

EVACUATING ORANGE COUNTY, CALIFORNIA, IN ABOUT ELEVEN (11) SECONDS

J. Riechel^{a,*}

^aCenter for Information Systems & Technology (CISAT), Claremont Graduate University (CGU), Claremont, CA 91711, USA,
james.riechel@cgu.edu

KEY WORDS: Vehicle Evacuation Planning by Waypoints, Approximate Driving Distance

ABSTRACT:

Orange County, California, residents must evacuate when there is a crisis at the San Onofre nuclear power plant in San Clemente, California. They must travel roughly north and east over safe roads. Depending on their location in Orange County (OC), residents will travel to the closest of four (4) waypoints located on the border between OC and neighboring counties. Once a waypoint is reached, evacuees can travel in any direction except back toward OC. The approximate driving distance algorithm is used to suggest a possible waypoint for each address—business or residential. The approximate driving distance algorithm makes this evacuation planning possible, as it takes only around eleven (11) seconds on a state-of-the-art laptop to route 1.1 to 1.2 million addresses to waypoints. Using actual driving distances would take too long and be too expensive, taking approximately fifty-three (53) days on the same platform. The waypoint suggestions are just that: suggestions. In some cases, the approximate driving distance algorithm might not choose the closest waypoint.

1. INTRODUCTION/LITERATURE REVIEW

First, you need to know what population(s) to evacuate. In this work, it is assumed that the entire (business and residential) County of Orange, California, needs to be evacuated. Yet this may not be the case. In Yanovskiy et al. (2020), it is determined that “[b]ased on the performed analysis, we suggest avoiding evacuation if the projected first-year dose is below 500 mSv.” One (1) millisievert (mSv) is the average amount of background radiation an individual will absorb in one (1) year. They base their analysis on data from the Fukushima Daiichi nuclear disaster in Japan.

“After the accidents of nuclear power plants at Chernobyl and at Fukushima, huge amounts of radioactive iodine were released into the atmosphere” (Ory et al., 2020). Of the two million children who lived close to Chernobyl, 7,000 cases of thyroid cancer were diagnosed in 2005. Further study is required to determine if these numbers are significant, as thyroid cancer is a very common disease. Yoshimura et al. (2020) report that there were significant differences between the types and amounts of radiation exposure at Chernobyl and Fukushima.

In Thompson et al. (2017), it is determined that people having an evacuation plan in hand are more likely to follow evacuation instructions: “Risk perception was a consistent positive predictor of evacuation, as were several demographic indicators, prior evacuation behavior, and having an evacuation plan.”

In Alabdouli (2017), the case of an Orange County, California, tsunami is considered. If the case of an OC tsunami were to be considered here instead of a San Onofre nuclear power plant crisis, the evacuation instructions would change to: head east to any higher ground!

The rest of this paper is as follows...

- Section 2.1: The approximate driving distance algorithm
- Section 3: A case study
- Section 3.1: The evacuation planning of the case study
- Section 4: The results of the case study
- Section 5: Limitations and a discussion
- Section 6: Conclusions and future work
- Acknowledgments, references, and an appendix containing source code used in the case study

2. MATERIALS AND METHODS

2.1. Approximate driving distance

The following closely follows Riechel (2020c):

Each line of the input file has a longitude, latitude pair (in degrees) of an address. First, convert these from degrees to radians. Then compute the x , y , z coordinates of each address:

$$\begin{aligned}x &= \text{longitude} * r * \text{Cos}(\phi_0) \\y &= \text{latitude} * r \\z &= \text{elevation}\end{aligned}$$

where r is the radius of Earth, and ϕ_0 is a centrally located latitude in the dataset. This forms an equirectangular projection (Hargitai et al., 2019).

Say (x_i, y_i, z_i) is the coordinates of an address, and (x_j, y_j, z_j) is the coordinates of a waypoint. The approximate driving distance between them is:

$$\text{distance}(i, j) = \text{Abs}(x_i - x_j) + \text{Abs}(y_i - y_j) + \text{Abs}(z_i - z_j)$$

* Corresponding author

This approximate distance, the Manhattan distance (Singh et al., 2013), is both a better approximation of the actual driving distance than the Euclidean distance and an order of magnitude faster to compute than it (Riechel, 2019, 2020a, 2020b). See Zeager and Stütz (2016) for a description of Euclidean distance.

