

# バス停留所におけるバス路線検索とその描画手法

加藤 りか<sup>1</sup> 山本 大介<sup>1</sup> 高橋 直久<sup>1</sup>

**概要:** 本論文ではバス停留所におけるバス路線検索とその描画手法に関して提案する。近年, Google Maps や NAVITIME, 各地域の交通事業者などによる経路検索サービスの普及が見られる。その一方で路線バスの停留所で得られる情報は限られており, 利用者が必要とする情報が足りていない場合もある。また, 従来の経路検索サービスにおいてもそれぞれ問題点が見られる。そこでバス停における路線案内を提供するシステムを従来の経路検索サービスの問題点をふまえた上で提案する。本研究では路線検索機能, 時刻表動的統合機能, 描画機能の3つの機能を提案, 実装しプロトタイプシステムを作成して評価実験を行った。

## Bus Route Search and Drawing Method at Bus Stop

RIKA KATO<sup>1</sup> DAISUKE YAMAMOTO<sup>1</sup> NAOHISA TAKAHASHI<sup>1</sup>

### 1. はじめに

近年, Google Maps や NAVITIME, 交通事業者などによる経路検索サービスの普及が見られる。これらは出発地や目的地, 駅名などを地図上もしくは地名により定め, 利用者の希望に応じた条件で経路を検索するものである。次に路線バスの停留所に関して考える。経路検索サービスが普及しているにも関わらず路線バスの停留所で得られる情報は限られている。次発までの時間の掲示をしている停留所も見られるが, 大まかな運行路と時刻表の掲示にとどまる場合が多い。そこでバス停における路線案内を提供するシステムを従来の経路検索サービスの問題点をふまえた上で提案する。また, 停留所では複数の利用者が同時にシステムを利用することも考えられる。限られたスペースで複数の利用者がそれぞれが必要とする情報を得られるようなシステムを目指す。本研究では以下の問題点に着目した。

**問題点 1** 停留所に掲示されている時刻表は系統ごとに分かれており行き先によっては複数の時刻表を比較し, 乗車するバスや時刻を確認する手間を要する。

**問題点 2** バス路線は実地図上の道路と対応付けたほうが地理的把握の観点で分かりやすいと考えられ, 路線をバス停間を直線で結ぶことにより表すことも考えられるがこの場合, 実際の運行路と離れてしまう。また,

複数の路線を同一の道路に表示する際, 重なってしまい描画が難しい。

問題点 1 に関して, 複数の時刻表を見比べる不便さがあることと, 全ての時刻表を掲示するスペースが必要となる。問題点 2 に関して, 例えば図 1 のように始めから 2 本目以降の路線をずらして描画すると重複区間以外の箇所においてずれが生じてしまう。また図 2 のように線の細さを変化させ, 太い線の上に細い線を描画すると重複区間以外の箇所でも細い線が見づらくなる可能性がある。このように路線全体に差をつけてしまうと見づらくなってしまう。これらの課題に対して次のアプローチを考えた。

**アプローチ 1** 乗車区間に合わせて複数路線分の時刻表を 1 つにまとめた時刻表を作成, 提示する。

**アプローチ 2** 重複区間を判別し, 重複区間のみに対して描画方法を変更する。

アプローチ 1 に関して, 利用者ごとの路線検索結果に応じた複数路線分を纏めた時刻表をその都度作成し, 地図上に提示することでいくつもの時刻表を見比べる手間を無くす。また, 全ての時間帯をまとめた時刻表を提示するのではなく現在時刻から数本のみを表示することでスペースを削減する。また鉄道と違いバスは時間帯によって停留所間の所要時間が異なる為, 時刻表検索と同時に乗車候補のバスの時刻表の直近の時刻を用いて所要時間の検索も行う。アプローチ 2 に関して, 複数の路線が重複している区間にそれぞれ応じて線をずらす方向, 量を定めて再描画をする

<sup>1</sup> 名古屋工業大学大学院工学研究科 Graduate School of Engineering, Nagoya Institute of Technology

