研究論文

鳴門教育大学情報教育ジャーナル No.9 pp.11-14 2012

ネットワークカメラを用いた学内消費電力可視化の試み

曽根直人*，泉　佐也加**

持続可能な社会へ向けてさまざまな場所で消費エネルギーを削減する試みが行われている。本学でもキャンパスにおける消費エネルギーを下げるため，さまざまな活動が実施されている。しかしこれらの活動によってどの程度消費電力を下げられたのか大学構成員に知らしめる手段が提供されていない。学内にはキャンパスの消費電力を記録するデマンド監視装置が設置されているが，その値は装置の前で確認するしかなく，学内で広く利用することはできていない。本稿では，ネットワークカメラを利用しデマンド監視装置の示す値を読み取りグラフ化することで学内へ電力消費の状況を示す試みについて述べる。

［**キーワード**：消費電力，ネットワークカメラ，グラフ化，エネルギー教育］

1. はじめに

2011年3月11日の震災以降，原発の停止によるエネルギー供給不足の懸念が続いている。関東・東北地区では事業所に対して15%以上の節電を求められたことも記憶に新しい。また四国電力管内でも3基ある原発が2012年1月には全て停止しており，電力供給の逼迫が報道されている。本学でもエコアクション21という環境保護活動を行っており，キャンパスをより低エネルギー消費で維持すべく多くの取り組みがなされている。エアコンの室温設定や小まめな消灯の推奨，照明のLED化などはその一例である。しかしこれらの活動によってどの程度消費電力を下げられたのか大学構成員に知らしめる手段は提供されていない。活動を適切に維持していくためにはその効果を知ることがモチベーションを維持するために重要である。

本研究では，学内に於ける省エネ活動の成果を実感できるよう学内の消費電力を計測，グラフ化し，Webページにより公開するシステムを試作した[2]。

2. 消費電力可視化への取り組み

すでに幾つかの大学ではキャンパスの消費電力を可視化する取り組みが行われている。例えば東京大学ではグリーンICTプロジェクトと呼ばれる大規模な取り組みが実施されている。このプロジェクトのWebページでは電力リアルタイム・モニタリング情報や昨年との電力使用状況の比較，その他節電に関するさまざまな情報が集約されている。報道発表資料によればこのプロジェクトによりピーク電力平均44%，使用電力量平均31%削減を達成するなど非常に大きな成果を上げている[1]。

このプロジェクトでは，建物の配電盤にネットワーク接続された電力計を設置することで消費電力データをサンプリングし，集計している。この方法ではリアルタイムに正確なデータの取得が可能であるが，分電盤の工事が必要になるため導入するには工期やコストの問題が発生する。消費電力を測定する方法としては，他にもテーブルタップ型の消費電力測定機器が存在する。しかしこれは装置に接続された機器の消費電力を測定するのみであるため棟全体やキャンパス全体の消費電力を可視化するためには利用できない。工期やコストの問題を避けるため，本研究では大学の消費電力を監視しているデマンド監視装置のデータを利用することとした。デマンド監視は施設課に設置されており，7セグメント赤色LEDにて各種数値が表示されている(図1)。図1の枠で囲った数値は予想デマンド値であり，現在の負荷が続いた場合，時限終了時のデマンド値を表示している。今回はこの予想デマンド値を読み取りグラフ化することで消費電力の状況を知らせることにした。

図１　デマンド監視装置の表示（9時撮影）

＊　鳴門教育大学 大学院 自然・生活系教育部
＊＊　鳴門教育大学 学校教員養成課程 小学校教育専修技術科教育コース

3. 消費電力可視化システム

先に述べたようにデマンド監視装置の値を読み取り，その値を使って学内の消費電力を可視化する。具体的な手順を以下に示す。

1. １分間隔でデマンド監視装置の表示をネットワークカメラにより撮影し，画像ファイルをFTPサーバへアップロードする。
2. アップロードされた画像を処理し，予想デマンド値として表示されている数値を得る。
3. 数値をデータベースに保存する。
4. 保存された数値を使ってグラフを描く。

