SQL ライクな操作言語を用いた可視化システムの応用とライフログ分析のためのデータ操作高速化

能條 太悟 [1]　杉本 航洋 [2]　富井 尚志 [3]

我々は，複数の属性からなるデータを PCP（Parallel Coordinates Plot）により可視化し，その状態を SQL ライクな独自の言語表現で保存・再現可能なシステム (PC)^{2}DV （Parallel Coordinates Plot Commutative Data Visualizer）を提案してきた．本稿では特に，ライフログ分析を目的としたデータ操作支援を試みる．まず，電力データの分析に (PC)^{2}DV を用いる例を挙げ，有用性を示す．日常的に取得される時系列データを PIVOT 操作により周期データに変換し，PCP を用いて可視化する．これにより，ライフログデータの持つ周期的な情報を可視化することができる．次に，(PC)^{2}DV においてライフログ分析のためのデータ操作の高速化を行う．(PC)^{2}DV はデータを PCP により可視化するため，データの件数が増えると線の数が多くなり，描画やデータ操作に時間を要するという課題があった．この課題に対し，GROUP BY 集約によるデータ件数の削減により描画の高速化を図る．高速化の検証として，従来手法との描画時間の定量的な比較を行う．これにより，(PC)^{2}DV においてビッグデータ分析が可能であることを示す．

1　はじめに

近年，センサデータの取得や各省庁等による様々なデータの公開により，実世界の状況がデータとして取得可能になった．また，ストレージの大容量化，低価格化により，取得したデータ全てを蓄積・保存することが可能になった．これらのデータを分析し，活用することが考えられる．その一例として，日常的な生活や活動を記録した「ライフログ」をデータとして取得することが挙げられる．その人，その場所ならではの特徴を含蓄するライフログを分析し，そのデータ固有の知見を示すことは，個々の事例においても社会全体においても有用である．今後の社会では，様々なソースから多様なデータが取得できるようになると考えられる．このような，出処や形式，粒度，種類が異なるデータを同じ時間や場所で結合して分析を行うことは重要である．すなわち，関係データベース（RDB：Relational Database）と SQL を用いた分析が有効であるといえる．ここで，SQL のクエリ結果は表形式であり，その形式のまま知見を得ることは容易ではない．そのため，データを可視化し分析を行うことが必要であると考えられる．

複数の属性からなるデータを可視化する手法の一つに平行座標プロット（PCP：Parallel Coordinates Plot）[1,2] がある．PCP ではデータの属性を平行な軸に割り当て，軸を結ぶ折れ線一本でデータ一件を表す．PCP でデータ全件を可視化することにより，大まかな属性間の相関やクラスタ，外れ値を容易に把握することができる．

ここで，PCP で可視化するデータは表形式であるため，PCP 上で表に対する操作，すなわち SQL と同等の操作が可能である．この特徴に注目し，我々は複数の属性からなるデータを PCP により可視化し，その状態を SQL ライクな言語表現により保存・再現することが可能なシステム (PC)^{2}DV（Parallel Coordinates Plot Commutative Data Visualizer）を提案してきた [3–6]．(PC)^{2}DV は GUI を有するデータ分析支援ツールであり，PCP 上で選択・射影・結合といった基本的な関係代数演算によるデータ操作がアドホックなクエリとして実行可能である．また分析の過程において，分析者はデータ操作やデータ可視化の状態を，SQL ライクな独自の言語表現 (PC)^{2}L（Parallel Coordinates Plot Commutative Language）により保存・再現することが可能である．(PC)^{2}L により，SQL を熟知する分析者に対して試行錯誤を伴うデータ分析を支援する．これらにより，リアルタイムにデータ操作が反映される GUI と，その状態と等価である論理的な言語表現を相互に活用した分析が可能である．

本稿では (PC)^{2}DV の有用性を示すライフログデータ分析の例題として，建物の需要電力データと太陽光発電（PV：Photovoltaic power generation）の発電電力データを対象とした分析を行う．日常的に取得される時系列データ，すなわちライフログデータを一般的なデータ操作である PIVOT の操作により 1 日ごとの周期データに変換し，PCP を用いて可視化する．分析の過程において，電力データと気象データという異なるデータを結合し，GROUP BY 集約により PCP の折れ線をグループごとの代表値を表す折れ線一本にまとめるデータ操作を行う．これにより，異なるデータソースから得られたデータを結合するという SQL の特徴的なデータ操作を GUI でデータを可視化しながら行うことのできる (PC)^{2}DV の有用性を示す．

ここで，データ解析例において実行したデータ操作である GROUP BY 集約に着目する．GROUP BY 集約によりデータから代表値を求め，PCP で描画する折れ線の数を意図的に削減することにより，データ全体の傾向やグループごとの特徴を損なわないまま，より高速にデータを可視化することができる．これは特に，日常的に取得できるライフログといった大量のデータを可視化・分析するときに有効であると考えられる．そこで，本稿では (PC)^{2}DV において大量のデータの分析を可能にするために，データ削減によるデータ描画の高速化を行う．GROUP BY 集約による可視化データの削減により高速化を図り，その検証として，従来手法 [6] とのデータ処理および描画時間の定量的な比較を行う．先行研究における (PC)^{2}DV の実装では可視化・分析のためのデータをブラウザのローカルメモリに保存し，そのデータを保持したままデータを処理する方法がとられた．そのため，

[1] 非会員　横浜国立大学 大学院環境情報学府 情報環境専攻 (現 株式会社富士テクニカルリサーチ)
[2] 学生会員　横浜国立大学 大学院環境情報学府 情報環境専攻
sugimoto-koyo-zf@ynu.jp
[3] 正会員　横浜国立大学 大学院環境情報研究院
tommy@ynu.ac.jp

データを可視化・分析する際に，データ量の増加に伴いメモリ上でのデータ処理や PCP の線の描画に要する時間が増加し，インタラクティブ性が失われることが問題であった．この先行研究と本稿における提案手法の定量的な比較により，可視化データ削減による高速化の有効性とビッグデータの分析可能性を示す．

2 関連研究

2.1 データ可視化とデータ分析

PCP は 1985 年，Inselberg によって初めて概念が定義された [1]．それ以降，PCP を用いたデータ分析が行われている．Tong らは，センサにより取得した身体の動作データを PCP により可視化・分析をし，評価実験により PCP がデータを解釈するために有効であることを示した [7].

また，PCP に関する議論がなされている．Johansson らによれば，PCP の研究カテゴリーは次の 4 つに分類される [2]：

1. PCP の軸のレイアウトの評価
2. PCP の乱雑さ（clutter）の削減方法の比較
3. PCP の実用性の提示
4. PCP と他のデータ分析手法との比較

上記のように，PCP の見せ方に関して議論がなされているものがほとんどであり，PCP の操作過程に着目した議論はされていない．

複数の属性からなるデータを可視化するその他の一般的な手法として，複数の散布図を表示する散布図行列が挙げられる [8]．散布図行列は，2 つの属性間の相関を直感的に把握できるが，散布図数が属性数の 2 乗に比例して増加する．そのため，分析過程でデータに対し結合の操作を行うことには不向きであるといえる．また，Bouali らは，対話型遺伝的アルゴリズムにより可視化手法の推薦を行い，データや利用者の要求に応じてより適切な可視化手法の選択を支援するシステムを提案した [9]．$(PC)^2DV$ は関係代数演算における選択・射影・結合が表現可能な可視化システムであるため，データ一件を 1 本の折れ線で表し，詳細に参照・分析可能である PCP が適切である．

データを可視化し，分析を行う研究（Visual Analytics）が盛んに行われている．Cui [10] による分類では，PCP を用いたデータ分析および本研究は多次元データをアルゴリズムに基づき変形し二次元空間で可視化する “Multi-Dimensional-Transformation-2D” かつ，データ操作によりデータや可視化空間を探索する “Exploratory-Oriented” に分類される．Visual Analytics の中で，我々の手法と同様にインタラクティブな操作と PCP による可視化を組み合わせた可視化・分析手法の提案がされている．Itoh らは，属性軸間の相関に基づいてインタラクティブに次元削減を行い，PCP から所望する情報の発見を支援するシステムを構築した [11]．Zhou らは，エントロピーの概念を導入することで，PCP の属性軸の整列順序をクラスタに基づいて決定する手法を提案した [12]．Bok らは，任意の属性値を基準としたデータの分布を表すヒストグラム（PHP：Parallel Histogram Plot）を PCP 上に表示し，PCP の軸上のデータの分布や属性間の相関の把握を支援する手法を提案した [13]．Gruendl らは，時間を新たな次元と考え，PCP の二軸間の奥行き方向に時間軸を導入し，PCP と時系列プロットを統合した，時間依存のデータを可視化・分析する手法を提案した [14]．これらの研究と比較して我々は，PCP 上でのデータ操作が関係代数演算と同等であり，可視化の状態と言語が可換である点に着目している．

