Energy Analytics in Public Buildings using Interactive Histograms

Ignacio Díaz Blanco[a,*], Abel Alberto Cuadrado Vega[a], Daniel Pérez López[a], Manuel Domínguez González[b], Serafín Alonso Castro[b], Miguel Ángel Prada Medrano[b]

[a] *Área de Ingeniería de Sistemas y Automática, University of Oviedo, Gijón, Spain*
[b] *SUPPRESS Research Group, University of Leon, Escuela de Ingenierías. Campus de Vegazana, 24007, León, Spain*

Abstract

In this paper we propose a visual analytics approach based on *data cube* methods to provide an insightful analysis of how energy is being used in a group of public buildings according to many different factors. The analysis is done by means of a web-based visual interface featuring "live" coordinated views —histograms— that show the distribution of demand data, according to different attributes, under different scenarios defined by user-driven filters on these attributes. We use the `crossfilter.js` library to achieve real-time computation of data cube aggregations for constantly changing user-defined filters, resulting in a fluid visualization of demand parameters (active power, power factor, total harmonic distorsion, etc.) aggregated according to many different factors or dimensions such as *time* (hour, day of week, month, etc.), *building* or *environment* (outside temperature).

Keywords: visual analytics, energy efficiency, multiway analysis, data cube

1. Introduction

Energy efficiency and savings policies have become a challenge of extraordinary strategic importance today. Despite the huge volumes of energy-related data and information available today in most buildings, industrial facilities, and even in households thanks to smart meters, that information is seldom presented in an intuitive way that the user can assimilate, making it difficult to obtain useful knowledge about the energy use. As a result, decisions in energy management are often taken under insufficient or ill-defined information. Thus, it has become increasingly important to have tools that, based on the large amounts of data obtained from sensors installed in energy facilities, allow to increase our *energy awareness*, that is, our perception of how the energy is being spent. In a close approach to this idea, highly cited works like [1, 2], explicitly highlight the importance of *feedback* on energy consumption. Such tools should also be able to handle large volumes of information of different kinds, deal with uncertainty, present information in a clear and intuitive way and also to provide the user with a global view, in order to suggest strategies to improve energy efficiency.

A promising approach in this scenario is the so called *visual analytics* approach. *Visual analytics* (VA) [3] allows the user getting insight from data, through an efficient combination of intelligent data analysis, data visualization and interaction mechanisms, harnessing the ability of the human visual system in detecting interesting patterns quickly and efficiently, and being an outstanding approach to exploit problem domain knowledge. Moreover, since the user becomes part of the analysis loop, this approach results in high degrees of confidence in the results, which favors the adoption of actions. Successful applications of VA for energy analytics have been presented in last years, including household and residential demand analytics [4, 5, 6], as well as analysis of large electric power grids using network analysis, such as force directed algorithms [7, 8] and other visualization techniques [9]. Also, the combination of dimensionality reduction algorithms and data visualization for insightful analysis of energy demand data in buildings in terms of 2D maps, has been proposed in [10, 11]. In [12], prediction and clustering-based novelty detection algorithms are computed for later visualization of anomalies in power consumption data. More recently, in [13], heat map visualizations along pivot table analysis were proposed to enable cross-buildings comparisons on a university campus.

Despite the advantages of VA over pure algorithmic approaches to provide a natural way to integrate a broad spectrum of information types, it is still a rather unexplored topic. Energy demand analysis in buildings is a multifaceted problem that involves many qualitatively different factors, including time factors (periodic, such as day, week, or year patterns and non-periodic, such as holiday periods, special demand patterns related to specific activities or singular events), spatial factors (departments, buildings, areas, regions, etc.) or even environmental or exogenous factors (such as economic or weather conditions), just to mention a few highly relevant for energy analysis. Despite the previous work on VA approaches for energy analysis provides great solutions to specific problems, the problem still requires to address the development of comprehensive tools allowing to visualize the energy demand from many qual-

*Corresponding author
Email addresses: `idiaz@uniovi.es` (Ignacio Díaz Blanco), `cuadrado@isa.uniovi.es` (Abel Alberto Cuadrado Vega), `dperez@isa.uniovi.es` (Daniel Pérez López), `manuel.dominguez@unileon.es` (Manuel Domínguez González), `saloc@unileon.es` (Serafín Alonso Castro), `ma.prada@unileon.es` (Miguel Ángel Prada Medrano)

itatively different factors, leading to a *multiway analysis* problem. As it will be detailed later, arranging the data into a hypercube structure according to the analysis factors, the problem can be posed in terms of *data cube* operations [14] that can be efficiently solved by state-of-the-art software libraries.

This paper proposes the use of coordinated views with interactive histograms that show the distribution of demand data from several buildings and/or facilities, with respect to different attributes to allow the user to discover and understand the factors that affect energy efficiency. Our approach is based on combining a highly interactive (fluid) visualization of histogram-like barcharts, with real-time computation of aggregated demand parameters (active power, power factor, total harmonic distorsion, etc.), for user-defined filters and aggregations across many different factors or dimensions such as *time* (hour, day of week, month, etc.), *building* or *environment* (outside temperature), using a *data cube* approach. The rest of the paper is organized as follows. In section 2 we pose the problem from a twofold perspective: the requirements for energy demand data analysis and the requirements for a visual interface. In section 3 we lay out the theoretical framework —namely a *data cube* formulation— and operations. In section 4, we describe implementation aspects, including real-time cube operations, rendering of visualizations in the web interface, data preparation and the description of the web interface. In section 5 we present and discuss the results and findings of applying the proposed approach and the interactive visualization tool to the analysis of one year of demand on 13 buildings of a university campus. Finally, section 6 provides a general discussion and concludes the paper.

2. Problem formulation

In this section we shall first introduce the problem from two different but closely related aspects: the data analysis problem and the data visualization interface. Then, we shall give a short description of the specific problem to be addressed later, in the results section.

2.1. Data cube approach for energy demand analysis

Energy demand information in buildings is often presented as large tables of data from SQL databases, Excel XLS files, CSV/TSV text files or from more structured formats such as XML (*eXtended Markup Language*) or JSON (*JavaScript Object Notation*). A typical case may consist of a large table, showing several energy demand variables per building (active energy, power factor, etc.), having thousands or even millions of records, sampled at regular intervals (typically, hours or quarters of an hour), for periods of one or more years, involving several buildings.

