Implementing Approximate Mobile Computing

Octavian Machidon, Tine Fajfar, Veljko Pejović
octavian.machidon@fri.uni-lj.si,tf5442@student.uni-lj.si,veljko.pejovic@fri.uni-lj.si
Faculty of Computer and Information Science, University of Ljubljana, Slovenia

Abstract

While approximate computing has shown potential to reach an acceptable trade-off between result accuracy and computational cost in desktop and server environments, its application in the mobile computing field is still lagging. In this position paper we first identify opportunities for approximate mobile computing (AMC), by observing how users' expectations with respect to the result accuracy vary as the context in which a mobile application is used varies. We assess the energy savings enabled by closing this gap between the expected and the actually delivered result quality, propose a general framework for AMC, and discuss a road towards its realization.

1 Approximate Mobile Computing

Approximate Computing Techniques (ACTs) have already been demonstrated on various levels of computer architecture, from hardware [3] to compiler-level optimizations [5] that sacrifice result accuracy for energy efficiency or reduced computational time [2][9]. However, approximate computing faces important challenges preventing its applicability to mobile computing, a domain that could benefit from ACTs due to the context-dependent mobile user requirements which offer the perfect occasions for adaptable approximations [7].

Today the technological advances paved the road for Approximate Mobile Computing (AMC) since the hardware capabilities of the latest mobile devices allow very complex on-device computation (e.g. running deep learning algorithms). Moreover, the growing popularity of mobile personal assistants (e.g. Google Assistant) opens up opportunities for inexact computation (they answer user queries where usually there is no ”golden” answer). Finally, the increased mobile device usage for a wider range of tasks over longer periods of time foster a better understanding of users' expectations from mobile computation.

We aim at making AMC a reality by answering key questions: how to enable approximation on mobile devices, how to infer a user's context-dependent result accuracy expectations, and how to adjust the approximation so that the expectations are met in the most resource-efficient way.

2 Use case - mobile video decoding

We hypothesise that in the mobile domain, a user's real-time context impacts his/her accuracy expectations. Furthermore, we assume that the context could be sensed and that the accuracy expectations could be inferred as the application is used. Finally, we expect that the accuracy of mobile computation could be, perhaps via ACTs, adjusted in real-time and that such adjustment might reduce resource consumption.

We conduct a preliminary investigation of some of the above hypotheses with a video playback application we have modified to capture users' viewing preferences. In our study 22 subjects each watched three short videos while in different mobility states (sitting, walking, running and riding in a car), and we monitored their satisfaction with the video resolution in each scenario (they could change it during playback in the range: 144p, 240p, 360p, 480p, 720p and 1080p). The experiments were carried out on a Samsung Galaxy S III Android smartphone and we used a Monsoon power monitor to measure the current consumption in each resolution.

The results the average current consumption in each resolution together with the standard deviation of the measurements are shown in Figure 1. Due to the similarity of current consumption for the first three resolutions, we divided the resolutions in two groups: low resolutions (144p, 240p and 360p) and high resolutions (480p, 720p and 1080p).

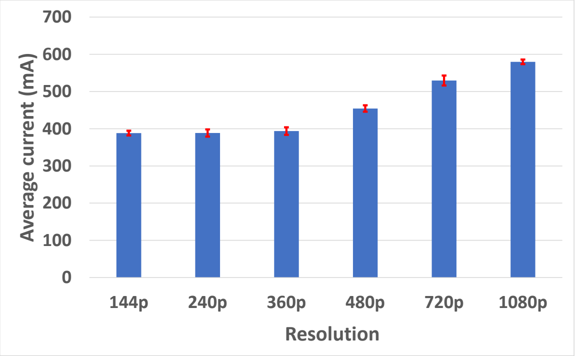

Figure 1. Smartphone current consumption during video playback.

We then analyzed the cases when the user was satisfied with a lower resolution, indicated by not changing the playback to a higher resolution. As evident from Table 1, the users were more likely to find a lower resolution acceptable when engaged in physical activities like *walking* or *running*, than when standing *still* or as a passenger *in a vehicle*. Together with the varying energy requirements of different resolutions (Figure 1), this observation opens opportunities for energy savings by using context-based approximate computing for dynamically scaling the playback resolution according to

Table 1. Number of cases when users found lower resolutions acceptable while watching videos in each mobility state.

Mobility state	Number of cases (%)
Still	28.9
In vehicle	34.9
Walking	53.5
Running	72.5

the user's real-time mobility state. If we downscale the resolution to lower values (e.g. 360p) when the user is running or walking, this extends the battery life of the phone with at least 16.5% and up to 48.6% compared to watching the video in 480p or 1080p, respectively. For a smartphone with a 3000mAh battery, this translates to extending the battery life with at least 1 h and 5 mins and up to 2 h and 31 mins, in the hypothetical case of continuous playback.

3 Towards a general support for AMC

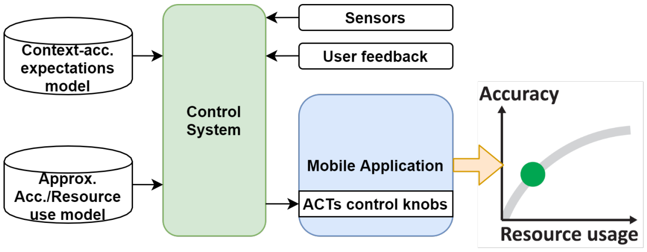

Figure 2. Generic architecture supporting Approximate Mobile Computing.

