Resilience Characterization of a Vision Analytics Application Under Varying Degrees of Approximation

Radha Venkatagiri† Karthik Swaminathan‡ Chung-Ching Lin‡ Liang Wang¶
Alper Buyuktosunoglu‡ Pradip Bose‡ Sarita Adve†
†University of Illinois at Urbana-Champaign ‡IBM Research ¶University of Virginia
†{venktgr2, sadve}@illinois.edu ‡{kvswamin, cclin, alperb, pbose}@us.ibm.com ¶lw2aw@virginia.edu

Introduction

The rapid growth in real-time edge computing requires sustained advances in energy-efficient, yet progressively higher performance embedded systems. An example application is computing aboard unmanned aerial vehicles (UAVs) engaged in surveillance, rescue and recovery missions. Video Summarization (VS) [7] refers to a class of key algorithms in the UAV-computing context. A basic function within such a summarizer is the creation of a single, stitched panorama of the landscape from input video streams. Resilience to soft errors under low voltage (low power) operation and at high altitude is a key attribute that is required of such embedded computing.

Approximate Computing [4] is an increasingly popular technique to trade off loss in output quality for system benefits such as gains in performance or energy efficiency. In this work, we apply three distinct software approximation techniques to a state-of-the-art video summarization algorithm which yield significant energy savings (up to 68%) while achieving desired performance targets and suffering only modest losses in the final output quality. To understand the interactions of software approximations with hardware resiliency, we undertake error injections experiments on the baseline as well as the approximate versions of the video summarization algorithm and study their resulting resiliency profiles. We show that the approximations chosen do not significantly degrade the underlying application's resiliency in the presence of soft errors.

Video Summarization Application Overview

The video summarization application (henceforth referred to simply as VS algorithm) takes an input video [3] captured by moving cameras and generates panoramas that provide a global view of the landscape. While a full detailed description of the algorithm is provided in [2], Figure 1 illustrates one of the fundamental functions performed by the VS algorithm – the comparison, transformation and stitching of two successive images from the input video. The algorithm first identifies key regions of interest (*key points*) within each image and then looks for matching key points between the images to identify potential common areas. If enough matching key points are found, the algorithm applies transformations (homography) to the two images so that they are aligned correctly and have the same scale, lighting, perspective etc., before proceeding to stitch them together. If enough matches are not found between the two images, one of them is dropped and the procedure is repeated with a subsequent image from the input stream.

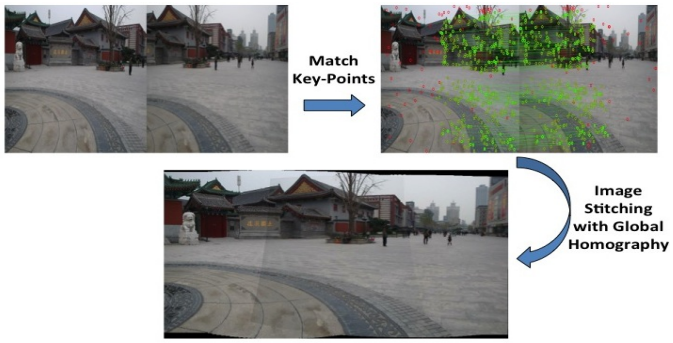

Figure 1: Simple example of stitching two images

The three approximate versions of the VS algorithm studied are briefly described below :
(1) Random Frame Dropping (VS_RFD): This approximation randomly drops 10% of the input frames.
(2) Key Point Down Sampling (VS_KDS): The number of key points used in matching two images are down-sampled.
(3) Simple Matching (VS_SM): In the VS algorithm, two key points (in successive images) are considered a good match based on a thresholded nearest neighbor distance calculation. In VS_SM, the match between two key points is determined using a faster, but less accurate technique.

Further details about the approximate algorithms, including the performance/energy benefits and quality of final output, are provided in [5].

Resiliency of Video Summarization Applications

Towards our goal to evaluate the application-level resilience of the VS algorithm and its different approximate

This work was supported in part by the Defense Advanced Research Projects Agency (Contract No. HR0011-13-C-0022), by the National Science Foundation (under Grant CCF-1320941) and by the Center for Future Architectures Research (C-FAR), one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