In most cases, it is of interest for the user to analyse the aggregated demand according to different criteria, such as by hour, week day, month, building or for user-defined ranges of known variables (e.g. active power, power factor or outside temperature) just to mention a few. This leads to a *multiway analysis* problem, since data can be conceived as a multidimensional array —a *data cube*—, with as many dimensions as defining attributes. Data cubes admit many different ways for summarizing information by means of aggregations according to any of their attributes, resulting in simpler summarized structures that can be easily represented, for instance, as pivot tables or chart visualizations, and explain different aspects of the problem.

2.2. Requirements for the data visualization interface

From the point of view of how information is presented to the user, a suitable analysis of energy data in this scenario should accommodate to the following requirements:

- It should allow to provide visualizations with summarized information of the demand behavior, based on the representation of aggregated values for groups defined on each dimension or factor being analyzed. Each dimension can admit different groupings (attributes). For instance, dimension *time* could be grouped in many ways, such as hour, weekday, month, quarter, year, etc.

- The user should also be able to *filter* the results for ranges in one or more attributes. For instance he/she should be able to quickly obtain answers to questions like "what is the aggregated hourly demand for Mondays in building 1?" or "in which hour of the day does the power factor most often fall below 0.95?", "can we answer the former question month by month?".

- The interface should allow a *fluid* selection of ranges by means of brushing gestures. Moreover, the intermediate results during brushing operation should be displayed "on the fly", resulting in lively animated views that provide the user with immediate *feedback*, which places the user in the loop, boosting the analysis process.

- Different kinds of aggregations for each group should be available, including total energy demand, the hourly average demand or simply a count of facts that fall on each group for the current dimensional filter specifications.

- It should establish mental *links* or *connections* between views. This can be achieved by coordinated views that get simultaneously (and quickly) updated on any user action.

2.3. Case description

The data being object of this study contained electric power demand information and other related data from a university campus over a period of exactly one year, from March 2010 to February 2011 with a sampling interval of 2 minutes. The dataset included *measures* of active power, power factor ($\cos\phi$), temperature and total harmonic distorsion (THD), but also attributes such as building id, weekday, hour, month, etc., listed in Table 1, that provide context for analyzing the measures. The dataset contained information from 13 buildings, listed in Table 2, having a broad spectrum of activities, including educational, research, services, sports, etc.

Table 1: dimension attributes used in the interface

id	label	attribute
1	`week`	week of year
2	`hour`	hour
3	`Pact (KW)`	active power (kW)
4	`month`	month
5	`T(C)`	temperature
6	`Edificio`	building id
7	`dayofyear`	day of year
8	`weekday`	day of the week
9	`THD Van (%)`	THD (%)
10	`cosPhi`	power factor ($\cos\phi$)

Table 2: list of buildings

id	Building name
5	Filosofía
15	Animalario
16	INCAFD
17	Pabellón Deportivo
18	Frontón
19	Biblioteca Central
21	Cafetería II
22	Molecular
23	Complejo Agrícolas
24	Colegio Mayor
25	Complejo Rectorado
27	Minas
28	Centro Idiomas

3. Methods and techniques

3.1. Data cube terminology and operations

In this section we shall present several definitions and terminology about data cube elements and operations, some of them adapted from previous work [15, 16] to provide a formal description of the operations used for the energy demand analysis.

Measures. We shall define *measures* $\mu_1, \mu_2, \ldots, \mu_p$, as scalar values that are the objects of analysis. Examples of measures can be `active-power`, `power-factor`, `temperature`, etc.

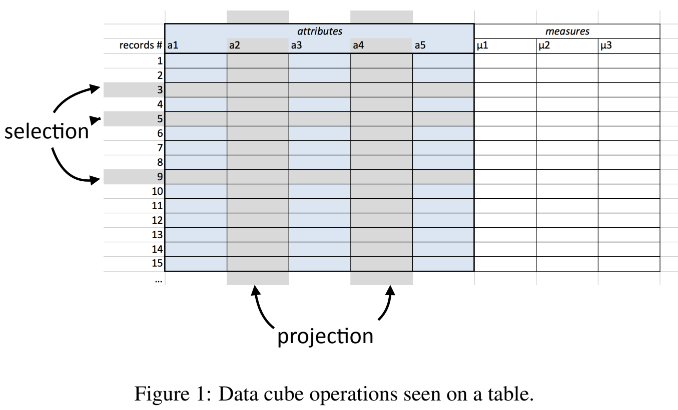

Figure 1: Data cube operations seen on a table.

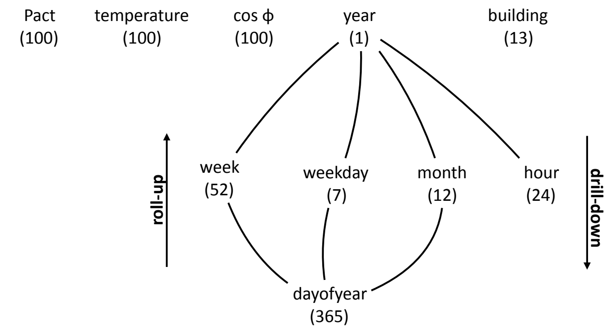

Figure 2: Dimension hierarchy.

Dimensions and attributes. The numeric measures depend on a set of *dimensions* which provide context for the measure [16], such as `time`, `space`, `temperature`, etc. A dimension may be grouped in different ways, according to one or more *attributes* with different levels of aggregation, that in some cases can be organized into hierarchies, as seen in Fig. 2. For instance, dimension `time` can be hierarchically grouped according to attributes such as `month`, `week`, `day`, etc., allowing also other interesting transversal categorizations such as `weekday` or `hour`. Also measures can be used as dimensions (e.g. `temperature`); a typical way to group continuously distributed measures is to partition the whole domain into intervals or "bins", as done in histograms. More formally, attributes can be defined as sets of groups $a_i = \{g^i_1, g^i_2, \ldots, g^i_{|a_i|}\}$, where the g^i_j denotes the j-th group into which a dimension is partitioned according to attribute a_i, and $|a_i|$ is the cardinality of the set a_i. Thus, for instance, {Su, Mo, Tu, …, Sa} are groups of attribute `weekday` of dimension `time`, while the temperature ranges $\{\,[-20^o, 0^o),\ [0^o, 20^o),\ [20^o, 40^o]\,\}$ can be groups forming an attribute of dimension `temperature`.

Data cube. We shall define a *data cube* as a multidimensional structure

$$C(a_1, a_2, \ldots, a_n) \tag{1}$$

where $a_1, a_2, \ldots, a_n$ are attributes. The data cube is a hypercube structure composed of $|a_1| \times |a_2| \times \ldots \times |a_n|$ cells. Each cell is defined by its coordinates $(g^1, g^2, \ldots, g^n)$, being g^k a single group chosen from attribute a_k, and contains the set of records that belong to the n groups $g^1, g^2, \ldots, g^n$ that define the cell position. Note that a cell can also contain an empty set.

