

Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH



Instytut Informatyki

Praca dyplomowa magisterska

na kierunku Informatyka
w specjalności Informatyka w Multimediacach

Zastosowanie algorytmów cyfrowego przetwarzania obrazów
do identyfikacji szczytów górskich w czasie rzeczywistym

Adamski Maciej

Numer albumu 300184

promotor
prof. dr hab. inż. Przemysław Rokita

WARSZAWA 2023

Zastosowanie algorytmów cyfrowego przetwarzania obrazów do identyfikacji szczytów górskich w czasie rzeczywistym

Streszczenie.

Celem niniejszej pracy dyplomowej magisterskiej było zbadanie oraz przetestowanie algorytmów cyfrowego przetwarzania obrazów do identyfikacji szczytów górskich na obrazie w czasie rzeczywistym. Prace badawcze były realizowane na stworzonej na potrzeby projektu platformie testowej na urządzeniach mobilnych z systemem Android. Opracowany proces rozpoznawania gór może zostać wykorzystany również w innych konfiguracjach sprzętowych.

Opierając się na analizie literatury oraz podobnych, istniejących już rozwiązańach przygotowany został wstępny proces rozpoznawania szczytów, który w dalszych iteracjach prac badawczych był modyfikowany dzięki poszerzanej wiedzy, a także na podstawie wyników testów oraz porównań algorytmów. Została przygotowana na potrzeby pracy dyplomowej aplikacja, która w momencie wykonywania zdjęcia zapisuje dodatkowe dane geolokalizacyjne na jego temat. Dzięki temu stworzono zbiór statycznych zdjęć gór z odpowiednimi danymi. Zasilana nimi platforma testowa pozwalała sprawdzić działanie przygotowanych metod, a także wybrać optymalne w kontekście procesu rozpoznawania szczytów górskich. Testy były zorientowane na czas wykonania algorytmów związany z działaniem projektu w czasie rzeczywistym, ale również na ich jakość i skuteczność.

Proces rozpoznawania szczytów górskich zaproponowany w ramach pracy dyplomowej składa się z kilku elementów. Na początku zbierane są dane z sensorów urządzenia. Na ich podstawie generowany jest trójwymiarowy model płaszczyzny Ziemi z perspektywy obserwatora. Wykorzystywane do tego są *numeryczne modele terenu* oraz interfejs *OpenGL*. Na tak wygenerowanej panoramie określone są widoczne góry. W tym celu przeprowadzane jest filtrowanie danych dotyczących szczytów górskich przy pomocy metod usuwania powierzchni niewidocznych - *frustum culling* oraz *occlusion culling*. Wykorzystując algorytm detekcji krawędzi *Canny* tworzone są na podstawie panoramy oraz zdjęcia wejściowego obrazu binarne zawierające kontury gór. Przy użyciu metody *dopasowania obrazów na podstawie szablonu* porównywane są odpowiednie fragmenty zdjęć potencjalnie zawierające dane szczyty. Na podstawie uzyskanych wyników stwierdzana jest widoczność poszczególnych szczytów górskich na kolejnych klatkach nagrania.

Opracowany proces został poddany weryfikacji końcowej w warunkach rzeczywistych. Potwierdziła ona zdolność takiego cyklu do prawidłowej identyfikacji szczytów górskich oraz wykazała możliwość jej działania w czasie rzeczywistym.

Słowa kluczowe: rozpoznawanie szczytów górskich, rejestracja obrazu, wyrównanie zdjęć, dopasowanie obrazu na podstawie szablonu, numeryczne modele terenu, grafika trójwymiarowa, przetwarzanie cyfrowe obrazu, czas rzeczywisty

Applying digital image processing algorithms in real time mountain peaks labelling

Abstract.

The main aim of the Master's Degree Thesis was to research and test digital image processing algorithms which allow to recognize mountain peaks in the video in real-time. The experiments were conducted on the custom made platform that was made for mobile devices with the Android operating system. The developed mountain peaks recognition process can also be transferred to different platforms.

Based on the analysis of the current state of the art and other existing approaches, an early process of mountain peak recognition was prepared. In the next stages of the conducted research the process has been updated due to growing knowledge. Moreover, it was also improved with numerous tests and comparisons of different algorithms. Furthermore, for the sake of the thesis there was also created a simple application, which saves the additional geolocation data at the moment a picture is taken. Hence, a set of static photos of mountain peaks with appropriate data was created. The application was loaded with this set. It allowed to evaluate the quality of the different methods. What is more, it helps to choose optimal methods in terms of the mountain peak recognition process. Tests were mainly focused on algorithms time complexity in real-time. However, the quality and accuracy of the solutions were also taken into account.

The process of the mountain peaks recognition that was proposed in this thesis is made of several elements. At the very beginning, the data is gathered from the device's sensors. It is used to generate a 3D model of the Earth's surface from the observer's point of view. The model is created with the *digital elevation model* and *OpenGL* interface. With the generated panorama view, the visibility of mountain peaks are assessed. So as to do that, there is a need to filter the data concerning mountain peaks. The hidden-surface determination methods are used - *frustum culling* and *occlusion culling*. With the *Canny* edge detector algorithm, on the basis of the panorama view and the input photo, the binary images containing the mountain's contour are created. *Template matching* is used to compare specific parts of the images that might include possible peaks. Obtained results are used to conclude the visible mountain peaks in the consecutive frames of the video.

Developed process was verified in the real-life conditions. The process's ability to properly recognize mountain peaks was confirmed. Furthermore, it was shown to work in the real-time.

Keywords: mountain peaks recognition, image registration, image alignment, template matching, digital elevation model, three-dimensional graphics, digital image processing, real-time



.....
miejscowość i data

.....
imię i nazwisko studenta

.....
numer albumu

.....
kierunek studiów

OŚWIADCZENIE

Świadomy/-a odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie, pod opieką kierującego pracą dyplomową.

Jednocześnie oświadczam, że:

- niniejsza praca dyplomowa nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.) oraz dóbr osobistych chronionych prawem cywilnym,
- niniejsza praca dyplomowa nie zawiera danych i informacji, które uzyskałem/-am w sposób niedozwolony,
- niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanego z nadawaniem dyplomów lub tytułów zawodowych,
- wszystkie informacje umieszczone w niniejszej pracy, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami,
- znam regulacje prawne Politechniki Warszawskiej w sprawie zarządzania prawami autorskimi i prawami pokrewnymi, prawami własności przemysłowej oraz zasadami komercjalizacji.

Oświadczam, że treść pracy dyplomowej w wersji drukowanej, treść pracy dyplomowej zawartej na nośniku elektronicznym (płycie kompaktowej) oraz treść pracy dyplomowej w module APD systemu USOS są identyczne.

.....
czytelny podpis studenta

Spis treści

1. Wstęp	9
1.1. Motywacja pracy	9
1.2. Cel pracy	9
1.3. Etapy pracy	10
2. Analiza literatury i dostępnych danych	12
2.1. Przegląd literatury	12
2.2. Rozszerzona rzeczywistość	13
2.3. Komercyjne rozwiązania	14
2.4. Numeryczny model terenu (NMT)	14
2.5. Bazy danych szczytów górskich	16
2.6. Rezultat przeglądu literatury - teoretyczne kroki procesu	17
3. Stos technologiczny platformy testowej	18
3.1. OpenGL	18
3.2. OpenCV	19
3.3. System operacyjny Android	19
3.4. Sensory	20
3.5. Języki programowania	21
4. Zestaw statycznych zdjęć do celów testowych	22
5. Przegląd algorytmów i szczegóły implementacyjne	24
5.1. Generowanie trójwymiarowego modelu terenu	25
5.2. Widoczność szczytów górskich na trójwymiarowym modelu terenu	26
5.2.1. Filtracja szczytów poza polem widzenia kamery	27
5.2.2. Filtracja zasłanianych szczytów	28
5.2.3. Przykładowy rezultat widoczności szczytów górskich na wygenerowanym modelu	29
5.3. Wyznaczanie odległości między dwoma punktami geograficznymi	29
5.3.1. Przegląd wybranych algorytmów	29
5.4. Detekcja krawędzi gór	31
5.4.1. Oczekiwany rezultat	31
5.4.2. Algorytm Canny	32
5.5. Dopasowanie obrazów binarnych	33
5.6. Dopasowanie na podstawie wzorca	34
5.7. Dopasowanie na podstawie cech	36
5.7.1. Punkty kluczowe i ich deskryptory	37
5.7.2. Dopasowanie punktów kluczowych obu zdjęć	39
5.7.3. Projekcja położenia z wykorzystaniem homografii	40
6. Dodatkowe problemy implementacyjne	43

6.1. Niedokładność sensorów urządzeń	43
6.2. Skalowanie trójwymiarowego modelu terenu	43
6.3. Migotanie wskazań lokalizacji widocznych szczytów	44
7. Porównanie wybranych algorytmów i próby optymalizacji	45
7.1. Wpływ rozdzielczości danych SRTM na dokładność odwzorowania terenu	45
7.2. Odrzucenie niepotrzebnych danych SRTM	47
7.3. Porównanie algorytmów obliczających odległość między dwoma punktami geograficznymi	49
7.4. Testowanie i optymalizacja algorytmów dopasowania obrazów	52
7.4.1. Wpływ dylacji na skuteczność dopasowania obrazów	56
7.4.2. Porównanie algorytmów dopasowania obrazów	61
7.4.3. Podsumowanie badania algorytmów dopasowania obrazów	65
7.5. Wpływ kolejności przetwarzania pikseli zdjęcia i wykorzystania pamięci podręcznej procesora	66
8. Weryfikacja końcowa	69
8.1. Prezentacja działania rozwiązania	69
8.2. Złożoność czasowa rozwiązania	70
8.3. Podsumowanie weryfikacji końcowej	70
9. Podsumowanie	71
9.1. Wykonane prace	71
9.2. Zdobyta wiedza i wyciągnięte wnioski	73
9.3. Możliwość rozwoju i dalszych badań	74
Bibliografia	75
Spis rysunków	81
Spis tabel	82
Spis załączników	83

1. Wstęp

1.1. Motywacja pracy

Dzisiejsza technologia oraz możliwość ciągłej geolokalizacji naszego położenia mimo, że bardzo ułatwiają turystom wędrówki, to nie zawsze pozwalają jednoznacznie określić prawidłowy kierunek, w którym powinni się oni poruszać. Posiadanie informacji o nazwach pobliskich gór może być bardzo ważne dla osób, które zgubiły się na szlaku. Dzięki rozwiązaniom pozwalającym identyfikować i określić nazwy szczytów górskich, turysta ma możliwość odczytania widocznych w danym momencie gór. Dzięki takiej wiedzy mogliby oni lepiej nawigować lub przekazać te dane służbom ratowniczym w celu łatwiejszego namierzenia ich w przypadku ewentualnych poszukiwań. Z tego powodu może mieć to realny wpływ na bezpieczeństwo na szlakach górskich.

Pozytywny wpływ na aspekt dotyczący bezpieczeństwa mogłoby mieć również wykorzystanie takiego rozwiązania do celów analizy warunków atmosferycznych w okolicach danego szczytu lub ścieżkach turystycznych na niego prowadzących. Łącząc algorytmy rozpoznawania szczytów i pogody możliwe jest automatyczne wykrywanie zmian meteorologicznych, nadchodzących burz czy dużego zachmurzenia w okolicach danej góry. Tego typu monitoring może informować turystów o możliwych niebezpiecznych warunkach pogodowych. Możliwe też jest sprawdzenie przez użytkowników, na który szczyt nie powinno się planować wycieczek.

Aplikacja pozwalająca etykietować widoczne szczyty może mieć też działanie edukacyjne. Gdy chcemy poznać nazwę jakiegoś pasma czy gór w danym regionie możemy skorzystać z takich rozwiązań oraz uzyskać pożądane informacje. Dzięki temu pozwoli to zwiększyć ogólną wiedzę użytkowników na temat geografii i geologii. Może zachęcić ich również do eksploracji i poznawania mniej popularnych szlaków czy rejonów.

Rozwiązanie potrafiące identyfikować szczyty nie musi być wykorzystywana jedynie przez turystów spędzających czas w górach czy osoby ciekawe tego jaki wierzchołek widzą. Sieci społecznościowe również mogłyby korzystać z takich rozwiązań. Ich użytkownicy mogliby dodawać etykiety z nazwami szczytów, tak jak jest to aktualnie możliwe na przykład z miastami, znanymi budowlami czy miejscowościami kultury i pomnikami. Dałoby to również możliwość automatycznego oznaczania zdjęć odpowiednimi nazwami gór, a takie dane można wykorzystać przykładowo w procesie uczenia sztucznej inteligencji.

1.2. Cel pracy

Celem niniejszej pracy dyplomowej było poznanie oraz przebadanie i porównanie, a także próby optymalizacji algorytmów przetwarzania cyfrowego obrazów pozwalających identyfikować szczyty górskie na obrazie. Finalnie, miało to pozwolić na stworzenie i opisanie procesu rozpoznającego góry w czasie rzeczywistym.

1. Wstęp

Na podstawie aspektów opisanych w podrozdziale dotyczącym motywacji pracy sformułowano założenie, że poszczególne algorytmów miały być badane na platformie testowej działającej na urządzeniach mobilnych. Związane jest to z faktem, że urządzenia takie jak smartfony większa część społeczeństwa ma cały czas przy sobie. Dzięki temu możliwe byłoby użytkowanie potencjalnego projektu opartego na opracowanym procesie na przykład podczas wycieczki turystycznej w góry. Z tego powodu platforma testowa wykorzystywana do badania algorytmów w ramach pracy dyplomowej została zaimplementowana na system Android, który jest jednym z najpowszechniejszych rozwiązań na urządzenia mobilne.

Badania algorytmów, które było kluczowym elementem pracy magisterskiej, opierały się nie tylko na testowaniu ich funkcjonalności, ale również dotyczyło aspektów takich jak wydajność czy użyteczność. Miały one przede wszystkim umożliwiać działanie procesu identyfikacji szczytów górskich w czasie rzeczywistym. Ze względu na ograniczoną moc docelowych urządzeń, jakimi są urządzenia mobilne, by to osiągnąć zastosowane algorytmy nie mogły mieć zbyt dużej złożoności obliczeniowej, zachowując przy tym zadowalającą jakość. Na podstawie wyników takich badań miały zostać wybrane optymalne algorytmy zarówno pod względem czasu potrzebnego na obliczenia, jak i wartości stosowanych metryk.

Elementem końcowym pracy dyplomowej miało być zweryfikowanie procesu składającego się ze zdefiniowanych jako optymalne algorytmów pod kątem jakości rozpoznawania szczytów oraz zdolności do działania w czasie rzeczywistym na urządzeniach mobilnych.

1.3. Etapy pracy

Pierwszym etapem przygotowania pracy magisterskiej była analiza literatury traktującej o zagadnieniach powiązanych z rozpoznawaniem szczytów górskich na obrazie. Na ich podstawie przygotowany został teoretyczny proces rozpoznawania gór, wymagane biblioteki oraz źródła danych wykorzystywane w projekcie. W szczególności określono wstępную listę kroków potrzebnych do prawidłowej detekcji szczytów i opisano rozwiązania służące do generowania grafiki trójwymiarowej oraz przetwarzania cyfrowego obrazów. Jako źródła danych przybliżono numeryczne modele terenu oraz bazy zawierające wpisy na temat szczytów na świecie.

Wykorzystując uzyskaną wiedzę teoretyczną zaimplementowano platformę testową służącą do testowania i badania poszczególnych elementów procesu identyfikacji szczytów górskich na obrazie. Następnie przygotowano zestaw zdjęć gór oraz odpowiadające im dane dotyczące geolokalizacji. Wykonane zostały przez autora pracy przy pomocy aplikacji stworzonej na potrzeby pracy dyplomowej. Podczas zapisywania obrazu tworzył on również plik zawierający takie wpisy jak lokalizacja GPS czy kąt o jaki obrócone było urządzenie. Służyły one jako zbiór testowy, na którym przeprowadzano badania oraz porównanie algorytmów.

Przeprowadzone zostały testy mające na celu określenie jakości poszczególnych metod jako klasyfikatorów. Podjęte zostały również próby optymalizacji oraz poprawienie dokładności wskazań przez dodatkowe operacje, na przykład poprzez dylacje obrazów binarnych. Jakość rozwiązań była mierzona przy pomocy metryk takich jak precyzyja, czułość czy miara F1. Pod uwagę brany był też kontekst wydajnościowy ze względu na ograniczoną moc obliczeniową urządzeń mobilnych. Z tego powodu przeprowadzono szereg modyfikacji mających na celu zmniejszenie złożoności algorytmów całego procesu.

Badano między innymi wpływ ilości danych SRTM na odwzorowanie modelu i czas jego generowania. Podjęte były próby odrzucenia częściowych wpisów numerycznego modelu terenu w zależności od możliwej wizji z pozycji obserwatora. Porównano metody obliczające odległość między dwoma punktami mając na uwadze ich dokładność i złożoność obliczeniową. Przetestowano również szczegółowo algorytmy dopasowania obrazów poświęcając im najwięcej uwagi ze względu, że jest to kluczowy element opracowywanego procesu. Przeprowadzano także weryfikacje końcową całościowego procesu złożonego z wybranych algorytmów pod względem skuteczności detekcji szczytów oraz działania w czasie rzeczywistym. Celem potwierdzenia działania procesu wykonane zostały nagrania prezentacyjne w paśmie góra, dołączone w formie załącznika numer 1.

2. Analiza literatury i dostępnych danych

W rozdziale tym przeprowadzono przegląd literatury dotyczącej zagadnienia identyfikacji szczytów górskich. Przedstawiony został tutaj proces jako całość, natomiast bardziej szczegółowe źródła traktujące o poszczególnych elementach związanych z tym zagadnieniem opisane zostały w kolejnych rozdziałach poświęconym konkretnym aspektom. Przedstawiono komercyjne rozwiązania dostępne na rynku umożliwiające etykietowanie szczytów górskich. Opisano również rodzaj danych wykorzystywanych w projekcie. Są to m.in.: numeryczny model terenu oraz zbiór nazw geograficznych takich jak rzeki czy góry. Przybliżone zostało także pojęcie rozszerzonej rzeczywistości związane z nakładaniem pewnych elementów na obraz na żywo. Na koniec zobrazowano teoretyczny proces rozpoznawania szczytów stworzony na podstawie przeprowadzonej w tym rozdziale analizy.

2.1. Przegląd literatury

Problem rozpoznawania szczytów górskich na obrazie w ostatnich latach zyskiwał na popularności. Skutkiem tego jest coraz większa dostępność opracować naukowych poruszających ten problem. Baboud i in. [1] przedstawili rozwiązanie wykorzystujące pole widzenia urządzenia, a także trójwymiarowy model terenu. Ich metoda opiera się na porównaniu obrazu rzutowanego na widok sferyczny $\angle 360^\circ$. Opisali oni sposób dopasowania dwóch modeli przy pomocy korelacji wzajemnej (krzyżowej), przekształcając wcześniej domenę problemu w transformatę Fouriera. Wykorzystali oni metryczny sposób detekcji krawędzi gór o dużej złożoności algorytmicznej. Badania w ich pracy wykonywane były przy pomocy komputerów stacjonarnych z dedykowanymi kartami graficznymi, co może prowadzić do hipotezy o wysokim zapotrzebowaniu na moc obliczeniową w ich rozwiązaniu. Stoi to w sprzeczności z założeniami pracy dyplomowej o testowaniu i działaniu w czasie rzeczywistym procesu rozpoznawania szczytów na urządzeniach mobilnych. Z tego powodu ważne było znalezienie rozwiązania, które było mniej skomplikowane pod względem obliczeniowym.

Fedorov w swojej pracy magisterskiej [2] oraz w innych artykułach naukowych [3], [4], wraz ze swoimi promotorami rozwinięli podejście przedstawione w [1]. Zastosowali oni modyfikację algorytmu zwiększając skuteczność detekcji krawędzi pasm górskich, a także późniejszego filtrowania szumów. Dodatkowo, zmienili sposób dopasowania obrazu. Opiera się on teraz na modelu walca, a nie sfery, co pozwoliło na znaczne zmniejszenie złożoności obliczeniowej. Jednak dalej wykorzystuje korelację krzyżową, lecz w tym wypadku w postaci wektorowej. Rozszerzeniem osiągnięć w [2] jest rzutowanie na sferę panoramy dwuwymiarowej wykorzystującą estymowane pole widzenia urządzenia opisane w [3]. W kolejnym ich artykule [4], po raz kolejny zmienili oni podejście do wykrywania krawędzi wprowadzając konwolucyjne sieci neuronowe oraz modyfikując sposób badania korelacji między dwoma zdjęciami.

Georges Baatz i in. w swoim opracowaniu [5] opisali temat określania geolokalizacji zdjęcia na podstawie terenów górzystych. Z problemem rozpoznawania szczytów ma wiele wspólnego, ponieważ operują na zdjęciach przedstawiających pasma górskie oraz wykorzystuje generowanie numerycznego modelu terenu, czy rozpoznawanie konturów gór. Duża część artykułu poświęcona jest segmentacji nieba, ponieważ w zaproponowanym rozwiązaniu rozważane są tylko krawędzie najwyższych partii gór, a także częściowe ich kontury. Pomijane są tutaj pasma górskie widoczne na bliższym planie, które mają mniejszą wysokość. Takie podejście uwrażliwia program na dodatkowe obiekty występujące na obrazie, a będące w jego wyższej części niż góry.

