

平成 30 年度 特別研究報告書

屋内における簡易位置推定と経路探索の検討

龍谷大学 理工学部 情報メディア学科

T150503 竹垣 陵

指導教員 三好 力 教授

内容梗概

近年, スマートフォンなどの携帯端末が普及し, ほとんどの人が通信端末を持つようになった. それらの, ほぼ全てのものに位置情報取得サービス GPS 機能が備わっており, 誰でも簡単に現在位置を取得できるようになった. 屋内測位の技術は様々な方式での先行技術が存在する. しかし, どの先行技術においても位置情報取得の正確さとコストの低さが両立できない. 本実験では, 正確さの向上のため, 屋内を含む施設内において道案内のための利用者の現在位置の取得に限定し, 低コストの実現のために現在位置の周辺情報からデータベース検索により, システム上で現在位置を推定する手法を用いてシステムを作成し検討する. 実験の結果としては, 屋内の特定の施設内での位置情報の取得に限定したことで正確かつ, 低コストの位置情報推定システムを作成することができた. さらに, 道案内プログラムを作成することで位置情報推定の実用性を確認することができた.

目次

第1章	はじめに	1
第2章	既存技術と問題点	2
2.1	Wi-Fi 測位	2
2.2	BLE (Bluetooth Low Energy)	2
2.3	地磁気測位	2
第3章	提案手法	3
3.1	提案概要	3
3.1.1	現在位置の位置推定	3
3.1.2	QRコードの使用	3
3.1.3	道案内プログラムの作成	3
3.2	システム概要	4
3.2.1	データベース	4
3.2.2	QRコードの作成	4
3.2.3	道案内のアルゴリズムとフローチャート	5
第4章	実験	7
4.1	実験の対象	7
4.2	実験環境	8
第5章	実験および考察	9
5.1	実験 1:位置推定	9
5.2	実験 2:QRコードの作成と読み取り	9
5.3	実験 3:経路探索	11
5.4	実験 4:利便性の確認	15
5.5	考察	15
第6章	おわりに	16

第1章 はじめに

近年、スマートフォンなどの携帯端末が普及し、ほとんどの人が通信端末を持つようになった。それら、ほとんどのものに位置情報取得サービス GPS 機能が備わっており、誰でも簡単に現在位置を取得できるようになった。

これを受けて、地方自治体やテーマパークなどではデジタルスタンプラリーを実施するなど位置情報を用いた産業が増加した。また、多くの人々がスマートフォン等でマップアプリを使用し、現在地から目的地までの経路を検索している。

最近では、GPS の精度も上がり非常に正確な現在位置の取得が可能である。しかし、これは GPS の受信がしやすい屋外(空の見える場所)に限り、屋内(空の见えない場所)に入ると位置情報の精度は格段に下がるか取得不能になる。そのため、屋内測位の技術が求められる。屋内測位が可能になるとショッピングモール内や大きな駅等での道案内が可能になり、利用者の利便性が大きく向上する。

屋内測位の技術は様々な方式での先行技術が存在する。しかし、どの先行技術においても位置情報取得の正確さとコストの低さが両立できない。本実験では、正確さの向上のため、屋内を含む施設内において道案内のための利用者の現在位置の取得に限定し、低コストの実現のために現在位置の周辺情報からデータベース検索により、システム上で現在位置を推定する手法を用いてシステムを作成し検討する。

施設内で現在位置から目的地まで誘導できると、施設公式のアプリやサイト、施設内案内板で顧客をうまく誘導できるので、施設側の収益機会が増加すると考えられる。あるいは GPS では収集できない利用者の移動データを入手することができる。

第2章 既存技術と問題点

2.1 Wi-Fi 測位

Wi-Fi 測位は、複数の Wi-Fi アクセスポイントから受信した電波強度の違いから三点測位演算を用いて、現在位置を算出する方式である。利点として、Wi-Fi アクセスポイントは既に広く普及していることから、新たな投資を必要としないことが挙げられる。しかし、Wi-Fi アクセスポイントのカバーエリアは半径数十メートルであるため、精度の高い位置情報の取得をする場合はかなりの密度でポイントが必要になるため、コストが高くなる。

2.2 BLE (Bluetooth Low Energy)

BLE は、複数のビーコンから受信した Bluetooth 電波強度から三点測位演算をして現在位置を取得する方式である。この方式の利点は、Bluetooth 電波を発信するものが動いていても計測が可能なことである。信号強度は半径十メートル程度なので発信器を高密度に設置する必要があるのは Wi-Fi 測位と同様であるため、コストが高くなる。

