

中間点を用いた案内粒度変更可能な音声経路案内システム

浮田 弥¹ 山本 大介¹ 高橋 直久¹

概要：本研究では，経路案内を音声のみで行う場合に歩行者にとって分かりやすい案内手法を提案するために，Android 端末用の音声案内アプリの開発を行った．既存の音声経路案内システムではユーザを経路通りに進ませるよう音声案内を行うが，その手法は GPS の誤差に影響されやすく，またユーザ自身の直感に合っていない道を進ませる場合があるのではないかと考えた．本研究はこの問題の解決のため，経路案内における案内粒度が変更可能なシステムを提案する．提案システムは，スマートフォン端末上で音声と地図による案内を行う際，既存経路に小目的地を複数配置することにより，案内粒度を変更可能とする機能や，ユーザの進むべき方向と距離などの情報を音声で伝える機能を持つ．この音声経路案内システムの実現のため，手軽に会話が作れる音声インタラクションシステム構築ツールキット”MMDAgent”[1] の音声を利用した．さらに，作成したプロトタイプシステムに対する評価を行い，提案手法の有用性を評価した．

Navigation granularity variable voice route navigation system using midpoints

WATARU UKITA¹ DAISUKE YAMAMOTO¹ NAOHISA TAKAHASHI¹

1. はじめに

近年，モバイル端末における道案内システムが普及している GoogleMaps の音声案内 [2] や Android の標準マップ [3] などの既存のスマートフォン向けの地図アプリにおける音声案内システムでは，細かく決められた経路に沿って右左折を音声で伝える形で案内を行うため，ユーザは自由に道を選択することができない [4]．また，経路辺が短く入り組んだ道では GPS の誤差の影響を受けやすくなり，かえって案内が分かり辛くなる場合がある．一方で「目的地への方向と距離のみを案内する」という，前述のような煩わしさを解消した道案内アプリも存在する [5]．しかし，示された方向に道が無い場合があるなどの問題がある．本研究では，この 2 種のシステムの利点を活かした案内手法を提案する．

ここで既存の音声経路案内システムの問題点をまとめると次のようになる．

問題点 1 従来の経路案内システムでは，経路探索結果に従って交差点毎に曲がるべき箇所を逐次ユーザに提示していく方式が一般的である．しかし，システムの提示する経路は往々にしてユーザの直感に合った経路と一致しない場合がある．

問題点 2 従来の経路案内システムでは，GPS の精度に大きく依存しており，特に都心部などの道路密集地域においては GPS の誤差により曲がるべき道を間違えてしまうことがある．

問題点 3 目的地への方向と距離のみを案内する経路案内システムでは，上記の問題の一部を解決可能であるが，示された方向に道があるとは限らない場合がある．

本研究では GoogleMaps などから得られる案内経路上に「中間点」という小目的地を複数配置することでこれらの問題点の解決を試みた．「中間点」とは GoogleMaps などから得られる案内経路上の頂点の一部のことであり，1 つの案内経路上には複数の中間点を持ちうる．システムはこれらの中間点までの案内を繰り返すことでユーザを目的地に案内する．また，「案内粒度」とは案内の細かさのことで目的地までの中間点の数が多いほど案内粒度が細かく，案

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

案内粒度が細かい
中間点が多い

案内粒度が粗い
中間点が少ない

2. 提案システムの概要

要件１ スマートフォン端末上で、音声と地図による案内を行う。システムの基本的なユーザーインターフェースについては既存システムに倣い、地図や現在位置、経路表示など最低限の GUI を備えつつ、それらを見なくとも音声のみで案内が可能なものとする。

要件 3 ユーザの進むべき方向と距離などの情報を音声で伝える。先に紹介した「目的地の方向と距離のみを案内する」という方法を一部採用し、特徴 2 で得られた中間点への方向と距離のみによる案内を行う方法を取った。

