

G-5

フリープラン観光ルート推薦システムに関する検討
Study on Recommendation System of Sightseeing Route for Free Plan Travel

○田 京寧¹, 滕 琳², 泉 隆²

*JingNing Tian¹, Lin Teng², Takashi Izumi²

Abstract: In recent years, the number of travelers all over the world has continued to increase. Among them, the number of people in the free plan accounts for more than half. In the free plan, it is quite difficult to make plans that are fulfilling and not tired. In this research, we will consider the optimum route recommendation system considering tourism factors using the Google Places API in response to user's requests and Real-time navigation using D* algorithm.

1. まえがき

近年, 世界中の旅行人数は増え続けていき, 特に中国の旅行人数は世界一位となった. その中ではフリープラン旅行の人数が半分以上を占めている^[1]. 一般的な旅行方式とは違い, フリープラン旅行では添乗員がいなく, 自由行動が中心である. 観光客が自分で観光地を探し, 観光ルートを計画するなど自由度が高まる一方, 充実で, かつ疲れない計画を立てるのはなかなか難しい.

そこで, 本稿では, ユーザの希望に従い, 観光要素を考慮した観光ルート推薦システムを構築した. アルタイム経路案内機能を検討したので報告する.

2. システムの概要

本システムでは, ユーザの観光計画を充実させるため, ユーザが入力した始点と目的の間にある寄り道観光地を推薦すると考える.

ユーザが入力し, 希望条件により観光地情報の検索を行う. Google Places API を用いて, 観光地情報を取得し, 観光地の推薦などに用いる. 最後に始点と複数観光地を含む巡回経路探索により最適な観光ルートを推薦し, リアルタイム経路案内をする. システムの流れを図 1 に示す.

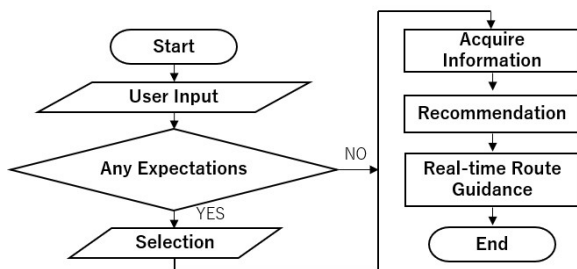


Figure1. System Flowchart

3. システムの構築

3.1 観光地情報取得機能

観光地推薦及び経路探索には先ず観光地情報が必要である. 本研究では, Google Places API^[2]を用いて観光地情報の取得を行った. Google Places API の Place Search 機能 (地点検索機能) について調査を行った結果を表 1 に示す.

Table 1. Place Search Feature Comparison

	返信地点数	必須入力	追加パラメータ	共通取得情報
Find Place requests	単一地点	キーワード	言語、部分返信、検索範囲指定	アドレス、座標、アイコン、ID、
Nearby Search requests	地点リスト	検索範囲	言語、キーワード、距離配列、重要性配列、タグ指定	名称、写真、タグ、PlaceID、営業状態、消費レベル、評判 (0-5)
Text Search requests	地点リスト	テキスト	言語、部分返信、国別コード指定	

Place Search 機能は HTTP URL によりアクセスし, JSON または XML 形式で情報を受信する. Nearby Search requests は観光範囲の指定, 地点数リストで返信するため, 観光地情報の取得に適している.

Place Search で取得した PlaceID を用いて, 営業時間, 消費レベル, 評価値と評価レビューなど地点の詳細情報を更に取得することができる.

3.2 アプリケーション構成

ユーザの利便性を配慮し, スマホアプリによるシステムを構築した. 開発環境は Android Studio を利用し, 観光地情報の取得, 経路探索, 案内結果の表示は Google Map を用いてウェブ画面上に描画し表示するのを実現した.

最初にユーザが入力した始点, 終点及び希望条件を取得し観光地の検索を行う. 一般検索であればデフォルト値で検索を行う. リストにある各アイテム

1 : 日大理工・院 (前) ・情報 2 : 日大理工・教員・情報

は観光地の基本情報を表示し,地点の PlaceID を非表示で格納する. 地図表示は Google Map を用いる.

システムの実行イメージを図 2 に示す.