2.3 地磁気測位

地磁気測位は、建物の鉄材から発生する地磁気のパターンを測定してデータベースを作成し、スマートフォンの地磁気センサーによって現在位置を取得する方式である。この方式は設備投資が不要で、電気も使わないのでコストが低い。建物に大きな変化がなければ一度作成したデータベースは比較的安定するが、大型車両や鉄道車両などの地磁気を乱す物体が近くにあると正確な測定ができなくなる。

第3章 提案手法

3.1 提案概要

3.1.1 現在位置の位置推定

屋内での位置取得が廉価と精巧を兼ねることが難しいので、位置取得ではなく位置入力に対応することにした。屋外と違って屋内の施設ならば目の前、周辺に何らかのブースが見え、その数は有限である。ブースごとに数字を割り振り、その数字をノード ID と呼ぶことにし、ブース名をノード ID に変換するデータベースを作成する。

具体例（本屋,01）（100円ショップ,02）…（蕎麦屋,n）

ノード ID と用意した地図上の座標を結び、現在位置の推定を行う。

ブース名を利用者が入力する場合、データベース内のブース名と完全一致しないことが予想される。そういった呼称の違いは、例えば本屋が目の前にあった場合「本屋」「書店」「〇〇書店」など様々な呼び方を同一のノード ID で複数登録する。また、目的地入力も文字入力で行う。

3.1.2 QRコードの使用

各ブースの前などにあらかじめ作成したノード ID が読み取れる QR コードを貼り付けておく。QR コードはその名の通り Quick な Response を返してくれるので現在位置の位置推定には適している方法だと思われる。

前述のようなブース名の呼称が異なる問題や、高齢者など文字入力が困難な利用者を想定すると QR コードは良い手法と考えられる。

しかし、QR コード自体をブースのどこに、いくつ、どのような間隔で貼り付けることや、ブース同士での貼り付け方の統一をとるなどの手間がかかることが欠点である。

3.1.3 道案内プログラムの作成

目的地の入力にて現在位置と繋げて道案内プログラムを作成し、位置推定の利用価値を図る。

経路探索手法には「ベルマンフォード法」を用いて、ノード ID で二次元の接続行列を生成し、隣接ブース同士を距離や段差などを考慮して移動コストを計算した値で結ぶ。離れたブース同士は原則移動不可とし、間に何も無い場合以外は結ばず負の閉路にする。全てのブース同士が直接または別ブース経由の間接的に線で結ばれたら経路探索可能状態になる。出発地(現在地)と目的地を入力すると2つのブース同士を結ぶ全ての経路を探索し、一番移動コストの低い経路を出力する。アルゴリズムの詳細は後述する。

3.2 システム概要

3.2.1 データベース

利用者の位置情報を、利用者自身に入力してもらうことで起こる、類語による呼称の違いや、かな漢字表記の違いを統一するためのデータベースを作成する。また、作成したデータベース内の各ノード ID と各類語等を結びつけて登録を行う。目的位置の入力も同時に行う。

3.2.2 QR コードの作成

ブースの見やすい場所に貼りつけて、読み取りやすかつ簡単に、QR コードを作成できるツールを探して利用する。QR コードにはブース ID 情報を載せ、ブース名入力の手間を省く。

QR コードを Web 上で簡単に作成できるサイトは以下のサイトなど多数存在する。

QR のススメ

(<https://qr.quel.jp/>)

CMAN インターネットサービス

(<https://www.cman.jp/QRcode/>)

QR コードメーカー

(<https://m.qrqrq.com/>)

その中でも QR コード作成と同時に文字を打ち込める CMAN インターネットサービスを今回採用した。

読み取る情報はブースのノード ID で、店舗名を囲み文字という機能を使って外枠に表示した。QR コードを使わず文字入力の場合でも、作成した QR コードを店前に貼り付けておくことで標準の店舗名が分かり、入力の手助けとなる仕組みである(図1)。

大垣書店



図1 作成した QR コード (店舗名,ノード ID)=(大垣書店,24)

3.2.3 道案内のアルゴリズムとフローチャート

施設内の店舗, 昇降方法, 目印等を計測し, ブースの地点数を決定.

ブース同士の二次元配列を生成する. 隣り合うブース同士に移動コスト, 移動不可なブース同士に-1 を格納する.

前章から得られた現在位置と目的位置のノード ID を入力する.

現在位置から目的位置までの距離を一筆書き可能な経路の全ての総移動コストを算出し, それぞれのルートと比較し移動コストの少ないものを格納する. 最も移動コストの少ないルートが決定すると経路上の店舗を順に出力する.

これらの処理を図 2 に示す.

