



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
WYDZIAŁ GEODEZJI GÓRNICZEJ I INŻYNIERII ŚRODOWISKA

KATEDRA FOTOGRAMETRII, TELEDETEKCJI ŚRODOWISKA
I INŻYNIERII PRZESTRZENNEJ

Praca dyplomowa magisterska

*Automatyzacja generowania optymalnych tras na podstawie
danych rastrowych i wektorowych w środowisku QGIS
z wykorzystaniem języka Python.*

*Automation of optimal routes generation based on
raster and vector data in the QGIS environment
using Python language.*

Autor:	<i>Karolina Katarzyna Aksamić</i>
Kierunek studiów:	<i>Geodezja i Kartografia</i>
Opiekun pracy:	<i>dr inż. Mariusz Twardowski</i>

Kraków, 2020

Kraków, dnia 17.09.2020

Karolina Katarzyna Aksamit
Imiona i nazwisko studenta

Dr inż. Mariusz Twardowski
Imiona i nazwisko opiekuna pracy dyplomowej

Automatyzacja generowania optymalnych tras
na podstawie danych rastrowych i wektorowych
w środowisku QGIS z wykorzystaniem języka Python.
Tytuł pracy dyplomowej w języku polskim

Automation of optimal routes generation
based on raster and vector data
in the QGIS environment using Python language.
Tytuł pracy dyplomowej w języku angielskim

STRESZCZENIE

Główne cele badań: Zaprojektowanie i utworzenie wtyczki do programu QGIS pozwalającej wyznaczyć optymalne trasy dla zadanych zestawów parametrów wraz z analizą wiarygodności oraz zmienności generowanych wyników.

Zakres badań: Teoretyczne podstawy automatyzacji w integracji i przetwarzaniu danych w aspekcie GIS, logistyki i transportu. Ocena poprawności działania zaimplementowanych algorytmów wraz z analizą wiarygodności generowanych wyników. Weryfikacja zmienności obliczanych atrybutów względem cech trasy oraz zastosowania zmiennego rozmiaru piksela obrazu wysokościowego. Wpływ metody interpolacji na pozyskiwane wartości. Określenie skali i przyczyn rozbieżności danych. Wykorzystanie aplikacji w logistyce i transporcie.

Metodyka badań: Implementacja algorytmów w języku Python w środowisku QGIS z wykorzystaniem generatora wtyczek Plugin Builder. Przeprowadzenie pomiaru terenowego czasów przejazdów i przejazdów na zaprojektowanych trasach w wybranym rejonie Krakowa z użyciem aplikacji Endomondo. Sprawdzenie wiarygodności i zmienności wyników na podstawie obliczeń przeprowadzonych w programie Microsoft Excel. Wizualizacja danych w postaci wykresów wygenerowanych za pomocą platformy Google Colab oraz biblioteki Plotly Express.

Uzyskane wyniki: Błędy średnie czasu, obliczone na podstawie wyników generowanych przez aplikację na oryginalnym rastrze 1x1 m oraz wartościach pomierzonych w terenie, osiągnęły wartości: 32 s (ruch pieszy) oraz 1 min 3 s (ruch samochodowy). Wyniki najbliższe rzeczywistości uzyskano dla rozmiaru piksela 5x5 m z błędami średnimi: 29 s oraz 1 min 4 s, natomiast najgorsze dla rozmiaru piksela 100x100 m z wartościami odpowiednio: 2 min 5 s, 1 min 4 s. Różnica błędów średnich między dwoma metodami interpolacji wyniosła niecałe 4 sekundy.

Wnioski: Wartości błędów średnich wskazują na stosunkowo wysoką dokładność szacowania czasu przebycia drogi przez nowo utworzoną wtyczkę. Zmiana rozdzielczości rastra znacząco zmienia otrzymywane wyniki. Zastosowanie różnych metod interpolacji w niewielkim stopniu wpływa na wartości generowanych atrybutów. Głównymi przyczynami rozbieżności były: sygnalizacja świetlna, ruch drogowy, konieczność zmiany pasa ruchu bądź zatrzymania się na przejściach dla pieszych.

SUMMARY

The main objectives research: Designing and creating a plug-in for the QGIS program allowing to determine optimal routes for given sets of parameters together with analysis of the reliability and variability of the generated results.

Scope of research: Theoretical basis for automation in data integration and processing in terms of GIS, logistics and transport. Assessment of the correctness of operation of the implemented algorithms operation along with analysis of generated results reliability. Verification of the variability of calculated attributes in relation to route features and use of a variable pixel size of the height image. Impact of the interpolation method on acquired values. Determining the scale and causes of data discrepancies. Application use in logistics and transport.

Methodology of research: Implementation of Python algorithms in the QGIS environment using the Plugin Builder. Conducting a field measurement of the times of passes and journeys on the designed routes in a selected area of Cracow using the Endomondo application. Credibility and variability check of the results on the basis of calculations made in Microsoft Excel. Data visualization in the form of graphs generated using the Google Colab platform and the Plotly Express library.

Obtained results: Average time errors calculated on the basis of the results generated by the application on the original 1x1 m raster and the values measured in the field reached the following values: 32 s (pedestrian traffic), 1 min 3 s (car traffic). The results closest to reality were obtained for a pixel size of 5x5 m with average errors: 29 s and 1 min 4 s, while the worst for 100x100 m with the values: 2 min 5 s, 1 min 4 s. The difference in mean errors between the two interpolation methods was less than 4 seconds.

Conclusions: The mean error values indicate a relatively high accuracy of estimating the travel time of the newly created plug-in. Changing the raster resolution significantly changes the obtained results. Use of various interpolation methods has little influence on the values of the generated attributes. The main reasons for the divergence were: traffic lights, road traffic, the need to change lanes or stop at pedestrian crossings.

Spis treści

WSTĘP	4
1. SYSTEMY INFORMACJI GEOGRAFICZNEJ W LOGISTYCE I TRANSPORCIE	5
1.1. ZNACZENIE I OBSZARY ZASTOSOWANIA GIS	5
1.1.1 <i>Logistyka i transport</i>	5
1.1.2 <i>GIS w procesie zarządzania danymi</i>	6
1.2. INTELIGENTNE SYSTEMY LOKALIZACYJNO-NAWIGUJĄCE	6
1.3. PROBLEMY IMPLEMENTACJI OPTIMALNYCH ROZWIĄZAŃ	8
2. AUTOMATYZACJA W INTEGRACJI I PRZETWARZANIU DANYCH	10
2.1. MODELE DANYCH PRZESTRZENNYCH	10
2.1.1 <i>Model rastrowy</i>	10
2.1.2 <i>Model wektorowy</i>	11
2.2. CYFROWE PRZETWARZANIE OBRAZÓW	11
2.3. WPROWADZENIE DO JĘZYKA PYTHON	12
2.4. ALGORYTMY WYZNACZANIA TRAS	14
3. METODYKA I DANE	18
3.1. ZAŁOŻENIA I TECHNOLOGIE PRACY	18
3.2. CHARAKTERYSTYKA WARSTW RASTROWYCH I WEKTOROWYCH	19
3.3. OPROGRAMOWANIE I APLIKACJE	21
4. OPRACOWANIE WTYCZKI QGIS	22
4.1. IMPLEMENTACJA ALGORYTMÓW	22
4.2. INTERFEJS GRAFICZNY	26
4.3. INSTRUKCJA OBSŁUGI	29
5. OPIS PRZEPROWADZONYCH BADAŃ	33
5.1. POZYSKANIE DANYCH PORÓWNAWCZYCH Z TERENU	33
5.2. GENEROWANIE TRAS Z WYKORZYSTANIEM WTYCZKI	36
5.2.1 <i>Dane testowe</i>	36
5.2.2 <i>Dane kontrolowane</i>	38
5.2.3 <i>Zmienna rozdzielczość rastra</i>	39
5.3. WYNIKI BADAŃ	45
5.3.1 <i>Poprawność działania algorytmów</i>	45
5.3.2 <i>Wiarygodność wyników dla różnego rozmiaru piksela</i>	46
6. ANALIZA DOKŁADNOŚCI	53
7. WNIOSKI	58
8. LITERATURA	60
9. SPIS TABEL	62
10. SPIS RYSUNKÓW	65
11. SPIS WZORÓW	67
12. SPIS ZAŁĄCZNIKÓW	68

Wstęp

Rozwój nowoczesnych technologii wiąże ze sobą potrzebę automatyzacji procesów umożliwiających podniesienie efektywności pracy przy jednoczesnym obniżeniu kosztów oraz oszczędności czasu. W logistyce i transporcie duże znaczenie odgrywają analizy przestrzenne, które umożliwiają wydajne zarządzanie zasobami ludzkimi i materialnymi.

Jednym z najbardziej popularnych narzędzi nawigacyjno-lokalizacyjnych we współczesnym świecie są Mapy Google, które zapewniają wyznaczenie zoptymalizowanej trasy, dostarczają informacji o ruchu drogowym w czasie rzeczywistym, a także umożliwiają wyświetlanie widoku Street View oraz Satelity. Aplikacja ta wyposażona jest w wiele funkcji skupiających się na spełnieniu podstawowych wymagań zwykłego użytkownika. Jednak w przypadku potrzeby wprowadzenia własnych danych sytuacyjno-wysokościowych bądź uwzględniania bardziej szczegółowych parametrów generowanej drogi, Mapy Google nie są już w stanie dostarczyć oczekiwanego rozwiązania, wymaga to bowiem opracowania i zbudowania narzędzia dostosowanego do indywidualnych potrzeb.

W niniejszej pracy dyplomowej poruszony zostanie temat automatyzacji, integracji oraz analiz danych przestrzennych w środowisku QGIS z wykorzystaniem języka Python. Głównym celem przeprowadzonych w niej badań jest określenie poprawności działania nowo utworzonej wtyczki wraz z wykonaniem analizy wiarygodności generowanych wyników.

Pierwsze dwa rozdziały posłużą przybliżeniu podstaw teorii związanej z GIS, logistyką i transportem. Poruszony zostanie tam także temat przetwarzania danych oraz opracowywania rozwiązań dla potrzeb wyznaczania optymalnych tras.

Następna część zostanie poświęcona opisowi głównych założeń pracy, przedstawieniu metodyki przeprowadzonych badań oraz przybliżeniu wykorzystanych technologii. Dodatkowo będzie ona zawierać ogólną charakterystykę danych, programów i aplikacji wykorzystanych w projekcie.

W kolejnym rozdziale przedstawiony zostanie schemat opracowywania algorytmów oraz szczegółowy opis budowy interfejsu graficznego. Zostanie w nim również załączona dokładna instrukcja obsługi, służąca zapoznaniu się z możliwościami zastosowania oraz sposobem działania nowo utworzonej wtyczki.

Jednym z ostatnich etapów będzie zaprezentowanie przeprowadzonych badań oraz wykonanie analizy dokładności uzyskanych wyników. Otrzymane wartości zostaną przedstawione z wykorzystaniem tabel oraz graficznej wizualizacji w postaci wykresów. W tej części pojawi się opis skali i przyczyn występujących rozbieżności (między trasami pomierzonymi w rzeczywistości oraz za pomocą aplikacji), a także charakterystyka zachowywania się wyników w przypadku wprowadzania zmian związanych z rozdzielczością obrazu bądź manipulowaniem dostępnymi parametrami.

Na zakończenie sformułowane zostanie podsumowanie zawierające główne spostrzeżenia wynikające z przeprowadzonych badań. Pojawią się tam także wnioski dotyczące oceny poprawności działania nowo zbudowanej aplikacji oraz skomentowane zostaną wyniki przeprowadzonych analiz. Podsumowanie badań obejmie rozważania dotyczące zastosowania rastrów o odmiennych rozdzielczościach lub różnych metodach interpolacji, a także zawierać będzie wnioski związane z generowaniem tras dla zestawów o różnie zdefiniowanych parametrach. Na końcu znajdą się przemyślenia wskazujące drogę rozbudowy oraz możliwe miejsca zastosowania nowo utworzonej wtyczki.

Kody źródłowe do opracowanej aplikacji oraz wszystkich wykresów są ogólnodostępne i znajdują się na platformie GitHub pod adresami:

https://github.com/AksamitKarolina/Praca_Magisterska_FindRoute_Aksamit_Karolina.git

https://github.com/AksamitKarolina/Praca_Magisterska_Wykresy.git

1. Systemy Informacji Geograficznej w logistyce i transporcie

1.1. Znaczenie i obszary zastosowania GIS

1.1.1 Logistyka i transport

Sytuacja społeczna-gospodarcza XXI wieku wymaga ciągłego rozwoju nowych technologii oraz rozwiązań w procesach zarządzania zasobami i czasem. Jednym z najważniejszych filarów efektywności wprowadzanych działań jest logistyka. Próbę jej definicji podjął Frederick J. Beier wraz z Krzysztofem Rutkowskim. „Przez logistykę rozumiemy z reguły pojęcie oznaczające zarządzanie działaniami przemieszczania i składowania, które mają ułatwić przepływ produktów z miejsc pochodzenia do miejsc finalnej konsumpcji, jak również związaną z nimi informacją w celu zaoferowania klientowi odpowiedniego poziomu obsługi po rozsądnych kosztach.” [1].

Istotną rolę w usługach logistycznych stanowi transport, którego optymalizacja jest głównym narzędziem oszczędności, zarówno w odniesieniu do czasu, jak i wydatków pieniężnych. Dodatkowo działania zwiększające efektywność pracy mogą pozytywnie wpłynąć na czynniki służące ochronie środowiska oraz podnoszeniu standardów życia ludzi. Największym udziałem w rozwoju branży logistycznej charakteryzują się kraje wysoko rozwinięte [24]. Wskazuje to na światowy trend podążania za coraz silniej zoptymalizowanymi rozwiązaniami umożliwiającymi ograniczenie korzystania z zasobów ludzkich przy jednakowym lub wydajniejszym poziomie świadczonych usług bądź produkcji.

W celu umożliwienia porównywania skuteczności różnych państw w odniesieniu do logistyki handlu, Bank Światowy stworzył narzędzie Indeksu Wydajności Logistycznej (ang. Logistics Performance Index, LPI), które opiera się na obliczeniu średniej ważonej dla sześciu głównych filarów. Biorą one pod uwagę: wygląd procesu rozliczania się przy odprawie celnej (szybkość, intuicyjność, przewidywalność), poziom infrastruktury transportowej (sieć dróg i autostrad, koleje, porty, lotniska), konkurencyjność kosztu przewozów, jakość usług logistycznych (operowanie językami obcymi, kontaktowość, sprawność w obsłudze), specyfikę śledzenia przesyłek oraz terminowość ich dostarczania do punktów docelowych. W celu ujednolicenia danych, wszystkie elementy są poddawane analizie składowych głównych (ang. Principal Component Analysis, PCA). Należy przy tym podkreślić, że całość procesu wyznaczenia indeksu jest związana nie tylko z zastosowaniem odpowiednich obliczeń, ale również z opracowaniem zbioru warunków oraz sposobów kategoryzacji. [35]

W swoim zestawieniu Bank Światowy przypisał największą skuteczność następującym państwom: Niemcy, Szwecja, Belgia, Austria, Japonia, Holandia, Singapur (na stan z 2018 roku). Dwudziesta ósma pozycja Polski w rankingu z wynikiem 3.54, w porównaniu z pierwszą pozycją, gdzie indeks wyniósł 4.20, wskazuje na potrzebę wprowadzania zmian poprzez opracowywanie oraz wdrażanie coraz nowocześniejszych i wydajniejszych systemów logistycznych (Tabela 1). Szczególnie mocne usprawnienie powinno zostać zaimplementowane w odniesieniu do dwóch pierwszych wymiarów indeksu. [24]

Tabela 1. Porównanie wartości oraz składowych wskaźnika LPI dla Polski oraz Niemiec [24].

Kraj	LPI	Obsługa celna	Infrastruktura	Przesyłki	Logistyka	Śledzenie przesyłek	Terminowość
Niemcy	4.20	4.09	4.37	3.86	4.31	4.24	4.39
Polska	3.54	3.25	3.21	3.68	3.58	3.51	3.95

1.1.2 GIS w procesie zarządzania danymi

Systemy Informacji Geograficznej (ang. Geographics Information System, GIS) służą głównie do pozyskiwania, modyfikacji oraz przechowywania danych przestrzennych. Stanowią także narzędzie do ich weryfikacji, integracji oraz udostępniania. W znacznie szerszym znaczeniu ich strukturę można rozumieć jako sieć powiązanych ze sobą elementów, na którą składają się: metody, oprogramowanie, sprzęt, bazy danych, zasoby finansowe i ludzkie [10]. Aby zapewnić odpowiednie funkcjonowanie tych systemów należy zaopatrzyć je w narzędzia umożliwiające szybki i łatwy odczyt informacji oraz zdolność do przejścia między różnymi formatami danych. Zadaniem GIS, podobnie jak w przypadku map, jest przedstawienie świata rzeczywistego z wykorzystaniem odpowiedniej symbolizacji graficznej i opisu tekstowego.

GIS jest powszechnie wykorzystywany w wielu dziedzinach życia. Wykonywanie analiz sieciowych umożliwia szybkie wyznaczanie optymalnych tras przejazdu, co znacznie upłynnia ruch samochodowy w miastach, a także minimalizuje koszty ponoszone przez firmy transportowe i usługowe. Przykładem aplikacji wspomagającej wybór najszybszej trasy są Mapy Google, które na podstawie dynamicznie analizowanych danych pochodzących od użytkowników na bieżąco aktualizują warunki panujące na trasie. GIS znajduje zastosowanie również w przypadku organizacji imprez masowych, przeprowadzaniu akcji służb mundurowych, turystyce, sporcie, rekreacji czy w logistyce miejskiej.

Zagadnienie wykorzystania Systemów Informacji Geograficznej w odniesieniu do zjawisk transportowych zostało poruszone w artykule Jadwigi Brzuchowskiej [2]. Tematem jej pracy było badanie dostępności komunikacyjnej wybranego obszaru Wrocławia. Autorka przeprowadziła szereg obserwacji opierających się na analizie danych przestrzennych, których wyniki zostały zaprezentowane w formie map zakumulowanego kosztu. W ten sposób wyznaczony został czas dotarcia do miejsca na rastrze względem wskazanego punktu. Umożliwiło to sprawną ocenę wizualną obsługi miejsc zamieszkania wrocławian w odniesieniu do połączeń komunikacyjnych.

Inne zastosowanie Systemów Informacji Geograficznej można znaleźć w pracy Mateusza Malinowskiego i Andrzeja Woźniaka [11]. Przy połączeniu informacji z danych statystycznych oraz przestrzennych autorzy referatu zaproponowali rozwiązanie problemu optymalizacji transportu odpadów komunalnych z warunkiem wykorzystania określonej liczby śmieciarek oraz hakowców w jednym z małopolskich eko-przedsiębiorstw.

1.2. Inteligentne systemy lokalizacyjno-nawigujące

Statystyki w 2020 roku wykazały, że Mapy Google są wykorzystywane przez ponad miliard użytkowników w ciągu miesiąca [34]. Wskazuje to na ogromną skalę popularności oraz potrzeby korzystania z zaawansowanych technik szybkiej lokalizacji i nawigacji.

W Internecie znajduje się wiele portali i aplikacji umożliwiających wyznaczenie odpowiednio uwarunkowanej drogi: przejścia dla turystów górskich (mapa-turystyczna, szlaki.net), połączeń komunikacji miejskiej (jakdoajde, e-podróżnik), przejazdu samochodem (Yanosik, HERE WeGo), a także zapewniających możliwość zapisu statystyk pokonanej trasy (Endomondo, Mi Fit).

Dostosowywanie optymalnych tras może być rozwiązywane poprzez dynamiczną analizę aktualnego stanu ruchu drogowego, bazującą na informacjach pozyskiwanych bezpośrednio od użytkowników aplikacji lokalizacyjnych lub w oparciu o techniki statyczne wykorzystujące między innymi: numeryczne modele wysokościowe, formy pokrycia terenu, dane statystyczne, prawne i ekonomiczne. Połączenie narzędzi GIS z informacjami z innych dziedzin nazywane jest przestrzennym systemem wspomagania decyzji (ang. Spatial Decision Support System, SDSS) [5].

Inteligentne Systemy Transportu (ang. Intelligent Transportation System, ITS) są narzędziami korzystającymi z technik informatycznych, elektroniki pojazdowej oraz komunikacji bezprzewodowej. Ich zadanie polega na informowaniu podróżujących, monitoringu zdarzeń oraz technicznym wspomaganiu i zarządzaniu transportem. Celem zastosowania tego typu inteligentnych systemów jest osiągnięcie poprawy bezpieczeństwa, zwiększenie płynności ruchu drogowego, jak i ograniczenie zużycia paliwa. [6]

Architektura ITS składa się ze struktury „ogólnej, funkcjonalnej, fizycznej i komunikacyjnej”. Pierwsza ze struktur określa główne założenia systemu wraz z zasadami działania. Druga zawiera relacje, zbiory danych oraz funkcje konieczne do spełnienia wymogów użytkowników. W przypadku struktury funkcjonalnej, podejmowana jest charakterystyka wyposażenia technicznego, służącego realizacji zadań w praktyce. Natomiast część komunikacyjna zapewnia środki przesyłu informacji i danych. [6]

Rozwiązania Intelligentnych Systemów Transportu znajdują powszechne zastosowanie w logistyce miejskiej. Ich ważnym elementem jest praktyczna optymalizacja rozwiązań, wiążąca się z potrzebą dogłębnego zapoznania z czynnikami wpływającymi na stopień realizacji zamierzonego efektu.

Opracowanie Aleksandry Koźlak [6] stworzone na podstawie publikacji „Zarządzanie bezpieczeństwem ruchu drogowego w systemie Tristar” [17] pokazuje przewidywaną skalę wpływu ITS w odniesieniu do możliwości zmian badanych cech (Tabela 2).

Tabela 2. Efekty zastosowania inteligentnych systemów transportowych [6][17].

Efekt zastosowania ITS	Rodzaj zastosowanych ITS	Skala efektu [%]
Wzrost przepustowości sieci ulic	systemy zarządzania ruchem na DSR	do 25
	systemy kierowania pojazdów na trasy alternatywne przez znaki o zmiennej treści	do 22
	zastosowanie elektronicznych systemów poboru opłat	200-300
Zmniejszenie strat czasu w sieci ulic	zastosowanie sygnalizacji świetlnej	do 48
	sterowanie ruchem na wjazdach na DSR	do 48
	systemy zarządzania zdarzeniami drogowymi	do 45
	zastosowanie elektronicznych systemów poboru opłat	do 71
	priorytet sygnalizacji świetlnej dla pojazdów transportu zbiorowego (oprócz redukcji strat czasu pozwala na wzrost punktualności do 59%)	do 54
Poprawa bezpieczeństwa ruchu drogowego (zmniejszenie liczby wypadków)	kamery nadzoru prędkości	do 80
	sterowanie ruchem na wjazdach na DSR	do 50
	zaawansowane systemy sterowania ruchem	do 80
	systemy zarządzania zdarzeniami drogowymi	do 50
Poprawa skuteczności służb ratowniczych	zastosowanie systemów zarządzania zdarzeniami drogowymi i służbami ratowniczymi - skrócenie czasu: wykrycia zdarzenia, dojazdu służb ratowniczych do miejsca wypadku	do 66 do 43
	zastosowanie systemów automatycznej lokalizacji pojazdu służb ratowniczych i nawigacji pojazdu do miejsca wypadku – skrócenie czasu dojazdu	do 40
Wpływ na środowisko naturalne	systemy zarządzania popytem - redukcja emisji spalin	do 50
	zarządzanie ruchem na DSR - redukcja zużycia paliwa	do 42
	systemy zarządzania ruchem miejskim - redukcja emisji spalin	do 30

1.3. Problemy implementacji optymalnych rozwiązań

Dostępne na rynku oprogramowanie i aplikacje nastawione są na zaspokajanie najbardziej powszechnych potrzeb. Dla zwykłego użytkownika są one zwykle wystarczające, jednak w odniesieniu do bardziej wymagających jednostek zdarza się, że wynik nie jest możliwy do osiągnięcia z wykorzystaniem prostych metod lub możliwości programów. W ten sposób proces staje się czasochłonny, znacznie bardziej skomplikowany i podatny na wystąpienie błędów.

W przypadku generowania tras, utrudnieniem może być nastawienie na jeden format danych. Obiekty liniowe, takie jak: drogi, ścieżki rowerowe i szlaki turystyczne zwykle zapisywane są w postaci wektora, natomiast ukształtowanie terenu, jako obraz złożony z pikseli zawierających wartości wysokości. Brak możliwości integracji warstw wektorowych z rastrowymi wymaga przejścia między różnymi modułami, programami lub wywołuje potrzebę wprowadzenia zmian ręcznych, z czym wiąże się konieczność poświęcenia czasu na znalezienie i zapoznanie się z funkcjonowaniem nowych narzędzi.

Innym problemem jest ograniczenie możliwości obliczeniowych urządzenia lub brak wystarczającej pamięci na dysku dla zbyt dużej objętości analizowanych danych. Sytuacja wymaga w takim przypadku wprowadzenia stosownej generalizacji lub zawężenia obszaru poszukiwań. Zmniejszenie ilości informacji można uzyskać poprzez zwiększenie rozmiaru pikseli obrazu, filtrację eliminującą dane nieistotne bądź zadanie koniecznych do spełnienia warunków, czego przykładem może być przypisanie wartości wysokości warstwom liniowym wyłącznie w miejscach ich oryginalnych wierzchołków.

Firmy transportowe, oprócz minimalizacji czasu przejazdu, potrzebują również informacji o nachyleniach terenu. Zbyt duży kąt podjazdu może być barierą nie do pokonania dla samochodów ciężarowych o bardzo dużym obciążeniu. Natomiast turyści mogą być zainteresowani informacją o stosunku ilości podejść do zejść na trasie. W przypadku organizacji wyścigów kolarskich każdy odcinek trasy musi spełniać zadane kryteria dotyczące średniego nachylenia, stosunku podjazdów do zjazdów oraz pokonywanych różnic wysokości.

Wiarygodność wyników może zostać zaburzona poprzez wprowadzenie zbyt mało precyzyjnych danych. Analizy z wykorzystaniem wyłącznie rastrów ograniczają dokładność sieci dróg, które są zwykle zbyt wąskie, aby ich przebieg został w całości poprawnie odwzorowany. Innym problemem jest dostosowanie algorytmów do sytuacji w rzeczywistości. W miastach sygnalizacja świetlna może wpłynąć na znaczne wydłużenie czasu przejazdu. Podobnie jest w sytuacji nagłego zakorkowania ulicy spowodowanego remontami, wypadkiem lub inną sytuacją drogową. Na wyniki wpływa również stan jakości nawierzchni, sytuacja demograficzna oraz popularność turystyczna danego regionu. Rozwiązaniem mogą być aplikacje działające na bazie dynamicznie zmieniających się danych, które są pozyskiwane od osób pokonujących określone trasy. Wiąże się to jednak z koniecznością pozyskania zgody na pobieranie informacji oraz pracą na bardzo dużych zbiorach wartości, które często mogą zawierać błędy i fałszować wyniki.

Wyznaczanie czasu potrzebnego do przejazdu drogi między wskazanymi punktami wymaga dostosowania prędkości do sytuacji komunikacyjnej danego obszaru. Artykuł widniejący na portalu oponeo.pl „Ranking miast przyjaznych kierowcom” [33] zestawia ze sobą statystyki dotyczące ruchu drogowego oraz udogodnień dla kierowców (Tabela 3, Tabela 4). Badane miasta zostały podzielone na dwie grupy, powyżej oraz poniżej 300 tysięcy mieszkańców. Na podstawie informacji udostępnionych z urzędów miast, zarządów dróg oraz pomiarów z systemu Yanosik udało się uzyskać średnie prędkości przemieszczania się pojazdów w centrum oraz w pozostałych częściach miasta. Wyniki charakteryzują się dużym zróżnicowaniem dla różnych obszarów. W przypadku prostych aplikacji uzyskane wartości mogą stanowić podstawę do zdefiniowania średniej prędkości ruchu aut.

Tabela 3. Średnie prędkości w centrum miast [33].

> 300 000 MIESZKAŃCÓW		< 300 000 MIESZKAŃCÓW	
NAZWA MIASTA	ŚREDNIA PRĘDKOŚĆ [KM/H]	NAZWA MIASTA	ŚREDNIA PRĘDKOŚĆ [KM/H]
Gdańsk	31	Katowice	35
Lublin	31	Gliwice	31
Warszawa	29	Częstochowa	30
Łódź	27	Sosnowiec	30
Wrocław	25	Toruń	29
Poznań	25	Białystok	27
Szczecin	25	Gdynia	27
Bydgoszcz	25	Kielce	26
Kraków	24	Rzeszów	26
		Radom	25

Tabela 4. Średnie prędkości poza centrum miast [33].

> 300 000 MIESZKAŃCÓW		< 300 000 MIESZKAŃCÓW	
NAZWA MIASTA	ŚREDNIA PRĘDKOŚĆ [KM/H]	NAZWA MIASTA	ŚREDNIA PRĘDKOŚĆ [KM/H]
Szczecin	43	Częstochowa	53
Lublin	42	Gliwice	52
Warszawa	41	Katowice	46
Łódź	41	Radom	44
Poznań	41	Białystok	43
Gdańsk	41	Sosnowiec	43
Bydgoszcz	40	Toruń	42
Kraków	37	Kielce	42
Wrocław	36	Rzeszów	42
		Gdynia	40

2. Automatyzacja w integracji i przetwarzaniu danych

2.1. Modele danych przestrzennych

GIS opiera się na zdefiniowanym kształcie, położeniu, właściwościach i relacjach. Obiekty przedstawiane są za pomocą punktów, linii lub powierzchni. W celu organizacji danych stosowane są określone modele, których główny podział opiera się na zróżnicowaniu pod względem reprezentacji graficznej. W ten sposób wyróżniono model rastrowy, skupiający się na zawartości wewnętrznej oraz wektorowy, skierowany na informacje związane z granicami [16].

2.1.1 Model rastrowy

Raster jest zbiorem wartości w postaci regularnej siatki złożonej z pojedynczych pól, zwanych pikselami [14], które mogą być charakteryzowane przez określoną rozdzielczość, położenie oraz wartość. Podczas przeprowadzania analiz każdej komórce przypisywany jest odpowiedni numer wiersza oraz kolumny w macierzy i w takiej postaci interpretowane są zdjęcia cyfrowe oraz dokumenty, które poddano skanowaniu.

Obrazy charakteryzują się określoną zdolnością do rozpoznawania szczegółów, która nazywana jest rozdzielczością. Wyróżnia się jej trzy podstawowe typy: geometryczna (związana z wymiarami lub liczbą pikseli w wierszach i kolumnach), radiometryczna (definiująca zakres skali wartości zapisu) i spektralna (zależna od liczby kanałów spektralnych). Im rozdzielczość jest większa, tym obraz można przechowywać bardziej precyzyjne i obszerne dane. Liniowa zmiana rozdzielczości powoduje jednak kwadratowy wzrost objętości plików oraz czasu wymaganego do ich przetwarzania [18]. Przy działaniach na rastrach dobierana jest również odpowiednia liczba kolorów reprezentacyjnych. Do najczęściej stosowanych formatów można zaliczyć: binarny (1 bpp, piksele przyjmują wyłącznie jedną z dwóch wartości), monochromatyczny (8 bpp, obrazy w 256 stopniowej skali szarości), kolorowy (24, 32 bpp, umożliwia przedstawienie barw RGB, CMY) [18]. Stosunek częstości występowania każdego z poziomów jasności jest przedstawiany za pomocą histogramu.