Projection operation. Projection refers to selecting the attributes used to view the cube —see Fig.1. This operation is roughly equivalent to projection Π in relational databases. Using the cube definition above

$$\Pi_{a_{i_1},\ldots,a_{i_p}}\left[C(a_1, a_2, \ldots, a_n)\right] \longrightarrow C(a_{i_1}, a_{i_2}, \ldots, a_{i_p})$$

where $\{a_{i_1}, \ldots, a_{i_p}\} \subset \{a_1, a_2, \ldots, a_n\}$. Note that the projection operation implies reducing the cube dimension by collapsing it to leave only the attributes $a_{i_1}, \ldots, a_{i_p}$ (that is, "flattening" the cube), which results in a reduced number of cells in the data cube, but preserving the total number of records. This means that some cells will store larger sets of records that were formerly divided according to groups of other attributes.

Selection operation. Let's define the *selection* operation

$$\sigma_\varphi C(a_1, \cdots, a_n)$$

as the result of selecting the records contained in C for which the predicate φ is true. In general, the predicate φ can be any logical expression on the cube attributes as in relational algebra terminology. However, in typical cube operations, φ is often limited to selections of subsets of groups in one or more attributes[1]. This results in a *dice selection* (since it yields a smaller cube or "dice"; see Fig. 3 for an illustration of the idea), as described in the data cube literature [15, 16, 14]. For instance, for $a'_k = \{a, b\}$, $a'_j = \{p, q, r\}$ we get

$$\sigma_{\substack{a_k=a'_k\\a_j=a'_j}} [C(a_1, a_2, \ldots, a_n)] \longrightarrow \tag{2}$$
$$\longrightarrow C(a_1, \ldots, \underbrace{\{a, b\}}_{a'_k}, \ldots, \underbrace{\{p, q, r\}}_{a'_j}, \ldots, a_n)$$

The *slice selection* is a particular case of the dice selection that arises when a single group a is selected from a single attribute a_k, that is, $a'_k = \{a\}$

$$\sigma_{a_k=a'_k} [C(a_1, a_2, \ldots, a_n)] \longrightarrow C(a_1, \ldots, \underbrace{\{a\}}_{a'_k}, \ldots, a_n)$$

Aggregation operation. Let's define the *aggregation function* $\mathcal{A}$ as a function that takes the set of records associated to a cell and yields a single value or object for summarizing purposes[2]. Let's assume that $\mathcal{A}$ can be applied to all the cells of the data cube, producing an aggregated value or object per cell. We define the *aggregation operation* as

$$\mathrm{Agg}_{\mathcal{A}} C(a_1, \ldots, a_n)$$

which returns a n-array of $|a_1| \times |a_2| \times \ldots \times |a_n|$ aggregated values, one per cell in the data cube. This array is often the last step previous to reporting results in the data cube workflow.

The slice and dice selections are often complemented with projection and aggregation in the data cube workflow for producing summaries. An instance of a *slice* operation can be

$$\mathrm{Agg}_{\mathrm{count}} \Pi_{\mathrm{P_{act}}} \sigma_{\mathrm{month=May}} C$$

It selects May from the month attribute (slice), projects the resulting slice on attribute P_{act} (in this case, actually a specific partition of *measure* P_{act} into "bins" or intervals), resulting in a reorganization of the records on a 1D cube with the groups (bins) of attribute P_{act}, and then counts the number of elements of each group. In other words, these operations return a *histogram* of P_{act} for May.

Another example with a *dice* operation and a different aggregation function can be

$$\mathrm{Agg}_{\mathrm{avg(cosPhi)}} \Pi_{\mathrm{hour}} \sigma_{\substack{\mathrm{building=\{1,3,4\}}\\ \mathrm{weekday=\{Fr,Sa\}}\\ \mathrm{hour=\{11,12,13,14\}}}} C$$

which does a *dice* selection by building, weekday and hour, resulting in a smaller cube (a *dice*, see Fig. 3) and then a *projection* on the hour attribute, resulting in 1D cube with 4 cells, one per selected hour, that contain all the records regardless the building and weekday of the dice. Finally, the aggregation implies in this case the computation of the average cosPhi for the records on each cell, resulting in a 1-array of four values. In Fig. 3 a picture of this operation along with the following two operations

$$\mathrm{Agg}_{\mathrm{sum(P_{act})}} \Pi_{\mathrm{building}} \sigma_{\substack{\mathrm{building=\{1,3,4\}}\\ \mathrm{weekday=\{Fr,Sa\}}\\ \mathrm{hour=\{11,12,13,14\}}}} C$$

$$\mathrm{Agg}_{\mathrm{avg(P_{act})}} \Pi_{\mathrm{weekday}} \sigma_{\substack{\mathrm{building=\{1,3,4\}}\\ \mathrm{weekday=\{Fr,Sa\}}\\ \mathrm{hour=\{11,12,13,14\}}}} C$$

is shown.

Roll up and drill down. According to the previous operations, roll-up and drill-down operations can be defined as recomputing the aggregation on a different attribute of the same dimension with a different level of aggregation. Suppose two different attributes of dimension time, namely, `year = {2010,2011}` and lower level aggregation attribute `quarter = {Q1-10, Q2-10, Q3-10, Q4-10, Q1-11, Q2-11, Q3-11, Q4-11}`, drill-down and roll-up are achieved by recomputing aggregations of active power P_{act} on both attributes

$$\mathrm{Agg}_{\mathrm{avg(P_{act})}} \Pi_{\mathrm{year}} C \quad \underset{\overset{\textstyle\text{roll−up}}{\longleftarrow}}{\overset{\text{drill−down}}{\longrightarrow}} \quad \mathrm{Agg}_{\mathrm{avg(P_{act})}} \Pi_{\mathrm{quarter}} C$$

Fig. 2. provides a schematic description of the roll-up and drill-down relationships among the attributes used in the tool described in this paper.

4. Implementation

4.1. Data cube implementation

We used Crossfilter.js [3] for implementing efficient client side data cube operations. Crossfilter is a JavaScript library for exploring large multivariate datasets in a web browser, that supports fast data cube operations with datasets containing a million or more records.

The Crossfilter library allows to build a data cube object from an array of javascript objects, using the `crossfilter` constructor method. The properties for each object in the array can be selected to define dimension attributes (`dimension` method), and specific groupings (`group` method) can be defined within

[1] This limitation results in "cubic" or "dice" selections of records inside the cube. Note, however, that despite the more general expressions for φ, may produce arbitrary empty cells in the cube, resulting in "non-cubic" selections of records, this does not affect the data cube nature and operations.