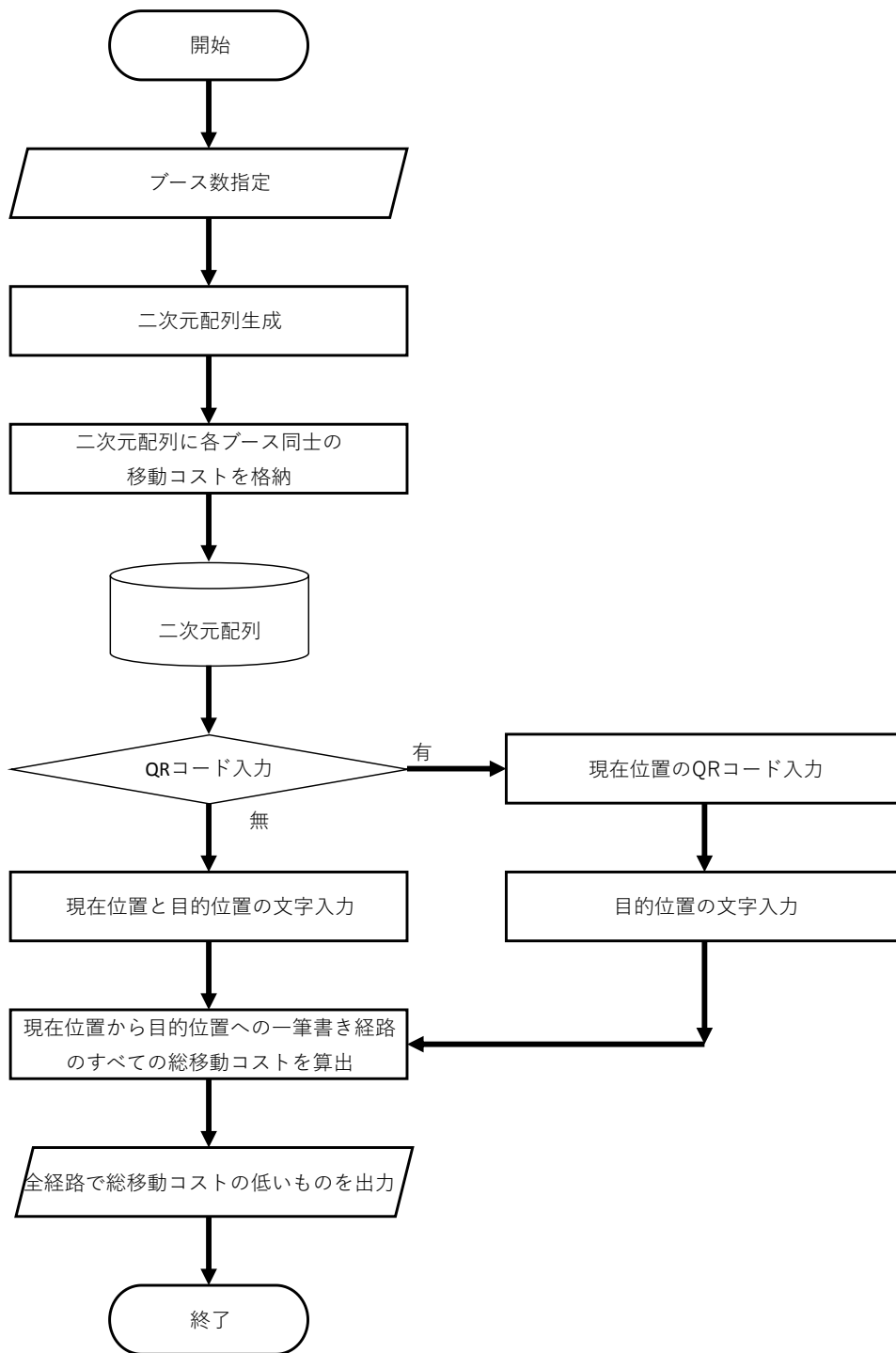


図2 道案内のフローチャート

第4章 実験

4.1 実験の対象

提案手法の有効性を調べるために以下の4つを実験で確認する。

1. 現在位置と目的位置の入力部分を作成し、店舗名からノードID、位置座標まで正しく得られるかを確認する。
2. QRコードを用いたノードID取得が正しく、かつ手軽であるかを確認する。
3. 経路探索アルゴリズム「ベルマンフォード法」のプログラムを作成し、実用的に動作することを確認する。
4. 実際に出力された経路を自分の足でなぞり、利便性を確認する。

