DEIM Forum 2018 G7-5

時空間演算に基づく Verbal Search の提案

阪田　晴香† 　王　　元元†† 　Panote Siriaraya† 　河合由起子†

† 京都産業大学コンピュータ理工学部　〒 603–8555 京都府京都市北区上賀茂本山
†† 山口大学大学院創成科学研究科　〒 755–8611 山口県宇部市常盤台 2-16-1
E-mail: †{g1544647, kawai}@cc.kyoto-su.ac.jp, spanote@gmail.com ††y.wang@yamaguchi-u.ac.jp

あらまし　本研究は，範囲や方向，時間長，大きさを組み合わせた検索の実現を目的とする．従来，検索キーワードを中心とした距離や範囲検索，検索キーワード間の範囲検索は，検索意図の表現が困難であり，また検索解の計算コストが高いという課題があった．本論文では，新たな時空間演算子を定義し，それら範囲や方向，時間長の時空間検索を可能にする検索方式を提案する．提案手法は，検索キーワード間の空白（␣）に着目し，空白の長さを距離や時間長として直感的に表現可能にすることで，1 トランザクションでの検索が可能な時空間演算を実現する．具体的には，「四条烏丸駅␣800m␣蕎麦屋」の問い合わせにより，現在地（四条烏丸駅）から 800m の距離にある蕎麦屋を検索できる．また，集合演算子に加え，[^] や [_]，[θ] の方向演算子，[*] の範囲演算子といった新たに 7 つの空間演算子を定義し，時空間範囲検索システムを開発し，従来不可能だった集合演算による時空間検索を可能とする．これにより，「（四条烏丸␣*800m␣蕎麦屋）*（八坂神社␣800m␣蕎麦屋）」（四条烏丸と八坂神社双方の 800m 圏内に共通する蕎麦屋）や「四条烏丸駅␣^800m␣蕎麦屋」（四条烏丸から 800m 圏内で北方向にある蕎麦屋）といった集合演算検索が可能になる．本論文では，12 種類の演算子を定義し，それら演算子に基づく時空間範囲検索システムによる地図検索および Web テキスト検索の実行例を示す．

キーワード　spatial-temporal operation，space query，時空間範囲検索，verbal search system

1. はじめに

スマートフォン等の携帯端末の普及により，Web や映像，地図への Nonverbal 検索サービスの需要が増しているが，テキストによる空間および時間の Verbal Query 生成および演算は困難である．例えば，地図検索において，「現在地から 500m 圏内の蕎麦屋」の問い合わせは，「蕎麦屋」の地図検索結果に「500m」の nonverbal となる制約指定をかける 2 トランザクションにより検索可能であるが，（Q1）現在地から 500m 圏内で「北側にある」蕎麦屋や「南西方向にある」蕎麦屋，（Q2）現在地から「200〜500m 圏内にある」蕎麦屋を 1 トランザクションで，かつテキストによる問い合わせはできない．また，（Q3）現在地から 500m 圏内に加えて，「B 地点からも 500m 圏内の共通する」蕎麦屋の問い合わせ生成も同様に 1 トランザクションのテキスト検索はできない．本研究では，方向や範囲，時間長となる演算子を新たに定義することで，テキストによる時空間クエリ生成を可能にする時空間範囲演算子による直感的かつ効率的検索方式を提案する．

提案手法は，従来のテキストによる問い合わせで用いられている「空白（␣）」に着目し，これまで考慮されていなかった空白の長さを空間を表現する演算子として利用する．従来のテキスト検索では「A␣B」も「A␣␣␣␣␣B」も「A と B が同時に含まれる」という問い合わせが生成され検索結果は同じになる．我々は URL リクエストの「A+B」と「A+++++B」は異なる問い合わせと捉え，「A+B」は A から最も近い B の検索であり，「A++B」は A から 2 番目に近い B を含む検索であり，「A+++++B」は A から 5 番目に近い B を含む検索となり，空間的差異を考慮した検索が実現できる．本論文では，それらを表現可能な空間演算子を定義する．例えば，A から近い 5 件の B を空間検索したい場合は，「A␣␣␣␣␣B」となる．また，A から 5 番目に近い B の空間検索の場合は「A␣␣␣␣␣*B」となる．さらに，空白長を距離に変換し用いることを可能とし，A から 5km 離れた B の空間検索の問い合わせは「A␣5km␣B」（「A+5km+B」）となる．また，距離（空間）検索は地図検索に有用となるが，文章にも応用でき，例えば A から 5 単語離れた B を含む文章も「A␣5 単語␣B」として検索でき，同様に映像の空間検索も可能となる．