W przypadku przeprowadzania analizy obrazów wykorzystywane są zdefiniowane modele barw. Jednym z najpopularniejszych przykładów jest RGB, który opiera się na sześcienniej reprezentacji palety kolorów. Jest addytywny, co oznacza, że barwy pośrednie uzyskiwane są w procesie przestrzennego sumowania strumieni światła o barwach podstawowych: czerwony, zielony, niebieski. Do celów przetwarzania komputerowego stworzony został model HSV, który bazuje na zdefiniowaniu trzech składowych: odcienia (ang. Hue), nasycenia (ang. Saturation) i wartości (ang. Value) [18]. Natomiast CMY jest subtrakcyjny i opiera się na odejmowaniu (pochłanianiu) mieszanych kolorów: C (turkus, ang. Cyan), M (purpura, ang. Magenta), Y (żółty, ang. Yellow) [12].

Modele rastrowe najlepiej pokazują obiekty o charakterze ciągłym, których granice są nieprecyzyjne bądź nie ma możliwości ich określenia. Mogą być to obszary, gdzie odwzorowywane są właściwości zjawisk: temperatura, ilość opadów, zanieczyszczenie powietrza lub ilustrowana jest zmienność cech, na przykład: położenie nad powierzchnią morza, ilość biomasy.

Zaletą rastrowego sposobu zapisu informacji jest przejrzysta struktura oraz możliwość prowadzenia analiz zróżnicowania w obrębie jednego obiektu. Poza tym, możliwe są do wykonywania działania typowe dla struktur tablicowych: klasyfikacja, interpolacja, filtracja, operacje logiczne na wielu zbiorach danych, analizy numerycznego modelu terenu [4]. Wadą jest duża objętość obrazów wysokorozdzielczych oraz ograniczenie liczby atrybutów przyporządkowanych do danego piksela.

2.1.2 Model wektorowy

Modele wektorowe mają zastosowanie głównie przy opisywaniu obiektów dyskretnych, zatem potrzebują precyzyjnej informacji o konturach i kształcie. Elementy są samodzielne i przedstawiane z użyciem wektorów: 0-wymiarowych (punkty), 1-wymiarowych (linie), 2-wymiarowych (powierzchnie), a także sposobów prezentujących trzeci (objętość) lub czwarty (czas) wymiar. Wektor stanowi cyfrowy zapis przestrzeni, gdzie lokalizacja jest wyznaczana poprzez zdefiniowanie pary współrzędnych geograficznych bądź płaskich prostokątnych. Wszystkie obiekty tego typu są możliwe do opisu pod względem charakterystyki jakościowej oraz ilościowej. Wartości cech atrybutów umieszczane są w tabeli pod odpowiednio oznaczonymi kolumnami, gdzie pierwsza z nich powinna zawierać unikatowy identyfikator (ID). [14]

Przy opisywaniu form ciągłych wykorzystywany jest model TIN (ang. Triangulated Irregular Network), który powstaje w wyniku podzielenia przestrzeni na trójkąty o różnych wymiarach. Wynikowa siatka składa się z węzłów o określonych współrzędnych (x, y, z) a jej elementy są ze sobą powiązane topologicznie. TIN znajduje wykorzystanie przy tworzeniu numerycznych modeli terenu lub projektowaniu obiektów 3D. [14]

Miejscom zastosowania modelu wektorowego mogą być wizualizacje położenia w przestrzeni: budynków, dróg, elementów przyrody ożywionej lub nieożywionej, a także analizy dostępności komunikacyjnej połączeń sieci pojazdów lub szlaków turystycznych. Odzworowywanie danych za pomocą wektora może zostać wykonane z bardzo dużą dokładnością przy zwartym formacie zapisu. Dodatkowo istnieje teoretyczna możliwość dołączania nieograniczonej liczby atrybutów do jednego obiektu.

2.2. Cyfrowe przetwarzanie obrazów

W poprzednim podrozdziale zaprezentowano dwa sposoby reprezentacji danych: wektorowy i rastrowy. Przetwarzanie tych danych opiera się na: pozyskiwaniu, wstępnej obróbce (filtracji, wzmacnianiu kontrastu, binaryzacji), segmentacji, wyodrębnieniu elementów, analizie informacji, rozpoznawaniu. Celem przetwarzania jest uzyskanie przydatnych informacji o obiektach lub ich otoczeniu z wykorzystaniem pewnych algorytmów. Aby efektywnie wykorzystać informacje z obrazu konieczne jest zapisanie go w postaci cyfrowej. [18]

Przetwarzanie obrazów opiera się głównie na opracowywaniu i wdrażaniu odpowiednich algorytmów, spośród których można wyodrębnić pięć podstawowych grup.

Pierwszym rodzajem są przekształcenia geometryczne, bazujące na obrotach, przesunięciach, odbiciach oraz innych transformacjach geometrii. Może być to także skalowanie wykonywane globalnie bądź z uwzględnieniem przebiegu wyznaczonych osi. Zmiany geometrii często służą niwelacji zniekształceń wywołanych brakiem doskonałości układu optycznego narzędzi pomiarowych. [18]

Drugim przykładem są przekształcenia punktowe. Należą do nich: przyciemnienie, rozjaśnienie, negatyw, sepia. Przekształcenia te opierają się na pojedynczych elementach obrazu (punktach), które są niezależnie od siebie modyfikowane. Tego typu operacje są zwykle stosunkowo proste i szybkie oraz przeprowadzane z wykorzystywaniem LUT (ang. Look Up Tables), czyli w oparciu o przygotowane tablice korekcji. Wynikiem działań jest zmiana skali jasności bez wpływu na geometrię obrazu. Dla poprawy jakości i czytelności informacji z rastrów wykorzystywane są również takie operacje jak: normalizacja (sprowadzająca wartości wejściowe do ustalonego zakresu), modulacja gamma (zmniejszająca nadmierny kontrast za pomocą funkcji o stałym wykładniku potęgi), wyrównanie histogramu, binaryzacja (przypisanie pikselom wyłącznie wartości 0 lub 1) oraz przeliczenia z zastosowaniem wyrażeń algebraicznych. [18]

Kolejnym rodzajem przetwarzania obrazów są przekształcenia kontekstowe, których nazwa oznacza, że zmiany dla jednego punktu będą opierać się nie tylko na nim samym, ale również na analizie wielu innych pochodzących z tego samego obrazu źródłowego. Operacje te wykorzystują filtry zmieniające wyznaczone elementy w zależności od ich stanu lub otoczenia. Często wiąże się to z długim czasem obliczeń, jednak algorytmy nie zawsze są skomplikowane i wymagające. Mogą być one również proste i regularne oraz wykonywane na wszystkich punktach jednocześnie. Problemem przekształceń kontekstowych może być przeprowadzenie działań na brzegach rastrów, gdzie pojawia się brak wartości bocznych. Dużą zaletą jest jednak możliwość pozbycia się zbędnych obiektów, czego przykładem jest stłumienie losowych szumów. [18]

Przekształcenia widmowe opierają się na działaniach dla całych obrazów. Pierwszym etapem jest obliczenie dwuwymiarowego widma (transformacja Fouriera), które dalej poddawane jest pewnym operacjom (na przykład usuwaniu składowych niespełniających założonych warunków) i rekonstrukcji (odwrotna transformacja Fouriera). Procesy te są bardzo precyzyjne, jednak wymagają czasu i stanowią duże obciążenie dla pamięci RAM (odpowiadającej za wydajność i szybkość działania systemu i programów) komputera. [18]

Ostatnim przykładem są przekształcenia morfologiczne. W tym przypadku określony element (wycinek obrazu) nie musi być zawsze modyfikowany, decyzja o zmianach jest uzależniona od spełnienia zadanego warunku logicznego. Pozwala to na wykonywanie najbardziej złożonych i efektywnych operacji, które mogą brać pod uwagę analizę kształtu, czy wzajemnego położenia. Każde przekształcenie wymaga przyłożenia centralnego punktu (wyróżnionego w bazowym elemencie strukturalnym) do wszystkich punktów obrazu, porównaniu lokalnej konfiguracji i podjęciu decyzji wykonania operacji lub jej pominięcia. Do przykładowych przekształceń morfologicznych należy erozja. Zakłada ona, że istnieje pewien obszar X oraz koło o danym promieniu R , które stanowi element strukturalny. Proces definiowany może być w następujący sposób: „Koło B przetacza się po wewnętrznej stronie brzegu figury. Kolejne położenia środka koła B wyznaczają brzeg figury zerodowanej”. W ten sposób w pojęciu implementacji komputerowej wszystkie punkty posiadające jakiegokolwiek sąsiada o wartości 0 są usuwane. Działanie przypomina filtr minimalny, przypisujący punktom minimum z dostępnych wartości. Innymi typowymi przekształceniami morfologicznymi są także: dylatacja (działanie odwrotne do erozji), otwarcie i zamknięcie (eliminujące wadę zwiększania lub zmniejszania powierzchni, która występuje przy wykonywaniu erozji lub dylatacji), detekcja ekstremów (minimów i maksimów), szkieletyzacja (umożliwiająca wyodrębnienie osi obiektów). [18]

Istnienie tak szerokiego wyboru algorytmów przekształcających wiąże się z koniecznością testowania i wyboru najlepszego z dostępnych rozwiązań. Głównym celem omawianych w tym podrozdziale operacji jest: ekstrakcja cech, opracowanie statystyk liczbowych lub pomiar wartości. Komputerowa analiza obrazów znajduje wykorzystanie w takich dziedzinach jak: fotogrametria (detekcja znaczków pomiarowych), teledetekcja (klasyfikacja form użytkowania terenu), meteorologia (przewidywanie pogody na podstawie zachowania się chmur), medycyna (automatyczne rozpoznawanie wad postawy), bezpieczeństwo publiczne (wykrywanie obecności ludzi w miejscach z zakazem wstępu).

2.3. Wprowadzenie do języka Python

Postęp technologiczny wymaga od ludzi znajdowania coraz bardziej efektywnych rozwiązań opartych na automatyzacji działań. Wspomaganie komputerowe procesów jest możliwe poprzez wykorzystywanie: programów (samodzielnie działających aplikacji w określonym systemie operacyjnym), skryptów (plików zawierających tekst z kodem wymagających interpretera), wtyczek (zestawu plików uruchamianych w środowisku danej aplikacji) lub bibliotek (zbiorów funkcji) [20].

Środowisko Python służy implementacji zarówno prostych i szybkich prototypów, jak i skomplikowanych, wielomodułowych rozwiązań. Posiada dostęp do bibliotek, które są wyposażone w wydajne funkcje matematyczno-statystyczno-analityczne. Jest to język wysokiego poziomu, który cechuje się: brakiem konieczności kompilacji przed uruchomieniem, zorientowaniem obiektowym, uniwersalnością w zastosowaniu, nastawieniem na intuicyjność oraz niewielką ilością zapisywanego kodu względem wykonywanych działań [3].

Historia języka Python sięga końca lat osiemdziesiątych XX wieku. Jego twórcą jest holenderski programista Guido van Rossum [3]. Język ten stał się szybko powszechnie znany i wykorzystywany. Według światowych rankingów z 2020 roku sprawdzających możliwości znalezienia pracy, zastosowanie, popularność i rozwój, Python widnieje w czołówce najlepszych języków programowania [25][22]. Z wykorzystaniem jego środowiska funkcjonują takie serwisy i aplikacje jak: Google, YouTube, Instagram, Dropbox, ESRI, Netflix [9].

Istnieje wiele powodów dla czego ludzie chętnie korzystają z języka Python. Temat ten został poruszony również przez jednego z najbardziej znanych i docenianych instruktorów tego języka Marka Lutz'a. Na podstawie jego książki „Learning Python” [9] przytoczono kilka przykładów największych zalet Python'a:

- jakość oprogramowania (czytelność, logiczność, wielokrotność zastosowania, utrzymywalność, jednolitość, prostota w zrozumieniu, skierowanie na programowanie obiektowe i funkcyjne),
- wydajność (nastawienie na krótki i zwięzły kod, szybkość uruchamiania),
- przenośność programu (niezmienne działanie na różnych platformach komputerowych, dostosowanie do potrzeb przenoszenia interfejsu, dostępu do bazy danych, systemów webowych),
- biblioteki wspomagające (biblioteka standardowa zawiera duży zasób funkcji wspomagających tworzenie skryptów i aplikacji, istnieje również dostęp do innych zbiorów, takich jak: NumPy, Pandas, Matplotlib, Django, Flask),
- integracja komponentów (łatwe powiązanie i komunikacja osobnych części aplikacji, możliwość odwoływania się i integracji z innymi językami programowania),
- przyjemność (łatwość w użyciu i rozumieniu, wbudowane zestawy narzędzi).

Python jest językiem szczególnie popularnym i docenianym w przypadku eksploracji danych, statystyki, uczenia maszynowego oraz celów predykcyjnych. Praktycznym przykładem wykorzystania może być analiza preferencji zakupowych potencjalnych klientów, przewidywanie zachowań giełdowych lub dostosowanie ceny produktów do wskaźnika popytu. Sprawna i wydajna praca jest możliwa poprzez wykorzystanie wbudowanych metod należących do określonych zbiorów funkcji nazywanych bibliotekami.

Poniżej wykonano krótką charakterystykę bibliotek wykorzystanych przy realizacji praktycznej części pracy dyplomowej związanej z budową wtyczki oraz analizą generowanych przez nią wyników.

Math, to moduł wyposażający użytkownika w podstawowe działania matematyczne. Po jego zaimportowaniu za pomocą komendy „import math” możliwe jest korzystanie z funkcji takich jak: pow (potęgowanie), sqrt (pierwiastkowanie), sin, cos, tan (trygonometria), degrees, radians (zamiana jednostek kątowych), pi, tau (wprowadzanie wartości stałych), ceil, floor (zaokrąglanie liczb), log (obsługa logarytmów). [28]

OSGeo jest zbiorem modułów (pakietem) ułatwiającym import, eksport oraz odczyt danych w różnych formatach. Wyposażony jest w algorytmy przydatne podczas analiz w zakresie GIS, generowania prostych wykresów oraz działaniach na atrybutach warstw rastrowych. Polecenie GetRasterBand daje możliwość odczytu wartości pikseli obrazów,

GetProjection zwraca rodzaj odwzorowania, natomiast zastosowanie przedrostka „set” pozwala na zdefiniowanie i wprowadzenie własnych właściwości. [30]

NumPy (ang. Numerical Python), to podstawowy pakiet dla osób realizujących projekty związane z przetwarzaniem danych. Jest wyposażony w narzędzia umożliwiające wykonywanie zaawansowanych analiz oraz przeprowadzanie szybkich obliczeń na tablicach. Algorytmy są nastawione na zaopatrzenie użytkownika w łatwy dostęp do gotowych rozwiązań z dziedziny algebry, transformacji, generowania losowych elementów. Możliwe jest sprawne sortowanie oraz filtracja danych. [13]

Biblioteka PyQt5 służy obsłudze interfejsów graficznych, które tworzą obudowę dla implementowanych rozwiązań. Zawarte w niej metody pozwalają na kierowanie procesem programu przez zewnętrznego użytkownika. Dodatkowo algorytmy dają możliwość zarządzania sygnałami i slotami, dzięki czemu emitowana jest odpowiednia odpowiedź na określone czynności, takie jak kliknięcie, wciśnięcie klawiszy klawiatury, śledzenie położenia kursora myszy. Budowanie wyglądu interfejsu ułatwione jest poprzez program Qt Designer, która umożliwia wizualne dodawanie obiektów QtWidgets, na przykład: QCheckBox, QComboBox, QDialog. [31][27]

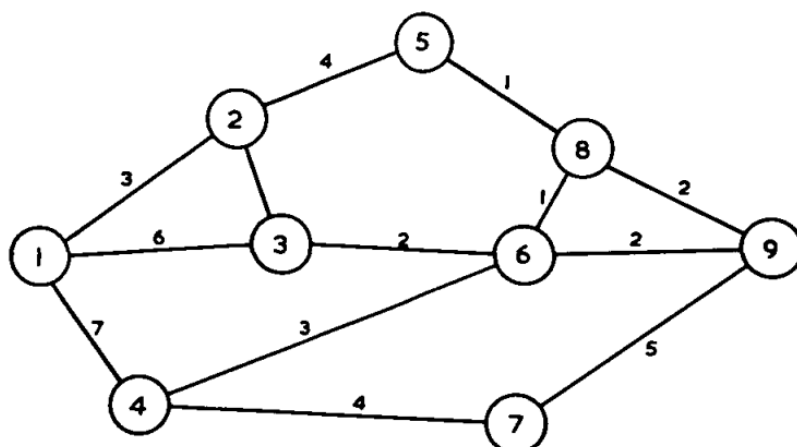
Pakiet QGIS zaopatrzony w moduły z grupy qgs pozwala między innymi na: obsługę warstw wektorowych oraz rastrowych, zarządzanie projektami w programie QGIS, czy również zapis obiektów w określonym formacie z nadaniem wskazanych atrybutów. Służy łatwieszemu tworzeniu wtyczek i automatyzacji procesów w zakresie działania oprogramowania Quantum GIS. [32]

2.4. Algorytmy wyznaczania tras

Sieć jest jedną z form organizacji relacji występujących między danymi dyskretnymi. W matematycznym ujęciu może być zdefiniowana jako „graf”. Struktura ta składa się z zestawu punktów, które posiadają określone odległości oraz wzajemne powiązania. Przykładami prostej formy sieci są: drzewo, pierścień, regularna siatka kwadratów i trójkątów, natomiast postać złożona charakteryzuje się pewną losowością budowy i powiązań. Przykładami skomplikowanych relacji mogą być: systemy komunikacji transportowej, system nerwowy, transakcje finansowe, media społecznościowe, oddziaływania kulturowe. W trakcie analiz uwzględniany jest także kierunek działania, który decyduje o pozytywnym bądź negatywnym znaku wartości podczas przejścia między wierzchołkami. [21]

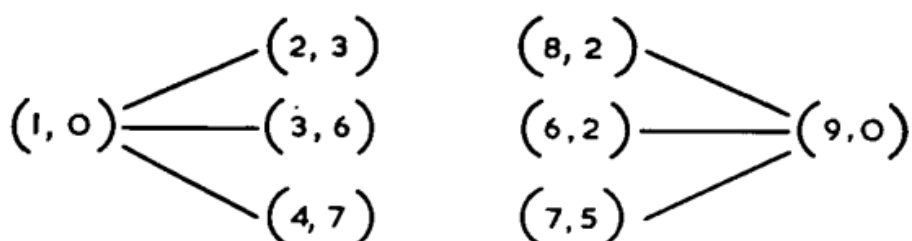
Jednym ze sposobów zastosowania sieci jest wyznaczanie optymalnej trasy między zadeklarowanymi punktami. Podstawowe rozwiązania skupiają się na badaniu połączeń oraz sprawdzaniu warunków między dwoma kolejnymi wierzchołkami. Reprezentatywnym przykładem jest wyszukiwanie najkrótszej drogi.

W jednej ze swoich publikacji T. A. J. Nicholson [15] przedstawił podstawowe problemy związane z zagadnieniem matematyczno-programistycznych rozwiązań wykorzystywanych w trakcie przeprowadzania analiz sieciowych. Opracowana przez niego procedura opiera się na jednoczesnym badaniu wszystkich możliwości przejść trasy od miejsca początkowego, poprzez kolejno związane ze sobą wierzchołki, aż do osiągnięcia punktu końcowego. Proces porównuje w międzyczasie długość odcinków, wybierając najkrótsze dla danych par punktów. Sprawdzana jest również kompletność i brak powtarzalności wierzchołków. Graficzna interpretacja jest przedstawiona z wykorzystaniem przykładowej sieci składającej się z dziewięciu punktów (Rysunek 1).



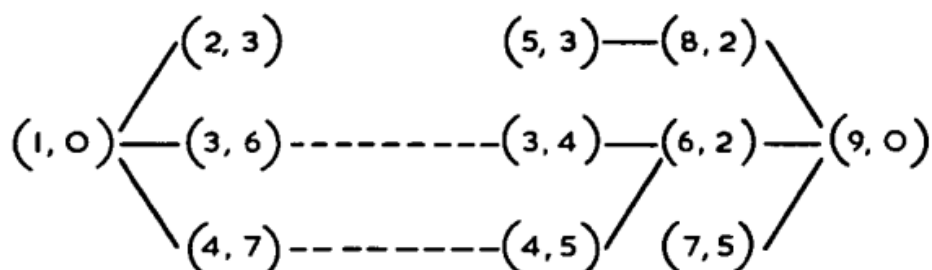
Rysunek 1. Schematu rozkładu punktów i odległości w analizowanej sieci
(źródło: [15]).

Pierwszym krokiem jest wyznaczenie dróg między punktem 1 a 9 w postaci (r,x) , gdzie r – punkt przylegający, x – odległość (Rysunek 2).



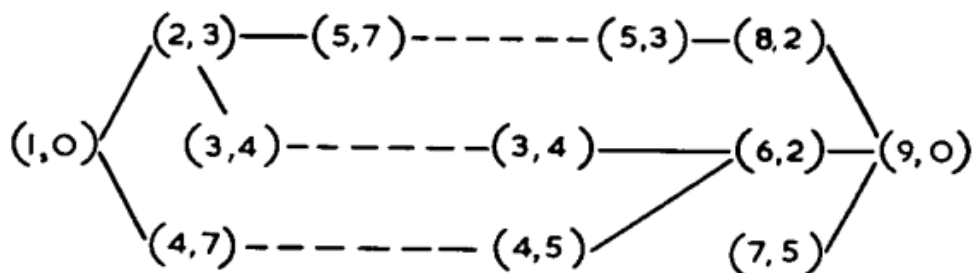
Rysunek 2. Wyznaczenie połączeń dla pierwszego stopnia powiązania wierzchołków
(źródło: [15]).

Następnie po wyselekcjonowaniu sześciu dróg badane są dalsze połączenia dla kolejnych elementów (Rysunek 3). W międzyczasie sprawdzane są również warunki, które mogą wyeliminować dany punkt spod dalszych analiz, czego przykładem jest brak uwzględnienia połączenia między wierzchołkiem 6 i 8 ze względu na warunek minimalizacji odległości.



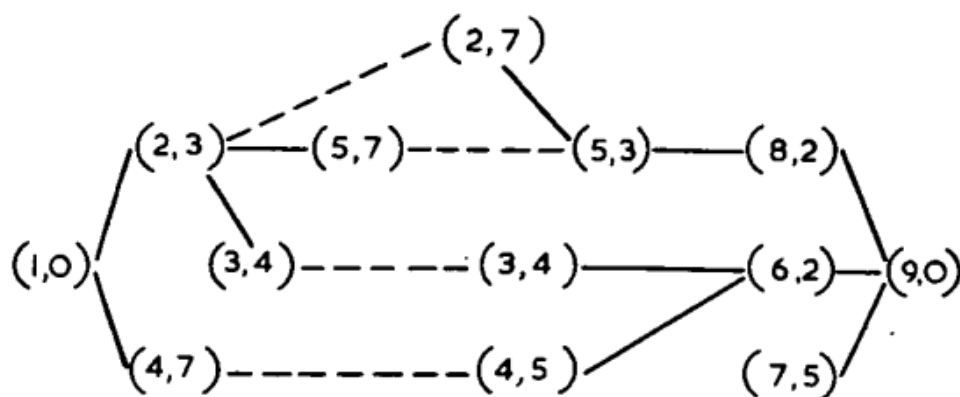
Rysunek 3. Zbadanie drugiego stopnia połączeń
(źródło: [15]).

W trzecim etapie następuje wyodrębnienie i porównanie kolejnych możliwości przebiegu dróg (Rysunek 4).



Rysunek 4. Wyodrębnienie powiązań umożliwiających złączenie się gałęzi z pierwszego stopnia połączeń (źródło: [15]).

Ostatni krok opiera się na wyselekcjonowaniu ostatecznych i kompletnych tras, z których wybierana jest najbardziej optymalna (Rysunek 5). W tym przypadku najkrótsza droga przechodzi przez punkty: 1, 2, 3, 6, 9 a jej odległość wynosi 8.



Rysunek 5. Wyodrębnienie ostatecznej wersji możliwych tras (źródło: [15]).

Innym aspektem jest wyznaczenie czasu potrzebnego do przejścia danej trasy. Do najbardziej popularnych sposobów należy metoda Naismith'a oraz Tobler'a (Rysunek 6). Oryginalna postać pierwszej z formuł zakłada, że człowiek w czasie godziny pokona 5 km, a na każde 600 m przewyższenia przypisywana jest kolejna godzina (1). Po uwzględnieniu poprawki Langmuir'a pojawia się powiązanie nie tylko z długością drogi, ale także z wartością spadku. Gdy pochylenie jest większe niż 12° średnia prędkość przejścia jest zmniejszana, powodując przyrost 10 min na każde 300 m. Odwrotna sytuacja następuje w przypadku zejścia dla zakresu $5-12^\circ$ (2). Korekcja Tranter'a dodatkowo uwzględnia indywidualny stan kondycji danej osoby [8]. Druga z analizowanych formuł wyznaczających czas przejścia zadanej trasy (metoda Tobler'a) przyjmuje postać funkcji wykładniczej. Opiera się na relacjach prędkości, różnicy wysokości oraz odległości poziomej (3).

Wzór dla metody Naismith'a:

$$\text{Dla } |S| > 12^\circ : t = \frac{ds}{5} + \left| \frac{dh}{10} \right| \quad \text{Dla } |S| \leq 12^\circ : t = \frac{ds}{5}$$

(1) [8]

gdzie:

S – spadek [°]

t – czas [min]

ds – odległość rzeczywista [m]

dh – przewyższenie [m]

Wzór dla metody Naismith'a z uwzględnieniem poprawki Langmuir'a:

$$\text{Dla } |S| > 12^\circ : t = \frac{ds}{5} + \left| \frac{dh}{30} \right| \quad \text{Dla } 5^\circ \leq S \leq 12^\circ : t = \frac{ds}{5} - \left| \frac{dh}{30} \right|$$

(2) [8]

gdzie:

S – spadek [°]

t – czas [min]

ds – odległość rzeczywista [m]

dh – przewyższenie [m]

Wzór dla metody Tobler'a:

$$V = 6e^{-3.5 \left| \frac{dh}{dx} + 0.05 \right|}$$

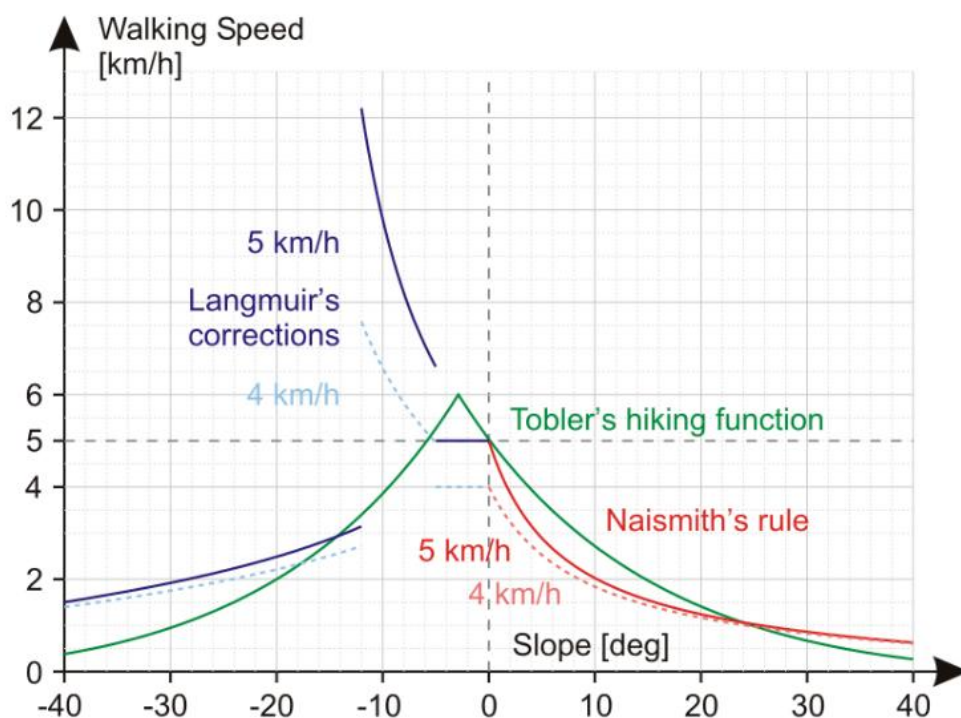
(3) [19]

gdzie:

V – prędkość chodu [km/h]

dh – przewyższenie

dx – odległość pozioma



Rysunek 6. Wykres zmian wartości prędkości dla reguł Naismith'a, Langmuir'a, Tobler'a. (źródło: [29]).

3. Metodyka i dane

3.1. Założenia i technologie pracy

Głównym celem badania jest zbudowanie wielofunkcyjnej aplikacji umożliwiającej szybkie wyznaczenie trasy zoptymalizowanej względem zestawu zadanych warunków. Ważnym aspektem jest także sprawdzenie jej wiarygodności z wartościami pozyskanymi z pomiarów w rzeczywistości, a także określenie wpływu metody interpolacji i zmiany rozdzielczości numerycznego modelu terenu na wielkość generowanych atrybutów. Wykonana zostanie również analiza źródeł prawdopodobnych przyczyn wystąpienia różnic bądź błędów. Wyniki będą podstawą do sformułowania spostrzeżeń dotyczących wpływu oraz sposobu wykorzystania badania w odniesieniu do zagadnień z dziedziny logistyki i transportu. Praca może stanowić podstawę do konstrukcji znacznie bardziej profesjonalnej aplikacji o specjalistycznym przeznaczeniu.

Pierwszy etap będzie polegał na opracowaniu zestawu funkcji, który umożliwi odpowiednie działanie algorytmów programistycznych w środowisku QGIS. Wymaga to głębokiego zapoznania się ze schematami działania, komendami oraz różnymi bibliotekami dostępnymi dla języka Python. Szczególnie istotne są moduły obsługujące działania matematyczne, pracę na interfejsie graficznym oraz te, które umożliwiają odczyt informacji z plików w różnych formatach lub pozwalają na wizualizację i zapis wyników. W celu poprawnego i szybkiego powiązania zawartości gotowego skryptu z programem QGIS rozwiązaniem może być pobranie i uruchomienie odpowiednich wtyczek. Przykładem ogólnodostępnego narzędzia umożliwiającego automatyczne połączenie przygotowanego kodu z interfejsem programu jest Plugin Builder.

Zaimplementowane rozwiązania wymagają przemyślenia sposobu kontroli poprawności. Może być to wykonane z wykorzystaniem danych próbkowych o małej objętości, ale uwzględniających jak największe zróżnicowanie oraz ewentualne wystąpienie błędów. Testy muszą charakteryzować się uniwersalnością oraz łatwością porównywania wyników. Dużym wyzwaniem jest uwzględnienie jak największej liczby przypadków, które mogą się zdarzyć. Dane mogą być niepełne, w nieodpowiednim formacie, wybiegać poza obszar opracowania lub zawierać błędy topologiczne. Innym problemem są uchybienia ze strony użytkownika. Może on nieumyślnie wprowadzić błędną ścieżkę zapisu lub odczytu plików, a także pomylić nazwy warstw bądź opuścić pola wymagające uzupełnienia. Wszystkie niedopatrzenia i błędy wymagają zabezpieczenia. Jedną z możliwości jest wysyłanie wizualnych komunikatów w postaci sprecyzowanych informacji o napotkanych problemach.

W celu analizy dokładności generowanych wyników należy wykonać porównanie z danymi otrzymanymi w wyniku pomiarów w rzeczywistości lub w odniesieniu do istniejących już rozwiązań. Pozyskanie wartości porównawczych może być wsparte aplikacjami mierzącymi czas i drogę wraz z zaznaczeniem jej przebiegu. Podczas kontroli uwzględnieniu musi podlegać odpowiedni typ lokomocji: pieszo, samochodem, rowerem. Ważnym elementem jest zwrócenie uwagi na bariery i utrudnienia, takie jak: sygnalizacja świetlna, roboty drogowe, korki. Mogą stanowić one przyczynę wystąpienia znacznych różnic między wartością kontrolną a kontrolowaną. Dzień tygodnia oraz pora dnia ma duży wpływ na ruch. Najbardziej zatłoczone są godziny poranne 7.30-10.00 oraz popołudniowe 15.00-17.30. Pomiary wykonane w różnym momencie będą się znacząco różnić, dlatego należy stosownie dopasować czas do aktualnych potrzeb. Dodatkowo powinna zostać uwzględniona ogólna sytuacja komunikacyjna rozważanego obszaru. Średnie prędkości jazdy samochodem są różne nie tylko w obrębie osobnych miast, ale także między ich centrami i obrzeżami (Tabela 3, Tabela 4).