実験1では、実際に現在位置と目的位置の文字入力を行い、正確な位置座標が得られるか確認することで、文字入力からシステム上の現在位置推定を可能とする。

実験2では、実験1での文字を入力する手間を無くし、QRコード入力を用いて現在位置周辺のノードIDが正確に出力されるか、手軽なUIであるか、確認できる。

実験3では、本実験の位置推定が実用的であるかを実証するために、位置情報を用いたシステムとして利用率の高い、道案内システムを作成する。

実験4では、実験3で出力された経路を実際に自分の足で確認することで最適かどうかについて検討する。

4.2 実験環境

イオンモール京都五条店で実験を行った。

他の施設でもすぐ使えるように店舗、施設、地図の編集をすぐに行える簡単なものにした(図3)。

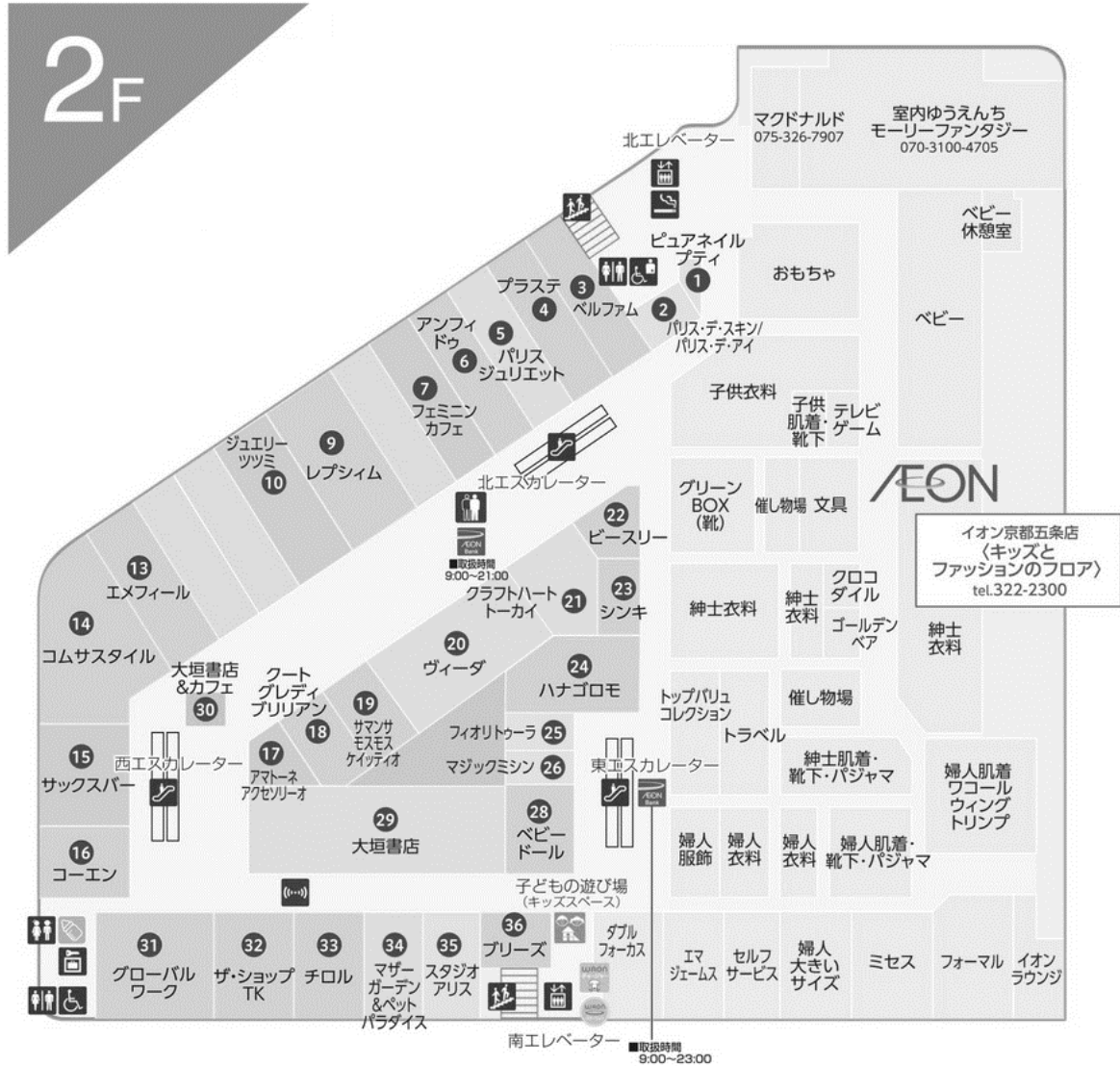


図3 イオンモール京都五条店 2F の案内図

第5章 実験および考察

5.1 実験 1:位置推定

文字入力部分はブース名を受け付ける. ブース名はデータベースを通してブース ID に変換され, ブース ID の座標情報から位置推定を行う. ブースの属性情報の例を以下に示す.

店舗番号 3 番:ベルファム

ノード ID:04, 座標:(495,295)

店舗番号:15 番:サックスバー

ノード ID:13, 座標:(105,610)

店舗番号:29 番:大垣書店

ノード ID:24, 座標(225,665)

5.2 実験 2:QR コードの作成と読み取り

QR コードの作成は CMAN インターネットサービスを利用した. (図 1)

QR コードの読み取りは, 手持ちの iPhone6s を用い以下の「QR コード」検索上位アプリで認識実験を行った.