本論文で定義する時空間演算子は，集合演算子と新たに範囲，方向，時間等の 12 種類である．これらを用いることで，(Q1) の方向検索（「*」「^」「_」「θ」），(Q2) の範囲検索（「[]」），(Q3) の範囲と四則演算による問い合わせ「（A␣0.5km␣蕎麦屋）＋（B␣0.5km␣蕎麦屋）」が可能となる．また，検索キーワード（オブジェクト）の特性検索のためのプロパティ演算子「$」，さらに時間演算子「#」を定義する．これにより，例えば，「現在地から 500m 圏内で 20 人規模の蕎麦屋」は「現在地␣500m$20 人␣蕎麦屋」の問い合わせ演算より検索可能となる．

本論文の構成は以下の通りである．2. 章では意味的演算および時間演算に関連する既存研究，地図や映像検索の関連研究を紹介する．3. 章では時空間演算子を定義し，演算によるシンプルで直感的な地図での問い合わせ例を提示する．また，Web 検索および映像検索での問い合わせ例を示す．4. 章では時空間演算による時空間検索システム構成による利用方法について述べ

る．5. 章では時空間演算子の一部を実装した地図検索・Web 検索システムを用いた実行例を提示する．最後に 6. 章で本論文のまとめと今後の課題を述べる．

2. 関 連 研 究

2.1 ビデオ検索

これまでに，ビデオ検索モデルに関する研究は活発に行われている．多次元空間の演算の研究では，インデックスを付加されたビデオ区間群を時系列的に連結することによって，あるビデオデータ問い合わせに対して答えとなり得るすべての区間を動的に生成するグルー（glue）演算と呼ばれる区間演算群がある [1] [2] [3]．しかしながら，このグルー演算のみでは不適切な解も生成してしまうため，そのようなノイズを含んだ区間を解から取り除く区間フィルタの概念も提案されている [4]．

また，近年では，効率的なビデオ検索のためには，色やテクスチャなどの表層的な特徴だけでなく，そこに何が映っているのかといった意味内容に基づいた解析が重要である．ビデオ中に出現するオブジェクトやイベントを識別する手法としては，SIFT [5] や SURF [6] といった局所特徴の出現頻度に基づく bag-of-features モデル [7] がある．これらの研究では，ビデオ区間の意味的演算やビデオの局所特徴に着目し，効率的にビデオを検索することができる．本研究では，空間と時間の単位を空白の長さで表現し，時空間範囲検索のための新たな演算モデルを提案する．

2.2 地 図 検 索

地図検索やロケーション検索においても様々な研究が行われている．そのうち，特にルート検索の研究が注目されている．景色のよい，心地の良い，安全性などの基準を考慮し，二地点間の最適なルートを検索するシステムが提案された [8] [9] [10]．また，最短なルートだけでなく，方向転換（曲がる）回数をできるだけ少なくすることで，記憶しやすいルートを検索する手段も提案されている [11]．最近，ソーシャルサーチ（社会的情報）を用いた地図検索の研究が行われている．これはソーシャルメディアや他ユーザが各地物に対して評価するクラウドソーシングを利用することで，より正確な地物検索を提供している [12] [13]．また，スマートフォンといったモバイル端末における地図検索を利用する際の画面制約の解消や操作性の向上を目指したインターフェイスの研究も存在する [14]．