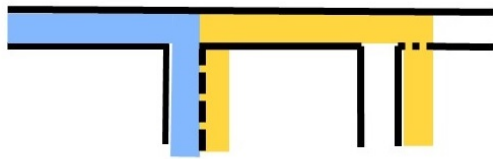


図 1 路線全体をずらした場合

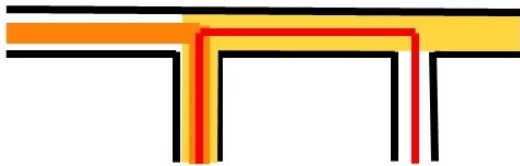


図 2 線の細さを変えた場合

ことで視認性の高い路線図の作成を可能にする。

## 2. 提案システムの概要

以下のような特徴をもつシステムを提案する。

**特徴 1** バス路線図を実際の運行路通りに地図上に描画することにより、バス路線と地図の関係性を強調する。

**特徴 2** 乗車停留所と降車停留所を含む複数のバス路線の時刻表を動的に統合し時間帯に合わせた時刻表と所要時間を表示する。

**特徴 3** 重複区間を判別し、区間ごとに重複路線をずらし路線図を描画する。

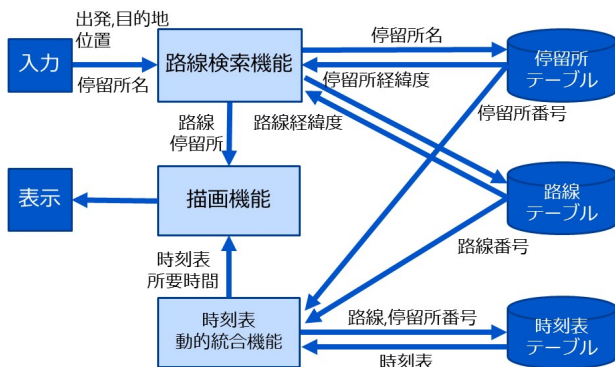


図 3 システム構成図

### 2.2 システムの各機能

システムの機能について説明する。なお、各機能内で用いられたデータに関しては第 3 章 1 節で説明する。

#### 2.2.1 路線検索機能

路線検索機能では指定した停留所までの路線検索を行い、バス路線のストロークと停留所に関する情報を得る。まず、指定した 2 停留所間を通る路線のストローク、路線番号、路線名を路線テーブルと停留所テーブルから取得する。また、バス路線のストロークからどの区間にどの路線が重複しているのかを ArrayList に格納する。停留所の指定方法は降車停留所名を入力する方法と、地図上で任意の位置をクリックすることによりその場所から最寄の停留所を取得する方法の 2 つがある。

#### 2.2.2 時刻表動的統合機能

路線検索機能によって得られた路線番号と現在時刻を用いて時刻表テーブルから時刻表を取得し統合する。これにより複数路線分をまとめた時刻表を得ることができる。また、時刻表テーブルから求めた乗車停留所の直近の時刻を用いて所要時間の計算を行い乗車停留所から各停留所までの所要時間を計算する。

#### 2.2.3 描画機能

路線検索機能によって得られた路線のストロークと重複区間を地図上に描画する。また、時刻表動的統合機能によって得られた時刻表と通過する各停留所までの所要時間も描画する。時刻表は複数路線分まとめられており、どの路線の時刻か判別が難しくなる為、各時刻の隣に路線名も表示する。

各機能の詳細は第 3 章 3 節で説明する。

## 3. 提案システムの実現法

本システムではバス路線に関するデータベースに 3 つのテーブルを用意し、利用した。それぞれについて説明する。

### 3.1 停留所テーブル

バス停留所に関するデータが格納されている。これは国土数値情報のバス停留所データを PostgreSQL に格納したものである。表 1 にテーブルの様子を示した。停留所番号

表 1 停留所テーブル

カラム名	型	説明
gid	int	停留所番号
p11_001	varchar	停留所名
p11_003	varchar	事業者名
p11_004	varchar	通過路線名
geom	geometry(Point)	停留所位置

は各停留所に与えられた番号である。停留所名は各停留所の名前である。事業者名は当該停留所を通過する路線の事業者名である。通過路線名は当該停留所を通過する路線の路線名である。停留所位置には geometry 型のデータとして停留所の緯度と経度が格納されている。

### 3.1.1 路線テーブル

バス路線に関するデータが格納されている。こちらも国土数値情報のバスルートデータを PostgreSQL に格納している。表 2 にテーブルの様子を示した。路線番号は各路線