の中間点を順に辿っていき目的地までの案内を行う。

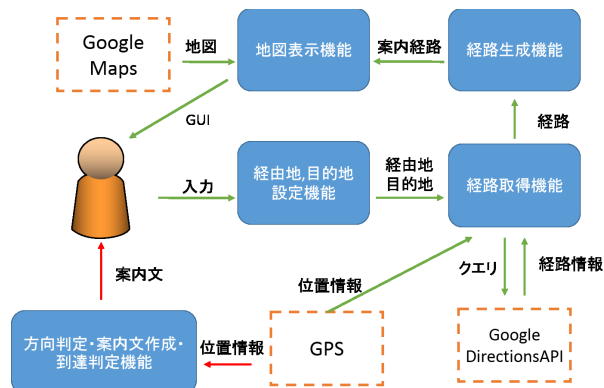


図 2 システムの構成図

3. 提案システムの実現法

本章では提案システムの実現法について述べる。

3.1 地図描画、目的地、経路地設定機能の実現方法

描画する地図は、既存のシステムを利用する。Android 端末で扱えるものは GoogleMaps[6] や OpenStreetMap[7] などが代表として挙げられるが、今回のプロトタイプシステムでは、簡易に表示が可能な GoogleMaps を採用している。今回のシステムでは経路情報の取得のために目的地と経由地の座標の取得が必要となる。ユーザは目的地、経由地に設定したい地点を画面の中央に表示させ、ボタンを押すとその地点が目的地、経由地として設定される。このとき指定された地点の座標を取得し保存する。

3.2 経路情報取得機能の実現方法

経路情報の取得については、ウェブサービス「Google Directions API[8]」を利用して取得する。これは次に示すソースコードの例のように始点、終点などの必要パラメータを含めた URL をクエリとして送ると、経路情報が XML 形式や JSON 形式で得られるものである。システムは URL に現在地の座標と前節で保存した目的地、経由地の座標を含めて送ることによって経路情報を取得する。

```
http://maps.googleapis.com/maps/api/directions/
xml?origin="現在地の座標"&destination="目的地の座
標"&mode=walking"
//経由地あり

"https://maps.googleapis.com/maps/api/directions/
xml?origin="現在地の座標"&destination="目的地の座
標"&waypoints="経由地の座標"&mode=walking"
```

— 429 —

グで構成されているが図3のように,DOM はそれを木構造として扱うことができる. 具体的には,XML では各経路辺の情報は複数ある<step>タグ以下に1 つずつ配置されており, その子孫ノードに経路辺の始点と終点の経度, 緯度(<start_location>,<end_location>下の<lat>,<lng>) が格納されているので, それを取り出す.

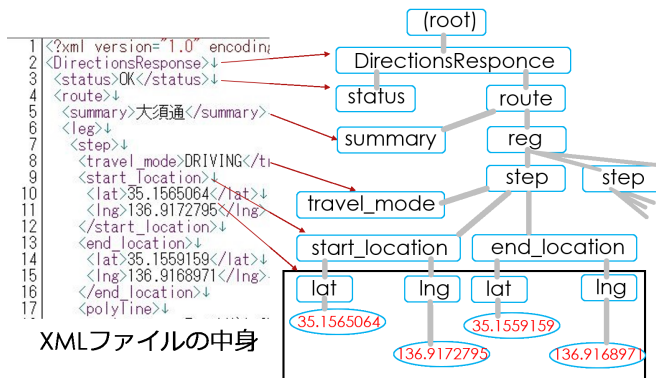


図3 XML の構造

3.3 経路生成機能の実現

提案手法の経路は, 先の章で述べた通り, 既存の案内経路を再構築する形で行う. 前節で得られた案内経路上に, 「中間点」という小目的地を複数配置し, 経路案内をすべて中間点に向けて行うという手法をとった. 中間点は, 出発地点を第0中間点とし, 以降は経路の線分の各頂点から選択し, 目的地を最終中間点とする. また, 中間点の生成は経路の長さによるアルゴリズムと面積によるアルゴリズムのどちらかで行う.

3.3.1 経路辺の長さによる中間点の生成

1 つ目の中間点の配置方法は, 経路辺の長さの合計がある一定の値を超えた時に中間点を配置していくという方法をとる. さらに, 経験的な手法から角度での条件を付け加えたアルゴリズムで中間点を生成する. 実際には, ユーザにより決定される基準長を B とする. 目的地までの経路辺を順に $e_1, e_2, \dots, e_n$ とし, 辺 e_i の長さを $len[i]$ とする. また, 初期値が0である変数 x を用意し, 以下の手順に従い中間点リスト C を決定する.