地図検索システムのユーザがいかに的確な情報を得られるのかは，入力した検索クエリによって大きく影響される．しかしながら，近年，提供している地域検索サービス（Google Maps や Bing Maps など）の検索クエリはほとんど，単純なロケーションを含めたキーワードや文章で構成されている．例えば，ユーザが京都にあるレストランを検索したい場合は，「Restaurants␣in␣Kyoto」．東京駅の近くにあるスーパーの情報が求めたいなら，「Supermarkets␣near␣Tokyo station」といった文章による検索クエリを利用する． 本論文で提案している新たな時空間演算子は，それらのキーワードを空間演算式の形式のクエリに組み立てることが可能になり，例えば，「Tokyo station␣500m␣Supermarkets」といった直感的な検索を実現できる．

2.3 テキスト検索

テキスト検索では，Mikolov ら [15] は，ニューラルネットワークを用いて，Word2Vec という語のベクトル表現を獲得する手法を提案している．この手法では，類似語を抽出するのみではなく，「king − man + woman = queen」のように語の意味演算を行うことが可能とされている．

また，近接演算子を用いた検索対象となる項目内での検索語間の距離や位置関係を指定して検索する近接演算がある．例えば，ScienceDirect や Scopus では近接演算子（W/n）を用いて「microscopy W/3 gfp」のように「microscopy」と「gfp」を語順を問わず 3 語以内に含む文献を検索できる [16]．Oracle Text では近接演算（near）を用いて「near((dog, cat), 6)」のように「cat」の 6 ワード以内に「dog」を含むすべてのドキュメントを検索できる [17]．本研究では，検索キーワード間の空白（␣）に着目し，検索キーワードの時空間演算によって，検索キーワードを中心とした距離や範囲検索，検索キーワード間の時空間範囲検索を可能にすることでテキスト検索だけでなく，地図検索および映像検索にも応用できる．

3. 時空間演算

本章では，12 種類の時空間演算子を定義し，時空間検索演算例を地図検索を中心に提示する．また，Web 検索と映像検索への適用例も提示する．なお，下記を問い合わせ演算の前提とする．

定義 1： 最小の時空間演算の問い合わせ SQ は「A␣（時空間演算子）時空間長␣α」(注1)の形式とする．

A と α はキーワードであり，A を起点とした時空間上の α の検索を意味する．また，ユーザの入力時の操作性を考慮し，時空間長の前に出現する空白（␣）は 1 つ以上の入力を許す．

（例） 「（A␣␣800m␣α）」は「A から 800m 圏内の α」

定義 2： 定義 1 の最小の時空間演算 SQ の問い合わせは，集合演算子を組み合わせた「$SQ^1 \cup SQ^2 \cup SQ^3 ... = SQ^+$」の形式で記述可能とする．

（例） 「（A␣800m␣α）＋（B␣800m␣β）」は「A から 800m 圏内の α と B から 800m 圏内の β の和集合」

定義 3： 時空間長は数値を含む単位を記述する．

「␣時空間演算子」に続いて数値と単位（例えば「800m」）を記述する．

単位には単語を許さない（例えば「3 たばこ」）．ただし，システム設計者がユーザの URL リクエストから演算処理のインタプリタに渡す前に，単語から単位変換を行うことで，単位に単語の使用を許せる．例えば，たばこの縦 3.5 インチと設計者が定義することで「3 たばご」＝「10.5 インチ」として演算処理が可能となる．

