

テクニカルノート

# 検索ワード間の空間演算の提案と地図検索への応用

阪田 晴香<sup>1,a)</sup> パノット シリヤーラヤ<sup>1,b)</sup> 王 元元<sup>2,c)</sup> 河合 由起子<sup>1,3,d)</sup>

受付日 2018年6月7日, 採録日 2018年7月28日

**概要:** 本研究は, 距離や範囲, 方向, 大きさを組み合わせた空間検索の実現を目的とする. 従来複数キーワードによる検索は, ワード間の距離や範囲といった空間上の位置関係を考慮した検索が困難であった. 特に, 地図検索での任意のワード (現在地等の起点) から距離や範囲等の空間的制約に合致したワード (店舗等の対象物) の検索は, 効率性と精度ともに低かった. 本論文では, 地図検索において検索ワード間の空白に着目し, 各ワードを起点と対象物とし, ワード間の空白を起点から対象物までの位置関係を表現する第3のワードにとらえ, さらにそれらの演算を可能とする新たな問合せ空間演算子を定義し, 距離や範囲, 方向等の空間演算の検索方式を提案する. 具体的には, 10種類の演算子を定義し, それら演算子を使った空間演算に基づく地図検索 Web サービスを実装し, Google Maps との比較実験により検索精度および効率性を検証する.

**キーワード:** 空間演算, 空間検索, 地図検索, 空白キー

## Designing a Spatial Query Language for Space-Key Search

HARUKA SAKATA<sup>1,a)</sup> PANOTE SIRIARAYA<sup>1,b)</sup> YUANYUAN WANG<sup>2,c)</sup> YUKIKO KAWAI<sup>1,3,d)</sup>

Received: June 7, 2018, Accepted: July 28, 2018

**Abstract:** Our goal is to design a query language which can express complex spatial queries by a single textual query in an intuitive way. The new language can express queries including conditions on range, direction, time lengths, and sizes which are difficult to express by textual queries in conventional keyword-based search systems. To express these conditions, our proposed language introduces several spatial operators, and also uses “space characters” between keywords in order to express geographical distance or time-length between matching objects in an intuitive way. We explain the overview of our search method where we can retrieve maps by using complex spatial queries and discuss the results of a study carried out to evaluate the system through Amazon MT crowd sourcing.

**Keywords:** query language, space-key search, spatio-temporal operators

### 1. はじめに

映像や地図の検索サービスの需要は増しているが, 既存のキーワードによる空間や時間制約を含めた検索は利便性が低い. たとえば空間の問合せとして代表的な地図検索では「現在地から 500 m 圏内の蕎麦屋」を検索する場合, そのまま文章による問合せでは正解を得ることは困難であり, 一般的にまず, 地図上の現在地 (位置) を検索し, 次にその地図上で「蕎麦屋」のキーワードを用いて検索し, さらに「500 m」を距離目盛りからおおよその目視で絞り込むか, あるいは「500 m 圏内」の制約指定が検索サービ

<sup>1</sup> 京都産業大学コンピュータ理工学部  
Faculty of Computer Science and Engineering, Kyoto Sangyo University, Kyoto 603-8555, Japan  
<sup>2</sup> 山口大学大学院創成科学研究科  
Graduate School of Sciences and Technology for Innovation, Yamaguchi University, Ube, Yamaguchi 755-8611, Japan  
<sup>3</sup> 大阪大学サイバーメディアセンター  
CyberMedia Center, Osaka University, Ibaraki, Osaka 567-0047, Japan  
a) g1544647@cc.kyoto-su.ac.jp  
b) spanote@gmail.com  
c) y.wang@yamaguchi-u.ac.jp  
d) kawai@cc.kyoto-su.ac.jp



図 1 空間演算による地図検索結果例

Fig. 1 Example of map search result using proposed spatial operators.