ルール1 経路辺 e_1 の始点を中間点のリスト C に加える

ルール2 経路辺 e_1 から e_n に対し, 先頭から順にそれぞれ以下の処理を行う.

- (a) $len[i]$ が B より長いとき
 - (i) e_i と e_{i+1} のなす角度が 135° 以上 (ほとんど) 道なりである場合, 中間点は生成しない
 - (ii) そうでない場合, e_i の終端を C に加え, x を0にする.
- (b) $len[i]$ が B 以下のとき
 - (i) e_i と e_{i+1} のなす角が 45° 以下 (曲がる方向が後方) である場合, e_i の終端を C に加え, x を0にする.
 - (ii) そうでない場合, x に $len[i]$ を加算する.
 - (iii) x が B を超えた時, e_i の終端を C に加え, x を0にする.
- (c) $i=n$ の場合, e_i の終端を C に加えてすべての処理を終了する.

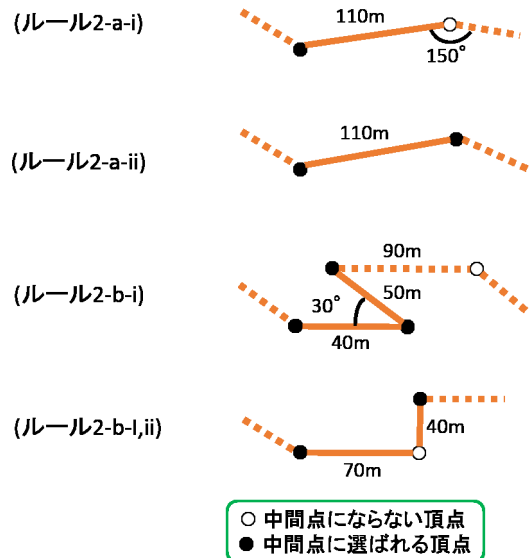


図4 経路辺の長さによる中間点生成ルール

3.3.2 面積による中間点の生成

2 つ目の中間点の配置方法は, 直線化のアルゴリズムである L^2 error norm weight アルゴリズムに条件を付け加えたアルゴリズムにしたがって行う. 出発地点を p_0 , 目的地までの頂点を $p_1, p_2, \dots, p_n$ とする. また初期値が0である変数 $area$ と変数 x を用意し, 以下の手順を繰り返して指定された頂点の数になるまで頂点を1 つずつ減らしていく. 残ったものを中間点とする.

ルール 1 頂点 $p_k (1 \leq k \leq n-1)$ に対してそれぞれ以下の処理を行う.

- (a) $\angle p_{k-1} p_k p_{k+1} > 45^\circ$ かつ $\Delta p_{k-1} p_k p_{k+1} < area$ のとき
 もしくは
 $\angle p_{k-1} p_k p_{k+1} > 45^\circ$ かつ $area = 0$ のとき
 (i) $area = \Delta p_{k-1} p_k p_{k+1}$ にする
 (ii) $x = k$ にする

ルール 2 ルール 1 の処理が終わると x には一番面積の小さい三角形を形成する中央の頂点の番号が入っているので p_x を除いた頂点で経路を生成し, $x = 0$ $area = 0$ にする

つまり, 各頂点と隣り合う前後の頂点によってできる三角形の面積を計算していき, 角度による例外を除いた一番小さい面積を形成する中央の頂点を除外することによって 1 つずつ頂点を減らしていく. $n=5$ のときの経路生成の様子を図 5 に示す.