3.1 集合演算子

定義 1 の問い合わせは集合演算で可能となる．集合演算子は，和「+」，差「-」，積「*」とする（図 1）．

(注1)：RESTFUL リクエスト「A++演算式+α」となる

表 1　空間演算子

演算子	␣*	␣^	␣_	␣θ	␣[x-y]	␣$	␣#
処理	場所	方向（北・上）	方向（南・下）	角度	範囲	プロパティ	時間

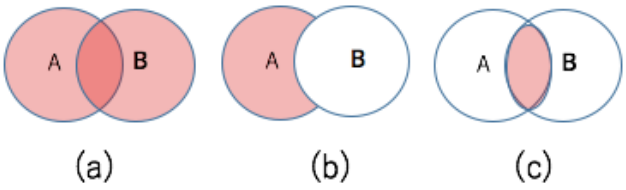

図 1　（a）和演算（b）差演算（c）積演算

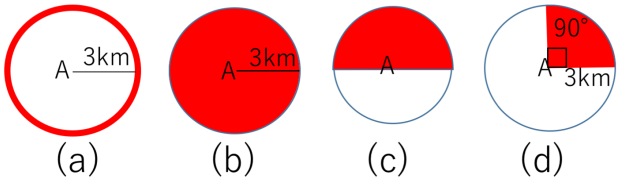

図 2　（a）周辺（b）周囲（c）方向（北・上）（d）角度（90 度の例）

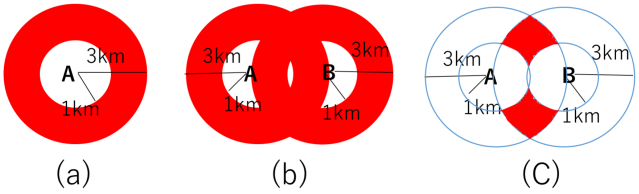

図 3　（a）範囲（b）範囲と和演算例（c）範囲と積演算例

和演算（例 1）　「(A␣3km␣α) ＋ (B␣3km␣α)」
　A から 3km 圏内と B の 3km 圏内の和演算（図 1（a））
差演算（例 2）　「(A␣3km␣α) - (B␣3km␣α)」
　A の 3km 圏内から B の 3km 圏内の差演算（図 1（b））
積演算（例 3）　「(A␣3km␣α) * (B␣3km␣α)」
　A から 3km 圏内と B から 3km 圏内の積演算（図 1（c））

3.2　方向演算子

表 1 に新たに定義する 7 つの時空間演算子を示す．周囲，方向，角度の演算子を用いた演算例 4 つを下記に示す（図 2）．

距離演算（例 4）　「A␣*3km␣α」
　A から 3km の周辺にある α（図 2（a））
周囲演算（例 5）　「A␣3km␣α」
　A の 3km 圏内にある α（図 2（b））
方向演算（例 6）　「A ␣^3km␣α」
　A から 3km 圏内の北側方向にある α（図 2（c））
角度演算（例 7）　「A ␣3km θ 90␣α」
　A から 3km 圏内の角度 90 度方向にある α（図 2（d））

3.3　範囲演算子

表 1 の範囲演算子を用いた演算例 3 つを下記に示す（図 3）．

範囲演算（例 8）　「A␣[3km-1km]␣α」
　A から半径 1km から 3km の範囲にある α（図 3（a））
　または下記の式より，周囲演算子と差演算でも記述可能
　「(A␣3km␣α) - (A␣1km␣α)」

表 2　時間演算子

演算子	␣#*	␣#<	␣#>	␣#[x-y]	␣#$
処理	前後	前	後	範囲	現在

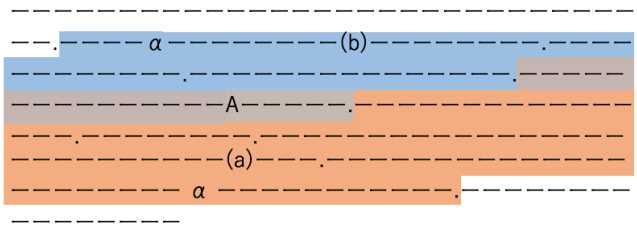

図 4　Web 検索による文章抽出の範囲演算例

（例 9）　「(A␣[3km-1km]␣α) ＋ (B␣[3km-1km]␣α)」
　A と B の半径 1km から 3km の範囲内にある α（図 3（b））
（例 10）　「(A␣[3km-1km]␣α) * (B␣[3km-1km]␣α)」
　A と B の半径 1km から 3km 範囲内の共通の α（図 3（c））

3.4　プロパティ演算子

表 1 のプロパティ演算子「$」は検索対象となる α のプロパティ（属性）の大きさを単位とする．ただし，α にそのプロパティが含まれていない場合は NULL を返す．

プロパティ演算（例 11）　「A␣3km$10 台␣α」
　A から半径 3km 圏内で 10 台の規模の α

3.5　時間演算子

表 1 の時間演算子を用いることで，時間における範囲検索を実現する．表 2 に時間演算子 5 種類を定義する．

地図における時間演算子は，A から α までに必要となる時間間隔となる．A から α までの経路導出とそれに伴う時間はシステムに依存する（詳細は 4. 章参照）．なお，映像における時間演算子の利用例は次節 3.6.2 で述べる．