表 2 路線テーブル

カラム名	型	説明
gid	int	路線番号
n07_002	varchar	事業者名
n07_003	varchar	路線名
geom	geometry(MultiLineString)	路線データ

に与えられた番号である。事業者名は当該路線を運行する事業者名である。路線名は各路線の名前である。路線データには geometry 型のデータとして路線を構成する点の緯度と経度が格納されている。

### 3.1.2 時刻表テーブル

それぞれのバス停留所における路線ごとに分けた時刻表が格納されている。これは新たに作成したものである。表 3 にテーブルの様子を示した。停留所番号は表 1 の停留所

表 3 時刻表テーブル

カラム名	型	説明
gidstop	int	停留所番号
gidline	int	路線番号
houkou	int	方向
youbi	int	曜日
time	varchar	時刻

番号と同じく各停留所に与えられた番号である。路線番号は表 2 の路線番号と同じく各路線に与えられた番号である。方向は上りなら 0, 下りなら 1 が格納されている。曜日は平日ダイヤなら 0, 土曜ダイヤなら 1, 日曜祝日ダイヤなら 2 が格納されている。時刻は該当する時刻表が順に格納されている。

### 3.2 時刻表からの所要時間の算出法

路線検索機能によって得られた通過路線の路線番号が格納された ArrayList から路線番号を取得し、時刻表テーブルに乗車停留所の時刻表を問い合わせる。ここで問い合わせた時刻表データは時刻ごとに ArrayList に格納する。現在時刻から直近のものが ArrayList の何番目に格納されているかを求める。同じ路線であれば格納された順番が同じものは同一のバスの時刻とみなせる為、時刻表テーブルに路線番号から停留所番号と時刻表を取得し、それぞれの停留所において先ほど求めた順番にあたる時刻を求め、乗車停留所の発車時刻を引くことにより算出した所要時間とその該当する停留所番号をそれぞれ ArrayList に格納する。

### 3.3 重複区間に関する処理

#### 3.3.1 重複区間の取得

2本の路線のストロークの一区間の緯度と経度がそれぞれ等しい時、その区間を重複しているとみなす。地図上で出発地点と目的地を指定するかもしくは乗車停留所と降車停留所を入力することにより路線検索を行う。地図上で指定した場合は最寄の停留所を停留所テーブルより取得する。PostgreSQL によりデータベースに格納した停留所テーブルと路線テーブルから乗車停留所と指定した停留所名を用いて通過路線の路線番号と路線名、路線のストロークを取得し路線ごとに緯度と経度に分け ArrayList に格納する。路線のストロークは MULTILINESTRING 型で格納されている。これらを文字列として取得後、数値に変換し格納する。比較は路線のストロークの緯度と経度を順に 2 つずつ取り出す。あらかじめ、重複区間の情報を入れる ArrayList を用意しておき、検索結果から得られた路線に対して重複している区間があればその都度データを追加していく。重複が起きていることがわかった場合、ArrayList に該当区間の緯度と経度と重複が起きていた路線の路線番号を格納する。最終的に全ての重複が起きている区間の始点の緯度と経度と終点の緯度と経度と重複している路線番号が格納された ArrayList を得る。図 4 に最終的に得た ArrayList の 1 つのデータを示す。これが重複している 1 区間分のデータとなるので更に重複している分だけ ArrayList に格納されている。

始点の経度
始点の緯度
終点の経度
終点の緯度
路線番号
路線番号
⋮

図 4 重複区間の ArrayList

#### 3.3.2 重複区間の描画

重複区間は 3.3.1 で最終的に得た ArrayList から順に 1 区間ずつ取り出して描画する。図 5 は 1 区間を取り出したものである。橙色の線を 1 本目の路線として考える。まず、青色の矢印のような始点から終点までの線に対する垂直ベクトルを考える。この垂直ベクトルの長さを赤い矢印のように変化させることによりわずらす量を路線によって変化させていく。1 本目の路線を中心とし、重複している路線をそこから垂直ベクトルの長さ分左右にずらして描画する。全体を 1 方向にずらした際には重複区間以外の箇所までもがずれてしまったり、重複数の異なる区間の描画が見やすいものでなかったりしたが、重複区間ごとに路線のわずらす量を変化させることにより視認性があがった。

