

G-19

テーマパークナビゲーションシステム Dnavi におけるカスタムルート作成および検討

Custom Route Creation and Evaluation in Theme Park Navigation System Dnavi

○木下将磨¹, 五味悠一郎²*Shoma Kinoshita¹, Yuichiro Gomi²

Abstract: The objective of this paper is to ensure that theme park navigation systems display routes that are actually passable from the current location to attractions. This paper created custom routes within the park and displayed them. However, when destinations were distant, the shortest path could not be displayed. Possible solutions include increasing the number of random route candidates or utilizing the A* algorithm, which is currently under development.

1. はじめに

テーマパークの1つである東京ディズニーリゾートでは、新型コロナウイルス対策による入場制限がされる前の2013年から2019年までは約3000万人が来場し、コロナ禍が収束後の2023年および2024年では約2750万人が来場していた。来場者が多いことからパーク内で混雑が発生し、アトラクションの待ち時間が長時間化していた。^[1]

2. 先行研究

中野はダイクストラ法を用いて、アトラクション待ち時間を考慮した最短経路を生成する、テーマパークナビゲーションシステム「Dnavi Ver.3.0」を開発した(Figure 1)^[2]。



Figure 1. Dnavi Ver.3.0

ダイクストラ法は、最短経路問題を効率的に解くグラフ理論であり、スタートとゴールまでの最短距離と経路を求めることができる^[3]。

村山はGoogle Maps JavaScript API v3を用いて、Webで地図表示をするシステム「Dnavi Ver.4.0」を開発した。Dnavi Ver.4.0では、Dnavi Ver.3.0で不可能だった地図の拡大縮小、および回転ができない問題を解決した。^[4]

徳岡は現在地からアトラクションまでのルート表示を目標に、Maps SDK for Androidを用いて現在地の取得と、現在地およびアトラクションのマーカースを設置したシステム「Dnavi Ver.3.1」を開発した。^[5]

島津はMaps SDK for AndroidとDirection APIを用いて、Dnavi Ver.3.1における現在地の取得機能を修正し、現在地からアトラクションまでのルート表示機能を追加したシステム「Dnavi Ver.3.2」を開発した。また、修正および追加した機能が正しく動作することを、園内各所で検証した。しかし、ルートが園外を經由しており、目的地から離れた場所で途切れるという課題があった。^[6]

3. 先行研究の課題および本稿の目標

2の先行研究では、現在地からアトラクションまでのルート表示機能において、ルート表示はされたもののルートが園外を經由しており、目的地から離れた場所で途切れるという課題が示された。

本稿の目標は、Maps SDK for Androidのポリライン描画機能を使用して園内にカスタムルートを作成し、ルート表示を行うことによって、現在地からアトラクションまで実際に通れるルートが示されることとする。

4. カスタムルートの作成とルート表示

カスタムルートを作成するため、園内にルートの中点地点となるノードを設置した。ノードは道の分岐点を中心とし、園内において主要な道の分岐が40箇所あったため、40個設置した。カスタムルートは、Maps SDK for Androidのポリライン描画機能を用いて作成した。ルート表示は、Pixel 3aというエミュレータを用いて行った。新たにカスタムルートを追加したテーマパークナビゲーションシステムを「Dnavi Ver.3.3」とする。カスタムルートが表示される流れを以下に示す。

1. 現在地と目的地の間に位置するノードを絞り込む。
2. 絞り込んだノードの中からランダムに通過順序を決め、ルートの候補を作成する。

3. 候補である各ルートの総距離を計算し、最短距離のルートを選択する。
 4. 最短距離のルートをポリラインとして表示する。
5. 結果と考察

設置したノードは、DNavi Ver.3.3 上では視覚的に確認できないようにしたため、一時的に確認できるようにした Google Maps の画面を Figure 2 に示す。



Figure 2. Nodes installed within the park

作成したカスタムルートにおける表示結果の一部を Figure 3 と Figure 4 に示す。現在地はエミュレータ上で園内の適当な場所に設定し、ルートの候補は絞り込んだノード数によって複数通り生まれるため、作成するルートの候補数は 50 個に制限した。



Figure 3. Display results for custom routes(Swiss Family Treehouse)

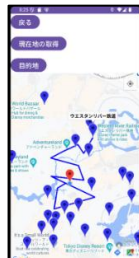


Figure 4. Display results for custom routes(Western River Railway)

Figure 2 より、園内に 40 箇所のノードを設置できた。設置場所に偏りが無いことから、ノードは機能していると考えられる。

Figure 3 と Figure 4 より、作成したカスタムルートによって、ノードを経由したルート表示はできたといえる。Figure 3 は経由しているノードの数が 4 個であり、迂回したルートは表示されず最短経路といえる。しかし、Figure 4 は経由しているノードの数が 10 個であり、迂回したルートが表示され最短経路とはいえない。他の結果も同様であり、現在地から目的地が離れている場合、最短経路は表示できなかった。その原因として、ランダムなルートを候補数 50 個で生成したため、最短経路が候補に含まれていない可能性が考えられる。現在地からノードすべてをランダムに通過し、目的地にたどり着く場合、ルートは 2^n 通り (n はノード数) ある。Figure 4 の場合は、通過したノード数が 10 個で

あるため、1024 通りの候補があると考えられ、最短経路が候補に含まれていない可能性が高い。よって、ランダムなルートの候補数を 50 個とする場合、ノード数を 5 個以下にする必要がある。

6. まとめと今後の課題

本稿では、Maps SDK for Android のポリライン描画機能を使用して園内にカスタムルートを作成し、ルート表示を行った。Figure 4 の結果より、「現在地から目的地が離れている場合、最短経路は表示できない」という問題点がわかった。よって、本稿の目標である「Maps SDK for Android のポリライン描画機能を使用して園内にカスタムルートを作成し、ルート表示を行うことで、現在地からアトラクションまで実際に通れるルートが示されること」は達成できなかった。

今後は、ランダムなルートの候補数を増やす、もしくは開発途中である A* アルゴリズムを用いることで、現在地から目的地が離れていても最短経路が表示されるようにする。

謝辞

本研究にご協力いただいた島津侑希様に感謝申し上げます。

7. 参考文献

- [1] 株式会社オリエンタルランド：入園者数データ，東京ディズニーリゾート。
<https://www.olc.co.jp/ja/ir/olc/group05.html> (参照 2025.09.25).
- [2] 中野雅允，宇田川祐介，五味悠一郎，“アトラクションナビゲータにおける最適経路表示システムの改善と実地検証”，第 61 回日本大学理工学部学術講演会，2017.
- [3] 茨木俊秀．”C によるアルゴリズムとデータ構造”．株式会社オーム社．2019.
- [4] 村山歩，五味悠一郎，“テーマパークアトラクションナビゲーター（D Navi）における地図表示手法の検討”，日本ソフトウェア科学会第 36 回大会，2019.
- [5] 徳岡美華，五味悠一郎，“Maps SDK を用いたテーマパークナビゲーションシステムの検討”，第 67 回日本大学理工学部学術講演会，2023.
- [6] 島津侑希，五味悠一郎，“Maps SDK for Android を用いたテーマパークナビゲーションアプリへのルート表示機能追加と評価”，日本ソフトウェア科学会第 41 回大会，2024