Zmiana rozdzielczości numerycznego modelu terenu może wywołać jednocześnie pozytywny, jaki i negatywny wpływ. W sytuacji zwiększenia rozmiaru piksela pierwotna informacja staje się mniej precyzyjna, ale z drugiej strony możliwe jest wykonywanie znacznie szybszych obliczeń. Natomiast w przypadku mniejszych komórek dane są w stanie dokładniej odwzorować powierzchnię, dzięki czemu profil terenu może być bardziej zbliżony do rzeczywistej sytuacji. Problemem rastrow o większej rozdzielczości jest wydłużenie czasu generowania wyników lub brak wystarczającej pamięci do przeprowadzenia uruchomionych procesów. Rozwiązaniem może być zapis relacji piksel-wektor w postaci tablicowej, ograniczenie obszaru analiz lub właściwe dopasowanie rozdzielczości obrazu do aktualnych potrzeb. Przy wykonywaniu resamplingu, pojawia się konieczność ustalenia granicy dopuszczalnych rozbieżności. Wiąże się to z częściową i nieodwracalną utratą informacji, co może prowadzić do istotnych zmian w generowanych wartościach.

Wyniki badań zostaną przedstawione w formie opisowej, tabelarycznej oraz z wykorzystaniem wykresów. Połączenie różnych metod prezentacji ułatwi zrozumienie relacji występujących między określonymi danymi. Na koniec sformułowane zostanie podsumowanie zawierające najważniejsze spostrzeżenia dotyczące poprawności działania aplikacji oraz przeprowadzonych analiz.

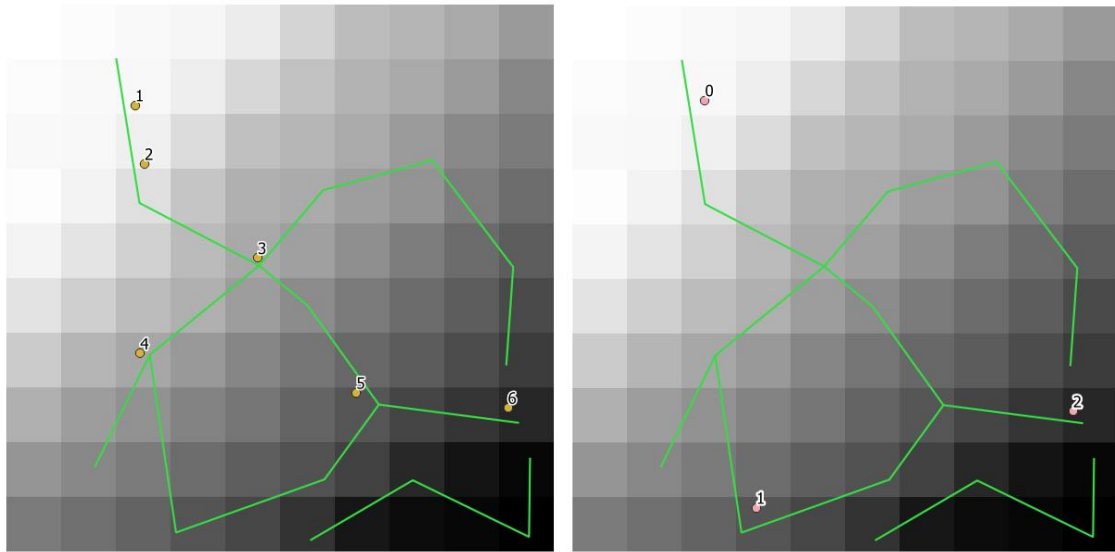
3.2. Charakterystyka warstw rastrowych i wektorowych

Do badań zostały wykorzystane dwa zestawy danych rastrowo-wektorowych, dzielące się na zbiór: testowy i kontrolowany. Zestaw testowy służył kontroli poprawności zapisu algorytmów. Składał się na: niewielki wycinek rastra o wielkości 10x10 pikseli i komórce 1x1 m, przykładowej sieci dróg oraz kilku zbiorów różnie rozmieszczonych punktów decydujących o kolejności przebiegu trasy (Rysunek 7). W tym przypadku czas generowania wyników przez aplikację był bardzo szybki i wynosił mniej niż 1 s. Zbiór kontrolowany obejmował: wysokorozdzielczy raster 1x1 m dla obszaru o powierzchni ponad 6.5 km², który pokrywał rejon pomiarów terenowych (Rysunek 8). Czas dla obliczeń wynosił znacznie więcej, około 30 s.

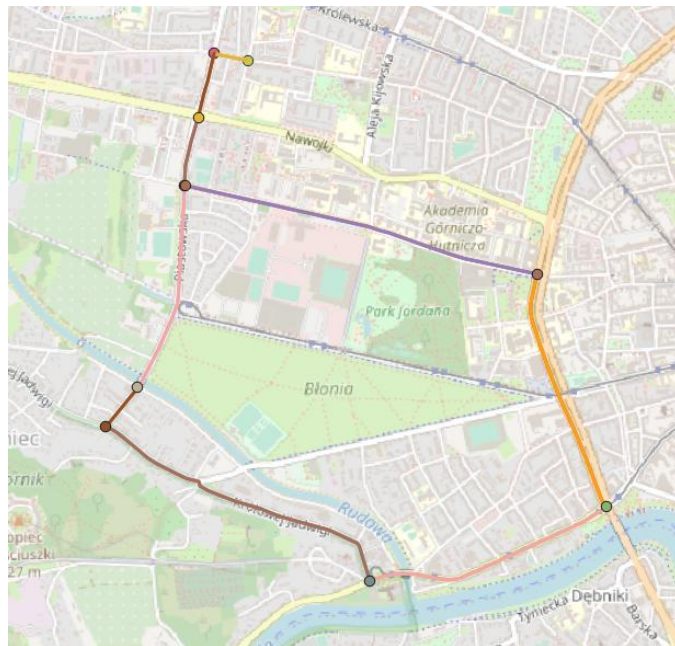
Numeryczny model terenu został pozyskany w wyniku filtracji i przetworzenia danych LIDAR (ang. Light Detection and Ranging) pochodzących z lotniczego skaningu laserowego wykonanego nad obszarem Krakowa w ramach projektu ISOK (Informatycznego Systemu Osłony Kraju). Dane zostały pozyskane w standardzie II, gęstość wynosiła co najmniej 12 pkt/m² [7]. Na podstawie punktów należących do powierzchni terenu wygenerowano model GRID o siatce 1x1 m. Następnie w wyniku zastosowania resamplingu utworzono także dodatkowe warstwy o mniejszej rozdzielczości pikseli: 3x3, 5x5, 10x10, 20x20, 30x30, 50x50, 70x70, 100x100 m (Rysunek 9).

Liniowe warstwy wektorowe w przypadku zestawu testowego zostały utworzone ręcznie, w wyniku losowego wrysowania. Składały się na 4 polilinie, które przecinały się ze sobą w 3 miejscach, stanowiących jedyne obszary możliwego przeskoczenia trasy z pierwotnego na inny obiekt. Drugi zbiór powstał w wyniku wektoryzacji dróg na podstawie danych z Open Street Map, gdzie odwzorowana została przybliżona trasa przebiegu odcinków z pomiaru terenowego wykonywanego przy wykorzystaniu aplikacji Endomondo.

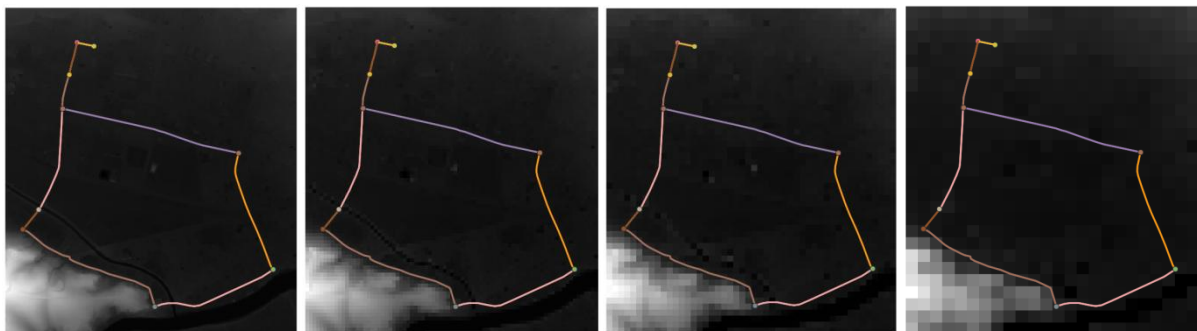
Utworzono kilka zestawów punktów wyznaczających kolejność trasy. Umożliwiło to sprawdzenie dużego zakresu możliwości jej przebiegu. Zbiory obejmowały od 2 do 7 punktów ulokowanych w różnych miejscach. Uwzględniano przykładowe położenia punktów: w środku i na krańcach tras, blisko miejsc przecięć, poza obszarem rastra, najbliższej odłączonego obiektu liniowego, między jednym bądź między różnymi obiektami. W przypadku analizy danych kontrolowanych punkty były lokowane zgodnie z miejscami każdego z odcinków pomiarowych, tak aby istniała możliwość wykonania stosownego porównania wartości wynikowych z aplikacji oraz otrzymanych w rzeczywistości.



Rysunek 7. Przykładowy zestaw danych testowych
(źródło: opracowanie własne).



Rysunek 8. Przykładowy zestaw danych kontrolowanych
(źródło: opracowanie własne).



Rysunek 9. Przykładowe rastry o rozdzielczościach: 10x10, 30x30, 50x50, 100x100 m
(źródło: opracowanie własne).

3.3. Oprogramowanie i aplikacje

W projekcie korzystano z oprogramowania QGIS Desktop w wersji 3.12.0, które jest darmowym i otwartym oprogramowaniem działającym na wielu platformach (Windows, Mac Os, Linux, Android, Unix) [23]. Jest to narzędzie umożliwiające edycję, eksplorację, analizy oraz integrację danych przestrzennych. QGIS umożliwia odczyt kilkuset formatów wektorowych i rastrowych, a także zapewnia połączenie z bazami danych oraz usługami on-line [23]. Narzędzie to znajduje szczególne zastosowanie przy wykonywaniu różnego rodzaju inwentaryzacji (przewodów wodociągowych, dróg, terenów zieleni miejskiej) oraz tworzeniu map. Powyższe oprogramowanie w trakcie badań posłużyło jako obszar implementacji nowo zaprojektowanej wtyczki, jak również pozwoliło na wykonanie niezbędnych procesów związanych z obróbką obrazów, wizualizacją danych, czy utworzeniem warstw testowych i kontrolowanych. Możliwość korzystania z wbudowanej konsoli dla języka Python ułatwia rozbudowę istniejących funkcji oraz wprowadzania własnych rozwiązań. QGIS posiada również ciągle poszerzający się zbiór ogólnodostępnych wtyczek, które można pobrać na potrzeby danego projektu.

Pozyskiwanie danych porównawczych wymagało zastosowania narzędzi do zapisu czasu oraz drogi pokonywanej osobno dla odpowiednich odcinków. Taką funkcjonalnością charakteryzuje się aplikacja Endomondo, która oprócz zapamiętywania informacji liczbowych ma możliwość generowania graficznego przedstawienia przebiegu trasy. Aplikacja jest przeznaczona głównie dla osób uprawiających sport lub planujących prowadzenie zdrowego i aktywnego trybu życia. W niepełnej wersji, która oferuje podstawowe funkcje pomiarów i analiz danych, nie wymaga opłat. Endomondo zapewnia rejestrację czasu, dystansu, prędkości, tempa, czy ukształtowania terenu z wykorzystaniem GPS oraz Map Google. Korzystanie z funkcji jest proste i intuicyjne. W przypadku obszarów z odsłoniętym horyzontem jego działanie jest precyzyjne, natomiast dla miejsc zadrzewionych lub otoczonych wysokimi budynkami dokładność lokalizacji staje się znacznie niższa. Nie stanowi to jednak problemu dla odwzorowania przybliżonego przebiegu pokonanej trasy.

W celu wizualizacji otrzymanych wyników badań, skorzystano z Google Colab. Jest to narzędzie pozwalające na wykorzystanie języka Python, w szczególności do przeprowadzania różnego rodzaju analiz na danych liczbowych lub kategoriycznych. Poza tym, umożliwia również szybkie połączenie się z repozytorium GitHub lub dokumentami Google, a także jest zintegrowany z wieloma bibliotekami służącymi badaniu i wizualizacji danych oraz uczeniu maszynowemu, czego przykładem może być Plotly Express, PyTorch bądź TensorFlow.

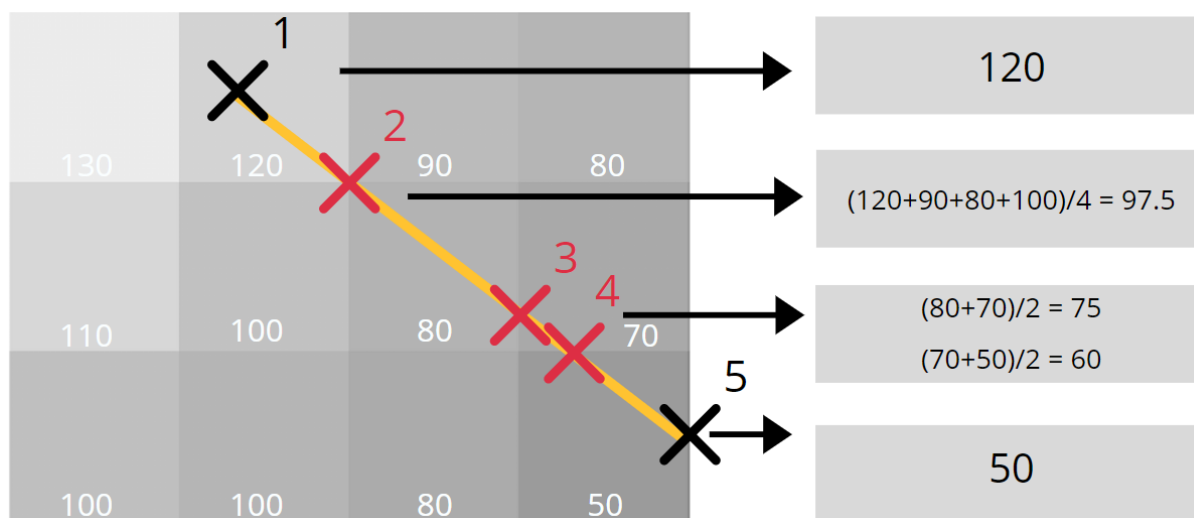
4. Opracowanie wtyczki QGIS

4.1. Implementacja algorytmów

Przygotowanie wtyczki do programu QGIS wymagało opracowania kompletnego zestawu wzajemnie powiązanych ze sobą funkcji. Budowanie kodu oraz sprawdzanie jego działania przed implementacją interfejsu graficznego odbywało się głównie z wykorzystaniem wbudowanej w programie konsoli Python.

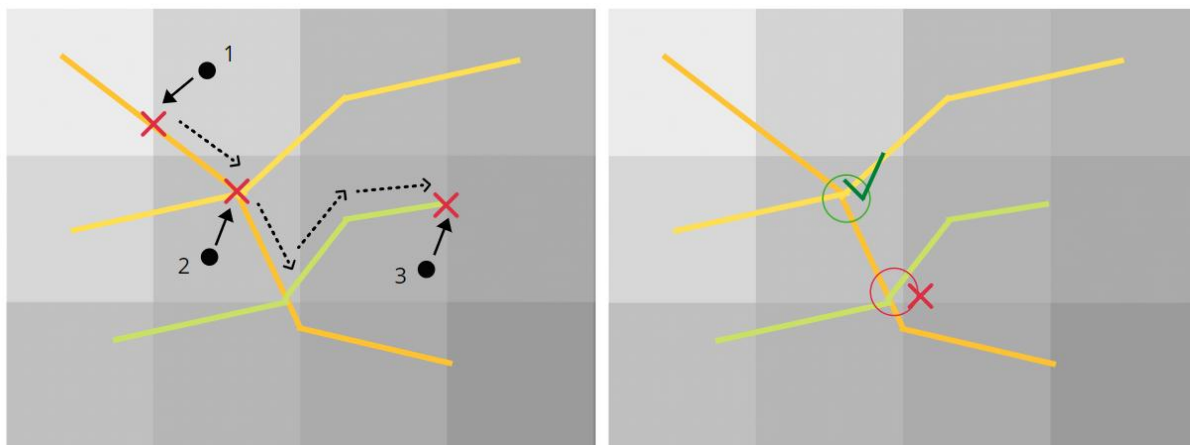
W trakcie pracy nad aplikacją konieczne było ustalenie zestawu warunków oraz wybranie najlepszych bądź najprostszych rozwiązań dla problematycznych przypadków. Przykładem takiej sytuacji był wybór sposobu interpolacji wysokości na krawędziach stykających się pikseli. W tym przypadku skorzystano z najprostszego rozwiązania polegającego na wyciąganiu średniej z wartości sąsiadujących ze sobą komórek. Innym przykładem było nadanie odpowiedniej formy zapisu relacji między obiektami i ich elementami. W tej sytuacji zastosowano tablice z biblioteki NumPy. Należało również opracować sposób analizowania i generowania tras dla okoliczności gdy punkt znajduje się w tej samej odległości do więcej niż jednego wierzchołka. W tym wypadku do tablicy z przebiegiem drogi zapisywane są wszystkie równoodległe punkty. Następnie wybierana jest jedna lub więcej tras najlepiej bądź równie dobrze spełniających warunek optymalizacji. Problemem wymagającym nadania pewnych warunków było także wytypowanie miejsc ewentualnego przejścia trasy na inny obiekt. Tutaj założono, że jest to możliwe wyłącznie w miejscach przecinania się oryginalnych wierzchołków warstwy liniowej (Rysunek 11).

Na początku opracowywania algorytmów sformułowano funkcje służące odczytowi informacji z danych rastrowych (wymiary i liczba pikseli, położenie, układ współrzędnych). Umożliwiło to skonstruowanie tablicy przechowującej wysokość oraz położenie każdej komórki obrazu. Następnie dodano opcje odczytu warstwy liniowej. W celu nadania poliliniom informacji o wysokości opracowano odpowiednie algorytmy. Bazują one na obliczaniu wysokości dla oryginalnych wierzchołków oraz punktów powstałych w wyniku przecinania się linii z krawędziami pikseli. Interpolację oparto na prostych zasadach: gdy wierzchołek znajduje się wewnątrz piksela lub na krańcach obrazu, to wysokość jest równa wartości odpowiedniego piksela, natomiast w przypadku gdy wierzchołek leży na krawędzi przynależnej do więcej niż jednej komórki, wyciągana jest wartość średnia (Rysunek 10).



Rysunek 10. Wizualizacja interpolacji wysokości z rastra na warstwę liniową. Punkty 1 oraz 5 – oryginalne, od 2 do 4 – powstałe w wyniku przecięcia linii na krawędziach komórek (źródło: opracowanie własne).

W kolejnym kroku dokonano odczytu warstwy z punktami przebiegu trasy. Względem nich wyznaczane są najbliższe wierzchołki dróg, co służy dalszym analizom (Rysunek 11). Kluczowym etapem było sformułowanie funkcji sprawdzających wszystkie możliwości przejścia między każdą parą podanych wierzchołków. Wymagało to uwzględnienia dwóch kierunków działania między odpowiednio posortowanymi w tablicy elementami obiektów (w przód, w tył) a także nadania zasad eliminacji lub zakończenia trasy. Za warunki odrzucenia wyznaczanego przebiegu drogi uznano: brak możliwości kontynuacji trasy bez osiągnięcia punktu docelowego bądź powtórzenie się tego samego wierzchołka. Nadano również kryteria mówiące o sposobie przejścia z jednego na drugi obiekt (wyłącznie w miejscach przecinania się oryginalnych załamań warstw liniowych) (Rysunek 11).



Rysunek 11. Po lewej: wyznaczenie miejsc przebiegu trasy na warstwie liniowej. Po prawej: przykład poprawnego oraz błędnego miejsca przejścia z jednego obiektu na drugi (źródło: opracowanie własne).

Dalszy etap opierał się na selekcji najbardziej optymalnej trasy spośród wszystkich dostępnych opcji. Aby możliwe było porównywanie ze sobą odpowiednich odcinków dróg należało wprowadzić funkcje przeliczające oraz akumulujące atrybuty: długość, czas, przewyższenia, podejścia, zejścia, spadki (4-9). Po spełnieniu przez dany zestaw warunku optymalizacji program pozwala zapisać wyniki z załączeniem wybranych atrybutów oraz wyświetlić go w projekcie programu QGIS (Rysunek 12).

Wzory na czas:

a) Pieszko

$$\text{Dla } |S| > 12^\circ : t = \frac{ds}{5 \frac{km}{h}} + \left| \frac{dh}{10 \frac{m}{min}} \right| \quad \text{Dla } |S| \leq 12^\circ : t = \frac{ds}{5 \frac{km}{h}} \quad (4)$$

b) Rowerem

$$\begin{aligned} \text{Dla } S > 12^\circ : t &= \frac{ds}{5 \frac{km}{h}} + \left| \frac{dh}{10 \frac{m}{min}} \right| & \text{Dla } |S| \leq 12^\circ : t &= \frac{ds}{5 \frac{km}{h}} \\ \text{Dla } S < -12^\circ : t &= \frac{ds}{5 \frac{km}{h}} - \left| \frac{dh}{10 \frac{m}{min}} \right| \end{aligned} \quad (5)$$

c) Samochodem

$$t = \frac{ds}{40 \frac{km}{h}} \quad (6)$$

Wzór na spadek:

$$S = \arctg \left(\frac{dh}{ds_{2D}} \right) \frac{180^\circ}{\pi} \quad (7)$$

Wzór na odległość między punktami:

$$ds_{2D} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (8)$$

$$ds = \sqrt{ds^2 + dh^2} \quad (9)$$

gdzie:

S – spadek terenu [°]

t – czas [min]

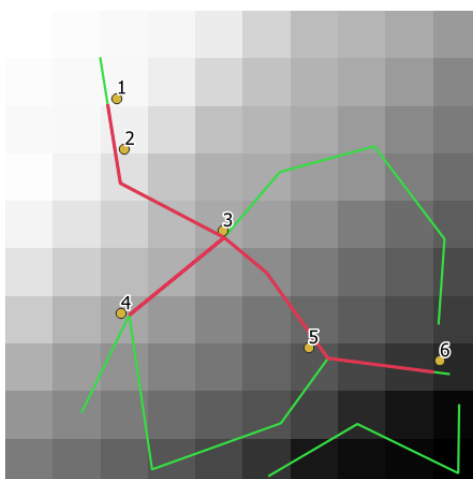
ds – odległość rzeczywista 3D między punktami [m]

dh – przewyższenie [m]

ds_{2D} – odległość pozioma 2D między punktami [m]

x₁, y₁ – współrzędne punktu początkowego [m]

x₂, y₂ – współrzędne punktu końcowego [m]



0_w :: Features Total: 1, Filtered: 1, Selected: 0									
id	length_2D	length_3D	time	elevation	height_dif	approaches	descents	mean_slope	
1	0	14.892	16.752	0.79	-5.040	35.135	2.692	14.060	25.024

Rysunek 12. Wizualizacja przykładowo wygenerowanej najszybszej trasy między punktami 1-6 wraz z przedstawieniem wartości zawartych w tabeli atrybutów (źródło: opracowanie własne).

Cała wtyczka składa się na zestaw wielu współpracujących ze sobą plików. Główna część wiążąca funkcje z interfejsem graficznym oraz pozwalająca na uruchomienie aplikacji znajduje się w pliku findRoute.py, gdzie proces generowania tras został podzielony na 22 segmenty.

Główne etapy służące aktywacji opracowanych algorytmów zamieszczonych w pliku functions.py:

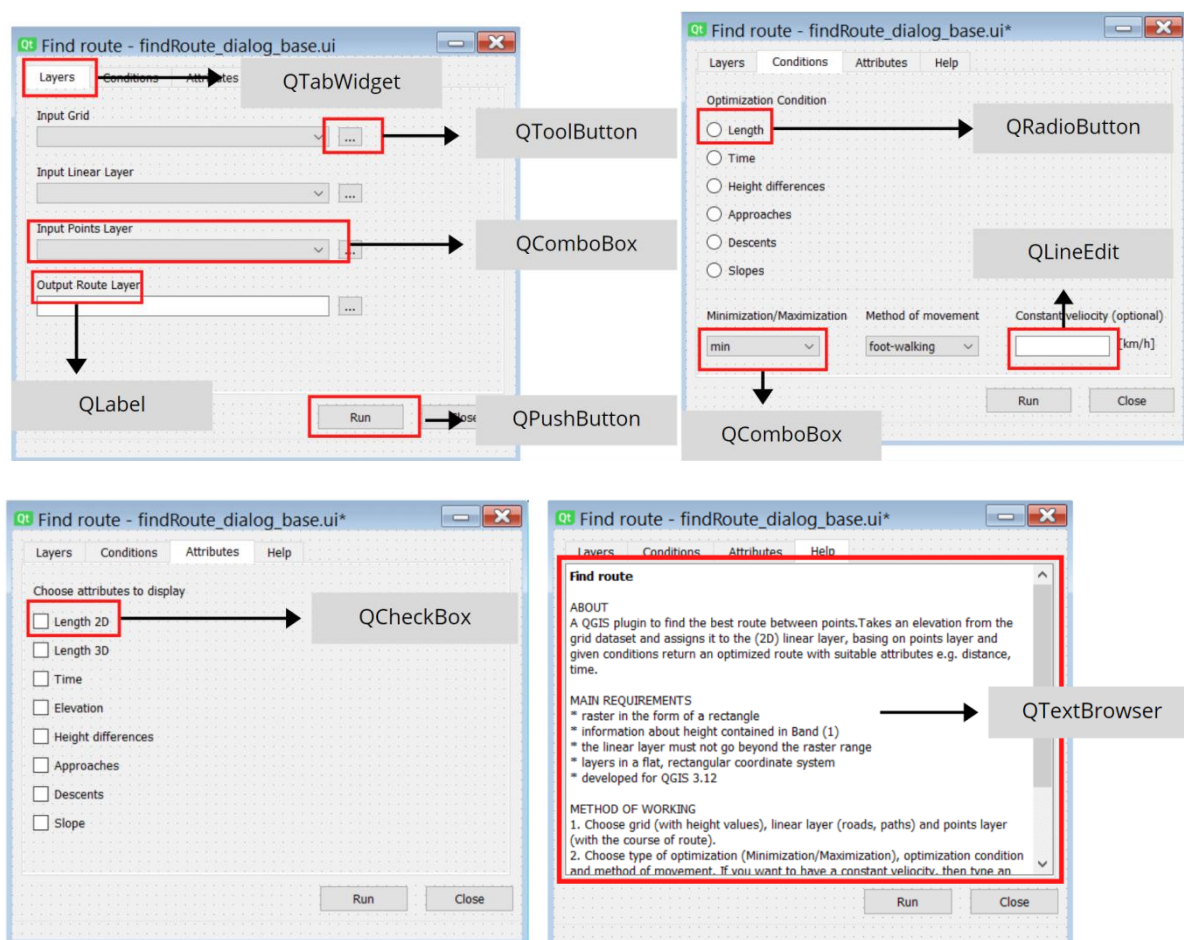
```
#[1] get paths
all_paths = self.read_input_file(layers)
raster_file_path = all_paths[0]
lines_file_path = all_paths[1]
points_file_path = all_paths[2]
result_file_path = all_paths[3]
#[2] open a raster file
raster_file = open_raster(self,raster_file_path)
#[3] create heights array from the grid
raster_height_arr = raster_height_array(self,raster_file)
#[4] return raster parameters
raster_param_arr = raster_parameters(self,raster_file,raster_height_arr)
#[5] return x,y,z array for all raster pixels
XYZ = raster_XYZ_array(self,raster_param_arr[3],raster_param_arr[4],raster_param_arr[5],\
raster_param_arr[6],raster_param_arr[7],raster_param_arr[8],raster_height_arr)
#[6] open the file a the linear layer
lines = open_vector(self,lines_file_path)
#[7] return objects from linear layer
fts = vector_objects(self,lines)
#[8] save information about existing vertex into an array
vertex_array = vertex_information_array(self,lines,fts)
#[9] find an intersection between pixels and linear layer, finally return an array: [object number, first
vertex value, next vertex value, new vertex number, x, y]
end_std_new_vert_arr = raster_line_intersection(self,lines,raster_param_arr[4],raster_param_arr[3],\
raster_param_arr[5],raster_param_arr[6],raster_param_arr[7],raster_param_arr[8],vertex_array)
#[10] assign heights to all vertices
new_z_arr = new_vertex_elevation_array(self,end_std_new_vert_arr,XYZ,raster_param_arr)
#[11]return an array with sorted information about vertices
new_info_arr = new_vertex_information_array(self,end_std_new_vert_arr,new_z_arr,4,4)
#[12] open a file with the points layer
route_pts = open_vector(self,points_file_path)
#[13] return an array with information about point layer
route_pts_arr = points_information_array(self,route_pts)
#[14] find the closest vertices to points
closest_pts_arr0 = closest_point(self,route_pts_arr, new_info_arr)
#[15] return an array with the possible combinations of routes between the closest vertices
closest_pts_arr = closest_pts_possibility(self,closest_pts_arr0,new_info_arr)
#[16] find all possible routes
possible_routes_arr = possible_routes(self,closest_pts_arr,new_info_arr)
#[17] complete route array
compl_poss_route_arr = complete_possible_route(self,possible_routes_arr,new_info_arr)
#[18] fill shortages on "compl_poss_route_arr"
new_compl_poss_route_arr = check_array_shortages(self,compl_poss_route_arr,possible_routes_arr)
#[19] make a table with attributes for sections of the routes (routes between two points)
sections_attr_arr = route_sections_attributes(self,new_compl_poss_route_arr)
#[20] choose the best route using optimization parameters
route_optim_array = route_optimization(self,sections_attr_arr,closest_pts_arr0)
#[21] choose attributes to be attached
atr = self.choose_attributes()
#[22] create and display an output file
create_shp_route(self,atr[0],atr[1],atr[2],atr[3],atr[4],atr[5],atr[6],atr[7],result_file_path,\
raster_param_arr[11],route_optim_array[0],new_compl_poss_route_arr,route_optim_array[1],\
raster_param_arr,3,2,2)
```

4.2. Interfejs graficzny

Po opracowaniu odpowiedniego zestawu funkcji zaprojektowano zewnętrzny wygląd aplikacji. Wykonano to przy użyciu biblioteki PyQt5, która umożliwia skorzystanie z opcji Designera pozwalającego na wizualne budowanie interfejsu. Jako podstawowy język aplikacji wybrano angielski, który zapewnia możliwość łatwego tłumaczenia oraz wykorzystania wtyczki również przez osoby pochodzące z innych państw. Zaprojektowano cztery osobne zakładki: warstw, warunków, atrybutów oraz pomocy. Interaktywność skryptu z użytkownikiem została zapewniona poprzez zastosowanie następujących widgetów:

- QComboBox, umożliwiający rozwinięcie listy z elementami do wyboru oraz dodanie nowych poprzez odpowiednie połączenie z funkcjonalnością innych ikon,
- QToolButton, łączący z eksploratorem plików Windows,
- QTabWidget, zakładki, dzielące aplikację na osobne tematycznie części,
- QLabel, obszary do nadawania opisów tekstowych,
- QPushButton, przyciski reagujące na kliknięcie klawiszem myszy,
- QRadioButton, służy wyborze wyłącznie pojedynczego elementu spośród wielu opcji,
- QLineEdit, linia z miejscem do wpisania wartości z klawiatury,
- QCheckBox, dający możliwość wyboru wszystkich, kilku bądź żadnej z dostępnych opcji,
- QTextBrowser, miejsce na tekst z możliwością edycji jako html.