3.1. デマンド監視装置の撮影

デマンド監視装置の撮影はパナソニック社製のネットワークカメラ BB-HCM371 で行う。カメラはデマンド監視装置の直前に設置し，障害物による撮影不良が発生しないようにしている(図 2)。

図２ デマンド監視装置の撮影

このカメラは撮影した画像を一定間隔で FTP サーバへアップロードする機能を持っている。今回はこの機能を利用し，毎分撮影した画像をFTPサーバへアップロードする設定を行った。その他，この製品はブラウザからパン＆チルト＆ズームを行う機能があるが，撮影画像の位置を変化させないために無効にした。さらにホワイトバランスも自動から固定に設定を変更した。これは日中と夜間では周りの照度が異なるため，撮影画像の色味が変わってしまうことをできる限り防ぐためである。

図３ デマンド監視装置の表示(23時撮影)

図1および図 3 はそれぞれ 9 時，23 時に撮影したデマンド監視装置の表示である。周囲の明るさにより撮影した画像の色調が変化し，LED の色味が異なっていることがわかる。また 23 時に撮影した画像(図 3)はノイズも多くざらついた画像となっている。

3.2. LED の認識

デマンド監視装置を撮影した画像から，予想デマンド値の部分を切り取り，LED の認識処理を行う。LED の認識は当初フリーソフトとして公開されている 7 セグメント表示の数値用 OCR である SSOCR を利用することを考えた。しかし予備実験の結果，SSOCR は入力として二値化画像が必要になるが，画像にノイズが含まれると小数点と誤認識するなど正常な認識結果を得ることが困難であることが判明した。前節で述べたように時間帯により撮影される画像の色調，ノイズは異なるため，SSOCR での認識に必要な品質で画像の二値化を行うことは困難であった。そこで今回は LED 数字の認識を自作プログラムにより行うこととした。

自作プログラムは，Ruby により記述され，デマンド監視装置の画像が一定の位置に固定されていることを利用し，LED セグメントの座標が固定されているものとして認識処理を行う。したがってカメラの位置が変化した場合は手動での調整が必要になる。以下に自作プログラムにおける処理の流れを示す。

1. デマンド監視装置の画像データの取得
2. 画像データから予想デマンド値の部分を切り取り
3. LED の色情報をサンプリング
4. サンプリングした色情報を利用し，画像を二値化
5. 数値の読み取り
6. 数値をデータベースへ保存

LED の色情報サンプリング

撮影された画像を二値化するために LED の色情報を調べる。7 セグメント LED の各セグメントを図 4 に示すようにラベルを割り当てる。③のセグメントは 5, 6 以外の数字では点灯しており，④のセグメントは 2 以外の数字で点灯している。したがって，③，④のセグメントの位置をサンプリングすることで点灯している LED の色情報が得られる。予想デマンド値は 4 桁の表示部分があるが，最上位は表示されていない場合が多いため，残り 3 桁の LED について③，④セグメントの

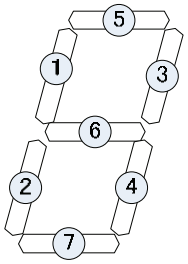

図４ ７セグメント LED へのラベル

座標をサンプリングし，点灯しているセグメントの色情報を調べる。LED が点灯しているか否かは注目しているピクセルの色情報が式(1)の条件を満たす場合とした。ピクセルの色成分は 16bit の値で得られる。

赤成分>50000 || (赤成分>30000 && 緑>30000) (1)

点灯していると判断したピクセルの色情報を平均し，LED の色情報とする。

画像の二値化

得られた LED の色情報を元に切り抜いた画像の各ピクセルが式(2)を満たすか確認し，満たす場合つまりLEDが点灯していると判断できる部分は黒，それ以外は白に変換することで二値化を行う。

Abs(ピクセルの赤成分−LED の赤成分) < 10000 &&
Abs(ピクセルの緑成分−LED の緑成分) < 20000 && (2)
Abs(ピクセルの青成分−LED の青成分) < 20000