W opracowaniach [1]–[3] wykorzystany został trójwymiarowy model terenu, który wizualizuje to co jest widoczne na zdjęciach. Wykorzystują oni tak zwane numeryczne modele terenu (rozdz. 2.4) zawierające wysokość nad poziomem morza w danym punkcie i na ich podstawie generują taką wizualizację.

W podejściu prezentowanym przez Porzi i in. [6] do celów detekcji krawędzi gór wykorzystywany jest klasyfikator *Fernes* (ang. Random Fernes) autorstwa Mirona Kursa [7] oraz rozpoznawanie cech.

2.2. Rozszerzona rzeczywistość

Rozszerzona rzeczywistość (ang. *AR - Augmented Reality*) [8] zyskuje coraz większą popularność w wielu dziedzinach nauki, rozrywki czy przemysłu. Jest to połączenie dwóch światów - rzeczywistego oraz wygenerowanego komputerowo jako grafiki 3D. Obie rzeczywistości w tym systemie przeplatają się oraz wchodzą ze sobą w interakcję.

W obszarze transportu istnieje wiele znanych rozwiązań wykorzystujących rozszerzoną rzeczywistość. Występują nawigacje oznaczające prawidłowy kierunek na obrazie z kamery. Kluczowe informacje na temat drogi, ograniczeń prędkości czy zdarzeń na niej występującej mogą być prezentowane na szybie samochodu lub motocykla - w przypadku tego drugiego może to mieć duże znaczenie dla bezpieczeństwa ze względu na specyfikę jazdy jednoślada. Pozwala to również bardziej skupić się na samej drodze, bez odrywania od niej wzroku.

AR jest również szeroko wykorzystywana w segmencie rozrywki, w szczególności w grach mobilnych. W ciągu ostatnich lat fenomenem była gra *Pokemon GO* - generująca rocznie średnio miliard dolarów przychodu. W szczytce miała 230 milionów aktywnych użytkowników [9]. Na obraz widziany z poziomu kamery nakłada on fikcyjne stworzenia, tak jakby występowały w świecie realnym oraz z którymi możemy wchodzić w interakcję.

Wizualizacja i projektowanie wnętrz lub przestrzeni, dzięki nakładaniu mebli czy obiektów na aktualny wygląd pomieszczeń staje się bardziej immersywne i łatwiejsze. Sieć sklepów meblowych *IKEA* oferuje swoim klientom możliwość wirtualnego umieszczenia swoich produktów np. w naszym własnym mieszkaniu, przez co możemy bez potrzeby zakupu sprawdzić jak dany produkt wkomponuje się w nasze wnętrze [10].

2.3. Komercyjne rozwiązania

Na rynku dostępnych jest kilka aplikacji komercyjnych etykietujących szczyty górskie na obrazie z kamery. Niektóre z nich bazują tylko na danych pochodzących z sensorów takich jak kompas i lokalizacji GPS (np. *PeakFinder App* [11]). Nie analizują one jednak zawartości obrazu, co skutkuje wrażliwością na dodatkowe obiekty w scenie, zmienne warunki atmosferyczne lub chmury zasłaniające szczyty. Aplikacja pomija je i traktuje jakby ich nie było, mimo że mogą przykrywać poszczególne wierzchołki. Są to proste rozwiązania, mające wiele niedoskonałości, objawiających się przede wszystkim niską dokładnością wskazań szczytów. Pomimo to, do zalet niewątpliwie trzeba zaliczyć prostotę implementacji oraz małą złożoność obliczeniową, dzięki czemu mogą być wykorzystywane nawet na sprzętce o słabych parametrach, w szczególności na urządzeniach mobilnych.

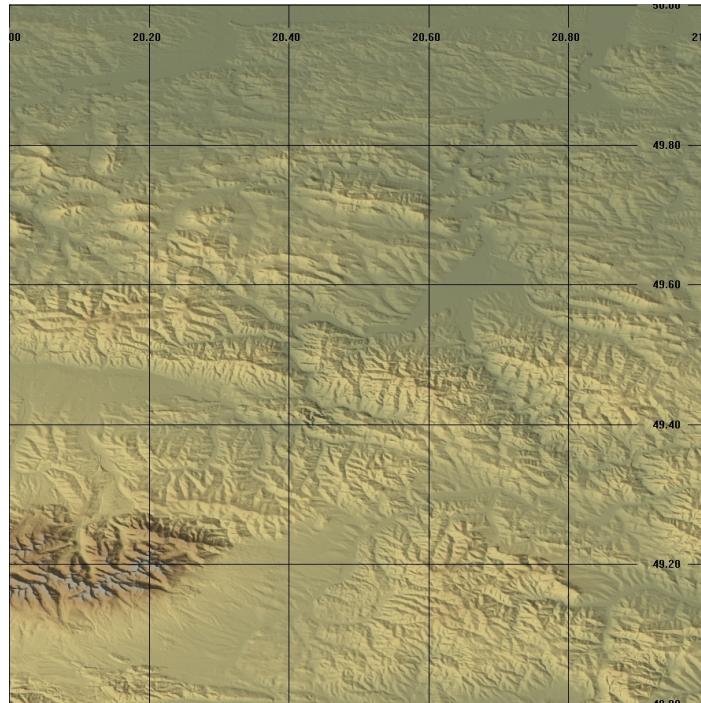
Występują również bardziej zaawansowane programy, które w swoim działaniu korzystają z większej liczby algorytmów przetwarzania obrazu czy nawet sztucznej inteligencji. Przykładem takiego oprogramowania jest *PeakLens* [12] (aplikacja autorów licznych artykułów cytowanych w tej pracy dyplomowej - Fedorov i in.). Takie podejście do problemu rozpoznawania szczytów wymaga mocniejszych układów obliczeniowych, ponieważ wykorzystane są nietywialne algorytmy o większej złożoności, a także pobieranie i interpretowanie dodatkowych danych pochodzących z zewnątrz systemu. Model terenu generowany jest z większą szczegółowością. Skutkuje to dużo większą skutecznością detekcji oraz eliminacją licznych problemów względem prostego podejścia, ponieważ brane są pod uwagę obiekty oraz przesłanianie szczytów.

Przykładami dostępnych przy pomocy przeglądarki rozwiązań umożliwiających generowanie widoków z oznaczonymi szczytami są: *Generate a panorama* [13] oraz *PeakFinder* [14]. Pierwsza aplikacja stworzona przez dr. Ulricha Deuschla na podstawie parametrów takich jak lokalizacja, kąt patrzenia czy wysokość obserwatora generuje płaską panoramę. Drugie rozwiązanie jest autorstwa tej samej firmy co opisane w poprzednim podrozdziale *PeakFinderApp*. Dzięki niej dostępny jest trójwymiarowy, interaktywny model terenu. Na obu z nich widoczne szczyty oznaczane są nazwami. Nie pozwalają jednak rozpoznawać szczytów górskich na obrazie i pełnią funkcję głównie informacyjną. Były jednak wykorzystywane w trakcie tworzenia pracy dyplomowej jako źródło porównań do tworzonych terenów w ramach projektu oraz dodatkowych danych na temat szczytów górskich.

2.4. Numeryczny model terenu (NMT)

W wielu regionach świata dostępne publicznie są numeryczne dane topograficzne o danym terenie [15]. Są to tak zwane Numeryczne Modele Terenu - NMT (ang. Digital Elevation Model - DEM lub Digital Terrain Model - DTM). DTM jest rozszerzeniem DEM o takie dane jak rzeki czy nieciągłości terenu [16]. Zawierają one informacje o wysokości nad poziomem morza w danym punkcie geograficznym, pomijając obiekty typu drzewa czy budynki.

W ramach pracy dyplomowej wykorzystywane są dane numeryczne zebrane, a następnie udostępnione przez Narodową Agencję Aeronautyki i Przestrzeni Kosmicznej w ramach międzynarodowej misji Shuttle Radar Topography Mission [17]. Dzięki niej stworzony został model terenu lądów pomiędzy równoleżnikami 56°S i 60°N. Dane SRTM występują w kilku wersjach w zależności od rozdzielczości. Na potrzeby pracy dyplomowej zintegrowane zostały wersje SRTM1 ($1''$) oraz SRTM3 ($3''$) o rozdzielczości terenowej odpowiednio 1 i 3 sekundy kątowej. Przekłada się to na dokładność średnią 30 i 90 metrów. Wersja $3''$ została uzyskana przez uśrednienie SRTM1. Opierają się na punktowej reprezentacji terenu w postaci siatki kwadratowej. Dla jednego elementu siatki terenu (o rozmiarach jednej jednostki szerokości i długości geograficznej) zawierają, w zależności od wersji, 3601x3601 lub 1201x1201 próbek. Przykładowy rzut pionowy terenu wygenerowany przy pomocy oprogramowania *3DEM Visualization Software* [18] dla danych SRTM o rozdzielczości $1''$ i współrzędnych geograficznych 49 – 50N, 20 – 21E pokazany został na rysunku 2.1.



Rysunek 2.1. Rzut pionowy danych SRTM 49-50N, 20-21E wygenerowany z wykorzystaniem oprogramowania 3DEM.

Wykorzystując dane typu NMT możliwe jest wygenerowanie trójwymiarowej mapy terenu, a następnie rzutowanie jej na dwuwymiarową panoramę. Uzyskany w ten sposób obraz może zawierać dodatkowo informacje o szczytach górskich i może być dopasowany do wejściowego zdjęcia. Nakładając na siebie oba obrazy możemy zweryfikować widoczność poszczególnych szczytów - a dzięki temu następnie odrzucić te niewidoczne z perspektywy użytkownika, a finalnie również etykietować te widoczne.

2. Analiza literatury i dostępnych danych

Posiadając jedynie dane dotyczące geo-lokalizacji możliwe jest wygenerowanie modelu trójwymiarowego (cylindrycznego lub sferycznego) [1][2][5] z danego punktu ziemi, a następnie przeszukiwanie całej przestrzeni wokół obserwatora. Natomiast jeśli prócz lokalizacji dostępne są również informacje o polu widzenia urządzenia lub estymowana jest jego wartość możliwe jest przeprowadzenie dopasowania na płaskim obrazie. Ze względu na mniejszą powierzchnię możliwych porównań podejście takie powinno dawać dokładniejsze wyniki. W przypadku opracowania Fedorov i in. [3] obie opisane techniki są ze sobą łączone - przybliżane jest pole widzenia, które następnie rzutowane jest na przestrzeń trójwymiarową $\angle 360^\circ$.

W pracy dyplomowej jednym z założeń było posiadanie danych pochodzących z sensorów urządzenia, w szczególności informacji dotyczących lokalizacji użytkownika, kątów obrotu urządzenia oraz specyfikacji technicznej zamontowanego w nim aparatu. Z tego powodu mogły być one wykorzystane przez badane algorytmy oraz poszczególne elementy opracowywanego projektu weryfikującego działanie procesu identyfikacji szczytów górskich na obrazie w czasie rzeczywistym. Takie dane pozwalają uzyskać dokładniejsze odwzorowanie terenu oraz generować panoramę terenu w taki sposób, jakby była widoczna z punktu, w którym znajduje się obserwator oraz z takim samym polem widzenia. Może mieć to realny wpływ na skuteczność dopasowania obrazów oraz zmniejszyć złożoność algorytmów ze względu na ograniczenie przestrzeni, którą trzeba porównać ze zdjęciem wejściowym.

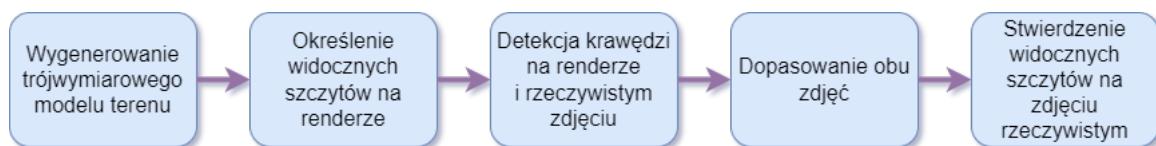
2.5. Bazy danych szczytów górskich

Projekt oraz platforma testowa tworzona w ramach pracy dyplomowej zasilana jest bazą danych nazw geograficznych *GeoNames* [19]. Jest to zbiór danych udostępnianych na licencji Creative Commons (CC) [20]. Kolekcja ta składa się z około 25 milionów wpisów, z których prawie 60 tysięcy dotyczy Polski. Baza danych podzielona została na różne kategorie takie jak wody, parki, miasta czy góry. Jednak, ze względu na charakter pracy dyplomowej interesujące są jedynie wpisy dotyczące danych należących do ostatniego z wymienionych typów. W Polsce zdefiniowano około 1200 nazw gór, które stanowią klu-czowe źródło informacji dla projektu podczas określania widocznych na wygenerowanym modelu szczytów.

Dane *GeoNames* podzielone są ze względu na kraj. System opracowany na potrzeby projektu pracy dyplomowej pobiera wpisy dla określonej puli krajów, następnie przetwarza je usuwając nieinteresujące typy oraz dane. W kolejnym kroku dzieli je ze względu na pola zależne od długości i szerokości geograficznej. Działania te pozwalają na zebranie dużej ilości danych na temat gór w wybranej przestrzeni geograficznej, co może przekładać się na precyzyjniejsze określanie widocznych szczytów. Podział na mniejsze zbiory pozwala zaoszczędzić pamięć operacyjną oraz moc obliczeniową potrzebną do ewentualnego procesu filtrowania poszczególnych wpisów.

2.6. Rezultat przeglądu literatury - teoretyczne kroki procesu

Na podstawie przeprowadzonego przeglądu literatury stworzony został teoretyczny opis procesu rozpoznawania szczytów górskich. Na początku zakłada on wygenerowanie trójwymiarowego modelu topograficznego wykorzystując dane numeryczne SRTM. Na takiej wizualizacji terenu określone mają być widoczne szczyty z bazy danych GeoNames. W trakcie kolejnego etapu użyte mają być algorytmy rozpoznawania konturów celem stworzenia obrazów binarnych z krawędziami gór na podstawie zwizualizowanej panoramy oraz zdjęcia rzeczywistego. Wygenerowane w ten sposób obrazy powinny być ze sobą porównane by określić podobieństwa poszczególnych ich wycinków w zależności od badanego szczytu. Na podstawie takich wskaźników stwierdzona może być widoczność gór na zdjęciu rzeczywistym. Wizualizację takiego hipotetycznego procesu w postaci diagramu przedstawiono na rysunku 2.2. W dalszej części pracy w sposób bardziej szczegółowy przeanalizowano i opisano poszczególne elementy tego cyklu.



Rysunek 2.2. Teoretyczne etapy procesu identyfikacji szczytów górskich na obrazie.

3. Stos technologiczny platformy testowej

Na potrzeby pracy magisterskiej ustalony został stos technologiczny związany z pracami badawczymi, a także z opracowywaną platformą testową, przy pomocy której testowano implementowane algorytmy cyfrowego przetwarzania obrazów odnoszące się do zagadnienia identyfikacji szczytów górskich na obrazie. Opisano między innymi wybrane biblioteki służące do generowania grafiki trójwymiarowej oraz modyfikowania i analizowania obrazów. Ponadto, w ramach tego rozdziału zostały określone i opisane sensory urządzeń potrzebne do stworzenia takiego projektu. Wybrano również języki programowania użyte do stworzenia bazy testowej oraz system operacyjny, na który była ona implementowana.

3.1. OpenGL

OpenGL (ang. Open Graphics Library) [21] to interfejs służący do tworzenia i renderowania grafiki - zarówno dwu, jak i trójwymiarowej. Aktualnie standaryzowany przez konsorcjum Khronos Group, zrzeszające największe firmy związane z grafiką komputerową. Biblioteka ta bazuje na niskopoziomowym interfejsie. Dzięki temu możliwe jest osiągnięcie lepszych zależności wydajnościowych.

W swoim działaniu OpenGL wykorzystuje tzw. potok graficzny. Składa się on z kolejnych etapów przetwarzania danych wejściowych - wierzchołków i trójkątów. Na wyjściu potoku wysyłany jest finalny obraz odwzorowujący widoczną scenę po odpowiednich przekształceniach. Poszczególne etapy wykonują takie operacje jak transformacje, skalowanie czy obroty, ale też kolorowanie, nakładanie tekstuur, aż do finalnego przetworzenie widocznych punktów na obraz płaski.

W projekcie wykorzystywany był podziób OpenGL zaprojektowany dla systemów wbudowanych - OpenGL ES (z ang. OpenGL for Embedded Systems) [22]. Ze względu na ograniczone zasoby sprzętowe takich systemów, rozszerzenie to koncentruje się na usprawnieniach związanych z wydajnością oraz na umożliwieniu wykonywania obliczeń w czasie rzeczywistym. Z tego powodu pasuje do zagadnienia wykorzystania procesu identyfikacji szczytów górskich przy pomocy urządzeń mobilnych w czasie rzeczywistym. Dodatkowo, używany był również interfejs EGL (z ang. Embedded-System Graphics Library) [23]. Podobnie jak OpenGL ES, stworzony został z myślą o systemach wbudowanych. Służy on do tworzenia kontekstu graficznego oraz powierzchni renderingu. W projekcie, dzięki niemu, możliwe było generowanie obrazów przy pomocy OpenGL bez konieczności wyświetlania ich na ekranie (tzw. offscreen rendering).

Na wybór OpenGL jako interfejsu do generowania modelu terenu wpływ miała powszechność i popularność tego rozwiązania. Zapewnia on szybkie i stabilne działanie, a także opisany jest bogatą dokumentacją. Wspierany jest on przez dużą liczbę producentów sprzętu, systemów operacyjnych czy oprogramowania.

3.2. OpenCV

Do celów manipulacji obrazami oraz wykonywania na nich obliczeń wybrana została biblioteka OpenCV - (ang. Open Source Computer Vision Library) [24]. Dostarcza mnogi wybór dobrze udokumentowanych algorytmów przetwarzania cyfrowego obrazu. Dzięki temu, że została stworzona z użyciem języka C/C++, zapewnia wysoką szybkość działania wykorzystując do tego niskopoziomowe mechanizmy. Może być użyta również w środowiskach opartych o Javę czy Pythona, ponieważ stworzone są dla niej wiązania do wielu popularnych języków programowania. Dostarcza ona także odpowiednie obiekty przechowujące obraz w pamięci, które w prosty sposób można modyfikować.

Ze względu na to, że jest to powszechnie uznana biblioteka została wytypowana jako główny element pracy magisterskiej związanej z przetwarzaniem obrazu. Ważnym czynnikiem przy tym wyborze był również szeroki wachlarz dostarczanych metod, które mogły zostać wykorzystane w trakcie prac badawczych czy tworzenia platformy testowej. Algorytmy takie często opatrzone są odpowiednimi przypisami bibliograficznymi do artykułów, na podstawie których były implementowane.

3.3. System operacyjny Android

Docelowym systemem, na który implementowana była aplikacja testowa na urządzenia mobilne był *Android*. Jest to system opracowany przez firmę Google i systematycznie rozwijany od 2007 roku. Obecnie jest on powszechnie stosowany w wielu urządzeniach mobilnych, ale także innych akcesoriach elektronicznych życia codziennego - np. zegarkach, telewizorach, systemach audio czy nawet lodówkach. Opiera się on na odpowiednio zmodyfikowanym jądrze Linux, co pozwala na wykorzystanie narzędzi oraz bibliotek pierwotnie pisanych z myślą o tym systemie.

Wybór platformy Android został podyktowany ogromną liczbą urządzeń mobilnych z tym systemem w skali światowej. Dzięki temu, rozwiązanie bazujące na opracowywanym oraz testowanym w ramach pracy dyplomowej procesie mogłoby dotrzeć do większej liczby potencjalnych odbiorców. Dodatkowo, dostępna jest duża liczba artykułów i dokumentacji związanych z projektowaniem oraz implementacją programów na ten system, co może mieć wpływ na szybsze rozwiązywanie ewentualnych problemów z tworzonym oprogramowaniem testowym. Z tego powodu weryfikacja działania projektowanego w ramach pracy procesu odbyła się na tej platformie.

Mimo, że eksperymenty przeprowadzane były na projekcie stworzonym na konkretny system operacyjny to możliwe jest, dzięki opracowaniu odpowiednich metod opisanych w pracy dyplomowej, dostosowanie procesu rozpoznawania szczytów górskich na dowolne urządzenie działające pod innym oprogramowaniem systemowym. Warunkiem koniecznym do spełnienia jest dostępność opisanych sensorów i odpowiednich dla nich interfejsów API.

3.4. Sensory

W ramach platformy testowej pracy dyplomowej wykorzystane zostały dostępne w mobilnych urządzeniach sensory. Były to przede wszystkim globalny system pozycjonowania oraz czujnik obrotu.