LINE

(<https://itunes.apple.com/jp/app/line/id443904275?mt=8>)

読み取り結果:「読み取れない形式の QR コードです」

QR コードリーダー , バーコードリーダー-アイコン

(<https://itunes.apple.com/jp/app/qr%E3%82%B3%E3%83%BC%E3%83%89%E3%83%AA%E3%83%BC%E3%83%80%E3%83%BC-%E3%83%90%E3%83%BC%E3%82%B3%E3%83%BC%E3%83%89%E3%83%AA%E3%83%BC%E3%83%80%E3%83%BC-%E3%82%A2%E3%82%A4%E3%82%B3%E3%83%8B%E3%83%83%E3%83%88/id480090210?mt=8>)

読み取り結果:「24」

QR コードリーダー for iPhone

(<https://itunes.apple.com/jp/app/qr%E3%82%B3%E3%83%BC%E3%83%89%E3%83%AA%E3%83%BC%E3%83%80%E3%83%BC-for-iphone/id585561686?mt=8>)

読み取り結果:「24」

結果の一例を図 4 に示す.

図 4 を見ると, iPhone 純正カメラによって読み取った QR コードが変換され, 読み取り結果:「24」を示していることがわかる.



図 4 iPhone 純正カメラで QR コードを認識

5.3 実験 3:経路探索

二次元配列での移動コストの調整実験を以下に示す。

隣り合うブース同士を距離に応じて移動コストを設定して結ぶ。

「隣り合う」基準は別のブースを通らず移動可能で、互いのブース同士がハッキリ見えることである。

移動コストは、実際の距離から算出する。つながれたブース同士を実際に歩き、経路ごとに手動入力する。そのあとにいくつか最適経路出力を表示し、明らかに遠回りの経路の移動コストを手動調整する。以下に 2F 専門店側での実験の調整作業結果を示す。