予想デマンド値の画像を図５にそれを二値化した画像を図６に示す。

図５ 予想デマンド値

図６ 二値化された予想デマンド値

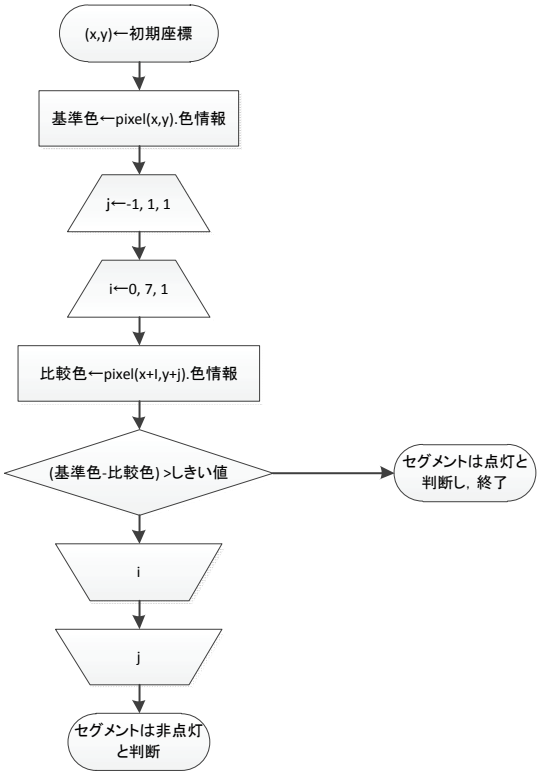

図７ 数値の読み取りアルゴリズム

数値の読み取り

二値化した画像から１桁ずつ数字を読み取る。数字の読み取りは LED のどのセグメントが点灯しているか図7に示すアルゴリズムにより判断し，その情報から表示されている数字を求める。初期座標は読み取るセグメント毎に設定しておく。また，横向きのセグメント(図 4 の⑤，⑥，⑦)は比較色を求める座標を pixel(x+j，y+i)に変更する。

上記のアルゴリズムにより，セグメントの点灯を確認するため，画像をスキャンしたピクセル部分をピンク色に置き換えた画像を図8に示す。

図８ セグメント認識のためのスキャン

図8を見ると，1ライン分スキャンした際にセグメントの点灯が判断できなかった場合，次のラインをスキャンしているためセグメントが点灯していない部分はスキャン跡を示すピンクの線分が太くなっていることがわかる。

数字の認識は配列を用意し，セグメントのラベル順にセグメントが点灯している場合はTrue，それ以外はFalseを格納し，その配列に保存されたパターンが表1のパターン中，どの数字と一致するかで行う。配列のパターンが表1と一致しなかった場合は数字の認識に失敗しており，数字の代わりに“*”を返す。このような単純なアルゴリズムでもカメラが固定されており，毎回同じ位置に認識すべき LED が表示されている場合には，ほぼ完全数値の読み取りに成功している。2012 年 2 月 1 日から 14 日までの 2 週間では全 4032 回の読み取りに対して，数字の認識に失敗したのは 1 回のみであった。

表１ ７セグメント LED の点灯パターン

数字	セグメントのラベル						
	1	2	3	4	5	6	7
0	F	F	F	F	F	F	F
1	F	F	T	T	F	F	F
2	F	T	T	F	T	T	T
3	F	F	T	T	T	T	T
4	T	F	F	T	F	T	F
5	T	F	F	T	T	T	T
6	T	T	F	T	T	T	T
7	T	F	T	T	T	F	F
8	T	T	T	T	T	T	T
9	T	F	T	T	T	T	T
0	T	T	T	T	T	F	T

各桁の数字を読み取り，得られた数値を時刻情報と共に SQLite3 のデータベースに保存している[3]。

3.3. 消費電力の可視化

可視化はLEDの認識とは別のプログラムで処理している。可視化プログラムも Ruby により作成しており，グラフ描画には Ruby の拡張パッケージである Gruff を用いた[4]。