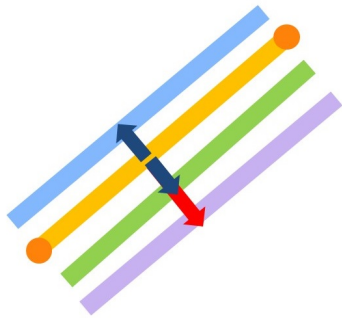


図 5 重複区間の描画

## 4. プロトタイプシステム

提案システムの実現の為、Java 言語を用いて名古屋市の市バスの路線検索を行うプロトタイプシステムを作成した。出発地と目的地を地図上でクリックして最寄の停留所を指定するか、乗車停留所と降車停留所を入力することで重複区間を考慮した路線図と各通過停留所までの所要時間、複数路線分の時刻表をまとめた時刻表、次発発車予定までの時間を表示する。尚、プロトタイプシステムでは重複区間の描画と時刻表の掲示に重点を置いた為、複数の利用者が同時に検索をするのではなく、一人の利用者がシステムを使うこととした。その為、利用者ごとの経路を表示するのではなく一人の利用者が利用可能な路線を重複を考慮して表示させた。

### 4.1 システムの動作

システムの画面は図 6 のようになる。地図上部にある 2 つのテキストフィールドに乗車停留所名と降車停留所名を入力するか、出発地と目的地を地図上でクリックし停留所を指定する。そしてバス路線検索と書かれたボタンをクリックすることにより指定した任意の 2 停留所間の路線検索が行われる。検索結果が表示された画面を図 7 に示す。表示結果について詳しく説明する。

#### 4.1.1 停留所、路線に関する表示

図 7 を拡大したものを図 8 に示す。円は停留所の位置を表しており、青枠に水色の円は乗車停留所もしくは降車停留所である。また、青い円は途中の停留所である。円の中に書かれた数字は、乗車停留所からの所要時間を表している。円右上の青い四角内には各停留所名を表示した。検索した区間の停留所のみを表示させることで分かりやすくした。各路線は色で違いを表した。乗車区間は青、緑、赤、黄、水色の 5 色で色分けされているため、5 路線が通過していることがわかる。

#### 4.1.2 時刻に関する表示

図 7 の左中央部の時刻表部分を拡大したものを図 9 に示す。一番上の緑色の四角には乗車停留所名と降車停留所名を表示した。赤色の 5 つの四角には複数の時刻表を 1 つにまとめたものを上から現在時刻に近い順に 5 つ表示した。時刻の隣の枠にはどの路線であるかが表示されている。一番下の緑色の四角には現在時刻から直近で発車するバスが来るまでの時間を表示した。

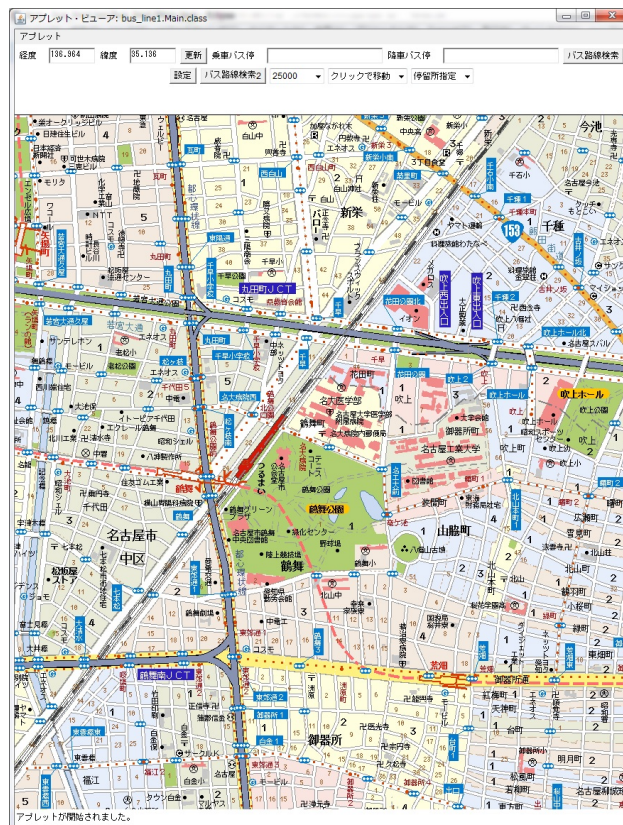


図 6 システム画面

## 5. 評価実験

### 5.1 実験方法

被験者(名古屋工業大学の学生 8 名)に従来システムと作成したプロトタイプシステムの両方を使ってもらい、システムに関するアンケートに回答してもらい。従来のシステムとして名古屋市交通局が提供していたなごや地図ナビを用いた。また、事前に用意したバス路線に関する質問に回答してもらい、システムを利用してもらった。また、提案システムで挙げた 3 つの機能について計算時間を測定した。路線検索の方法は地図上で出発地と目的地を指定する方法ではなく、事前に用意した停留所名を入力してもらい実験を行った。