また，インタラクティブにデータ可視化を行う研究の中で，大量のデータを対象とした研究については多くの事例が見られる [15]．中でも，関係データベーススキーマに基づくデータに対し，GUI 上でクエリの記述や複数の可視化の連携を可能にし，データ解析を支援する研究も複数行われている．Derthick らは，データオブジェクトを可視化しつつ，インタラクティブに GUI でクエリが表現可能な環境を構築した [16]．North らは，データの可視化と，表示した複数の可視化間の連携をユーザーが自由に変更可能なインターフェースの構築を行った [17]．杉渕らは，クエリフローモデルによる直感的かつ段階的なクエリが構築可能な GUI を機能として備えた，可視化フレームワークを実装した [18]．これらの研究は，可視化とクエリを GUI 上で連携させることで，インタラクティブなデータ解析を支援する点では，我々と立場が同じといえる．しかし，これらの研究は「データベースに習熟していないデータ解析者を支援する」点を重視している．本研究は，「SQL に類似した言語を用いてデータ分析の過程の任意の状態を保存し，データベースや SQL を熟知するデータ分析者を支援する」ことを目的としているため，これらの研究とは立場が異なる．

2.2 データ操作過程管理（Data Provenance）

データやシステムの操作過程を管理する研究（Provenance）が行われている [19]．Herschel らは文献 [19] 内で，特にデータやシステム，プログラミングコードなどの操作過程や操作の意図を保存することは，複雑なデータ処理を支援するために重要なことであると述べている．さらに，分析結果データの操作過程や操作の意図を示すことは，SQL のような関係代数演算をサポートする問合せ言語で記述することが有効であるとも述べている．この点において，$(PC)^2L$ を用いて $(PC)^2DV$ のデータの操作過程の状態を保存することは有効な手段であるといえる．

また，データやシステムの操作過程を保存することでユーザの支援を行う手法が提案されている．Waldner らは，PC のアプリケーションの閲覧履歴や操作履歴を保存し，それらを時系列が理解できるように可視化することで，ユーザが過去に行った情報探索の詳細を再現する支援を行った [20]．Mindek らは，画像データと，分析過程で利用する他のソースのデータを同時に表示し，分析者の文脈を含蓄したスナップショットを保存することで，シミュレーションデータの可視化や文書分析の支援を行った [21]．Gratzl らは，PCP やヒートマップ，散布図行列など様々な可視化手法を組み合わせて複数のソースから得られたデータとその解析過程を可視化し，データ解析の支援を行った [22]．これらの手法と比較して我々の手法は，「可視化システムのデータ解析過程を可視化して見せる」のではなく，「SQL に類似した言語を用いてデータ分析の過程の任意の状態を保存し，問合せ言語として一般的な SQL を熟知するデータ分析者を支援する」ものであり，立場が異なる．また，言語を用いて状態を保存することにより，言

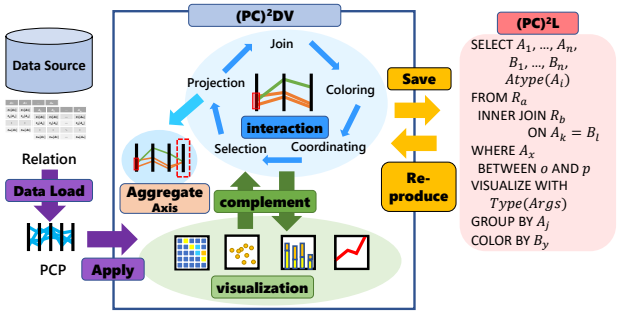


図 1　(PC)²DV の概要図

語の一部を書き換えるだけで容易にデータ分析の改善をすることができる．その点でこれらの研究と比較して優位性をもつ．

また，Holger らは分析プロセスの可視化状態と操作を，検索可能なグラフ構造として保存・再現する分析支援システムを提案した [23]．彼らの手法は，分析における過去の状態を言語情報により保存・再現可能である，という点が我々共通する．しかし，彼らは「グラフ構造により分析の履歴を示し」，操作を含めた分析過程を「検索可能な形ですべて保存する」一方で，我々の手法は「可視化状態と言語が可換である」という点を重視し，分析過程の「任意の状態を保存する」ことにより支援を行うものである．この点において，我々の研究とは立場が異なる．

3　SQL ライクな操作言語を用いた可視化システム (PC)²DV

我々は先行研究 [3–6] において (PC)²DV を提案してきた．(PC)²DV は，SQL ライクな操作言語を用いた可視化システムである．本章では (PC)²DV の概要とその表示例を述べる．

3.1　(PC)²DV の概要

(PC)²DV の概要図を図 1 に示す．(PC)²DV では，SQL ライクな独自の言語 (PC)²L によりデータ操作を行った状態を保存・再現することが可能である．このシステムでは，以下のようなデータ操作手順により分析を行う分析者に対して支援を行うことを想定する．

1. 任意のデータソースへ接続し，リレーションを PCP により可視化する．
2. PCP を補完する形で，任意のグラフの描画を行う．
3. 可視化結果をもとに，PCP 上でインタラクション（データ操作）を行う．その際，データ操作結果はリアルタイムに（2）で表示したグラフに反映される．
4. データ分析者が任意に，(2)，(3) の分析過程の中間状態（スナップショット）を (PC)²L で保存する．
5. （2）から（4）を繰り返す．その際，過去の状態に戻りたい場合は該当する (PC)²L を入力し，その状態を再現する．
6. データ分析者が所望の可視化結果を獲得する．

3.2　(PC)²DV の表示例

(PC)²DV の表示例を図 2 に示す．実装システムは，先行研究 [3–6] と同様に，個々のユーザの環境に依存しないようにするために Web ブラウザを通して利用できるように構築した．開発言語は，サーバサイドの処理に PHP，フロントエンド側の処理に HTML，CSS，JavaScript を使用した．

図 2 の A：PCP View では，任意のデータソースから取得したリレーションを可視化した PCP が表示される．ここでは先行研究 [3] で定義したデータに対するインタラクションのうち，選択（Selection），色分け（Coloring），軸配置（Coordinating）の操作が利用可能である．PCP の各軸上を上下にドラッグし範囲選択をすることで，範囲内に含まれる折れ線のみが PCP に表示される．これにより選択（Selection）の操作が可能である．各軸の軸名をクリックすることで，その属性の値を基準にした色分け（Coloring）の操作が可能である．なお，色分けの色は指定した軸の属性値が数値の場合は昇順に「青，緑，黄，赤」がグラデーションとなるよう配色され，文字列の場合は固定の色が割り振られる．PCP の各軸を左右にドラッグ&ドロップすることで軸の配置を変更することができる．これにより軸配置（Coordinating）の操作が可能である．

図 2 の B：Graph View では，(PC)²L で指定したグラフが表示される．グラフ上には A，C 上で行ったインタラクションを反映したデータセットが表示される．

図 2 の C：Join & Projection Button では，横一行がリレーション一つに対応するボタンが表示される．このボタンでは，先行研究 [3] で定義したデータに対するインタラクションのうち，結合（Join）と射影（Projection）の操作が利用可能である．各行の最左部には，リレーション名のボタンが配置されている．このボタンをクリックすることで，結合条件を選択する画面が表示される．そこで指定した結合条件に応じた結合（Join）の操作を実行することが可能である．なお，既に Join されているリレーション名のボタンを再度クリックすることで，Join を解除することができる．このボタンの右側には，リレーションの属性名のボタンが配置されている．これをクリックすることでその属性に対応した PCP の軸の表示/非表示を切り替えることができる．これにより射影（Projection）の操作が可能である．