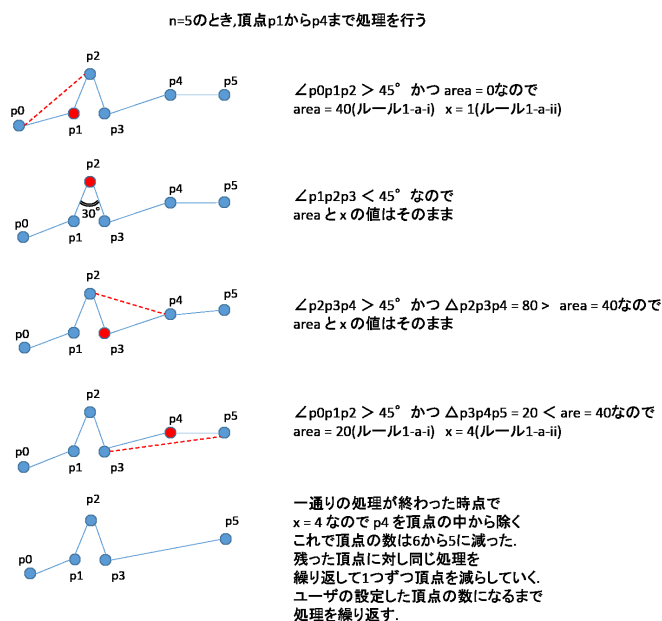


図 5 面積による中間点生成の例

3.4 中間点や目的地への音声案内と到達判定の実現方法

現在位置の取得については, 端末の GPS 機能を利用した. 今回は右や左などのユーザ視点での方向案内が必要であるため, ユーザの歩行軌跡からユーザの向いている方向を判定し, 中間点や目的地への方向判定を行う. 図 6 のようにユーザが歩いている状況で現在位置, 数秒前の位置, 中間点 (目的地) の 3 点が成す角度からユーザから見た中間

点 (目的地の) の方向を判定可能である [10].

今回作成したプロトタイプシステムでは, 図 7 のように時計盤の 0 時をユーザから見て正面とし, 0 時から 11 時の 12 通り 30 度刻みで方向を示す.

音声案内については先の章で述べたように, MMDAgent を利用する. そのために, 今回のシステムは MMDAgent からプラグインとして呼び出す方式を取っている. ユーザが端末のアプリである MMDAgent を起動後, 提案システムを呼び出すことで提案システムの出力としてだけでなく, ユーザの入力インターフェースとしても音声を利用できるようにする. 自動案内については MMDAgent へ案内文をクエリとして渡すことで, 音声案内を定期的呼び出す方式を取った. 案内文の内容は「(第 n 中間点 or 目的地) は, (0 時から 11 のいずれか) 方向に (距離) m です」とした.

中間点や目的地への到達判定は, 以下の手順で判定している.

- 手順 1 現在地と巡回済みでない中間点との距離をすべて求める
 手順 2 近傍に中間点があれば, それ以前の中間点を巡回済みとする
 手順 3 その次の中間点を目標とし, それより後の中間点を未巡回とする
 手順 4 複数の中間点が近傍にある場合は, 一番目的地に近い中間点を選択される

これにより, ユーザが途中の中間点を飛ばして次の中間点に到着してしまった場合も, 以前の中間点に引き返す案内をすることを防いでいる. また現在地と中間点または目的地の距離が 15m 以内になったら到着したと判定する.

方向案内については, ユーザに定期的に伝える必要があるので, プロトタイプシステムでは TimerTask 関数を利用し, 以下の機能を 20 秒感覚で呼び出している.

- 1 ユーザが進むべき方向の判定
- 2 中間点や目的地への距離と到達判定
- 3 上記を伝える音声案内

なお, このタイマーはアプリを終了させるか, 目的地に到達すると停止する.

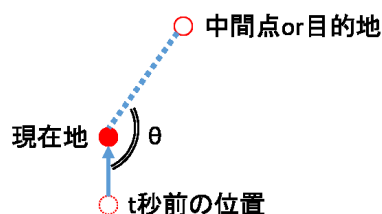


図 6 方向判定の概要

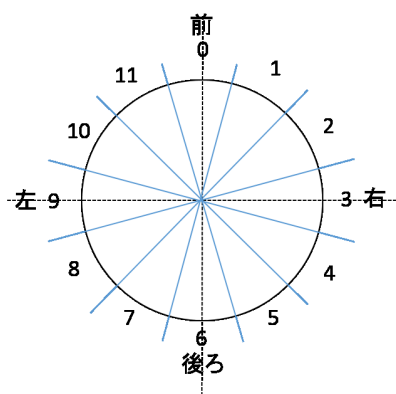


図 7 方向に対する案内

4. プロトタイプシステム

4.1 プロトタイプシステムの開発

プロトタイプシステムの開発は、Windows7 のもとで Eclipse を用いて Java によって行った。完成したシステムは nexus5 端末に送り、アプリとして起動させている。