[2] In a more general sense, the aggregation result can be an object, for example a structure with several summarizing properties.

[3] `https://github.com/square/crossfilter`

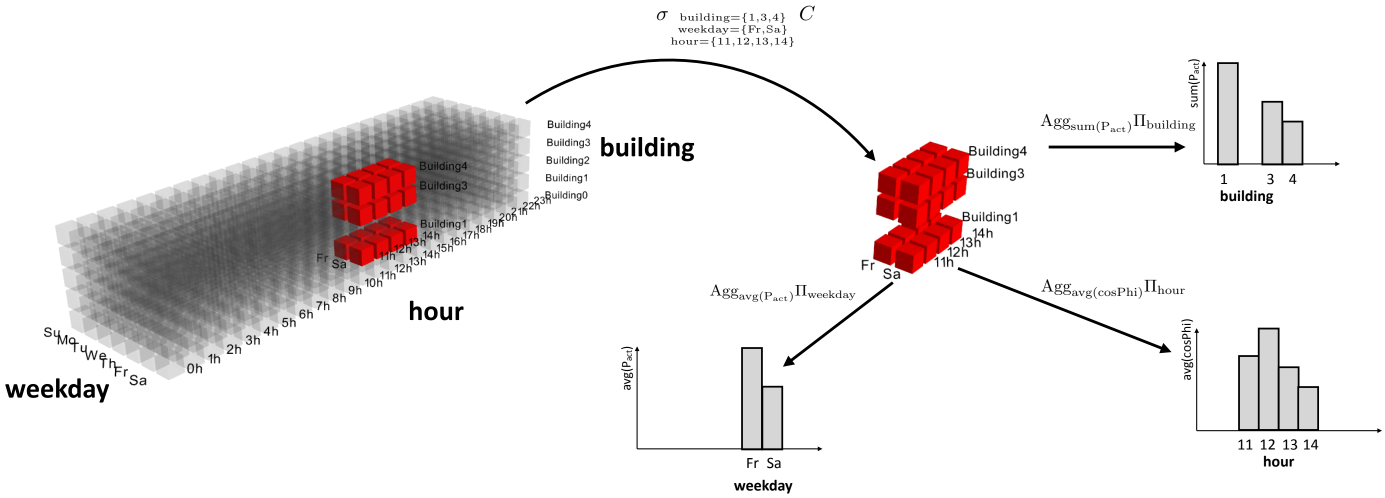

Figure 3: Dice and aggregation operation on a data cube.

attributes, resulting in attribute group sets for further cube operations. Once the data cube, the attributes and the groups have been defined, the `filter` method allows to dice the cube by selecting subsets of groups within one or more attributes (e.g. for a range of values).

Finally, Crossfilter returns the aggregation values (`top` and `bottom` methods) with count values as default or with user-defined aggregations using map-reduce methods (`reduceAdd`, `reduceRemove`, `reduceInitial`), allowing to implement aggregations (e.g. sums and averages) on attribute groups in a highly efficient way.

4.2. Rendering visualizations

The Crossfilter methods for fast computation of aggregated values are tightly integrated in a javascript web app that uses D3.js [17] for rendering the results into coordinated and interactive histogram views, one per attribute in the data cube. Each histogram also includes a vertical red bar showing the mean value. Each view features a brush behavior, activated by mouse drag gestures on the view, that allows the user to define a filter, that defines a selection operation on the corresponding attribute. The selection can be changed both modifying the limits of the defined range by dragging the sides of the selection rectangle and moving the current selection range by dragging the inner area.

By default, all the views are configured as histograms, since the default aggregation function is `count`, which simply counts the number of elements in each group of the attribute for the current selection. More precisely, for the view of attribute a_i

$$\text{Agg}_{count}\Pi_{a_i}\sigma_{current\ selection}C$$

An input textbox allows the user to modify the behavior of the views, by setting the type of aggregation (“avg”, “sum”, or “count”) and the measure being aggregated, using a simple syntax “`attribute,aggregation,measure`”. For instance `week,count,samples` to define the view for attribute week as a histogram,

$$\text{Agg}_{count}\Pi_{\text{week}}\sigma_{current\ selection}C$$

`hour,avg,cosPhi` show average values of cosPhi per hour

$$\text{Agg}_{avg(cosPhi)}\Pi_{\text{hour}}\sigma_{current\ selection}C$$

During mouse drag operation, all aggregations are recomputed and all the views are instantaneously rendered, showing the updated results for the current selection. This occurs on the fly, in a fluid manner at rates close to 20 frames per second (using Safari browser, version 9.0.3, on an 2,4GHz intel core i7 macbookpro retina 8Gb, Early 2013).

In Fig. 4 a detailed description of the elements of a chart are described for two different aggregation configurations.

4.3. Data preparation

We used python (specially pandas and JSON libraries) for data import, curation, and organization into a JSON input data file for the client application. Data originally available in large binary files were imported into a python/pandas `dataframe` structure, and timestamps were generated for every record, being later downsampled to 1 hour periods using average values. Missing data within the hour period were ignored and only the valid samples were used; when no valid data were available for the hour period, the hourly measure was considered a missing value. A further imputation method for such missing values was carried out creating a pivot table with the average values for the measures, using hour and weekday as pivot dimensions, from which the missing values were taken.

To allow defining new attributes for meaningful categorizations of the data, additional helper columns `week day`, `hour`, `week of year`, `day of year` and `month` were created from the available timestamps. All these helper columns were used as attributes for cube filtering and aggregation in the web application.

Finally, data were restructured into javascript objects, one per sample record (8760×13 records, a whole year of data for each building) and packed in a JSON file.