図 2 の D：Table View & (PC)²L Interface では，A，C 上で行ったインタラクションを反映したデータセットのテーブル表示や (PC)²L の入出力を行うユーザインターフェースを持つ．(PC)²DV のデータ可視化状態と可換な (PC)²L の出力や，テキストボックスでの (PC)²L の入力と編集，txt 形式での (PC)²L の出力が可能である．(PC)²DV ではユーザの操作に応じて，まず可視化するデータセットを作成し，その後で PCP やグラフなど GUI の描画を行う．なお，先行研究の実装ではデータソースに接続して取得したデータをブラウザのメモリに JSON の形式で保存し，それを保持したまま (PC)²DV のプログラム内でデータ操作を行い，可視化するデータを変更する．ユーザが GUI や言語を介して操作を行うと，メモリに保存したデータを参照し，可視化するデータセットを作成する．

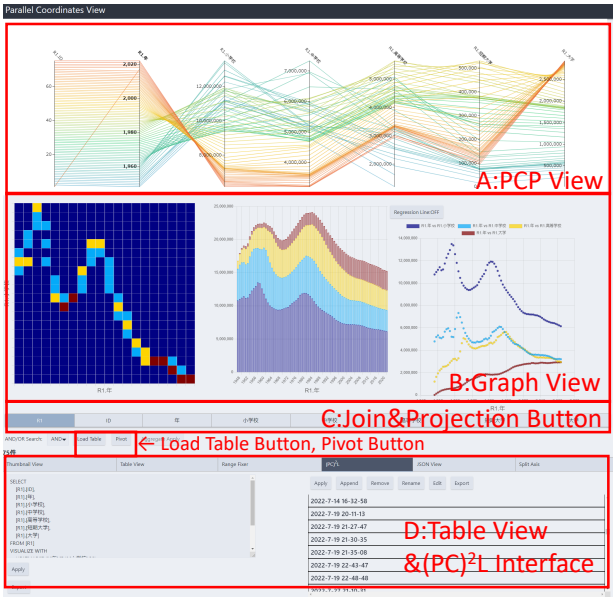


図 2　(PC)^{2}DV の表示例

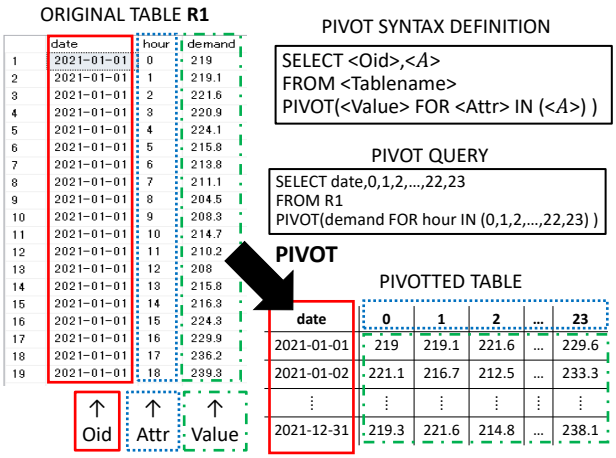


図 3　参考文献 [25–27] による PIVOT の文法定義と PIVOT の概要

4　PIVOT によるデータ変換

本稿では，(PC)^{2}DV によりライフログ分析をすることを目的とする．そのために，一般的なデータ操作である転置（PIVOT）[24] の操作を (PC)^{2}DV に導入した．(PC)^{2}DV による PIVOT の操作を用いたライフログ分析は 5 章で示す．

4.1　PIVOT の概要と参考文献における定義

PIVOT とはテーブルにおいて，行を削減し列に変換する操作である．この操作により，データを束ね件数を減らしつつ属性数を増やすことでデータ一件の単位を変えることが可能である．図 3 に PIVOT の概要と文法の定義を示す．参考文献 [25–27] によると，PIVOT の操作は以下の文法により記述される．

SELECT ＜ Oid ＞，＜ A ＞
FROM ＜ Tablename ＞
PIVOT（＜ Value ＞ FOR ＜ Attr ＞ IN （＜ A ＞））

＜ Tablename ＞とは PIVOT を行うテーブルの名前である．図 3 では属性 date，hour，demand からなるテーブル R1 に対して PIVOT の操作を行うため，＜ Tablename ＞は R1 である．＜ Oid ＞とは PIVOT 後のテーブルの主キーとなる属性であり，＜ Oid ＞の値ごとにデータが束ねられる．図 3 における＜ Oid ＞となる属性は date である．PIVOT 後のテーブルはデータが日ごとに束ねられ，レコード一件が日ごとのデータとなる（図中赤実線）．＜ Attr ＞とは PIVOT 後のテーブルのカラム名となる値を持つ属性であり，＜ A ＞は＜ Attr ＞の属性値の集合である．＜ Attr ＞となる属性の，属性値集合＜ A ＞の値が PIVOT 後のテーブルのカラム名となる．図 3 では＜ Attr ＞となる属性は hour であり，＜ A ＞は hour の値（0，1，2，...，23）である．この値が PIVOT 後のテーブルのカラムとして並ぶ（図中青点線）．＜ Value ＞とは PIVOT 後のテーブルの各カラムの値となる値を持つ属性である．図 3 では，＜ Value ＞となる属性は demand である．例えば，属性 date の値が 2021-01-01 かつ属性 hour の値が 0 のレコードの，属性 demand の値は 219 である．この値が，PIVOT 後のテーブルの属性 date の値が 2021-01-01 のレコードの属性 0 の値となる．このようにして，date ごとに hour ごとの demand の値が PIVOT のテーブルの値として並ぶ（図中緑一点鎖線）．

4.2　(PC)^{2}DV における PIVOT の定義

4.1 節の定義をもとに我々は，(PC)^{2}L における PIVOT を (PC)^{2}DV において必要な情報のみで定義するために，PIVOT の記法を以下のように変え，(PC)^{2}DV に導入した．なお，各記号は 4.1 節の定義に則る．

SELECT ＜ Oid ＞，＜ A ＞
FROM ＜ Tablename ＞
PIVOT（＜ Value ＞ FOR ＜ Attr ＞）

参考文献の PIVOT の定義と (PC)^{2}L の PIVOT の定義には一点のみ相違点がある．これは，属性値集合「IN（＜ A ＞）」の部分をを削除した点である．この理由として，以下の三点が挙げられる．ただし，*Attr* は＜ Attr ＞の全ての属性値の集合である．

1. A=*Attr* の場合，自明のため列挙の必要がない
2. A ⊂ *Attr* の場合，Selection の操作により実現可能
3. 属性値の列挙が SELECT 句での記述と重複し，冗長である

以上により，参考文献をもとに (PC)^{2}DV において必要な情報のみで (PC)^{2}L の PIVOT を定義した．

5　(PC)^{2}DV を用いたライフログ分析

5.1　分析の目的

本章では，3 章で述べた (PC)^{2}DV を用いて日々取得される電力データを分析する．建物の 1 時間ごとの電力データと PV の発電電力データを PIVOT により 1 日ごとの電力データに変換し PCP で日々の電力波形を可視化する．また，変換したデータを日ごとの天気や季節のデータと結合し，季節ごと，天気ごとに電力データを集約する．これにより，季節ごと，天気ごとの建物の需要電力や PV の発電電力の違いを代表値を用いて可視化する．

表 1　リレーション Building_Demand の属性

属性名	説明
date	日付（yyyy-mm-dd）
hour	時間（0〜23）
demand	1 時間ごとの建物の需要電力量（kWh/h）

表 2　リレーション Solar_Generate の属性

属性名	説明
date	日付（yyyy-mm-dd）
hour	時間（0〜23）
generate	1 時間ごとの PV の発電電力量（kWh/h）

表 3　リレーション Daily_Weather の属性

属性名	説明
date	日付（yyyy-mm-dd）
season	季節（四季）
weather	日ごとの天気（晴，曇，雨，雪）

建物の需要電力は季節や天気ごとに異なり，PV の発電電力は天気に大きく影響される．これらのことをデータから説明することで，ライフログデータ分析における $(PC)^2DV$ の有用性を示す．