時間演算（例 12）　「A␣#3 分␣α」
　A の 3 分圏内の距離にある α
時間演算（例 13）　「A␣#*3 分␣α」
　A から 3 分の距離にある α
時間演算（例 14）　「A ␣^#3 分␣α」
　A から 3 分圏内の北側方向にある α
時間演算（例 15）　「A ␣#3 分 θ 90␣α」
　A から 3 分圏内の角度 90 度方向にある α

例 14 と例 15 のように演算子を連続して記述することで，組み合わせ演算が可能である．

3.6　時空間演算子による Web 検索および映像検索

時空間範囲検索は地図検索のみでなく，Web 検索，映像検索にも適応可能である．

3.6.1　Web 検索への適用例

Web 検索における時空間演算では，A から α までの文章を検索し抽出結果を返す．

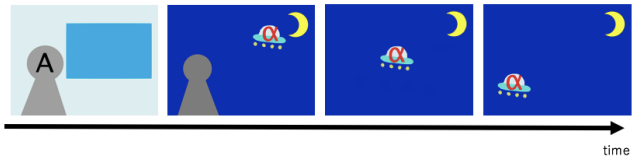

図 5　映像検索による時間抽出の範囲演算例

距離演算（例）　「A␣*3 行␣α」（図 4（a）（b））
　A から 3 行内でαがマッチした場合にそのαを含む行
範囲検索（例）　「A␣3 行␣α」（図 4（a））
　A から 3 行目のαがマッチした場合に A から 3 行分
方向検索（例）　「A␣#<3 行␣α」（図 4（b））
　A から 3 行前のαがマッチした場合にαから A の 3 行分
αを拡張子（例えば.png）やリンクとすることも可能である．
方向検索（例）　「A␣#<3 行␣.jpg」
　A から 3 行前の jpg（画像）がマッチした場合に画像から A までの 3 行分
方向検索（例）　「A␣#<3 行␣url」
　A から 3 行前のリンクがマッチした場合にリンクから A までの 3 行分

行だけでなく，パラグラフを単位として利用できる．また，DOM 解析により視覚的な方向を考慮した検索が可能となる．これによりページ内のどこに存在するかという視覚的検索ができる．

方向検索（例）　「A␣^3 行␣.jpg」
　A から 3 行上にある jpg（画像）がマッチした場合に jpg 画像から A までの 3 行分
方向検索（例）　「A␣[θ 90-θ 180]␣.jpg」
　A からθ 90 からθ 180（左横）の jpg（画像）がマッチした場合に画像から A まで全て

3.6.2　映像検索への適用例

映像検索における時空間演算では，A からαまでの映像を検索し抽出結果を返す．

距離演算（例）　「A␣#3 分␣α」
　A から 3 分圏内でαがマッチしたフレーム
範囲検索（例）　「A␣#*3 分␣α」
　A から 3 分目のαがマッチした場合に A から 3 分間
方向検索（例）　「A␣#<3 分␣α」
　A から 3 分前のαがマッチした場合にαから A の 3 分間

また，映像フレーム解析により視覚的な方向を考慮した検索が可能となる．これにより映像中のどこに存在するかという視覚的検索ができる．

方向検索（例）　「A␣^3 分␣α」
　A から 3 分間に画面の上方向のαがマッチした場合に A から 3 分間
方向検索（例）　「A␣[θ 90-θ 270]␣α」（図 5（a））
　A から画面のθ 90 からθ 270 の画面左でαがマッチした場合に A からαまで全ての映像