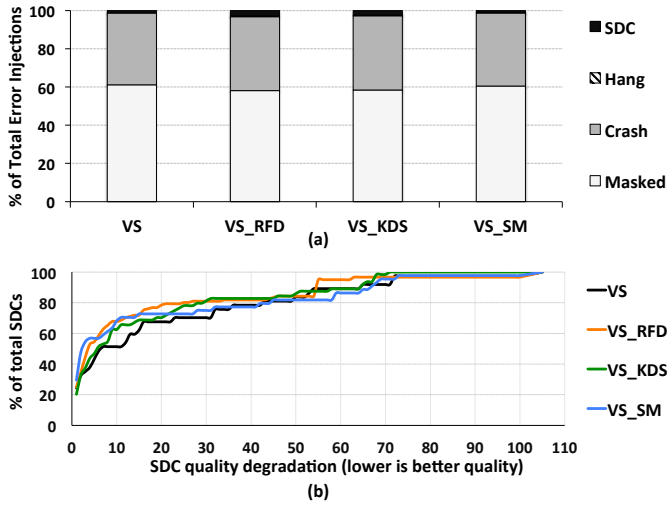

Figure 2: Graph (a) shows the resiliency profiles (SDC, hang, crash and masked rates) for the different VS algorithms. Graph (b) shows the quality degradation distribution for the SDCs across the different VS algorithms.

versions in the presence of hardware transient errors, we use a tool called Application Fault Injection (AFI) to randomly inject 2000 single bit errors (one at a time) in the general purpose (GPR) and floating point (FPR) architectural registers (1000 each for GPR and FPR), and observe the rate of masked, crash, hang, and silent data corruption (SDC) outcomes [8].

Figure 2(a) shows the outcomes for the error injection experiments in the GPRs for the different VS algorithms (FPR error injections produce a very high masking rate, >99.5%, and hence, for brevity, we do not show those results). The crash and hang rates of the approximate algorithms are very similar to those of the baseline VS algorithm. The silent data corruption (SDC) rates increase from 1% (VS) to 3% and 2.5% for VS_RFD and VS_KDS respectively. This increase in SDCs is due to the loss of redundancy in the approximate algorithms. Some errors that are masked in the baseline algorithm – due to redundant subsequent frames being stitched on top of the erroneous portions of the image – are now exposed.

Analyzing SDC Quality

SDCs corrupt the program output without any easily detectable software symptoms [1], and hence protection against them normally incurs high overheads. The cost of protection can be minimized if we can quantify the output quality degradation produced by different SDCs and then only protect those SDCs whose quality degradation is deemed unacceptable to the application or user [6]. Hence, it is important to understand the quality of the SDCs produced along with the SDC rate.

To measure the quality degradation of a corrupted output image with respect to the golden error-free output image, we first transform the images to remove minor variations in color, lighting, perspective etc. We do this (instead of directly using a quality metric like PSNR) because these types of erroneous outputs may be tolerable (while still providing valuable information) to the human beings observing the end panorama for surveillance and tracking. Once the two images are transformed, the deviation across them is measured by the relative difference (in percentage) in the L2 norms of the two images and is stored as the quality degradation metric (hereby referred to as QD) of the SDC.

Figure 2(b) plots the SDCs generated by the different video summarization algorithms according to their QD. The graph shows that the overall trend for the SDC quality for the VS and its approximate algorithms are very similar. The approximations do not fundamentally change the quality of the SDCs produced. Another trend seen is that many of the SDCs produced by the VS algorithms are relatively benign. For example, ∼50% of the SDCs across all the algorithms produce a quality degradation of 5% or less. These SDCs need not be protected if a quality degradation of 5% is acceptable to the user. Hence, although approximating the VS algorithm minimally changes its resiliency profile by modestly increasing the number of SDCs generated, this is offset by the fact that a large percentage of these SDCs may be tolerable and hence the cost of protecting them is low.

Conclusion

In this work we study a state-of-the-art video summarization application that serves as a representative emerging workload for the domain of real time edge computing. We further examine three different approximation techniques to improve the power and performance efficiency of the workload. A detailed resiliency study of the application as well as its approximate versions show that the approximations do not degrade the resiliency of the baseline algorithm. Thus, we conclude that it is possible to realize safe, yet efficient approximations for this state-of-the-art video summarization algorithm from the point of view of performance, energy and reliability.

References

[1] S. K. S. Hari *et al.*, "mSWAT: Low-cost Hardware Fault Detection and Diagnosis for Multicore Systems," in *MICRO*, 2009.

[2] C. Lin *et al.*, "Moving camera analytics: Emerging scenarios, challenges, and applications," *IBM JRD*, 2015.

[3] S. Oh *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *CVPR*, 2011.

[4] S. Sidiroglou-Douskos *et al.*, "Managing performance vs. accuracy trade-offs with loop perforation," in *FSE*, 2011.

[5] K. Swaminathan *et al.*, "A case for approximate computing in real-time mobile cognition," in *Workshop on Approximate Computing Across the System Stack (WACAS)*, 2015.

[6] R. Venkatagiri *et al.*, "Approxilyzer: Towards a systematic framework for instruction-level approximate computing and its application to hardware resiliency," *To appear in MICRO*, 2016.

[7] R. Viguier *et al.*, "Resilient mobile cognition: Algorithms, innovations, and architectures," in *ICCD*, 2015.

[8] L. Wang *et al.*, "Power-efficient embedded processing with resilience and real-time constraints," in *ISLPED*, 2015.