5.2　使用データと分析プロセス

本稿では，建物の需要電力データと PV の発電電力データを対象としたデータ分析を行う．建物の需要電力データとして，横浜国立大学の研究棟 3 棟の 1 時間ごとの需要電力の実データ[*1]を使用する．このリレーション Building_Demand を表 1 に示す．また，PV の発電電力データとして，建物の屋上全体に太陽光パネルが設置されている想定のもと，日射量のオープンデータ[*2]により算出した，実データに基づく仮想データを使用する．このリレーション Solar_Generate を表 2 に示す．いずれもデータ取得期間は 2021 年 1 月 1 日から 2021 年 12 月 31 日までであり，データ件数は 24 時間× 365 日で 8760 件である．これらのデータを季節ごと，天気ごとに集約するためのデータとして，気象庁[*3]から取得した神奈川県横浜市の日ごとの天気と月に対応する季節のデータを用いる．このリレーション Daily_Weather を表 3 に示す．リレーション Buliding_Demand，Solar_Generate について，$(PC)^2DV$ を用いた以下のプロセスで可視化・分析を行う．

1. データソースに接続し，クエリを実行することでデータを取得する．取得したデータを PCP で可視化する．
2. 可視化されたデータに対して，PIVOT の操作を行い，日ごとのデータに変換する．
3. 変換されたデータを PCP で可視化する．
4. 同一の尺度で値を比較するため，軸の範囲を統一する．
5. 日付をキーとして，リレーション Daily_Weather を結合する．
6. データを天気ごと，季節ごとに集約し，代表値として 1 時間ごとの平均値を可視化する．

なお，データは Microsoft SQL Server 上のデータベースに格納されている．このデータベースに ODBC 接続し，データを取得する．

5.3　分析例１：需要電力データの分析

まず，建物の需要電力について可視化・分析を行う．取得したデータを可視化した様子とその状態を表す $(PC)^2L$ を図 4 に示す．折れ線は日付（date）を基準に色分けされている．なお，図に示した PCP の可視化状態は同図の下部に記述した $(PC)^2L$ と可換である．これは以降の分析例における PCP を示す図においても同様である．このように，分析における任意の可視化状態が言語表現と可換であるという点が，$(PC)^2DV$ および $(PC)^2L$ の特徴である．しかしながら，この状態からデータの持つ周期的な情報を読み取ることは困難である．

そこで，データに対して PIVOT の操作を行う．図 2 左中央辺りの Pivot Button を押すことで，図 5 のような画面が表示される．この画面から＜ Oid ＞，＜ Attr ＞，＜ Value ＞となる属性をそれぞれ選ぶことで PIVOT の操作を実行することが可能である．ここでは，可視化したデータを日ごとの，1 時間ごとの電力データに変換するため，＜ Oid ＞は date，＜ Attr ＞は時間（hour），＜ Value ＞は需要電力（demand）とした．PIVOT の操作を行った後のデータを PCP により可視化した様子を図 6 に示す．ここで，PCP で可視化してるデータセットが PIVOT の操作により作成されたものであるという情報が，図中に示したように 4.2 節で定義した $(PC)^2L$ の文法で保持されていることがわかる．この状態では各軸の範囲がまばらであるため，軸の範囲を指定し統一する．軸の範囲を 0kWh/h から 550kWh/h までに統一した PCP を図 7 に示す．これにより，1 日の電力推移を波形として把握することができる．この図から，この建物の需要電力は人々の活動に合わせて 7 時ごろから増加し，昼過ぎから夕方の間に大きくなり，夜になるにつれて減少するという傾向があることがわかる．建物の 1 日の電力波形がこのような推移を取ることはごく当然であるが，元の形式のデータからは得られなかった周期的な情報を可視化することができたといえる．また，折れ線の色により，季節ごとの需要電力の違いを把握することができる．この建物の需要電力は，春（緑）と比較して夏（黄色）や冬（橙，青）のほうが概ね高いことや，冬よりも夏のほうが概ね高いことがわかる．このように季節ごとの大まかな違いを把握できたことは，データを PCP で可視化をすることのメリットの一つである．

ここで，季節ごとの違い，天気ごとの違いをより明確に可視化するために，データを季節ごと，天気ごとに集約する．そのためにまず，リレーション Daily_Weather のデータを取得し，日付をキーとして電力データと結合する．図 2C のリレーション名（ここでは，Daily_Weather）のボタンを押すことで図 8 のような画面が表示され，結合の条件を指定し結合の操作を行うことができる．データを日付（date）結合し季節（season）ごと，天気（weather）ごとの平均値を求め可視化した PCP を図 9 に示す．線の色は季節を表す．この図から，１日を通して春（赤）や秋（桃）

[*1] 横浜国立大学施設部，http://shisetsu.ynu.ac.jp/gakugai/shisetsu/（学内限定アクセス）
[*2] 横浜市環境創造局，http://www.city.yokohama.lg.jp/kankyo/
[*3] https://www.data.jma.go.jp/gmd/risk/obsdl/index.php

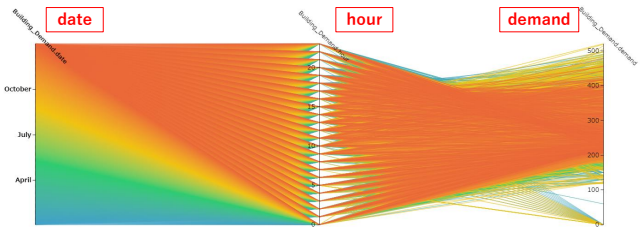


図 4　建物の需要電力量を可視化した PCP

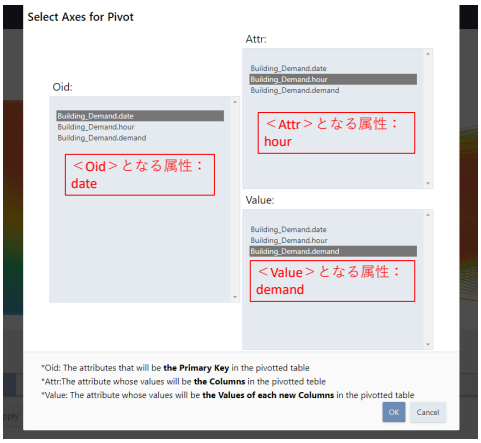


図 5　リレーション Building_Demand に対する PIVOT 操作の様子

と比べて夏（緑）や冬（青）の電力が大きいことがわかる．また，冬と夏では冬の電力のほうが概ね大きいように見える．このように，異なるデータを結合してデータ分析の試行錯誤を行える点に (PC)^{2}DV の特徴がある．

ここで，建物の需要電力は天気の影響を受けると考え，天気の値で選択することを考える．図 10 に天気が晴のデータを選択した様子を，図 11 に天気が雨または曇のデータを選択した様子を示す．図 10 より，晴れていた日の建物の需要電力は冬よりも夏のほうが大きいことがわかる．また，図 11 より，天気が悪い日の建物の需要電力は夏よりも冬のほうが大きいことがわかる．これらのことは，天気の違いによって冷暖房の使用量に違いがあることを示していると考えられる．夏の冷房は天気のいい日は多く使用し，天気の悪い日は比較的使用しない一方で，冬の暖房は天気の悪い日に天気のいい日より多く使用する．このように，SQL を用いた試行錯誤を伴うデータ分析を，GUI と連動させてインタラクティブに行える点もまた，(PC)^{2}DV の特徴である．以上のことがデータを天気ごと，季節ごとに集約し可視化することで示された．

5.4　分析例２：発電電力データの分析

次に，PV の発電電力について可視化・分析を行う．Building_Demand と同様のスキーマのデータに対して同一プロセスで分析を行うため，保存した (PC)^{2}L の一部を書き換えることで同