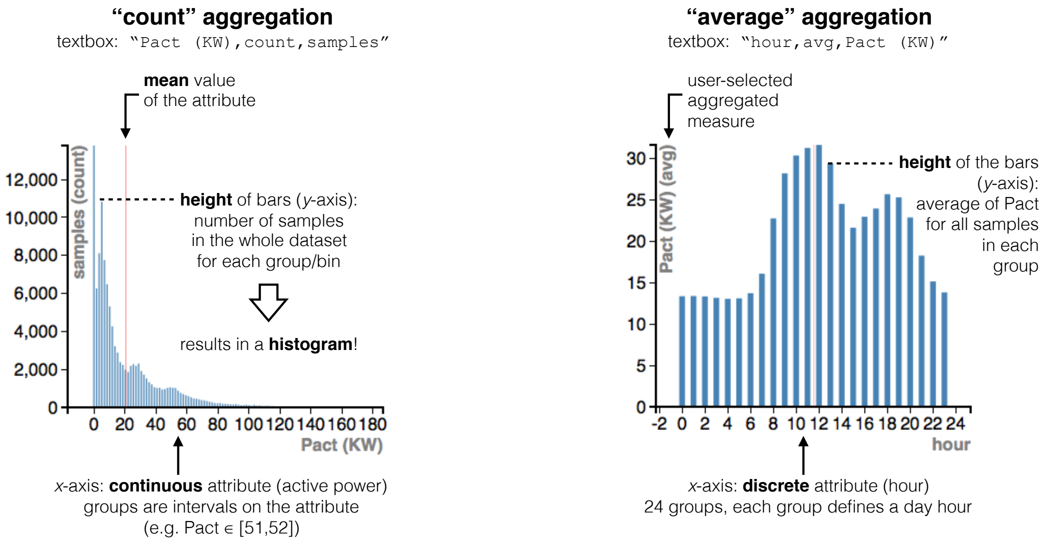

Figure 4: Description of the elements in a chart.

4.4. The web based interface

Interface layout. The application interface (see Fig. 5) can be easily run by opening an HTML5 page with any modern browser (e.g. Chrome, Safari, Firefox) with javascript SVG render capabilities. The page includes a brief explanation of the interface and the domain problem, as well as a textbox that can be used for individual configuration of the views. At the bottom, a set of 10 coordinated views (see table 1) show the current aggregated values for the attributes being analyzed.

User changes on the view configuration. Using the small text box, the user can quickly change the configuration of any view using a very simple syntax

`attribute,aggregation,measure`

where `attribute` stands for the attribute to be represented, `aggregation` is the type of aggregation {`avg`, `sum`, `count`} and `measure` is the variable being aggregated. This simple syntax allows the user to quickly change from viewing the total demanded energy by months `<month,avg,Pact>`, the histogram of power factor values `<cosPhi,count,samples>` or the average harmonic distorsion on phase “*a*” by weekday `<weekday,avg,THD Van (%)>`.

It must be pointed out that these aggregated values are computed *within a context* given by the current user-defined filters on other attributes (allowing, for instance, to condition the results of aggregation to contexts like “Mondays from 10:00 to 12:00”, for “building 2 on February” or “for temperatures higher than 30°C on working days”) in a completely straightforward manner.

Filtering attributes (brushing). When the user sets the mouse on a view, he/she can drag the mouse to select a filter range on this attribute, that is shown by means of a shaded rectangle covering the selected groups in the associated attribute. This action modifies the σ operation on the cube. For instance, by selecting $\{11, 12, 13, 14\}$ in the view `hour` and then selecting {Mo,Tu} in the view `weekday`, the user makes a selection (filters) on the attributes `hour` and `weekday`, resulting in chained *dice* and *aggregation* operations, namely

$$\mathrm{Agg}_{avg(P_{act})}\Pi_{month}\sigma_{\substack{hour=\{11,12,13,14\}\\ weekday=\{\mathrm{Mo,Tu}\}}} \tag{3}$$

$$\mathrm{Agg}_{sum(P_{act})}\Pi_{building}\sigma_{\substack{hour=\{11,12,13,14\}\\ weekday=\{\mathrm{Mo,Tu}\}}} \tag{4}$$

The previous aggregations describe, respectively, the average power aggregated by month and the accumulated energy demand aggregated by building, both evaluated on Mondays and Tuesdays at peak hours —from 11:00 to 14:00. These operations, therefore, recompute the aggregated values for all views from the “diced” cube that results from user’s brush-filtering gestures.

Note that, unless otherwise stated, the charts provide information aggregated from all buildings, all hours, all weekdays, etc. However, if specific information were needed for, say, the library, it could be easily obtained by filtering this building, e.g.

$$\mathrm{Agg}_{avg(P_{act})}\Pi_{month}\sigma_{building=\{19\}} \tag{5}$$

that would display the monthly average demand of the library.

Fluid interaction. A key factor in the interface is that the previous *dice* and *aggregation* operations are literally computed “on the fly” during the brush gesture, with a very low latency, resulting in coordinated animated transitions of all the views, while the user drags the mouse. This behavior favours a high degree of involvement of the user in the analysis loop. Indeed, the mere posibility of seeing immediately the results for every action that the user does on the interface boosts the ability of elaborating hypothesis based on problem domain knowledge and immediately testing them with a rapid increase of the problem knowledge on every action.

