Multimodal data acquisition on mobile devices

Ales Lanik Jozef Mlich Pavel Zemcik Roman Juranek Petr Chmelar
Department of Computer Graphics and Multimedia
Faculty of Information Technology, Brno University of Technology
Brno, Czech Republic, 612 66
[ilanik, imlich, zemcik, ijuranek, chmelarp]@fit.vutbr.cz

Abstract — *This paper deals with acquisition of data on mobile devices in order to consequently process, analyses and browse. It mobile devices provide wide variety of stream data, for example audio-visual, accelerometer, location, wireless status, and other data available on particular devices. The data acquisition and experimental results achieved on the acquired data set are presented.*

1 Introduction

Data acquisition directly influences performance of subsequent methods. The modalities captured and their precision is thus crucial issue for a wide variety of tasks. It has been shown that fusion of the information from different modalities can improve the overall precision of systems.

The motivation of the presented work is to acquire and provide information that is not available using other sensing methods. For instance, users may want to organize photos and videos not only according to the time, but also according to the geographical position or according to the objects they contain. It can be e.g. castle, restaurant or a traffic crossing. Similarly to Google Goggles, different kinds of objects and places can be visually identified, using domain-dependant pictures only.

From the technological point of view, we need an appropriate data set for computer vision and image processing. An example of processing is the creation of complex panorama photos, 3D object reconstruction or classification from fusion of the visual, position and motion information.

If a possibility exists to make use of GPS data and accelerometer, the position and orientation of the device can be estimated in 3D more precisely.

Moreover, camera motion estimation, moving object detection and camera lens intrinsic parameters may be estimated and refined from the data. We suppose, the additional information to the visual can improve the performance and precision of these methods. The collected data should also provide information for such tasks as general and spatio-temporal data mining (e.g. traffic density, closed road or Wi-Fi networks coverage).

2 Related work

Lots of general purpose multimedia frameworks such as FFMpeg, Direct Show and GStreamer. These frameworks deal with temporal synchronization of audio-visual data. These frameworks, however, does not provide access to other modalities [3]. On the contrary, frameworks which provide unified access to hardware components do exist. For instance, the Maemo [2] liblocation framework provides information about location from fusion of GPS, GSM and Wifi modules. The Maemo provides similar application interfaces for whole hardware stack. The other application interfaces also unifies access to the hardware such as MeeGO [4] and Google Android [1]. However, these application interfaces do not support the temporal data synchronization and unified storage capabilities. To fulfill all requirements for information retrieval purposes, which are summarized in industrial standard MPEG-7 [7], it is necessary to filter gathered data and extract meta-data. For information retrieval, computer vision and audio processing modules are crucial. The meta-data extraction modules are usually provided by separate libraries, which cover only single modalities. In case of computer vision module the OpenCV [6] framework can be used. In case of the audio content search, general sound recognition methods [8] and speech recognition systems [10] are widely used .

3 Data acquisition

For the above purposes (see Section 1), we attempt to record all information available on a particular device. Information, such as latitude and longitude, elevation, gravity of acceleration, directional information using compass, Wi-Fi network identifiers (SSIDs), and signal strength, information from GSM transceiver station (BTS) and, of course video.

Figure 1 shows OMAP 3430 System on Chip, which is included in Nokia N900 cell phone and it is typical example of contemporary smart phone hardware. It provides the above mentioned data sources.

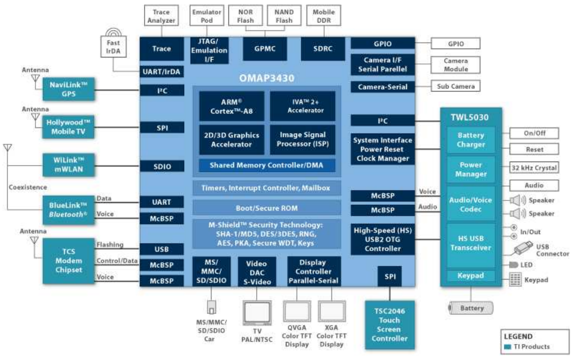

Figure 1: OMAP 3430 SoC Block Diagram [5]

We gather the audio-visual data, which are processed by the meta-data extraction modules. The meta-data consists from low-level and high-level image features sets. Along with the audio-video data accelerometer, location, wireless status, and other data are extracted.