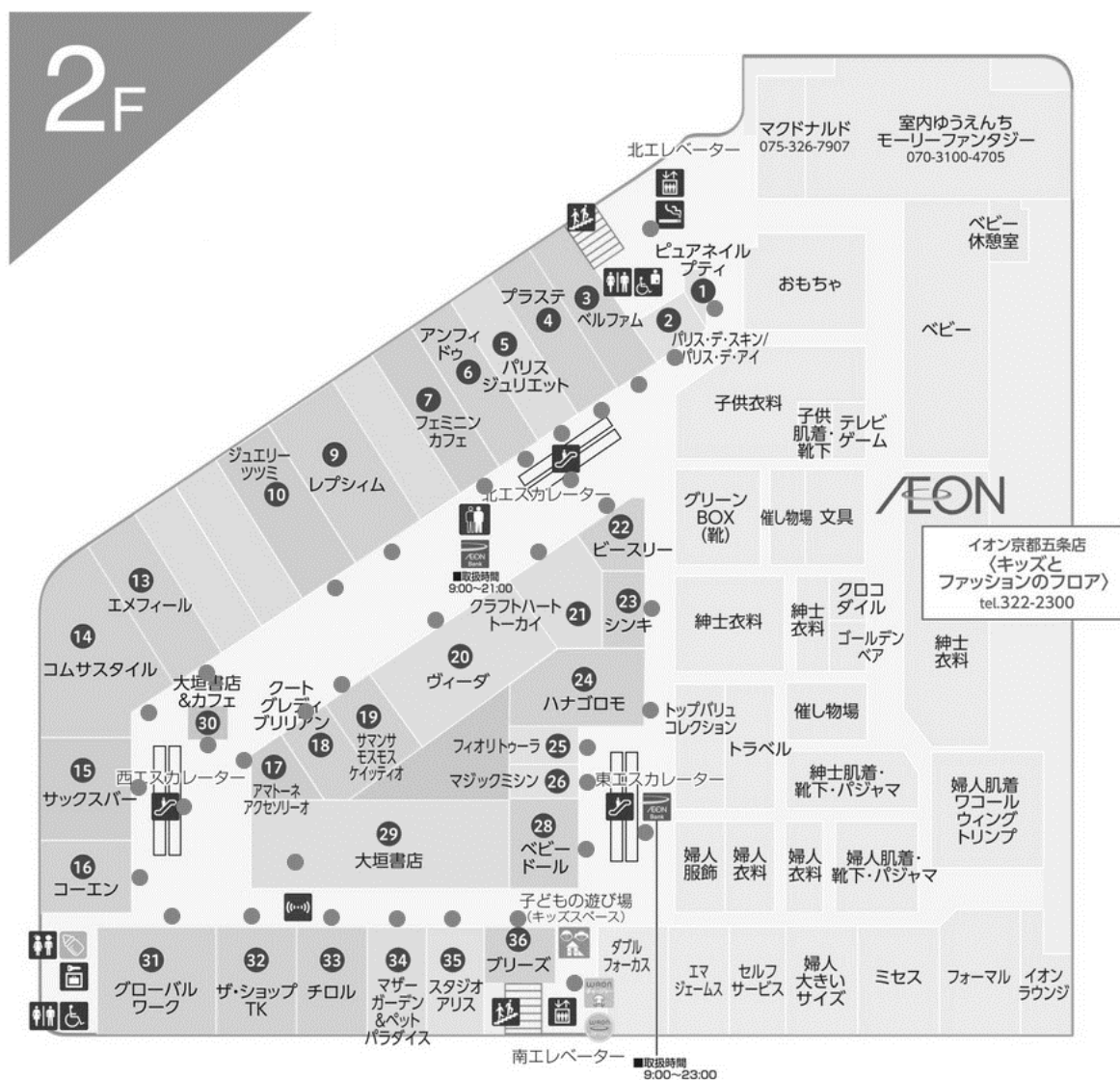


図 5 店舗座標の登録

今回は北エレベーターから西端のサックスバー(予め番号が振ってあるので以下 15 と省略する)へ向かう。

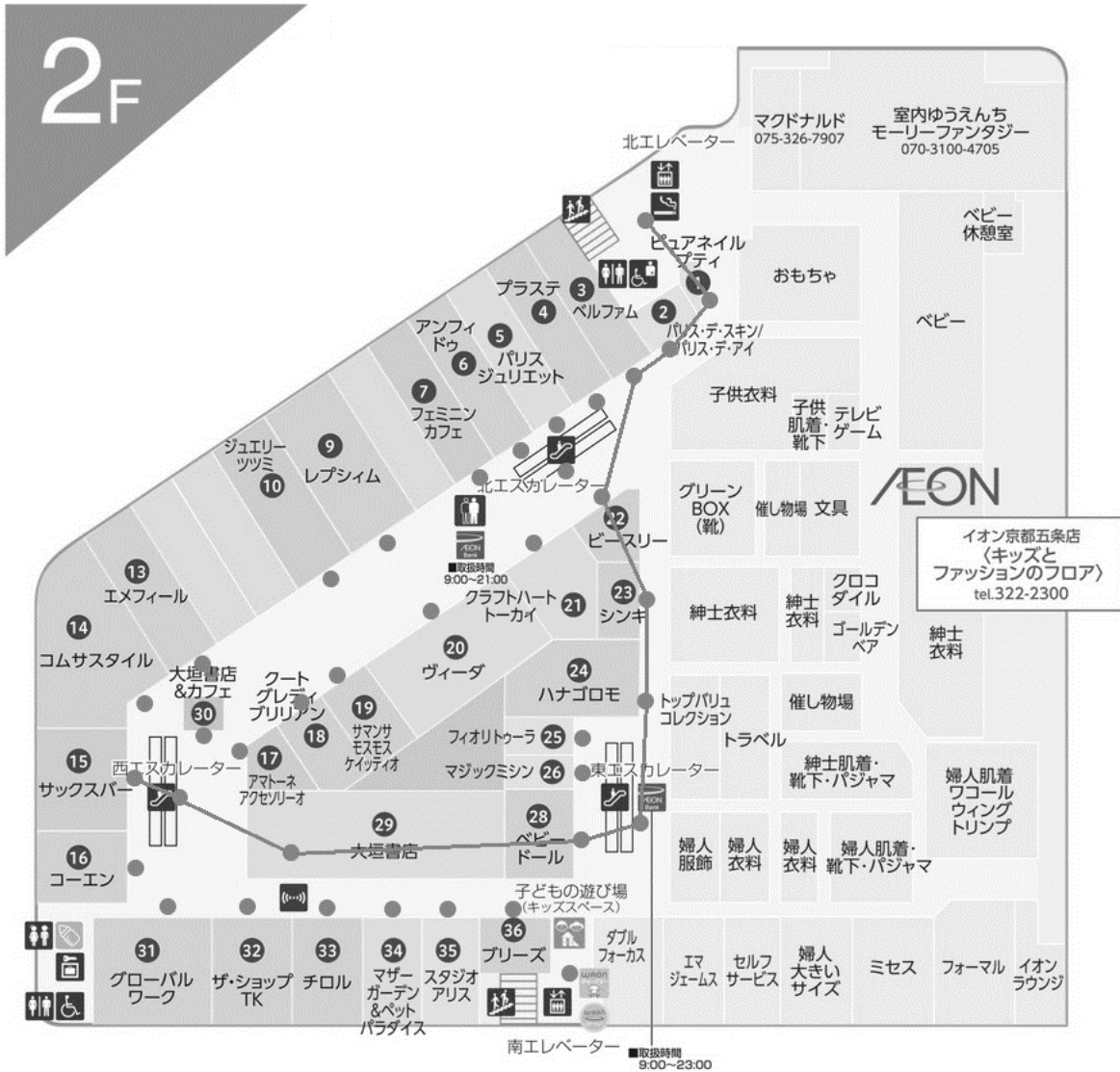


図6 経路出力1

この時点での出力は
北 EV-1-2-3-22-23-24-東 ES-28-29-西 ES-15
と南側を迂回するルートをとる。