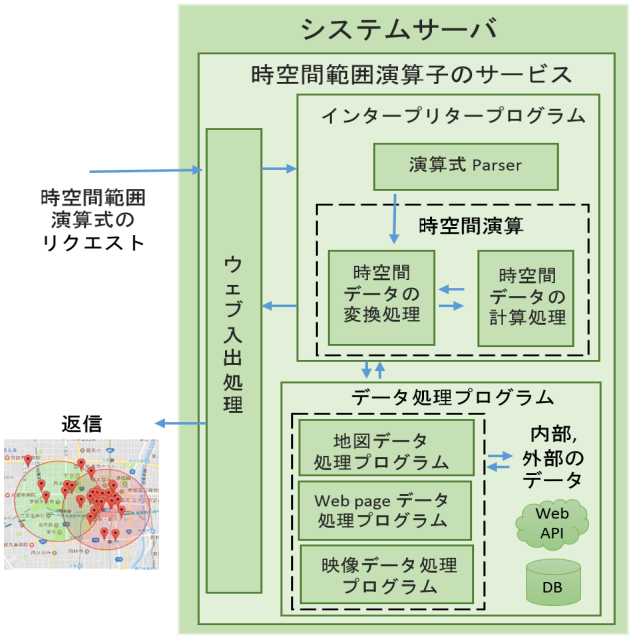

図 6　時空間範囲演算子のサービスのシステムの概要

4.　時空間検索システム

本章では，時空間演算の実現方法として時空間検索システム構成に基づいた利用方法について述べる．

時空間検索システムは，ユーザからの Web リクエストの入出力となる Web 入出力処理の部分と時空間範囲演算子を処理するインタープリタプログラムとシステム設計者が開発したデータ処理するプログラムという 3 つの要素で構成される．

4.1　Web 入出力処理

Verbal Search のシステムを利用するとき，クライアントは時空間演算とデータタイプ（地図，Web ページ，映像）の RESTFUL の HTTP リクエストを送信する．例えば，「www.verbalsearch.com」サーバに地図の時空間検索の演算式「A␣^3km␣α」をリクエストしたい場合は，「www.verbalsearch.com/map/?q=A+^3km+α」の GET リクエストがサーバに送られる．/map とするか，またはリクエストに「&type=map」等して含めるかはサーバ設計者に依存する．

これら取得したデータタイプと演算式を時空間演算処理に渡す．時空間演算処理はデータ取得抽出に処理にデータタイプに合わせた範囲を含む配列を渡す．データ取得抽出処理は内部または外部のデータからデータタイプに基づき配列範囲内のデータを取得し，それら配列を時空間演算処理へ返す．

時空間演算処理は受け取った配列から計算処理し，計算結果を Web 入出力処理へ渡し，Web 入出処理はユーザへ HTTP レスポンスとして返信する．

4.2　インタプリタプログラムの構成と概要

インタプリタプログラムはユーザからの時空間範囲演算式を処理する役割を持つ．プログラムの構成は演算式のパーサ，時空間データの変換機，時空間データの計算機の三要素とする．

時空間範囲演算自体はリクエストのパラメータとして送る．

そのリクエストは Web サービスシステムの入出力処理部分によって受け取り，ユーザが入力した演算式をパーサで処理する．そのパーサプログラムはユーザが入力した時空間範囲演算式の構文を解析し，必要な時空間範囲演算処理とその処理順を決める．例えば，ユーザが入力した時空間範囲演算式が

「(A␣^3km␣α ＋ B␣^3km␣α」) ＋
(C␣^5km␣α」＋D␣^1km␣α」)」

であれば，パーサは次のように処理順を決定する．

変数 1　＝ A␣^3km␣α * B␣^3km␣α」(作業 1)
変数 2　＝ C␣^5km␣β ＋D␣^1km␣β」(作業 2)
結果　＝ 変数 1 ＋変数 2 (作業 3)

次に，処理リストにある時空間範囲演算処理を順に行う．まず，時空間範囲演算式が時空間データ変換機に送られる．その変換機が演算式の要素（例えば，A␣^3km　または B␣^3km）を時空間データの流域の行列に変換する．変換機のプログラムは設計者が書くデータ処理プログラムを参考に，時空間データの流域を計算する．例えば，「A␣^3km␣店」という地図の時空間範囲演算の入力の場合，時空間変換機は設計者が書く地図のデータ処理プログラムを参考に，A の位置と，その場所から 3km の範囲内にある店の情報を得てくる．例えば，設計者が書くデータ処理プログラムが Google Maps API のデータソースを指定すれば，A の位置と指定範囲内にある店の情報は Google Maps API のリクエストにより求められる．また，設計者が内部の SQL データベースを指定すると，SQL クエリによって A の位置と指定範囲内にある店情報を得て，時空間範囲の流域行列を計算する．同様の処理が　B␣^3km␣α，C␣^5km␣β，D␣^1km␣β にも行われる．