図 7 検索結果表示



図 9 時刻に関する表示

- 手順 4 作成したシステムの利用方法を説明する。
- 手順 5 作成したシステムを用いて質問に回答してもらう。
- 手順 6 作成したシステムに関するアンケートに回答してもらう。

### 5.1.2 使用した質問とアンケート

手順 2, 手順 5 で用いた質問について説明する。被験者には図 11 のような乗車停留所名, 降車停留所名と以下の 2 つの質問が書かれた紙を渡し, 回答してもらった。従来システムで検索を行ってもらった区間と, 作成したシステムで検索を行ってもらった区間は異なる。

- 直近で発車するバスは何時何分発ですか。
- この区間はいくつの路線が通過していますか。



図 8 停留所, 路線に関する表示

乗車  
停留所名 → 降車  
停留所名

Q 直近で発車するバスは何時何分発ですか。

Q この区間はいくつの路線が通過していますか。

図 10 質問用紙

### 5.1.1 実験の手順

以下の手順で実験を行った。

- 手順 1 従来システムの利用方法を説明する。
- 手順 2 従来システムを用いて質問に回答してもらう。
- 手順 3 従来システムに関するアンケートに回答してもらう。

手順 3, 手順 6 で用いたアンケートについて説明する。被験者にはそれぞれのシステムを使い, 全ての質問に回答してもらったあとに以下の 6 つの項目に関して 1 から 5 の 5 段階評価でそれぞれのシステムに関して評価してもらった。5 段階は 1 はそう思わない, 2 は少し思わない, 3 はどちらでもない, 4 は少しそう思う, 5 はそう思う, である。

- 停留所の指定は容易であった
- 路線の把握ができた
- 十分な情報を得られた

- 操作は簡単であった
- 時間がかからないと感じた
- 不安なくバスに乗車できそうである

また、各システムを使って初めての感想や意見を自由記述欄を設けて書いてもらった。

## 5.2 結果と考察

システム利用後に回答してもらったアンケートの結果を表4と表5にまとめた。表4は従来システムに関して、表5は作成したシステムに関するものである。8人の各被験者の項目別の評価の値と、項目別の被験者分の評価の値の平均値をそれぞれ示した。また、それぞれのアンケートの各項目の平均値を棒グラフにまとめたものが図12である。図12を見ると、全ての項目において従来システムより作成したシステムのほうが良い評価を得られている。

また、路線が一気に表示されるので見やすかったとの意見が得られた。これは本研究において取り組んだ複数路線の重複区間の描画方法の改善によるものだと考えられる。一般的な路線検索システムでは、従来システムのように複数路線は地図上に1路線ずつしか表示されなかったり、重なって表示されてしまっていたりしていた。しかし重複区間の描画の改善により同時に必要とする路線のみを複数表示させることができるようになったことからこの意見が得られたと考えられる。これにより作成したシステムにより路線検索の利便性の向上が見られる。

しかし、従来システムと作成したシステムの項目ごとの差をみていくと十分な情報を得られた、という項目が一番差が少ない。表4と表5を比較すると被験者によっては従来システムのほうが十分な情報を得られたと感じている。これは作成したシステムから得られる情報の量に問題があると思われる。本研究では時刻表と路線図の問題点に重点を置いており、なるべく利用者が必要な情報のみを表示するようにしていた。このため被験者は得られる情報量が少ないと感じたと考えられる。必要なのに足りないと感じた情報に関しては後程この節内で述べる。

また、目的のバス停に印をつけてほしいとの意見が挙げられた。停留所の表示に関するものである。これは情報量の多い地図上では停留所を見つけづらいことが原因だと考えられる。各停留所位置の円の中に記された所要時間も初めは何を示しているのか分からなかったという意見も実験中に出された。この2点に関しては表示方法の改良が必要だと考えられる。今より更に文字を大きくしたり配色の変更を検討したい。

