ClockDrift: A Mobile Application for Measuring Drift in Multimedia Devices

Mario Guggenberger
mg@itec.aau.at

Mathias Lux
mlux@itec.aau.at

Laszlo Böszörmenyi
lb@itec.aau.at

Institute of Information Technology
Alpen-Adria-Universität Klagenfurt
9020 Klagenfurt am Wörthersee, Austria

ABSTRACT

Parallel recordings made at the same event with different devices, e.g. by visitors of a concert, contain semantically the same content but do not run at the same speed when played back in parallel on a computer, which makes their synchronization difficult. This effect, time drift, concerns all current consumer multimedia recording devices and results from their internal clocks not running at the same speed, leading to deviations from their nominal sampling rates. We present a mobile application capable of conducting instant measurements of this time drift, thus helping in determining devices that go well together, or correcting the speed differences in post processing.

Categories and Subject Descriptors

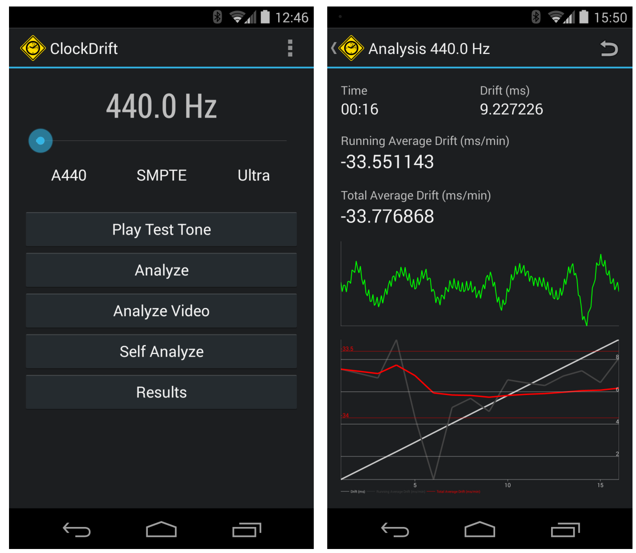
H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*Audio input/output*; H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing—*Signal analysis, synthesis, and processing*

Keywords

Audio, recording, clock drift, time drift, synchronization

1. INTRODUCTION

Electronic devices like smartphones, tablets, audio recorders and video cameras are usually based on complex electronic circuits, whose actions are coordinated by clock signals. These signals are generated by crystal oscillators, which usually do not exactly run at their specified nominal frequency, but deviate a little bit and are also prone to environmental influences. This means that a clock signal oscillating at 1 Hz nominal frequency will in reality, when measured in UTC time, oscillate at a rate of $1 + \epsilon$ Hz, where the error, called *drift*, is usually very small but measurable. Since the audio codec's sample rate and the video codec's frame rate are derived from this signal, they are also influenced by the drift. This leads to playback and recording processes being carried out slightly off their nominal rates (like e.g. specified in the file header) and results in stretched or reduced nominal run times, whose deviation we call *time drift*. This deviation leads to problems when synchronizing parallel recordings made with different devices at the same event, e.g. at a music concert. When comparing the recordings side by side, one can often notice that their speeds do not match perfectly and they, when synchronized at some point, will drift apart and not stay in sync during their whole length. Depending on the drift rate, it becomes noticeable to humans after some hours in the best case, and after a few minutes in the worst case. This concerns recordings from uncontrolled environments, e.g. crowd sourced videos from music concerts, as well as coordinated productions by amateurs. A very basic example for an amateur production is a recording of a talk or interview with a video camera and a separate audio record-



(a) Main menu (b) Analysis screen

Figure 1: The main screens in our app: (a) contains the main functions, (b) shows the drift in ms/min (can be switched to ppm, ms/h, s/h, frequency and percentage by tapping), the zero-crossing distances in green and the measurement progress chart on the bottom.

MM'14, November 3–7, 2014, Orlando, Florida, USA.
ACM 978-1-4503-3063-3/14/11.
http://dx.doi.org/10.1145/2647868.2654886.

ing from a high quality voice recorder, and adding a second camera perspective or an additional audio recording of the interviewer or audience only makes the problem worse. Due to cost, many amateurs do not have any alternatives but using unsynchronized consumer/prosumer equipment which is susceptible to the drift problem. In highly professional environments, devices are usually fed by a common master clock signal at recording time to avoid this problem. In the scientific multimedia community, we could only find one recent paper addressing the problem [1] and none proposing a solution. In this demo, we are going to focus on the audio sampling rate drift and present a mobile application specifically made to quickly measure the drift rate in multimedia devices.