スに用意されていれば利用する。このように既存の検索方法では、ユーザは 2 回以上の検索操作を必要とし、2 地点間の距離を考慮した検索は困難といえる。また、次のような検索はさらに困難をきわめ、ユーザはより多くの検索操作が必要となる。

- (Q1) 現在地から 500m 圏内で、さらに「北側にある」蕎麦屋や「南西方向にある」蕎麦屋 (方向検索)
- (Q2) 現在地から「50~90m 圏内にある」蕎麦屋 (範囲検索)
- (Q3) 現在地から 500m 圏内に加えて、「他の地点から 300m 圏内」と「共通する」蕎麦屋 (集合検索)

そこで本研究では、方向や範囲、時間長となる演算子を定義することで、問合せ回数の減少および検索精度を向上させる空間演算を提案する。また、提案した空間演算を地図検索 Web サービスとして実装・公開し、検索精度および効率性を、既存の地図検索として代表的な Google Maps の Web サービスと比較検証する。

提案手法は、従来のテキスト検索で用いられている「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定義する。たとえば、既存手法では「A B」も「A B」も「A と B が同時に含まれる」という問合せ文が生成されるため、検索結果は同じになる。ここで 2 つのワード間の空白を考慮することで「A を起点とした B」の問合せを実現でき、「A B」は A から最も近い B、「A B」は A から近い 5 件の B となる空間制約の検索が可能になる。しかしながら、A を起点として 100 個の空白を入力することは有用ではないため、空白長を距離として用いることを許す。たとえば、A を起点として 5km 離れた B の空間検索の問合せは「A 5km B」となる。

本論文で定義する空間演算子は、集合演算子と新たな範囲、方向、時間等を含む 10 種類である。これらを用いることで (Q1) の方向検索 (Q2) の範囲検索 (Q3) の集合演算をユーザは 1 回の問合せにより検索できる。図 1 に各空間演算による地図検索結果例を示す。(a) は方向演算「^」による (Q1) の北方向の問合せ結果例を示しており、(b) は差演算「-」による (Q2) の周囲検索結果、(c) は積演算「\*」による (Q3) の 2 地点間で共通するワード (店舗) 検索例で

ある。また、検索ワードの特性 (たとえば、レビュー評価や収容人数等) 検索のためのプロパティ演算子「\$」、時間演算子「#」を定義する。これによりたとえば、「現在地から 500m 圏内で 20 人規模の蕎麦屋」を「現在地 500m \$20 人 蕎麦屋」の演算より検索できる。

## 2. 関連研究

### 2.1 ビデオおよび地図における空間検索

空間演算として代表的な映像検索では、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、答えとなり得るすべての「区間」を動的に生成するグルー演算と呼ばれる区間演算群がある [1]。また、グルー演算のみでは不適切な解となるノイズを取り除く区間フィルタの概念も提案されている [2]。近年では、効率的な映像検索として、色やテキスト等の表層的な特徴だけでなく、映像中に出現するオブジェクトやイベントを識別し [3], [4], 意味内容に基づいた解析がなされている [5]。本研究では、空間と時間の単位を空白の長さで表現した空間検索のための新たな演算モデルを提案しており、これら映像の局所特徴検索への適用が可能となる。

地図検索ではユーザの入力した検索クエリによって検索精度や効率は大きく影響される。しかし、既存の地図検索サービス (Google Maps や Bing Maps 等) の検索クエリは、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。たとえば、ユーザが京都駅にあるレストランを検索する場合に「Restaurants in Kyoto station」の文章で検索可能だが、距離や件数といった明示的な空間制約を含めた検索は困難である。本論文で提案する新たな空間演算は、それらのワードの空間の関係を演算式の形式としてクエリ生成でき、たとえば、「Tokyo Station 500m Supermarkets」といった直感的で精度の高い検索を実現できる。

### 2.2 テキストによる意味的検索

テキスト検索では、Mikolov ら [6] 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。たとえば、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 [7]。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 [8]。本研究では、検索ワード間の空白 ( ) に着目し、起点に対する距離や範囲等のワード間の空間制約を用いた集合演算を可能にしてお