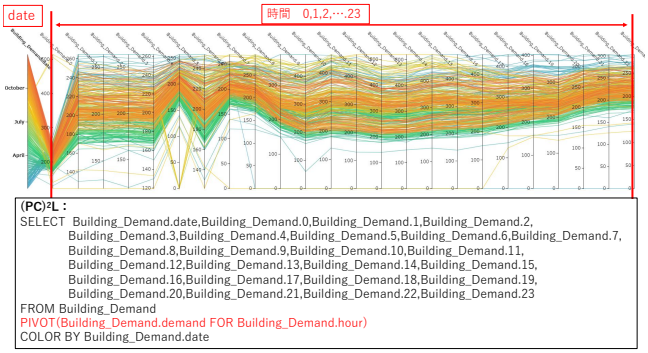


図 6　日ごとの建物の需要電力量を PIVOT で変換し可視化した PCP

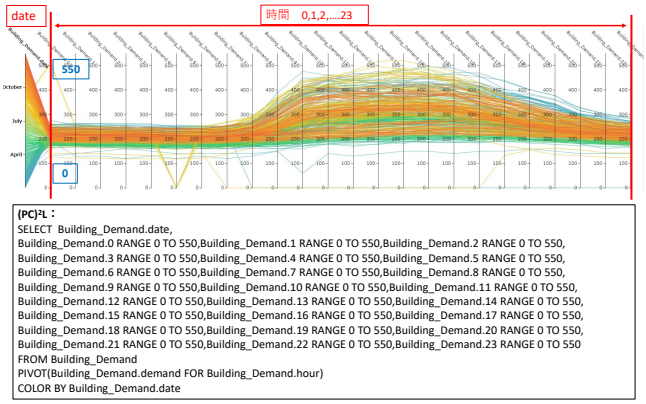


図 7　日ごとの建物の需要電力量を可視化した PCP

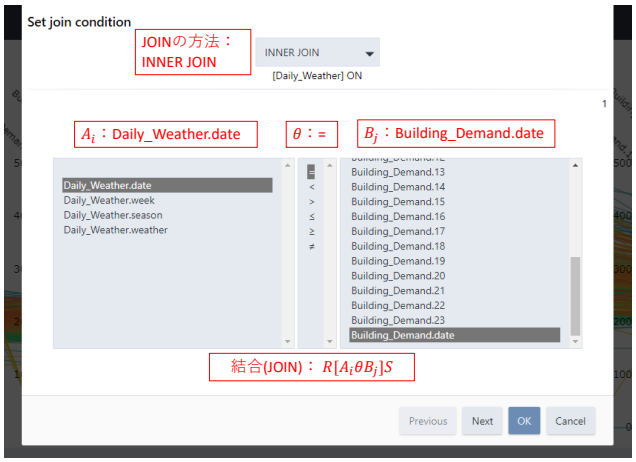
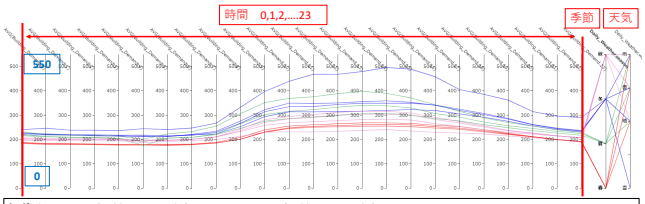


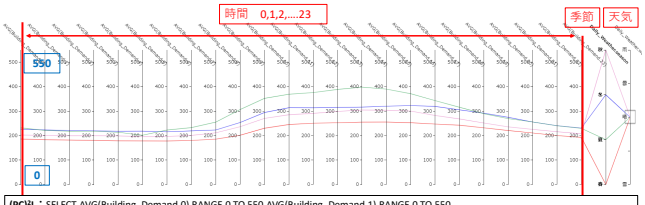
図 8　両リレーションを日付で結合する様子

じプロセスの分析が容易に可能である．取得したデータを可視化した様子とその状態を表す (PC)^{2}L を図 12 に示す．折れ線は日付（date）を基準に色分けされている．Building_Demand と同様にこの状態からデータの持つ周期的な情報を読み取ることは困難であるため，データに対して PIVOT の操作を行い，データを軸の範囲を統一し PCP で表示する．ここでは日ごとの，1 時間ごとの電力データに変換するため，＜ Oid ＞は date，＜ Attr ＞



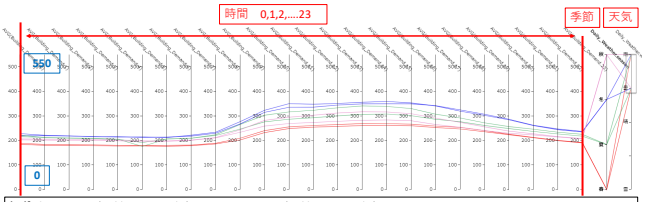

```
(PC)²L：SELECT AVG(Building_Demand.0) RANGE 0 TO 550,AVG(Building_Demand.1) RANGE 0 TO 550,
AVG(Building_Demand.2) RANGE 0 TO 550,AVG(Building_Demand.3) RANGE 0 TO 550,AVG(Building_Demand.4) RANGE 0 TO 550,
AVG(Building_Demand.5) RANGE 0 TO 550,AVG(Building_Demand.6) RANGE 0 TO 550,AVG(Building_Demand.7) RANGE 0 TO 550,
AVG(Building_Demand.8) RANGE 0 TO 550,AVG(Building_Demand.9) RANGE 0 TO 550,AVG(Building_Demand.10) RANGE 0 TO 550,
AVG(Building_Demand.11) RANGE 0 TO 550,AVG(Building_Demand.12) RANGE 0 TO 550,AVG(Building_Demand.13) RANGE 0 TO 550,
AVG(Building_Demand.14) RANGE 0 TO 550,AVG(Building_Demand.15) RANGE 0 TO 550,AVG(Building_Demand.16) RANGE 0 TO 550,
AVG(Building_Demand.17) RANGE 0 TO 550,AVG(Building_Demand.18) RANGE 0 TO 550,AVG(Building_Demand.19) RANGE 0 TO 550,
AVG(Building_Demand.20) RANGE 0 TO 550,AVG(Building_Demand.21) RANGE 0 TO 550,AVG(Building_Demand.22) RANGE 0 TO 550,
AVG(Building_Demand.23) RANGE 0 TO 550,Daily_Weather.season,Daily_Weather.weather
FROM Building_Demand
PIVOT(Building_Demand.demand FOR Building_Demand.hour)
INNER JOIN Daily_Weather on Daily_Weather.date = Building_Demand.date
GROUP BY Daily_Weather.season,Daily_Weather.weather
COLOR BY Daily_Weather.season
```

図 9　日ごとの建物の需要電力量の季節ごと，天気ごとの平均値を可視化した PCP




```
(PC)²L：SELECT AVG(Building_Demand.0) RANGE 0 TO 550,AVG(Building_Demand.1) RANGE 0 TO 550,
AVG(Building_Demand.2) RANGE 0 TO 550,AVG(Building_Demand.3) RANGE 0 TO 550,AVG(Building_Demand.4) RANGE 0 TO 550,
AVG(Building_Demand.5) RANGE 0 TO 550,AVG(Building_Demand.6) RANGE 0 TO 550,AVG(Building_Demand.7) RANGE 0 TO 550,
AVG(Building_Demand.8) RANGE 0 TO 550,AVG(Building_Demand.9) RANGE 0 TO 550,AVG(Building_Demand.10) RANGE 0 TO 550,
AVG(Building_Demand.11) RANGE 0 TO 550,AVG(Building_Demand.12) RANGE 0 TO 550,AVG(Building_Demand.13) RANGE 0 TO 550,
AVG(Building_Demand.14) RANGE 0 TO 550,AVG(Building_Demand.15) RANGE 0 TO 550,AVG(Building_Demand.16) RANGE 0 TO 550,
AVG(Building_Demand.17) RANGE 0 TO 550,AVG(Building_Demand.18) RANGE 0 TO 550,AVG(Building_Demand.19) RANGE 0 TO 550,
AVG(Building_Demand.20) RANGE 0 TO 550,AVG(Building_Demand.21) RANGE 0 TO 550,AVG(Building_Demand.22) RANGE 0 TO 550,
AVG(Building_Demand.23) RANGE 0 TO 550,Daily_Weather.season,Daily_Weather.weather
FROM Building_Demand
PIVOT(Building_Demand.demand FOR Building_Demand.hour)
INNER JOIN Daily_Weather on Daily_Weather.date = Building_Demand.date
WHERE ( Daily_Weather.weather BETWEEN 174 AND 221 )
GROUP BY Daily_Weather.season,Daily_Weather.weather
COLOR BY Daily_Weather.season
```