29番(大垣書店)の店舗面積が広いいためこのような結果になった、これは明らかな遠回りのため29番への移動コストを増やす。

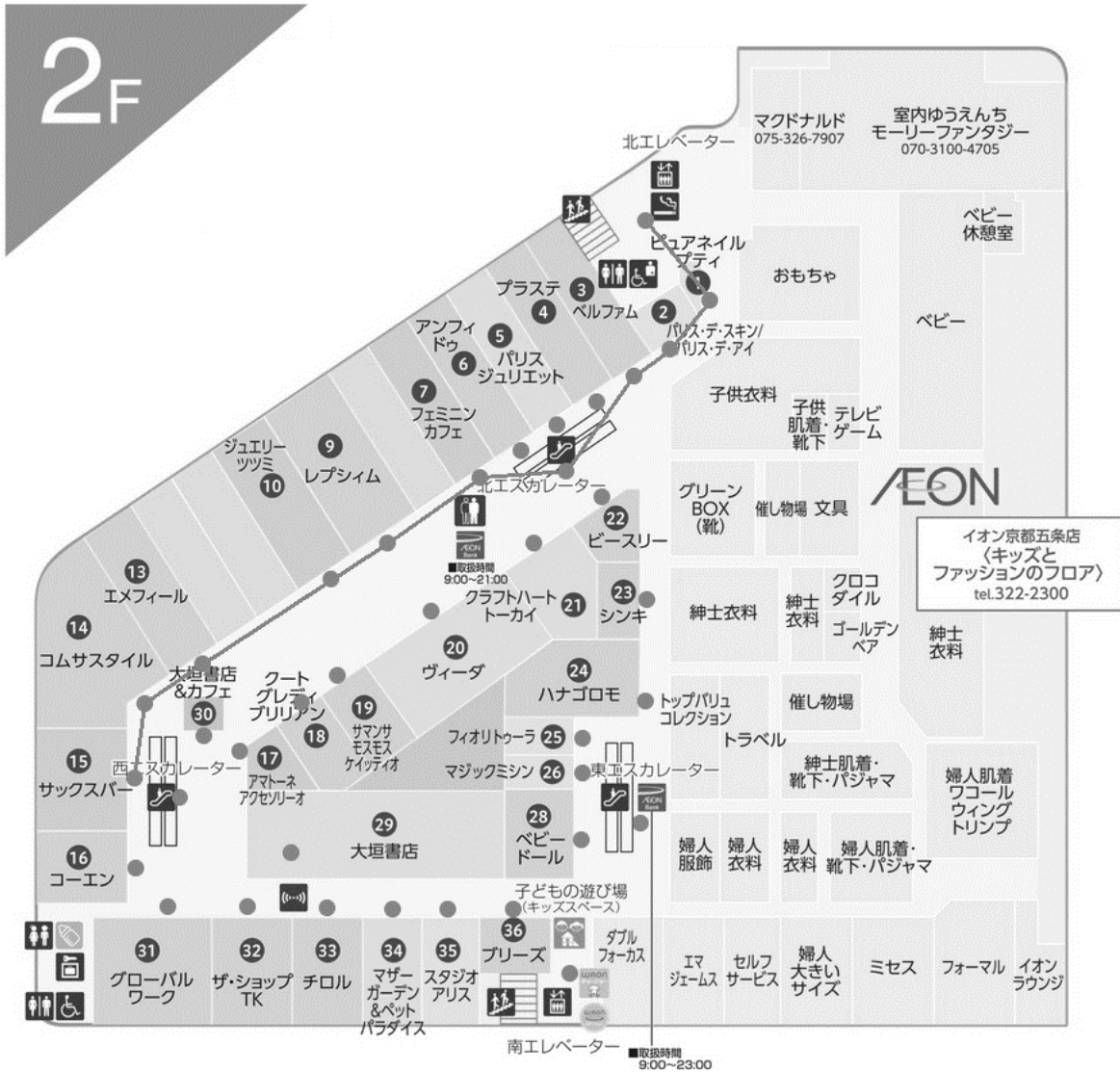


図 8 経路出力 3

3 番～22 番の通路向かい側同士の移動コストを増やした結果は以下ようになった。
 北 EV-1-2-3-北 ES-7-9-10-13-14-15
 地図で見る限りの最適ルートを示してくれた。
 このような調整を施設全体に施す。