時空間データの変換終了後，計算した流域の行列に対して時空間データの計算機を用い，時空間演算を行う．例えば，　作業 1 には「*」の時空間演算子があるため，A␣^3km が変換された流域の行列 と B␣^3km が変換された流域の行列の重複部分を計算する．全ての時空間演算が終了すると，計算された流域の行列の結果が再び時空間データ変換機に送られる．時空間データ変換機は，計算された流域の行列の結果を，設計者が作成したデータ処理プログラムが利用できるデータタイプに変換する（例えば，A と B の重複範囲にある店のオブジェクトの配列）．最後に，その結果は再度入出力処理モジュールに送られ，ユーザに返される．

5. Verbal Search プロトタイプシステム

4 章をもとに Google API を用いた地図と，Web 文書検索の一部を作成した．今回地図検索の例として，京都駅と京都水族館を用いた問い合わせを示す．京都駅とは京都府一大きな駅で，観光客によく利用されることで有名な施設である．京都水族館は，京都駅から 1.5km の位置にある施設である．

5.1 集合演算による検索

地図検索で「(京都駅␣800m) ＋ (京都水族館␣800m) ␣レストラン」と検索すると，図 7 のように，京都駅から周囲 800m と京都水族館から周囲 800m の範囲を合わせた部分に該当するレストランが表示される．

図 7　＋の例：京都駅と水族館の 800m 圏内にあるレストラン検索結果例

「(京都駅␣800m) -(京都水族館␣800m) ␣レストラン」の検索で，図 8 のように京都駅から周囲 800m から京都水族館から周囲 800m を除いた範囲にあるレストランが表示される．

図 8　-の例：京都駅から 800m 圏内にあり水族館から 800m 圏内にないレストラン検索結果例

「(京都駅␣800m) *(京都水族館␣800m) ␣レストラン」の検索で，図 9 のように京都駅から周囲 800m と京都水族館から周囲 800m の共通部分の範囲にあるレストランが表示される．

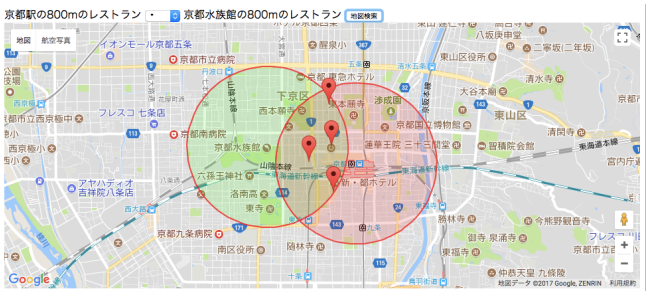

図 9　*の例：京都駅と水族館の 800m 圏内に共通するレストラン検索結果例

5.2 Web 検索実行例

Web での文書検索に応用した例を示す．検索キーワード「A␣10 行」で A というキーワードから 10 行分の文書のみを問い合わせが可能であるし，「A␣*10 行」なら A というキーワードから 10 行目にあたる文書の問い合わせとなる．「A␣^」で A より上に位置する文字列を問い合わせる．二区間では，[A-B] という問い合わせで，A と B というキーワードが出てくる間の文書を調査可能である．例として，キーワード「シャンシャン」「シンシン」で検索するとする．[シャンシャン-シンシン] で，シャンシャという単語が現れてからシンシンという単語が現れ

るまでの区間の文章，図の黄と緑の色がついた部分が返される．「シャンシャン␣^シンシン」でシャンシャンより上にシンシンが出てくる，緑，青，灰が該当する．「シャンシャン␣#<.png」でシャンシャンより左側にある png ファイル，赤い画像が対象となる．