次に、作成したシステムにおいて不安なくバスに乗車できそうである、という項目の平均値が他の項目の平均値と比べて低いことが表5から読み取れる。作成したシステムを利用することによってこのバスに乗車すれば必ず目的地に到着することができる、といった確信に繋がれなかつ

たと思う。これは前に述べた情報量が少ないことや視覚的な問題や検索方法の問題に起因するものだと考えられる。個々の問題を解決することにより利用者の不安を減らすことができると考えられる。

表4 従来システムに関するアンケート

	A	B	C	D	E	F	G	H	平均値
停留所の指定は容易であった	1	1	1	4	3	1	3	1	1.875
路線の把握ができた	1	4	4	4	2	1	5	1	2.75
十分な情報を得られた	1	5	2	4	2	2	4	3	2.875
操作は簡単であった	1	2	1	2	1	2	2	2	1.625
時間がかからないと感じた	1	1	1	2	1	2	2	2	1.5
不安なくバスに乗車できそうである	1	4	1	2	2	1	4	1	2

表5 作成したシステムに関するアンケート

	A	B	C	D	E	F	G	H	平均値
停留所の指定は容易であった	5	5	3	4	4	5	5	5	4.5
路線の把握ができた	4	4	5	5	5	5	3	5	4.5
十分な情報を得られた	5	3	5	4	5	4	4	5	4.375
操作は簡単であった	5	5	5	4	5	4	4	4	4.5
時間がかからないと感じた	5	4	5	4	5	4	4	5	4.5
不安なくバスに乗車できそうである	4	4	5	4	5	4	3	4	4.125

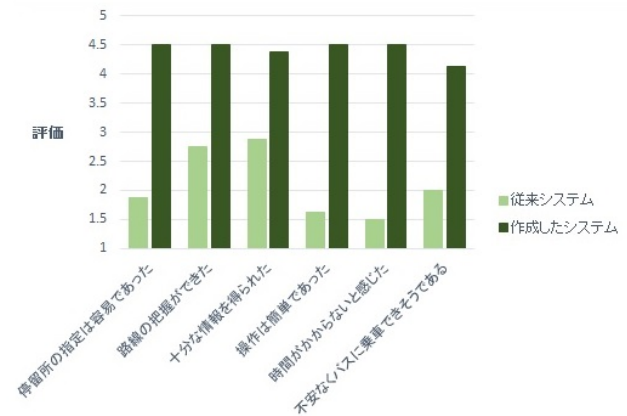


図11 アンケート結果

最後に、機能ごとの計算時間の平均を表6に示した。描画機能に関しては路線、重複区間、時刻表の3つに分けて計測した。路線検索機能と時刻表動的統合機能の計算時間に差があるのが見られる。これは、路線検索機能で用いた停留所テーブルと路線テーブルは第3章1節で述べたように国土数値情報のデータをPostgreSQLに格納したものであるが、時刻表動的統合機能で用いた時刻表テーブルは本研究のために新たに作成したものであるためデータの数が限られている。そのためデータベース内での検索時間の差と考えられる。また、描画機能では路線の描画が最も時間

を要している。これは描画方法に問題があると思われる。路線は1路線ずつ上描きしており、重複区間の場合は更にも上から重複区間を描画している。つまり3路線の重複が起きている区間であれば同じ区間を4回描画していることになる。このような不要な描画を減らすことにより描画に要する時間を減らすことができるのではないかと考えられる。

表 6 システムの計算時間

	平均計算時間 (ミリ秒)
路線検索機能	204.67
時刻表動的統合機能	3.57
描画機能 (路線)	15.55
描画機能 (重複区間)	3.33
描画機能 (時刻表)	1.48

## 6. 関連研究

文献 [3] では既存のバス時刻表検索システムを地図と Google マップの連動により利便性を向上させた携帯端末向けシステムの提案をしている。本研究と同様に地図を利用した時刻表の検索が行われている。GPS を用いてユーザの現在地近くの停留所と路線が地図上で把握できる。また、路線から乗降車停留所名を選択することにより時刻表検索ができる為、事前情報が少ない場合でも容易に検索ができると考えられる。また、機能集約型インテリジェントバス停の研究開発 [9] ではバス路線を含めた経路探索のアルゴリズムの考案と実際に開発した停留所向けのシステムに関して述べている。本研究の目標とする目的地によって複数の時刻表をまとめる動的な時刻表の作成も行われている。このシステムでは情報表示用パネルと入力用タッチパネルの2つのパネルを用いて操作する。また、検索時間の短縮を目標としてあげており、利用者の利便性の向上につながっていると考えられる。