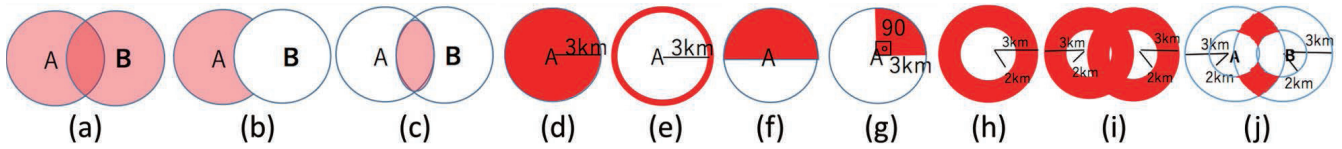


図 2 (a) 和 (b) 差 (c) 積 (d) 周辺 (e) 周囲 (f) 方向 (北) (g) 角度 (90) (h) 範囲 (i) 和と範囲 (j) 積と範囲

Fig. 2 (a) Union (b) Difference (c) Intersection (d) Within (e) Distance (f) Direction (north) (g) Angle (90 degree) (h) Range (i) Range and Union (j) Range and Intersection.

表 1 空間演算子

Table 1 Spatial operators.

演算子	$\sqcup^*$	$\sqcup^\wedge$	$\sqcup_-$	$\sqcup@$	$\sqcup[x-y]$	$\sqcup\$$	$\sqcup\#$
処理	場所	方向 (北・上)	方向 (南・下)	角度	範囲	プロバティ	時間

り、直感的な空間演算による検索を実現できる点が特徴となる。

### 3. 空間演算

本章では、10種類の空間演算子を定義し、空間検索演算を地図検索例を中心に提示する。まず、空間演算子を用いた問合せ演算の3つの前提条件を記す。

**定義 1:** 最小の空間演算の問合せ (SQ) は「 $A \sqcup [\text{空間演算子}] \text{空間長} \sqcup \alpha$ 」の形式とする。

A と  $\alpha$  はキーワードであり、A を起点とした空間上の  $\alpha$  の検索を意味する。なお、空間演算子がない場合は、空白 ( $\sqcup$ ) の個数分、近い順を検索する問合せとなる。

(例) 「 $A \sqcup \sqcup \sqcup \sqcup \sqcup \alpha$ 」は「A から近い 5 件の  $\alpha$ 」

(例) 「 $A \sqcup^\wedge 1 \text{km} \sqcup \alpha$ 」は「A から 1 km 圏内で北方向の  $\alpha$ 」

**定義 2:** 空間長は数値を含む単位の記述を許す。

空間演算子を用いる場合は、空間演算子に続いて数値と単位を記述する。

(例) 「 $A \sqcup 800 \text{m} \sqcup \alpha$ 」は「A から 800 m 圏内の  $\alpha$ 」

**定義 3:** 最小の空間演算 SQ の問合せを集合演算子の組合せ「 $SQ^1 \cup SQ^2 \cup \dots = SQ^+$ 」で記述できる。

(例) 「 $(A \sqcup 800 \text{m} \sqcup \alpha) + (B \sqcup 800 \text{m} \sqcup \beta)$ 」は「A から 800 m 圏内の  $\alpha$  と B から 800 m 圏内の  $\beta$  の和集合」

#### 3.1 集合演算子と方向演算子

集合演算子は、和「+」、差「-」、積「\*」とする。

和演算 (例 1) 「 $(A \sqcup 3 \text{km} \sqcup \alpha) + (B \sqcup 3 \text{km} \sqcup \alpha)$ 」

A から 3 km 圏内と B から 3 km 圏内の和演算 (図 2 (a))

差演算 (例 2) 「 $(A \sqcup 3 \text{km} \sqcup \alpha) - (B \sqcup 3 \text{km} \sqcup \alpha)$ 」

A の 3 km 圏内から B から 3 km 圏内の差演算 (図 2 (b))

積演算 (例 3) 「 $(A \sqcup 3 \text{km} \sqcup \alpha) * (B \sqcup 3 \text{km} \sqcup \alpha)$ 」

A から 3 km 圏内と B から 3 km 圏内の積演算 (図 2 (c))

周囲演算 (例 4) 「 $A \sqcup 3 \text{km} \sqcup \alpha$ 」

A の 3 km 圏内にある  $\alpha$  (図 2 (d))