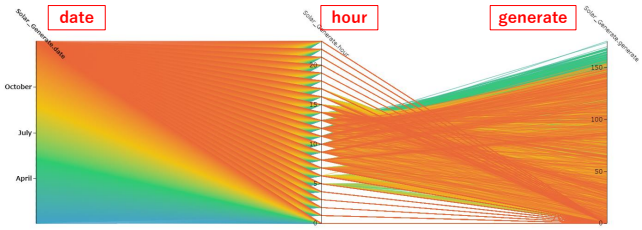
図 10　図 9 から天気が晴のデータを選択した PCP




```
(PC)²L：SELECT AVG(Building_Demand.0) RANGE 0 TO 550,AVG(Building_Demand.1) RANGE 0 TO 550,
AVG(Building_Demand.2) RANGE 0 TO 550,AVG(Building_Demand.3) RANGE 0 TO 550,AVG(Building_Demand.4) RANGE 0 TO 550,
AVG(Building_Demand.5) RANGE 0 TO 550,AVG(Building_Demand.6) RANGE 0 TO 550,AVG(Building_Demand.7) RANGE 0 TO 550,
AVG(Building_Demand.8) RANGE 0 TO 550,AVG(Building_Demand.9) RANGE 0 TO 550,AVG(Building_Demand.10) RANGE 0 TO 550,
AVG(Building_Demand.11) RANGE 0 TO 550,AVG(Building_Demand.12) RANGE 0 TO 550,AVG(Building_Demand.13) RANGE 0 TO 550,
AVG(Building_Demand.14) RANGE 0 TO 550,AVG(Building_Demand.15) RANGE 0 TO 550,AVG(Building_Demand.16) RANGE 0 TO 550,
AVG(Building_Demand.17) RANGE 0 TO 550,AVG(Building_Demand.18) RANGE 0 TO 550,AVG(Building_Demand.19) RANGE 0 TO 550,
AVG(Building_Demand.20) RANGE 0 TO 550,AVG(Building_Demand.21) RANGE 0 TO 550,AVG(Building_Demand.22) RANGE 0 TO 550,
AVG(Building_Demand.23) RANGE 0 TO 550,Daily_Weather.season,Daily_Weather.weather
FROM Building_Demand
PIVOT(Building_Demand.demand FOR Building_Demand.hour)
INNER JOIN Daily_Weather on Daily_Weather.date = Building_Demand.date
WHERE ( Daily_Weather.weather BETWEEN 1 AND 113 )
GROUP BY Daily_Weather.season,Daily_Weather.weather
COLOR BY Daily_Weather.season
```

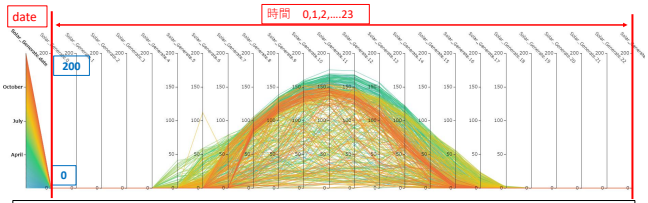
図 11　図 9 から天気が雨，曇のデータを選択した PCP

は時間（hour），＜ Value ＞は発電電力（generate）とした．Solar_Genarete に対して PIVOT の操作を行い，軸の範囲を 0kWh/h から 200kWh/h までに統一した PCP を図 13 に示す．この図から，PV は日の出に合わせて発電し始め，10 時から 12 時頃に 1 日のピークを迎え，日の入りに合わせて発電量が減少していることがわかる．また，昼間の時間帯の折れ線が乱雑であることから，PV の発電電力が天候に依存し不安定であることが示唆されている．




```
(PC)²L：
SELECT Solar_Generate.date, Solar_Generate.hour,
       Solar_Generate.generate
FROM Solar_Generate
COLOR BY Solar_Generate.date
```

図 12　PV の発電電力量を可視化した PCP




```
(PC)²L：
SELECT  Solar_Generate.date,
Solar_Generate.0 RANGE 0 TO 200,Solar_Generate.1 RANGE 0 TO 200,Solar_Generate.2 RANGE 0 TO 200,
Solar_Generate.3 RANGE 0 TO 200,Solar_Generate.4 RANGE 0 TO 200,Solar_Generate.5 RANGE 0 TO 200,
Solar_Generate.6 RANGE 0 TO 200,Solar_Generate.7 RANGE 0 TO 200,Solar_Generate.8 RANGE 0 TO 200,
Solar_Generate.9 RANGE 0 TO 200,Solar_Generate.10 RANGE 0 TO 200,Solar_Generate.11 RANGE 0 TO 200,
Solar_Generate.12 RANGE 0 TO 200,Solar_Generate.13 RANGE 0 TO 200,Solar_Generate.14 RANGE 0 TO 200,
Solar_Generate.15 RANGE 0 TO 200,Solar_Generate.16 RANGE 0 TO 200,Solar_Generate.17 RANGE 0 TO 200,
Solar_Generate.18 RANGE 0 TO 200,Solar_Generate.19 RANGE 0 TO 200,Solar_Generate.20 RANGE 0 TO 200,
Solar_Generate.21 RANGE 0 TO 200,Solar_Generate.22 RANGE 0 TO 200,Solar_Generate.23 RANGE 0 TO 200
FROM Solar_Generate
PIVOT(Solar_Generate.generate FOR Solar_Generate.hour)
COLOR BY Solar_Generate.date
```

図 13　日ごとの PV の発電電力量を可視化した PCP

ここで，季節ごと，天気ごとの PV の発電電力の違いをより明確にするために，データを季節ごと，天気ごとに集約する．建物の需要電力の分析と同様に，リレーション Daily_Weather のデータを取得し，日付をキーとして電力データと結合する．データを日付（date）で結合し季節（season）ごと，天気（weather）ごとの平均値を求め可視化した PCP を図 14 に示す．線の色は季節を表す．この図から，PV の発電電力は天候に大きく依存し，同じ季節（同じ色の折れ線）でも大きく違いがあることがわかる．そこで，データを天気が晴のときのデータを選択し，天気が同じグループ同士での比較を行うことを考える．その様子を図 15 に示す．この図から，天気が晴の日の PV の発電電力は季節ごとの大きな差がないことがわかる．そこで，PCP の折れ線の色を天気で色分けし，天気ごとの違いを見ることを考える．図 14 の PCP を天気で色分けした様子を図 16 に示す．この図より，どの季節も晴（青）のときの発電電力に比べ雨（桃）や曇（赤）の発電電力は大きく減少していることがわかる．よって，PV の発電電力が天気の影響を受けることが示された．

5.5　分析例に関する考察

分析例 1 によって建物の需要電力量が季節と天気の組み合わせによって異なることが示された．また，組み合わせごとに集約して生成した代表値は個々の日の特徴をよく表していることもわかった．可視化によって，日ごとの電力需要波形の傾向は概ね一致することが示された．