## 7. 終わりに

本論文では複数路線分の時刻表を比較する手間や重複区間の見づらさといった従来の経路検索サービスの問題点の解決を行い、より利用しやすい路線検索のためのシステムを提案した。路線をバス路線のストロークを用いて描画し、複数路線が通過する場合の複数路線分をまとめた時刻表の作成と提示をし、重複区間を考慮した描画方法をそなえたプロトタイプシステムを名古屋市の市バスの路線情報を用いて作成した。さらに提案システムによって利用者の路線検索の利便性が向上したかどうかを明らかにするため、評価実験を行った。実験によって、提案システムは従来システムと比べて路線図に関する視認性を高めることができると示せた。今後の課題としてバス停留所における利用を考えた場合、複数の利用者が同時に操作する状況が考えられ

る。プロトタイプシステムは一人の利用者のみが操作するものとして実装したが、複数の利用者が同時に操作できるように実装したい。本論文で提案した機能が応用できるのではないかと考えている。例えば重複区間の描画方法に関しては全ての路線を表示するのではなく、利用者一人につき乗車区間のみを一路線に表示することで一枚の地図上で複数の路線検索結果が表示できると考える。

## 謝辞

本研究は JSPS 科研費 26330136, 25700009 の助成を受けたものです。この場を借りまして感謝の意を表します。

## 参考文献

- [1] GoogleMaps  
入手先 <<https://maps.google.co.jp/>> (参照 2016-02-02)
- [2] NAVITIME  
入手先 <<http://www.navitime.co.jp/>> (参照 2016-02-02)
- [3] 吉村元秀, 松田佳奈実, 地図を用いたバス路線および時刻表検索システムの作成, 長崎県立大学研究紀要 Vol.15, pp.225-229, 2014
- [4] 川村尚生, 菅原一孔, バスネットワークのための実用的な経路探索システム, 情報処理学会論文誌 Vol.48, No.2, pp.780-790, 2007
- [5] 清水畑朋子, 大信田祥代, 佐藤永欣, 高山毅, 村田嘉利, FeliCa 乗車券と降車バス停情報によるバスの遅延に対応した観光案内システム, 第 73 回全国大会講演論文集 2011 Vol.1, pp.67-68, 2011
- [6] 菊池新, 水本理子, 大西啓介, 寺尾浩俊, 時刻表データベースに基づく経路探索エンジンとその応用, 全国大会講演論文集 第 57 回 (コンピュータと人間社会), pp.398-399, 1998
- [7] 道田英明, 田辺政昭, 中川剛志, 駅における情報提供サービスに関する研究, Technical review JR East Vol.24, pp.41-46, 2008
- [8] 原田麻紀, 山本大介, 片山喜章, 高橋直久, 道や角の分かりやすさを考慮した構内道案内システム, 全国大会講演論文集 Vol.1, pp.683-685, 2012
- [9] 機能集約型インテリジェントバス停の研究開発  
入手先 <[http://www.soumu.go.jp/main\\_sosiki/joho\\_tsusin/scope/event/h21yokousyu/poster/tiiki25.pdf](http://www.soumu.go.jp/main_sosiki/joho_tsusin/scope/event/h21yokousyu/poster/tiiki25.pdf)> (参照 2016-05-07)
- [10] 名古屋市交通局 なごや地図ナビ  
入手先 <<http://map.kotsu.city.nagoya.jp/map/>> (参照 2016-02-02)
- [11] Java  
入手先 <<https://java.com/ja/>> (参照 2016-02-04)
- [12] PostgreSQL  
入手先 <<http://www.postgresql.org/>> (参照 2016-02-04)
- [13] PostGIS  
入手先 <<http://postgis.net/>> (参照 2016-02-04)
- [14] 国土数値情報  
入手先 <<http://nlftp.mlit.go.jp/ksj/>> (参照 2016-02-04)