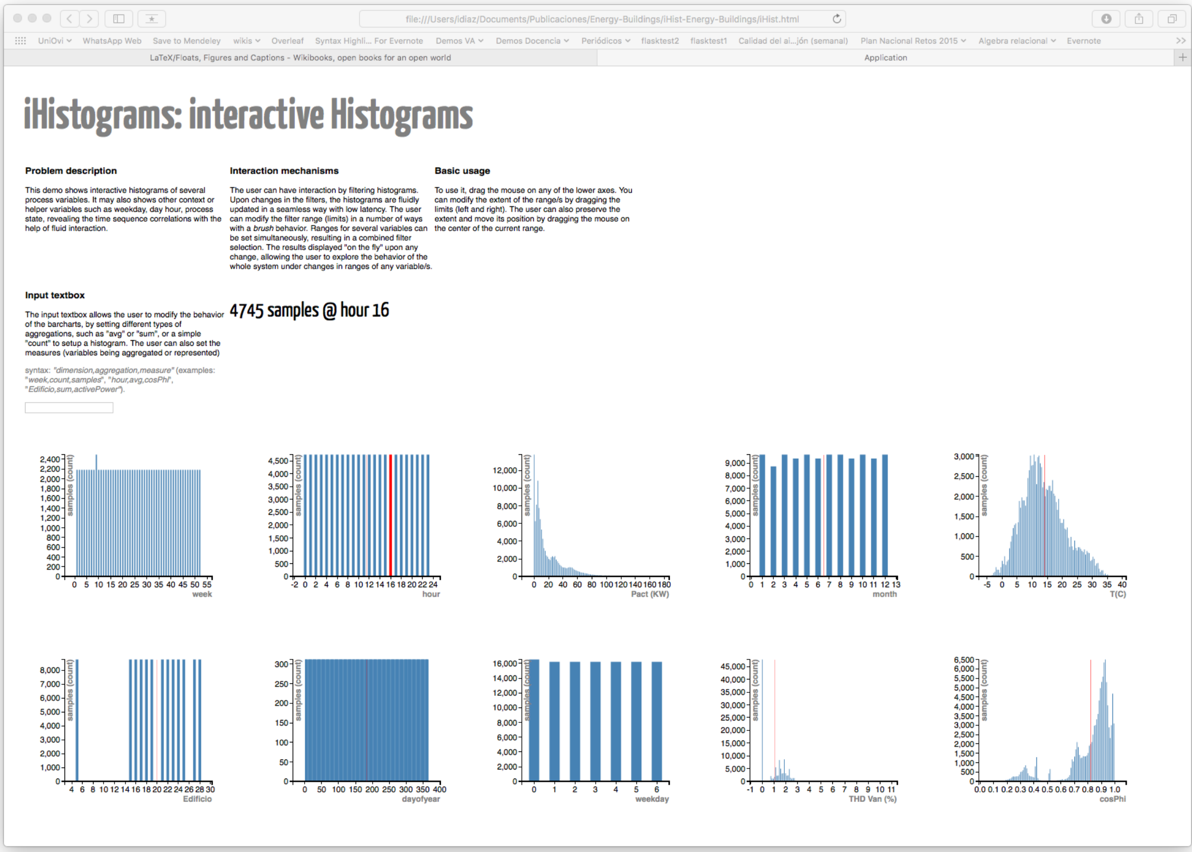

Figure 5: Application snapshot.

5. Results: study of cases

Through this section we will consider a series of particular configurations of the tool for the given dataset that focus on specific cases from a broad spectrum of cases occurred in the campus during the whole year under analysis. Each case of analysis is defined by a specific configuration of the tool, which in turn is described by

1. The filters that the user has defined (i.e. their range), and the attributes to which they apply.
2. The kind of aggregation (*count*, *avg*, *sum*) considered for the representation of each dimension.

For the dataset under analysis, given the two previous *configuration coordinates*, the outcome of the tools (that is, all the displayed bar charts) is fully specified.

5.1. Comparison of the overall consumption of the buildings and temporal behavior

This operation can be easily done by setting the *sum* aggregation to the building chart (attribute id=6), just typing "`Edificio,sum,Pact (KW)`" in the input textbox. The result is shown in Fig. 5. Upon entering the text, the user can quickly observe a change in the building barchart that now shows the overall power consumption in kW for every building.

Additional filters can be added, with interesting results. For instance, by setting a range for `dayofyear`, the user can see the overall consumption in the buildings for this period. By dragging the selection, the user can move this range and the results are smoothly updated, revealing the temporal behavior of the building demands. As a particular case, for instance, a significant increase of the demand in the library (building id=19) is shown when the range contains the month of July, revealing an extended schedule due to a heavy examination period occurring this month.

5.2. Correlation between outside temperature and demand

In certain buildings like Molecular (building id=22) the presence of air conditioning and cooling systems shows a strong correlation between the active power P_{act} and the outside temperature $T(C)$. Such a strong correlation emerges for ranges of small overall active powers, as shown in Figure 7. This behavior can be explained since cooling systems are working permanently, thereby causing a "latent demand" that is isolated when the user excludes the remaining "variable demand" by filtering by low overall active powers.

It is noteworthy that such low values of active power demand occur mainly at low activity periods, that is, early in the morning (from 0:00 to 8:00), late night (from 22:00 to 0:00) and on weekends (`weekday` $= \{5, 6\}$).

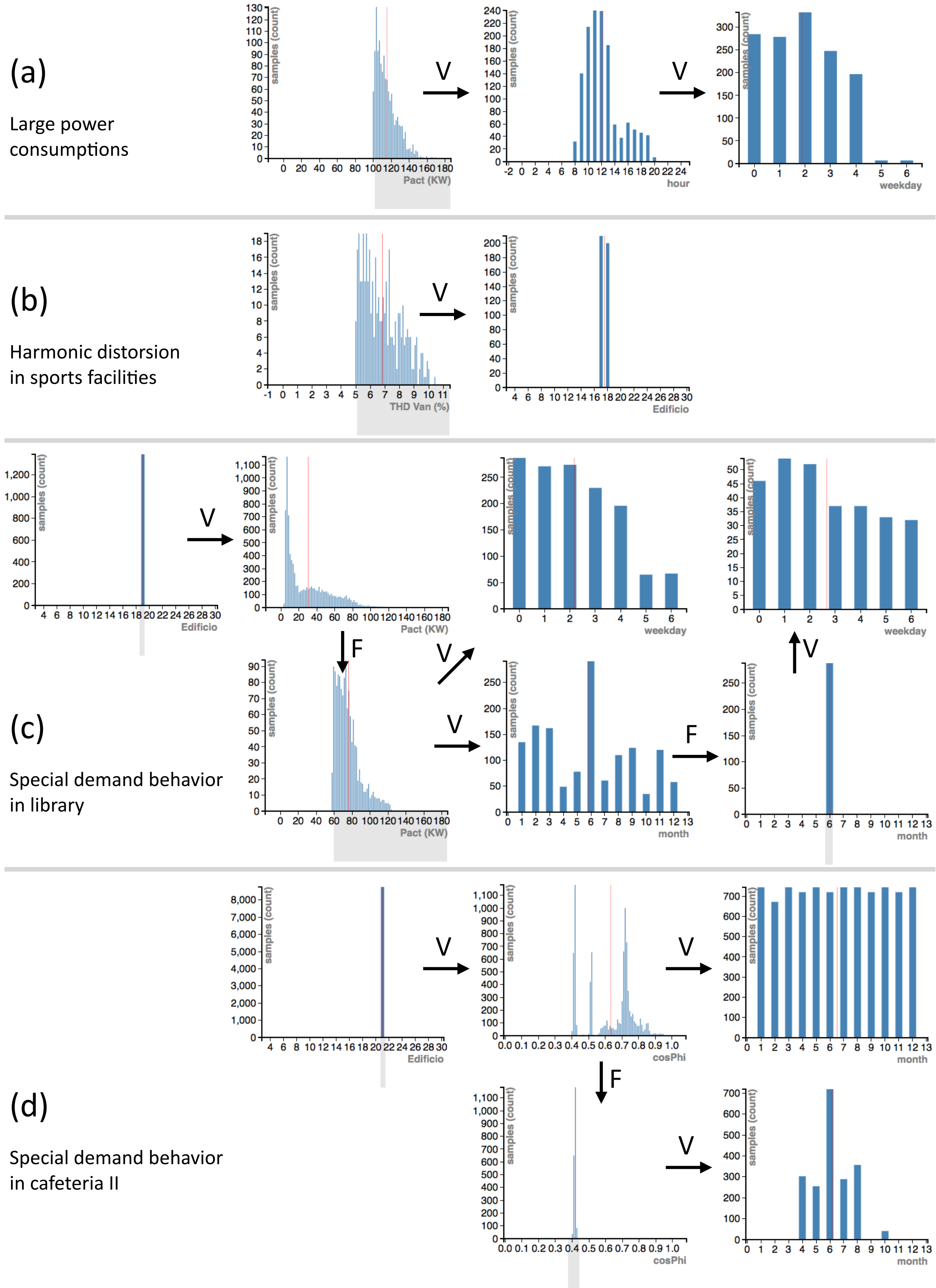

Figure 6: Selected snapshots of some case studies: (a) large power consumptions; (b) harmonic distorsion in sports facilities; (c) special demand behavior of the library; (d) special demand behavior of the cafeteria II. Label “F” indicates a user filter action; label “V” indicates that the user “views results” of the previous filter action.