5.4 実験 4:利便性の確認

2018年11月21日イオンモール京都五条店で、3回経路を確認したところ、 unnecessary 遠回りなどはなく、目的地に到着できた。

5.5 考察

実験の結果を以下に述べる。

全ての店名からノード ID への変換が確認できた。

「QRコード作成」検索上位サイトの全てにおいて QRコードの作成が簡単であることを確認した。

QRコード読み取り用アプリではいずれも正しく素早くノード ID を得られることが出来た。

実験 3 の結果は、前項で示したとおり。

実際に店舗にて徒歩実験を行ったところ数十秒単位で最終出力したルートが他のルートより早くたどり着くことが確かめられた。

以上により、提案手法が実現可能であることが示された。

第6章 おわりに

今回の実験により実際にある施設内で現在位置から目的地まで誘導するという目的は達成された。ここから、施設内通路が広がったり目の前に何も目印ない状況があったりする施設など、本プログラムの単純な編集では対応が難しい例について考えていきたい。ただし、屋内で位置情報を無線で取得することが難しい課題は解決することができた。

また利便性等において今後の課題として以下がある。

- ・現在位置の入力は端末付属カメラで周辺店舗の撮影から位置を分析
- ・目的地の入力は地図を表示してその中からタップ等で入力する選択が求められる。

謝辞

本研究課題を与えていただき、終始ご指導、ご鞭撻を賜りました三好 力 教授に深く感謝の意を表します。

参考文献

[1]基本情報技術者試験ドットコム

https://www.fe-siken.com/kakomon/29_haru/pm08.html

(2018年11月15日アクセス)

[2]連想類語辞典 日本語シソーラス

<https://renso-ruigo.com/>

(2018年12月20日アクセス)

[3]アルゴリズム講習会

<http://dai1741.github.io/maximum-algo-2012/docs/shortest-path/>

(2018年11月15日アクセス)

[4]CMAN インターネットサービス

<https://www.cman.jp/QRcode/>

(2018年12月20日アクセス)

[5]LINE

[https://itunes.apple.com/jp/app/line/id](https://itunes.apple.com/jp/app/line/id443904275?mt=8)

[443904275?mt=8](https://itunes.apple.com/jp/app/line/id443904275?mt=8)

(2018年12月22日アクセス)

[6]QRコードリーダー , バーコードリーダー-アイコンット

<https://itunes.apple.com/jp/app/qr%E3%82%B3%E3%83%BC%E3%83%89%E3%83%AA%E3%83%BC%E3%83%80%E3%83%BC-%E3%83%90%E3%83%BC%E3%82%B3%E3%83%BC%E3%83%89%E3%83%AA%E3%83%BC%E3%83%80%E3%83%BC-%E3%82%A2%E3%82%A4%E3%82%B3%E3%83%8B%E3%83%83%E3%83%88/id480090210?mt=8>

(2018年12月22日アクセス)

[7]QRコードリーダー for iPhone

<https://itunes.apple.com/jp/app/qr%E3%82%B3%E3%83%BC%E3%83%89%E3%83%AA%E3%83%BC%E3%83%80%E3%83%BC-for-iphone/id585561686?mt=8>

(2018年12月22日アクセス)

付録

・ソースコード(一部)

```
int nPoint=N;
int sRoute[nPoint];
int sDist;

int pDist[nPoint];
int pRoute[nPoint];
bool pFixed[nPoint];
int sPoint,i,j,newDist;

sDist=99999;

for(i=0;i<nPoint;i++){
    sRoute[i]=-1;
    pDist[i]=99999;
    pFixed[i]=false;
}

pDist[sp]=0;

while(true){
    i=0;
    while(i<nPoint){
        if(pFixed[i]==0){
            break;
        }
        i=i+1;
    }

    if(i==nPoint){
        break;
    }
    for(j=i+1;j<nPoint;j++){
        if(pFixed[j]==0 && (pDist[j] < pDist[i])){
            i=j;
        }
    }
    sPoint=i;
    pFixed[sPoint]=true;

    for(j=0;j<nPoint;j++){
        if((Distance[sPoint][j]>0) && (pFixed[j]==0)){
            newDist=pDist[sPoint]+Distance[sPoint][j];
            if(newDist<pDist[j]){
                pDist[j]=newDist;
                pRoute[j]=sPoint;
            }
        }
    }
}

sDist=pDist[dp];
j=0;
i=dp;

while(i!=sp){
    sRoute[j]=i;
    i=pRoute[i];
    j=j+1;
}
sRoute[j]=sp;
```