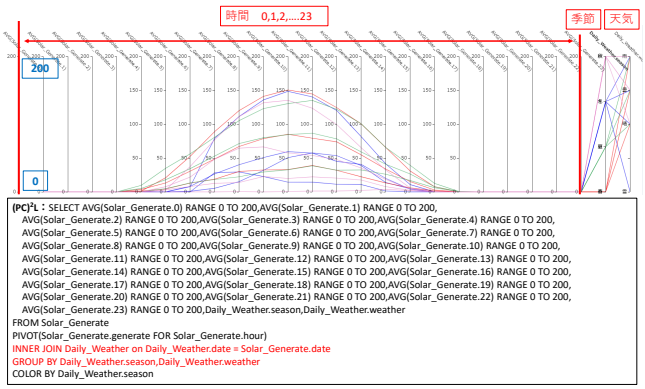


図 14　日ごとの PV の発電電力量の季節ごと，天気ごとの平均値を可視化した PCP

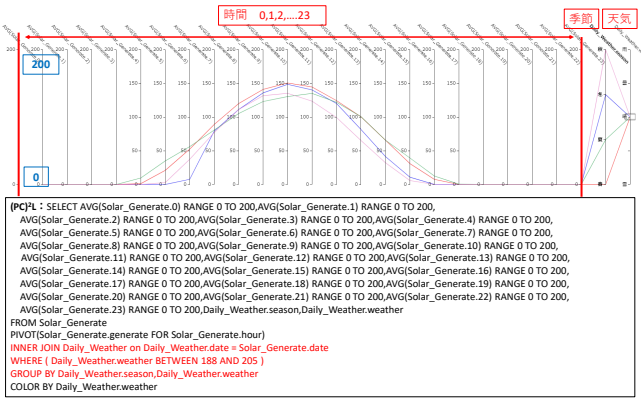


図 15　晴れの日のみを選択した，日ごとの PV の発電電力量の季節ごと，天気ごとの平均値を可視化した PCP

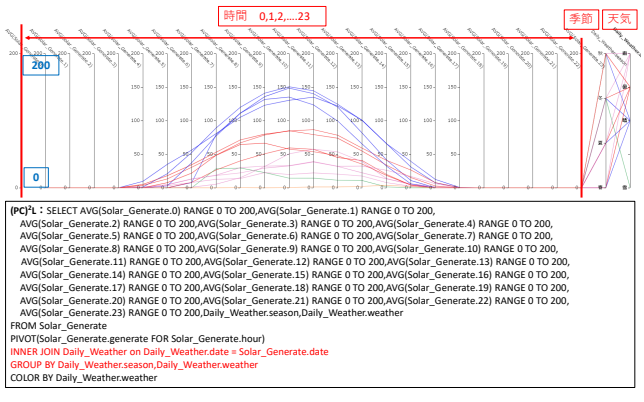


図 16　図 14 を天気で色分けした PCP

一方，分析例 2 によって PV の発電パターンは天気に応じて様々なパターンが生じることも可視化により示された．例えば，晴れた日の発電パターンは集約して生成された代表値と概ね一致する．その一方で，曇りの日は発電力が安定しないため，代表値と大きく異なることがわかった．

我々の別のプロジェクトでは「太陽光発電と電気自動車を統合する負荷平準指向スマートグリッド」の研究に取り組んでいる．本稿の分析例により，天気に依存して大きく異なる PV と，比較的予測が容易な需要電力を組み合わせて，発電エネルギーを地産地消することが可能かどうかの分析に有用であることが明らかになった [28].

6　データ操作高速化とその評価

本章では，(PC)^{2}DV におけるライフログ分析のためのデータ操作高速化とその効果について，描画時間の定量的な比較により評価を行う．

6.1　(PC)^{2}DV における GROUP BY 集約の有効性

5 章では, (PC)^{2}DV 上で大量の時系列データを可視化し, PIVOT の操作によりデータを変換し，異なるデータと結合し，データをグループごとの代表値に集約する分析によりデータの周期的な情報を示した．分析の過程において，GROUP BY 集約により PCP に描画するデータの件数を削減した．この操作は大量のライフログデータの分析において一般的に有用なデータ操作であると考えられる．

6.2　先行研究の (PC)^{2}DV における問題点

3.2 節で述べたように，先行研究 [6] の実装では，データソースに接続して取得したデータをブラウザのメモリに JSON の形式で保存し，それを保持したまま (PC)^{2}DV のプログラム内でデータに対する処理を行い，可視化するデータを変更していた．データ件数の増加に伴いメモリ上でのデータ処理や PCP の線の描画に要する時間が増加し，インタラクティブ性が損なわれることが問題であった．そこで本稿では，データ操作による可視化されるデータセットの変更に伴いメモリに保存されたデータを削減することにより高速化を行う．メモリ上のデータを削減することで PCP で描画する線の数を削減し，(PC)^{2}DV におけるデータ描画の高速化を図る．高速化の定量的な評価のために (PC)^{2}DV のデータ処理時間と描画時間の比較を行う．

6.3　評価概要

本稿では，データ件数の異なるデータセット毎に，(PC)^{2}DV の先行研究の実装と提案手法の実装に対してデータ処理時間とデータ描画時間を計測する．ここで，データ処理時間は，ユーザの操作終了から PCP で描画するデータセットを作成するまでの時間とし，データ描画時間は，PCP で描画するデータセットを作成した時点から PCP でのデータ描画が完了するまでの時間とする．GROUP BY 集約のデータ操作で時間を計測し，二つの実装の比較・評価を行う．Microsoft SQL Server に格納した評価用のデータセットを表 4 に示す．これは，2022 年 4 月 1 日から 2022 年 9 月 26 日までの神奈川県における新型コロナウイルス感染症の感染者のデータであり，神奈川県が公開しているオープンデータ[*4]である．このデータセットのデータ件数は 874678 件であった．このデータセットからデータ件数 N が，N = {100, 200, 500, 1000, 2000, 5000, 10000, 20000, 50000, 100000, 200000} となるようにデータをハッシュにより抽出し，評価用のデータセットを作成した．評価に用いる，GROUP BY 集約を伴う (PC)^{2}L を

[*4] https://www.pref.kanagawa.jp/docs/ga4/covid19/occurrence_list.html#patient_opendata

表 4　評価用データセットの属性

属性名	説明
date	日付（yyyy-mm-dd）
area	感染者が報告された保健所の場所
age	感染者の年代
gender	感染者の性別

```
SELECT [TableName].[area],[TableName].[age],
       COUNT(TableName.date)
FROM [TableName]
GROUP BY [TableName].[area],[TableName].[age]
COLOR BY COUNT(TableName.date)
```

図 17　GROUP BY 集約を伴う (PC)^{2}L

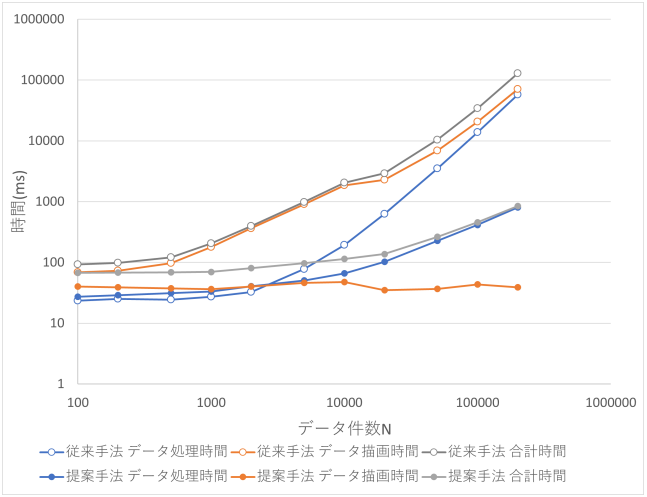


図 18　GROUP BY 集約を伴うデータ操作に要する時間の計測結果

図 17 に示す．図中の TableName はデータセットのテーブル名を表す．これは area と age ごとのデータ件数，すなわち感染者数を求めるクエリであり，県内の地域ごとの感染者の年齢層の傾向の違いを把握するような分析を想定している．なお，実験を行った PC について，CPU は Intel Core i5-10500，3.10GHz，メモリは 32GB × 2（64GB），OS は Windows10 Education 64bit，ブラウザは Microsoft Edge version110.0.1587.46（64bit）を用いた．

6.4　評価結果と考察

GROUP BY 集約を伴うデータ操作におけるデータ処理時間と描画時間を各データセットに対して計測した．その結果を両対数グラフとして図 18 に示す．図中の白抜きの点は従来手法，塗りつぶされた点は提案手法を表し，各線がデータ処理時間（青），データ描画時間（橙）とそれらの合計時間（灰）を表す．なお計測結果から，N が 20 万以上のデータセットではインタラクティブ性が大きく損なわれると考えた．そのため，N を 20 万で打ち止めた．この図より，先行研究の実装では N の増加に伴いデータ処理時間と描画時間がどちらも大きく増加していることがわかる．これは先行研究では GROUP BY 集約を実行してもデータ件数が減らず，PCP で描画する折れ線の数も減らないためである．また，N が 10 万以上のデータセットでは合計時間が 10000ms（10s）を超えており，インタラクティブ性が損なわれているといえる．一方，提案手法ではデータ描画時間は N に依らずほぼ一定であることがわかる．これは，データ件数に関わらずグループ数が概ね同じあり，描画する線の数が概ね変わらないためである．すなわち，GROUP BY 集約によりデータ件数を削減したことでデータ描画時間が一定になったといえる．そのため，合計時間は先行研究と比較して大きく削減された．この時間が 1000ms 程度であるため，インタラクティブ性が確保されているといえる．よって，GROUP BY 集約を伴うデータ操作によるデータ件数の削減が，(PC)^{2}DV におけるデータ操作高速化に有効であることが検証された．なお，データ処理時間に関して，提案手法でも取得したデータに対して (PC)^{2}DV のプログラム上でデータ処理を行っている．(PC)^{2}DV 上のデータ操作はすべて (PC)^{2}L に可換であり，(PC)^{2}L は SQL と同等の文法を持つ言語である．そのため，SQL と同等の操作は DBMS 上で実行する構成が妥当である．このようにすることでデータ処理時間や実行プランの妥当性が DBMS により保証される．これについては，今後の課題とする．