4.2 プロトタイプシステムの構成

今回作成したプロトタイプシステムの構成図は先に示した図 1 と同じになっている。入力としては経由地、目的地の設定と案内粒度の指定が必要である。出力としては、GoogleMaps の地図を中心とする GUI、その上での経路表示、経路案内システムが MMDAgent を通じてユーザーに行う音声案内となる。

4.3 目的地、経由地の設定

ユーザーは経由地または目的地として設定したい場所を画面の中心に表示させる。そこで、経由地または目的地のボタンを押すとその地点が経由地または目的地として設定される。この際、設定された地点の緯度経度座標を変数に格納しておく。

4.4 経路情報の取得と経路生成

目的地などが設定されたあと経路検索ボタンを押すと変数に格納していた目的地などの座標と現在位置の座標から GoogleDirectionsAPI のクエリを作成し、経路情報を取得する。取得した経路情報から頂点の経度緯度座標の情報を取り出し、端末画面上に得られた経路を表示する。ユーザーは案内粒度を指定し、経路生成ボタンを押すと、すると、案内する経路と目的地までの案内文が表示される。目的地までの案内文はボタンになっており、このボタンを押すと音声案内が開始される (図 8,9)。

4.5 音声案内

音声案内が開始されると約 7 秒おきに現在位置を取得す

る。案内文は約 21 秒おきに MMDAgent を通じてユーザーに伝えられる。7 秒前の位置と現在位置、目的地または中間点の座標から方向判定機能でユーザー視点からの目的地または中間点の方向を計算する。このとき、現在位置から目的地または中間点への直線距離を計算する。それらを使って案内文を作成する。実際には「(第 n 中間点 or 目的地) は (0~11 時の方向に), x m です。」というような案内文が作成される。

また、中間点の 30m 以内に近づいた場合は 7 秒前の位置、現在位置、次の中間点の座標から方向を計算し「(第 n 中間点 or 目的地) は (0~11 時の方向に), x m です。その後は (0~11 時の方向) です。」というような案内文を作成し、第 n 中間点に到着した後の第 $n+1$ 中間点の方向をユーザーに伝える。そして、第 n 中間点に 15m 以内に近づいた場合は「第 n 中間点に到着しました。次は (0~11 時の方向) です。」というような案内文を作成する。これらの案内によって中間点に着いた際、ユーザーが立ち止まらずに次の中間点に迎えるようになっている。目的地の 15m 以内に近づいた場合は、「目的地周辺です。お疲れ様でした。」とユーザーに伝え案内を終了する。

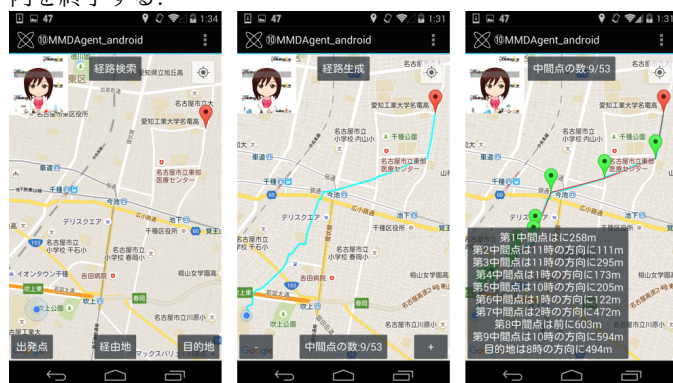


図 8 経路情報取得前 (左), 経路取得 (中央), 案内経路表示 (右)

