization for 3D vision could significantly improve the power consumption. Computer vision applications commonly have predictable memory access patterns (e.g. streaming data). For this access pattern, with little temporal locality, it is better for energy consumption to have streaming buffers instead of a cache that consumes power without improving performance. Instead of caching, memory pre-fetching may be used to improve the memory performance and reduce the total energy consumption. For non-stream data, caches minimize the off-chip memory bandwidth and thus save energy. Different memory requirements and access patterns are observed for within and across computer vision applications. Thus, the best cache configuration for power/performance will vary depending on the heterogeneous resource used, requiring platform support for a variety of cache configurations including streaming, scratchpads, and traditional caches. Together, the wide range of behaviour and requirements makes dynamic intra and inter task cache reconfiguration important to reduce energy consumption.

3) *Power& performance modeling.*

Accurate power and performance modelling plays a critical role in low-power design. Embedded system designers must estimate the power/energy budget for each application scenario. For a heterogeneous platform, each resource type may require different power and performance models to accurately represent implementation technology, computation latency, and availability of hardware features such as clock gating or DVFS. These estimations are critical to the design process – incorrect workload estimations during platform design may result over or under-provisioned computation resources and battery system.

4) *Task scheduling.*

Based on task dependence graphs, performance, power and energy estimates, tasks are distributed among the platform resources. In simple systems, tasks can be statically assigned to resources. However, workloads will include changing mixtures of applications, which will require dynamic task scheduling to meet system goals. Task scheduling is a complex trade-off between performance, instantaneous power and total energy consumption. For example, the OS may delay beginning one task's execution in order to wait for a better resource, or begin execution immediately on a less-optimal resource [12]. To improve battery life, the OS may use lower instantaneous power (to reduce heating and battery load) at the cost of performance and total energy consumption. In heterogeneous platforms, each task may have multiple implementations with different power, energy, computation resource requirements and computation latency [13]. In GPU and FPGA platforms, there may even be multiple alternative implementations that use different amounts of GPU or FPGA resources.

5) *DVFS and clock gating*

As mentioned above, DVFS and clock gating can significantly improve energy consumption. Although not currently available in high performance GPUs, DVFS with embedded GPUs guided by workload characterization and prediction can be used to achieve the target throughput (real-time computation). GPU code may be designed and optimized for different amounts of physical resources; therefore, the system may scale the voltage, frequency and number of turned-on cores to match the performance and resource requirements of the current application. Similarly, FPGAs may use dual supply voltages for power optimization [11], or clock gating to turn off unused portions of the FPGA fabric.

IV. Conclusion

Mobile 3D vision is an emerging application area with great opportunity to improve the interactivity and quality of mobile displays. However, there are significant challenges in developing algorithms and platforms that can effectively meet the power and performance constraints of next generation applications. In this paper, we outline the main challenges at the algorithmic and platform levels for introducing mobile real-time 3D video to mobile platforms.

V. References

[1] W.-H. Cho and T.-C. Kim, "Image enhancement technique using color and edge features for mobile systems," Proc. SPIE, 2011.

[2] P.-S. Tsai, C.-K. Liang, T.-H. Huang, and H.-H. Chen, "Image Enhancement for Backlight-Scaled TFT-LCD Displays," IEEE Trans. on Circuits and Systems for Video Technology, 2009.

[3] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision," Cambridge University Press, 2003.

[4] http://vision.middlebury.edu/stereo

[5] D. Min and K. Sohn, "Cost Aggregation and Occlusion Handling With WLS in Stereo Matching," IEEE Trans. on Image Processing, vol. 17, no. 8, pp. 1431-1442, 2008.

[6] R. Szeliski et. Al., "A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors," IEEE Transactions on Pattern Analysis and Machine Intelligence, Jun. 2008.

[7] C.-K. Liang, C.-C. Cheng, Y.-C. Lai, H. H. Chen, and L.-G. Chen, Hardware-efficient belief propagation, IEEE Trans. CSVT, vol. 21, no. 5, pp. 525 - 537, May. 2011.

[8] D. Min, J. Lu, and M. N. Do, "Depth Video Enhancement Based on Joint Global Mode Filtering," IEEE Trans. on Image Processing. (submitted)

[9] A. Malik, B.Moyer and D.Cermak, "A Low Power Unified Cache Architecture Providing Power and Performance Flexibility," ISLPED, June 2000.

[10] S. Segars. "Low power design techniques for microprocessors," Int. Solid-State Circuits Conf. Tutorial, 2001.

[11] D. Chen et. al. "Technology Mapping and Clustering for FPGA Architectures with Dual Supply Voltages". IEEE TCAD Vol 29, No. 11, 2010.

[12] HK Tang, K Rupnow, P Ramanathan, K Compton "Dynamic Binding and Scheduling of Firm-Deadline Tasks on Heterogeneous Compute Resources", in RTCSA, 2010 pp. 275-280

[13] K Rupnow, K Compton, "SPY vs. SLY: Run-time Thread-Scheduler Aware Reconfigurable Hardware Allocators", in FPT 2009, pp. 353-356