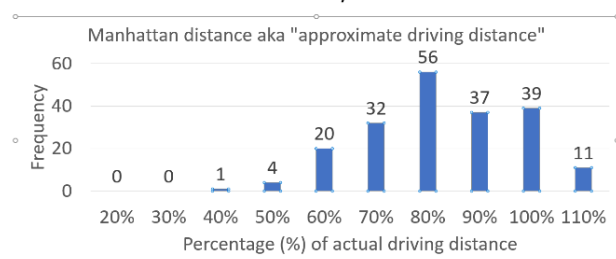
The Manhattan distance is extremely fast to compute:

	Running time of 100 million calls (milliseconds)	Operations per second
Manhattan distance, aka “approximate driving distance”	296	337,837,838

Table 1. Execution speed of Manhattan distance, aka “approximate driving distance”

A pilot study was performed to determine how accurate Manhattan distances are to actual driving distances. In this nonrepresentative study of 200 green taxi cabs rides in New York City on January 1, 2016, starting at 12 AM EST, the following distribution was produced:

Accuracy of Manhattan distance in NYC pilot study



Graph 1. Accuracy of Manhattan distance in NYC pilot study

This distribution is confirmed as normal. On average, the Manhattan distance is about 80% of the actual driving distance, and the Manhattan sometimes overestimates actual driving distance.

There are many reasons why Manhattan distance might underestimate actual driving distance, including:

- Traffic controls (U-turns, one-way streets)
- Manmade obstacles (bridges)
- Natural obstacles (lakes, hills, mountains)
- Etc.

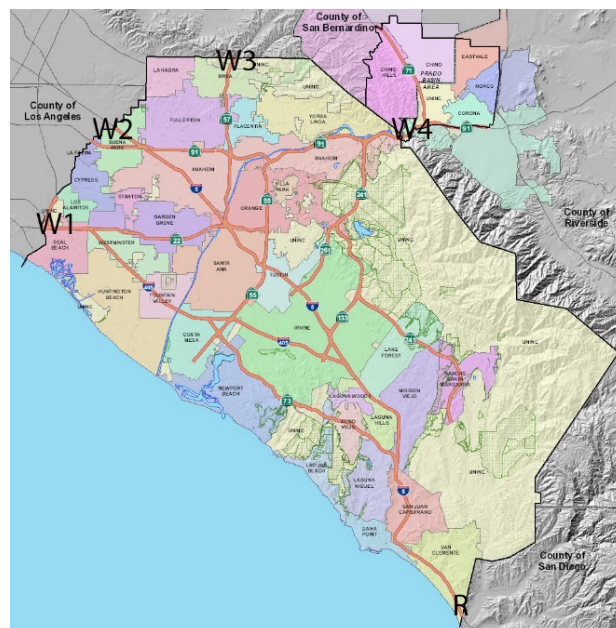
Also, the equirectangular projection tends to underestimate distance.

The approximate driving distance algorithm (Riechel, 2019, 2020a, 2020b) makes this evacuation planning possible. Using actual driving distances would take too long and be too expensive. The waypoint suggestions are just that: suggestions. In some cases, the approximate driving distance algorithm might not choose the closest waypoint.

3. CASE STUDY

In the following map, the San Onofre nuclear power plant is labeled “R” for “reactor.” The four waypoints out of Orange County, California, are labeled “W1,” “W2,” “W3,” and “W4.”

The four waypoints exit Orange County on the 405, 5, 57, and 91 freeways, respectively. Once a waypoint is reached, evacuees can travel in any direction, except back toward Orange County.



Map 1. The reactor and the four waypoints: map courtesy of Orange County, California (<https://www.ocgis.com/ocpw/landrecords/>)

The following table gives more detailed information on the waypoints:

Waypoint	Description	Longitude (degrees)	Latitude (degrees)
W1	405 on LA/Orange County border	-118.0931346	33.7859273
W2	5 on the LA/Orange County border	-118.0114088	33.8748110
W3	57 on the LA/Orange County border	-117.8683537	33.946003
W4	91 on Riverside/Orange County border	-117.6717187	33.8695321

Table 2. The four waypoints

3.1. Evacuation planning

Each address is sent a color map of Orange County (much like Map 1 above), showing the nuclear reactor R and all four waypoints: W1, W2, W3, and W4.

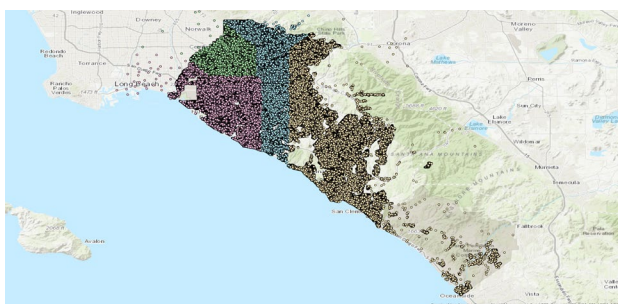
For each address in Orange County (residential or business), the approximate driving distance algorithm is used to find the nearest waypoint.

Along with the map, a suggested or recommended waypoint is given.