Globalny System Pozycjonowania (GPS). GPS to system nawigacyjny wykorzystujący satelity umieszczone na orbicie okoziemskiej [25]. Dzięki określaniu czasu dotarcia emitowanych sygnałów radiowych możliwe jest obliczenie położenia danego urządzenia. System Android udostępnia interfejs umożliwiający pobieranie aktualnych danych o lokalizacji urządzenia w czasie rzeczywistym. Oprócz informacji dotyczących położenia takich jak długość i szerokość geograficzna zwracane są również dodatkowe dane: wysokość na jakiej znajduje się odbiornik czy prędkość z jaką przemieszcza się urządzenie [26].

Sensor ten może pracować w dwóch trybach: dokładnym i przybliżonym. Na potrzeby pracy dyplomowej został wybrany tryb dokładny, dzięki czemu określone położenia urządzenia są bardziej precyzyjne. Przekłada się to na dopasowywanie map topograficznych z większą skutecznością.

Wektor obrotu. Kolejnym czujnikiem wykorzystywanym w projekcie jest wektor obrotu. Jest to rodzaj fikcyjnego sensora, który umożliwia określenie orientacji telefonu w trzech wymiarach. Wykorzystuje do swojego działania dwa lub trzy (w zależności od wybranego trybu) czujniki fizyczne: akcelerometr, żyroskop oraz czujnik pola geomagnetycznego. Pozwala on określić kąt odchylenia urządzenia wzduż trzech osi układu XYZ [27]. Wynikowy kąt zwracany jest w postaci kwaternionu:

$$\begin{pmatrix} \cos(\theta/2) \\ x * \sin(\theta/2) \\ y * \sin(\theta/2) \\ z * \sin(\theta/2) \end{pmatrix}$$

Dzięki wykorzystaniu tego czujnika możliwe jest stwierdzenie, w którym kierunku i pod jakim kątem obserwator ogląda dane pasmo górskie.

Istnieją trzy tryby pracy wektora obrotów:

- Użycie wszystkich trzech czujników fizycznych. Zwracanym rezultatem jest kąt o wartości absolutnej względem ziemi.
- Użycie jedynie akcelerometru i żyroskopu. Kąt Y nie wskazuje wartości bezwzględnej, a względną do pewnej wartości referencyjnej. Skutkuje to brakiem informacji o odchyleniu względem północnego bieguna ziemi.
- Użycie jedynie akcelerometru i czujnika pola geomagnetycznego. Jest to mniej dokładna odmiana pierwszego trybu, zwracająca wartości bezwzględne ziemi.

3.5. Języki programowania

Wybór odpowiedniego języka programowania jest kluczowym elementem większości projektów związanych z wytwarzaniem oprogramowania. Stworzona w ramach projektu dyplomowego aplikacja na system Android będąca platformą testową została napisana z wykorzystaniem w głównej mierze języka Java, zapewniającego wiele przydatnych bibliotek czy narzędzi. Natomiast do tworzenia interfejsu graficznego wykorzystano język znaczników XML. Służy on do opisu struktury, stylizacji, a także rozmieszczenia interfejsu użytkownika. W prosty sposób pozwala na hierarchiczne tworzenie UI (z ang. User Interface).

Prócz Javy oraz XML, w trakcie prac wykorzystany był również język Python. Jest to język wysokiego poziomu, dzięki czemu pozwala on w szybki sposób prototypować hipotetyczne rozwiązania. Miało to niebagatelny wpływ na testowanie wybranych algorytmów i rozwiązań, ponieważ pozwalało na wstępna ich weryfikację w skróconym czasie.

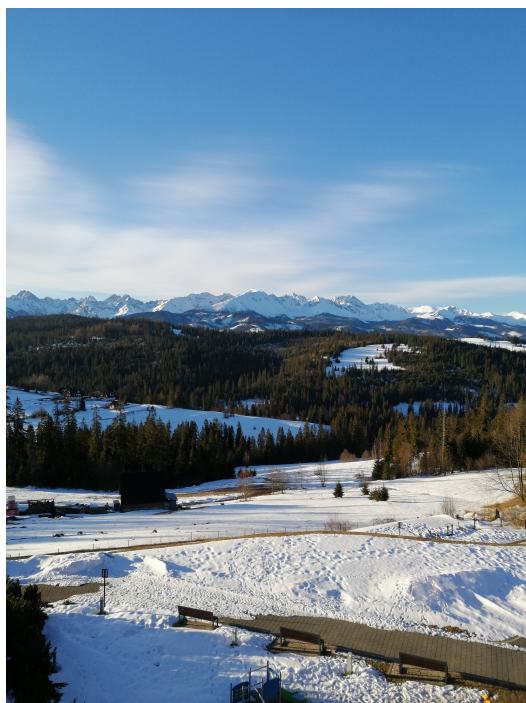
4. Zestaw statycznych zdjęć do celów testowych

Na potrzeby przeprowadzonych w ramach pracy dyplomowej testów przygotowany został zestaw statycznych zdjęć. Do tego celu stworzono prostą aplikację na urządzenia mobilne, która podczas wykonywania zdjęcia zapisywała dodatkowe informacje.

Zbierane były następujące dane:

- Pole widzenia (FoV, z ang. Field of View) - szerokość widzenia obiektywu (charakterystyczna dla danego urządzenia i matrycy aparatu) w pionie i poziomie
- Lokalizacja - położenie w postaci szerokości i długości geograficznej
- Wysokość - wysokość nad poziomem morza względem elipsoidy WGS84 [28]
- Kąty obrotu - absolutne wartości odchyлеń urządzenia względem ziemi wokół trzech osi układu odniesienia
- Orientacja - zmienna logiczna, określająca czy urządzenie było w trybie orientacji pionowej (ang. portrait) czy poziomej (ang. landscape)

W ramach tej części pracy dyplomowej zostało zebranych kilkadziesiąt zdjęć gór i szczytów w paśmie górskim Tatr. Zdjęcia zostały wykonane przez autora pracy dyplomowej w okolicach Zakopanego oraz Bukowiny i Białki Tatrzańskiej. Były one zróżnicowane pod względem miejsca oraz wysokości nad poziomem, z której były wykonywane, pory dnia, warunków pogodowych oraz widocznych szczytów znajdujących się na obrazie. Przykładowe zdjęcie oraz odpowiadające mu dane zostały pokazane na rysunku 4.1 i w tabeli 4.1. Na obrazie widoczne są szczyty takie jak: Wielka Koszysta, Świnica czy Rysy. Oddalone są one od obserwatora o około 12 – 18km.



Rysunek 4.1. Przykładowe zdjęcie gór i terenu zebrane w paśmie górskim Tatr.

Tabela 4.1. Zebrane dane dotyczące przykładowego zdjęcia w paśmie górskim Tatr.

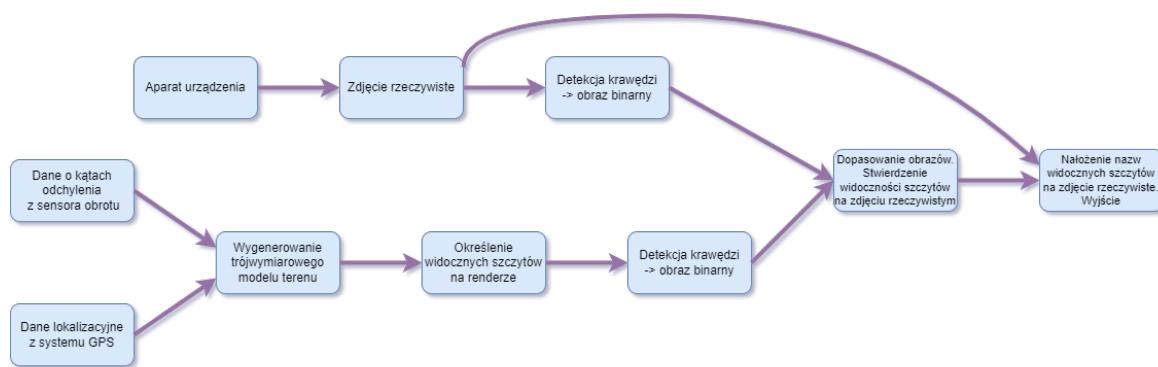
Kąty widzenia (FoV)	Położenie geograficzne	Kąty obrotu	Orientacja
65,8946° 51,8465°	49,3393948 N 20,0812903 E 1006,09 m n.p.m	-167,37733° 1,7193204° -0,98883315°	Pionowa

Wszelkie ilustracje oraz wizualizacje prezentowane w dalszej części pracy dyplomowej odnoszą się głównie do przedstawionego zdjęcia oraz parametrów zebranych podczas jego wykonywania. Przykładowo, rysunek w rozdziale 5.1, na którym prezentowany jest trójwymiarowy model terenu wygenerowany w ramach prototypu zawiera wizualizację widocznych na tym zdjęciu gór.

Na podstawie zebranych danych generowany był później trójwymiarowy model terenu. W teorii wizualizował on przestrzeń zbliżoną do tej, która była widoczna z perspektywy obserwatora na wykonanym zdjęciu. Pozwalało to testować proces identyfikacji szczytów górskich symulując wejściowe klatki nagrani pochodzących z aparatu urządzenia.

5. Przegląd algorytmów i szczegóły implementacyjne

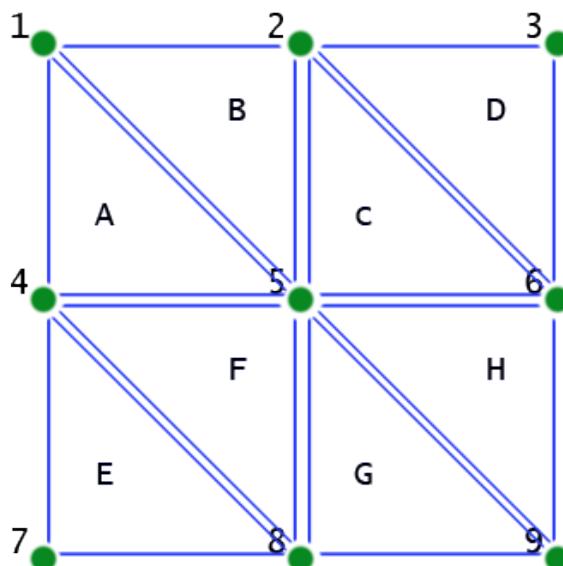
Zaimplementowany w ramach platformy testowej proces rozpoznawania szczytów przebiega w podobny sposób, jak opisane w rozdziale 2.6 podejście teoretyczne na podstawie analizy literaturowej. Na początku zbierane są dane z sensorów dotyczące kątów obrotu urządzenia oraz jego położenia GPS. Wykorzystując lokalizację określany jest współczynnik skalowania modelu związanego ze zmianą wielkości pojedynczych kratek siatki terenu w zależności od szerokości geograficznej. Następnie generowany jest trójwymiarowy model ziemi z wykorzystaniem biblioteki OpenGL, a kamera ustawiana jest zgodnie z zebranymi danymi z czujników. Dla takiej wizualizacji określone są widoczne na niej szczyty górskie filtrując listę gór pochodząą ze zbioru danych GeoNames. Zarówno dla wygenerowanej panoramy, jak i wykonanego za pomocą aparatu urządzenia zdjęcia, wykrywane są krawędzie gór tworząc dwa obrazy binarne. Następnie wykorzystując metody dopasowania i sprawdzania podobieństwa między obrazami oraz listę widocznych wierzchołków na trójwymiarowym modelu stwierdzana zostaje ich widoczność na zdjęciu rzeczywistym. Odbywa się to na wygenerowanych obrazach binarnych, z których pobiera się odpowiedni wycinek w zależności od badanego szczytu. Dla tych gór, dla których stwierdzone zostanie podobieństwo z odpowiednio dużym prawdopodobieństwem poprawności nakładane są etykiety nazw na zdjęcie wejściowe. Tak zmodyfikowany obraz zostaje wysłany na wyjście procesu, a finalnie wyświetlony użytkownikowi. W przypadku działania na urządzeniu mobilnym w czasie rzeczywistym cykl ten powtarzany jest dla każdej kolejnej klatki nagrania zarejestrowanego na żywo. Diagram przedstawiający ten proces przedstawiono na rysunku 5.1. Względem diagramu teoretycznego rozszerzony został on o dane pochodzące z sensorów oraz wykorzystanie aparatu urządzenia do pobrania kolejnych klatek obrazu.



Rysunek 5.1. Przepływ danych i sterowania procesu identyfikacji szczytów górskich z uwzględnieniem sensorów i urządzeń.

5.1. Generowanie trójwymiarowego modelu terenu

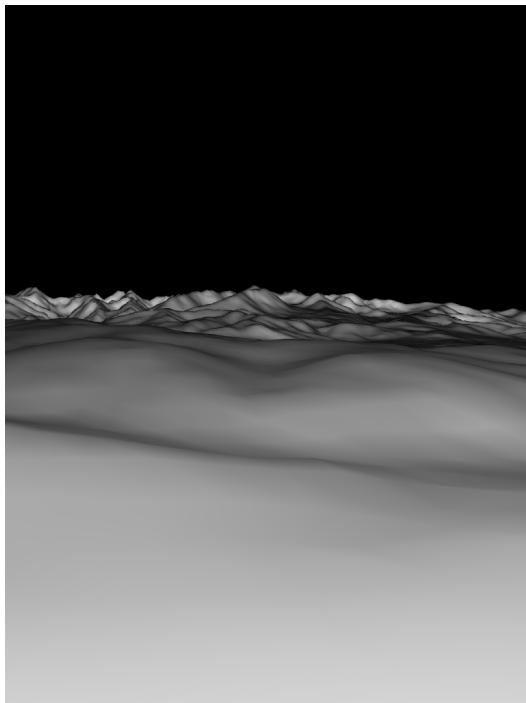
Dane SRTM zapisane są w plikach jako ciąg wartości posortowanych zachód-wschód oraz północ-południe, które w ramach implementacji platformy testowej konwertowane są do postaci tablicy dwuwymiarowej. Numery rzędów i kolumn odpowiadają kolejno szerokości i długości geograficznej danego punktu, a wartości w poszczególnych komórkach tabeli to wysokość nad poziomem morza w danym miejscu. Takie przeskalowane „trójkąty” (rzad, kolumna, wartość) odpowiadają współrzędnym trójwymiarowym OpenGL, a w praktyce tworzą poszczególne wierzchołki modelu. W OpenGL prymitywy tworzy się z siatki trójkątów i w ten sam sposób generowana jest mapa terenu w projekcie - trójkąty zostają utworzone z kolejnych trzech posortowanych wierzchołków. Wizualizacja takiej siatki dla danych SRTM pokazana jest na rysunku 5.2. Numerowane punkty odpowiadają kolejnym wartościom zapisanym w tym zbiorze.



Rysunek 5.2. Wizualizacja tworzenia siatki trójkątów między punktami ze zbioru SRTM.

Dla tak stworzonych prymitywów dodawane jest cieniowanie zależne od położenia światła oraz wysokości poszczególnych punktów. Wymaga to obliczenia wektorów normalnych dla każdego wierzchołka. Uzyskuje się to poprzez normalizację sumy takich wektorów wszystkich trójkątów, w których skład wchodzi dany wierzchołek (wierzchołek częściowo opisuje od dwóch do sześciu trójkątów). Dzięki temu możemy w łatwy sposób kolorować i cieniować teren, co ma wpływ na detekcję krawędzi w dalszej części projektu, a także na poprawę walorów wizualnych przy ewentualnym podglądzie modelu.

Przykładowa wizualizacja trójwymiarowego terenu pokazano na rysunku 5.3. Przedstawia ona wygenerowany model z ustawieniami lokalizacji i kątów obrotu odpowiadającym perspektywie zdjęcia 4.1.



Rysunek 5.3. Przykład wizualizacji terenu z wykorzystaniem interfejsu OpenGL i danych SRTM.

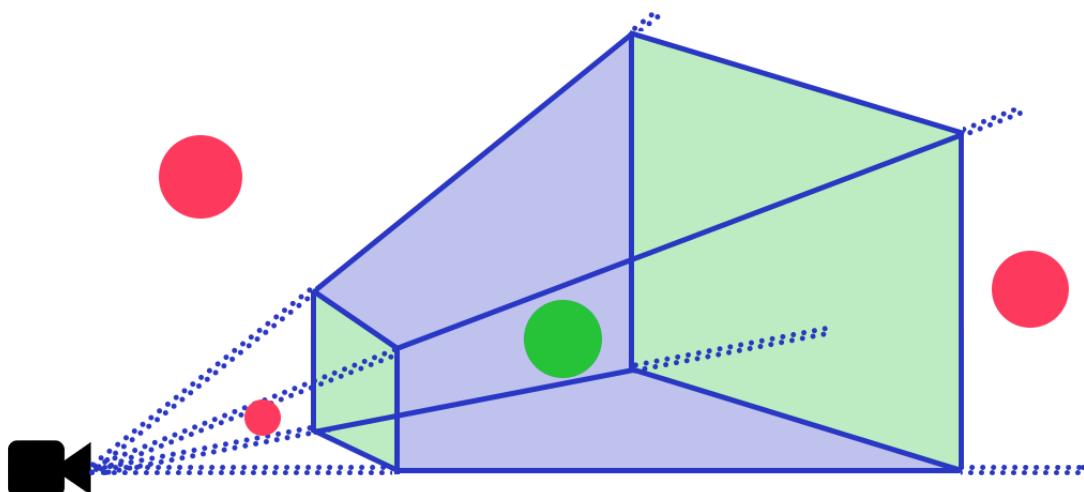
5.2. Widoczność szczytów górskich na trójwymiarowym modelu terenu

Na model trójwymiarowy nanoszone są dane dotyczące szczytów górskich uzyskane ze zbioru danych GeoNames. Zawierają one szerokość i długość geograficzną oraz wysokość nad poziomem morza dla każdego szczytu, które są przeliczane na odpowiadające położenie w przestrzeni wygenerowanego terenu. Następnie, uzyskane w ten sposób wartości mapowane są na konkretny wierzchołek modelu. Tak przygotowane dane tworzą listę potencjalnie widocznych na zdjęciu wejściowym szczytów górskich. Ze względu jednak na to, że niektóre z nich nie są widoczne na wizualizacji z perspektywy obserwatora musi zostać ona przefiltrowana.

Z tego powodu, na podstawie znajomości zagadnienia grafiki trójwymiarowej przez autora, został zaproponowany prosty, dwuetapowy proces filtracji listy potencjalnie widocznych wierzchołków. Oba kroki bazują na przystosowanych na potrzeby projektu metodach optymalizacji procesu renderingu poprzez usuwanie powierzchni niewidocznych. Jeśli potraktujemy szczyty górskie jako prymitywy i obiekty w generowanej scenie, to wykorzystanie takich algorytmów do określania ich widoczności wydaje się w pełni logiczne i uzasadnione. W pierwszym etapie sprawdzane jest położenie poszczególnych punktów względem przestrzeni widzenia kamery. Następnie weryfikowana jest widoczność danego wierzchołka w kontekście bycia zasłanianym przez inne elementy terenu. Po przejściu tego procesu stwierdzana jest widoczność poszczególnych szczytów na wygenerowanym modelu.

5.2.1. Filtracja szczytów poza polem widzenia kamery

Pierwszym etapem filtrowania szczytów jest *Frustum Culling* [29]. Algorytm ten jest jedną z najpopularniejszych metod wykorzystywanych w grafice trójwymiarowej, której celem jest określenie wierzchołków znajdujących się w polu widzenia kamery. Dzięki temu pozwala usunąć obiekty, które nie znajdują się w zasięgu wzroku obserwatora. Polega on na projekcji współrzędnych trójwymiarowych na płaszczyznę dwuwymiarową obrazu. Na podstawie położenia kamery, kątów widzenia, kierunku patrzenia oraz innych, specyficznych dla danego rodzaju projekcji atrybutach tworzona jest matryca *MVP* (z ang. Model View Projection [30]). Opisuje ona przestrzeń widoczną dla kamery, a wizualnie charakteryzuje ostrosłup lub stożek ścięty. Przykładową wizualizację bryły widoczności dla projekcji perspektywicznej pokazano na rys. 5.4. Widoczne na niej obiekty w zacienionej przestrzeni - zielone koło - są odwzorowywane na obrazie, natomiast reszta jest usuwana - czerwone koła. Jeśli dany wierzchołek po przekształceniach z wykorzystaniem macierzy *MVP* ma prawidłowe współrzędne dwuwymiarowe - znajdujące się w zakresie szerokości i wysokości tworzonego obrazu - to jest on widoczny dla obserwatora.