図 10　文書検索の例

6. おわりに

本論文では，時空間検索を可能にする新たな演算方式を提案した．検索キーワード間の空白（␣）に着目し，空白の長さで距離や時間を表現できる，新たな７つの範囲演算子と，集合演算子での範囲検索を定義した．その後，それぞれを地図検索の例を用いて紹介し，演算を実現するための時空間検索システムの構成について述べ，最後に地図検索と web での文書検索の実例を示した．今後の課題として，提案したインタプリタまたは API を開発し，普及を目指す．

謝　辞

本研究の一部は，JSPS 科研費 16H01722，17K12686，17H01822 の助成を受けたものである．ここに記して謝意を表す．

文　献

[1] プラダンスジット, 田島敬史, 田中克己. ビデオデータ検索のための区間グルー操作と解のフィルタリング. 情報処理学会論文誌データベース（TOD）, 40(SIG03(TOD1)): 80-90, 1999.

[2] Sujeet Pradhan, Takashi Sogo, Keishi Tajima, Katsumi Tanaka. A New Algebraic Approach to Retrieve Meaningful Video Intervals from Fragmentarily Indexed Video Shots. In Proc. of Advances in Visual Information Management Visual Databased Systems (VDB5), pages 11-30, 2000.

[3] Katsumi Tanaka, Keishi Tajima, Takashi Sogo, Sujeet Pradhan. Algebraic Retrieval of Fragmentarily Indexed Video. In New Generation Computing, 18(4): 359-374, 2000.

[4] Sujeet Pradhan, Keishi Tajima, Katsumi Tanaka. A Query Model to Synthesize Answer Intervals from Indexed Video Units. In Transactions on Knowledge and Data Engineering, 13(5): 824-838, 2001.

[5] David G. Lowe. Object Recognition from Local Scale-Invariant Features. In Proc. of the International Conference on Computer Vision-Volume 2 (ICCV '99), page 1150, 1999.

[6] Herbert Bay, Tinne Tuytelaars, Luc Van Gool. SURF: Speeded Up Robust Features. In Proc. of European Conference on Computer Vision (ECCV 2006), pages 404-417, 2006.

[7] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, Cedric Bray. Visual Categorization with Bags of Keypoints. In Proc. of International Workshop on Statistical Learning in Computer Vision (ECCV 2004), pages 59-74, 2004.

[8] K. Fu, Y. C. Lu, C. T. Lu. Treads: A safe Route Recommender using Social Media Mining and Text Summarization. In Proc. of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2014), pages 557-560, 2014.

[9] J. Kim, M. Cha, T. Sandholm. SocRoutes: Safe Routes Based on Tweet Sentiments. In Proc. of the 23rd International Conference on World Wide Web (WWW 2014), pages 179-182, 2014.

[10] D. Quercia, R. Schifanella, L. M. Aiello. The Shortest Path to Happiness: Recommending Beautiful, Quiet, and Happy Routes in the City. In Proc. of the 25th ACM conference on Hypertext and social media (HT 2014), pages 116-125, 2014. DOI=http://dx.doi.org/10.1145/2631775.2631799

[11] M. Duckham, L. Kulik. "Simplest" Paths: Automated Route Selection for Navigation. In COSIT, Vol. 2003, pages 169-185, 2003.

[12] M. F. Bulut, Y. S. Yilmaz, M. Demirbas. Crowdsourcing Location-based Queries. In Proc. of 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops),pages 513-518, 2011.

[13] M. Choy, J. G. Lee, G. Gweon, D. Kim. Glaucus: Exploiting the Wisdom of Crowds for Location-Based Queries in Mobile Environments. In Proc. of ICWSM 2014, 2014.

[14] K. Church, J. Neumann, M. Cherubini, N. Oliver. The "Map Trap"? An Evaluation of Map Versus Text-based Interfaces for Location-based Mobile Search Services. In Proc. of the 19th international conference on World wide Web (WWW 2010), pages 261-270, 2010.

[15] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proc. of the 26th International Conference on Neural Information Processing Systems (NIPS '13), pages 3111-3119, 2013.

[16] ノイズを抑えた文献検索を行うには？ScienceDirect & Scopus での "近接演算子" の活用. エルゼビア・ニュースレター: 2015 年 8 月 19 日号, 2015.

[17] Oracle Text リファレンス, 11g リリース 2(11.2). B61357-06, 2015 年 10 月.