Wykonano wizualizację przybliżającą schemat budowy interfejsu wtyczki względem osobnych elementów umieszczonych w każdej z zakładek (Rysunek 13).



Rysunek 13. Budowa i zawartość zakładek wtyczki Find Route (źródło: opracowanie własne).

Zaprojektowana część graficzna została zapisana do pliku w formacie XML, który wykorzystywany jest przez QT Designer jako format zapisu swoich obiektów. W opracowanym kodzie uwzględniono obsługę sygnałów, które informują o stanie elementów interfejsu, co umożliwia wywołanie odpowiedniej reakcji w przypadku wystąpienia określonych zmian. Część wizualna aplikacji pozwala użytkownikowi w dużym stopniu wpływać na działanie wtyczki. Widoczne jest to zarówno w zakładce warstw, gdzie można wybrać własne pliki wejściowe lub wskazać docelowe miejsce zapisu wyznaczonej trasy, jak i w pozostałych częściach dających możliwość wskazania warunków optymalizacji, zdefiniowania sposobu przemieszczania się oraz wyboru rodzajów i liczby wartości dołączanych do tabeli atrybutów.

W celu informowania użytkownika o wystąpieniu problemów podczas obliczeń lub wczytywaniu danych wtyczka została wyposażona w komunikaty QMessageBox (Rysunek 14). Ułatwiają one zrozumienie oraz sprawną poprawę błędów występujących w wyniku nieprawidłowej obsługi aplikacji. Zastosowanie tego rodzaju zabezpieczeń zostało użyte w przypadkach takich jak: wykroczenie warstwy z trasami poza obszar rastra (Rysunek 15), wprowadzenie warstwy punktowej zamiast liniowej bądź odwrotnie (Rysunek 16), brak wskazania metody optymalizacji oraz wpisanie liter w miejscu określającym wartości prędkości (Rysunek 17). Poza tym komunikaty zostały wykorzystane również w sytuacji braku możliwości wyznaczenia rozwiązania, błędów odczytów ścieżki dostępu do plików lub problemów z ich otwarciem.

```
def open_raster(self,raster_file_path):
    try:
        raster_file = gdal.Open(raster_file_path)
        return raster_file
    except IOError:
        QMessageBox(QMessageBox.Warning, "Error","Failed to open raster dataset.", QMessageBox.Ok).exec()

def vector_objects_elements(self,ft):
    geom_ft = ft.geometry()
    #check if the type is: single or not
    geom_type = QgsWkbTypes.isSingleType(geom_ft.wkbType())

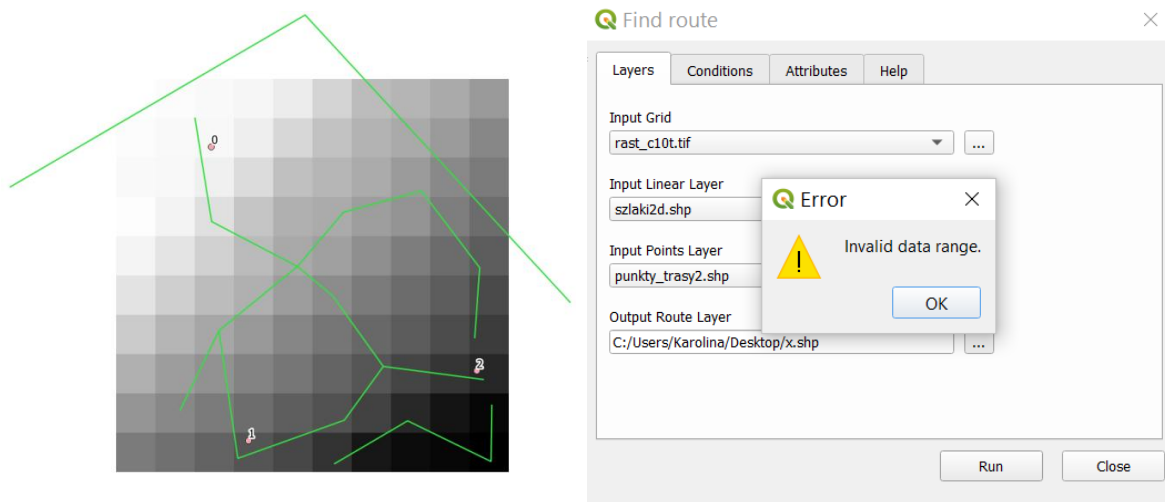
    #check a detailed type of the geometry: polyline, multi-polyline or else
    if geom_ft.type() == QgsWkbTypes.LineGeometry:
        if geom_type:
            elem_xy = geom_ft.asPolyline()
        else:
            elem_xy = geom_ft.asMultiPolyline()
    else:
        QMessageBox(QMessageBox.Warning, "Error","Invalid linear layer format.", QMessageBox.Ok).exec()

    return elem_xy

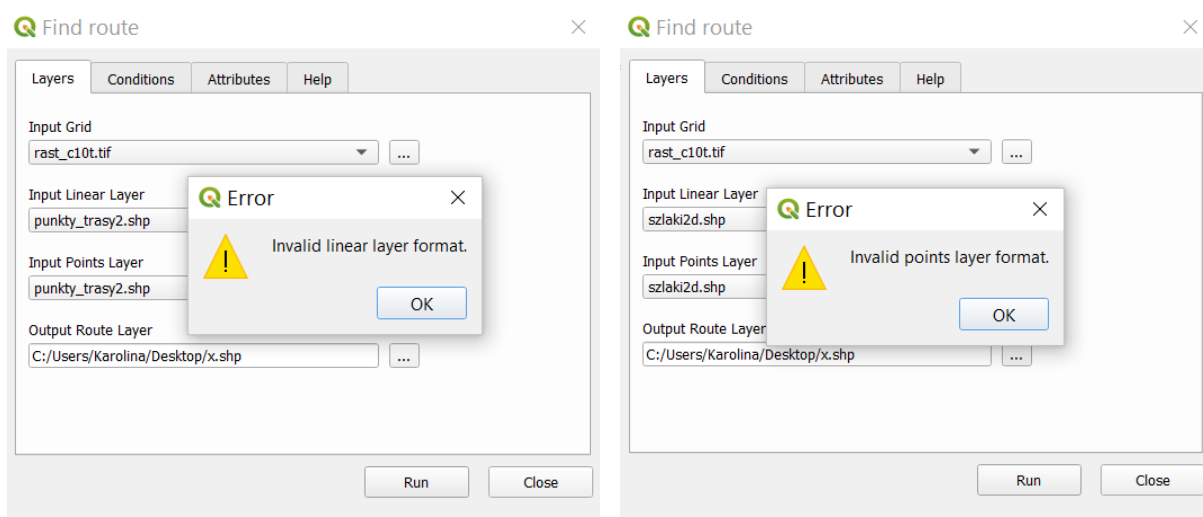
def route_optimization(self,sections_attr_arr,closest_pts_arr0):

    #choose parameter to optimize route
    optimization = self.check_optimization()
    if optimization == "none":
        QMessageBox(QMessageBox.Warning, "Error","Optimization method not selected.", QMessageBox.Ok).exec()
```

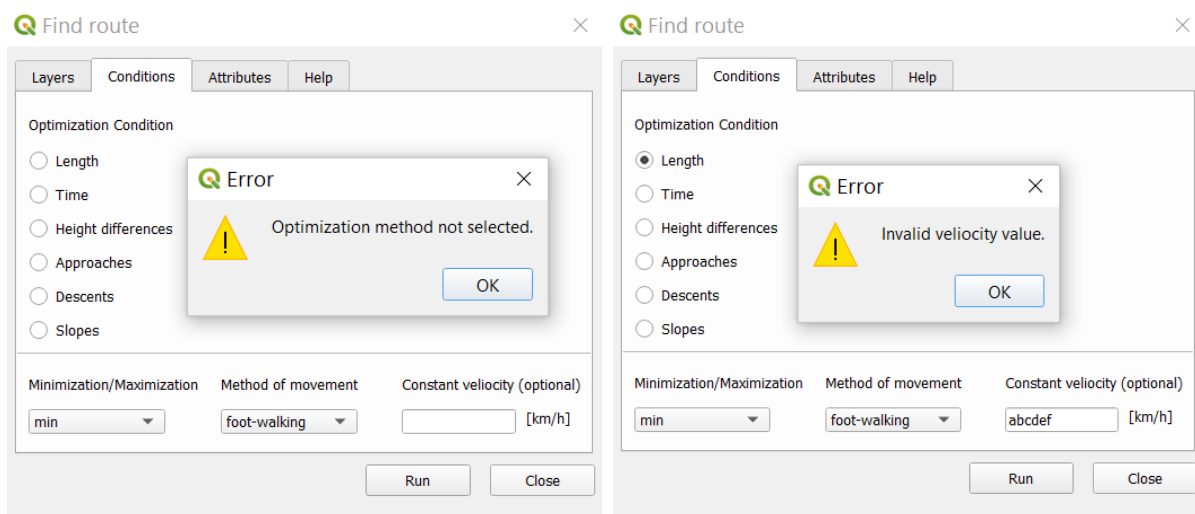
Rysunek 14. Wykorzystanie i zapis komunikatów QMessageBox w przykładowych funkcjach: odczytującej plik z rastrem, sprawdzającej rodzaj obiektów, odczytującej warunek optymalizacji (źródło: opracowanie własne).



Rysunek 15. Przykładowy wygląd niepoprawnej warstwy z komunikatem błędu (źródło: opracowanie własne).



Rysunek 16. Sygnalizacja błędów formatu wprowadzanych danych (źródło: opracowanie własne).

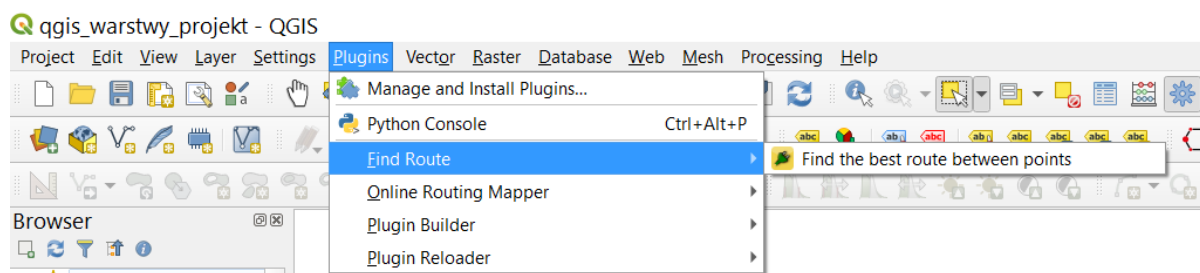


Rysunek 17. Komunikat braku wskazania optymalizacji oraz błędu wartości prędkości (źródło: opracowanie własne).

Aplikacja została zbudowana z wykorzystaniem szablonu projektowego pochodzącego z darmowej i ogólnodostępnej wtyczki PluginBuilder, która pomogła w poprawnym połączeniu skryptu z oprogramowaniem QGIS. Wygląd zewnętrzny został zaprojektowany w taki sposób, aby użytkownik bez wglądu do dokumentacji potrafił sam poprawnie uruchomić program. Zakładki umożliwiają szybki powrót i ewentualną zmianę zastosowanych parametrów. Głównym założeniem projektu była prostota, czytelność i wielofunkcyjność.

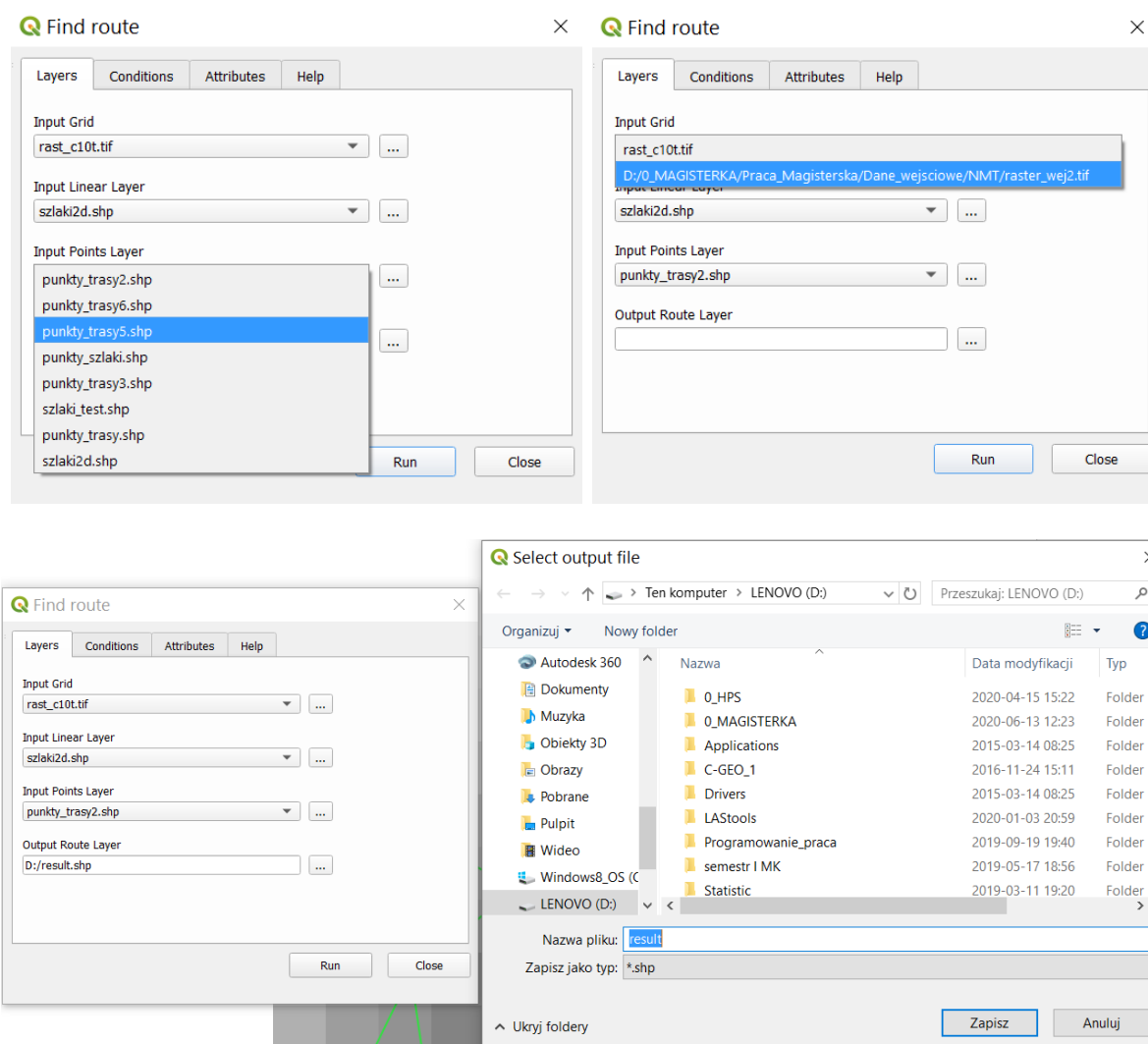
4.3. Instrukcja obsługi

Find Route jest przeznaczona do działania w połączeniu z oprogramowaniem QGIS (testy prowadzono w wersji 3.12.0). W celu jej implementacji należy wgrać folder z wszystkimi niezbędnymi plikami, które umieszczono na platformie GitHub (https://github.com/AksamitKarolina/Praca_Magisterska_FindRoute_Aksamit_Karolina.git). Przykładowa ścieżka zapisu: C:\ProgramFiles\QGIS\3.12\apps\qgis\python\plugins. Aby mieć możliwość bieżącego aktualizowania oraz resetowania zapisu ustawień aplikacji można skorzystać z opcji zawartych we wtyczce Plugin Reloader. Uruchomienie następuje z zakładki wtyczek głównego menu programu QGIS (Rysunek 18).



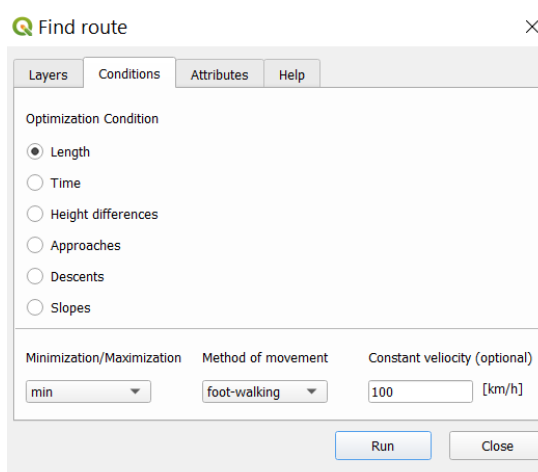
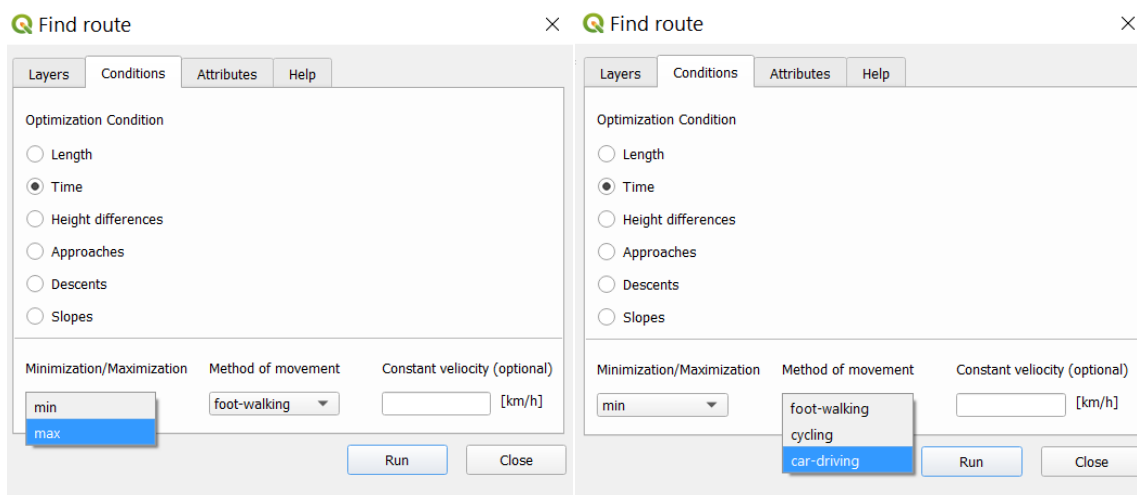
Rysunek 18. Lokalizacja miejsca uruchamiania wtyczki Find Route
(źródło: opracowanie własne).

Po uruchomieniu wtyczki pojawia się interfejs graficzny składający się z czterech głównych części, z czego ostatnia stanowi wyłącznie funkcję informacyjną. W pierwszej zakładce w miejscu „Input Grid” należy wskazać numeryczny model terenu w postaci prostokątnej, regularnej siatki pikseli. Inna forma danych wywoła błąd związany z powiązaniem informacji o wysokości względem wprowadzonej sieci dróg. Dla „Input Linear Layer” wymagane jest załadowanie liniowej warstwy przebiegu połączeń w postaci Shapefile. Ten sam format, ale z punktami definiującymi kolejność przebiegu trasy jest wymagany dla „Input Points Layer”. Pliki wprowadzane do aplikacji są możliwe do wskazania bezpośrednio z automatycznie wygenerowanej listy, w której znajdują się warstwy zamieszczone w projekcie lub z wykorzystaniem okna eksploratora Windows. Ostatnim wymaganiem polem do uzupełnienia jest „Output Route Layer”, gdzie zamieszcza się ścieżkę zapisu dla wyznaczonej trasy (Rysunek 19). Wtyczka nie wymaga zdefiniowania układu współrzędnych dla warstwy wynikowej, jako domyślny przyjmowany jest układ wprowadzonego rastra.



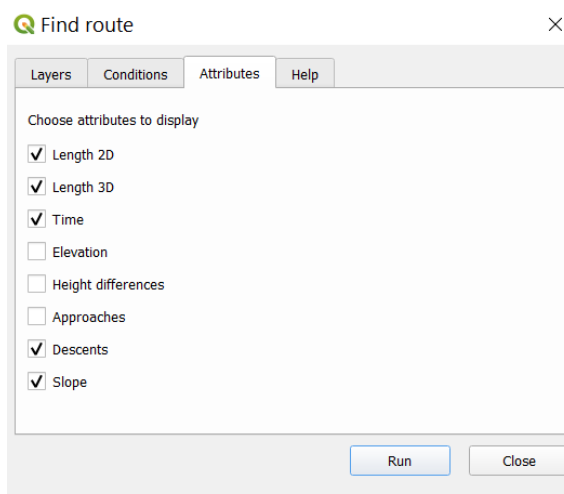
Rysunek 19. Prezentacja obsługi zakładki warstw
(źródło: opracowanie własne).

Druga część zawiera zestaw parametrów: długość, czas, różnice przewyższeń, podejścia, zejścia, spadki, spośród których wybierany jest wyłącznie jeden warunek optymalizacji. Brak jego zdefiniowania powoduje wystąpienie komunikatu o błędzie. Poza tym istnieje również możliwość wskazania, czy atrybuty mają osiągnąć wartość minimalną, czy maksymalną. „Method of movement” wskazuje aplikacji w jaki sposób należy przeliczać prędkość poruszania się. Wprowadzono trzy główne tryby: pieszy, rowerem, samochodem. Dla pierwszych dwóch sposobów przemieszczania się zaimplementowano odpowiednie wzory, natomiast dla aut przyjęto przybliżoną średnią prędkość jazdy w mieście, to jest 40 km/h. Opcjonalne jest wpisanie stałej wartości prędkości, która powoduje brak uwzględniania metody lokomocji z rozwijanej wcześniej listy (Rysunek 20).



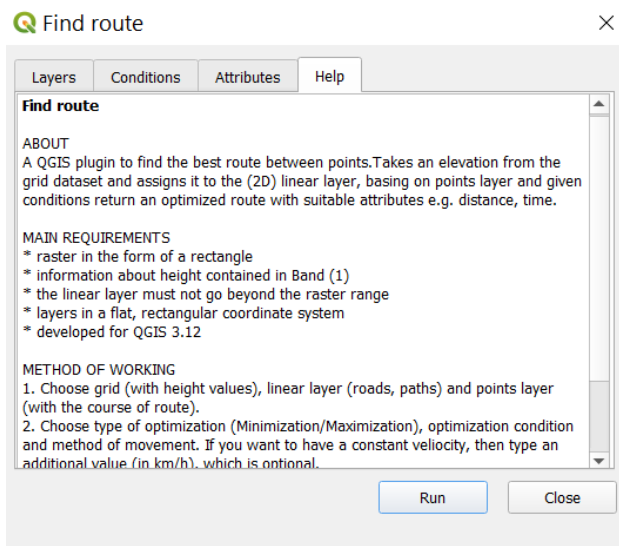
Rysunek 20. Prezentacja obsługi zakładki warunków
(źródło: opracowanie własne).

Zakładka z atrybutami pozwala dołączyć do wyniku informacje o długości poziomej i rzeczywistej, czasie, przewyższeniu, sumie bezwzględnych różnic wysokości, podejść, zejść oraz średnim spadku trasy. Dozwolony jest brak wskazania żadnego z elementów, wybór jednego, kilku bądź załączenie wszystkich (Rysunek 21).



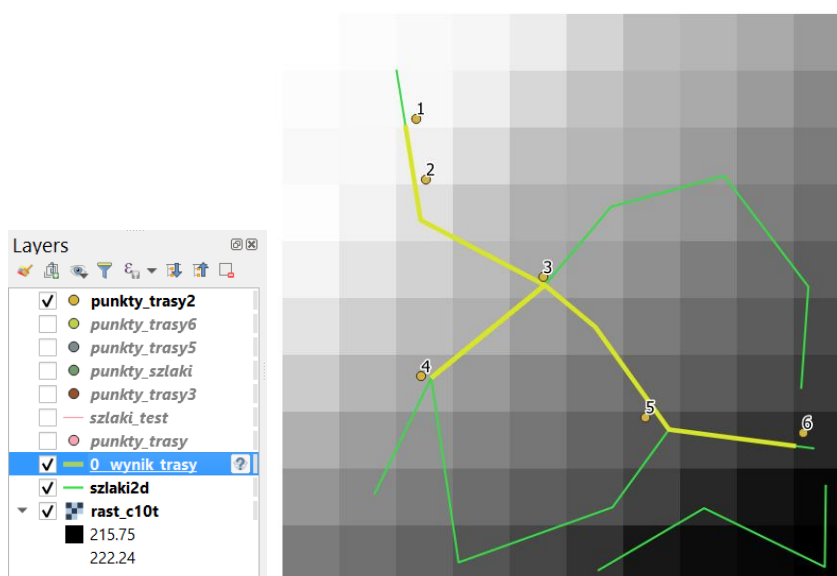
Rysunek 21. Prezentacja obsługi zakładki atrybutów
(źródło: opracowanie własne).

„Help” stanowi część opisową wtyczki (Rysunek 22). Przedstawia ogólną ideę, przeznaczenie oraz podstawowe wymagania związane z formatem warstw. Dodatkowo znajduje się tam również krótki opis etapów przebiegu pracy. Umożliwia to określenie, czy aplikacja spełnia założone wymagania i będzie odpowiednia do zastosowania w rozpatrywanym przypadku.



Rysunek 22. Prezentacja zakładki pomocy
(źródło: opracowanie własne).

Po uzupełnieniu wymaganych pól w celu uruchomienia kodu należy wybrać przycisk „Run”. W razie wystąpienia błędów pojawi się odpowiedni komunikat informujący o przyczynie problemu. Jeśli proces przebiegnie pomyślnie, wynikiem będzie jedna bądź więcej wektorowych warstw liniowych pokazujących przebieg najlepszej drogi w układzie zdefiniowanym przez raster. Większa liczba oznacza, że więcej niż jedna trasa spełnia równie dobrze warunek optymalizacji. Poza zapisem wyniku do pliku pojawia się także jego wizualizacja w obecnie otwartym projekcie (Rysunek 23).

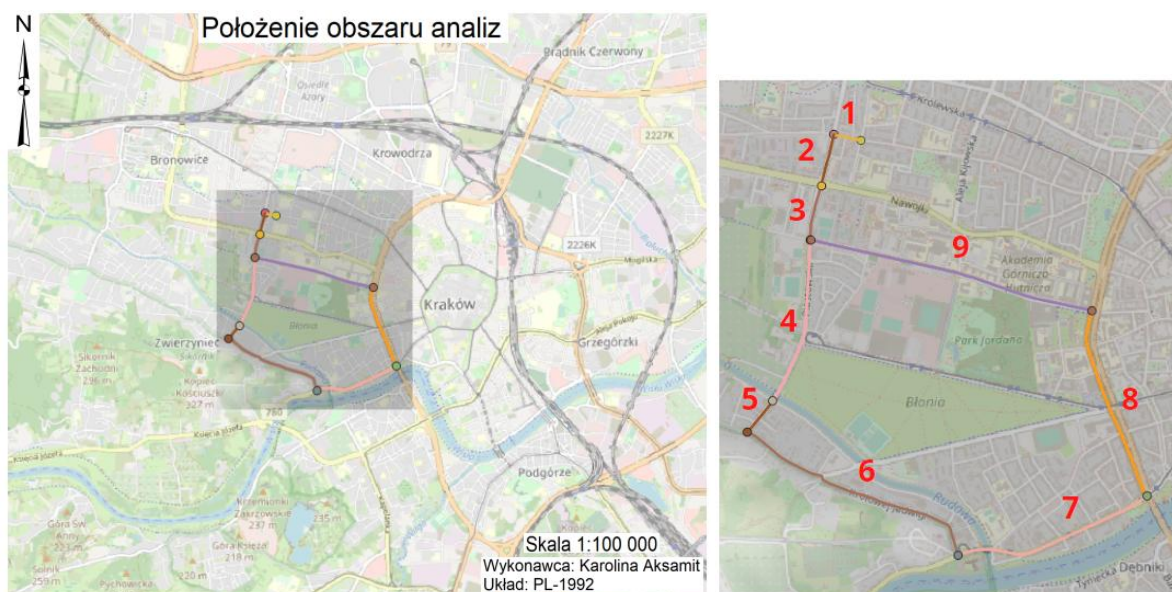


Rysunek 23. Prezentacja warstwy wynikowej
(źródło: opracowanie własne).

5. Opis przeprowadzonych badań

5.1. Pozyskanie danych porównawczych z terenu

Do celów analizy wiarygodności wykonano pomiar rzeczywistych wartości czasu potrzebnego do przejścia między zadanymi odcinkami znajdującymi się w terenie. Jako obszar badań wybrano fragment centralno-zachodniej części Krakowa. Zaplanowano trasę o długości ponad 7 km, składającą się z 9 części (Rysunek 24, Tabela 5). Dane zbierano stosując dwa sposoby lokomocji: pieszo oraz samochodem. Badania przeprowadzono w czwartek, 4 czerwca 2020 roku między godziną 12.20 a 14.40.



Rysunek 24. Położenie obszaru analiz na tle warstwy Open Street Map
(źródło: opracowanie własne).

Tabela 5. Opis odcinków mierzonej trasy.

Numer	Początek	Koniec
1	Skrzyżowanie ulic: Warmijska, Juliusza Lea	Skrzyżowanie ulic: Piastowska, Juliusza Lea
2	Skrzyżowanie ulic: Piastowska, Juliusza Lea	Skrzyżowanie ulic: Piastowska, Armii Krajowej, Nawojki
3	Skrzyżowanie ulic: Piastowska, Armii Krajowej, Nawojki	Skrzyżowanie ulic: Piastowska, Jana Buszka, Reymonta
4	Skrzyżowanie ulic: Piastowska, Jana Buszka, Reymonta	Skrzyżowanie ulic: Piastowska, Emaus
5	Skrzyżowanie ulic: Piastowska, Emaus	Skrzyżowanie ulic: Piastowska, Królowej Jadwigi
6	Skrzyżowanie ulic: Piastowska, Królowej Jadwigi	Skrzyżowanie ulic: Królowej Jadwigi, Księcia Józefa, Tadeusza Kościuszki
7	Skrzyżowanie ulic: Królowej Jadwigi, Księcia Józefa, Tadeusza Kościuszki	Skrzyżowanie ulic: Tadeusza Kościuszki, Zwierzyniecka, alei Zygmunta Krasińskiego
8	Skrzyżowanie ulic: Tadeusza Kościuszki, Zwierzyniecka i alei Zygmunta Krasińskiego	Skrzyżowanie ulicy Reymonta i alei Adama Mickiewicza
9	Skrzyżowanie ulicy Reymonta i alei Adama Mickiewicza	Skrzyżowanie ulic: Piastowska, Jana Buszka, Reymonta

Wyniki pomiarów pozyskano z wykorzystaniem aplikacji Endomondo, która zapisywała zestaw parametrów: czas, dystans, tempo, prędkość, spalane kalorie, minimalną, maksymalną wysokość, ilość podejść i zejść, a także wizualne odwzorowanie przebytej drogi. Ze względu na niską dokładność pozyskiwania większości z mierzonych cech, skupiono się na analizie czasu oraz odległości (Tabela 6).

Tabela 6. Atrybuty dla odcinków trasy pomierzonej w terenie, rodzaj lokomocji: pieszo (po lewej), samochodem (po prawej).

