SURF and SURF-PI: A File Format and API for Non-Intrusive Load Monitoring Public Datasets

Lucas Pereira, Nuno Nunes
Madeira-ITI, University of Madeira
Polo Científico e Tecnológico da Madeira, floor -2
Caminho da Penteada, Funchal, Madeira, Portugal
lucas.pereira@m-iti.org, njn@uma.pt

Mario Bergés
Civil and Environmental Engineering,
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA, USA
marioberges@cmu.edu

ABSTRACT

In this paper we propose a common file format and API for public Non-Intrusive Load Monitoring (NILM) datasets such that researchers can easily evaluate their approaches across the different datasets and benchmark their results against prior work. The proposed file format enables storing the power demand of the whole house along with individual appliance consumption, and other relevant metadata in a single compact file, whereas the API supports the creation and manipulation of individual files and datasets in the proposed format.

Categories and Subject Descriptors

D.2.13 [**Software Engineering**]: Reusable Software – *reusable libraries.*

Keywords

Energy Disaggregation, Datasets, File Format, API.

1. INTRODUCTION

NILM, first introduced by George Hart in his seminal work [1], is the process of estimating the energy consumption of individual appliances given only current and voltage measurements taken at a limited number of locations in the electric distribution of a building. Yet, despite decades of research and recent efforts towards creating public datasets (e.g. [2] and [3]) to validate and improve the existing approaches, very few formal evaluations (e.g. [4]) of the technology have been carried out so far, thus raising questions about the large scale applicability of this technology. We argue that one of the reasons for this is the difficulty of objectively comparing the performance of different algorithms given the lack of public datasets and the wide differences between the ones currently available. In fact, only recently there has been a serious effort to homogenize the existing datasets and provide a single interface to run evaluations [5] to which we wish to contribute by proposing SURF and SURF-PI, a common file format and programming interface to support the creation and manipulation of public NILM datasets, to help homogenize the whole process of systematically evaluating NILM algorithms across different datasets.

e-Energy'14, June 11--13, 2014, Cambridge, UK.

ACM 978-1-4503-2819-7/14/06…$15.00.
http://dx.doi.org/10.1145/2602044.2602078

2. SURF FILE FORMAT

The proposed format is an extension of the Waveform Audio File Format (WAVE) that supports the storage of digital audio data and metadata annotations according to the underlying chunk structure that is defined by the Resource Interchange File Format (RIFF) standard. There are four main reasons behind expanding this format and not another: i) data and metadata are all stored in a single compact file, thus limiting the number of artifacts that need to be managed; ii) the possibility of adding custom chunks without breaking the file consistency; iii) the resulting files are optimized to have little overhead; and iv) the mature programming interfaces that exist for a diversity of programming languages, hence facilitating the expansion and portability of the proposed format and API.

The SURF file format is currently composed of 13 chunks each one containing its own header and data bytes. Eight chunks are inherited from the WAVE format, one from the RIFF standard, and the remaining four are custom chunks created to supplement the files with relevant metadata. Next we describe the underlying structure of the SURF format.

2.1 Power Demand Data

The power demand data is defined in the Format chunk (*sampling rate*, *sample size* and *channels*) and stored in the Data chunk. The data values are stored uncompressed (to preserve the original signal) in little-endian byte order and scaled to the interval]-1,1[.

2.2 Individual Appliance and User Activities

Individual appliance activities correspond to the changes in the power demand that are triggered when individual loads change their operating mode (e.g. going from *on* to *off* and vice-versa), whereas user activities are groups of related individual appliances activities (e.g. combine the clothes-washer, dryer and iron activities to form the *"laundry"* user activity). All these activities have a corresponding timestamp (user activities also have an end timestamp) that are mapped to the corresponding sample number in the power demand data.

These activities are embedded in the SURF files using the Cue, Associated Data List, Label and Labeled Text chunks. Each activity is represented by their respective positions in the power demand data and a JSON formatted string with its details (see figure 1). For example, the following two JSON strings

correspond to a refrigerator activity that was mapped to sample (position) 19394633, and a working activity that involves using the desktop computer, monitor and printer:


```
{"ID": 1101, "Type": 1, "Position": 19394633,
"Timestamp": "2011-10-24 05:45:57.040",
"Appliance_ID": 111, "Appliance_Label":
"Refrigerator"}
```

```
{"ID": 10021, "Activity_ID": 111,
"Activity_Label": "Working" "Start_Position":
14300225, "End_Position": 15000377,
"Start_Timestamp": "2011-10-21 21:05:14.940",
"End_Timestamp": "2011-10-22 00:23:18.056",
"Appliance_Activity_IDs": [1201, 1209, 1303, 1304,
1305, 1401, 1402, 1403], "Total Power": 1290}
```


2.3 Metadata

By default the RIFF standard enables metadata fields in the Info chunk, some of which we have repurposed according to our requirements: *artist* (renamed *dataset creator*), *title*, *date created*, *comment* and *copyright*. Likewise, we have also used the Note chunk to add localized metadata directly in the power demand data (e.g. when some appliance is added or removed from the buildings' electric circuit).