7　まとめと今後の課題

我々は，複数の属性からなるデータを PCP により可視化し，その状態を (PC)^{2}L で保存・再現可能な可視化システム (PC)^{2}DV を提案してきた．本稿では特に，ライフログ分析を目的としてデータ操作支援を行った．大量の時系列データを可視化し，PIVOT の操作によりデータを変換し，異なるデータと結合し，データをグループごとの代表値に集約する分析によりデータの持つ周期的な情報を示した．

また，データ分析における GROUP BY 集約の有用性に着目し，データ操作の高速化として，GROUP BY 集約によるデータ件数の削減を行い，先行研究の実装との処理時間と描画時間の定量的な比較を行った．これにより，(PC)^{2}DV におけるデータ削減の有効性と，ビッグデータの分析可能性を示した．

今後の課題として，SQL で可能な操作であるデータの更新，挿入，削除といった操作を (PC)^{2}DV 上で実行可能にし，データ分析支援を拡充することが挙げられる．また，取得元が異なる複数のデータを (PC)^{2}DV を用いて組み合わせるデータ分析や，今回の分析で扱った電力データとは全く異なる種類のデータを対象とした分析により，有用な知見を示すことが挙げられる．これにより，多様なデータの分析において (PC)^{2}DV がキラーアプリケーションとして機能する事例を示す．

8　謝辞

本研究の一部は JSPS 科研費 (課題番号 22H03810) の支援による．

参考文献

[1] Alfred Inselberg. The plane with parallel coordinates. *The Visual Computer*, Vol. 1, No. 2, pp. 69–91, 1985.
[2] Jimmy Johansson and Camilla Forsell. Evaluation of parallel coordinates: Overview, categorization and guidelines for future research. *IEEE Trans. on Visualization and Computer Graphics (TVCG)*, Vol. 22, No. 1, pp. 579–588, 2016.
[3] 濱崎裕太, 植村智明, 富井尚志. 多変量データを SPJ 質問により統合する平行座標プロット型情報可視化システムと操作言語. 情報処理学会論文誌データベース（TOD）, Vol. 12, No. 4, pp. 27–39, October 2019.
[4] 植村智明, 吉田顕策, 吉瀬雄大, 富井尚志. 試行錯誤を許容するデータ解析支援システムと電気自動車の走行ログ解析. 情報処理学会論文誌データベース（TOD）, Vol. 13, No. 4, pp. 13–26, October 2020.
[5] 植村智明, 能條太悟, 吉瀬雄大, 富井尚志. 解析者の興味に基づく道路区間集計が可能な EV 推定消費エネルギーデータ解析システムの構築と応用. 情報処理学会論文誌データベース（TOD）, Vol. 14, No. 4, pp. 70–85, October 2021.
[6] 能條太悟, 稲澤朋也, 富井尚志. PCP と言語表現を組み合わせた多変量データ分析支援システムの拡張. 第 14 回データ工学と情報マネジメントに関するフォーラム (DEIM2022), J23-1, pp. 1–8, 2022.
[7] Chuxuan Tong, Jinglan Zhang, Alok Chowdhury, and Stewart G. Trost. An interactive visualization tool for sensor-based physical activity data analysis. In *Proceedings of the Australasian Computer Science Week Multiconference*, ACSW 2019, New York, NY, USA, 2019. Association for Computing Machinery.
[8] G. Grinstein, M. Trutschl, and U. Cvek. High dimensional visualizations. In *In Proceedings of KDD Workshop on Visual Data Mining*, 2001.
[9] Fatma Bouali, Abdelheq Guettala, and Gilles Venturini. VizAssist: An interactive user assistant for visual data mining. *The Visual Computer: Int'l Journal of Computer Graphics*, Vol. 32, No. 11, pp. 1447–1463, 2016.
[10] Wenqiang Cui. Visual analytics: A comprehensive overview. *IEEE Access*, Vol. 7, pp. 81555–81573, 2019.
[11] Takayuki Itoh, Ashnil Kumar, Karsten Klein, and Jinman Kim. High-dimensional data visualization by interactive construction of low-dimensional parallel coordinate plots. *Journal of Visual Languages & Computing*, Vol. 43, pp. 1–13, 2017.
[12] Z. Zhou, Z. Ye, J. Yu, and W. Chen. Cluster-aware arrangement of the parallel coordinate plots. *Journal of Visual Languages & Computing*, Vol. 46, pp. 43–52, 2018.
[13] Jinwook Bok, Bohyoung Kim, and Jinwook Seo. Augmenting parallel coordinates plots with color-coded stacked histograms. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 28, No. 7, pp. 2563–2576, 2022.
[14] Henning Gruendl, Patrick Riehmann, Yves Pausch, and Bernd Fröhlich. Time‐series plots integrated in parallel‐coordinates displays. *Computer Graphics Forum*, Vol. 35, , 2016.
[15] P. Godfrey, J. Gryz, and P. Lasek. Interactive visualization of large data sets. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, No. 8, pp. 2142–2157, 2016.
[16] Mark Derthick, John Kolojejchick, and Steven F. Roth. An interactive visual query environment for exploring data. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, UIST '97, pp. 189–198, 1997.
[17] Chris North and Ben Shneiderman. Snap-together visualization: A user interface for coordinating visualizations via relational schemata. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '00, pp. 128–135, 2000.
[18] 杉渕剛史, 田中譲. 関係データベースモデルに基づくデータベース可視化フレームワークの提案と実装. 電子情報通信学会論文誌. D, 情報・システム = The IEICE transactions on information and systems (Japanese edition), Vol. 90, No. 3, pp. 918–932, mar 2007.
[19] Melanie Herschel, Ralf Diestelkämper, and Houssem Ben Lahmar. A survey on provenance: What for? what form? what from? *The VLDB Journal*, Vol. 26, No. 6, pp. 881–906, Dec 2017.
[20] Manuela Waldner, Stefan Bruckner, and Ivan Viola. Graphical histories of information foraging. *Proc. of the 8th Nordic Conf. on Human-Computer Interaction: Fun, Fast, Foundational(NordiCHI '14)*, pp. 295–304, 2014.
[21] Peter Mindek, Stefan Bruckner, and M. Eduard Gröller. Contextual snapshots: Enriched visualization with interactive spatial annotations. *Proc. of the 29th Spring Conf. on Computer Graphics(SCCG '13)*, pp. 49–56, 2013.
[22] S. Gratzl, N. Gehlenborg, A. Lex, H. Pfister, and M. Streit. Domino: Extracting, comparing, and manipulating subsets across multiple tabular datasets. *IEEE Trans. on Visualization and Computer Graphics(TVCG)*, Vol. 20, No. 12, pp. 2023–2032, Dec 2014.
[23] Holger Stitz, Samuel Gratzl, Harald Piringer, Thomas Zichner, and Marc Streit. Knowledgepearls: Provenance-based visualization retrieval. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 25, No. 1, pp. 120–130, 2019.
[24] Columbus Salley and E. F. Codd. Providing olap to user-analysts: An it mandate. 1998.
[25] Rakesh Agrawal, Amit Somani, and Yirong Xu. Storage and querying of e-commerce data. In *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pp. 149–158, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
[26] Conor Cunningham, César A. Galindo-Legaria, and Goetz Graefe. Pivot and unpivot: Optimization and execution strategies in an rdbms. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pp. 998–1009. VLDB Endowment, 2004.
[27] Gang Luo and Lewis J. Frey. Efficient execution methods of pivoting for bulk extraction of entity-attribute-value-modeled data. *IEEE Journal of Biomedical and Health Informatics*, Vol. 20, No. 2, pp. 644–654, 2016.
[28] 石毛大貴, 廣居樹, 片平昇輝, 鈴木博登, 本藤祐樹, 富井尚志. 太陽光発電と EV を統合する負荷平準指向スマートグリッド DB を用いた VGI シミュレーション評価. 第 15 回データ工学と情報マネジメントに関するフォーラム (DEIM2023), 5c-9-3, pp. 1–10, 2023.