Residents do not have to follow the recommended waypoint. Instead, they can choose any of W1, W2, W3, or W4 in the event of a crisis at R.

4. RESULTS

The following map shows the results of evacuating County of Orange, California, to the four waypoints:



Map 2. The results

The addresses are color-coded to specify suggested evacuation waypoints:

Color	Evacuation waypoint
Purple	Evacuate by 405 freeway
Green	Evacuate by 5 freeway
Blue	Evacuate by 57 freeway
Brown	Evacuate by 91 freeway

Table 3. Explanation of color codes in Map 2

The following table presents the number of addresses for each color code, and the total number of addresses:

Waypoint	Count (addresses)
W1	306,249
W2	129,219
W3	266,898
W4	445,429
TOTAL	1,147,795

Table 4. The number of addresses routed to each waypoint, and the total number of addresses

5. LIMITATIONS/DISCUSSION

The presented software routed 1.1 to 1.2 million addresses in OC to the closest of four (4) waypoints in about eleven (11) seconds, because the approximate driving distance algorithm was used on a state-of-the-art laptop.

It is estimated that computing actual driving distances instead of approximate driving distances would take more than 53 days.

The main drawback of the approximate driving distance algorithm is its accuracy. On average, the approximate driving distance is about 80% of actual driving distances (Riechel, 2019, 2020a, 2020b).

Li et al. (2016) introduce the “six Vs” of Geospatial Big Data: volume, variety, velocity, veracity, visualization, and visibility. “In a world filled with Big Data, where the volume of data points to compute distances between, and the velocity at which these distances are expected to be computed, are both extremely high, a fast algorithm for computing approximate distances may be the only choice. This introduces the issue of veracity: How reliably accurate these approximate distances are” (Riechel, 2020c).

6. CONCLUSION/FUTURE WORK

Every evacuation scenario differs and requires custom software to solve.

The approximate driving distance algorithm might prove key to other evacuation problems, not just that of OC.

The approximate driving distance algorithm aids in the development and testing of evacuation software as it can be done quickly in real time.

For future work, the author would like to try solving more evacuation problems in the United States.

ACKNOWLEDGMENTS

Mike Vuong, GIS Supervisor, County of Orange, California, Fire Authority

Full Clinical Professor Brian N. Hilton and Professor Warren Roberts, Center for Information Systems & Technology (CISAT), Claremont Graduate University (CGU)

REFERENCES

- Alabdouli, K. (2017). Hypothetical tsunami scenarios and evacuation intention: A case study of Orange County, California. *Journal of Geography and Natural Disasters*, 7(3), 1–9. <https://doi.org/10.4172/2167-0587.1000212>
- Hargitai, H., Willner, K., & Hare, T. (2019). Fundamental Frameworks in Planetary Mapping: A Review. In H. Hargitai (Ed.), *Planetary Cartography and GIS* (pp. 75–101). Springer International Publishing. https://doi.org/10.1007/978-3-319-62849-3_4
- Li, S., Dragicevic, S., Castro, F. A., Sester, M., Winter, S., Coltekin, A., Pettit, C., Jiang, B., Haworth, J., Stein, A., & Cheng, T. (2016). Geospatial big data handling theory and methods: A review and research challenges. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115, 119–133. <https://doi.org/10.1016/j.isprsjprs.2015.10.012>
- Ory, C., Leboulleux, S., Salvatore, D., Le Guen, B., De Vathaire, F., Chevillard, S., & Schlumberger, M. (2020). Consequences of atmospheric contamination by radioiodine: The Chernobyl and Fukushima accidents. *Endocrine*. <https://doi.org/10.1007/s12020-020-02498-9>

Riechel, J. (2019). A fast algorithm for computing approximate distances in the Cartesian plane. *Proceedings of URISA GIS-Pro '19*, New Orleans, LA. <https://drive.google.com/file/d/126GnuqNw2V9dMUgl8He4j95JXs0VoUQ/view?usp=sharing>

Riechel, J. (2020a). *Comparing Manhattan, Euclidean, and Actual Driving Distances*. Proceedings of CalGIS 2020, Long Beach, CA. https://drive.google.com/file/d/1_wgjePJH6LXM6OAYHg-PJkr2o-wVr8AK/view?usp=sharing

Riechel, J. (2020b). *Extending Manhattan, Euclidean, and Actual Driving Distances into 3D*. Unpublished. <https://drive.google.com/file/d/16rkw9Ysn8BWfT7CpflnlaSwUgBw4Y4v/view?usp=sharing>

Riechel, J. (2020c). *Extremely Fast "Solution" to the Large-Scale and Very Large-Scale Vehicle Routing Problem*. Unpublished. <https://drive.google.com/file/d/1dq756Oua9iCrVG764ug7OVg55BRFzBO3/view?usp=sharing>