2. TIME DRIFT

Clock drift leads to deviations in the audio playback and recording sample rates in devices, which result in pitch shifts in, and bandwidth changes of the transmitted signals, but also impact the run times of played or recorded files. This change of run time is what we call the time drift. While drift in oscillators is commonly specified in parts per million (ppm), we specify it in milliseconds per minute (ms/min) for more intuitive understanding.

To measure the relative drift between two devices A and B, we play back a sine wave with a nominal frequency f_n on device A, and record it with device B. We then measure the frequency f_r in the recorded data and use it to calculate the total drift factor $d_t = f_n/f_r$. In case of drift, f_r is slightly off f_n and therefore the factor slightly off 1. This kind of measurement is sufficient to compensate the drift in recordings made with two or more devices by measuring all devices from one reference playback device and resampling or stretching the recordings according to the calculated drift factor. Measuring the absolute drift of a device off UTC time requires a calibrated measurement device, e.g a high-precision spectrum analyzer. Again, the device to be measured plays back a sine wave at f_n Hz and the playback drift factor $d_p = f_n/f_p$ can be determined through the measured frequency output f_p from the playback device. Since the hardware audio codec of a device C derives the playback sample rate from the same clock signal as the recording sample rate, we can assume that both sample rates are the same, thus both drift factors are equal and therefore $d_p^C \equiv d_r^C$.

Table 1 lists measurements of several different devices. It is important to note that the measured values are not absolute constants since temperature has measurable influence on the drift, and there is a variance between devices of the same model. Exemplary measurements of different devices have shown a variance of $\approx 1\,\text{ms/min}$ between $-20\,^\circ\text{C}$ and $+50\,^\circ\text{C}$ in temperature, and a standard deviation of $\approx 0.1\,\text{ms/min}$ between five LG Nexus 5 and eight Asus Nexus 7 devices. These values clearly show that there are huge differences between different devices, which make certain multimedia use cases, e.g. the synchronization of parallel and overlapping recordings of events, a non-trivial task.

3. CLOCKDRIFT APP

Our app is available for free through the Google Play Store[1]. Its two fundamental functions are to (i) act as a playback source by playing out a test signal, and to (ii) act as a recording target that analyzes the incoming signal on the fly and displays the measured drift to the user. The main screens are shown in Figure 1. A measurement works by playing a sine wave at the source and isolating it at the target from the incoming signal with a narrow band-pass filter centered around the chosen frequency, which can be freely adjusted by the user up into the imperceptible ultrasonic band. The filtered signal then gets further processed by measuring the distances between consecutive zero-crossings of the waveform and averaging them over time, resulting in a precise frequency measurement that can be transformed to a drift reading. The distances are calculated to sub-sample precision by linearly interpolating the zero-crossing points from their neighboring sample values, which almost instantly leads to a precise drift rate determination. The measurement is further improved by determining the strength of the incoming signal and only measuring while its energy is sufficient, else the analysis gets paused and a hint shown to the user. The measurements can be stored locally on the device. This data also contains additional device metadata, which can, together with the measurement data, be submitted to our server with the goal to build a database of device drift indicators. A detailed user guide explaining all the app's functionality is available online[1].

[1] http://www-itec.aau.at/~maguggen/clockdrift/

Table 1: Measured drifts of various multimedia recording devices in ms/min.

Device	Type	Drift
Acer Iconia A200	Tablet	0.8
Apple iPad 2 Wi-Fi	Tablet	0.8
Apple iPod touch 4G	Media Player	25.0
Asus Nexus 7 2012 Wi-Fi	Tablet	0.2
LG Nexus 4	Smartphone	0.4
Samsung Galaxy Note 10.1	Tablet	1.0
Samsung Galaxy S II	Smartphone	16.4
Samsung Galaxy Spica	Smartphone	-0.9

4. CONCLUSION

We presented a method to measure the clock drift or time drift in multimedia devices capable of audio playback or recording, and described our Android app specifically developed for the purpose of quick on-the-go measurements of the drift. We have shown that common, often used devices like smartphones can suffer from enormous drift that makes the synchronization of even short recordings with a few minutes length impossible without further post-processing. In possible future work we will take a look specifically at video recording, which in a minority of devices seems to depend on another clock source and leads to different measurement results compared to audio only recordings.

5. ACKNOWLEDGMENTS

This work was supported by Lakeside Labs GmbH, Klagenfurt, Austria, and funding from the European Regional Development Fund (ERDF) and the Carinthian Economic Promotion Fund (KWF) under grant 20214/22573/33955.

6. REFERENCES

[1] A. L. Casanovas and A. Cavallaro. Audio-visual events for multi-camera synchronization. *Multimedia Tools and Applications*, March 2014.