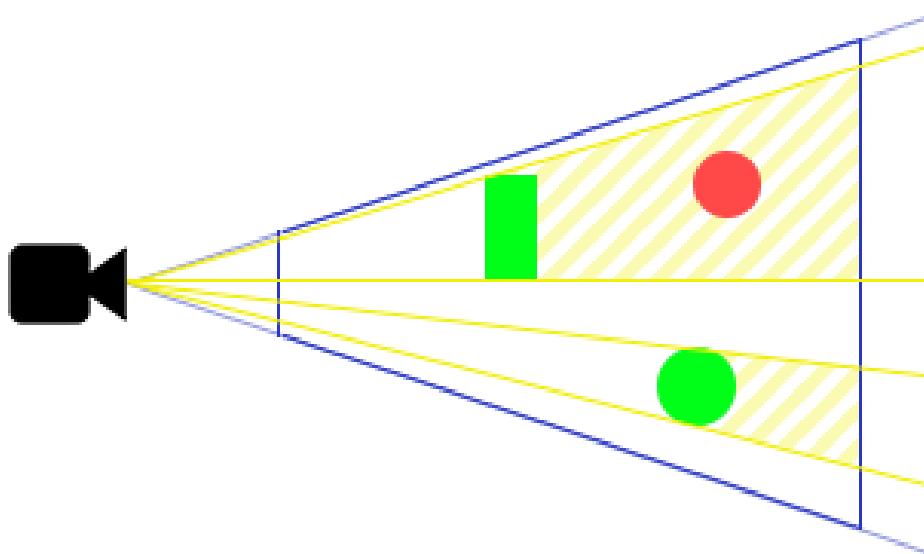
Rysunek 5.4. Wizualizacja frustum culling. Ilustracja stożka widoczności kamery przy projekcji perspektywicznej. Odwzorowane zostaną tylko obiekty znajdujące się w przestrzeni stożka ściętego - zacieniona część. Czerwone koła będące poza obszarem widzenia kamery zostaną odrzucone.

Technika ta jest powszechnie stosowana w renderingu trójwymiarowym do optymalizacji tego procesu, ponieważ pozwala zmniejszyć liczbę wierzchołków oraz obiektów, które muszą być przetworzone i narysowane w generowanej scenie.

Na potrzeby pracy dyplomowej algorytm frustum culling został zaadaptowany do celów filtracji szczytów górskich, które na pewno nie są widoczne z perspektywy obserwatora, niezależnie od wygenerowanego terenu. Pozwala to znacznie zredukować listę potencjalnie widocznych szczytów na obrazie.

5.2.2. Filtracja zasłanianych szczytów

Dla pozostałych szczytów, które znajdują się w przestrzeni widocznej dla kamery, przeprowadzany jest tzw. *Occlusion Culling* [31]. Jest to kolejna technika optymalizacji procesu renderingu trójwymiarowego, której celem również jest odrzucenie struktur, które nie są widoczne z pozycji obserwatora. W tym wypadku znajdują się one jednak w stożku widoczności ale położone są za innymi obiektami, więc są przez nie przesłaniane. Powoduje to, że nie muszą być przetwarzane i rysowane. Najprostsza implementacja wykorzystuje *test głębokości* charakteryzujący to, jak blisko kamery znajduje się dany punkt, a co za tym idzie czy zasłania inne piksele (takie, które uzyskują mniejszą wartość współczynnika głębokości).



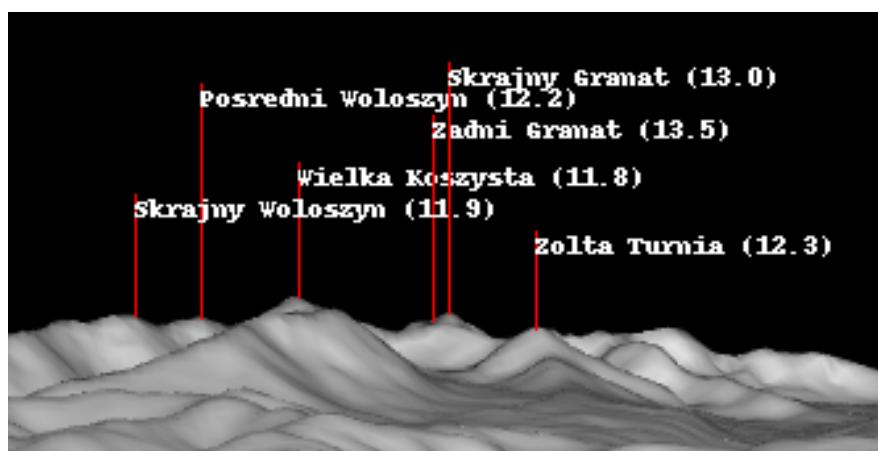
Rysunek 5.5. Wizualizacja Occlusion Culling. Czerwone koło jest przesłaniane przez zielony prostokąt, więc zostaje usunięte z procesu renderingu.

W ramach opracowywanego procesu identyfikacji gór na obrazie occlusion culling pozwala sprawdzić, które wierzchołki odpowiadające poszczególnym szczytom przechodzą test głębokości. Jeśli dana góra uzyska dodatni wynik takiej próby to jest ona widoczna i powinna zostać oznaczona na obrazie. Test ten przeprowadzany jest z wykorzystaniem bufora zapytań (ang. queries [32]) w OpenGL. Taki mechanizm pozwala sprawdzać wynik wykonania danego zapytania. W projekcie sprawdzany jest stan wierzchołków (*GL_SAMPLES_PASSED*) po przeprowadzeniu testu głębokości. Dla każdego szczytu wykonywana jest kwerenda i zależnie od odpowiedzi może on zostać usunięty z listy potencjalnych szczytów górskich.

Uzyskana po tym etapie lista zawiera szczyty, które na pewno są widoczne z perspektywy obserwatora na wygenerowanym trójwymiarowym modelu terenu. Następnym elementem jest dopasowanie ich do widocznych na zdjęciu wejściowym gór.

5.2.3. Przykładowy rezultat widoczności szczytów górskich na wygenerowanym modelu

Na rysunku 5.6 przedstawiono wcześniej wygenerowany model terenu (rys. 5.3) z nanesionymi nazwami widocznych na nim szczytów górskich. Ich widoczność została stwierdzona przy wykorzystaniu wyżej opisanego procesu filtrowania listy potencjalnie widocznych górz. Ze względu na duże zagęszczenie szczytów w widocznym na modelu pasmie górkim, liczba wyświetlanych nazw została ograniczona celem poprawy czytelności. Dodatkowo obraz został wykadrowany z powodu zajmowania dużej części obrazu przez niebo oraz nieistotne elementy terenu.



Rysunek 5.6. Przykładowy rezultat identyfikacji szczytów górskich na trójwymiarowym modelu terenu.

5.3. Wyznaczanie odległości między dwoma punktami geograficznymi

Jednym z obliczeń wykonywanych w trakcie procesu identyfikacji szczytów górskich wykonywanego na platformie testowej jest wyznaczanie odległości między dwoma punktami geograficznymi. Wyniki takich działań wykorzystywane są do ustalania odpowiedniej skali modelu trójwymiarowego względem rzeczywistości, ale również do wyliczenia odległości między obserwatorem, a widocznymi szczytami. Dlatego istotne było wybranie odpowiedniego algorytmu, który przy małej złożoności obliczeniowej będzie osiągać niski błąd. Przybliżone w następnym podrozdziale algorytmy były testowane w dalszej części pracy dyplomowej.

5.3.1. Przegląd wybranych algorytmów

W literaturze dostępnych jest wiele metod służących do obliczenia odległości między dwoma położeniami geograficznymi. Niektóre z nich wykorzystują elementy przybliżenia lub traktują kule ziemską jako inne figury trójwymiarowe, np. sferę lub walec. Na potrzeby pracy dyplomowej przybliżone zostały cztery algorytmy: sferyczne prawo cosinusów, metoda Haversiana, przybliżenie walcowe i iteracyjny algorytm Vincenta.

We wzorach poniżej występują następujące oznaczenia:

- **d** - odległość między dwoma punktami geograficznymi.
- **R** - promień ziemi. Ze względu na różnicę pomiędzy promieniem równikowym i biegunowym w projekcie wykorzystywany jest średni promień ziemi o długości $R = 6371,0$ km [33].
- **lat_i** - (ang. latitude) szerokość geograficzna punktu i
- **long_i** - (ang. longitude) wysokość geograficzna punktu i

Sferyczne Prawo Cosinusów. Jednym z podstawowych praw trygonometrii sferycznej jest *Sferyczne Prawo Cosinusów* (ang. Spherical Law of Cosines). [34] Jest to odpowiednik Prawa Cosinusów z geometrii płaskiej. Wykorzystywane jest do obliczania odległości między dwoma punktami na powierzchni sfery. Przy pomocy tej metody można uzyskać wynik charakteryzujący się dużą dokładnością, nawet przy małych odległościach.

$$d = R * \arccos(\sin lat_1 * \sin lat_2 + \cos lat_1 * \cos lat_2 * \cos(long_2 - long_1))$$

Haversian. Metoda *Haversian* [34], [35] jest jedną z najczęściej wybieranych przy projektowaniu aplikacji wykorzystujących GPS czy bazujących na modelu ziemi. Została oparta na trygonometrii sferycznej jako jej uogólnienie. Pierwsze tabele haversianów były opisywane już na początku XIX wieku, ale termin ten został sprecyzowany i opisany dopiero przez James'a Inmana w 1835 roku.

$$d = R * 2,0 * \text{atan}2(\sqrt{a}, \sqrt{1,0 - a})$$

gdzie a dane jest wzorem:

$$\begin{aligned} a = & \sin((lat_2 - lat_1)/2,0)^2 \\ & + \cos lat_1 * \cos lat_2 * \sin((lat_2 - lat_1)/2,0) * \sin((long_2 - long_1)/2,0) \end{aligned}$$

Przybliżenie walcowe równoodległościowe. Nastecną z zaprezentowanych metod jest *Przybliżenie Walcowe Równoodległościowe* (ang. Equirectangular Approximation). [34] Algorytm ten opiera się na uproszczeniu, że Ziemia jest idealną sferą i traktuje ją jako płaszczyznę walcową. Sprawia to, że punkty o równej odległości względem równika na kuli ziemskiej są również tak samo odległe od równika na mapie. Dzięki takim założeniom zmniejszona zostaje złożoność obliczeniowa. Z tego powodu metoda ta jest często stosowana przy ograniczonych zasobach sprzętowych, na przykład w systemach wbudowanych.

$$d = R * \sqrt{((long_2 - long_1) * \cos((lat_1 + lat_2)/2,0))^2 + (lat_2 - lat_1)^2}$$

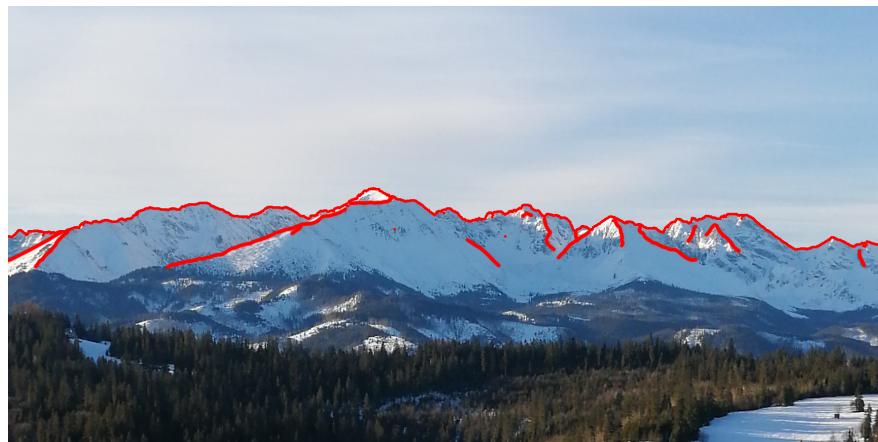
Algorytm Vincenta. Ostatnim z opisywanych podejść jest iteracyjny algorytm Vincent'a. Jest to jedna z najbardziej zaawansowanych metod obliczania odległości geograficznej dającą bardzo dokładne rezultaty. Wadą tego rozwiązania jest duża złożoność, zarówno obliczeniowa jak i w kontekście opisu teoretycznego. Z tego powodu nie jest ona w pełni przybliżana w tej pracy dyplomowej. Dogłębnie opisana została w opracowaniu [36].

5.4. Detekcja krawędzi góra

Kolejnym etapem jest detekcja krawędzi na wygenerowanej panoramie oraz na rzeczywistym zdjęciu. Jest to istotny element procesu ze względu na to, że wyniki dla obu obrazów są do siebie porównywane, a następnie stwierdzana jest widoczność na nich poszczególnych szczytów.

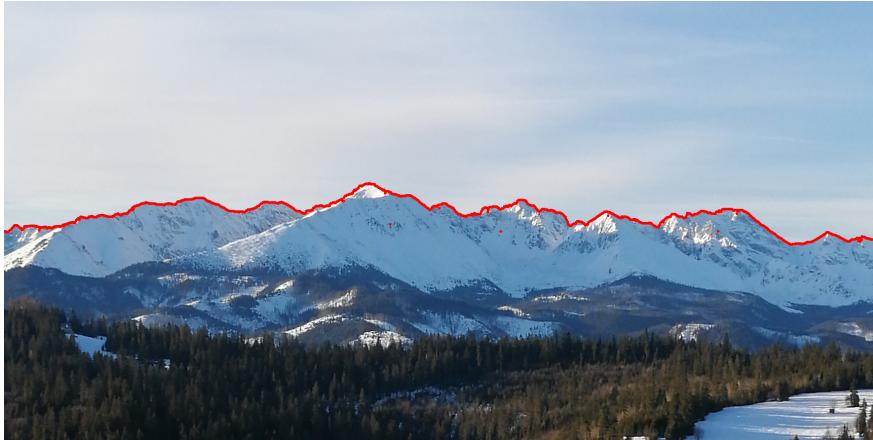
5.4.1. Oczekiwany rezultat

Wstępnie w ramach przeprowadzonych badań planowano wykrywać wszelkie krawędzie gór. Przykładowy oczekiwany rezultat zaprezentowano na rysunku 5.7. Jednak opracowanie odpowiedniej metody lub dobranie parametrów znanych algorytmów, tak by osiągnąć założony na początku cel okazało się niemożliwe w racjonalnej perspektywie czasu w kontekście tworzenie pracy dyplomowej magisterskiej, której tylko jednym z elementów jest rozpoznawanie konturów. Głównym problemem było znalezienie uniwersalnego rozwiązania pozwalającego wykrywać w ten sposób krawędzie niezależnie od warunków pogodowych, jakości zdjęć czy innych zakłóceń.



Rysunek 5.7. Przykład wstępnie oczekiwanej rezultatu detekcji krawędzi góra.

Z tego względu zdecydowano się na ograniczenie detekcji krawędzi do najwyższych gór tworzących linię nieba. Jest to podejście podobne do opisanego w [5]. Przykład wykrytych w ten sposób konturów pokazano na rysunku 5.8. Konsekwencją takiego uproszczenia jest analiza jedynie szczytów górskich znajdujących się na tej linii oraz w bliskim jej sąsiedztwie ze względu na okno analizy o pewnym rozmiarze.



Rysunek 5.8. Przykład uproszczonej detekcji krawędzi gór.

Ma to szczególnie znaczenie przy zdjęciach gór robionych z bliskiej odległości, gdy zajmują one dużą część obrazu w pionie oraz zawierają widoczne pasma górskie zróżnicowane pod względem wysokości. W takim przypadku efektem będzie pominięcie analizy wierzchołków znajdujących się w dolnej części zdjęcia, co finalnie oznacza niemożność ich identyfikacji.

5.4.2. Algorytm Canny

W ramach badań oraz platformy testowej, do celów detekcji krawędzi, zdecydowano się wybrać algorytm Canny [37]–[39]. Mimo, że została opracowana w 1986 roku to dalej jest szeroko stosowaną metodą w przetwarzaniu cyfrowym obrazu rozwiązującą problem rozpoznawania konturów czy zmian w intensywności pikseli. Składa się ona z czterech głównych etapów:

1. **Usunięcie szumów.** Odbiera się to poprzez wygładzenie obrazu z wykorzystaniem filtru Gaussowskiego [40] o rozmiarze jądra 5x5. Pozwala to usunąć szum w postaci pojedynczych wyróżniających się pikseli. Efektem ubocznym jest bardziej rozmyte zdjęcie.
2. **Obliczenie siły i kierunku gradientu pikseli.** Na wygładzonym obrazie zastosowany zostaje operator Sobela [41] w pionie i poziomie celem określenia pierwszej pochodnej w obu tych kierunkach. Operator ten opiera się na splocie z macierzą 3x3 wypełnioną wartościami zależnymi od orientacji. Następnie dla tych wyników oblicza się gradient oraz jego kierunek.
3. **Tłumienie nie-maksymalne.** (ang. Non-maximum Suppression) [42] Polega na badaniu lokalnych maksymów dla znalezionych krawędzi. Dla piksela tworzącego dany kontur porównuje się go z wynikiem sąsiadów wzdłuż gradientu. Jeśli dany punkt jest lokalnym maksimum to uznawany jest za składową krawędzi. W innym przypadku jest odrzucany. Pozwala zachować piksele z najbardziej widoczną zmianą intensywności w danym otoczeniu.

4. Progowanie Histerezowe. (ang. Hysteresis Thresholding) [43] Jest to filtrowanie krawędzi w zależności od siły gradientu. Opiera się na dwóch progach. Jeśli piksel znajduje się powyżej tych wartości to jest on uznawany za pewną (mocną) krawędź, natomiast jeśli poniżej to jest odrzucany. Piksele, których wartość gradientu znajduje się między progami zostają uznane za kontur w zależności od tego czy są połączone z jakąś mocną krawędzią, czy nie.

Parametry progowania histerezowego w implementacji platformy testowej zostały ustawione w taki sposób, żeby odrzucać głównie zaszumienie w części zdjęcia zawierającej niebo. Związane jest to z wcześniej opisanym problemem i decyzją o analizie jedynie najwyższych gór. Dla wynikowego obrazu binarnego wyszukuje się najwyższe krawędzie odrzucając te znajdujące się poniżej. Dlatego szum i dodatkowe wskazania w niższej części obrazu nie mają tak wielkiego znaczenia jak te na niebie, ponieważ nie są brane w ogóle pod uwagę w trakcie analizy. Uzyskany efekt zawiera pojedynczy kontur w danej kolumnie zdjęcia tak jak na rysunku 5.8. Taki sam schemat detekcji krawędzi stosowany jest w przypadku panoramy trójwymiarowego modelu terenu. W tym przypadku dobranie parametrów było dużo prostsze ze względu na stałe odwzorowanie kolorów takiego modelu.

5.5. Dopasowanie obrazów binarnych

Pomijając dodawanie nazw widocznych szczytów na zdjęciu rzeczywistym, dopasowanie obrazów binarnych celem stwierdzenia widoczności owych gór jest ostatnim etapem opracowywanego procesu ich identyfikacji. Wydaje się on również najważniejszym w kontekście całego projektu, ponieważ to wyniki porównania ze sobą obrazów wygenerowanego modelu oraz faktycznego zdjęcia gór wpływają na odrzucenie lub etykietowane poszczególnych szczytów. Jeśli dany algorytm nie będzie w stanie dopasować do siebie badanych zdjęć z odpowiednio wysoką pewnoścą, szczyt może być niesłusznie uznany za niewidoczny, lub odwrotnie, niewidoczny szczyt jako widoczny. Z tego powodu, analiza tego zagadnienia oraz dostępnych algorytmów była bardziej dogłębna, a metody lepiej przetestowane.

Operacja dopasowania obrazów zawierających krawędzie gór odbywa się na ich wycinkach o pewnym rozmiarze. W przypadku obrazu binarnego wyrenderowanego modelu centralnym punktem takiego fragmentu był piksel odpowiadający wierzchołkowi danego szczytu. Natomiast, dla zdjęcia wejściowego centrum było estymowane jako takie samo jak wizualizacji. Można przyjąć, że jest to miejsce hipotetycznego szczytu, gdyby wskazania sensorów urządzenia były idealne. Jednak ze względu, że w rzeczywistości takie mogą nie być, wycinek ten jest większy niż obrazu wygenerowanego terenu. Powiększony on jest o pewną tolerancję, celem zwiększenia pewności zawierania się w nim widocznego szczytu.

W poniższych podrozdziałach opisano dwie metody, które w ramach pracy dyplomowej były badane. Są to algorytmy oparte na dopasowaniu: wzorca (ang. template matching) oraz cech (ang. feature matching). Pierwsza z nich bada wartości pojedynczych pikseli, natomiast druga bierze pod uwagę kontekst poszczególnych punktów

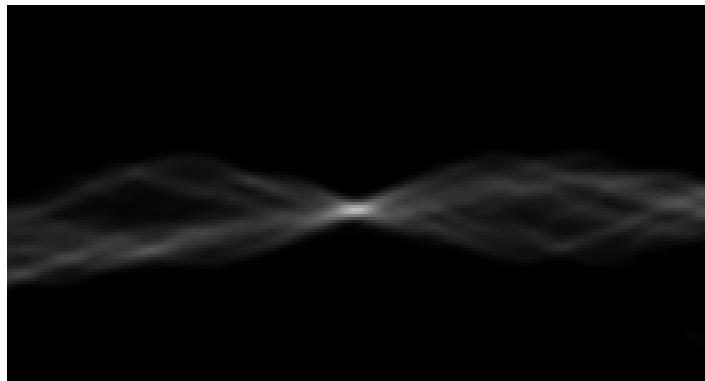
5.6. Dopasowanie na podstawie wzorca

Pierwszą z opisanych metod porównywania obrazów jest dopasowanie na podstawie wzorca (czasem określana także jako dopasowanie szablonu) [44]–[48]. Do swojego działania wykorzystuje metody obliczania korelacji między dwoma zbiorami (w tym wypadku pikseli) dla każdego położenia. Dla danego położenia okna wynikiem takich obliczeń jest podobieństwo między dwoma obrazami. Finalnie, metoda ta zwraca najbardziej prawdopodobną lokalizację, w której szablon może znajdować się na zdjęciu wejściowym oraz jak pewne jest ich podobieństwo.