Singh, A., Yadav, A., Block, A. E., Rana, A., Block, E., & Floor, G. (2013). K-means with Three Different Distance Metrics. *International Journal of Computer Applications*, 67(10), 13–17. <https://doi.org/doi:10.5120/11430-6785>

Thompson, R. R., Garfin, D. R., & Silver, R. C. (2017). Evacuation from Natural Disasters: A Systematic Review of the Literature. *Risk Analysis*, 37(4), 812–839. <https://doi.org/10.1111/risa.12654>

Yanovskiy, M., Levi, O. N., Shaki, Y. Y., & Socol, Y. (2020). Consequences of a large-scale nuclear accident and guidelines for evacuation: A cost-effectiveness analysis. *International Journal of Radiation Biology*, 96(11), 1,382–1,389. <https://doi.org/10.1080/09553002.2020.1779962>

Yoshimura, K., Saegusa, J., & Sanada, Y. (2020). Initial decrease in the ambient dose equivalent rate after the Fukushima accident and its difference from Chernobyl. *Scientific Reports*, 10(1), 3,859. <https://doi.org/10.1038/s41598-020-60847-0>

Zeager, J., & Stitz, C. (2016). *College Algebra*. <http://dspace.calstate.edu/handle/10211.3/180387>

APPENDIX

A1. Source code file, 'main.cpp'

```
#include "math.h"
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>

using namespace std;

#define EARTH_RADIUS (6371000)
#define PI (3.14159265359)
#define DEG2RAD(deg) (deg/180.0*PI)

int main() {
```

```
    const int number_of_rows = 1147795;

    // this is a two-pass algorithm. on the
    // first pass, compute the average latitude:

    ifstream input_file("xy.csv");
    string line;
    getline(input_file, line); // read past
    "longitude,latitude" line

    double average_latitude = 0.0;
    for (int i = 0; i < number_of_rows; i++)
    {

        getline(input_file, line);
        stringstream s(line);
        string word;
        getline(s, word, ',');
        std::string::size_type pos;
        double longitude = stod(word, &pos);
        getline(s, word, '\n');
        double latitude = stod(word, &pos);
        average_latitude += latitude;
    }

    average_latitude =
    DEG2RAD(average_latitude /
    ((double)number_of_rows));

    input_file.close(); // end of first pass

    // begin of second pass: do everything
    else

    ifstream input_file2("xy.csv");
    getline(input_file2, line); // read past
    "longitude,latitude" line

    ofstream output_file("Output2.csv");

    output_file <<
    "longitude,latitude,W1,W2,W3,W4,Waypoint" <<
    endl;

    double cos_Phi_0 = cos(average_latitude);
    double x_mult = EARTH_RADIUS * cos_Phi_0;
    double y_mult = EARTH_RADIUS;

    double Waypoint[4][2];

    Waypoint[0][0] = DEG2RAD(-118.0931346) *
    x_mult;
    Waypoint[0][1] = DEG2RAD(33.7859273) *
    y_mult;
    Waypoint[1][0] = DEG2RAD(-118.0114088) *
    x_mult;
    Waypoint[1][1] = DEG2RAD(33.8748110) *
    y_mult;
    Waypoint[2][0] = DEG2RAD(-117.8683537) *
    x_mult;
    Waypoint[2][1] = DEG2RAD(33.946003) *
    y_mult;
```

```
Waypoint[3][0] = DEG2RAD(-117.6717187) *
x_mult;;
Waypoint[3][1] = DEG2RAD(33.8695321) *
y_mult;

for (int i = 0; i < number_of_rows; i++)
{

    getline(input_file2, line);
    stringstream s(line);
    string word;
    getline(s, word, ',');
    std::string::size_type pos;
    double longitude = stod(word, &pos);
    getline(s, word, '\n');
    double latitude = stod(word, &pos);

    double x = DEG2RAD(longitude) *
x_mult;
    double y = DEG2RAD(latitude) *
y_mult;

    double distances[4];
    for (int j = 0; j < 4; j++) {

        distances[j] = abs(Waypoint[j][0]
- x) + abs(Waypoint[j][1] - y);
    }

    double shortest_distance =
distances[0];
    int shortest_index = 0;
    for (int j = 1; j < 4; j++) {

        if (distances[j] <
shortest_distance) {

            shortest_distance =
distances[j];
            shortest_index = j;
        }
    }

    output_file << longitude << "," <<
latitude << "," << distances[0] << "," <<
distances[1] << "," << distances[2] << ","
<< distances[3] << "," << (shortest_index +
1) << std::endl;
    }

    input_file2.close();
    output_file.close();
}
```