Trasa	Czas [min]	Długość [km]	Trasa	Czas [min]	Długość [km]
1	1.50	0.14	1	0.38	0.13
2	2.93	0.29	2	1.37	0.28
3	3.50	0.32	3	0.38	0.26
4	11.15	0.91	4	1.37	0.82
5	2.30	0.18	5	1.18	0.21
6	16.78	1.48	6	2.02	1.31
7	13.15	1.15	7	2.43	1.09
8	12.83	1.23	8	4.30	1.12
9	18.45	1.42	9	2.43	1.49

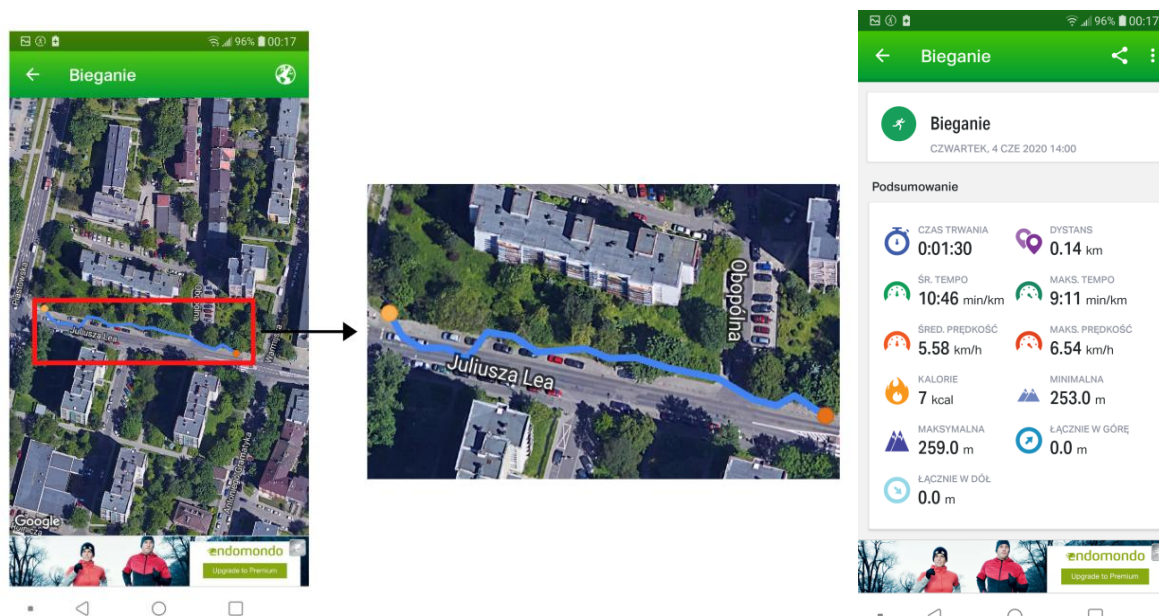
Z powyższej tabeli wynika, że najdłuższa trasa piesza (1.48 km), nie uzyskała największej wartości czasu. Osiągnął to krótszy odcinek 9 (1.42 km) z różnicą około 2 min oraz 60 m. Podobna sytuacja pojawiła się dla ruchu samochodowego. Droga numer 8 (1.12 km) była najwolniejsza a 9 (1.49 km) najdłuższa. W przypadku obu rodzajów lokomocji najszybsza trasa, była również najkrótszą (numer 1).

Odcinki pomiarowe charakteryzowały się dużym zróżnicowaniem pod względem długości oraz specyfiki terenu. Trasa 6 przebiegała przez obszary o najbardziej zmiennym ukształtowaniu terenu, gdzie w bliskim otoczeniu występowały skarpy oraz drzewa. Odcinek 8 rozciągał się wzdłuż drogi otoczonej wysoką i zwartą zabudową. Fragmenty 1, 2, 3, 5 były krótkie i proste, a budynki znajdowały się już w większej odległości. W odniesieniu do ruchu samochodowego największe problemy pojawiły się dla odcinka 8, który prowadził przez drogę o silnie natężonym ruchu, zawierającą kilka miejsc wymagających zwolnienia bądź zatrzymania się, takich jak: przejścia dla pieszych, sygnalizacja świetlna oraz zmiana pasa ruchu, w wyniku czego średnia prędkość wyniosła około 15 km/h, prawie trzykrotnie mniej niż najszybsza (Tabela 7). Najwolniejszą trasą okazał się jednak numer 5. Powodem tej sytuacji była konieczność zatrzymania się w korku utworzonym przed światłami na skrzyżowaniu ulicy Piastowskiej z Królowej Jadwigi.

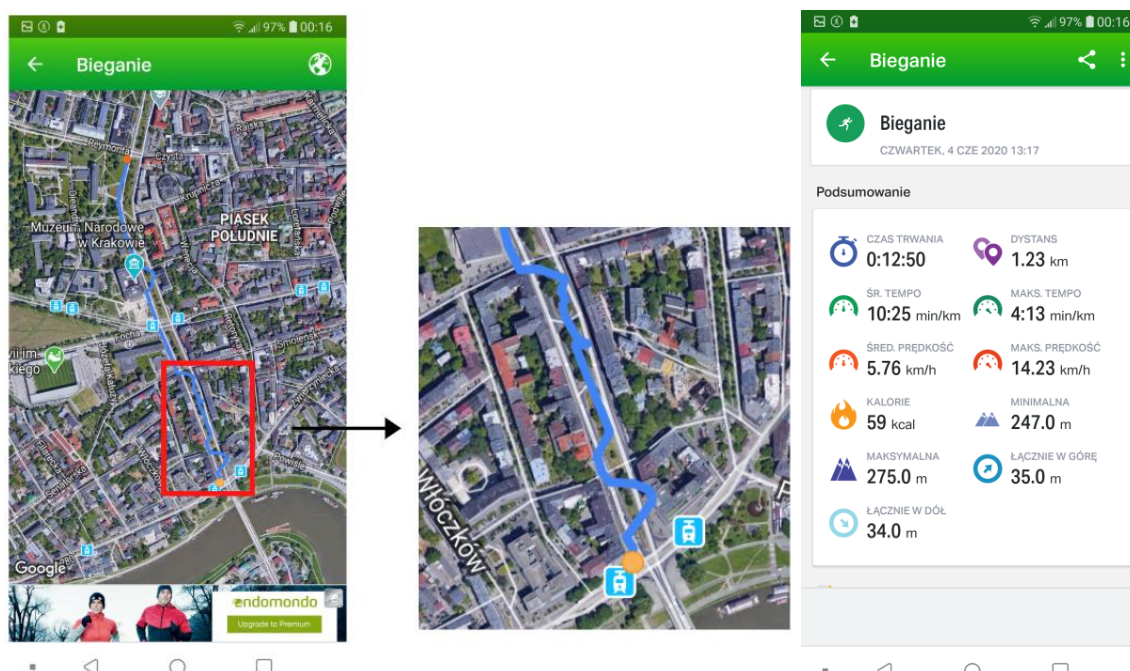
Tabela 7. Zestawienie średnich prędkości dla odcinków trasy pomierzonych w terenie.

Numer	Prędkość – ruch pieszy [km/h]	Prędkość – ruch samochodowy [km/h]
1	5.58	20.95
2	5.96	12.26
3	5.47	40.35
4	4.91	36.01
5	4.73	10.65
6	5.30	38.86
7	5.24	26.81
8	5.76	15.65
9	4.63	36.64

W trakcie pomiarów zauważono, że aplikacja Endomondo generuje zapis trasy często znacząco odbiegający od prawdy z przesunięciami sięgającymi kilkunastu metrów. GPS działał niepoprawnie w miejscach o zakrytym horyzoncie, na przykład między ulicami z wysokimi kamienicami lub blisko usytuowanymi drzewami o rozłożystych koronach. Takie rozbieżności można zauważyć w szczególności na trasach 1 i 8 (Rysunek 25, Rysunek 26).



Rysunek 25. Wizualizacja rozbieżności wywołanych zadrzewieniem – trasa 1 (źródło: opracowanie własne na podstawie Endomondo).



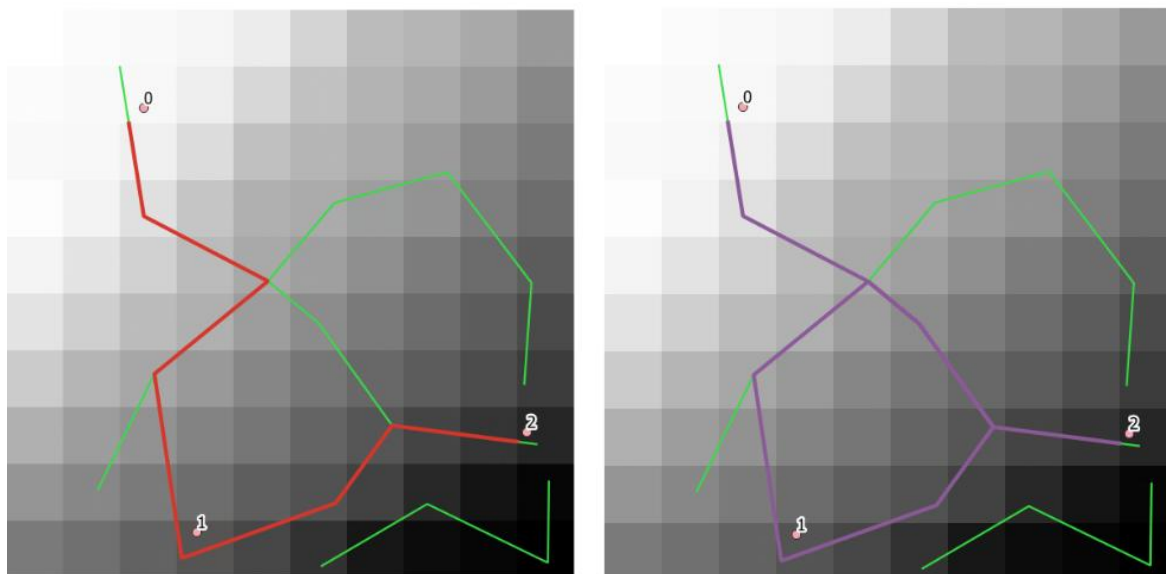
Rysunek 26. Wizualizacja rozbieżności wywołanych gęstą i wysoką zabudową – trasa 8 (źródło: opracowanie własne na podstawie Endomondo).

5.2. Generowanie tras z wykorzystaniem wtyczki

5.2.1 Dane testowe

W celu sprawdzania poprawności działania algorytmów konieczna była ciągła kontrola wyników, którą wykonywano z wykorzystaniem niewielkiego wycinka rastra, ręcznie wrysowanej warstwy liniowej oraz kilku zestawów punktów. Głównym założeniem kodu było zapewnienie jak największej uniwersalności rozwiązania dla różnych przypadków rozmieszczenia punktów. Wszystkie trasy generowane były w czasie mniejszym niż 1 s. Sprawdzono również jak długo należy czekać w przypadku znacznie większego rastra i czy możliwości obliczeniowe komputera będą wystarczające. W tym celu wykonywano adekwatne analizy opierając się na wysokorozdzielczych danych kontrolnych. Dla zrozumienia rozważanych kombinacji testowych przytoczone zostanie kilka przykładów generowania trasy z załączeniem odpowiednich komentarzy (Rysunek 27 - Rysunek 30).

Wtyczka zapewnia wybór opcji minimalizacji bądź maksymalizacji określonego parametru, co rozszerza możliwość jej optymalizacji (Rysunek 27, Rysunek 28). Wprowadzona zasada braku powtórnego przejścia przez ten sam wierzchołek między każdą parą punktów ogranicza liczbę analizowanych możliwości. W przypadku firm transportowych głównym priorytetem optymalizacji będzie wyznaczenie jak najszybszej drogi. Poza tym, innymi ważnymi aspektami może być minimalizacja ilości podjazdów lub wielkość nachylenia na trasie (szczególnie w przypadku przewożenia bardzo ciężkich ładunków). Jednak w odniesieniu do turystów wymagania mogą być całkiem inne, gdyż chcieliby oni zwiedzić jak najwięcej. Wtyczkę można zastosować także w planowaniu przebiegu torów lub dróg, gdzie istotną rolę odgrywa ukształtowanie terenu.

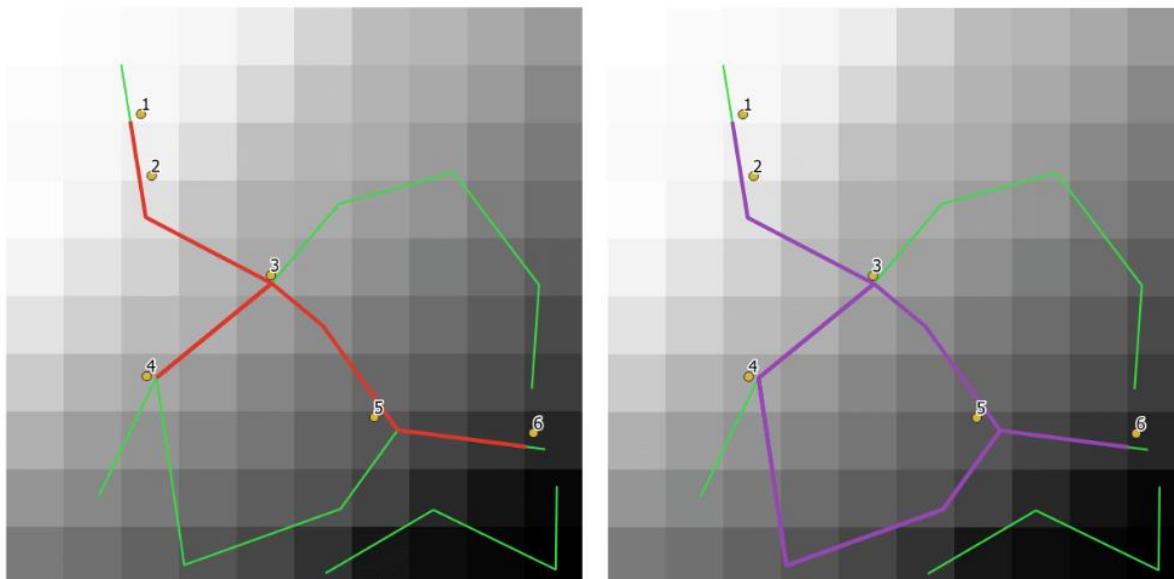


Rysunek 27. Wpływ minimalizacji i maksymalizacji warunku optymalizacji na wygląd trasy (źródło: opracowanie własne).

	id	length_2D	length_3D	time	elevation	height_dif	approaches	descents	mean_slope
1	0	16.789	18.815	0.83	-5.040	16.380	3.540	15.275	26.122
	id	length_2D	length_3D	time	elevation	height_dif	approaches	descents	mean_slope
1	0	23.553	27.150	1.35	-5.040	27.300	8.074	19.075	29.764

Rysunek 28. Wpływ minimalizacji i maksymalizacji warunku optymalizacji na atrybuty trasy (źródło: opracowanie własne).

Generowana warstwa jest wynikiem zestawienia różnych kombinacji parametrów. W zależności od nich jej przebieg jest dobierany tak, aby spełnić oczekiwania użytkownika, który może decydować o tym co jest dla niego najważniejsze. Tabela atrybutów zapewnia dostęp do informacji o wielu parametrach, co pozwala szybko ocenić czy wynik jest satysfakcjonujący lub czy istnieje konieczność wykonania innej optymalizacji (Rysunek 29, Rysunek 30). W logistyce każdy z parametrów jest mniej bądź bardziej znaczący. Aby zapewnić w mieście najszybszy transport ludzi lub zaplanować najlepszy dojazd służb ratunkowych należy kierować się czasem, jednak gdy teren jest górzysty ważniejszą rolę mogą odgrywać wielkości nachylenia terenu, a dla osób na wózkach inwalidzkich ilość podjazdów i zjazdów jest kluczowa.



Rysunek 29. Wpływ optymalizacji względem parametru czasu oraz spadków
(źródło: opracowanie własne).

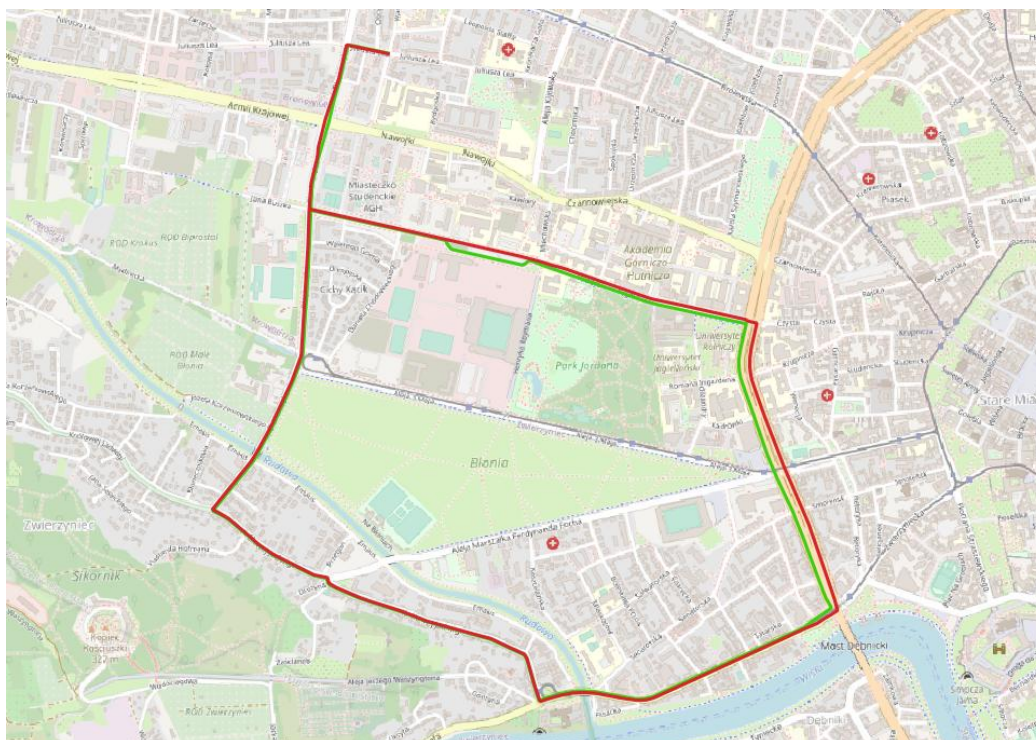
	id	length_2D	length_3D	time	elevation	height_dif	approaches	descents	mean_slope
1	0	14.892	16.752	0.79	-5.040	35.135	2.692	14.060	25.024
	id	length_2D	length_3D	time	elevation	height_dif	approaches	descents	mean_slope
1	0	30.619	34.597	1.55	-5.040	45.245	11.247	23.350	24.544

Rysunek 30. Atrybuty trasy po optymalizacji względem parametru czasu oraz spadków
(źródło: opracowanie własne).

Aplikacja służy nie tylko do wyznaczania optymalnych tras, ale może być narzędziem badania cech ukształtowania powierzchni. Wrysowanie pojedynczej linii wraz z załączeniem punktów znajdujących się na jej krańcach pozwala badać profile terenu oraz zapewnia szybki dostęp do gotowego zestawu parametrów. W ten sposób można między innymi usprawnić proces planowania inwestycji pod budowę. Zdarza się, że inwestor ma do wybrania kilka działek i musi zdecydować, która z nich będzie najlepsza. W przypadku budowy bloków mieszkalnych lub domów jednorodzinnych wtyczka może zapewnić szybkie sprawdzenie czasu dojścia do przystanków. Dla dróg i autostrad stanowiłaby źródło informacji o stosunku średniej wartości nachyleń, czyli poglądowe miary potrzeby wyrównania powierzchni. Napisany kod jest podstawą do stworzenia znacznie bardziej specjalistycznego narzędzia. W prosty sposób możliwa jest jego edycja oraz rozwój poprzez opracowanie nowych funkcjonalności, usprawnienie metod obliczeń bądź dostęp do większej liczby atrybutów.

5.2.2 Dane kontrolowane

Wiarygodność funkcjonowania wtyczki wymaga porównania wartości wynikowych z rzeczywistą sytuacją. W tym celu zaplanowano trasę testową, której odcinki zostały poddane pomiarowi. W programie QGIS z wykorzystaniem podkładu w postaci Open Street Map dokonano przybliżonego odwzorowania liniowego przebiegu drogi, osobno dla obu sposobów lokomocji pieszo oraz samochodem (Rysunek 31). W ten sposób możliwe było dokładniejsze rozpatrywanie obu przypadków oraz minimalizacja rozbieżności długości, co na dłuższych odcinkach przekłada się na znaczne zmiany czasu przejazdu.



Rysunek 31. Wizualizacja wzajemnego przebiegu drogi dla przejścia pieszego (kolor zielony) oraz przejazdu samochodem (kolor czerwony) (źródło: opracowanie własne).

Utworzono również warstwy z punktami wyznaczającymi miejsce rozpoczęcia oraz zakończenia każdego z odcinków. Następnie za pomocą wtyczki wygenerowano atrybuty dla badanych tras. Wyniki zostały zestawione w postaci tabelarycznej (Tabela 8).

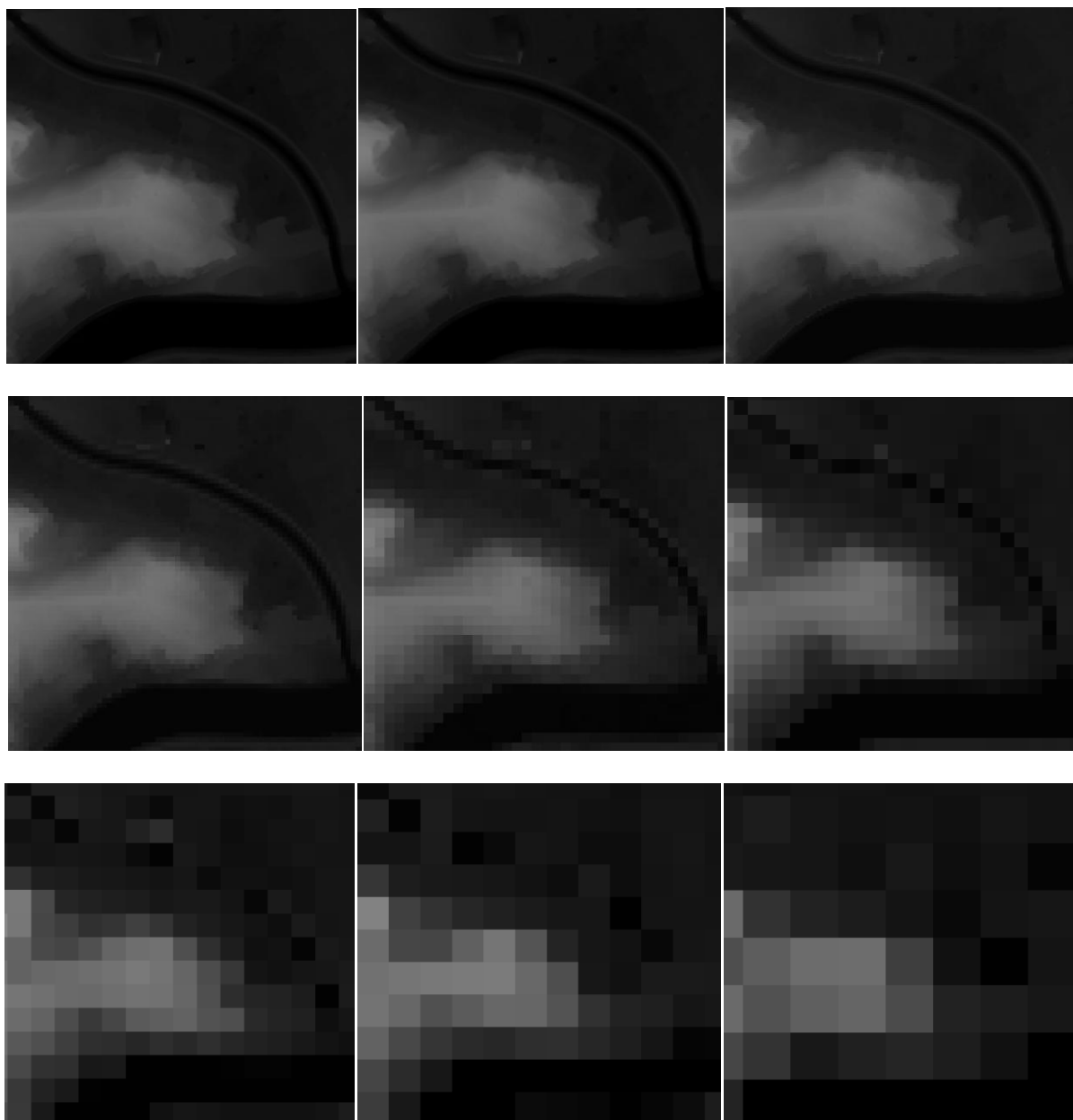
Tabela 8. Atrybuty dla odcinków trasy kontrolowanej wygenerowane na podstawowym rastrze o rozdzielczości 1x1m, rodzaj lokomocji: pieszo (po lewej), samochodem (po prawej).

Trasa	Czas [min]	Długość [km]
1	1.66	0.14
2	3.48	0.28
3	3.82	0.30
4	12.23	0.89
5	2.60	0.21
6	17.43	1.34
7	14.36	1.06
8	12.28	1.01
9	18.56	1.53

Trasa	Czas [min]	Długość [km]
1	0.21	0.14
2	0.41	0.28
3	0.44	0.29
4	1.33	0.89
5	0.31	0.21
6	2.01	1.34
7	1.62	1.08
8	1.52	1.02
9	2.32	1.55

5.2.3 Zmienna rozdzielczość rastra

W celu sprawdzenia wpływu zmiany rozmiaru komórki numerycznego modelu terenu na wartości generowanych wyników wykonano resampling. Jako algorytm interpolacji zastosowano dwie różne metody: najbliższego sąsiada, która przypisuje nowej komórce wartość piksela znajdującego się najbliżej jej środka oraz biliniową, bazującą na liczbach pochodzących z czterech najmniej odległych komórek. Utworzono zestaw rastrów o rozdzielczościach: 3x3, 5x5, 10x10, 20x20, 30x30, 50x50, 70x70, 100x100 m, które posłużyły jako źródło pozyskania wysokości dla warstwy liniowej. Wykonano wizualizację wyglądu wybranego fragmentu obrazu na różnych etapach generalizacji (Rysunek 32).



Rysunek 32. Wizualizacja wyglądu rozdzielczości komórek dla rastrów:
1x1, 3x3, 5x5, 10x10, 20x20, 30x30, 50x50, 70x70, 100x100 [m],
na przykładzie interpolacji metodą najbliższego sąsiada
(źródło: opracowanie własne).

Obniżanie rozdzielczości rastra powoduje mieszanie się informacji pochodzących z różnych komórek, w ten sposób powstają tak zwane miksele. W przypadku analiz wysokościowych wykorzystując resampling oraz określony rodzaj interpolacji wynikiem są wartości odpowiednio uśredniane, skalowane bądź filtrowane. Dla badanego obszaru piksele o wymiarze do 10 m pozwalają dobrze wyodrębnić wygląd przebiegu rzeki. Dla wartości 30 m lub więcej staje się to bardzo ciężkie, nie da się jednoznacznie wskazać dokładnego miejsca biegu cieku, który jest fragmentarycznie przerywany. Trudne do identyfikacji jest także wyznaczenie położenia jego brzegów oraz określenie szerokości. Informacja o wysokości jest szczególnie zaburzana w przypadku, gdy powierzchnia jest mocno zróżnicowana, czego przykładem w mieście może być występowanie: rzek, wzgórz, rowów, nasypów, wałów. Dlatego ważne jest dopasowanie sposobu interpolacji adekwatnie do powierzchni, a także posiadanie wiedzy na temat skali rozbieżności w wysokościach, które mogą się pojawić.

Wykorzystując aplikację Find Route wygenerowano trasy dla obrazu wejściowego oraz wszystkich utworzonych po przeprowadzeniu resamplingu. Wszędzie opierano się na tym samym zestawie punktów kolejności przebiegu oraz z zastosowaniem identycznych parametrów optymalizacji. Następnie odczytano wartości atrybutów, na podstawie których wykonano tabelaryczne zestawienie otrzymanych wyników (Tabela 9 - Tabela 24).

Tabela 9. Atrybuty wygenerowane na rastrze o rozdzielczości **3x3 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **najbliższego sąsiada**.

Nr	Czas [min]	Długość [km]
1	1.65	0.14
2	3.39	0.28
3	3.60	0.29
4	11.95	0.88
5	2.60	0.21
6	16.62	1.33
7	14.29	1.06
8	12.24	1.01
9	18.45	1.53

Nr	Czas [min]	Długość [km]
1	0.21	0.14
2	0.41	0.28
3	0.44	0.29
4	1.32	0.88
5	0.31	0.21
6	2.01	1.34
7	1.61	1.07
8	1.53	1.02
9	2.32	1.54

Tabela 10. Atrybuty wygenerowane na rastrze o rozdzielczości **5x5 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **najbliższego sąsiada**.

Nr	Czas [min]	Długość [km]
1	1.66	0.14
2	3.37	0.28
3	3.56	0.29
4	11.84	0.88
5	2.65	0.21
6	17.14	1.34
7	14.11	1.06
8	12.26	1.01
9	18.45	1.53

Nr	Czas [min]	Długość [km]
1	0.21	0.14
2	0.41	0.28
3	0.44	0.29
4	1.32	0.88
5	0.31	0.21
6	2.01	1.34
7	1.62	1.08
8	1.52	1.01
9	2.32	1.54

Tabela 11. Atrybuty wygenerowane na rastrze o rozdzielczości **10x10 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **najbliższego sąsiada**.

Nr	Czas [min]	Długość [km]
1	1.65	0.14
2	3.33	0.28
3	3.56	0.29
4	11.95	0.88
5	2.59	0.21
6	18.13	1.34
7	13.99	1.06
8	12.24	1.01
9	18.42	1.53

Nr	Czas [min]	Długość [km]
1	0.21	0.14
2	0.41	0.28
3	0.44	0.30
4	1.32	0.88
5	0.31	0.21
6	2.01	1.34
7	1.60	1.07
8	1.53	1.02
9	2.32	1.54

Tabela 12. Atrybuty wygenerowane na rastrze o rozdzielczości **20x20 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **najbliższego sąsiada**.

Nr	Czas [min]	Długość [km]
1	1.65	0.14
2	3.33	0.28
3	3.50	0.29
4	11.42	0.89
5	2.54	0.21
6	18.60	1.34
7	14.27	1.06
8	12.28	1.01
9	18.39	1.53

Nr	Czas [min]	Długość [km]
1	0.21	0.14
2	0.41	0.28
3	0.44	0.29
4	1.32	0.88
5	0.31	0.21
6	2.01	1.35
7	1.61	1.08
8	1.52	1.01
9	2.32	1.54

Tabela 13. Atrybuty wygenerowane na rastrze o rozdzielczości **30x30 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **najbliższego sąsiada**.

Nr	Czas [min]	Długość [km]
1	1.65	0.14
2	3.33	0.28
3	3.50	0.29
4	12.01	0.89
5	2.80	0.22
6	17.91	1.34
7	14.28	1.07
8	12.19	1.01
9	18.41	1.53

Nr	Czas [min]	Długość [km]
1	0.21	0.14
2	0.41	0.28
3	0.45	0.30
4	1.31	0.87
5	0.32	0.21
6	2.02	1.35
7	1.60	1.07
8	1.53	1.02
9	2.31	1.54

Tabela 14. Atrybuty wygenerowane na rastrze o rozdzielczości **50x50 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **najbliższego sąsiada**.

Nr	Czas [min]	Długość [km]
1	1.65	0.14
2	3.35	0.28
3	3.54	0.29
4	11.72	0.88
5	2.62	0.22
6	21.37	1.35
7	14.25	1.06
8	12.21	1.01
9	18.35	1.53

Nr	Czas [min]	Długość [km]
1	0.21	0.14
2	0.41	0.28
3	0.44	0.29
4	1.32	0.88
5	0.31	0.21
6	2.03	1.35
7	1.60	1.07
8	1.53	1.02
9	2.31	1.54

Tabela 15. Atrybuty wygenerowane na rastrze o rozdzielczości **70x70 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **najbliższego sąsiada**.