Ze względu na wykorzystanie jedynie numerycznych form porównania, metoda ta jest podatna na różne zniekształcenia między obrazami takie jak skalowanie czy rotacja, a także na różnice w intensywności. Z tego względu może być efektywna przy porównywaniu prostych obrazów oraz przy kontrolowanych warunkach ich pozyskiwania. W kontekście porównywania obrazów binarnych zawierających kontury gór założenia te wydają się spełnione, co może przekładać się na lepsze wyniki [49].

Algorytmy oparte o analizę dopasowania wzorca tworzą tzw. mapę odpowiedzi lub mapę podobieństwa. Jest to dwuwymiarowa tablica o takich samych rozmiarach jak zdjęcie wejściowe, na którym szukany jest dany szablon. Wypełniania jest obliczonymi współczynnikami podobieństwa $R(x, y)$ dla każdego położenia wzorca względem głównego obrazu. Wykonywane jest to przesuwając szablon po wszystkich pikselach obrazu wejściowego. Tak stworzona mapa zawiera w każdej komórce ustalony współczynnik dopasowania opisujący jak podobne są do siebie oba obrazy. Im wyższa jego wartość tym zdjęcie bardziej do siebie pasują. W przypadku wartości znormalizowanych pełną zgodność stwierdza się przy osiągnięciu wartości 1,0 - obrazy są wtedy identyczne przy danym położeniu. Przykładowy wynik dopasowania wycinka wygenerowanego modelu trójwymiarowego do zdjęcia rzeczywistego w projekcie pokazany został na rysunku 5.9. Bardziej intensywny biały kolor na grafice oznacza wyższe wskaźnik podobieństwa między nimi (zakres wartości podobieństwa 0,0 – 1,0 został przeskalowany na wartość pikseli 0 – 255 w skali szarości). Na środku widoczny jest najbardziej jasny punkt - jest to najbardziej prawdopodobne położenie poszukiwanego szczytu.

W bibliotece OpenCV występują trzy metody obliczania wartości w mapie dopasowania: kwadrat różnicy (ang. Squared Difference - SQDIFF), korelacja krzyżowa (ang. Cross-correlation - CCORR), współczynnik korelacji krzyżowej (ang. Cross-correlation coefficient - CCOEFF). Wszystkie z nich występują zarówno w wersji bez, jak i z normalizacją [44][45] do zakresu wartości 0,0 – 1,0.



Rysunek 5.9. Przykład mapy podobieństwa uzyskanej wykorzystując algorytm dopasowania wzorca przy porównaniu obrazów binarnych krawędzi góra.

W trakcie prac badawczych wykorzystane zostały tylko metody oparte na korelacjach (CCORR i CCOEFF). Kwadrat różnicowy został całkowicie pominięty z opisu teoretycznego oraz z porównań algorytmów z dalszej części pracy dyplomowej. Związane jest to z bardzo słabymi, bliskimi zeru, wynikami testów wstępnych. Dodatkowo, badania te pokazały przewagę wersji znormalizowanych korelacji krzyżowej i współczynnika korelacji, nad ich wersjami bez normalizacji. Związane to jest z łatwiejszym doborem parametrów przy ograniczonym zakresie wartości. Z tego powodu brane pod uwagę są tylko metody z normalizacją.

W opisanych poniżej równaniach stosowanych do wypełnienia mapy podobieństwa wykorzystywane są następujące oznaczenia:

- $R(x, y)$ - wyliczony współczynnik podobieństwa wzorca i zdjęcia wejściowego przy położeniu okna o środku x, y . Całosćiowo R oznacza tzw. mapę odpowiedzi/podobieństwa zawierającą obliczony współczynnik w każdej lokalizacji.
- x, y - współrzędne danego piksela zdjęcia wejściowego.
- x', y' - współrzędne danego piksela wzorca.
- $I(x, y)$ - wartość/intensywność/jasność piksela zdjęcia wejściowego o danych współrzędnych.
- $T(x', y')$ - wartość/intensywność/jasność piksela wzorca o danych współrzędnych.

Znormalizowana korelacja krzyżowa. Metoda ta wykorzystuje korelację krzyżową jako metrykę podobieństwa. Mierzy ona dopasowanie szablonu i zdjęcia wejściowego dla każdej możliwej pozycji drugiego obrazu. W praktyce polega ona na przesuwaniu okna analizy po całym obrazie wejściowym i dla każdej takiej lokalizacji obliczany jest iloczyn skalarny wartości pikseli. Wyższa wartość korelacji wskazuje na większe podobieństwo między szablonem, a obszarem docelowym.

Wersja ta skupia się wyłącznie na korelacji punktowej. Z tego powodu jest ona skuteczna głównie w przypadku, gdy dopasowane obrazy mają podobny zakres jasności oraz minimalne zniekształcenia względem siebie takie jak rotacja czy skalowanie.

Wartości w mapie dopasowania, dla tej metody, obliczane są zgodnie z równaniem:

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}}$$

Znormalizowany współczynnik korelacji. Metoda ta również wykorzystuje korelację krzyżową oraz iloczyn skalarny wartości pikseli. Stosuje ona dodatkowo normalizację uśredniającą jasności obrazu do zera. Z tego powodu w przeciwieństwie do poprzedniego rozwiązania nie skupia się wyłącznie na korelacji punktowej, ale również bierze pod uwagę różnice w intensywności pikseli w danym otoczeniu. Dzięki temu, wersja ta jest bardziej elastyczna i odporniejsza na zmianę warunków oświetleniowych czy kontrastu między zdjęciami oraz na różnego typu znieksztalcenia.

Miara podobieństwa opisana jest wzorem:

$$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} T'(x', y')^2 \cdot \sum_{x', y'} I'(x + x', y + y')^2}}$$

gdzie T' i I' dane są wzorami:

$$\begin{aligned} T'(x', y') &= T(x', y') - 1/(w \cdot h) \cdot \sum_{x'', y''} T(x'', y'') \\ I'(x + x', y + y') &= I(x + x', y + y') - 1/(w \cdot h) \cdot \sum_{x'', y''} I(x + x'', y + y'') \end{aligned}$$

T' i I' oznaczają wartość danego piksela, odpowiednio dla szablonu i zdjęciu wejściowego, poddaną normalizacji uśredniającej względem jasności otoczenia. Sprawia ona, że wskazania średnie w danym punkcie przyjmują wartość 0. Natomiast pozostałe wyniki równania określają odległość wartości jasności pikseli od średniej.

5.7. Dopasowanie na podstawie cech

Drugą metodą detekcji podobieństwa między obrazami wykorzystywaną w projekcie pracy dyplomowej jest dopasowanie na podstawie cech (z ang. Feature Matching). Polega ona na porównywaniu między zdjęciami pewnych, charakterystycznych cech - stąd nazwa metody. Mogą być nimi m.in.: krawędzie, tekstury, nieszablonowe zmiany jasności czy elementy wyróżniające się w danej przestrzeni. Na ich podstawie, wyszukując podobne struktury na obu zdjęciach, stwierdza się ich podobieństwo [50][51].

Metoda ta jest jedną z częściej wykorzystywanych w procesie tzw. rejestracji zdjęć (ang. Image Registration) lub inaczej nazywane wyrównaniem zdjęć (ang. Image Alignment) [52]. Są to techniki mające na celu zarejestrować i wyrównać obrazy zawierające tę samą scenę lub obiekt ale z różnych punktów widzenia czy w różnych warunkach

oświetleniowych [53], [54]. Pozwalają one uzyskać dla nich jednolity układ współrzędnych, przeprowadzić korekcję znieksztalceń, tworzyć z wielu zdjęć panoramę lub zdjęcie trójwymiarowe, a także, co istotne w kontekście pracy dyplomowej, porównywać obrazy. Z powodzeniem jednak rejestrowanie zdjęć wykorzystywane jest również w przetwarzaniu dokumentów czy śledzeniu obiektów na nagraniach [55].

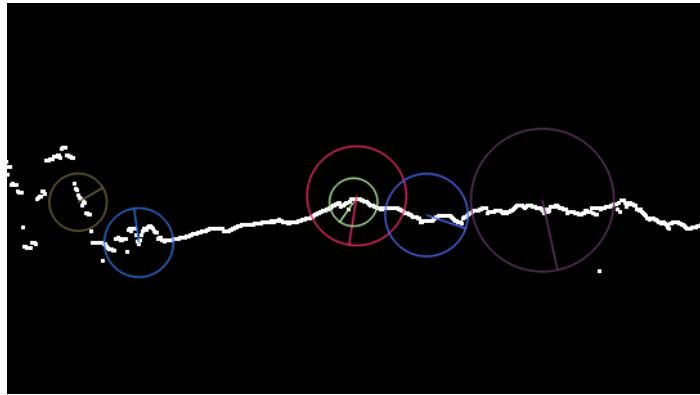
W teorii, feature matching jest bardziej elastyczny niż metoda oparta na szablonach oraz bardziej odporna na znieksztalceń, rotacje czy skalowanie. Daje ona dobre wyniki przy skomplikowanych scenach bogatych w unikalne cechy, ze zmiennymi widokami oraz przy poszukiwaniu złożonych struktur o nietrywialnych elementach. Jednocześnie radzi sobie gorzej z prostymi scenami, w szczególności podczas wyszukiwania konkretnego wzoru lub prymitywnych obiektów o ograniczonej możliwości transformacji. Z tego powodu może się okazać mało skuteczna przy porównywaniu obrazów binarnych krawędzi gór, które nie zawierają zbyt dużej liczby charakterystycznych cech, a jedynie zbiór białych linii na jednolitym tle.

Podejście to składa się z kilku etapów. Na początku wybierane są punkty charakterystyczne, zawierające pewnego rodzaju szczególne cechy. Określone są dla nich również deskryptory opisujące wybrane ich właściwości. Odbywa się to dla zdjęć wejściowych: poszukiwanego oraz przeszukiwanego. Następnie porównuje się punkty z obu obrazów, dopasowując do siebie te z największym podobieństwem. Ze względu na możliwe szумy i błędne przyporządkowania, należy je odpowiednio przefiltrować i odrzucić niepoprawne. Ostatnim krokiem jest przekształcenie szablonu na podstawie dopasowanych punktów kluczowych, w taki sposób by poprawnie określić miejsce szukanego obiektu na drugim obrazie.

5.7.1. Punkty kluczowe i ich deskryptory

Pierwszym etapem dopasowania obrazów na podstawie cech jest wyszukanie punktów kluczowych oraz ich deskryptorów [56][57]. Są one podstawą działania tego rodzaju algorytmów, ponieważ są ze sobą porównywane w celu stwierdzenia podobieństwa danych elementów zdjęć.

Punkty charakterystyczne określają specyficzne lub wyjątkowe obszary na zdjęciu, sklasyfikowane na podstawie wyróżnionych unikalnych cech. Z tego względu mogą być jednoznacznie identyfikowane w różnych widokach, a nawet na różnych zdjęciach. Ich obszary mogą być specyficzne ze względu na różne atrybuty, takie jak: wzór, tekstura, strukturę jaką tworzą lub poprzez krawędzie i kąty. Dany punkt by mógł zostać uznany za kluczowy musi spełniać określone niezmienniki, tzn. musi być niezależny od transformacji takich jak skalowanie, rotacja, zmiany jasności czy perspektywy. Kolejnym założeniem odnoszącym się do punktów kluczowych jest powtarzalność. Musi być możliwość identyfikacji tych samych cech na kolejnych scenach lub obrazach. Wizualizację wykrytych na obrazie binarnym punktów charakterystycznych wraz z ich kierunkiem oraz rozmiarem pokazano na rysunku 5.10.



Rysunek 5.10. Przykład części wykrytych punktów charakterystycznych wraz z ich kierunkiem i rozmiarem na obrazie binarnym krawędzi góra.

Cechy punktów kluczowych opisywane są przez tak zwane deskryptory tworzone dla każdego miejsca charakterystycznego. Zawierają głównie informację na temat otoczenia danego punktu. Opisują intensywność pikseli, kierunek ewentualnych gradientów, wzór, który tworzy czy jego histogram. Zestaw tych cech jest niezależny od lokalizacji, dzięki czemu pozostałe fragmenty obrazu nie mają na niego wpływu i może być porównywany między różnymi scenami. Deskryptory najczęściej przybierają formy wektorowe.

SIFT. Przykładem algorytmu pozwalającego na wyszukiwanie opisanych punktów i cech jest SIFT (z ang. Scale-Invariant Feature Transform) [58][59]. Składa się on z kilku etapów. Pierwszym jest poszukiwanie potencjalnych punktów kluczowych poprzez zastosowanie techniki *Różnicy Gaussów* (ang. Difference of Gaussian). Jest to technika nakładania filtrów gaussa o dwóch rozmiarach okna, a następnie obliczaniu różnicy wartości pikseli między nimi. Celem tego jest wyszukanie punktów niezależnych od skalowania. Wyszukując ich ekstrema uzyskuje się wstępную listę punktów charakterystycznych o różnych rozmiarach. Kolejnym etapem jest ich filtracja ze względu na kontrast (niski kontrast może oznaczać wrażliwość na szum) lub odrzucając mniej istotne krawędzie. Wykonywane jest to na podstawie progowania o pewnym ustalonym progu dla lokalnych ekstremów. Pozostałe punkty charakterystyczne traktowane są jako stabilne i wartościowe. Następnie dla każdego z nich określany jest kierunek oraz rozmiar, w zależności od jego sąsiedztwa. Pozwala to uodpornić algorytm na zmiany rotacji względem kolejnych scen. Na końcu obliczane są deskryptory cech okolicy danego punktu.

ORB. Drugą z opisywanych metod służących do określania punktów charakterystycznych oraz ich deskryptorów jest ORB (z ang. Oriented FAST and Rotated BRIEF) [60][61]. Jest to połączenie algorytmów FAST (z ang. Features from Accelerated Segment Test [62]) do wykrywania punktów kluczowych oraz BRIEF (z ang. Binary Robust Independent Elementary Features [63]) do opisu cech. Ze względu na jej hipotetyczną wydajność autor artykułu, w którym opisuje tę metodę, zatytułował go *ORB: Efektywna alternatywa dla SIFT*

oraz SURF (oryg. *ORB: An effective alternative to SURF and SIFT* [60]). Nazwa ta związana jest z mniejszą złożonością obliczeniową, a co za tym idzie krótszym czasem wykonania. Odnosi się również do patentów, którymi były objęte oba algorytmy. W przeciwieństwie do nich, ORB od razu był wolny od takich ograniczeń (SIFT był opatentowany w momencie powstania tamtego artykułu, lecz licencja ta wygasła w marcu 2020 r.). Ze względu, że FAST nie był odporny na obroty scen, w algorytmie ORB obliczana jest orientacja punktów, opisywana w formie kołowej. Modyfikuje on również metodę BRIEF wykorzystując obliczoną wcześniej rotację. Tworzone są na ich podstawie macierze obrotu, służące do nadawania deskryptorom kierunku. Ma to na celu zniwelowanie problemów algorytmu BRIEF z wydajnością. Dzieje się to jednak kosztem utraty wysokiej wariancji oraz nieskalowania punktów. By odzyskać te cechy ORB przeprowadza zachłanne przeszukiwanie potencjalnych wartości wynajdując takie, które posiadają te właściwości.

5.7.2. Dopasowanie punktów kluczowych obu zdjęć

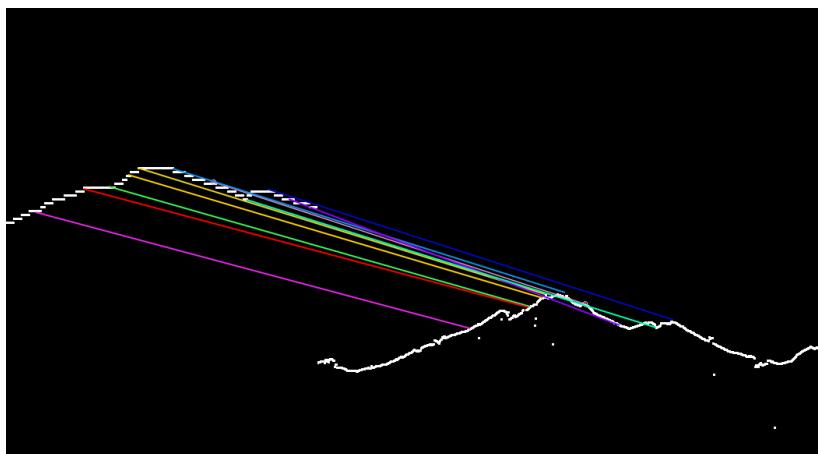
Obliczone punkty charakterystyczne oraz ich deskryptory dla obu zdjęć należy ze sobą porównać i przyporządkować do siebie najbardziej pasujące tworząc pary. Wykonywane jest to poprzez zestawienie ze sobą poszczególnych cech z obu zdjęć. Popularnymi metodami porównywania punktów kluczowych oraz deskryptorów są na przykład: algorytm siłowy (ang. Brute Matching) lub FLANN (z ang. Fast Library for Approximate Nearest Neighbors) [64][65].

Jak wszelkiego rodzaju algorytmy siłowe, tak i w tym przypadku polega on na przeszukiwaniu przestrzeni rozwiązań poprzez sprawdzenie wszystkich możliwych kombinacji i wybranie najlepszych. Dany punkt charakterystyczny z dopasowanego szablonu porównuje się z każdym punktem ze zdjęcia, na którym poszukujemy interesujący nas obiekt. Na podstawie takich porównań tworzy się pary o największym podobieństwie. Zaletą takiego rozwiązania jest gwarancja uzyskania rezultatów optymalnych z najlepszymi dopasowaniem. Wadą podejścia opartego na metodach siłowych jest natomiast długi czasu wykonania oraz wykorzystanie większych zasobów pamięciowych. Złożoność obliczeniowa tego algorytmu wynosi $O(n^2)$.

FLANN [66] [67] zorientowany jest na szybkie znajdowanie najbliższych sąsiadów. W praktyce nie jest to jedna metoda, a zbiór algorytmów służących do tego celu. W zależności od użytych parametrów takich jak sposób indeksowania, liczba epok, wymagana dokładność czy preferowana metoda możliwe jest uzyskanie różnych rezultatów. W cytowanej dokumentacji FLANN zasugerowane są optymalne ustawienia dla wybranych algorytmów wyszukujących punkty charakterystyczne, w tym dla opisanych w poprzednim podrozdziale ORB i SIFT. Zgodnie z założeniem mają być to szybkie rozwiązania, a osiągane jest to kosztem znajdowania tylko przybliżonego rozwiązania. Z tego względu może on dawać mniej dokładne wyniki niż na przykład metoda siłowa. W tym przypadku, średnia złożoność obliczeniowa jest na poziomie $O(\log(n))$, co ma znaczenie przy dużych zbiorach danych.

Przy obu podejściach, możliwe jest uzyskanie dla każdego punktu k najbliższych sąsiadów (w przypadku metody siłowej k najlepszych). Ze względu na liczbę punktów charakterystycznych oraz wartość ich dopasowania zasadnym staje się wybór tylko pewnej ich części w zależności od przyjętego kryterium. W projekcie magisterskim, jeśli dopasowywany jest tylko jeden punkt wykorzystywana jest filtracja za pomocą progowania podobieństwa. W przypadku $k \geq 2$ wykorzystywany jest tzw. test współczynników. Określa on jak bardzo lepsze musi być pierwsze z k dopasowań w porównaniu do pozostałych, by para mogła być uznana za wartościową. Jeśli przykładowo pierwszy i drugi mają porównywalną wartość podobieństwa to dopasowania dla tego punktu zostają odrzucone.

Na rysunku 5.11 przedstawiono przykład dopasowanych punktów kluczowych dla krawędzi gór wygenerowanego modelu oraz rzeczywistego zdjęcia. Do celów wizualizacyjnych zostało wybrane 10 dopasowań o najwyższym współczynniku podobieństwa, celem zwiększenia przejrzystości.



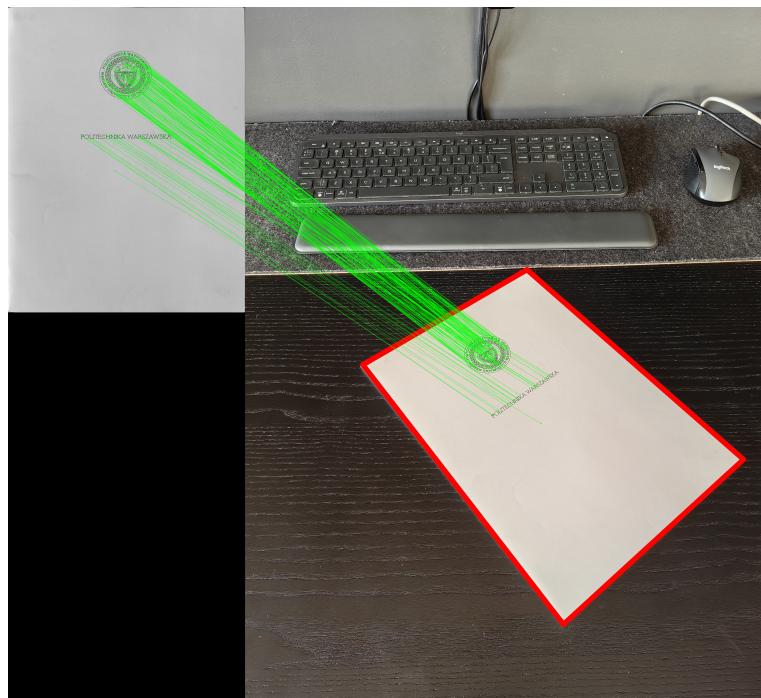
Rysunek 5.11. Przykład dopasowanych punktów kluczowych dwóch obrazów binarnych krawędzi gór.