Video playback is only one of the applications for which AMC is suitable. Approximation can also be used for energy reduction of video decoding even for not pre-transcoded videos (e.g. loop perforation for video decoding). Numerous other applications from domains as wide as online translation, 3D rendering, and activity recognition could also benefit from AMC. Providing a general framework for supporting AMC, however, remains challenging. In Figure 2 we depict our vision of such a framework. The framework would take into account varying context and users' result accuracy expectations and bringing context-awareness to the lower levels of the application stack where generally-applicable ACTs are implemented. In addition, we consider that putting the user directly in the loop controlling approximate execution enables a proactive control of the approximations based on the predicted user's requirements (a synergy between approximate and anticipatory mobile computing [8]). We envision that the framework contains mobile implementations of the most applicable ACTs that proved their efficiency in desktop scenarios. The ACTs should be flexible enough that a system controlling the approximation could tune the ACTs' knobs to enable real-time control of the result accuracy.

3.1 AMC and context inference

Inferring the context is a crucial part driving AMC adaptation, but it comes with a price in terms of energy consumption. Yet, approximation could also help reduce this cost. Mobile implementations of Deep Neural Networks (DNNs) for human activity recognition [4], for example, can use approximate computing to improve their efficiency [1].

Going back to the case study of an adaptive video resolution app, we could implement the activity classifier using hierarchical decomposition. Instead of having a single, more complex classifier to identify each of the four relevant user mobility states we can decompose it in three, more simpler classifiers corresponding to three stages. This classifier architecture is shown in Figure 3.

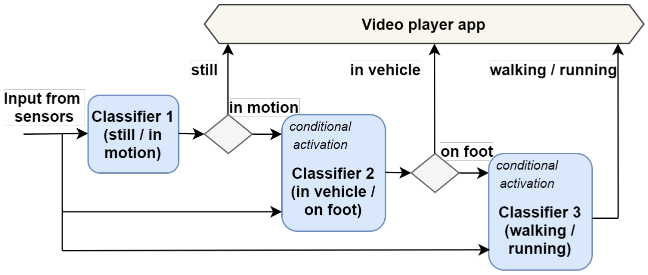

Figure 3. Proposed approach of a hierarchical decomposed classifier for human activity recognition.

Similar approaches of hierarchical classification frameworks [6] or scalable-effort classifiers [10] proved their energy efficiency by showing 1.5× to 2× improvements in energy consumption over traditional classifier architectures.

4 Conclusions and future steps

In this paper we discussed the motivation for implementing AMC starting from the observation that smartphone user's accuracy/quality expectations are context-driven. We showed that users are more likely to accept a lower video resolution, if engaged in physical activities, such as walking or running. Our experiments demonstrate that applying context-based adaptation of the video resolution during playback could lead to up to 48.6% energy savings. Further, we propose a general framework for supporting AMC by moving further down the stack and performing context-driven control of approximation techniques. Finally, we discuss applying ACTs to context inference in order to reduce the overhead of AMC.

Acknowledgments

The research presented in this paper was funded in part by the project "Bringing Resource Efficiency to Smartphones with Approximate Computing" funded by the Slovenian National Research Agency (ARRS), project number: N2-0136.

References

[1] Jungwook Choi and Swagath Venkataramani. 2019. *Approximate Computing Techniques for Deep Neural Networks*. Springer International Publishing, Cham, 307–329. https://doi.org/10.1007/978-3-319-99322-5_15

[2] Hadi Esmaeilzadeh, Adrian Sampson, Luis Ceze, and Doug Burger. 2012. Neural acceleration for general-purpose approximate programs. In *2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE, 449–460.

[3] Zvi M Kedem, Vincent J Mooney, Kirthi Krishna Muntimadugu, and Krishna V Palem. 2011. An approach to energy-error tradeoffs in approximate ripple carry adders. In *IEEE/ACM International Symposium on Low Power Electronics and Design*. IEEE, 211–216.

[4] Athanasios Lentzas, Andreas Agapitos, and Dimitris Vrakas. 2019. *IET Conference Proceedings* (January 2019), 1 (6 pp.)–1 (6 pp.)(1). https://digital-library.theiet.org/content/conferences/10.1049/cp.2019.0098

[5] Sasa Misailovic, Stelios Sidiroglou, Henry Hoffmann, and Martin Rinard. 2010. Quality of service profiling. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1*. 25–34.

[6] Priyadarshini Panda, Swagath Venkataramani, Abhronil Sengupta, Anand Raghunathan, and Kaushik Roy. 2017. Energy-efficient object detection using semantic decomposition. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25, 9 (2017), 2673–2677.

[7] Veljko Pejović. 2019. Towards Approximate Mobile Computing. *GetMobile: Mobile Computing and Communications* 22, 4 (2019), 9–12.

[8] Veljko Pejovic and Mirco Musolesi. 2015. Anticipatory mobile computing: A survey of the state of the art and research challenges. *ACM Computing Surveys (CSUR)* 47, 3 (2015), 1–29.

[9] Mehrzad Samadi, Davoud Anoushe Jamshidi, Janghaeng Lee, and Scott Mahlke. 2014. Paraprox: Pattern-based approximation for data parallel applications. In *Proceedings of the 19th international conference on Architectural support for programming languages and operating systems*. 35–50.

[10] Swagath Venkataramani, Anand Raghunathan, Jie Liu, and Mohammed Shoaib. 2015. Scalable-effort classifiers for energy-efficient machine learning. In *Proceedings of the 52nd Annual Design Automation Conference*. 1–6.