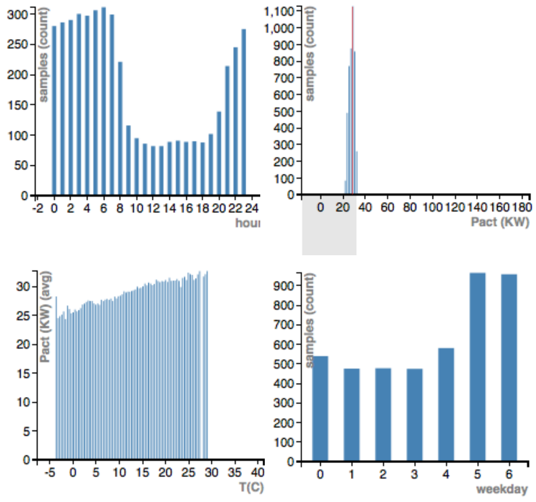

Figure 7: Filtering by low values of P_{act} ("latent demands") in Molecular building (id=22) reveals a strong correlation between outside temperature and active power, due to cooling systems.

5.3. *Large power consumptions*

Filters.

$$P_{act} > 100 \text{ kW}$$

Aggregations. All aggregations set to *count* (histograms)

Analysis. With this configuration, the user restricts the analysis to cases in which the hourly demand has been larger than 100 kW. Not having set any other filter, no further restrictions apply, so the analysis applies to all buildings, all months, all hours, etc. Many observations can be done —see Figure 6 (a):

- The count/histogram of `hour` shows the bimodal consumption profile typical from activity in Spain, with a larger bump corresponding to peak hours centered at 12:00, and a smaller bump corresponding to afternoon hours centered at 16:00.
- The count/histogram of `weekday` shows larger demands on {Mo,Tu,We,Th} (={0,1,2,3} in the interface), a slightly smaller demand on Fr (=4) and lowest demand in {Sa,Su} (={5,6}).

5.4. *Harmonic distorsion in sports facilities*

Filters.

THD $V_{an} > 5\%$

Aggregations. All aggregations set to *count* (histograms)

Analysis. With this configuration the user restricts the analysis to samples where the THD was very high, comparatively to the average. Looking at the histograms, one can quickly get these findings:

1. At the `buildings` chart, it can be quickly seen that the large harmonic distorsion condition only occurs in buildings 17 and 18 (sports pavillion and pelota court) —see Figure 6 (b).
2. In the hour chart this occurs from 18:00 to 21:00. This observation is compatible with the fact that these sports facilities have a peak activity at evening, at the end of the day.
3. The histogram of `month` reveals that this happens mainly in autumn and winter. Within this period the natural light is at its lowest levels.
4. A further look on the sports facilities reveals that the large THD was provoked by nonlinear loads used in lighting equipment, that included high-power mercury-vapor lamps.

5.5. *Special demand behavior of the library*

Filters.

$$\begin{aligned} P_{act} &> 60 \text{ kW} \\ \text{building} &= 19 \quad \text{(library)} \end{aligned}$$

additional filters at

$$\begin{aligned} month &= 6 \\ month &= 9 \end{aligned}$$

Aggregations. All aggregations set to *count* (histograms)

Analysis. The library opens continuously during final exam periods, that include mainly June as the period with largest activity. Specific details of this can be quickly obtained from the histograms —see Figure 6 (c):

- In the `month` histogram the largest peak is found in June, which confirms the previous fact.
- Adding another filter at *month* = 6, the user quickly observe that the demand at weekends is very large, as compared to the weekend demand at other months, when it is smaller or even zero (library closed). This also happens at *month* = 9 (September).
- Filtering by months other than June or September, the activity is restricted to {Mo,Tu,We,Th,Fr}, of which Fr is the day with less demand. Those months that fall out of exam periods (i.e. {Mar, Apr, Oct, Nov, Dec}) the demand at weekends is zero or negligible, which means that the building is closed to the public.

5.6. *Special demand behavior of the cafeteria II*

Filters

$$building = 21 \quad \text{(cafeteria II)}$$

Aggregations. All aggregations set to *count* (histograms)

Analysis. The cafeteria II is a singular facility, which was not open for the general public during the period under analysis, but was used for singular events, mainly on Thursday afternoons. Some singular findings were quickly spotted with our tool.

- Looking at the histogram of the power factor (`cosPhi`), two highly relevant peaks at very low power factors around 0.42 and 0.51 can be observed. Adding a filter on one of these factors ($[0.40, 0.43]$) we find that this occurs: a) for extremely low active power consumptions; and b) for three specific periods at April, May-June and July-August. This happens for periods where the cafeteria was closed to the public and residual power consumption from certain devices (emergency lamps, etc.) led to such low values —see Figure 6 (d).

  Interestingly, moving the `cosPhi` filter to the other peak $[0.50, 0.52]$, we observe now near-zero power demand happening at a different period in `week` or in `dayofyear`, that runs mainly from the second half of August to the beginning of October. The reason is probably due to residual consumptions from different kind of devices this period.

  It should be remarked, however, that the previous values of $\cos\varphi$ are unusually low, even admitting the aforementioned explanations. Another plausible explanation of these facts –this would deserve further analysis– could be a sensor malfunction/misconfiguration. Revealing sensor malfunction is another potential benefit of our proposed tool.

- Setting the active power to values larger than only 3.3 kW, we spot the periods, days and hours of activity, showing activity mainly on Thursdays (*weekday* = 3) in the evening (*hour* from 16:00 to 22:00), being this activity restricted to academic period (month from October to May, and excluding months from June to September).

5.7. Demand behavior of the residence hall

Filters

$$\begin{aligned} \textit{building} &= 24 \quad \text{(residence hall)} \\ P_{act} &> 30\ \text{kW} \end{aligned}$$

Aggregations. All aggregations set to *count* (histograms)

Analysis. The residence hall, providing accomodations for students and some faculty staff, has also a particular behavior.