5.7.3. Projekcja położenia z wykorzystaniem homografii

Rozłożenie połączonych punktów może tworzyć nieregularne kształty. Z tego powodu dopasowanie szablonu na zdjęciu rzeczywistym z odpowiednią skalą, rotacją i zniekształceniami jedynie na podstawie tych połączeń może być wizualnie trudnym zadaniem. Do tego celu może być wykorzystana estymacja z wykorzystaniem transformacji homograficznej [68].

Homografia opisuje relacje płaską między dwoma obrazami z różnych punktów obserwacji. Reprezentowana jest poprzez macierz 3×3 z ośmioma punktami swobody, tworzącą tak zwaną macierz transformacji perspektywicznej. Ten rodzaj przekształcenia zachowuje proste, czteropunktowe kontury obrazu, ale pozwala na deformacje obrazu bez zachowania sztywności elementów. Przy jej użyciu możliwa jest projekcja pikseli na wybraną płaszczyznę [69][70].

W kontekście porównania obrazów na podstawie cech technika ta pozwala na transformację szablonu względem dopasowanych punktów kluczowych. Wykonywane jest to w taki sposób, że możliwe jest jego nałożenie na obraz główny z jak najlepszym wyrównaniem i odwzorowaniem przekształceń. Pozwala to również na stwierdzenie dokładnego położenia szukanego fragmentu obrazu [71]. Przykład takiej transformacji z zaznaczeniem pewnej liczby punktów kluczowych oraz oznaczeniem szukanego obiektu pokazany został na rysunku 5.12 (ze względu na małą przejrzystość takiej grafiki przy dopasowaniu krawędzi gór, do celów prezentacyjnych wykorzystany został prosty obiekt świata rzeczywistego).



Rysunek 5.12. Przykład transformacji homograficznej na podstawie dopasowanych punktów kluczowych celem oznaczenia położenia szukanego szablonu.

W części badawczej pracy dyplomowej do znalezienia optymalnej macierzy homografii wykorzystywane są dwie metody: RANSAC i PROSAC. Mają one za zadanie przeszukiwanie przestrzeni danych oraz obliczenie wartości tej macierzy.

RANSAC. Pierwszą z opisywanych metod wykorzystywanych do obliczenia macierzy homografii na podstawie punktów kluczowych jest RANSAC (z ang. Random Sample Consensus) [72]. Jest to iteracyjny algorytm służący do estymowania wybranych parametrów na podstawie zestawu danych wejściowych. W przypadku feature matching jest to wyliczenie wartości macierzy transformacji homograficznej, gdy na wejściu zostaje przesłany zestaw punktów kluczowych. Jego celem jest znalezienie jak najlepszego modelu pasującego do jak największej części zbioru.

5. Przegląd algorytmów i szczegóły implementacyjne

Algorytm ten według literatury jest skuteczny przy pracy z zaszumionymi danymi oraz ze zbiorami zawierającymi wiele wartości odstających. Wynika to z faktu, że skupia się on na znalezieniu modelu, który pasuje do jak największego zbioru będącego w pewnej odległości od siebie, ignorując przy tym wartości znacznie się różniące i odstające.

W trakcie danej iteracji metody RANSAC wykonywane są następujące kroki:

1. Losowy wybór minimalnego zbioru danych potrzebnego do estymowania parametrów. W przypadku dopasowania obrazów zawsze są to co najmniej cztery punkty kluczowe.
2. Na podstawie wybranego podzbioru obliczane są parametry modelu. W opisywanym kontekście są to wartości macierzy homografii.
3. Stwierdzenie jaka część danych wykorzystując stworzony model mieści się w granicach ustalonej tolerancji.
4. Na podstawie danych będących w ustalanych granicach obliczana jest jakość wygenerowanego modelu.

Kroki te powtarza się przez określoną liczbę iteracji lub do znalezienia modelu o wystarczająco wysokim wskaźniku jakości. W pierwszym przypadku wybierany jest ten model ze wszystkich wygenerowanych, dla którego największy podziór danych spełnia założone dopasowanie.

PROSAC. Kolejnym prezentowanym rozwiążaniem jest PROSAC (ang. Progressive Sample Consensus) [73]. Jest to rozwinięcie algorytmu RANSAC poprzez zmianę sposobu wyboru podzbioru danych. W pierwszym z opisywanych podejść jest to wykonywane przez losowy wybór próbek. Natomiast, w przypadku algorytmu PROSAC próbowanie to odbywa się metodą progresywną na podstawie wyników z poprzednich iteracji. Biorąc pod uwagę oceny ostatnich modeli, prioryzuje on te dane, które są bardziej prawdopodobne. Skupienie się na prawidłowych próbkach ma na celu zwiększenie szansy na szybsze znalezienie dobrego i wystarczającego modelu. Pozostałe kroki są takie same jak w RANSAC. Rozwiążanie to jest częściej wykorzystywane, gdy istnieje duża liczba wartości odstających w zbiorze danych oraz potrzebne jest szybsze dopasowanie modelu.

6. Dodatkowe problemy implementacyjne

6.1. Niedokładność sensorów urządzeń

Jednym z elementów procesu identyfikacji szczytów górskich na obrazie jest generowanie trójwymiarowego modelu terenu. W stworzonej platformie testowej opiera się on na interfejsie OpenGL i danych NTM. Moduł ten umożliwia kalibrację kamery zgodnie z ustawieniami danego urządzenia. Dzięki temu, generowany model może być łatwo porównywany z rzeczywistymi, statycznymi zdjęciami. Możliwe jest również ustalenie obserwatora w dowolnym miejscu, np. zgodnie z danymi dotyczącymi zarejestrowanego wcześniej obrazu z urządzenia mobilnego oraz obserwacja terenu z różnych perspektyw.

Podczas testów wykorzystujących przygotowany zbiór statycznych zdjęćauważono, że generowany model na podstawie zapisanych danych, w szczególności kątów obrotu, był w niektórych przypadkach przesunięty o pewną odległość względem zdjęcia rzeczywistego. Wynikało to przeważnie ze wskazań czujnika obrotów. Przesunięcia odpowiadały najczęściej błędem o wielkości $2^{\circ} - 5^{\circ}$. Dzięki temu, w dalszej pozostałej części testowej pracy dyplomowej założono pewną tolerancję odpowiadającą takim niedokładnościom uzyskiwanych danych.

6.2. Skalowanie trójwymiarowego modelu terenu

Testowanie wizualizacji trójwymiarowej wykazało również błędne założenia na temat skalowania terenu. Wstępnie wykorzystywane były ogólnie przyjęte przybliżenia jednej jednostki szerokości oraz długości geograficznej. Wygenerowana w ten sposób panorama terenów Tatr była rozciągnięta w poziomie względem tego co widoczne na odpowiadających zdjęciach.

Na podstawie analizy tego problemu stwierdzono, że problem wynika ze zbyt dużej powierzchni generowanej na podstawie jednej kratki danych względem terenu rzeczywistego. Związane to jest z kształtem Ziemi, a w szczególności z rozłożenia siatki geograficznej. Wraz ze zbliżaniem się do jednego z biegunów maleje odległość między kolejnymi wartościami długości geograficznej. Przy równiku odległość ta wynosi około 111 km, natomiast przy biegunie już tylko 2 km. W Polsce, przykładowo w okolicach Tatr długość ta ma wartość około 73 km, a nad morzem 65 km. Natomiast liczba danych w jednym kwadracie SRTM się nie zmienia. Z tego powodu skalowanie (odległość między kolejnymi wartościami) powinno być różne w zależności od położenia geograficznego miejsca, dla którego generowany jest teren.

W konsekwencji wykorzystane zostały metody obliczania odległości między dwoma punktami geograficznymi, celem estymowania przybliżonej skali terenu. Pozwoliło to na wyeliminowanie efektu rozciągnięcia panoramy, dzięki czemu była ona bardziej zbliżona do rzeczywistego widoku.

6.3. Migotanie wskazań lokalizacji widocznych szczytów

W trakcie weryfikacji końcowej przetestowane zostało działanie całego procesu rozpoznawania szczytów górskich. Odbyło się to w okolicach Zakopanego oraz Bukowiny Tatrzańskiej. Testy polegały na sprawdzeniu czy potencjalne oprogramowanie wykorzystujące wyniki niniejszej pracy dyplomowej jest w stanie rozpoznawać widoczne oraz odrzucać zasłonięte szczyty na obrazie pobieranym na żywo z kamery, a także jej zdolności do działania w czasie rzeczywistym.

Badania wykazały wrażliwość projektu na różnego typu drgania czy wpływ wiatru, a także zmiany w odbieranej jasności w danych punktach. Skutkowało to wahaniem wykrywanych krawędzi góra. Objawiało się to niekontrolowanym przesuwaniem się oznaczeń na zdjęciu. Było to pewnego rodzaju migotanie. Spowodowało to konieczność zaimplementowania mechanizmu tłumienia tych drgań, ograniczającego ruchy etykiet na obrazie.

Mechanizm ten został opracowany z wykorzystaniem okna analizy pewnej liczby klatek wstecz. Zapamiętywał on ostatnie obliczone położenia danego szczytu i na ich podstawie określał najbardziej prawdopodobną lokalizację góry. Jednocześnie odrzucał wartości brzegowe mogące być wynikiem błędów algorytmów, chwilowych zmian widoczności, różnego typu szumów czy niedoskonałościom odbieranego obrazu. Skutkiem wymienionych zmian może być mniej dokładne etykietowanie w danym momencie ze względu na uśrednianie położenia, lecz dzięki nim zmniejszono wpływ migotania będącego nieprzyjemnym efektem dla użytkownika i utrudniającym mu odbiór wyników aplikacji.

7. Porównanie wybranych algorytmów i próby optymalizacji

W ramach pracy dyplomowej podjęte zostały próby usprawnienia i optymalizacji różnych aspektów projektu. Testowany był wpływ rozdzielczości danych SRTM terenu na dokładność odwzorowania modelu oraz czas ich przetwarzania. Podjęta została próba optymalizacji procesu poprzez odrzucenie niepotrzebnych w danym momencie próbek terenu, zmniejszając przetwarzaną ich liczbę. Z powodu mnogości dostępnych metod obliczania odległości między punktami geograficznym, zostały one przebadane ze względu na dokładność, a także na czas wykonania potrzebnych obliczeń. Skuteczność dopasowania obrazów binarnych zawierających kontury gór na modelu terenu oraz na zdjęciu rzeczywistym wydaje się kluczowym czynnikiem wpływającym na jakość rozwiązania. Mając to na uwadze zostały przetestowane i porównane metody dopasowania na podstawie szablonu oraz na podstawie cech. Przeprowadzono również próbę optymalizacji tych klasyfikatorów pod względem jakościowym poprzez zastosowanie operacji morfologicznej dylacji oraz zbadano jej wpływ na ten aspekt. Na koniec przeprowadzono profilowanie procesu rozpoznawania szczytów wykonywanego na platformie testowej, które pozwoliło wyłapać ewentualne błędy implementacyjne oraz tzw. wąskie gardła zaproponowanego procesu. Dzięki temu wykryta została możliwość usprawnienia systemu poprzez lepsze wykorzystanie pamięci podręcznej procesora.

W przypadku testów i porównań złożoności czasowej rozwiązań badania były przeprowadzane przy pomocy trzech różnych urządzeń z systemem Android. Były one zróżnicowane pod względem wydajnościowym, jak i lat produkcji. Wykorzystane były urządzenia marki Huawei: Mate 20 Lite, P30 Pro oraz Mate 50 Pro. Przedstawione zostały w kolejności od najmniejszej mocy obliczeniowej do największej.

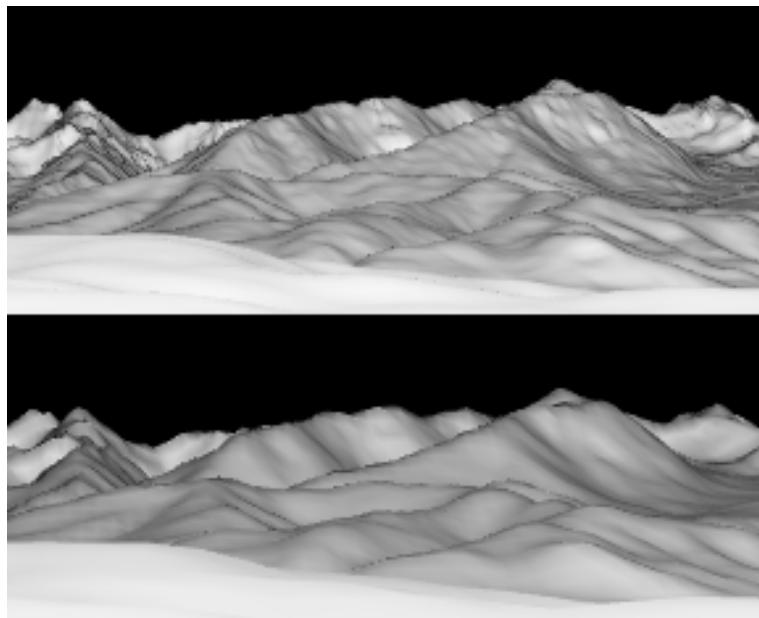
7.1. Wpływ rozdzielczości danych SRTM na dokładność odwzorowania terenu

Opisane wcześniej dane SRTM w wersji podstawowej 1'' zawierają 12 967 201 wartości na jedną kratkę (3601x3601) siatki geograficznej. Natomiast, w przypadku rozdzielczości 3'' próbka jest już tylko 1 442 401 (1201x1201). Jest to prawie 90% mniej wpisów.

Mimo, że rzadsze próbkowanie zawiera dziewięciokrotnie mniej danych, to jednak dla problemu detekcji szczytów górskich, dokładne odwzorowanie terenu nie zawsze musi być aż tak znaczące. Na rzeczywistych zdjęciach pasma górskie mogą być oddalone nawet o kilkanaście czy kilkadziesiąt kilometrów, więc ich szczegółowość nie będzie duża. Tak samo, trójwymiarowy model terenu w zależności od rozdzielczości danych, będzie różnić się tylko detalami. Odwzorowanie pasm górskich zostanie mimo to zachowane.

Ze względu na dużą różnicę w liczbie danych, porównane zostały modele generowane z wykorzystaniem obu rozdzielczości. Badanie polegało na subiektywnej ocenie dokładności krawędzi pasm renderowanego terenu w wybranych lokalizacjach geograficznych.

Przykładowe porównanie modeli pokazane zostało na rysunku 7.1. Przedstawia on wycinki wyrenderowanych górz oddalonych w rzeczywistości o 10 – 20 km.



Rysunek 7.1. Porównanie generowanego terenu w zależności od rozdzielczości danych. Górnny wyrenderowany model prezentuje dokładność 30 m, dolny natomiast 90 m.

Na podstawie pokazanego porównania można stwierdzić, że główne kontury pasm górskich są widoczne w podobnym i dostatecznym stopniu na obu wizualizacjach. Bardziej dokładny model posiada więcej szczegółów związanych z nierównościami terenu, drobnymi pagórkami czy gwałtownymi zmianami wysokości. Na potrzeby detekcji krawędzi gór obie wizualizacje są wystarczające, ze względu na to, że pasma horyzontalne są wyraźnie widoczne. Dodatkowo, bardziej dokładne dane mogą mieć negatywny wpływ na rozpoznawanie tych elementów poprzez zwiększenie zaszumienia wizualizacji mało istotnymi konturami i zmianami odcieni terenu.

Przeprowadzone zostało badanie wpływu rozdzielczości zbioru danych na czas generowanie trójkątów i wierzchołków modelu. Wyniki tego testu zostały przedstawione w tabeli 7.1. Długość trwania tych operacji w przypadku mniej dokładnego zestawu danych wyniosła około 10 – 12% czasu potrzebnego na te same czynności wykonane przy wykorzystaniu danych o większej rozdzielczości. Dane te potwierdzają opisane rozważania teoretyczne. Każdy przypadek testowy przeprowadzony był dziesięciokrotnie, a wynik uśredniony. Rezultaty jednoznacznie pokazują znaczne zmniejszenie okresu potrzebnego na przetworzenie wejściowych danych dotyczących terenu, co może mieć istotne znaczenie przy komforcie użytkowania takiego projektu. Dane wczytywane i przygotowywane są w czasie ładowania aplikacji, a nie jej działania, dlatego wpływ może mieć głównie na subiektywne odczucia podczas jej używania, a nie na wydajności w kontekście pracy w czasie rzeczywistym.

Tabela 7.1. Porównanie czasu generowania wierzchołków i trójkątów w zależności od rozdzielczości danych SRTM.

Urządzenie	Czas generowania wierzchołków			Czas generowania trójkątów		
	1"	3"	$\frac{t_1''}{t_3''}$	1"	3"	$\frac{t_1''}{t_3''}$
Mate 20 Lite	0,3576 s	0,0516 s	0,1444	0,3330 s	0,0364 s	0,1095
P30 Pro	0,1794 s	0,0209 s	0,1170	0,2256 s	0,0231 s	0,1026
Mate 50 Pro	0,1197 s	0,0135 s	0,1129	0,1526 s	0,0147 s	0,0969

Więcej danych oznacza również zwiększenie zapotrzebowania na zasoby pamięci celem przechowania znacznie większej liczby wierzchołków i trójkątów. Potrzebny rozmiar pamięci przedstawiono w tabeli 7.2. W projekcie wczytanych kratek danych może być nawet 4 w zależności od położenia geograficznego i maksymalnej renderowanej odległości. Wtedy różnice czasowe są jeszcze bardziej znaczące. W przypadku mniejszej rozdzielczości zostaje zaoszczędzone wtedy aż 1659 MB pamięci operacyjnej urządzenia.

Tabela 7.2. Porównanie pamięci zajmowanej przez wierzchołki i trójkąty w zależności od rozdzielczości danych SRTM.

Rozdzielczość	Pamięć zaalokowana dla wierzchołków	Pamięć zaalokowana dla trójkątów
1"	155,6 MB	311,0 MB
3"	17,3 MB	34,5 MB

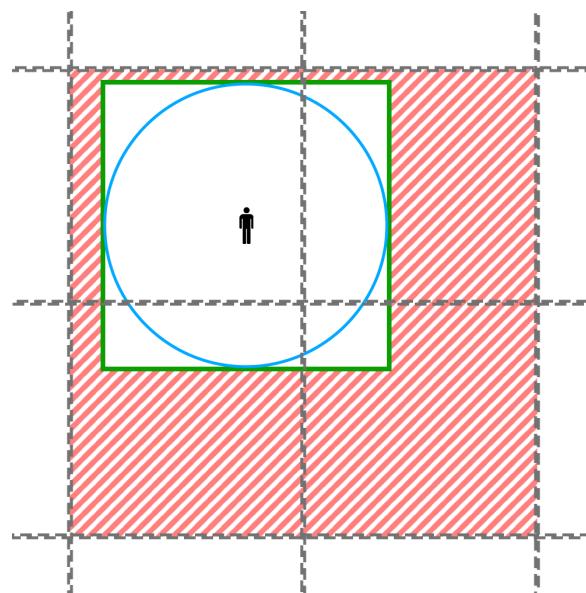
Porównanie graficzne wygenerowanych modeli pokazało małą stratę jakości w kontekście odwzorowania istotnych elementów terenu. Natomiast, różnica czasu tworzenia wierzchołków i trójkątów oraz zajmowanej pamięci przy mniejszej rozdzielczości jest znaczna. Wydaje się zatem, że wykorzystanie dokładniejszych danych jest niepotrzebne. Z tego powodu na platformie testowej używane były dane SRTM o rozdzielczości 3".

7.2. Odrzucenie niepotrzebnych danych SRTM

Nie wszystkie wczytane dane SRTM są potrzebne ze względu na parametryzowaną odległość na jaką renderowany jest dany teren i na jakiej rozpoznawane są szczyty górskie. Z tego powodu można usunąć niepotrzebne wpisy.

W zależności od położenia i ustawionej odległości widzenia wczytywane są od jednego do czterech plików SRTM, które następnie łączone są w jedną tablicę dwuwymiarową. W przypadku czterech kratek, mogą one obejmować prawie 50 tysięcy kilometrów kwadratowych. Natomiast, jeśli założymy, że maksymalny widoczny dystans wynosi 30 km, to koło o takim promieniu zajmuje powierzchnie raptem ~ 3 tysiące kilometrów kwadratowych. W tym wypadku, efektywne wykorzystanie przechowywanych danych wynosi około 6%.