次に空間演算子として、本項では方向演算子、プロバティ演算子、時間演算子の 7 種類を定義する。表 1 に新たに定義した 7 つの空間演算子を示す。また、下記にそれら演算子を用いた演算例をいくつか示す。

距離演算 (例 5) 「 $A \sqcup^* 3 \text{km} \sqcup \alpha$ 」

A から 3 km の周辺にある  $\alpha$  (図 2 (e))

方向演算 (例 6) 「 $A \sqcup^\wedge 3 \text{km} \sqcup \alpha$ 」

A から 3 km 圏内の北側方向にある  $\alpha$  (図 2 (f))

角度演算 (例 7) 「 $A \sqcup @ 90 \sqcup \alpha$ 」

A から 3 km 圏内の角度 90 度方向にある  $\alpha$  (図 2 (g))

範囲演算 (例 8) 「 $A \sqcup [3 \text{km} - 1 \text{km}] \sqcup \alpha$ 」

A から半径 1 km から 3 km の範囲にある  $\alpha$  (図 2 (h))

なお、差演算 (「 $(A \sqcup 3 \text{km} \sqcup \alpha) - (A \sqcup 1 \text{km} \sqcup \alpha)$ 」) としても記述することで、同様の演算結果となる。

#### 3.2 集合演算子と方向演算子による空間演算

定義 3 より集合・空間演算子による空間演算を実現する。

和演算と範囲演算 2 地点で周辺の  $\alpha$  を検索したい場合

(例 9) 「 $(A \sqcup [3 \text{km} - 1 \text{km}] \sqcup \alpha) + (B \sqcup [3 \text{km} - 1 \text{km}] \sqcup \alpha)$ 」

A と B の半径 1 km から 3 km の範囲内にある  $\alpha$  (図 2 (i))

積演算と範囲演算 2 地点で周辺にある共通の  $\alpha$  の検索

(例 10) 「 $(A \sqcup [3 \text{km} - 1 \text{km}] \sqcup \alpha) * (B \sqcup [3 \text{km} - 1 \text{km}] \sqcup \alpha)$ 」

A と B の半径 1 km から 3 km 範囲内の共通の  $\alpha$  (図 2 (j))

#### 3.3 プロバティ演算子と時間演算子

表 1 のプロバティ演算子「\$」は検索対象となる  $\alpha$  のプロバティ (属性) の大きさを単位とする。また、時間演算子「#」は他の演算子と組み合わせて使用する。

プロバティ演算 (例 11) 「 $A \sqcup 3 \text{km} \$ 10 \text{台} \sqcup \alpha$ 」

A から半径 3 km 圏内で 10 台の規模の  $\alpha$

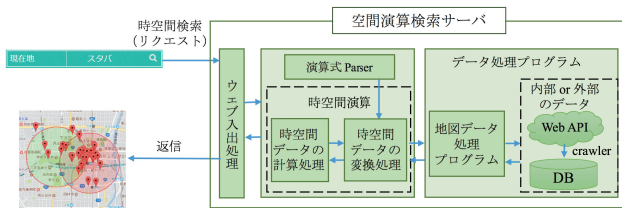


図 3 空間演算に基づく検索サービスのシステム構成例

Fig. 3 Overview of system configuration based on spatial operators.

#### 4. 空間演算に基づく空間検索 Web サービス

空間検索サービスは、Web リクエストの入出力となる Web 入出力処理、空間演算式の解析モジュール、データ管理の3つで構成される(図3)。空間演算式の解析モジュールは、演算式のParser、空間データの変換、空間データの計算の3つの処理機構で構成される。

まず、Parserはユーザが入力した空間演算式を構文解析し、空間演算処理とその処理順序を決定する。たとえば、演算式  $(A_{\square} \sim 3km_{\square} \alpha) * (B_{\square} \sim 3km_{\square} \alpha)$  は、まず、変数  $1 = (A, \sim, 3km, \alpha)$  と変数  $2 = (B, \sim, 3km, \alpha)$  となり、次に、各変数をデータ変換処理に渡す。