現在の仕様では図9に示すように，

- 当日の予想デマンド値（黄色）
- 前日の予想デマンド値（青色）
- 1 週間を平均した予想デマンド値（緑色）

を折れ線グラフ化している。グラフ以外に最新の予想デマンド値，当日最大の予想デマンド値およびそれぞれの値を得た時刻を表示している。

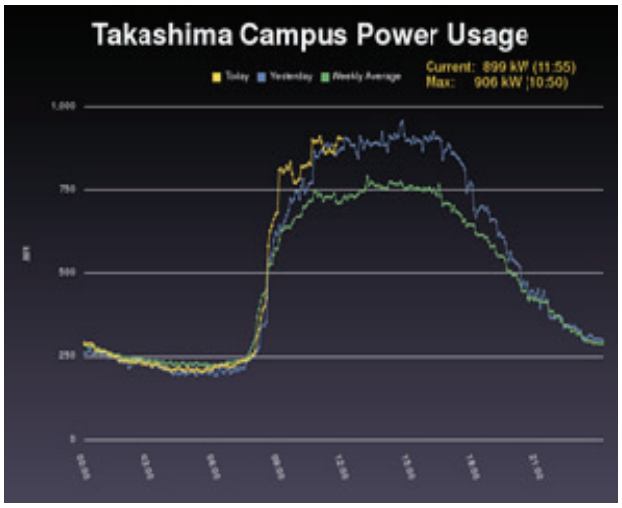

図９ 予想デマンド値のグラフ化

予想デマンド値読み取りプログラムにより，読み取られた値は SQLite3 形式のデータベースに保存されている。グラフ化プログラムではデータベースから必要なデータ，現在の仕様では過去一週間分のデータを読み出しグラフを描画する。Gruff ではグラフに描きたいデータを配列に保存しグラフ描画のメソッドを実行することでグラフ画像(png)が得られる。予想デマンド値の現在値および最大値は Gruff のみでは処理できないため，グラフ画像ファイルを読み込み RMagick(ImageMagick)の annotate メソッドを用いてグラフ画像に追記している[5,6]。

4. まとめ

電力削減への学内の取り組みを「見える化」するためにデマンド監視装置に表示される予想デマンド値を読み取りグラフ化する試みを行った。今回のシステムは非常に簡易なシステムではあるが，消費電力を公開する目的には十分な機能を持つと考える。

今後の展開としては，教職員や学生へより関心を持ってもらうため見せ方の工夫や twitter やメールを使った情報提供などが必要であると考える。また東大グリーン ICT プロジェクトの成果をベースとしたスマートグリッド用プロトコル IEEE1888 では「分析」，「見える化」，「制御」，「ストレージ」といった機能が利用できる[7]。この IEEE1888 が利用できれば応用の幅が広がるため，今回のシステムとの連係の可能を検討したい。

参考文献

[1] 国立大学法人東京大学　東大グリーンICTプロジェクト：“報道発表資料 2011 年 9 月 21 日”，http://www.gutp.jp/news/pdf/20110921-pressrelease.pdf.

[2] 泉 佐也加，曽根 直人：“節電に向けた学内電力計の可視化への試み”,日本産業技術教育学会第 27 回四国支部大会 講演要旨集, p.11 (2011).

[3] SQLite: “SQLite Home Page”，“http://www.sqlite.org/”.

[4] Geoffrey Grosenbach：“Geoffrey Grosenbach”，“http://nubyonrails.com/pages/gruff”.

[5] RMagic: “RMagick Download Page”，“http://rmagick.rubyforge.org/”.

[6] ImageMagick:”ImageMagick: Convert, Edit, Or Compose Bitmap Images”,”http://www.imagemagick.org/”.

[7] 国立大学法人東京大学　東大グリーン ICT プロジェクト: ”IEEE1888 (FIAP)誕生の背景”，” http://www.gutp.jp/fiap/”