Nr	Czas [min]	Długość [km]
1	1.65	0.14
2	3.33	0.28
3	3.51	0.29
4	10.52	0.88
5	2.61	0.22
6	18.90	1.34
7	13.22	1.06
8	12.16	1.01
9	18.54	1.53

Nr	Czas [min]	Długość [km]
1	0.21	0.14
2	0.41	0.28
3	0.44	0.29
4	1.32	0.88
5	0.32	0.21
6	2.02	1.35
7	1.60	1.07
8	1.53	1.02
9	2.31	1.54

Tabela 16. Atrybuty wygenerowane na rastrze o rozdzielczości **100x100 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **najbliższego sąsiada**.

Nr	Czas [min]	Długość [km]
1	1.65	0.14
2	3.33	0.28
3	3.50	0.29
4	10.52	0.88
5	2.61	0.22
6	22.93	1.35
7	13.03	1.06
8	12.16	1.01
9	18.40	1.53

Nr	Czas [min]	Długość [km]
1	0.21	0.14
2	0.41	0.28
3	0.44	0.29
4	1.32	0.88
5	0.31	0.21
6	2.04	1.36
7	1.60	1.07
8	1.53	1.02
9	2.31	1.54

Tabela 17. Atrybuty wygenerowane na rastrze o rozdzielczości **3x3 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **biliniową**.

Nr	Czas [min]	Długość [km]
1	1.65	0.14
2	3.39	0.28
3	3.60	0.29
4	12.06	0.88
5	2.60	0.21
6	16.68	1.33
7	14.24	1.06
8	12.26	1.01
9	18.39	1.53

Nr	Czas [min]	Długość [km]
1	0.21	0.14
2	0.41	0.28
3	0.44	0.29
4	1.32	0.88
5	0.31	0.21
6	2.01	1.34
7	1.61	1.07
8	1.53	1.02
9	2.31	1.54

Tabela 18. Atrybuty wygenerowane na rastrze o rozdzielczości **5x5 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **biliniową**.

Nr	Czas [min]	Długość [km]
1	1.66	0.14
2	3.35	0.28
3	3.54	0.29
4	11.92	0.88
5	2.62	0.21
6	17.51	1.34
7	14.08	1.06
8	12.26	1.01
9	18.41	1.53

Nr	Czas [min]	Długość [km]
1	0.21	0.14
2	0.41	0.28
3	0.44	0.29
4	1.32	0.88
5	0.31	0.21
6	2.01	1.34
7	1.62	1.08
8	1.52	1.01
9	2.31	1.54

Tabela 19. Atrybuty wygenerowane na rastrze o rozdzielczości **10x10 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **biliniową**.

Nr	Czas [min]	Długość [km]
1	1.65	0.14
2	3.34	0.28
3	3.56	0.29
4	11.94	0.88
5	2.59	0.21
6	19.97	1.34
7	13.77	1.06
8	12.23	1.01
9	18.40	1.53

Nr	Czas [min]	Długość [km]
1	0.21	0.14
2	0.41	0.28
3	0.44	0.30
4	1.32	0.88
5	0.31	0.21
6	2.01	1.34
7	1.62	1.07
8	1.52	1.02
9	2.31	1.54

Tabela 20. Atrybuty wygenerowane na rastrze o rozdzielczości **20x20 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **biliniową**.

Nr	Czas [min]	Długość [km]
1	1.65	0.14
2	3.33	0.28
3	3.50	0.29
4	11.41	0.89
5	2.54	0.21
6	18.69	1.34
7	14.27	1.06
8	12.28	1.01
9	18.37	1.53

Nr	Czas [min]	Długość [km]
1	0.21	0.14
2	0.41	0.28
3	0.44	0.29
4	1.32	0.88
5	0.31	0.21
6	2.02	1.35
7	1.61	1.08
8	1.52	1.01
9	2.31	1.54

Tabela 21. Atrybuty wygenerowane na rastrze o rozdzielczości **30x30 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **biliniową**.

Nr	Czas [min]	Długość [km]
1	1.65	0.14
2	3.33	0.28
3	3.50	0.29
4	11.98	0.88
5	2.80	0.22
6	18.15	1.34
7	14.28	1.07
8	12.19	1.01
9	18.41	1.53

Nr	Czas [min]	Długość [km]
1	0.21	0.14
2	0.41	0.28
3	0.45	0.30
4	1.31	0.87
5	0.32	0.21
6	2.02	1.35
7	1.60	1.07
8	1.53	1.02
9	2.31	1.54

Tabela 22. Atrybuty wygenerowane na rastrze o rozdzielczości **50x50 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **biliniową**.

Nr	Czas [min]	Długość [km]
1	1.65	0.14
2	3.35	0.28
3	3.54	0.29
4	11.71	0.88
5	2.62	0.22
6	21.40	1.35
7	14.25	1.06
8	12.21	1.01
9	18.35	1.53

Nr	Czas [min]	Długość [km]
1	0.21	0.14
2	0.41	0.28
3	0.44	0.29
4	1.32	0.88
5	0.31	0.21
6	2.03	1.35
7	1.60	1.07
8	1.53	1.02
9	2.31	1.54

Tabela 23. Atrybuty wygenerowane na rastrze o rozdzielczości **70x70 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **biliniową**.

Nr	Czas [min]	Długość [km]
1	1.65	0.14
2	3.33	0.28
3	3.51	0.29
4	10.52	0.88
5	2.61	0.22
6	18.89	1.34
7	13.24	1.06
8	12.16	1.01
9	18.54	1.53

Nr	Czas [min]	Długość [km]
1	0.21	0.14
2	0.41	0.28
3	0.44	0.29
4	1.32	0.88
5	0.31	0.21
6	2.02	1.35
7	1.60	1.07
8	1.53	1.02
9	2.31	1.54

Tabela 24. Atrybuty wygenerowane na rastrze o rozdzielczości **100x100 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **biliniową**.

Nr	Czas [min]	Długość [km]
1	1.65	0.14
2	3.33	0.28
3	3.50	0.29
4	10.52	0.88
5	2.61	0.22
6	22.93	1.35
7	13.02	1.06
8	12.16	1.01
9	18.40	1.53

Nr	Czas [min]	Długość [km]
1	0.21	0.14
2	0.41	0.28
3	0.44	0.29
4	1.32	0.88
5	0.31	0.21
6	2.04	1.36
7	1.61	1.07
8	1.53	1.02
9	2.31	1.54

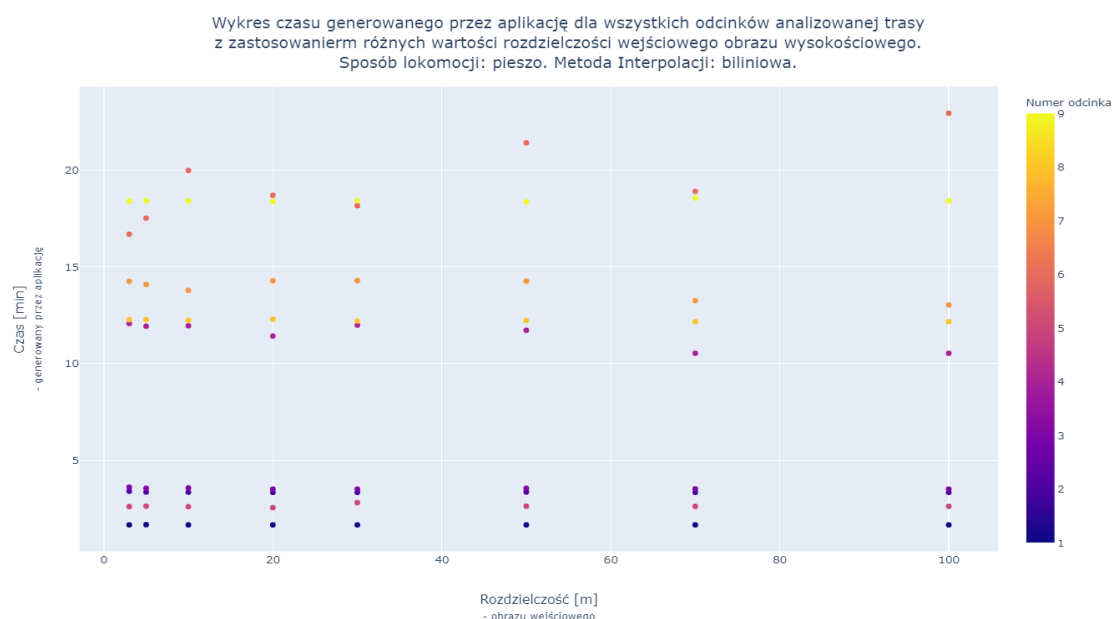
W trakcie generowania wyników zauważono, że aplikacja dla obrazów o komórce 1x1 m działała bardzo powoli, czas obliczeń wynosił średnio 30 s, dla rozmiaru 3x3 m około 5 s, natomiast w przypadku 5x5 m były to już 2 s. Dla mniejszych rozdzielczości, czyli rastrów o większym pikselu wyniki pozyskiwano szybciej niż w 1 s.

Po wykonaniu tabelarycznego zestawienia wartości opracowano wykresy punktowe pokazujące zachowanie się czasu względem zmieniającej się rozdzielczości dla dwóch metod interpolacji oraz różnych sposobów lokomocji (Rysunek 33 - Rysunek 36).

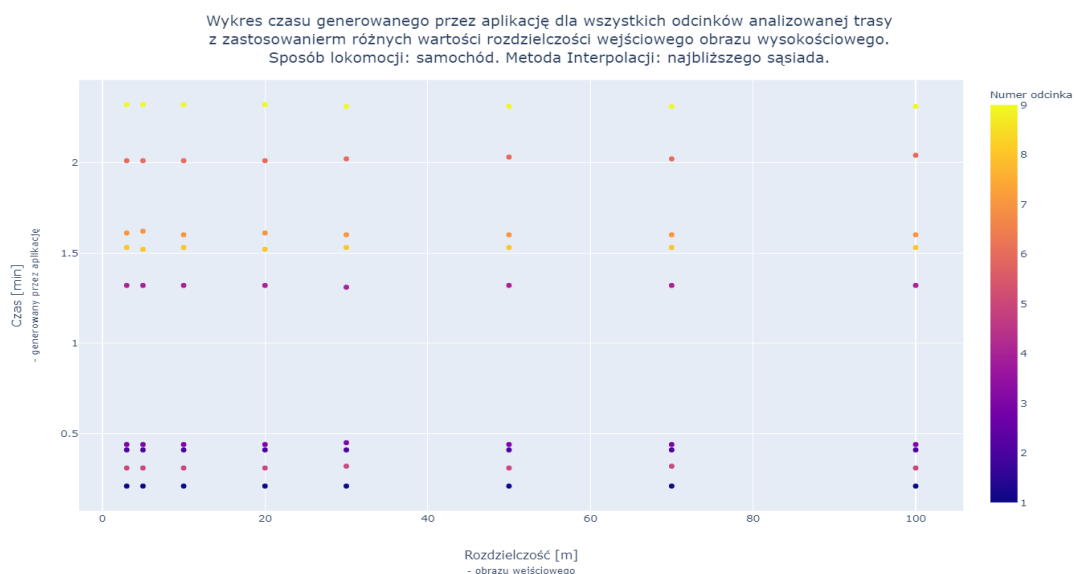
Jeżeli ułożenie punktów o tym samym kolorze przypomina bardziej jednolitą i prostą linię to oznacza, że wpływ zmiany rozdzielczości na generowane wartości jest mniejszy.



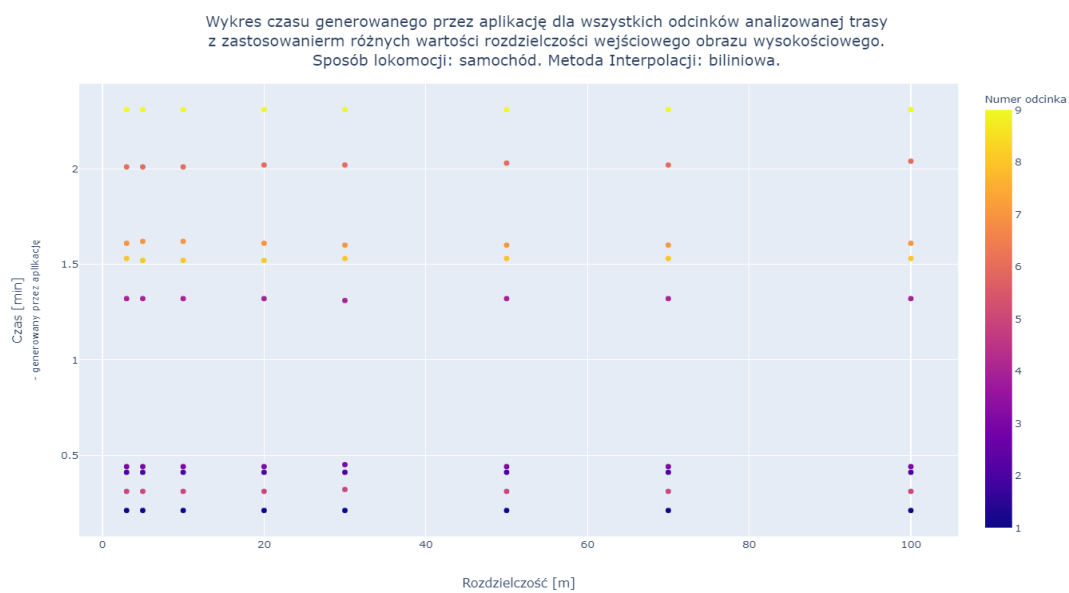
Rysunek 33. Wykres generowanego czasu względem różnych wartości rozdzielczości, sposób lokomocji: **pieszo**, metoda interpolacji: **najbliższego sąsiada** (źródło: opracowanie własne).



Rysunek 34. Wykres generowanego czasu względem różnych wartości rozdzielczości, sposób lokomocji: **pieszo**, metoda interpolacji: **biliniowa** (źródło: opracowanie własne).



Rysunek 35. Wykres generowanego czasu względem różnych wartości rozdzielczości, sposób lokomocji: **samochód**, metoda interpolacji: **najbliższego sąsiada** (źródło: opracowanie własne).



Rysunek 36. Wykres generowanego czasu względem różnych wartości rozdzielczości, sposób lokomocji: **samochód**, metoda interpolacji: **biliniowa** (źródło: opracowanie własne).

Wykresy pokazują, że zmiany rozdzielczości rastrów mają niewielki wpływ na czas generowany dla przemieszczania się samochodem, zmiany te osiągają zwykle 1-2 s. Większe rozbieżności pojawiają się w ruchu pieszym, który jest mocno związany nie tylko z odległością przestrzenną, ale również zmianami wysokości. Różnice osiągają w tym przypadku zarówno kilka sekund, jak i wartości minutowe. Taka sytuacja jest szczególnie widoczna w przypadku w przypadku trasy numer 6 dla obu rodzajów interpolacji. Wygląd rozmieszczenia punktów wskazuje na fakt, że długość częściowo wpływa na dokładność. Dla badanego przypadku odcinki dłuższe (4, 5, 7, 8, 9) charakteryzują się występowaniem większych odstępów czasowych niż pozostałe krótsze (1, 2, 3, 5). Największe zmiany można zauważyć dla trasy, która jest najbardziej różnorodna pod względem ukształtowania terenu oraz leży w pobliżu wzniesień, skarp i brzegów Wisły.

5.3. Wyniki badań

5.3.1 Poprawność działania algorytmów

Określenie stopnia wiarygodności wyników generowanych przez wtyczkę poza sprawdzeniem umiejętności wyznaczania odpowiednich połączeń pomiędzy zadanymi punktami wymaga wykonania analizy, pokazującej jak dokładnie obliczane są atrybuty tras. Najbardziej miarodajnym rozwiązaniem jest porównanie wartości pomierzonych w rzeczywistości oraz wyznaczonych przez aplikację (Tabela 6, Tabela 8). W tym celu wykonano zestawienie pokazujące różnice między wynikami z pomiaru terenowego oraz dla obrazu oryginalnego o rozdzielczości piksela 1x1 m (Tabela 25).

Tabela 25. Różnice atrybutów dla odcinków trasy kontrolnej pomierzonych w terenie oraz wygenerowanych na rastrze o rozdzielczości 1x1 m, rodzaj lokomocji: pieszo (po lewej), samochodem (po prawej).

Trasa	Czas [min]	Długość [km]	Trasa	Czas [min]	Długość [km]
1	-0.16	0.00	1	0.17	-0.01
2	-0.55	0.01	2	0.96	0.00
3	-0.32	0.02	3	-0.06	-0.03
4	-1.08	0.02	4	0.04	-0.07
5	-0.30	-0.03	5	0.87	0.00
6	-0.65	0.14	6	0.01	-0.03
7	-1.21	0.09	7	0.81	0.01
8	0.55	0.22	8	2.78	0.10
9	-0.11	-0.11	9	0.11	-0.06

Różnice dla długości wynikają z błędu podkładu mapowego, wektoryzacji oraz niedokładności wyznaczania lokalizacji telefonu. Największa rozbieżność pojawiła się dla odcinka 8, osiągając wartość 220 oraz 100 m, co mogło wynikać z powodu występowania na drodze przejść dla pieszych oraz konieczności omijania przechodniów lub zmiany pasa ruchu. Spodziewaną tendencją jest fakt, że częściej dłuższe trasy charakteryzują się większą różnicą odwzorowanej długości.

Odnosząc się do wartości odchylen czasu są one stosunkowo niewielkie. Tylko dwie trasy piesze: 4, 7 oraz jedna samochodowa: 8 wykazały rozbieżności większe niż 1 min. Dla trasy 4, 7 przyczyna mogła leżeć w nakładaniu się błędów nierównego tempa chodu, natomiast dla 8 wynik był spowodowany koniecznością zatrzymywania się w korkach oraz sygnalizacją świetlną. Zauważalna jest także skłonność do generowania zawyżonych predykcji czasu dla ruchu pieszego oraz zaniżonych dla samochodowego.

Wykonano także obliczenie błędów średnich dla odpowiednich parametrów (10). W wyniku czego uzyskano wartości: 39 s, 100 m oraz 1 min 3 s, 50 m. Wyeliminowanie ze zbiorów po jednej najgorszej obserwacji znacznie zmniejszyło wartości błędów na: 33 s, 70 m oraz 33 s i 30 m. Dla zbiorów odcinków do 1 km błędy dla obu rodzajów lokomocji prezentują się w następujący sposób: 35 s, 20 m i 35 s, 30 m, natomiast dla dłuższych: 45 s, 150 m oraz 1 min 27 s, 60 m.

Wzór na błąd średni:

$$m_{\bar{s}r} = \sqrt{\frac{\sum vv}{n}} \quad (10)$$

gdzie:

$m_{\bar{s}r}$ - błąd średni

v - różnica między wartością wygenerowaną a rzeczywistą

n - liczba obserwacji

5.3.2 Wiarygodność wyników dla różnego rozmiaru piksela

Dla sprawdzenia zachowania się generowanych atrybutów przy stopniowej zmianie rozdzielczości komórek wykonano zestawienie różnic dla czasu oraz długości przestrzennej względem wartości z pomiaru terenowego (Tabela 26 - Tabela 41). Dodatkowo obliczono i zestawiono w tabelach również błędy średnie (Tabela 42, Tabela 43).

Tabela 26. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości **3x3 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **najbliższego sąsiada**.

Nr	Czas [min]	Długość [km]
1	-0.15	0.00
2	-0.46	0.01
3	-0.10	0.03
4	-0.80	0.03
5	-0.30	-0.03
6	0.16	0.15
7	-1.14	0.09
8	0.59	0.22
9	0.00	-0.11

Nr	Czas [min]	Długość [km]
1	0.17	-0.01
2	0.96	0.00
3	-0.06	-0.03
4	0.05	-0.06
5	0.87	0.00
6	0.01	-0.03
7	0.82	0.02
8	2.77	0.10
9	0.11	-0.05

Tabela 27. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości **5x5 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **najbliższego sąsiada**.

Nr	Czas [min]	Długość [km]
1	-0.16	0.00
2	-0.44	0.01
3	-0.06	0.03
4	-0.69	0.03
5	-0.35	-0.03
6	-0.36	0.14
7	-0.96	0.09
8	0.57	0.22
9	0.00	-0.11

Nr	Czas [min]	Długość [km]
1	0.17	-0.01
2	0.96	0.00
3	-0.06	-0.03
4	0.05	-0.06
5	0.87	0.00
6	0.01	-0.03
7	0.81	0.01
8	2.78	0.11
9	0.11	-0.05

Tabela 28. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości **10x10 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **najbliższego sąsiada**.

Nr	Czas [min]	Długość [km]
1	-0.15	0.00
2	-0.40	0.01
3	-0.06	0.03
4	-0.80	0.03
5	-0.29	-0.03
6	-1.35	0.14
7	-0.84	0.09
8	0.59	0.22
9	0.03	-0.11

Nr	Czas [min]	Długość [km]
1	0.17	-0.01
2	0.96	0.00
3	-0.06	-0.04
4	0.05	-0.06
5	0.87	0.00
6	0.01	-0.03
7	0.83	0.02
8	2.77	0.10
9	0.11	-0.05

Tabela 29. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości **20x20 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **najbliższego sąsiada**.

Nr	Czas [min]	Długość [km]
1	-0.15	0.00
2	-0.40	0.01
3	0.00	0.03
4	-0.27	0.02
5	-0.24	-0.03
6	-1.82	0.14
7	-1.12	0.09
8	0.55	0.22
9	0.06	-0.11

Nr	Czas [min]	Długość [km]
1	0.17	-0.01
2	0.96	0.00
3	-0.06	-0.03
4	0.05	-0.06
5	0.87	0.00
6	0.01	-0.04
7	0.82	0.01
8	2.78	0.11
9	0.11	-0.05

Tabela 30. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości **30x30 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **najbliższego sąsiada**.

Nr	Czas [min]	Długość [km]
1	-0.15	0.00
2	-0.40	0.01
3	0.00	0.03
4	-0.86	0.02
5	-0.50	-0.04
6	-1.13	0.14
7	-1.13	0.08
8	0.64	0.22
9	0.04	-0.11

Nr	Czas [min]	Długość [km]
1	0.17	-0.01
2	0.96	0.00
3	-0.07	-0.04
4	0.06	-0.05
5	0.86	0.00
6	0.00	-0.04
7	0.83	0.02
8	2.77	0.10
9	0.12	-0.05

Tabela 31. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości **50x50 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **najbliższego sąsiada**.

Nr	Czas [min]	Długość [km]
1	-0.15	0.00
2	-0.42	0.01
3	-0.04	0.03
4	-0.57	0.03
5	-0.32	-0.04
6	-4.59	0.13
7	-1.10	0.09
8	0.62	0.22
9	0.10	-0.11

Nr	Czas [min]	Długość [km]
1	0.17	-0.01
2	0.96	0.00
3	-0.06	-0.03
4	0.05	-0.06
5	0.87	0.00
6	-0.01	-0.04
7	0.83	0.02
8	2.77	0.10
9	0.12	-0.05

Tabela 32. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości **70x70 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **najbliższego sąsiada**.

Nr	Czas [min]	Długość [km]
1	-0.15	0.00
2	-0.40	0.01
3	-0.01	0.03
4	0.63	0.03
5	-0.31	-0.04
6	-2.12	0.14
7	-0.07	0.09
8	0.67	0.22
9	-0.09	-0.11

Nr	Czas [min]	Długość [km]
1	0.17	-0.01
2	0.96	0.00
3	-0.06	-0.03
4	0.05	-0.06
5	0.86	0.00
6	0.00	-0.04
7	0.83	0.02
8	2.77	0.10
9	0.12	-0.05

Tabela 33. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości **100x100 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **najbliższego sąsiada**.

Nr	Czas [min]	Długość [km]
1	-0.15	0.00
2	-0.40	0.01
3	0.00	0.03
4	0.63	0.03
5	-0.31	-0.04
6	-6.15	0.13
7	0.12	0.09
8	0.67	0.22
9	0.05	-0.11

Nr	Czas [min]	Długość [km]
1	0.17	-0.01
2	0.96	0.00
3	-0.06	-0.03
4	0.05	-0.06
5	0.87	0.00
6	-0.02	-0.05
7	0.83	0.02
8	2.77	0.10
9	0.12	-0.05

Tabela 34. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości **3x3 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **biliniową**.

Nr	Czas [min]	Długość [km]
1	-0.15	0.00
2	-0.46	0.01
3	-0.10	0.03
4	-0.91	0.03
5	-0.30	-0.03
6	0.10	0.15
7	-1.09	0.09
8	0.57	0.22
9	0.06	-0.11

Nr	Czas [min]	Długość [km]
1	0.17	-0.01
2	0.96	0.00
3	-0.06	-0.03
4	0.05	-0.06
5	0.87	0.00
6	0.01	-0.03
7	0.82	0.02
8	2.77	0.10
9	0.12	-0.05

Tabela 35. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości **5x5 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **biliniową**.

Nr	Czas [min]	Długość [km]
1	-0.16	0.00
2	-0.42	0.01
3	-0.04	0.03
4	-0.77	0.03
5	-0.32	-0.03
6	-0.73	0.14
7	-0.93	0.09
8	0.57	0.22
9	0.04	-0.11

Nr	Czas [min]	Długość [km]
1	0.17	-0.01
2	0.96	0.00
3	-0.06	-0.03
4	0.05	-0.06
5	0.87	0.00
6	0.01	-0.03
7	0.81	0.01
8	2.78	0.11
9	0.12	-0.05

Tabela 36. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości **10x10 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **biliniową**.

Nr	Czas [min]	Długość [km]
1	-0.15	0.00
2	-0.41	0.01
3	-0.06	0.03
4	-0.79	0.03
5	-0.29	-0.03
6	-3.19	0.14
7	-0.62	0.09
8	0.60	0.22
9	0.05	-0.11

Nr	Czas [min]	Długość [km]
1	0.17	-0.01
2	0.96	0.00
3	-0.06	-0.04
4	0.05	-0.06
5	0.87	0.00
6	0.01	-0.03
7	0.81	0.02
8	2.78	0.10
9	0.12	-0.05

Tabela 37. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości **20x20 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **biliniową**.

Nr	Czas [min]	Długość [km]
1	-0.15	0.00
2	-0.40	0.01
3	0.00	0.03
4	-0.26	0.02
5	-0.24	-0.03
6	-1.91	0.14
7	-1.12	0.09
8	0.55	0.22
9	0.08	-0.11

Nr	Czas [min]	Długość [km]
1	0.17	-0.01
2	0.96	0.00
3	-0.06	-0.03
4	0.05	-0.06
5	0.87	0.00
6	0.00	-0.04
7	0.82	0.01
8	2.78	0.11
9	0.12	-0.05

Tabela 38. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości **30x30 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **biliniową**.

Nr	Czas [min]	Długość [km]
1	-0.15	0.00
2	-0.40	0.01
3	0.00	0.03
4	-0.83	0.03
5	-0.50	-0.04
6	-1.37	0.14
7	-1.13	0.08
8	0.64	0.22
9	0.04	-0.11

Nr	Czas [min]	Długość [km]
1	0.17	-0.01
2	0.96	0.00
3	-0.07	-0.04
4	0.06	-0.05
5	0.86	0.00
6	0.00	-0.04
7	0.83	0.02
8	2.77	0.10
9	0.12	-0.05

Tabela 39. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości **50x50 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **biliniową**.

Nr	Czas [min]	Długość [km]
1	-0.15	0.00
2	-0.42	0.01
3	-0.04	0.03
4	-0.56	0.03
5	-0.32	-0.04
6	-4.62	0.13
7	-1.10	0.09
8	0.62	0.22
9	0.10	-0.11

Nr	Czas [min]	Długość [km]
1	0.17	-0.01
2	0.96	0.00
3	-0.06	-0.03
4	0.05	-0.06
5	0.87	0.00
6	-0.01	-0.04
7	0.83	0.02
8	2.77	0.10
9	0.12	-0.05

Tabela 40. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości **70x70 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **biliniową**.

Nr	Czas [min]	Długość [km]
1	-0.15	0.00
2	-0.40	0.01
3	-0.01	0.03
4	0.63	0.03
5	-0.31	-0.04
6	-2.11	0.14
7	-0.09	0.09
8	0.67	0.22
9	-0.09	-0.11

Nr	Czas [min]	Długość [km]
1	0.17	-0.01
2	0.96	0.00
3	-0.06	-0.03
4	0.05	-0.06
5	0.87	0.00
6	0.00	-0.04
7	0.83	0.02
8	2.77	0.10
9	0.12	-0.05

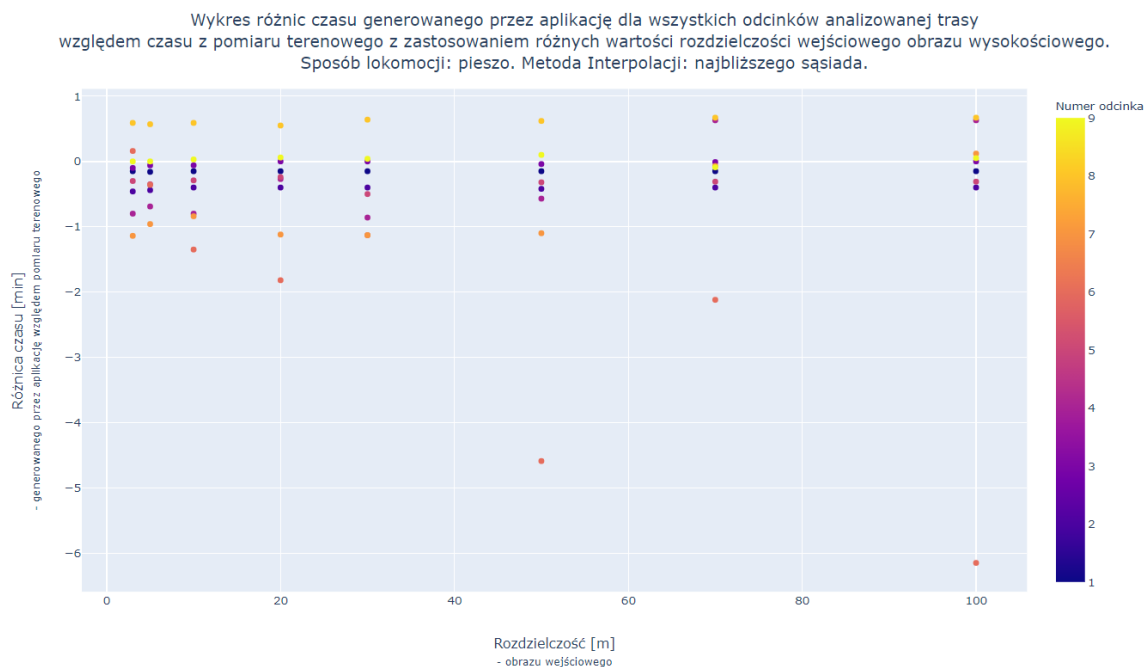
Tabela 41. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości **100x100 m**, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą **biliniową**.