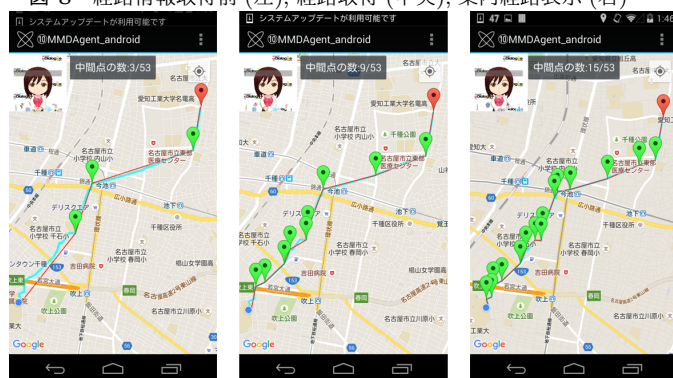


図 9 同じ経路で案内粒度を変えた場合

5. 提案システムの評価

プロトタイプシステムの評価としては2つの実験を行った。

5.1 実験1

5.1.1 評価実験の目的と内容

今回の実験の目的は、提案した中間点を用いたシステムと経路通りに案内する既存の手法、目的地の方向と距離のみを案内するシステムを比較することで、案内粒度によるユーザにとっての分かりやすさや目的地への辿りつきやすさなどの違いを調査することにある。

実験方法については、まず設定を案内粒度の細かい順に「A.GoogleDirectionsAPI から得られた経路データ通りに案内」「B. 中間点を生成する今回の提案手法で案内」「C. 目的地の距離と方角のみを案内」の3通りに分けた。Aの経路は出発点から目的地まですべての経路を指定して案内するので案内粒度は最も細かい。Cの経路は目的地の距離と方向だけを案内するので案内粒度は最も粗い。Bの経路は出発点と目的地の間にいくつかの中間点を設定し、それらを辿るように案内を行う。案内粒度はAとCの間といえる。比較は全て同じ端末で行い、それぞれの設定につき各3名、合計9名を被験者とし、9名全てに予め取得した、同じ出発点、目的地の経路を進んで貰った。具体的な実験の手順は、以下の通りである。

- (1) 被験者に全体的な地図と目的地、案内経路を見せる
- (2) Android 端末を持たせ、出発点から端末の音声のみに従って歩いてもらう。
- (3) 音声案内は一定間隔で、中間点または目的地の方向と距離が伝えられる。
- (4) 目的地に着いた時点で実験を終了する。

実験終了後は以下に示す10項目の質問に回答してもらった。項目1から7は5段階評価で、「最も当てはまる」を5としており、項目8から10は自由記述となっている。

項目1 案内がわかりやすいと感じたか (5段階)

項目2 自分の直感にあった道を進めていると感じたか (5段階)

項目3 案内が煩雑に感じたか (5段階)

項目4 迷わず進めていると感じたか (5段階)

項目5 システムが自分の位置を正確に把握していると感

じたか (5段階)

項目6 システムを使っでの満足度 (5段階)

項目7 案内時に不安を感じたか (5段階)

項目8 項目7で具体的に不安を感じた点はどこか (自由記述)

項目9 今まで使用した類似システムと比べてどう感じたか (自由記述)

項目10 その他システムに対する意見 (自由記述)

5.1.2 結果と考察

今回の実験で得られたデータを表1に示す。それぞれの項目について、自由記述の意見も交え考察する。

項目1:案内がわかりやすいと感じたか

Aが他の2つより低い結果となったのはGPSの誤差の影響を受け曲がり角などが分かりにくかったのではないかと考えられる。また、「〇時の方向」という案内だと直感的に方向が分からず一瞬戸惑ってしまうという意見があった。ユーザが直感的に中間点の方向が分かるような伝え方が必要であると考えられる。

項目2:直感にあった道を進めていると感じたか

案内粒度を粗くすることによって「正確に道が決まっていなくて自由度が高く、しばらく感じ方がよかった」という意見があった。案内粒度を粗くすることによってユーザからは「直感に合った道を進めている」という評価が得られる。またC>Bとはならなかったため、単純に案内粒度を荒くするほど自由度が上がるというわけではなく、ある程度の方向を示した方がよいのではないかと考えられる。