The hardware components operate on different sampling rates and therefore, it is necessary to perform temporal synchronization of the data. The synchronization is performed by inclusion of timestamp into the data stream.

The information is acquired using streaming protocols (RTSP), containing video information using MPEG-4 (3GP) and MPEG-7 (Multimedia Content Description Interface) for streaming of the related informations to a server. The visual information can be extended with low level image information, e.g. interesting points descriptors, color or texture features, and by high level information e.g. detection of certain objects (cars, signs, restaurants, etc.), events (doors opening, cars running, etc.), and concepts (indoor or outdoor scene, etc.).

The meta-data extraction should be done directly on mobile device. However, some feature extraction have high computational cost. Therefore, some feature extraction methods can be performed in offline mode either directly in the device or at the server side.

The implementation can be done using an experimental software similar to that we developed for the TRECVid (NIST TREC Video Retrieval Evaluation) workshop in recent years [9]. As a data storage, we use a database system capable of information retrieval functions and using image (features') similarity.

The above mentioned captured information is appropriately stored together with the audio-visual stream. MPEG-7 standard defines container for storing audio-visual data stream together with other custom data. All the captured data are indexed to simplify later usage. For example, spatio-temporal index, wireless network index, etc. Moreover, we plan to build a simple client (browser) of the captured data.

4 Data analysis

The data analysis provided by the implemented system includes online general object detection engine and offline (server-side) meta-data extraction, indexing and search methods. Contemporary mobile devices use operating systems that fulfill the hardware requirements for the use cases mentioned above. Some of the devices even support accelerated data processing by FPU or SIMD units or DSP processor. Some algorithms, however, are not suitable to be executed on mobile devices due to limited memory, cpu speed or other factor.

Therefore, we have implemented the object detector based on WaldBoost algorithm [14] which is extension of Real AdaBoost that has been used in original rapid object detection system developed by Viola and Jones [15]. Classifiers produced by the WaldBoost are suitable for real-time operation. The detection is based on classification of image subwindows by a classifier, which contains a set of simple image features. This type of detector gives sufficient detection rate while keeping low computational complexity [11].

5 Experimental results

The Android platform is based on Java. This language, however, is not suitable for image processing because it does not support low level access to camera device. Since version 1.5, the Android SDK enables to implement of algorithm partially in C language and call it from Java through JNI (Java Native Interface).

We have implemented basic form of AdaBoost framework on the mobile platform Android 1.6 (see above) and Maemo (see below) without optimizations. The core of this framework is identical with the desktop version and it is used by Java front-end.

The Android software stack, however, has an inappropriate design of the memory sharing. The problem affects transfer of the image data between Java and C. Moreover, it is not possible to access the camera device directly from the native C code. The memory sharing part was identified as the most significant source of processing latency. For these reasons, the Android implementation is slower than the implementation on Maemo (see below). The performance will be improved by direct camera access which is enabled in the Android SDK

simple	SSE	cuda	gpu	n900	htc desire
150	897	2270	2506	11.2	2.3

Table 1: Results of face recognition

2.2 and by parallel implementation of the detection (image feature extraction in particular) with SIMD instructions.

The Maemo (on N900 device) allows accessing the hardware resources directly using the video for linux (V4L) interface. However, for the temporaly synchronized data acquisition, it is more feasible to use the GStreamer interface.

Table 1 compares results of frontal face detection with classifier consisting of 1000 MB-LBP [13] image features on different architectures. The image input was sized to 160x120 pixels and the detection window was 24x24 pixels.Only single scale detection only was performed. The column simple refers to the implementation on CPU with no optimizations. This one is treated as baseline. The SSE extends the simple detection with the optimized feature extraction using SIMD instructions. The CUDA and GPU are high performance implementations that use special graphic hardware. The final two algorithms implemented on N900 and HTC Desire are using the simple version of the algorithm on these mobile platforms.

From the table, it is clear that the CPU/GPGPU implementations beat the mobile platforms due to processing architecture (speed, cache, memory bandwidth, etc.). On the other hand, the applications on mobile platforms in this case do not need such high processing performance.

The detection engine implemented does not use any optimizations so the performance is rather low (about 10 fps). However, SIMD optimized detection engines exists [12, 11] so in the future we will employ SIMD units (on devices where available) to gain performance and decrease overall load of the system.