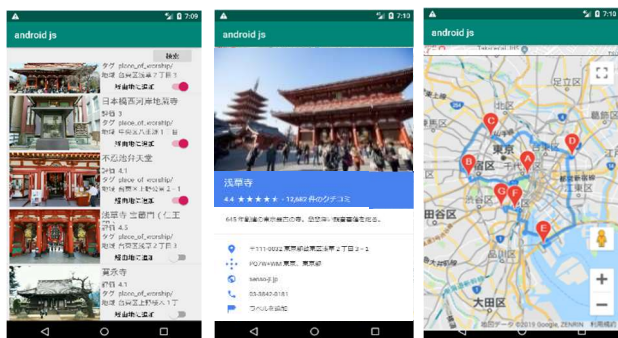


Figure2. System Image

図 2 (左) は抽出した観光地候補の結果リストであり, 観光地の簡易情報と選択スイッチが表示される. 各タブに格納した PlaceID (非表示) を通じて, 図 2 (中) のウェブ画面で観光地の詳細情報を表示する. 観光地を選択し, 巡回経路の経路案内画面を図 2 (右) のように示す.

3.3 巡回経路探索

巡回経路探索の仕組みは, 先ず A*アルゴリズムで各地点間の距離を計算し, 遺伝的アルゴリズムを用いて巡回の順番を計算することを考えた. 遺伝的アルゴリズムを実装し, 全組み合わせ探索との比較実験も行った. 地点数 10 個以下の場合, A*アルゴリズムを用いて地点間の距離を計算した後, 順番を計算する時間を比較した. 結果を以下に示す.

Table 2. Search Algorithm Comparison

地点数	5	6	7	8	9	10
GA(ms)	2.04	2.13	2.056	2.084	2.126	2.142
組合せ(ms)	0.005	0.028	0.134	1.056	8.134	69.29

結果により, 地点数が 10 個以下の場合, 遺伝的アルゴリズムのリアルタイム性を確認した. そして地点数が 9 個以上の場合に組み合わせ探索より効率が高いことが分かった.

4. リアルタイム経路探索

ユーザは渋滞などが原因で旅行時間を無駄にする可能性があり, 走行中常にルート把握できるように, リアルタイム経路案内が必要と考えられる.

リアルタイム経路探索の方法としては, 渋滞情報が更新された際に A*アルゴリズムを用いて再探索するのが一般的である. このような方法は静的再探

索と呼ぶ. D*アルゴリズム^[3]はコストパラメータを経路再探索の途中に変えられ, 動的探索するのが特徴である. A*再探索の機能を完全に実現でき, 快速な再探索の途中に中間データを保存し用いることができるので広域地図に対する少数更新の時, 無駄な重複探索が避けられる.

D*アルゴリズムに関する研究は主にロボットに使われるが^[4], 本研究はロボット地図をデジタル道路地図に変更して, 交通道路のリアルタイム経路案内に使われると考えた. 19881 ノードの模擬地図を用いて, 模擬実験を行って有効性を検証した. A*アルゴリズム再探索との比較実験も行った. 結果を以下に示す.

Table 3. D* Algorithm Result

ノード番号	6391	7379	8227	8935	15443	16291	16859
再探索時間(ms)	0.5041	0.9875	0.2128	3.4389	8.2261	11.6378	25.7686
access数	92	175	39	603	1460	2199	4584

Table 4. A* Algorithm Result

ノード番号	6391	7379	8227	8935	15443	16291	16859
再探索時間(ms)	6.2251	5.5651	6.0486	6.2164	6.0848	6.4657	6.0272
access数	3454	3411	3402	3437	3505	3473	3598

実験結果から, D*アルゴリズムによる再探索時間は A*アルゴリズムよりほとんどの場合少ない. D*アルゴリズムを改善し, 一回のアクセスにかかる時間を短縮すれば効率を大幅に改善できる.

5. まとめ

本研究では観光ルート推薦システムについて検討し, 一部の機能を実装した. 特に, リアルタイム経路案内機能について検討し, アルゴリズムの実装実験を行った. 今後は検討したシステム機能を完成し, リアルタイム経路案内機能を追加したい.

参考文献

[1] 中国旅行研究院, 「2017 全球自由行報告」, <http://www.mafengwo.cn/gonglve/zt-862.html>
 [2] Google Cloud プレイス, <https://cloud.google.com/maps-platform/places/?hl=ja>
 [3] Stentz, Anthony: "Optimal and efficient path planning for partially known environments.", Intelligent Unmanned Ground Vehicles. Springer, Boston, MA, 203-220, 1997.
 [4] Robotic Motion Planning: "A* and D* Search", http://www.cs.cmu.edu/~motionplanning/lecture/AppH-astar-dstar_howie.pdf
 [5] Seven Koenig, Maxim Likhachev: "D*lite", Eighteenth national conference on Artificial intelligence, 476-483, 2002.