SIMUTHON

Dokumentacja - Task1

FIND THE LOWEST-COST PATH BETWEEN TWO POINTS IN KRAKÓW OLD TOWN

Rafał Szygenda MSKN Decybel

2 grudnia 2023

Spis treści

1 Opis działania rozwiązania

Poniższa dokumentacja opisuje rozwiązanie 1 zadania z Simuthon 2023©. Pliki źródłowe znajdują się na repozytorium [1].

Funkcja dokonuje znalezienia optymalnej trasy względem podanej funkcji kosztów

$$cost = speedLimitCost \cdot traficIntensity + ObstacleCost \tag{1}$$

Do znalezienia optymalnej trasy wykorzystano zmodyfikowaną funkcję A-star, a dokładniej funkcję która dla tego algorytmu jest zminimalizowana. Aby móc wykorzystać funkcję A-star, dokonano konwersji mapy do "binaryOccupancyMap". Następnie zmodyfikowano funkcję kosztów. Funkcję kosztów algorytmu można określić jako:

$$f(n) = g(n) + h(n) \tag{2}$$

gdzie:

- f(n) - całkowita funkcja kosztu
- g(n) - funkcja kosztu trasy
- h(n) - jest heurystyczna funkcją która estymuje koszt "najtanszej" trasy od n do konca.

Modyfikacje wykonano poprzez zdefiniowanie niestandardowej funkcji kosztów $Gcost(n)$ danej równaniem 2. Dokładniejszy opis działania funkcji w Punkcie 2.

2 Funkcje dodatkowe

2.1 Custom G cost

Funkcja przyjmuje argumenty które są kolejno

- Aktualny punkt
- Następny punkt
- Mapa kosztów prędkości
- Mapa kosztów zatłoczenia
- Mapa kosztów przeszkód

Funkcja oblicza koszt wg równania 2, oraz sprawdza czy ruch jest diagonalny, jeżeli tak, przemnaża koszt o $\sqrt{2}$.

2.2 Custom H cost

Funkcja zwraca koszt = 0, tak aby równanie 2 było wyłącznie kosztem algorytmu. Testy wykazały, iż przyrównanie $h(n) = g(n)$, nie zmienia wyników operacji, generując tą samą trasę, oraz ten sam koszt.

2.3 startStopGenerator

Funkcja przygotowana przez organizatorów Simuthon©. Funkcja generująca prawidłowe punkty start stop dla zadanej mapy.

2.4 task 1

Funkcja implementuje wszystkie powyższe funkcje. Kilku warstwowa mapa zostaje podzielona na 4 oddzielne zmienne, w celu wykorzystania ich w funkcjach kosztów oraz planowaniu trasy algorytmem A-Star. Funkcja dokonuje obliczeń oraz zwraca *path* w punktach jako macierz$_{n\times 2}$, gdzie n -liczba punktów otrzymanej trasy. Zwracana lista punktów jest zgodna z wytycznymi, takimi, że punkty oddalone są od siebie o 1 pixel.

3 Generowanie Kodu do C/Cpp

Wykorzystano wewnętrzne narzędzie Matlab - **Matlab Coder**. Za pomocą tego narzędzia wygenerowano niezbędne pliki do uruchomienia funkcji za pomocą C++. Wygenerowane pliki znajdują się w repozytorium, w folderze *Task1/codegen*. Przed generacją kodu, funkcja $task1.m$ musi być skrócona o dwie linijki [lst.1], tak aby konwersja przebiegła pomyślnie.

```
%... reszta kodu task_1.m
show(planner);
disp(['total custom cost (implemented function) = ' num2str(solution_info.PathCost) ])
```

Listing 1: Fragment kodu wymagający usunięcia, w celu użycia **Matlab Coder**

Literatura

[1] *Repozytorium MSKN Decybel* - https://github.com/wiktornowacki/MSKN_Decybel_23