Nr	Czas [min]	Długość [km]
1	-0.15	0.00
2	-0.40	0.01
3	0.00	0.03
4	0.63	0.03
5	-0.31	-0.04
6	-6.15	0.13
7	0.13	0.09
8	0.67	0.22
9	0.05	-0.11

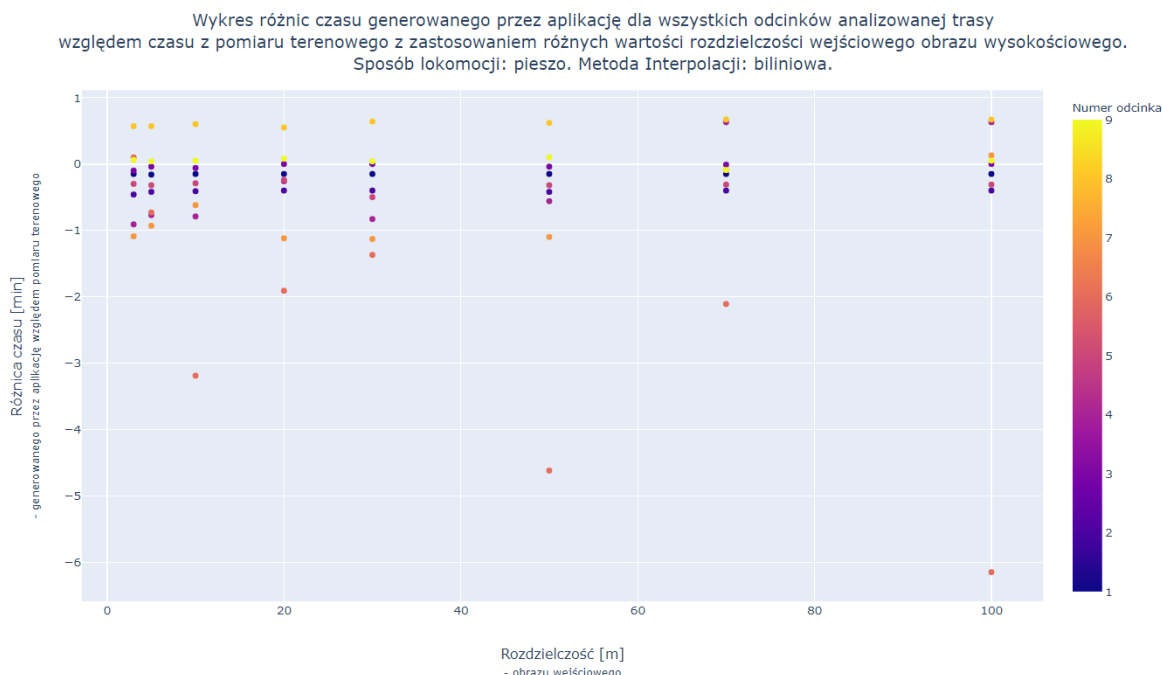
Nr	Czas [min]	Długość [km]
1	0.17	-0.01
2	0.96	0.00
3	-0.06	-0.03
4	0.05	-0.06
5	0.87	0.00
6	-0.02	-0.05
7	0.82	0.02
8	2.77	0.10
9	0.12	-0.05

Po wykonaniu tabelarycznego zestawienia różnic wartości generowanych przy zmiennej rozdzielczości i tych, które zostały pozyskane w terenie, opracowano wykresy punktowe pokazujące zachowanie się różnic czasu względem zmieniającej się rozdzielczości dla obu metod interpolacji oraz lokomocji pieszej i samochodem (Rysunek 37 - Rysunek 40).

Jeżeli ułożenie punktów o tym samym kolorze przypomina bardziej jednolitą i prostą linię, tym wpływ zmiany rozdzielczości na różnice czasu jest mniejszy.



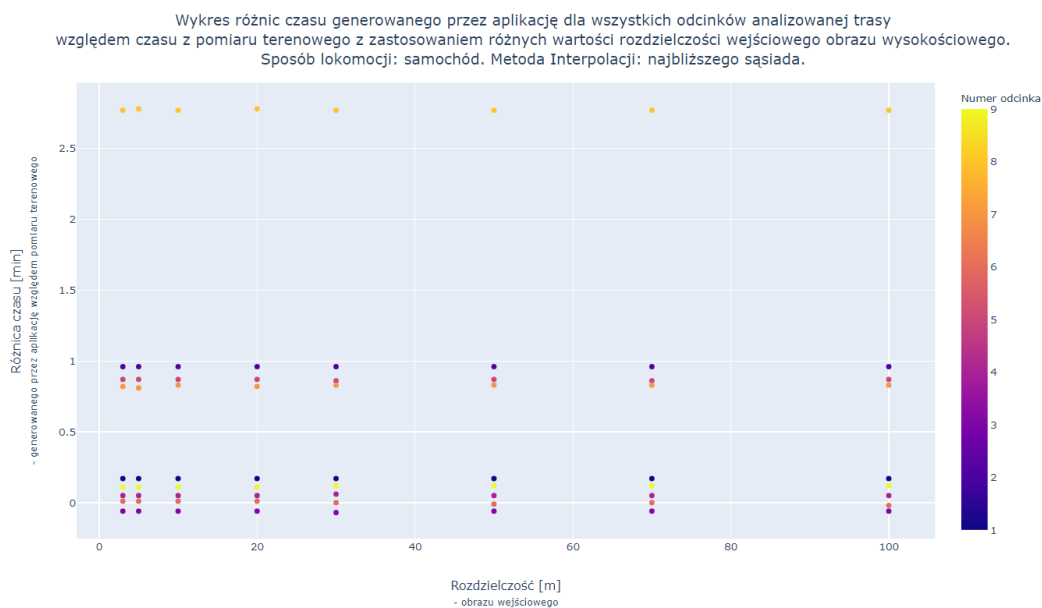
Rysunek 37. Wykres różnic generowanego czasu dla różnych wartości rozdzielczości względem pomiaru terenowego, sposób lokomocji: **pieszo**, metoda interpolacji: **najbliższego sąsiada** (źródło: opracowanie własne).



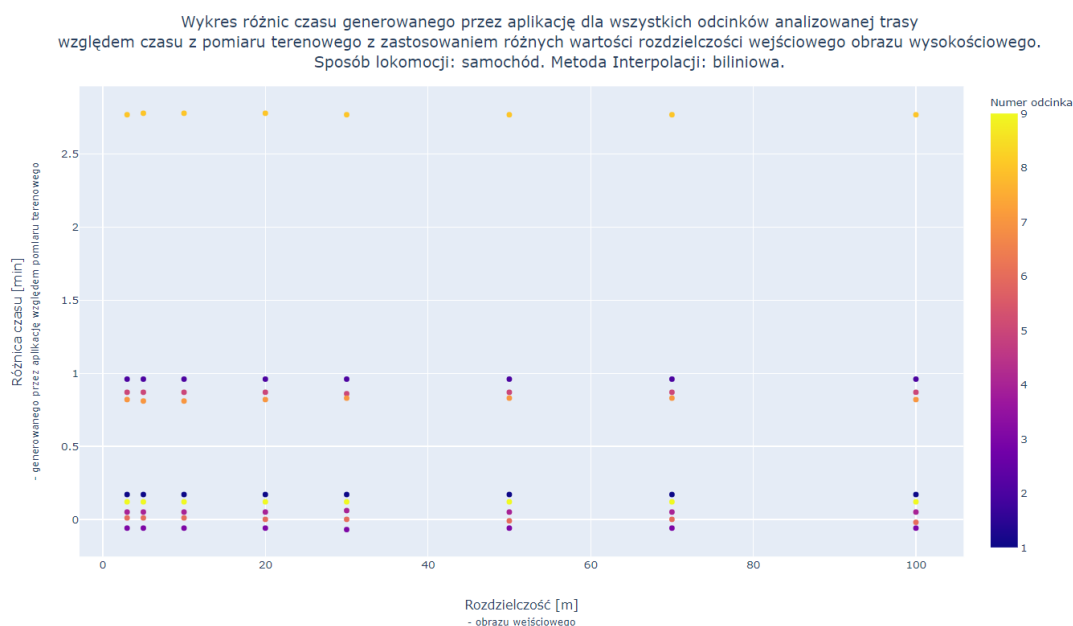
Rysunek 38. Wykres różnic generowanego czasu dla różnych wartości rozdzielczości względem pomiaru terenowego, sposób lokomocji: **pieszo**, metoda interpolacji: **biliniowa** (źródło: opracowanie własne).

Wykresy (Rysunek 37, Rysunek 38) pokazują, że dla trasy 6 i obrazu o komórce 100x100 m różnica osiągnęła wartość ponad 6 min, dla 70x70 m zmniejszyła się do około 2 min, a dla obrazu o pikselu 50x50 m powiększyła się w przybliżeniu do 4.5 min. Przykład ten ilustruje dużą losowość wpływu działania zastosowanych metod interpolacji.

Przyczyną tak wysokich rozbieżności dla danego fragmentu trasy mogło być jej bliskie położenie względem rzeki Rudawy i jej wałów, a także zróżnicowane ukształtowanie terenu oraz sąsiedztwo skarp znajdujących się na znacznie podwyższonym terenie pod Kopcem Kościuszki. Natomiast najlepiej odwzorowywanymi odcinkami dla ruchu pieszego okazały się 3 i 9.



Rysunek 39. Wykres różnic generowanego czasu dla różnych wartości rozdzielczości względem pomiaru terenowego, sposób lokomocji: **samochód**, metoda interpolacji: **najbliższego sąsiada** (źródło: opracowanie własne).



Rysunek 40. Wykres różnic generowanego czasu dla różnych wartości rozdzielczości względem pomiaru terenowego, sposób lokomocji: **samochód**, metoda interpolacji: **biliniowa** (źródło: opracowanie własne).

Odnosząc się do ruchu samochodowego największa różnica pojawiła się przy odcinku 8, który charakteryzował się występowaniem natężonego ruchu drogowego, sygnalizacji świetlnej oraz fragmentarycznymi korkami. Doprowadziło to do znacznego wydłużenia przejazdu oraz braku możliwości płynnego i jednostajnego przemieszczania się. Najmniejsze rozbieżności osiągnęły odcinki 3 i 4, gdzie warunki na drodze umożliwiły bezproblemowe pokonanie trasy, bez konieczności zatrzymywania się bądź znacznej zmiany prędkości.

Tabela 42. Błędy średnie czasu i długości dla kolejnych odcinków trasy z zastosowaniem wszystkich rozdzielczości, rodzaj lokomocji: pieszo (po lewej), samochodem (po prawej), metoda interpolacji: najbliższego sąsiada.

Błąd średni dla odcinków		
Trasa	Czas [min]	Długość [m]
1	0.15	0.00
2	0.41	0.01
3	0.05	0.03
4	0.68	0.03
5	0.34	0.04
6	2.95	0.14
7	0.91	0.09
8	0.62	0.22
9	0.06	0.11

Błąd średni dla odcinków		
Trasa	Czas [min]	Długość [m]
1	0.17	0.01
2	0.96	0.00
3	0.06	0.03
4	0.05	0.06
5	0.87	0.00
6	0.01	0.04
7	0.83	0.02
8	2.77	0.10
9	0.12	0.05

Tabela 43. Błędy średnie czasu i długości dla kolejnych odcinków trasy z zastosowaniem wszystkich rozdzielczości, rodzaj lokomocji: pieszo (po lewej), samochodem (po prawej), metoda interpolacji: biliniowa.

Błąd średni dla odcinków		
Trasa	Czas [min]	Długość [m]
1	0.15	0.00
2	0.41	0.01
3	0.05	0.03
4	0.70	0.03
5	0.33	0.04
6	3.16	0.14
7	0.88	0.09
8	0.62	0.22
9	0.07	0.11

Błąd średni dla odcinków		
Trasa	Czas [min]	Długość [m]
1	0.17	0.01
2	0.90	0.00
3	0.05	0.03
4	0.05	0.06
5	0.82	0.00
6	0.01	0.04
7	0.78	0.02
8	2.62	0.10
9	0.12	0.05

Wyniki analiz błędów średnich potwierdzają spostrzeżenia wyciągnięte z maksymalnych i minimalnych różnic między wartościami pozyskanymi z wykorzystaniem wtyczki oraz z pomiaru w terenie. Dodatkowo można zauważyć, że wyniki dla interpolacji metodą najbliższego sąsiada oraz biliniową wykazują podobny trend oraz są do siebie bardzo zbliżone. Największe różnice pojawiają się dla trasy numer 6 o najbardziej urozmaiconej formie terenu i jest to zauważalne wyłącznie w przypadku ruchu pieszego. Zsumowane błędy dla wszystkich odcinków i różnych rodzajów interpolacji wykazują, że w odniesieniu do czasu dla ruchu pieszego lepsza jest metoda najbliższego sąsiada, natomiast dla przemieszczania się samochodem biliniowa (Tabela 44). W obu przypadkach rozbieżności wyniosły około 4 s, co oznacza bardzo niewielkie odchylenie.

Tabela 44. Zsumowane błędy średnie czasu dla wszystkich odcinków trasy względem zastosowanej metody interpolacji.

Metoda interpolacji	Czas – pieszo [min]	Czas samochodem [min]
Najbliższego sąsiada	1.09	1.06
Biliniowa	1.15	1.00

Tabela 45. Błędy średnie czasu i długości dla wszystkich odcinków trasy w zależności od rozdzielczości rastra, rodzaj lokomocji: pieszo (po lewej), samochodem (po prawej), metoda interpolacji: najbliższego sąsiada.

Błąd średni		
Trasa	Czas [min]	Długość [m]
3x3	0.54	0.10
5x5	0.49	0.10
10x10	0.65	0.10
20x20	0.76	0.10
30x30	0.68	0.10
50x50	1.61	0.10
70x70	0.79	0.10
100x100	2.08	0.10

Błąd średni		
Trasa	Czas [min]	Długość [m]
3x3	1.06	0.05
5x5	1.06	0.05
10x10	1.06	0.05
20x20	1.06	0.05
30x30	1.06	0.05
50x50	1.06	0.05
70x70	1.06	0.05
100x100	1.06	0.05

Tabela 46. Błędy średnie czasu i długości dla wszystkich odcinków trasy w zależności od rozdzielczości rastra, rodzaj lokomocji: pieszo (po lewej), samochodem (po prawej), metoda interpolacji: biliniowa.

Błąd średni		
Trasa	Czas [min]	Długość [m]
3x3	0.55	0.10
5x5	0.54	0.10
10x10	1.15	0.10
20x20	0.78	0.10
30x30	0.72	0.10
50x50	1.62	0.10
70x70	0.79	0.10
100x100	2.08	0.10

Błąd średni		
Trasa	Czas [min]	Długość [m]
3x3	1.06	0.05
5x5	1.06	0.05
10x10	1.06	0.05
20x20	1.06	0.05
30x30	1.06	0.05
50x50	1.06	0.05
70x70	1.06	0.05
100x100	1.06	0.05

Zmiana rozdzielczości rastra z wykorzystaniem metody najbliższego sąsiada oraz biliniowej dla ruchu pieszego w większości przypadków jest powodem generowania wyższych błędów średnich. Natomiast w przypadku ruchu samochodowego widoczna jest stabilność występujących błędów. Najlepsze wyniki uzyskano dla rozdzielczości 5x5 m, natomiast najgorsze dla 100x100 m. W tabelach (Tabela 45, Tabela 46) kolorem zielonym oznaczono wartości najniższe, a czerwonym najwyższe.

Oba rodzaje interpolacji pokazują podobne zachowanie się wraz ze zmianą rozdzielczości. Wyjątkiem jest miejsce gdzie metoda biliniowa dla rozdzielczości 10x10 m oraz ruchu pieszego wykazuje widocznie większy błąd wynoszący o 30 s więcej niż dla metody najbliższego sąsiada. Zsumowane błędy średnie czasu wykazują różnicę w przypadku ruchu pieszego wynoszącą niecałe 4 s (Tabela 47).

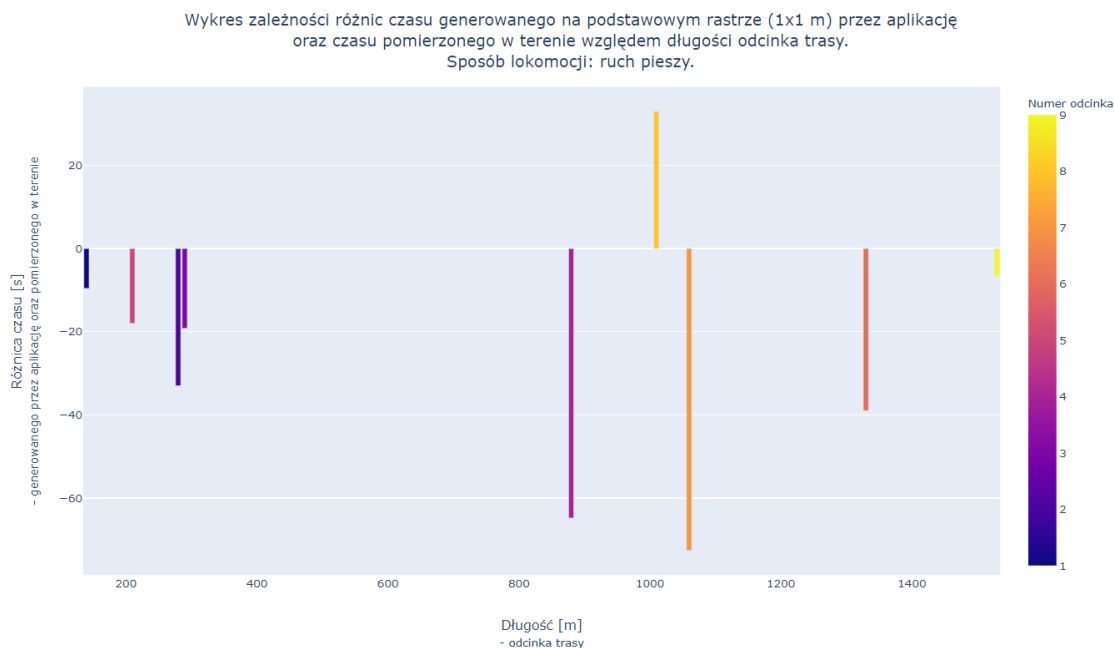
Tabela 47. Zsumowane błędy średnie czasu dla różnych rozdzielczości i metod interpolacji.

Metoda interpolacji	Czas – pieszo [min]	Czas samochodem [min]
Najbliższego sąsiada	1.03	1.06
Biliniowa	1.09	1.06

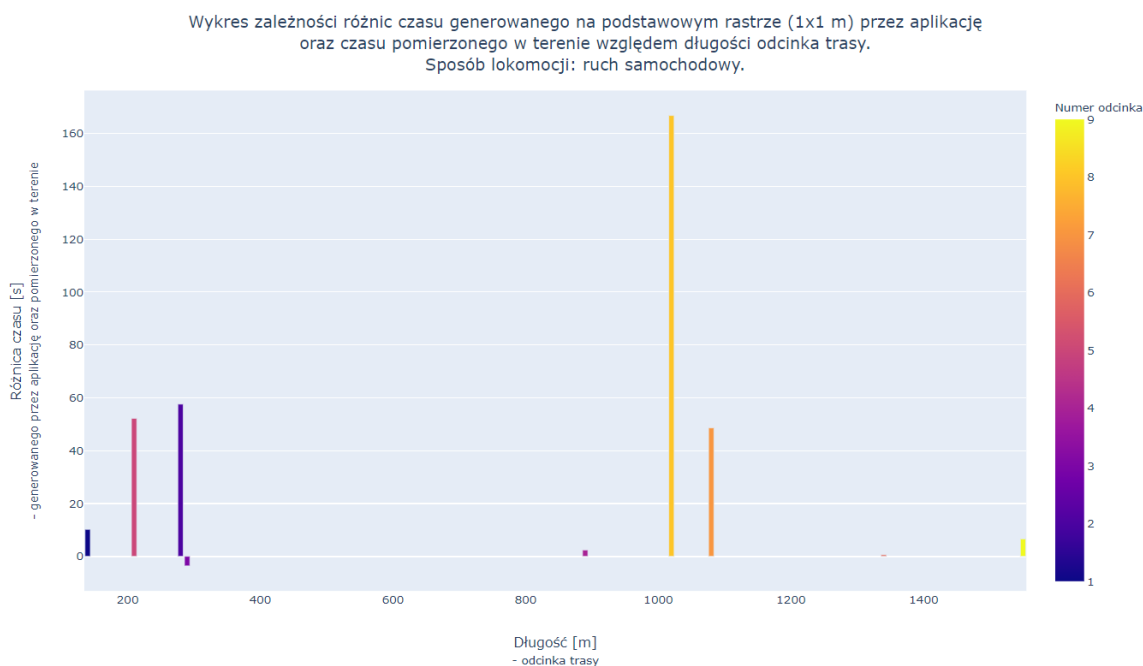
6. Analiza dokładności

W celu ułatwienia zrozumienia wartości uzyskanych podczas przeprowadzanych badań wykonano podsumowującą analizę dokładności.

Rysunek 41 i Rysunek 42 pokazują różnice czasu między wynikami generowanymi przez wtyczkę (na podstawowym rastrze o rozdzielczości 1x1 m) a otrzymanymi z pomiaru w terenie w zależności od długości danego fragmentu trasy (Tabela 6, Tabela 8, Tabela 25). Jeżeli słupek przyporządkowany do danego numeru trasy jest mniejszy to oznacza, że odchylenia czasu dla tej trasy są niższe.

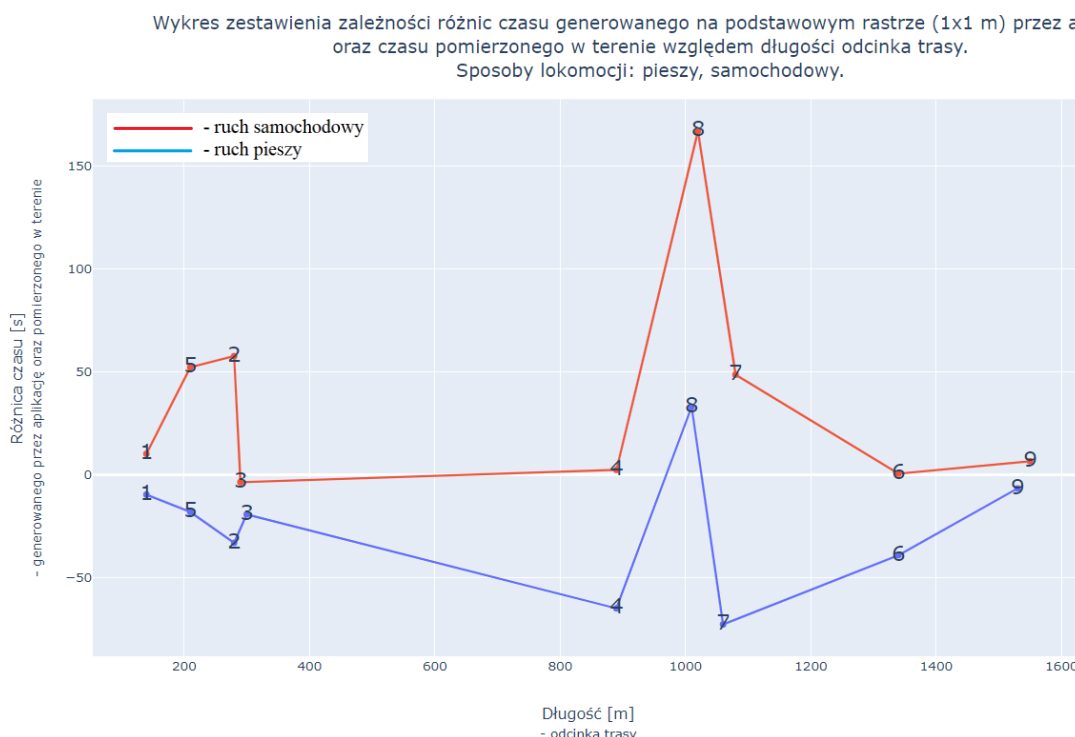


Rysunek 41. Wykres zależności różnic czasu od długości odcinka trasy – ruch pieszy
(źródło: opracowanie własne).



Rysunek 42. Wykres zależności różnic czasu od długości odcinka trasy – ruch samochodowy
(źródło: opracowanie własne).

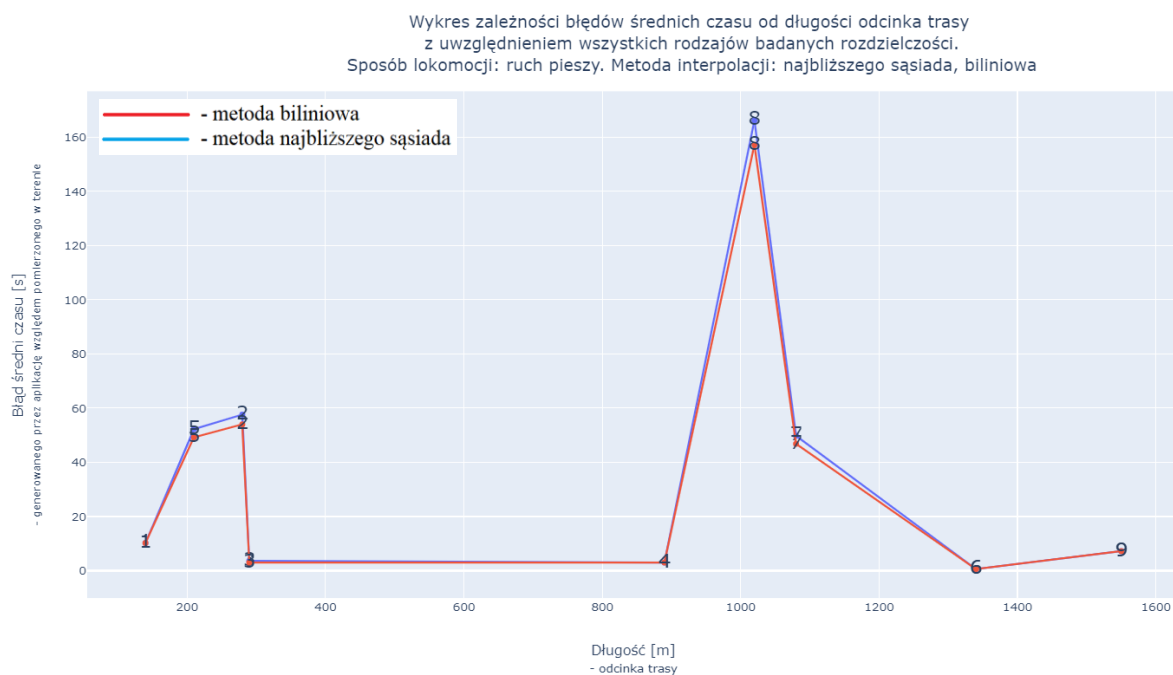
W przypadku ruchu pieszego (Rysunek 41) widoczna jest tendencja zawyżania wartości generowanych przez aplikację a także zauważalny jest wpływ długości odcinka na występowanie większych odchyłeń. Wykres dla przemieszczania się samochodem (Rysunek 42) pokazuje odwrotny trend względem czasu przejazdu szacowanego za pomocą wtyczki a wynikami z pomiaru w terenie, co oznacza zaniżanie wyliczanych wartości. Odnosząc się do występujących odchyłeń względem długości odcinków, widoczna jest ich pewna losowość. Zestawienie wyników dla obu metod lokomocji zostało przedstawione na zbiorczym wykresie (Rysunek 43). Numery oznaczają odpowiedni odcinek trasy.



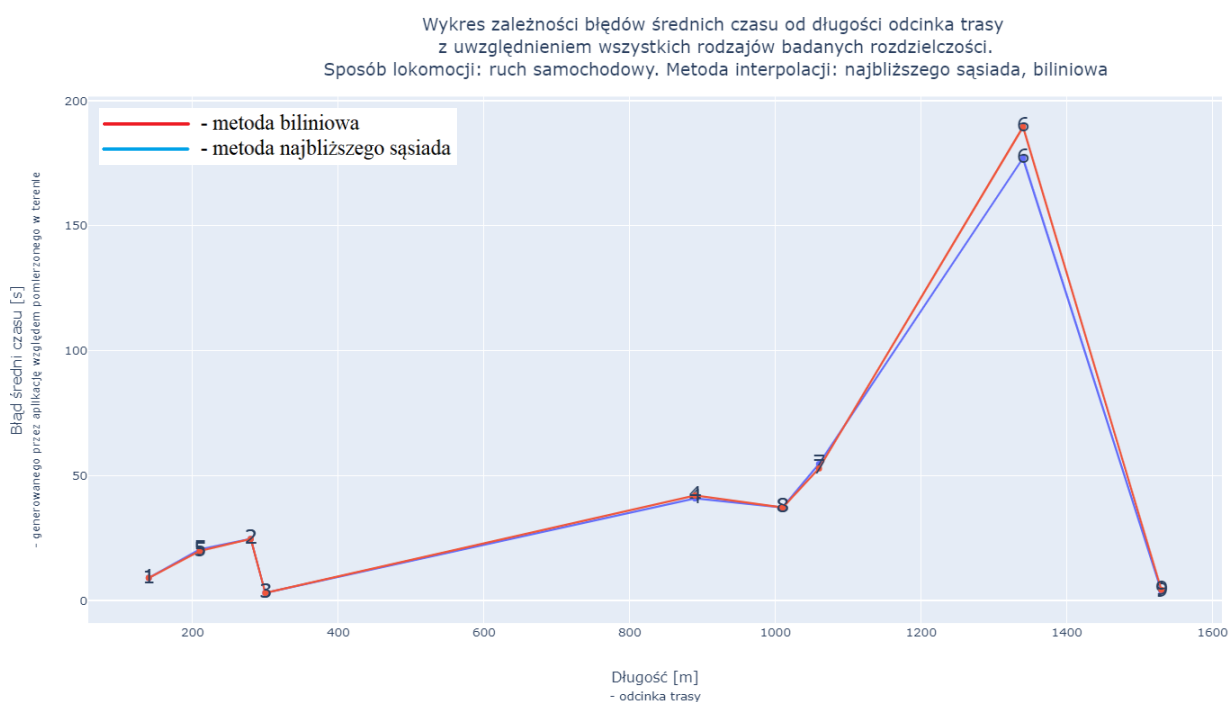
Rysunek 43. Wykres zależności różnic czasu od długości odcinka trasy
– ruch pieszego (niebieski), samochodowy (czerwony)
(źródło: opracowanie własne).

Powyższe zestawienie odchyłeń wykazuje pewną zależność danych. Dla większości odcinków możliwe jest zaobserwowanie zjawiska lustrzanego odbicia względem osi zerowej różnicy czasu, a w szczególności jest to widoczne dla numerów: 1, 2, 7, 9. Dodatkowo możliwe do zaobserwowania jest odwrotne zachowywanie się powiązanych par kolejnych fragmentów trasy dla ruchu pieszego oraz samochodowego, czego przykładem mogą być relacje między: 1-5, 5-2, 2-3, 3-4, 7-6. Silnie wyróżniającymi się obserwacjami są różnice otrzymane dla odcinka 8, który w obu przypadkach został niedoszacowany.

Wygenerowano także wykresy odwzorowujące zależności błędów średnich czasu w odniesieniu do długości odcinków dla dwóch rodzajów interpolacji oraz typów lokomocji (Rysunek 44, Rysunek 45). Wyniki pokazują, że dla badanego obszaru zastosowanie metody najbliższego sąsiada, czy biliniowej nie ma znaczącego wpływu na generowane wartości, ponieważ linie dla obu rodzajów interpolacji znajdują się przeważnie bardzo blisko siebie. Dla ruchu pieszego zauważalna jest rozbieżność na odcinku numer 6, co świadczy o tym, że większe odchylenia wykazują fragmenty położone na bardziej urozmaiconym terenie w pobliżu silnie wypukłych lub wklęsłych form. W przypadku przejazdu samochodowego widoczne jest zjawisko występowania większych rozbieżności między metodami interpolacji dla odcinków o najwyższych wartościach błędów średnich (5, 2, 8, 7).



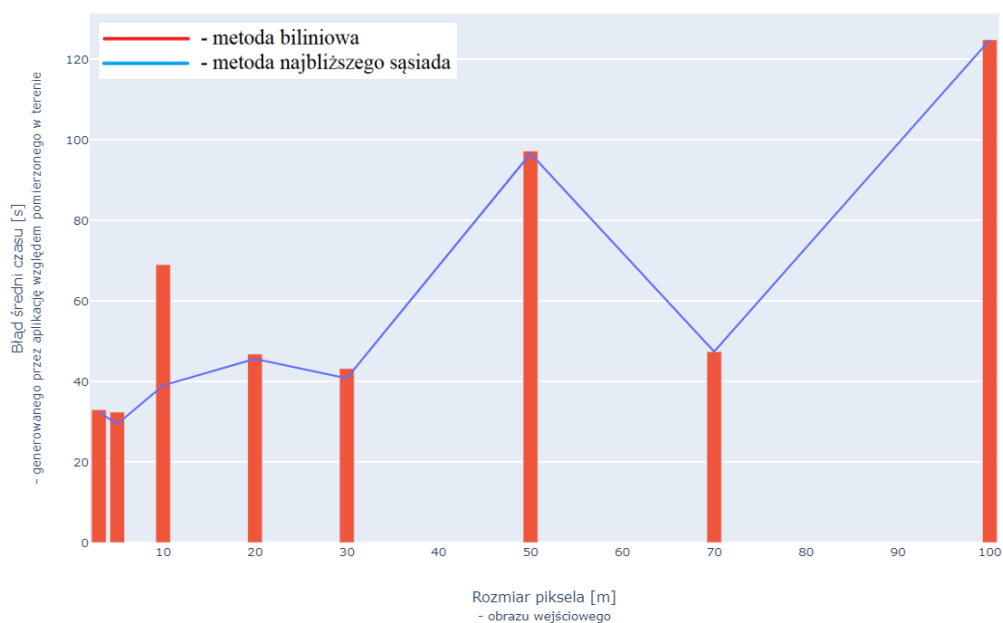
Rysunek 44. Wykres zależności błędów średnich czasu od długości odcinka trasy dla kolejnych odcinków uwzględniając wszystkie rodzaje rozdzielczości – ruch pieszy, metoda najbliższego sąsiada (niebieski), metoda biliniowa (czerwony) (źródło: opracowanie własne).