Furthermore, to enable a richer set of metadata we also complemented the SURF files with custom chunks, specifically created for this effect: i) Config: for configuration specific metadata (e.g. initial timestamp and calibration values); ii) External: metadata that refers to variables external to the dataset (e.g. the sensing hardware that was used); iii) Appliances: a list of the existing appliances; and iv) Activities: a list of the existing user activities.

3. SURF PROGRAMMING INTERFACE

The SURF-PI was implemented combining and extending several open source Java libraries for audio edition. The current version enables three main types of operations: i) **Create / Update**: functions to write, edit and delete the available chunks, e.g. *WritePower(powerData),* and *SetApplianceActivity(position, jsonString)*; ii) **Read**: functions to read the chunks data, e.g. *GetFormat(), GetUserActivity(id)*; *GetAppliances()*; and iii) **Query**: functions for NILM specific queries, e.g. *GetIntervalConsumption(startPosition, endPosition)*, and *GetActivityConsumption(activityID)*.

Additionally, since most of the annotation data are done using JSON we have implemented validation schemas using the JSON Schema Draft 4 to remove ambiguity and errors that may occur when creating the dataset files. We believe that having such a specification is especially important if we consider the possibility of extending or porting this API to other programming languages.

4. CONCLUSION AND FUTURE WORK

In this paper we have presented SURF and SURF-PI, a file format and programming interface for NILM datasets.

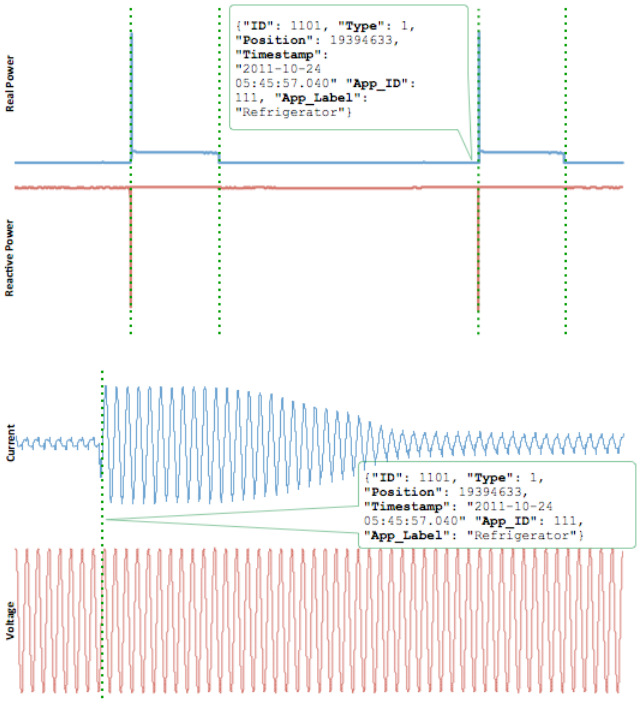

Figure 1. Refrigerator activity: real and reactive power at 60 Hz (top), current and voltage at 12 kHz (bottom).

We are now working towards proving the general applicability of this work and to this end we are converting some of the existing datasets to our format. Furthermore, since our work aims at generalizing NILM research, future work will involve evaluating and benchmarking previously proposed algorithms across the different datasets using SURF and SURF-PI. Likewise, it is very likely that different researchers will have different requirements regarding the proposed API, annotations and metadata, therefore we will be, soon, releasing a stable and documented open-source version of this work that will be accessible from: *http://aveiro.m-iti.org/software/surf*.

5. REFERENCES

[1] G. Hart, "Nonintrusive appliance load monitoring," *Proc. IEEE*, vol. 80, no. 12, 1992.

[2] Z. Kolter and M. Johnson, "REDD: A public data set for energy disaggregation research," *SustKDD '11*.

[3] K. Anderson et al., "BLUED: A Fully Labeled Public Dataset for Event-Based Non-Intrusive Load Monitoring Research," *SustKDD '12*.

[4] Electric Power Research Institute, "Non-Intrusive Load Monitoring (NILM) Technologies for End-Use Load Disaggregation: Laboratory Evaluation I." [Online]. Available: http://tinyurl.com/kloo5wq.

[5] N. Batra et al., "NILMTK: An Open Source Toolkit for Non-intrusive Load Monitoring," *e-Energy '14*.