項目3:案内が煩雑に感じた

ユーザによって「頻繁に案内してくれるので安心できた」という意見と「案内の間隔が短い」という意見があったので案内の間隔をユーザが指定できるようにする必要があると考えられる。

項目4:迷わず進めていると感じたか

この結果から案内粒度を粗くしてもユーザは迷わずに進めていると感じ、ある程度安心感を持って進めると考えられる。

項目5:システムが現在地を正確に把握していると感じたか

この項目の結果、Cが他の2つに比べ高い値を示した。これはパターンCにおける案内では、A,Bに比べ「次の目的地までの距離」が長くなるからであり、その分GPSによる誤差を吸収しやすいためと考えられる。

項目6:システムを使っでの満足度

どれも4前後の評価が得られており、高い順にC,B,Aとなった。この理由について、「9. 今まで使用した類似

表 1 実験結果

実験経路	A. 経路データ通り	B. 提案手法	C. 目的地のみ
評価項目	評価 (5 段階)		
平均到着時間 (分:秒)	8:42	6:04	6:39
案内が分かりやすい	3.33	4.66	4.00
直感にあった道を進めた	3.33	5.00	4.66
案内が煩雑に感じた	2.66	2.00	2.00
迷わず進めていると感じた	2.66	3.66	2.66
システムが位置を正確に把握している	3.00	3.00	4.33
システムを使っでの満足度	3.66	4.00	4.33
案内時に不安を感じた	3.33	2.33	2.66

システムと比べてどう感じたか」で得られた意見によると C では「探検的な楽しさがある」、B では「選ぶ楽しさがあった」「気楽」との解答があったため、それが影響しているものと考えられる、パターン A では平均到着時間からも分かるように迷うことが多くなったため、他の 2 つに比べるとやや低い値がでたと考えられる。

項目 7:案内時に不安を感じたか

ユーザの方向判定の精度が低く不安に感じてしまうという意見が多かったので方位センサーなどにより正確に方向判定を行う必要があると感じた。それによりユーザが立ち止まってしまった場合にも正しく方向判定を行うことができると考えられる。また、今回の実験では目的地までの案内は音声だけで行った。そのため、地図を見ることができず不安に感じたという意見が多かった。しかし、経路通りの案内をするシステムよりは地図を確認する回数は減るのではないかと考えられる。

6. 実験 2

6.1 評価実験の目的と内容

もう 1 つの実験の目的は経路生成アルゴリズムに対する実験である。経路辺の長さによる経路生成アルゴリズムと面積による経路生成アルゴリズム、それぞれに対して角度の条件がある場合と無い場合の 4 つの経路生成によって作成された経路について比較、評価を行った。角度の条件とは経路辺の長さによる経路生成アルゴリズムの (ルール 2-a-i)、(ルール 2-b-i) のことである。

出発点から目的地までの経路から案内に必要と思われる中間点を指定する。後ろに大きく曲がるような点や障害物を避けるために通らなければいけない点など重要度が高いと思われる中間点ほど点が高くなるように合計 10 点になるように中間点に点数を配分する。それぞれのアルゴリズムによって設定された中間点とどのくらい一致するのかを評価実験として行った。

6.2 結果と考察

得られたデータを表 2 に示す。



図 10 比較した経路の 1 つ

実験から経路辺の長さによる経路生成よりも面積による経路生成の方が一致数が多く、得点の高い中間点の配置が行われ、角度の条件もありの方が良い結果が得られた。経路辺の長さによる経路生成で角度の条件がない場合、曲がり角や直線上の点に関係なく中間点が生成されてしまうためユーザにとって分かりやすい中間点とは言えず得点も低い値となった。経路辺短い道が連続した場合、経路辺の長

表 2 実験結果

経路 (頂点数)	面積 (角度有)		面積 (角度無)		経路辺の長さ (角度有)		経路辺の長さ (角度無)	
	一致数	得点	一致数	得点	一致数	得点	一致数	得点
経路 1(4)	4	10	4	10	3	8	1	3
経路 2(3)	3	10	3	10	2	6	2	6
経路 3(3)	2	7	2	7	2	6	3	10
経路 4(3)	3	10	3	10	2	7	0	0
経路 5(3)	2	7	1	2	3	10	1	3
経路 6(4)	4	10	4	10	2	6	3	8