The Experimental setup was the following. A PC with hardware configuration CPU Intel Core2 Duo E8200 at 2.66 GHz, 3 GB DDR3 RAM and ASUS NVidia ENGTX280/HTDP. Mobile devices were n900 with ARMv7 Cortex A8 at 600 MHz, 256 MB RAM and device HTC desire (Android implementation) with ARMv7 Cortex A8 at 1 GHz, 512 MB RAM.

6 Conclusions

We developed a system for acquisition of data on mobile devices. We focused on two main platforms – Android and Maemo, which are typical representatives of contemporary mobile platforms. The data gathered by the system are suitable for a wide variety of tasks from the field of computer vision and information retrieval.

Currently, the work is focused on the server side of the system and on the optimization of client applications. The server side takes care of gathering the data from multiple devices and the data preprocessing. The preprocessing is currently done directly on the devices. The work continues also on the browser of gathered data.

In the future, we will focus on the extension of preprocessing modules and improvement of performance of applications on mobile devices. Some algorithms can benefit from SIMD processing, which can significantly reduce computation cost and thus load of the system, which is currently an issue.

Acknowledgement

The work presented in this paper was funded in part by the FRVŠ, Ministry of Education, Youth and Sports, FR2747/2010G1 ”Multimedia content on mobile devices” and the BUT FIT grant FIT-10-S-2 and the research plan MSM0021630528 and grant by the European Commission under contract FP7-215453 - WeKnowIt and the FP7 project TA2: Together Anywhere, Together Anytime ICT-2007-214793.

References

[1] The developer's guide | android developers. http://developer.android.com/guide/index.html.

[2] Documentation/Maemo 5 developer guide - maemo.org wiki. http://wiki.maemo.org/Documentation/Maemo_5_Developer_Guide.

[3] GStreamer: open source multimedia framework. http://www.gstreamer.net/.

[4] MeeGo 1.1 API reference. http://apidocs.meego.com/1.1/core/html/index.html.

[5] OMAP 3 processors - OMAP3430. http://focus.ti.com/general/docs/wtbu/wtbuproductcontent.tsp?contentId=14649&navigationId=12643&templateId=6123.

[6] OpenCV. http://opencv.willowgarage.com/wiki/.

[7] O. Avaro and P. Salembier. MPEG-7 systems: overview. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):760 –764, June 2001.

[8] M. Casey. MPEG-7 sound-recognition tools. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):737–747, 2002.

[9] Petr Chmelar, Vitezslav Beran, Adam Herout, Michal Hradis, Ivo Reznicek, and Pavel Zemcik. Brno university of technology at trecvid 2009. In *TRECVID 2009: Participant Notebook Papers and Slides*, page 11. National Institute of Standards and Technology, 2009.

[10] Thomas Hain, Lukas Burget, John Dines, N. Phillip Garner, Asmaa Hannani El, Marijn Huijbregts, Martin Karafiat, Mike Lincoln, and Vincent Wan. The AMIDA 2009 meeting transcription system. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*, volume 2010, pages 358–361. International Speech Communication Association, 2010.

[11] Adam Herout, Pavel Zemcik, Michal Hradis, Roman Juranek, Jiri Havel, Radovan Josth, and Martin Zadnik. Low-Level image features for Real-Time object detection. In *Pattern Recognition, Recent Advances*, pages 111–136. IN-TECH Education and Publishing, 2010.

[12] Adam Herout, Pavel Zemcik, Roman Juranek, and Michal Hradis. Implementation of the "local rank differences" image feature using simd instructions of cpu. In *Proceedings of Sixth Indian Conference on Computer Vision, Graphics and Image Processing*, page 9. IEEE Computer Society, 2008.

[13] Seong-Whan Lee, Stan Li, Lun Zhang, Rufeng Chu, Shiming Xiang, Shengcai Liao, and Stan Li. Face detection based on Multi-Block LBP representation. In *Advances in Biometrics*, volume 4642 of *Lecture Notes in Computer Science*, pages 11–18. Springer Berlin / Heidelberg, 2007.

[14] Jan Sochman and Jiri Matas. Waldboost - learning for time constrained sequential detection. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 150–156, Washington, DC, USA, 2005. IEEE Computer Society.

[15] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:511, 2001.