Dlatego, zdecydowano się odrzucać część wczytanych danych, które reprezentują punkty geograficzne znajdujące się poza określonym polem widzenia. Biorąc pod uwagę, że użytkownik urządzenia mobilnego może się z nim przemieszczać, promień takiego pola jest powiększony o jeden kilometr względem ustawionego parametru maksymalnej odległości renderingu. Przechowywanie danych dla pola o kształcie koła wiązałoby się z dodatkowymi obliczeniami, np. wyliczanie przesunięcia, a także z bardziej skomplikowanym procesem odrzucenia niepotrzebnych próbek. Z tego powodu, brane pod uwagę są wszystkie wpisy zawierające się w kwadracie opisany na takim kole. Mimo, że punkty znajdujące się w rogach tego czworokąta są nadmiarowe to pozwalają na uproszczenie przeprowadzanych obliczeń oraz przechowywanie danych w formie wypełnionej tablicy dwuwymiarowej. Wizualizacja odrzucenia niepotrzebnych danych pokazana jest na rysunku 7.2. Widoczny na nim zacieniony, czerwony obszar zostaje usunięty.



Rysunek 7.2. Wizualizacja usuwania niepotrzebnych danych SRTM.

Dzięki takiemu założeniu, ilość przetwarzanych danych została ograniczona nawet o 92,7%, jeśli maksymalna widoczność wynosi 30km oraz brane są pod uwagę cztery sąsiadujące kratki. W przypadku gdy obserwator znajduje się w okolicach Tatr, to ze względu na zmniejszenie odległości między kolejnymi punktami długości geograficznej, różnica ta wynosi około 88,9%. Natomiast, jeśli maksymalny dystans jest ustawiony na 50km, wartości te wynoszą odpowiednio 89,7% i 79,2%.

Dane teoretyczne oraz wyniki pokazane w tabeli 7.3, jednoznacznie potwierdzają zasadność takiego uproszczenia. Sumaryczny czas potrzebny na wygenerowanie modelu przy odrzuceniu części danych jest mniejszy o około 80 – 85% w porównaniu do cyklu bez takiej optymalizacji.

Tabela 7.3. Porównanie czasu generowania wierzchołków i trójkątów przy uproszczeniu i braku uproszczenia danych.

Urządzenie	Czas upraszczania danych	Czas generowania wierzchołków	Czas generowania trójkątów	Sumaryczny czas
Bez upraszczania danych (t_1)				
Mate 20 Lite	nd.	0,1730 s	0,1255 s	0,2985 s
P30 Pro	nd.	0,0980 s	0,1170 s	0,0937 s
Mate 50 Pro	nd.	0,0549 s	0,0586	0,1136 s
Z upraszczaniem danych (t_2)				
Mate 20 Lite	0,0033 s	0,0387 s	0,0227 s	0,0648 s
P30 Pro	0,0016 s	0,0142 s	0,0154 s	0,0314 s
Mate 50 Pro	0,0007 s	0,0081 s	0,0079 s	0,0168 s
Stosunek czasów $\frac{t_2}{t_1}$				
Mate 20 Lite	nd.	0,2241	0,1811	0,2171
P30 Pro	nd.	0,1458	0,1651	0,1640
Mate 50 Pro	nd.	0,1478	0,1362	0,1458

7.3. Porównanie algorytmów obliczających odległość między dwoma punktami geograficznymi

Ze względu na różne rozwiązania umożliwiające obliczenie odległości między dwoma punktami, zostały one przetestowane pod kątem dokładności wykonywanych obliczeń oraz szybkości działania.

Jako referencja wyników odległości wykorzystany został algorytm Vincenta, uznawany za najbardziej dokładny wśród prezentowanych. Ustawiony został dla niego limit iteracji na poziomie 1000 (w nagłówkach tabel, liczba w nawiasie przy metodzie Vincenta oznacza maksymalną liczbę iteracji) i względem niego obliczane były różnice i błędy pozostałych metod.

Niezależnie od odległości wszystkie algorytmy uzyskały akceptowalną niedokładność wyników. Przedstawione zostały one w tabeli 7.4. Największa różnica wyniosła około 0,32% dla przypadku Londyn - Warszawa, którego rzeczywista odległość wynosi ~ 1452 km. Wartość ta odpowiada w przybliżeniu błędowi 200 m. Na potrzeby obliczeń wykonywanych w ramach projektu dyplomowego, błąd przy tak odległych punktach jest całkowicie pomijalny i nie wpływa w istotny sposób na działanie programu i dokładność rezultatów całego procesu. Dodatkowo, obliczenia na tak dużych odległościach w projekcie są mało prawdopodobne, ponieważ widoczność tak oddalonych szczytów jest niemożliwa i w rzeczywistości nie występuje. Największe dystanse w aplikacji testowej obliczane są między dwoma bokami siatki geograficznej, ale nawet wtedy ich maksymalny wynik nie przekracza 220 km. Jest to mniej niż badana wartość dla przypadku Warszawa - Kraków, dla którego błąd wynosi około 0,07%.

7. Porównanie wybranych algorytmów i próby optymalizacji

Tabela 7.4. Porównanie dokładności wybranych algorytmów obliczających odległość między dwoma punktami geograficznymi.

Punkty geograficzne	Vincent (1000)	Vincent (100)	Haversian	Spherical Law of Cosines	Equirectangular Approximation
(49.0; 20.0) ->(50.0; 20.0)	111,21941 km $\Delta = 0,00000 \text{ km}$ (0,00000%)	111,21941 km $\Delta = 0,02448 \text{ km}$ (0,02201%)	111,19493 km $\Delta = 0,02444 \text{ km}$ (0,02197%)	111,19497 km $\Delta = 0,02444 \text{ km}$ (0,02197%)	111,19493 km $\Delta = 0,02448 \text{ km}$ (0,02201%)
(49.0; 20.0) ->(49.0; 21.0)	73,17126 km $\Delta = 0,00000 \text{ km}$ (0,00000%)	73,17126 km $\Delta = 0,22135 \text{ km}$ (0,30251%)	72,94991 km $\Delta = 0,22135 \text{ km}$ (0,30251%)	72,94991 km $\Delta = 0,22135 \text{ km}$ (0,30251%)	72,95044 km $\Delta = 0,22082 \text{ km}$ (0,30179%)
(51.509865; -0.118092) ->(52.237049; 21.017532) Londyn ->Warszawa	1452,71785 km $\Delta = 0,00681 \text{ km}$ (0,00047%)	1452,71104 km $\Delta = 0,00681 \text{ km}$ (0,00047%)	1448,08846 km $\Delta = 4,62939 \text{ km}$ (0,31867%)	1448,08846 km $\Delta = 4,62939 \text{ km}$ (0,31867%)	1453,24964 km $\Delta = 0,53179 \text{ km}$ (0,03661%)
(52.237049; 21.017532) ->(50.049683; 19.944544) Warszawa ->Kraków	254,66051 km $\Delta = 0,00006 \text{ km}$ (0,00002%)	254,66057 km $\Delta = 0,00006 \text{ km}$ (0,00002%)	254,74374 km $\Delta = 0,18677 \text{ km}$ (0,07334%)	254,47374 km $\Delta = 0,18677 \text{ km}$ (0,07334%)	254,48145 km $\Delta = 0,17906 \text{ km}$ (0,07031%)
(50.049683; 19.944544) ->(49.299030; 19.949047) Kraków ->Zakopane	83,49035 km $\Delta = 0,00000 \text{ km}$ (0,00000%)	83,49035 km $\Delta = 0,00000 \text{ km}$ (0,00000%)	83,46943 km $\Delta = 0,02092 \text{ km}$ (0,02506%)	83,46943 km $\Delta = 0,02092 \text{ km}$ (0,02506%)	83,46943 km $\Delta = 0,02092 \text{ km}$ (0,02506%)
(49.299030; 19.949047) ->(49.250332; 19.933662) Zakopane ->Giewont	5,53040 km $\Delta = 0,00000 \text{ km}$ (0,00000%)	5,53040 km $\Delta = 0,00000 \text{ km}$ (0,00000%)	5,52875 km $\Delta = 0,00165 \text{ km}$ (0,02984%)	5,52875 km $\Delta = 0,00165 \text{ km}$ (0,02984%)	5,52875 km $\Delta = 0,00165 \text{ km}$ (0,02984%)

W przypadku obliczania odległości między dwoma równoleżnikami błąd porównywanych algorytmów wynosił około 0,022%. Natomiast niedokładność wyników między południkami był na poziomie 0,30%. Te dwa testy ze względu na swoje znaczenie przy wyznaczaniu skali generowanego modelu trójwymiarowego, która ma realny wpływ na dokładność rozpoznawania szczytów, były bardzo istotne. Pomijając algorytm Vincenta, pozostałe metody dały porównywalne wyniki. Z tego powodu ich wpływ na wybór ostatecznej metody był zerowy. Tym bardziej, że przekłamanie w obliczonym dystansie wynosiło odpowiednio tylko 20 m i 200 m.

Algorytm Vincenta, niezależnie od wartości parametru maksymalnej liczby iteracji daje praktycznie takie same rezultaty. Może to wynikać z charakterystyki tej metody oraz wykorzystanych danych wejściowych w trakcie testu. Prawdopodobnie w badanych przypadkach algorytm przestawał obliczać kolejne przybliżenia wartości przed osiągnięciem zadanej, maksymalnej liczby iteracji. Mogło to wynikać z faktu osiągnięcia błędu mniejszego niż $\epsilon = 1e-12$.

Niedokładność podczas obliczania odległości między Krakowem, a Zakopanem wyniosła w przybliżeniu 20 m. Jest to wartość mniejsza niż odległość między kolejnymi punktami danych SRTM1, których rozdzielcość wynosi około 30 m. Błąd na tym poziomie uzyskały wszystkie testowane metody. Z tego powodu ostateczny wybór algorytmu jako optymalny z zaprezentowanych odbywał się z uwzględnieniem przede wszystkim złożoności czasowej poszczególnych rozwiązań.

Takie porównanie czasu trwania obliczeń dla poszczególnych rozwiązań przedstawiono w tabeli 7.5. Zgodnie z założeniami, w porównaniu do metody Vincenta trzy pozostałe algorytmy okazały się dużo szybsze.

Tabela 7.5. Porównanie czasu wykonania wybranych algorytmów obliczających odległość między dwoma punktami geograficznymi.

Urządzenie		Vincent (1000)	Vincent (100)	Haversian	Spherical Law of Cosines	Equirectangular Approximation
Mate 20 Lite	Średni czas jednego testu	10,97529 s	11,52881 s	2,25107 s	2,06024 s	0,59029 s
	Średni czas wywołania jednej funkcji	1097,52893 ns	1152,88094 ns	225,10736 ns	206,02364 ns	59,02929 ns
	Różnica czasowa jednego wywołania funkcji	nd.	-55,35201 ns (-4,80119%)	-927,77358 ns (-80,47436%)	-946,85730 ns (-82,12967%)	-1093,85165 ns (-94,87982%)
P30 Pro	Średni czas jednego testu	4,84974 s	4,83854 s	0,89253 s	0,91203 s	0,35220 s
	Średni czas wywołania jednej funkcji	484,97448 ns	483,85403 ns	89,25293 ns	91,20310 ns	35,22035 ns
	Różnica czasowa jednego wywołania funkcji	nd.	+1,12045 ns (+0,23157 %)	-394,60110 ns (-81,55375%)	-392,65093 ns (-81,15070%)	-448,63368 ns (-92,72087%)
Mate 50 Pro	Średni czas jednego testu	3,98607 s	3,92896 s	0,66648 s	0,60028 s	0,15715 s
	Średni czas wywołania jednej funkcji	398,60748 ns	392,8963 ns	66,64805 ns	60,02849 ns	15,71475 ns
	Różnica czasowa jednego wywołania funkcji	nd.	+5,71118 ns (+1,45361%)	-326,24825 ns (-83,03673%)	-332,86781 ns (-84,72154%)	-377,18155 ns (-96,00028%)

Przy badaniu złożoności obliczeniowej algorytmu Vincenta dla obu wartości parametrów czas wykonania funkcji i testu był podobny. Wynik ten wydaje się potwierdzać tezę, że maksymalna liczba iteracji nie jest osiągana, a proces kończy działanie wcześniejszej.

Metody Haverisan oraz Sferyczne Prawo Consinusów opierają się na bardzo podobnej metodzie i obliczeniach, dlatego czas ich wykonania jest bardzo zbliżony. Obliczenie jednego wywołania tych dwóch funkcji zajmuje jedynie około 20% czasu potrzebnego na wykonanie algorytmu Vincenta. W przypadku najsłabszego z wykorzystanych urządzeń jest to różnica prawie tysiąc nanosekund przy każdej obliczanej odległości.

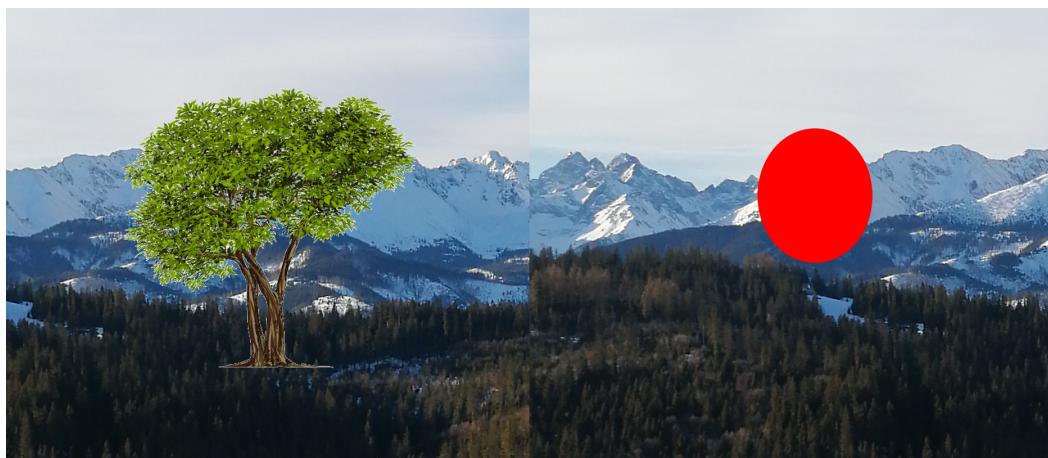
Najszybszym z testowanych sposobów obliczenia odległości między dwoma punktami geograficznymi jest przybliżenie Equirectangular. Nawet względem najbardziej popularnej formuły Haversiana wykonywany jest 3 – 4 razy szybciej. Jest to zgodne z przedstawionym teoretycznym opisem teoretycznym tej metody, gdzie stwierdzono jej popularność przy projektach o ograniczonych zasobach sprzętowych ze względu na jej małą złożoność obliczeniową.

Na podstawie przyprowadzonych testów dokładności i szybkości działania na urządzeniach testowych wybrany jako najbardziej optymalny został algorytm Equirectangular Approximation. Uzyskał on podobny błąd odległości jak pozostałe metody, ale jego złożoność czasowa była znacznie mniejsza. Szybkość działania była czynnikiem decydującym ze względu na złożoność czasową, co w przypadku urządzeń mobilnych, których zasoby są ograniczone może mieć bardzo istotne znaczenie.

7.4. Testowanie i optymalizacja algorytmów dopasowania obrazów

Dopasowanie obrazów celem stwierdzenia widoczności wybranych gór oraz określenia ich położenia na zdjęciu wydaje się zagadnieniem kluczowym w kontekście prawidłowego rozpoznawania szczytów. Z tego powodu przeprowadzone zostały testy porównujące algorytmy oparte na szablonach oraz na cechach. Przebadano została także ich skuteczność w zależności od ustawień wybranych parametrów. Odbywało się to poprzez przeszukanie przestrzeni możliwych ich wartości celem znalezienia najbardziej optymalnych zestawień. Przeprowadzono również badanie sprawdzające jaki wpływ ma operacja morfologiczna dylacji na jakość klasyfikatorów przy zastosowaniu jej na obrazach binarnych. Wykonane badania pozwoliły na wybranie lepszej metody, dobranie parametrów optymalnych oraz stwierdzenie czy wykorzystanie dylacji jest uzasadnione.

Opisane w poniższych podrozdziałach testy zostały przeprowadzone z wykorzystaniem dwóch zdjęć ze zbioru przygotowanego do statycznej analizy podczas jednego z etapów pracy dyplomowej. Na obu z nich widoczne są inne szczyty, jednak wszystkie należą do pasma górskiego Tatr. Na potrzeby testów, badane zdjęcia zostały kilkukrotnie powielone i zmodyfikowane poprzez dodanie na kolejnych kopii różnych typu artefaktów. Zmiany te miały symulować niewidoczność szczytów górskich, które mogą wystąpić z różnych względów w rzeczywistości. Takimi powodami mogą być na przykład przesłanianie innymi obiekttami, drzewami czy budynkami. Dwie z takich takich modyfikacji pokazane zostały na rysunku 7.3. Każde zdjęcie występuje w sumie w sześciu wariantach, na których zasłaniane są pojedyncze szczyty, kilka na raz lub całe pasma, większymi lub mniejszymi elementami.



Rysunek 7.3. Przykłady nałożenia artefaktów zasłaniających część szczytów w wybranych przypadkach testowych.

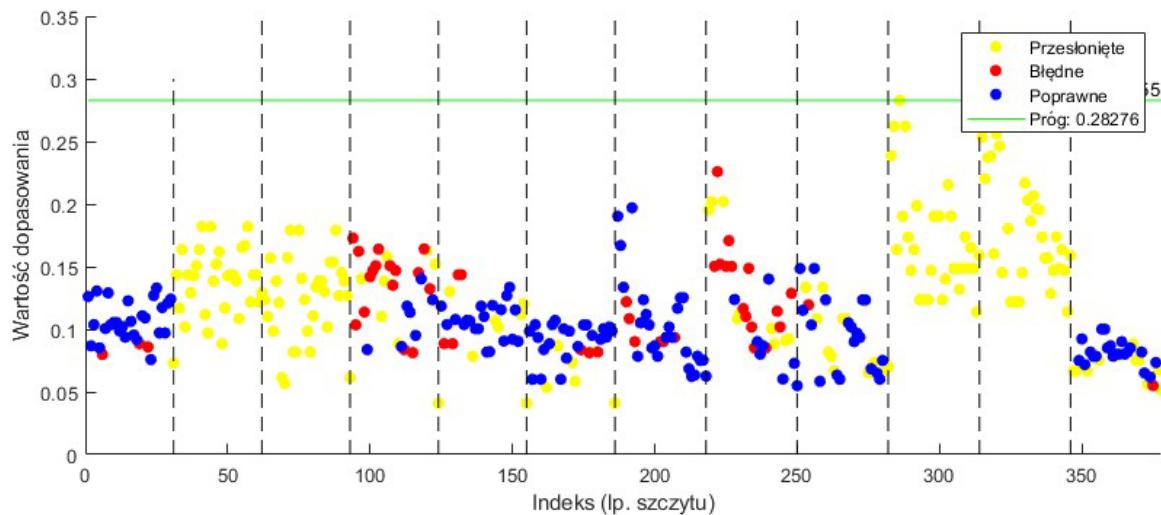
Sumarycznie daje to 378 badanych szczytów w pojedynczym teście (każda kopia jednego zdjęcia - 32 szczyty, natomiast drugiego - 31), w tym 199 widocznych oraz 179 zasłoniętych. Daje to rozkład klas na poziomie 53%/47%.

Badania były przeprowadzane przy dwóch rozdzielczościach: 768x1024 i 3840x5120. Większa z nich związana jest z wielkością zebranych wcześniej statycznych zdjęć. Model trójwymiarowy terenu był domyślnie generowany w rozmiarach 768x1024, dlatego w przypadku drugiej rozdzielczości zdjęć, uzyskany render interpolowany był w góre. Operacja ta była przeprowadzana dopiero na wynikowym obrazie binarnym procesu detekcji krawędzi.

W poniższych zestawieniach i porównaniach przedstawione zostały wyniki jedynie dla mniejszego rozmiaru zdjęć, ponieważ w procesie rozpoznawania szczytów w czasie rzeczywistym wykorzystywane są obrazy o mniejszych rozdzielczościach ze względu na złożoność czasową obliczeń. Dlatego rozmiar 768x1024 jest bliższy rzeczywistemu. Dodatkowo, przestrzeń w pracy dyplomowej zajmowana przez tak dużą ilość danych rozrosłaby się niepotrzebnie. Porównanie dwóch rozdzielczości pozwoliło jednak stwierdzić brak znaczących różnic w wynikach między nimi. Z tego względu można założyć, że wpływ rozdzielczości zdjęć na skuteczność dopasowania obrazów do siebie nie jest tak znaczny. Wszelkie dane zebrane dla obu rozdzielczości zostały umieszczone w repozytorium opisany w załączniku nr 1.