さによる経路生成では経路辺の長さが短く入り組んだ場所に中間点が生成されてしまう。案内をするときに入り組んだ場所に中間点があると GPS の誤差の影響を受け案内がうまくいかない場合がある。一方、面積による経路生成では 1 つ 1 つの経路辺の長さは短く面積も小さくなるため、経路辺の長さが短く入り組んだ場所には中間点が生成されない。また障害物を避けるという点でも、経路辺の長さによる経路生成では基準とする値によって障害物を避けるように中間点を生成することができない場合があるが、面積による経路生成では大きな障害物であるほど経路辺の長さが長くなり面積も大きくなるため障害物を避けるような点に中間点が生成されやすい (図 10)。また、ほぼ真後ろに大きく曲がるような点は案内において重要であると考えられるので角度の条件は必要であると考えられる。

これらより 4 つの経路生成アルゴリズムのなかでは、角度の条件をつけた面積による経路生成アルゴリズムが一番良いと考えられる。

7. まとめ

本研究では、経路案内を音声のみで行う場合に方向者にとってより分かりやすい案内手法を提案するために、Android 端末用の音声経路案内アプリの開発を行った。既存システムの案内手法の課題であると考えられる、「システムの経路選択がユーザーの直感的な経路選択に沿わない可能性があり、入り組んだ経路だと GPS の誤差にも弱い」という点を解決するべく、既存システムの経路上に「中間点」なるものを配置し、障害物を回避できる程度に案内粒度を下げることで音声道案内システムにも適応するように経路を単純化するシステムを提案した。ユーザに分かりやすい中間点の配置を行うため経路の単純化の際に用いるアルゴリズムを提案した。また提案した実現法をもとに、本論文であげた特徴「スマートフォン端末上で、音声と地図による案内を行う」、および特徴「案内粒度を変更可能な経路案内機能を持ち、それに適した音声案内を行う」の機能を有するプロトタイプシステムを実装し、これによって作られた経路に対する比較実験を行った。

評価実験は L^2 error norm weight アルゴリズムを使った面積による経路生成と経路辺の長さによるアルゴリズム、さらにそれぞれのアルゴリズムに角度の条件を付け加えた

もので中間点を生成しそれらを比較することにより行った。この結果より、面積による経路生成に角度の条件を付け加えたものがこの中では一番ユーザにとってわかりやすい中間点が生成できるという結果が得られた。

今後システムをより高めるための課題点としては、さらにわかりやすい中間点生成アルゴリズムや方位センサーの精度向上、ユーザが直感的に中間点の方向が分かるような案内文などが考えられる。

謝辞

本研究は JSPS 科研費 26330136, 25700009 および、科学技術振興機構 CREST の助成を受けたものです。

参考文献

- [1] 李 晃仲, 大浦 圭一郎, 徳田 恵一, 魅力ある音声インタラクションシステムを構築するためのオープンソースツールキット, 情報処理学会研究報告, Vol. 2011-SL-89, pp. 1-6, 2011.
- [2] Google でできること
<http://www.google.co.jp/dekiru/android/map.html>. (2016/2/6 に参照)
- [3] iOS8
<https://www.apple.com/jp/ios/maps/>. (2016/2/6 に参照)
- [4] iOS7 純正マップ VS Google マップ
<http://www.appps.jp/2105716/>
- [5] waaaaay
<http://waaaaay.com/>. (2016/2/6 に参照)
- [6] Google マップ
<http://maps.google.co.jp/>. (2016/2/6 に参照)
- [7] OpenStreetMap Japan
<https://openstreetmap.jp/>. (2016/2/6 に参照)
- [8] Google Maps API ウェブサービス
<https://developers.google.com/maps/documentation/directions/?hl=ja>. (2016/2/6 に参照)
- [9] 杉山博史, 土井美和子, “交差点形状が与える心理的影響を考慮した道案内システム”, 電子情報通信学会論文誌. A, 基礎・境界 J87-A(1), pp.59-67, 2004.
- [10] 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”, 情報処理学会第 74 回全国大会, 1P-8, 2012.