次にデータ変換処理では、変数の第1要素の位置と第3要素の距離から空間データ領域を抽出し、第3要素の条件と第4要素のワードを含む(ワード、緯度経度)のリストをデータ管理機構から取得する。たとえば「 $A_{\square} \sim 3km_{\square}$ 蕎麦」の変数の場合、データ管理機構からAの緯度経度を取得し、その緯度経度から3km範囲内かつAの緯度より大きい値かつ蕎麦屋のワードを含むタブルを問合せ、その結果のリストをParserへ返信する。

Parserは変数が2つ以上ある場合、データ変換処理のリストを計算処理にわたし、計算結果のリストをWeb入出力に渡す。たとえば「\*」の空間演算子では、「 $A_{\square} \sim 3km_{\square}$ 蕎麦」の(ワード、緯度経度)のリストと「 $B_{\square} \sim 3km_{\square}$ 蕎麦屋」のリストから重複部分を計算する。

データ管理は、内部のデータベースあるいは外部のデータベースとし、解析モジュールのデータ変換機構は各データベースの問合せ構文に従って変換する。具体的には、内部データベースの場合はSQL文等へ変換し、外部のGoogle Maps APIやFoursquare API, Twitter API等の場合は各APIリクエストへ変換する。

#### 5. 実装と検証

空間演算に基づく地図検索Webサービスを実装し、公開した\*1,\*2。なお、モバイル環境で一般ユーザを対象として、空間演算子を入力せず空白(□)のみで距離の近い順に店

\*1 <http://yklab.kyoto-su.ac.jp/~sakata/spatialQuery/>  
 \*2 シンプルバージョン(Android対応): <https://play.google.com/store/apps/details?id=com.kawaiLab.spatialQuery>

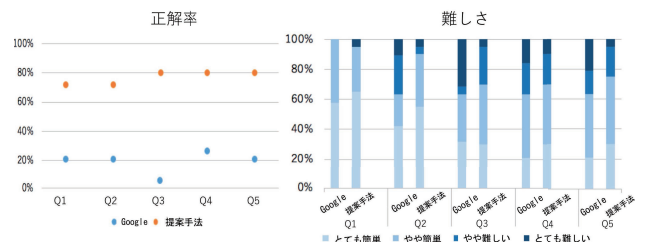


図 4 正答率と難しさ(4段階評価)の実験結果

Fig. 4 Experimental results of correct answer rates and difficulty (4-point scale).

舗等の対象物を空白の個数分だけ検索できるシンプルバージョンと、演算式による問合せが可能な開発者を対象としたフルバージョンの2種類を公開した。先に示した図1は本サービスを利用した演算式による問合せ結果例である。

#### 5.1 実験

提案した地図検索の検索精度および効率性を既存のGoogle Mapsと比較検証した。実験は、クラウドソーシング(AMT\*3)による40名の10代から60代の男女に、検索精度の検証として下記の5種類の質問(店名や店数の検索)を実施し、正解率を検証した。

- (Q1) Times square から4番目に近いピザ屋の名前
- (Q2) Times square から3km圏内のピザ屋の数
- (Q3) Times square の5km圏内と Empire state building の5km圏内両方に含まれるピザ屋の数
- (Q4) Times square の5km圏内に含まれるが Empire state building の3km圏内には含まれないピザ屋の数
- (Q5) Times square の5km圏内と Empire state building の5km圏内に共通のピザ屋の数

また、効率性検証として下記の質問をQ1からQ5の質問ごとに実施した。

- (Q6) この問題は簡単だったか(4段階評価)

なお、提案手法およびGoogle Mapsともにすべての質問の前に例題を1問提示した。Google Mapsの例題では、提供されている距離算出ツールの説明も行ったが、ツールの利用は任意とした。また、40名のうち提案手法は20名、Google Mapsは19名の回答を有効なものとし評価検証した。

#### 5.2 結果と考察

図4に実験結果を示す。Q1からQ5のすべての問いで提案手法の正解率が上回っており、全体平均はGoogle Mapsが19%に対して提案手法は75%の正解率となり、検索精度が大幅に向上した。一方で、Q6の効率性では、「4:とても簡単だった」と「3:やや簡単だった」の評価の全体平均はGoogle Mapsが71%、提案手法が80%と9%の向上にとどまった。これは、AMTでは正確な検索時間を計測