Rysunek 45. Wykres zależności błędów średnich czasu od długości odcinka trasy dla kolejnych odcinków uwzględniając wszystkie rodzaje rozdzielczości – ruch samochodowy, metoda najbliższego sąsiada (niebieski), metoda biliniowa (czerwony) (źródło: opracowanie własne).

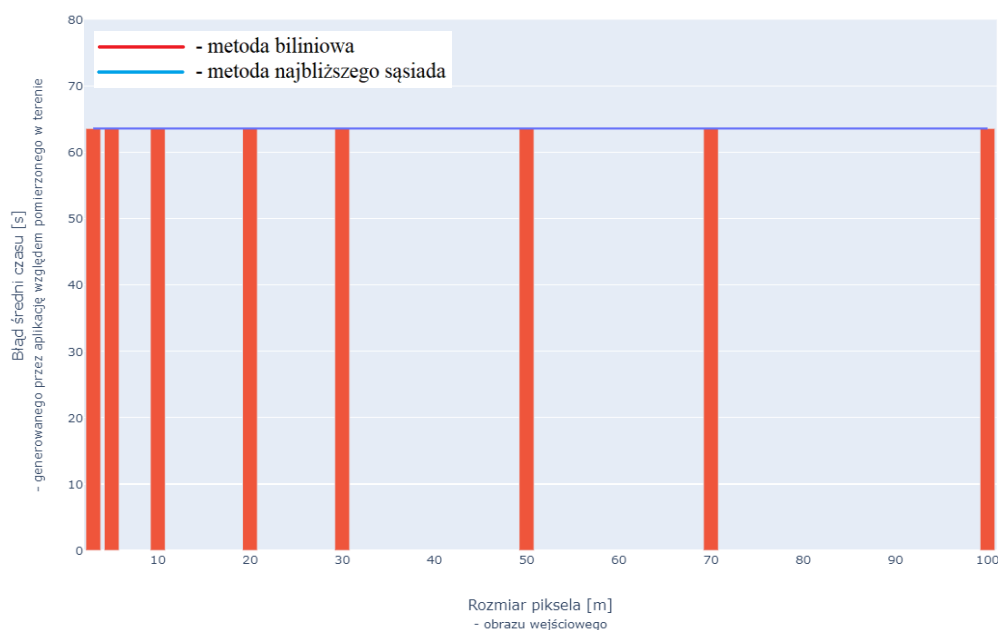
Korzystając z tabel (Tabela 45, Tabela 46) wykonano wizualizacje przedstawiające wpływ zmian rozdzielczości obrazu zawierającej informację o wysokości na błędy średnie przewidywanego czasu uzyskanego z aplikacji względem rzeczywistych wartości.

Wykres zależności błędów średnich czasu generowanego przez aplikację od zmian rozmiaru piksela obrazu wysokościowego.
Sposób lokomocji: ruch pieszy.



Rysunek 46. Wykres zależności błędów średnich czasu od rozmiaru piksela– ruch pieszy, metoda najbliższego sąsiada (niebieski), metoda biliniowa (czerwony)
(źródło: opracowanie własne).

Wykres zależności błędów średnich czasu generowanego przez aplikację od zmian rozmiaru piksela obrazu wysokościowego.
Sposób lokomocji: ruch samochodowy.



Rysunek 47. Wykres zależności błędów średnich czasu od rozmiaru piksela – ruch samochodowy, metoda najbliższego sąsiada (niebieski), metoda biliniowa (czerwony)
(źródło: opracowanie własne).

Na podstawie wykresu błędów średnich czasu względem rozmiaru piksela dla ruchu pieszego można zauważyć skokowe zmiany wartości, które są tym wyższe i mniej przewidywalne, im rozmiar piksela jest większy (Rysunek 46). Dla przejazdu samochodem wartość jest stała, co oznacza brak istotnego wpływu zmian rozdzielczości rastra na wyniki (Rysunek 47). Zastosowanie obu metod interpolacji generuje błędy na podobnym poziomie wartości. Jedynie w przypadku rozdzielczości 10 metrowej wystąpiła znacząca różnica.

Na podstawie Załącznika nr 1, można zaobserwować, że każda zmiana rozdzielczości obrazu wiąże się ze wzrostem lub obniżeniem wartości dla generowanych atrybutów. Nie jest to prosta zależność, którą można przedstawić za pomocą regresji liniowej, ale składa się na wiele czynników generujących pewne wahania.

Dla głębszego zrozumienia zachowania się wartości parametrów wykonano przykładowe zestawienie różnic wysokości oraz średniego spadku dla dwóch wybranych odcinków: 3 i 6 (Tabela 48, Tabela 49).

Tabela 48. Wartości wybranych atrybutów trasy numer 3 względem rozmiaru piksela, dla interpolacji: metodą najbliższego sąsiada.

Pieszem			Samochodem		
Trasa	Różnice wysokości [m]	Średni spadek [°]	Trasa	Różnice wysokości [m]	Średni spadek [°]
1x1	10.07	4.34	1x1	3.35	1.32
3x3	5.63	2.35	3x3	1.99	0.76
5x5	4.36	1.90	5x5	2.00	0.53
10x10	2.45	2.82	10x10	1.87	0.56
20x20	1.65	0.61	20x20	1.27	0.28
30x30	1.43	0.50	30x30	1.37	0.23
50x50	2.85	5.38	50x50	1.57	0.69
70x70	2.24	1.86	70x70	2.24	0.58
100x100	2.13	0.40	100x100	1.07	0.57

Tabela 49. Wartości wybranych atrybutów trasy numer 6 względem rozmiaru piksela dla interpolacji: metodą najbliższego sąsiada.

Pieszem			Samochodem		
Trasa	Różnice wysokości [m]	Średni spadek [°]	Trasa	Różnice wysokości [m]	Średni spadek [°]
1x1	57.22	4.42	1x1	24.17	2.06
3x3	41.36	3.31	3x3	18.95	2.36
5x5	40.95	4.54	5x5	16.88	1.43
10x10	55.19	3.94	10x10	31.52	2.44
20x20	58.42	3.64	20x20	53.54	5.67
30x30	56.17	5.45	30x30	47.37	4.14
50x50	75.92	6.78	50x50	74.60	6.49
70x70	62.26	5.80	70x70	67.78	3.71
100x100	93.31	6.68	100x100	93.31	7.50

Tabele wykazują, że wartości atrybutów ulegają zakresowym tendencjom spadkowym oraz wzrostowym a większe rozbieżności pojawiają się na trasie dłuższej. Zauważalne jest również znaczne spłaszczenie tras ruchu samochodowego szczególnie w zakresach rozdzielczości 1-10 m. Biorąc pod uwagę stabilność i zachowywanie się wyników, można przypuszczać, że najwyższa wiarygodność jest zachowana dla rozmiaru piksela do 5 m.

7. Wnioski

Podstawowym założeniem pracy było utworzenie wtyczki do programu QGIS, która umożliwi w szybki i prosty sposób wyznaczenie optymalnych tras dla zadanego zestawu warstw: wysokościowej, połączeń liniowych oraz punktów trasy. Ważnym aspektem była także analiza wiarygodności generowanych wyników, jak i zbadanie zmienności zachowywania się wartości obliczanych atrybutów. Ostatecznie osiągnięty wynik można uznać za zadowalający. Aplikacja może stanowić solidną podstawę dla znacznie bardziej rozwiniętych systemów nawigacyjnych, transportowych bądź logistycznych. Wnioski i spostrzeżenia wyciągnięte podczas całego procesu twórczego oraz badawczego pracy przedstawiono w poniższych podsumowaniach.

Nowo utworzona aplikacja jest w stanie poprawnie wyznaczyć trasę zoptymalizowaną pod względem zadanego parametru wraz z obliczeniem i załączeniem wskazanych przez użytkownika atrybutów. Oprócz opcji minimalizacji dostępna jest również maksymalizacja wyników. W ten sposób możliwe jest wygenerowanie trasy o najmniejszej lub największej wartości: długości, czasu, różnic wysokości, podejść, zejść, spadków. Interfejs graficzny wtyczki zapewnia prostą i sprawną obsługę bez konieczności zapoznawania się z dokumentacją narzędzia. Dużą zaletą jest wyposażenie aplikacji w zestaw zabezpieczeń i komunikatów, które ułatwiają wybór odpowiednich danych bądź informują użytkownika o wystąpieniu ewentualnych błędów. Szybkość obliczeń wtyczki jest w dużym stopniu uzależniona od wielkości i rozdzielczości rastra. Na przykładzie testowanych danych czas dla rozmiaru piksela 1 m wyniósł około 30 s, natomiast dla większych niż 10 m zmniejszył się do wartości nie przekraczającej 1 s.

Porównanie wyników generowanych przez aplikację na podstawowym rastrze 1x1 m z wartościami pomierzonymi w terenie wykazały błędy średnie czasu: 32 s (ruch pieszy), 1 min 3 s (ruch samochodowy), co oznacza wysoką dokładność szacowania i poprawne działanie algorytmów. Najmniejsza różnica czasu pomiędzy wartościami otrzymanymi z wtyczki a rzeczywistymi z terenu wyniosła 1 s, natomiast największa 2 min 45 s. Głównymi przyczynami rozbieżności były: sygnalizacja świetlna, miejscowe zakorkowania, przejścia dla pieszych oraz konieczność zmiany pasa ruchu.

Średnia prędkość ruchu samochodem jest silnie związana z charakterystyką komunikacyjną danego miasta, odległością od centrum oraz porą dnia. Dla Krakowa ustalona wartość 40 km/h jest stanowczo zbyt wysoka w stosunku do średniej otrzymanej z pomiaru 26.5 km/h. Jedynie odcinki 3 i 6 osiągnęły wartości prędkości bliskie założonej (40.3 i 38.9 km/h). W przypadku ruchu pieszego, to ukształtowanie powierzchni wpływa na prędkość, co jednocześnie wiąże się z generowaną wartością czasu. Jednak odnosząc się do przejazdu samochodowego wpływ ten jest już zaniedbywalny lub bardzo mały. Na podstawie tych obserwacji można stwierdzić, że numeryczny model terenu dla lokomocji pieszej wymaga danych o wysokiej dokładności, natomiast dla samochodowej nie jest to już konieczne.

Zmiany rozdzielczości pikseli z zastosowaniem obu metod interpolacji (najbliższego sąsiada, biliniowej) wpłynęły silnie i nieregularnie na wartości pozyskiwanych atrybutów. Wyniki najbliższe sytuacji rzeczywistej otrzymano dla rozmiaru piksela wynoszącego 5 m. Błąd średni w tym przypadku wyniósł 29 s. Zastosowanie różnych metod interpolacji wysokości pokazuje podobny sposób ich zachowania się dla zmiennych wartości rozdzielczości. W obu przypadkach uzyskane odchylenia są niewielkie a błędy średnie różnią się o pojedyncze sekundy. Wyjątkiem jest tylko sytuacja dla rozmiaru komórki 10x10 m, gdzie różnice są znacznie większe.

Analizując długość trasy można zauważyć, że cecha ta częściowo wpływa na wartość otrzymywanych błędów. Zauważalna jest tendencja do występowania większych rozbieżności

(między czasem generowanym przez wtyczkę a czasem uzyskanym z pomiaru terenowego) dla odcinków powyżej 1 km niż dla pozostałych, które są krótsze. Nie jest to jednak tendencja stała i niezmienna.

Wtyczka może zostać wykorzystana jako fundament służący budowie specjalistycznej i rozwiniętej aplikacji pomagającej w optymalizacji ruchu w mieście, wyznaczaniu tras dla turystów, a także firm kurierskich, dostawczych lub transportowych oraz sprawdzaniu komunikacyjności miejsc planowanych pod inwestycję mieszkaniową. Aplikacja może stanowić także bazę dla narzędzi badających cechy terenu, co jest przydatne przy projektowaniu przebiegu dróg, tras wyścigów kolarskich i biegowych. Posiada ona również funkcje mogące wesprzeć prace poszukiwawcze służb mundurowych oraz przewidywanie i zapobieganie powodziom poprzez liniowe badanie charakteru spływu wody.

8. Literatura

Książki i publikacje:

1. Beier Frederick J., Rutkowski Krzysztof; Logistyka. Szkoła Główna Handlowa, s.16. Warszawa, 1995.
2. Brzuchowska Jadwiga; Propozycje analiz zjawisk transportowych oparte na mapach rastrowych i narzędziach GIS. Czasopismo Techniczne, nr 1-A/3, s. 128-131, 135-136. Kraków, 2010.
3. Gągolewski Marek, Bartoszek Maciej, Cena Anna; Przetwarzanie i analiza danych w języku Python. Wydawnictwo Naukowe PWN, s. 3 . Warszawa, 2016.
4. Głazewski Andrzej; Modele rzeczywistości geograficznej a modele danych przestrzennych. Polski Przegląd Kartograficzny, nr 3, s.219. Warszawa, 2006.
5. Kapler Monika, Wojtal Paweł; Systemy informacji przestrzennej w strategii rozwoju logistyki miejskiej. Zeszyty Naukowe Politechniki Śląskiej, nr 130, s.294. Gliwice, 2018.
6. Koźlak Aleksandra; Inteligentne Systemy Transportowe jako instrumenty poprawy efektywności transportu. „LOGISTYKA”, nr 2. Poznań, 2008.
7. Kurczyński Zdzisław, Bakula Krzysztof; Generowanie referencyjnego numerycznego modelu terenu o zasięgu krajowym w oparciu o lotnicze skanowanie laserowe w projekcie ISOK.Archiwum Fotogrametrii, Kartografii i teledetekcji, wydanie specjalne: Monografia „Geodezyjne Technologie Pomiarowe”, s. 62.
8. Langmuir Eric; Mountaincraft and Leadership. The Scottish Sports Council/MTLB. 1984.
9. Lutz Mark; Learning Python. O'Reilly Media, s. 9-10, 3-4. Sebastopol, 2013.
10. Ładysz Jerzy; Technologia GIS w inżynierii bezpieczeństwa. Wyższa Szkoła Oficerska Wojsk Lądowych imienia generała Tadeusza Kościuszki, s.9. Wrocław, 2015.
11. Malinowski Mateusz, Woźniak Andrzej; Problem optymalizacji logistycznych parametrów transportu odpadów komunalnych w aspekcie strategii ekofirmy. Infrastruktura i Ekologia Terenów Wiejskich, nr 10, s. 107-119. Kraków, 2011.
12. Materka Andrzej, Strumiłło Paweł; Wstęp do komputerowej analizy danych. Politechnika Łódzka, Instytut Elektroniki, s. 66. Łódź, 2009.
13. McKinnley Wes; Python for Data Analysis. O'Reilly Media, s. 4. Sebastopol, 2013.
14. Medyńska-Gulij Beata; Kartografia. Zasady i zastosowania geowizualizacji. Wydawnictwo Naukowe PWN, s. 28-29, 88-92. Warszawa, 2015.
15. Nicholson T. A. J.; Finding the shortest route between two points in a network. The Computer Journal, nr 9, s. 275-280. 1996.
16. Ogrodniczak Michał, Ryba Błażej, Piórecka Natalia, Piłat Łukasz, Dawidzionek Joanna; Implementacja oprogramowań typu GIS w logistyce. Autobusy: technika, eksploatacja, systemy transportowe. Instytut naukowo-Wydawniczy „SPATIUM”, nr 6, s. 1493-1494. Radom, 2017.
17. Oskarbski Jacek, Jamroz Kazimierz; Zarządzanie bezpieczeństwem ruchu drogowego w systemie Tristar. Fundacja rozwoju Inżynierii Lądowej, s. 251-260. Gdańsk, 2007.
18. Tadeusiewicz Ryszard, Korohoda Przemysław; Komputerowa analiza i przetwarzanie obrazów. Wydawnictwo Fundacji Postępu Telekomunikacji, s. 3, 5, 13 . Kraków, 1997.
19. Tobler Waldo; “Non-isotropic geographic modelling”. Technical Report, nr 93-1. Santa Barbara, 1993.
20. Twardowski Mariusz; Wykład: Język Python na potrzeby GIS. Kraków, 2019.
21. Zinoviev Dmitry; Complex network analysis in Python. The Pragmatic Programmers, s. 1-5. Raleigh, 2018.

Strony internetowe:

22. Belani Gaurav; Programming Languages You Should Learn in 2020. Dostęp: <https://www.computer.org/publications/tech-news/trends/programming-languages-you-should-learn-in-2020> [17.06.2020]
23. Co to jest QGIS? Dostęp: <https://gis-support.pl/co-to-jest-qgis/> [04.07.2020]
24. Global Rankings 2018. Dostęp: <https://lpi.worldbank.org/international/global/2018> [22.07.2020]
25. Goel Aman; Best Programming Language to Learn in 2020 (for Job & Future). Dostęp: <https://hackr.io/blog/best-programming-languages-to-learn-2020-jobs-future> [17.06.2020].
26. Introduction. Dostęp: https://docs.qgis.org/3.10/en/docs/pyqgis_developer_cookbook/intro.html [25.06.2020]
27. Introduction. Dostęp: <https://www.riverbankcomputing.com/static/Docs/PyQt5/introduction.html#pyqt5-components> [25.06.2020]
28. Math – Mathematical functions. Dostęp: <https://docs.python.org/3/library/math.html> [25.06.2020]
29. Naismith's Rule. Dostęp: https://en.wikipedia.org/wiki/Naismith%27s_rule [19.06.2020]
30. Package osgeo. Dostęp: <https://gdal.org/python/> [25.06.2020]
31. PyQt5 Reference Guide. Dostęp: <https://www.riverbankcomputing.com/static/Docs/PyQt5> [25.06.2020]
32. QtWidgets. Dostęp: <https://www.riverbankcomputing.com/static/Docs/PyQt5/api/qtwidgets/qtwidgets-module.html> [25.06.2020]
33. Ranking miast przyjaznych kierowcom. Dostęp: <https://www.oponeo.pl/artykul/ranking-miast-przyjaznych-kierowcom> [13.06.2020]
34. Rijo Daniel; Google Maps now used by over 1 billion people every month. PPC Land, 2020. Dostęp: <https://ppc.land/google-maps-now-used-by-over-1-billion-people-every-month/> [08.06.2020]
35. The LPI methodology. Dostęp: <https://wb-lpi-media.s3.amazonaws.com/LPI%20Methodology.pdf> [22.07.2020]

9. Spis tabel

Tabela 1. Porównanie wartości oraz składowych wskaźnika LPI dla Polski oraz Niemiec [24].	5
Tabela 2. Efekty zastosowania inteligentnych systemów transportowych [6][17].	7
Tabela 3. Średnie prędkości w centrum miast [33].	9
Tabela 4. Średnie prędkości poza centrum miast [33].	9
Tabela 5. Opis odcinków mierzonej trasy.	33
Tabela 6. Atrybuty dla odcinków trasy pomierzonej w terenie, rodzaj lokomocji: pieszo (po lewej), samochodem (po prawej).	34
Tabela 7. Zestawienie średnich prędkości dla odcinków trasy pomierzonych w terenie.	34
Tabela 8. Atrybuty dla odcinków trasy kontrolowanej wygenerowane na podstawowym rastrze o rozdzielczości 1x1m, rodzaj lokomocji: pieszo (po lewej), samochodem (po prawej).	38
Tabela 9. Atrybuty wygenerowane na rastrze o rozdzielczości 3x3 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą najbliższego sąsiada.	40
Tabela 10. Atrybuty wygenerowane na rastrze o rozdzielczości 5x5 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą najbliższego sąsiada.	40
Tabela 11. Atrybuty wygenerowane na rastrze o rozdzielczości 10x10 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą najbliższego sąsiada.	40
Tabela 12. Atrybuty wygenerowane na rastrze o rozdzielczości 20x20 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą najbliższego sąsiada.	40
Tabela 13. Atrybuty wygenerowane na rastrze o rozdzielczości 30x30 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą najbliższego sąsiada.	41
Tabela 14. Atrybuty wygenerowane na rastrze o rozdzielczości 50x50 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą najbliższego sąsiada.	41
Tabela 15. Atrybuty wygenerowane na rastrze o rozdzielczości 70x70 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą najbliższego sąsiada.	41
Tabela 16. Atrybuty wygenerowane na rastrze o rozdzielczości 100x100 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą najbliższego sąsiada.	41
Tabela 17. Atrybuty wygenerowane na rastrze o rozdzielczości 3x3 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą biliniową.	41
Tabela 18. Atrybuty wygenerowane na rastrze o rozdzielczości 5x5 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą biliniową.	41
Tabela 19. Atrybuty wygenerowane na rastrze o rozdzielczości 10x10 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą biliniową.	42
Tabela 20. Atrybuty wygenerowane na rastrze o rozdzielczości 20x20 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą biliniową.	42
Tabela 21. Atrybuty wygenerowane na rastrze o rozdzielczości 30x30 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą biliniową.	42
Tabela 22. Atrybuty wygenerowane na rastrze o rozdzielczości 50x50 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą biliniową.	42
Tabela 23. Atrybuty wygenerowane na rastrze o rozdzielczości 70x70 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą biliniową.	42
Tabela 24. Atrybuty wygenerowane na rastrze o rozdzielczości 100x100 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą biliniową.	42

Tabela 25. Różnice atrybutów dla odcinków trasy kontrolnej pomierzonych w terenie oraz wygenerowanych na rastrze o rozdzielczości 1x1 m, rodzaj lokomocji: pieszo (po lewej), samochodem (po prawej).	45
Tabela 26. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości 3x3 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą najbliższego sąsiada.	46
Tabela 27. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości 5x5 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą najbliższego sąsiada.	46
Tabela 28. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości 10x10 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą najbliższego sąsiada.	46
Tabela 29. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości 20x20 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą najbliższego sąsiada.	46
Tabela 30. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości 30x30 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą najbliższego sąsiada.	47
Tabela 31. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości 50x50 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą najbliższego sąsiada.	47
Tabela 32. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości 70x70 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą najbliższego sąsiada.	47
Tabela 33. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości 100x100 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą najbliższego sąsiada.	47
Tabela 34. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości 3x3 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą biliniową.	47
Tabela 35. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości 5x5 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą biliniową.	47
Tabela 36. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości 10x10 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą biliniową.	48
Tabela 37. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości 20x20 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą biliniową.	48
Tabela 38. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości 30x30 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą biliniową.	48
Tabela 39. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości 50x50 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą biliniową.	48
Tabela 40. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości 70x70 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą biliniową.	48

Tabela 41. Różnice atrybutów pomierzonych w terenie względem wygenerowanych na rastrze o rozdzielczości 100x100 m, ruch: pieszo (po lewej), samochodem (po prawej), interpolacja: metodą biliniową.	48
Tabela 42. Błędy średnie czasu i długości dla kolejnych odcinków trasy z zastosowaniem wszystkich rozdzielczości, rodzaj lokomocji: pieszo (po lewej), samochodem (po prawej), metoda interpolacji: najbliższego sąsiada.	51
Tabela 43. Błędy średnie czasu i długości dla kolejnych odcinków trasy z zastosowaniem wszystkich rozdzielczości, rodzaj lokomocji: pieszo (po lewej), samochodem (po prawej), metoda interpolacji: biliniowa.	51
Tabela 44. Zsumowane błędy średnie czasu dla wszystkich odcinków trasy względem zastosowanej metody interpolacji.	52
Tabela 45. Błędy średnie czasu i długości dla wszystkich odcinków trasy w zależności od rozdzielczości rastra, rodzaj lokomocji: pieszo (po lewej), samochodem (po prawej), metoda interpolacji: najbliższego sąsiada.	52
Tabela 46. Błędy średnie czasu i długości dla wszystkich odcinków trasy w zależności od rozdzielczości rastra, rodzaj lokomocji: pieszo (po lewej), samochodem (po prawej), metoda interpolacji: biliniowa.	52
Tabela 47. Zsumowane błędy średnie czasu dla wszystkich rodzajów rozdzielczości względem zastosowanej metody interpolacji.	52
Tabela 48. Wartości wybranych atrybutów trasy numer 3 względem rozmiaru piksela, dla interpolacji: metodą najbliższego sąsiada.	57
Tabela 49. Wartości wybranych atrybutów trasy numer 6 względem rozmiaru piksela dla interpolacji: metodą najbliższego sąsiada.	57

10. Spis rysunków

Rysunek 1. Schematu rozkładu punktów i odległości w analizowanej sieci.	15
Rysunek 2. Wyznaczenie połączeń dla pierwszego stopnia powiązania wierzchołków.	15
Rysunek 3. Zbadanie drugiego stopnia połączeń.	15
Rysunek 4. Wyodrębnienie powiązań umożliwiających złączenie się gałęzi z pierwszego stopnia połączeń.	16
Rysunek 5. Wyodrębnienie ostatecznej wersji możliwych tras.	16
Rysunek 6. Wykres zmian wartości prędkości dla reguł Naismith'a, Langmuir'a, Tobler'a.	17
Rysunek 7. Przykładowy zestaw danych testowych.	20
Rysunek 8. Przykładowy zestaw danych kontrolnych.	20
Rysunek 9. Przykładowe rastry o rozdzielczościach: 10x10, 30x30, 50x50, 100x100 m.	20
Rysunek 10. Wizualizacja interpolacji wysokości z rastra na warstwę liniową. Punkty 1 oraz 5 – oryginalne, od 2 do 4 – powstałe w wyniku przecięcia linii na krawędziach komórek.	22
Rysunek 11. Po lewej: wyznaczenie miejsc przebiegu trasy na warstwie liniowej.	23
Rysunek 12. Wizualizacja przykładowo wygenerowanej najszybszej trasy między punktami 1-6 wraz z przedstawieniem wartości zawartych w tabeli atrybutów.	24
Rysunek 13. Budowa i zawartość zakładki wtyczki Find Route.	26
Rysunek 14. Wykorzystanie i zapis komunikatów QMessageBox w przykładowych funkcjach: odczytującej plik z rastrem, sprawdzającej rodzaj obiektów, odczytującej warunek optymalizacji.	27
Rysunek 15. Przykładowy wygląd niepoprawnej warstwy z komunikatem błędu.	28
Rysunek 16. Sygnalizacja błędów formatu wprowadzanych danych.	28
Rysunek 17. Komunikat braku wskazania optymalizacji oraz błędu wartości prędkości.	28
Rysunek 18. Lokalizacja miejsca uruchamiania wtyczki Find Route.	29
Rysunek 19. Prezentacja obsługi zakładki warstw.	30
Rysunek 20. Prezentacja obsługi zakładki warunków.	31
Rysunek 21. Prezentacja obsługi zakładki atrybutów.	31
Rysunek 22. Prezentacja zakładki pomocy.	32
Rysunek 23. Prezentacja warstwy wynikowej.	32
Rysunek 24. Położenie obszaru analiz na tle warstwy Open Street Map.	33
Rysunek 25. Wizualizacja rozbieżności wywołanych zadrzewieniem – trasa 1.	35
Rysunek 26. Wizualizacja rozbieżności wywołanych gęstą i wysoką zabudową – trasa 8.	35
Rysunek 27. Wpływ minimalizacji i maksymalizacji warunku optymalizacji na wygląd trasy.	36
Rysunek 28. Wpływ minimalizacji i maksymalizacji warunku optymalizacji na atrybuty trasy.	36
Rysunek 29. Wpływ optymalizacji względem parametru czasu oraz spadków.	37
Rysunek 30. Atrybuty trasy po optymalizacji względem parametru czasu oraz spadków.	37
Rysunek 31. Wizualizacja wzajemnego przebiegu drogi dla przejścia pieszego (kolor zielony) oraz przejazdu samochodem (kolor czerwony).	38
Rysunek 32. Wizualizacja wyglądu rozdzielczości komórek dla rastrow: 1x1, 3x3, 5x5, 10x10, 20x20, 30x30, 50x50, 70x70, 100x100 [m].	39
Rysunek 33. Wykres generowanego czasu względem różnych wartości rozdzielczości, sposób lokomocji: pieszo, metoda interpolacji: najbliższego sąsiada.	43
Rysunek 34. Wykres generowanego czasu względem różnych wartości rozdzielczości, sposób lokomocji: pieszo, metoda interpolacji: biliniowa.	43
Rysunek 35. Wykres generowanego czasu względem różnych wartości rozdzielczości, sposób lokomocji: samochód, metoda interpolacji: najbliższego sąsiada.	44

Rysunek 36. Wykres generowanego czasu względem różnych wartości rozdzielczości, sposób lokomocji: samochód, metoda interpolacji: biliniowa.	44
Rysunek 37. Wykres różnic generowanego czasu dla różnych wartości rozdzielczości względem pomiaru terenowego, sposób lokomocji: pieszo, metoda interpolacji: najbliższego sąsiada.	49
Rysunek 38. Wykres różnic generowanego czasu dla różnych wartości rozdzielczości względem pomiaru terenowego, sposób lokomocji: pieszo, metoda interpolacji: biliniowa.	49
Rysunek 39. Wykres różnic generowanego czasu dla różnych wartości rozdzielczości względem pomiaru terenowego, sposób lokomocji: samochód, metoda interpolacji: najbliższego sąsiada.	50
Rysunek 40. Wykres różnic generowanego czasu dla różnych wartości rozdzielczości względem pomiaru terenowego, sposób lokomocji: samochód, metoda interpolacji: biliniowa.	50
Rysunek 41. Wykres zależności różnic czasu od długości odcinka trasy – ruch pieszy.	53
Rysunek 42. Wykres zależności różnic czasu od długości odcinka trasy – ruch samochodowy.	53
Rysunek 43. Wykres zależności różnic czasu od długości odcinka trasy.	54
Rysunek 44. Wykres zależności błędów średnich czasu od długości odcinka trasy dla kolejnych odcinków uwzględniając wszystkie rodzaje rozdzielczości – ruch pieszy.	55
Rysunek 45. Wykres zależności błędów średnich czasu od długości odcinka trasy dla kolejnych odcinków uwzględniając wszystkie rodzaje rozdzielczości – ruch samochodowy,	55
Rysunek 46. Wykres zależności błędów średnich czasu od rozmiaru piksela– ruch pieszy,	56
Rysunek 47. Wykres zależności błędów średnich czasu od rozmiaru piksela.	56

11. Spis wzorów

(1) Wzór dla metody Naismith'a.	17
(2) Wzór dla metody Naismith'a z uwzględnieniem poprawki Langmuir'a.	17
(3) Wzór dla metody Tobler'a.	17
(4) Wzór na czas przejścia pieszo.	23
(5) Wzór na czas przejazdu rowerem.	23
(6) Wzór na czas przejazdu samochodem.	23
(7) Wzór na spadek.	24
(8) Wzór na odległość między punktami (2D).	24
(9) Wzór na odległość między punktami (3D).	24
(10) Wzór na błąd średni.	46

12. Spis załączników

Załącznik nr 1. Tabele atrybutów dla odcinków tras w zależności od rozdzielczości rastra oraz zastosowanej metody interpolacji podczas resamplingu.