- By choosing $P_{act} > 30$ the user immediately sees on the *hour* chart a couple of peaks, at 14:00 and 21:00, which are the scheduled food serving times for this building.

- By looking at the *month* chart, a strong increase of the demand can be appreciated in the cold months (from January to March), due to intense activity of heating systems

- Conversely, a drastic drop of the demand is observed from July to September, where students leave to their homes, added to the fact that this building does not have cooling systems.

6. Conclusions

In this paper we have proposed a novel approach for energy analytics in public buildings based on interactive analysis of multiway data, including electrical demand, time, building and environmental data attributes. The proposed approach features fluid interaction for user-driven attribute filtering and definition of aggregation operators, followed by a real-time updated visualization of the results on a web application. Information is presented by means of coordinated barcharts (histograms) that are updated *"on the fly"* upon user changes in attribute filters or in aggregation functions. This allows the user to test any hypothesis, getting immediate feedback for further reconfiguration, resulting in a highly effective user-in-the-loop analysis for pattern finding and correlation analysis.

The results show that the approach is highly effective in several major tasks:

- Efficient analysis of the **temporal behavior** can be achieved by filtering and aggregation according to different attributes of the *time* dimension (`weekday`, `hour`, `month`, etc.) that make full sense for the user and therefore bear useful insight in the results.

- The approach proved also to be useful in exploring correlations of the demand behavior with **environmental variables** by filtering by outside temperature, allowing to spot and quantify the effects of temperature in the demand in different scenarios such as buildings with heat and/or cooling systems.

- It also provides insight in **comparing buildings** allowing aggregation of demand parameters.

- It allowed to spot **special behaviors** by focusing on specific attributes such as the THD (*Total Harmonic Distorsion*) or the *power factor* ($\cos\varphi$), revealing the influence on the demand of special equipment, such as mercury-vapor lighting in sports facilities.

- Also, any **combination of the former analyses** are possible, since the user can create and modify as many filters as attributes, allowing, for instance, to explore the demand for a temperature range, a certain building and/or a weekday or hour range.

Finally, it's worth to note that the information provided by this tool gives the best results if combined with available social science knowledge about the kind and nature of the activities that take place in the buildings.

Acknowledgements

The authors would like to thank financial support from the Spanish Ministry of Economy (MINECO) and FEDER funds from the EU under grant DPI2015-69891-C2-1/2-R.

References

[1] S. Darby, The effectiveness of feedback on energy consumption. A review for DEFRA of the literature on metering, billing and direct displays, Tech. rep. (2006).

[2] C. Fischer, Feedback on household electricity consumption: a tool for saving energy?, Energy Efficiency 1 (1) (2008) 79–104. doi:10.1007/s12053-008-9009-7.

[3] D. Keim, F. Mansmann, J. Schneidewind, J. Thomas, H. Ziegler, Visual analytics: Scope and challenges, Visual Data Mining (2008) 76–90.

[4] K. Ellegård, J. Palm, K. Ellegard, J. Palm, Visualizing energy consumption activities as a tool for making everyday life more sustainable, Applied Energy 88 (5) (2011) 1920–1926. doi:10.1016/j.apenergy.2010.11.019.

[5] J. Rodgers, L. Bartram, Exploring Ambient and Artistic Visualization for Residential Energy Use Feedback, IEEE Transactions on Visualization and Computer Graphics 17 (12) (2011) 2489–2497.

[6] L. Bartram, J. Rodgers, K. Muise, Chasing the negawatt: Visualization for sustainable living, IEEE Computer Graphics and Applications 30 (3) (2010) 8–14. doi:10.1109/MCG.2010.50.

[7] P. C. Wong, Z. Huang, Y. Chen, P. Mackey, S. Jin, Visual Analytics for Power Grid Contingency Analysis, Computer Graphics and Applications, IEEE 34 (1) (2014) 42–51. doi:10.1109/MCG.2014.17.

[8] P. C. Wong, K. Schneider, P. Mackey, H. Foote, G. C. Jr., R. Guttromson, J. Thomas, A Novel Visualization Technique for Electric Power Analytics, IEEE Transactions on Visualization and Computer Graphics 15 (3) (2009) 410–423.

[9] T. J. Overbye, J. Weber, Smart Grid Wide-Area Transmission System Visualization, Engineering 1 (4) (2015) 466–474. doi:10.15302/J-ENG-2015098.

[10] M. Domínguez, J. J. Fuertes, I. Díaz, A. A. Cuadrado, S. Alonso, A. Morán, Analysis of electric power consumption using Self-Organizing Maps, in: Proceedings of the 18th IFAC world congress, Vol. 18, Milano (Italy), 2011, pp. 12213–12218. doi:10.3182/20110828-6-IT-1002.02092.

[11] A. Morán, J. J. Fuertes, M. a. M. Prada, S. Alonso, P. Barrientos, I. Díaz, M. Domínguez, Analysis of electricity consumption profiles in public buildings with dimensionality reduction techniques, Engineering Applications of Artificial Intelligence 26 (8) (2013) 1872–1880. doi:10.1016/j.engappai.2013.04.005.

[12] H. Janetzko, F. Stoffel, S. Mittelstädt, D. A. Keim, Anomaly detection for visual analytics of power consumption data, Computers & Graphics 38 (0) (2014) 27–37. doi:http://dx.doi.org/10.1016/j.cag.2013.10.006.

[13] I. Yarbrough, Q. Sun, D. C. Reeves, K. Hackman, R. Bennett, D. S. Henshel, Visualizing building energy demand for building peak energy analysis, Energy and Buildings 91 (2015) 10–15. doi:10.1016/j.enbuild.2014.11.052.

[14] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, H. Pirahesh, Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals, Data Mining and Knowledge Discovery 1 (1) (1997) 29–53. doi:10.1023/A:1009726021843.

[15] A. Datta, H. Thomas, The cube data model: a conceptual model and algebra for on-line analytical processing in data warehouses, Decision Support Systems 27 (3) (1999) 289–301. doi:10.1016/S0167-9236(99)00052-4.

[16] S. Chaudhuri, U. Dayal, An overview of data warehousing and OLAP technology, ACM SIGMOD Record 26 (1) (1997) 65–74. doi:10.1145/248603.248616.

[17] M. Bostock, V. Ogievetsky, J. Heer, D3: Data-Driven Documents., IEEE Transactions on Visualization and Computer Graphics 17 (12) (2011) 2301–9. doi:10.1109/TVCG.2011.185.