\*3 <https://www.mturk.com/>

できず、定性的評価となっていることも影響していると考えられるため、今後、検索時間を計測し定量的評価を行う予定である。

以上より、既存手法として代表的な Google Maps と比較して検索精度が約 4 倍向上し、定性的評価ではあるが検索の容易さは提案手法は約 80% とより高く、提案手法の複数の空間制約による地図検索の有効性が明らかとなった。

## 6. おわりに

本論文では、検索ワード間の空白を考慮した新たな空間演算方式を提案し、地図検索として実装公開した。代表的な既存地図検索との比較実験では、検索精度、効率性ともに提案手法の高さを確認できた。今後、空間演算を用いた新たな店舗推薦やナビゲーションへの適用、時間演算子による時空間演算を実装し、さらなる精度と効率性の向上を目指す。また、Web ページや映像検索への拡張を検討する。

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。

## 参考文献

- [1] Tanaka, K., Tajima, K., Sogo, T. and Pradhan, S.: Algebraic Retrieval of Fragmentarily Indexed Video, *New Generation Computing*, Vol.18, No.4, pp.359-374 (2000).
- [2] Pradhan, S., Tajima, K. and Tanaka, K.: A Query Model to Synthesize Answer Intervals from Indexed Video Units, *Trans. Knowledge and Data Engineering*, Vol.13, No.5, pp.824-838 (2001).
- [3] Lowe, D.G.: Object Recognition from Local Scale-Invariant Features, *Proc. International Conference on Computer Vision-Volume 2 (ICCV '99)*, p.1150 (1999).
- [4] Bay, H., Tuytelaars, T. and Van Gool, L.: SURF: Speeded Up Robust Features, *Proc. European Conference on Computer Vision (ECCV 2006)*, pp.404-417 (2006).
- [5] Csurka, G., Dance, C.R., Fan, L., Willamowski, J. and Bray, C.: Visual Categorization with Bags of Keypoints, *Proc. International Workshop on Statistical Learning in Computer Vision (ECCV 2004)*, pp.59-74 (2004).
- [6] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J.: Distributed Representations of Words and Phrases and their Compositionality, *Proc. 26th International Conference on Neural Information Processing Systems (NIPS '13)*, pp.3111-3119 (2013).
- [7] Elsevier R&D Solutions: 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).
- [8] Oracle Database: Oracle Text リファレンス, 11g リリース 2 (11.2), B61357-06 (2015).



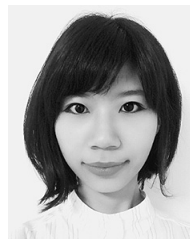
阪田 晴香

現在、京都産業大学コンピュータ理工学部ネットワークメディア学科在学中、問合せ文、自然言語処理技術に興味を持つ。



パノット シリアーラヤ

2013 年イギリスカンタベリーセント大学電子工学博士号取得。2014 年デルフト工科大学工業デザイン工学部研究員を経て、2017 年より京都産業大学情報理工学部研究員、現在に至る。博士 (電子工学)。仮想環境、高齢化人口のためのゲーミフィケーション設計技術、人間とコンピュータのインタラクションに関する研究に従事。



王 元元 (正会員)

2014 年兵庫県立大学大学院環境人間学研究科博士後期課程修了。同年京都産業大学コンピュータ理工学部研究員、2015 年名古屋大学大学院情報科学研究科研究員を経て、2016 年より山口大学大学院創成科学研究科助教、現在に至る。博士 (環境人間学)。主に異種メディア融合の研究に従事。日本データベース学会各会員。



河合 由起子 (正会員)

2001 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了。同年独立行政法人通信総合研究所 (現、国立研究開発法人情報通信研究機構)、2006 年京都産業大学理学部講師を経て、2018 年より京都産業大学情報理工学部教授、大阪大学サイバーメディアセンター特任教授 (常勤)、現在に至る。博士 (工学)。Web マイニング、情報推薦の研究に従事。電子情報通信学会、日本データベース学会各会員。本会シニア会員。

(担当編集委員 加藤 誠)