W zależności od rozmiaru zdjęcia badane były różne rozmiary wycinka obrazów. W przypadku rozdzielczości 768x1024 były to fragmenty o wymiarach: 20x20, 40x40, 60x60, 80x80, 120x120, 160x160, 200x200, 300x300. Natomiast, dla 3840x5120: 100x100, 200x200, 300x300, 400x400, 600x600, 800x800, 1000x1000. Wycinany segment zdjęcia rzeczywistego był powiększony o 15% rozmiaru zdjęcia. Związane jest to z założoną niedokładnością czujników dotyczących obrotu urządzenia. Założona tolerancja odpowiada około $+/- 5^\circ$ odchylenia w każdym kierunku. Sprawia to, że wycinki zdjęć mają rozmiary odpowiednio $\{250 \times 328, 270 \times 348, 190 \times 368, 310 \times 388, 350 \times 428, 390 \times 68, 430 \times 508, 530 \times 608\}$ (powiększenie o 230x308) i $\{1252 \times 1636, 1352 \times 1736, 1452 \times 1836, 1552 \times 1936, 1752 \times 2136, 1952 \times 2336, 2152 \times 2536\}$ (powiększenie 576x768).

W przypadku dopasowania opartego na szablonach, badano wszystkie rozmiary wycinków, zarówno dla korelacji krzyżowej, jak i współczynnika korelacji krzyżowej. Dla każdego wyniku klasyfikatora wybierana była jedynie najwyższa wartość prawdopodobieństwa uznawana za hipotetyczne położenie szczytu. Pozostałe rezultaty były odrzucane. Najpierw jednak, dla każdego ustawienia algorytmu ustalana była wartość progu. Odbywało się to poprzez wyliczenie dla każdej kombinacji współczynników wszystkich przypadków testowych. Na ich podstawie dobierana była taka wartość, przy której liczba poprawnych wskazań była największa. Przykład wartości wykorzystanych do wyznaczenia progu pokazano na rysunku 7.4. Wykres ten przedstawia uzyskane wartości prawdopodobieństwa dopasowania podzielone na trzy grupy: prawidłowo określone położenie, błędnie określone położenie oraz szczyt niewidoczny na zdjęciu rzeczywistym. Na rysunku została zaznaczona również wartość progu wybrana dla tego ustawienia algorytmu. Przy tak wyliczonych limitach przeprowadzano dalsze testy dla tej metody.



Rysunek 7.4. Przykład obliczonego podobieństwa dla poszczególnych przypadków przy użyciu algorytmu dopasowania szablonu. Na ich podstawie ustalany był próg wartości dla danych ustawień parametrów.

Feature matching zostało podzielone na cztery kategorie testów, po dwie dla każdego rodzaju dopasowania. Zarówno dla metody opartej na metodzie siłowej, jak i FLANN, były to progowanie punktów kluczowych i przeprowadzenie testu współczynników dla k -najbliższych sąsiadów. Każda z nich była badana dla deskryptorów i cech charakterystycznych uzyskanych przy pomocy algorytmów SIFT i ORB. W przypadku filtracji więcej niż jednej pary przy pomocą testu ratio, badane były następujące wartości tego parametru: {0, 9, 0, 75, 0, 6, 0, 5}. Natomiast, gdy wykorzystywane było progowanie najpierw ustalone zostały wartości graniczne dla poszczególnych przypadków testowych. Odbywało się to na podobnej zasadzie co w przypadku dopasowania na podstawie szablonu. Ze względu jednak, że dla metody opartej na cechach, liczba wartości obliczana na jeden szczyt jest większa niż w przypadku template matching, to określone zostały trzy wartości progów dla każdego ustawienia. Oznaczały one branie pod uwagę różnej liczby najlepszych trafień. W obu filtracjach i dla każdego algorytmu wykrywania cech, badano również maksymalną liczbę dopasowań braną pod uwagę przy procesie projekcji homograficznej: bez ograniczeń, 10 lub 20.

W poszczególnych przypadkach testowych, dla każdego szczytu przypisywany jest jeden z czterech możliwych wyników:

- **TP** (ang. True Positive) - prawdziwie pozytywne. Przypadki, które zostały poprawnie sklasyfikowane jako pozytywne - szczyt został poprawnie uznany za widoczny na zdjęciu, a jego położenie mieści się w granicach przyjętego błędu.
- **TN** (ang. True Negative) - prawdziwie negatywne. Przypadki, które zostały poprawnie sklasyfikowane jako negatywne - szczyt został poprawnie uznany za niewidoczny na zdjęciu.

- **FP** (ang. False Positive) - fałszywie pozytywne. Przypadki, które zostały błędnie sklasyfikowane jako pozytywne - szczyt został niepoprawnie uznany za widoczny lub jego estymowane położenie jest zbyt odległe od prawidłowego.
- **FN** (ang. False Negative) - fałszywie negatywne. Przypadki, które zostały błędnie sklasyfikowane jako negatywne - szczyt został błędnie uznany jako niewidoczny.

Na ich podstawie obliczane były cztery wskaźniki skuteczności klasyfikatora: dokładność, precyza, czułość i miara F1 [74]. W ogólności dla każdego z nich im większa wartość tym lepiej. Jednak dla lepszej interpretacji wyników wymagana jest analiza wszystkich wskaźników oraz liczby przypadków sklasyfikowanych jako TP, TN, FP i FN.

Dokładność. Stosunek poprawnie sklasyfikowanych przypadków do sumy wszystkich testów. Określa ogólną skuteczność klasyfikatora.

$$\text{Dokładność} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precyza. Liczba poprawnie pozytywnych przypadków do sumy prawdziwie i fałszywie pozytywnych wyników. Opisuje ile pozytywnych przypadków jest faktycznie prawdziwych.

$$\text{Precyza} = \frac{TP}{TP + FP}$$

Czułość. Proporcja prawdziwie pozytywnych rezultatów do sumy poprawnych pozytywów i fałszywych negatywów. Wyraża zdolność do wykrywania pozytywnych przypadków.

$$\text{Czułość} = \frac{TP}{TP + FN}$$

Miara F1. Jest to średnia harmoniczna precyzji i czułości. Pozwala określić jednym wskaźnikiem skuteczność klasyfikatora biorąc pod uwagę te dwie składowe.

$$F1 = \frac{2 * \text{Precyza} * \text{Czułość}}{\text{Precyza} + \text{Czułość}}$$

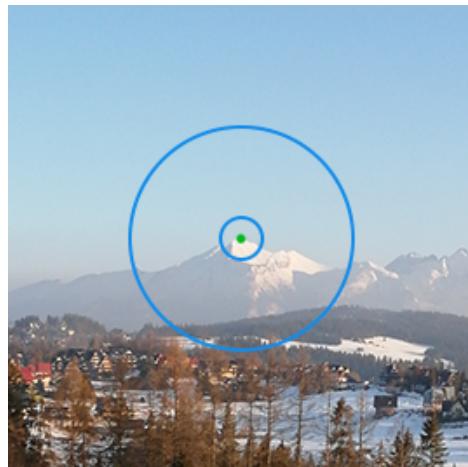
Estymowane położenie szczytu na obrazie było uznawane za poprawne, jeśli spełniało nierówność związaną z odlegością od oczekiwanej rezultatu:

$$\|P - E\| \leq K * \sqrt{w^2 + h^2}$$

gdzie:

- **P** (ang. Predicted) - obliczone położenie szczytu na obrazie.
- **E** (ang. Expected) - oczekiwane (prawidłowe) położenie szczytu na obrazie.
- **K** - współczynnik skalujący, określający dopuszczalną wielkość błędu.
- **w, h** - wymiary obrazu.

Współczynnik K pozwala określić wielkość dopuszczalnego błędu zwróconego położenia względem oczekiwanej. Oznacza on to jak dokładna predykcja położenia szczytu na obrazie jest uznawana za wystarczającą. W trakcie testów wykorzystywane były wartości tego współczynnika równe 0,05 i 0,01. Dla rozmiaru zdjęcia 768x1024 oznaczało to dopuszczalną odległość o wielkościach odpowiednio 64 i 12,8 pikseli. Interpretacja graficzna opisanych wartości współczynnika została pokazana na rysunku 7.5. Przedstawia on przestrzeń, w której musi znajdować się wyliczone położenie szczytu by wynik został uznany za prawidłowy. Zewnętrzny obszar przedstawia wartość $K = 0,05$, natomiast wewnętrzny $K = 0,01$. Punkt oznaczony kropką oznacza idealne dopasowanie. Wybrany do tej prezentacji szczyt oddalony jest od obserwatora o około 13 km. W tym przypadku wskazania na obwodzie szerszego okręgu wydają się zbyt odstające od rzeczywistego położenia by można uznać je za dokładne. Przy bliższych szczytach niedokładność nie byłaby tak wyraźna. Jednak z powodu, że w hipotetycznych warunkach rzeczywistych bardziej wartościowe wydają się wskazania dla gór będących w pewnej odległości od obserwatora, przy ostatecznym wyborze algorytmu większą uwagę zwracano na wyniki dla $K = 0,01$.



Rysunek 7.5. Dopuszczalny błąd dopasowania w zależności od wartości współczynnika tolerancji K dla wybranego szczytu.

7.4.1. Wpływ dylacji na skuteczność dopasowania obrazów

Dylacja [75] - jedna z operacji morfologicznych stosowanych w przetwarzaniu cyfrowym obrazu. Służy do zwiększenia oraz łączenia obiektów, wypełniania dziur czy pogrubienia konturów i linii. Polega na modyfikacji każdego piksela w zależności od jego sąsiedztwa określonego rozmiarem elementu strukturalnego (najczęściej macierzy kwadratowej) będącej parametrem dylacji. Jeśli badany punkt ma w okolicy przynajmniej jeden piksel o wartości 255 (lub 1,0 w zależności od sposobu reprezentacji koloru białego) to przyjmuje on również tę wartość.

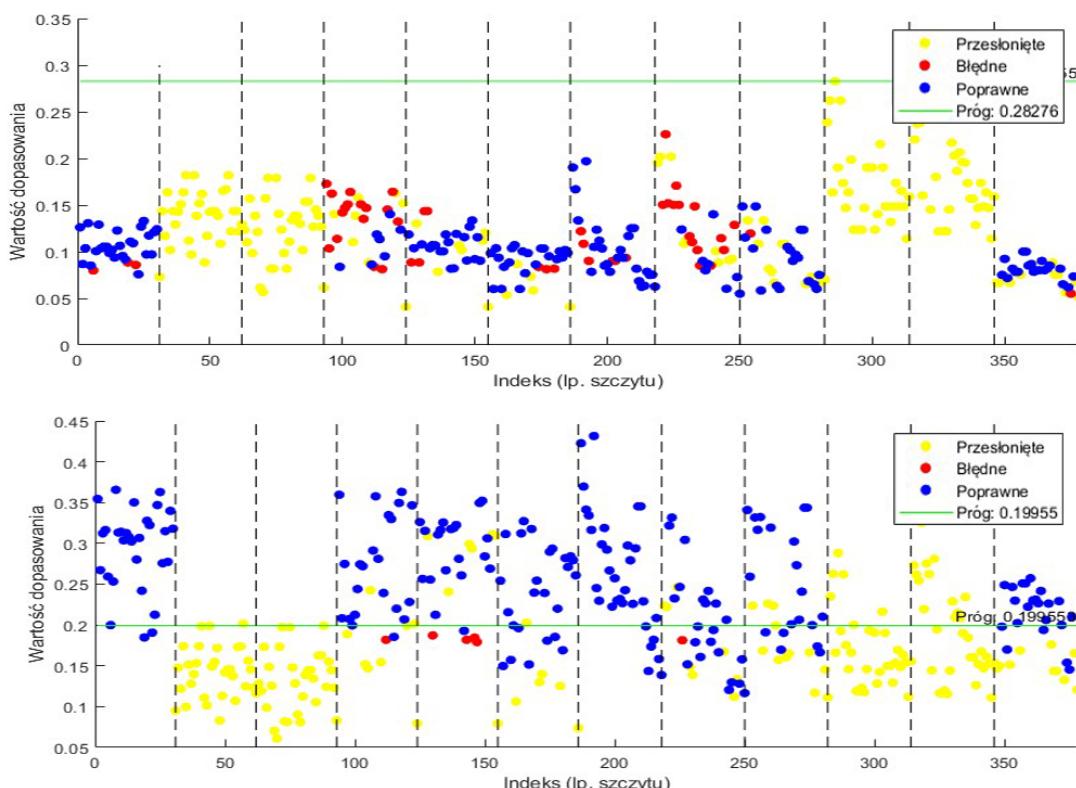
Operacja dylacji może być również przeprowadzona z wykorzystaniem funkcji zwracającej wartość maksymalną z danego zbioru. W tym wypadku ze wszystkich wartości pikseli będących w danym sąsiedztwie. Wtedy wzór na tę operację morfologiczną można zapisać jako:

$$(I \oplus B)(x, y) = \max \{I(x + b, y + c) : (b, c) \in B\}$$

gdzie $I(x,y)$ oznacza wartość danego piksela (luminancję), natomiast B element strukturalny.

Podczas generowania grafik związanych z dopasowaniem obrazów na potrzeby pracy magisterskiej, został wykorzystany mechanizm dylacji celem poprawy ich czytelności. W trakcie tego procesu, dla wybranego szczytu zaobserwowana została poprawa wskazania jego lokalizacji oraz zwiększenie wartości prawdopodobieństwa prawidłowego dopasowania. Mając to na uwadze, przeprowadzono wstępne testy wpływu dylacji na działanie algorytmów porównujących podobieństwo zdjęć przy losowych ustawieniach algorytmów.

Przedstawione na rysunku 7.6 wykresy punktowe prezentują wartości dopasowania szablonów poszczególnych szczytów dla jednakowych ustawień algorytmu, przy niemodyfikowanym zdjęciu oraz przy użyciu dylacji.



Rysunek 7.6. Przykład wpływu dylacji na wyniki algorytmu dopasowania obrazów na podstawie szablonów przy przykładowych ustawieniach parametrów. Górnny wykres przedstawia wyniki bez wykorzystania dylacji, dolny natomiast z wykorzystaniem dylacji.

Wskazania dla szczytów poprawnie oraz błędnie sklasyfikowanych, a także przesłoniętych przyjmują podobny zakres wartości w przypadku obrazów niemodyfikowanych. Widoczne to jest na górnym wykresie. Ciężko jest na nim wskazać jakikolwiek trend zachowania algorytmu i dobrać sensowną wartość progowania. Najlepszą wartość dokładności klasyfikatora metoda uzyskuje przy progu, dla którego wszystkie testy oznaczone zostają jako negatywne. Takie ewentualne działanie klasyfikatora nie jest w żaden sposób porządne i sensowne. Natomiast na wykresie dolnym, który pokazuje wyniki gdy obrazy zostały poddane operacji dylacji, można zaobserwować zwiększenie liczby poprawnych detekcji. Przede wszystkim jednak widoczny jest na nim pewien trend związany z wartościami w zależności od widoczności szczytu. Dla gór, które na zdjęciu rzeczywistym są niewidoczne, algorytm zwraca wartości zauważalnie niższe niż w przypadku, gdy szczyt znajduje się na zdjęciu. Dzięki temu, możliwe jest dobranie wartości progowania w sposób bardziej racjonalny. W tym przypadku, przekłada się to na zdecydowane zwiększenie dokładności klasyfikatora.

W tabeli 7.6 pokazano porównanie przykładowych wyników dopasowania obrazów z wykorzystaniem dylacji i bez. Są to rezultaty dla algorytmów wykorzystujących porównanie wzorca oraz cech przy losowo wybranych ustawieniach ich parametrów. W obu przypadkach użycie operacji morfologicznej pozwoliło zwiększyć wartości wszystkich metryk. Dla feature matching w przypadku czułości i miary F1 wskazania poprawione zostały o około 20 punktów procentowych. Udało mu się też sklasyfikować prawie czterokrotnie więcej przypadków jako prawdziwie pozytywne. Badane ustawienie template matching dużo bardziej widocznie poprawiło się po zastosowaniu operacji dylacji. Sprawiła ona, że klasyfikator nie wskazywał już wszystkich przypadków jako negatywne. Pozwoliło to określić aż 151 szczytów jako prawdziwie pozytywne, zwiększając dokładność o 30 punktów procentowych. Ze względu na pierwotne wskazanie wszystkich przypadków jako negatywne, gdy algorytm nie wykorzystywał dylacji, pozostałe wskaźniki były wtedy równe 0%. Użycie operacji morfologicznej oznaczało skok wartości tych metryk do poziomu ~ 80%. Jest to niebagatelna i znacząca poprawa jakości tego klasyfikatora.

Tabela 7.6. Porównanie skuteczności algorytmów dopasowania obrazów na podstawie szablonu oraz na podstawie cech przy przykładowych ustawieniach parametrów tych metod z wykorzystaniem operacji dylacji oraz bez.

	TP	TN	FP	FN	Dokładność	Precy-zja	Czułość	F1
Dopasowanie na podstawie szablonu								
Bez dylacji	0	179	0	199	47,3%	0,00%	0,00%	0,00%
Z dylacją	151	145	34	48	78,3%	81,6%	75,8%	78,6%
Dopasowanie na podstawie cech								
Bez dylacji	10	164	22	182	46,0%	31,2%	5,2%	8,9%
Z dylacją	39	144	71	124	48,4%	35,4%	23,9%	28,3%

Wyżej wymienione przykłady poprawy jakości klasyfikatora dzięki użyciu operacji dylacji sugerują zasadność wykorzystania tej techniki dla procesu porównania obrazów binarnych zawierających krawędzie gór. Z tego powodu, przeprowadzone zostały dalsze testy porównujące wyniki algorytmów dopasowania zdjęć przy pomocy szablonów oraz cech charakterystycznych, w zależności od wykorzystania tej techniki. Badane były kombinacje zastosowania dylacji na obrazach binarnych wyrenderowanego modelu oraz rzeczywistego zdjęcia:

- Brak dylacji na obu obrazach.
- Dylacja tylko na wyrenderowanym obrazie.
- Dylacja tylko na rzeczywistym zdjęciu.
- Dylacja na obu obrazach.

Trzy ostatnie przypadki były porównywane do rezultatów uzyskanych bez wykorzystania dylacji (test kontrolny). Wynik uznawany był za lepszy dla danej miary jakości, jeśli jej wartość była wyższa lub równa wynikowi testu kontrolnego. W przypadku dopasowania na podstawie cech, jeśli algorytm nie był w stanie zwrócić żadnego punktu kluczowego dla wersji bez dylacji, natomiast po zastosowaniu tej modyfikacji takie punkty już istniały to niezależnie od uzyskanego wyniku, druga wersja uznawana była zawsze za lepszą. W przypadku, gdy mimo dylacji występował w dalszym ciągu brak punktów kluczowych to taki przypadek testowy nie był brany pod uwagę i pomijany, mimo że spełniał on zależność lepszy lub tak samo dobry. Podane poniżej rezultaty dla algorytmów template i feature matching odpowiadają sumie wyników dla $K = 0,05$ i $K = 0,01$ w każdym przypadku testowym.

Dla algorytmu dopasowania na podstawie wzorca, porównanie wyników w zależności od zastosowania dylacji przedstawiono w tabeli 7.7. Dla wszystkich trzech kombinacji, wszystkie wskaźniki były przeciętnie lepsze niż w teście kontrolnym. Dokładność była aż w 90% testach wyższa dla dylacji na modelu i dylacji na obu obrazach. Wyniki te były lepsze niż bez wykorzystania dylacji odpowiednio w 29 i 28 przypadkach na 32. Oba te ustalenia w większości przypadków osiągały również lepszy wynik miary F1 niż test kontrolny - 81,3% oraz 78,1%. Dylacja na zdjęciu rzeczywistym uzyskała słabsze wyniki niż pozostałe dwa rodzaje, ale mimo to, w dalszym ciągu okazała się lepsza niż brak dylacji na obrazach. W jej przypadku największy zysk jest w czułości, bo lepsze rezultaty osiągnęła aż dla 27 na 32 testy.

Warto odnotować również fakt, że najlepsze wyniki dla wskaźników dokładności i miary F1, zarówno przy $K = 0,05$, jak i $K = 0,01$ metoda dopasowania szablonu uzyskiwała przy wykorzystaniu dylacji na obu obrazach. W przypadku funkcji precyzji i czułości, wartości równe 1,0 uzyskiwane były zarówno z wykorzystaniem dylacji jak i bez. W tym wypadku jednak trudno mówić o sensowności takich danych, ponieważ w przypadku precyzji występowało to, gdy klasyfikator zwrócił na przykład jedynie 10 poprawnie pozytywnych przypadków, a resztę klasyfikował jako negatywne.

Tabela 7.7. Wpływ operacji morfologicznej dylacji na algorytm dopasowania na podstawie szablonu.

	Dokładność	Precyzja	Czułość	F1
Dylacja na renderze	29/32 (90,6%)	26/32 (81,3%)	20/32 (62,5%)	26/32 (81,3%)
Dylacja na zdjęciu rzeczywistym	26/32 (75,0%)	24/32 (62,5%)	20/32 (85,7%)	22/32 (68,8%)
Dylacja na obu obrazach	28/32 (87,5%)	29/32 (90,6%)	21/32 (65,6%)	25/32 (78,1%)

Rezultaty na podobnym poziomie uzyskane zostały również w przypadku algorytmu porównującego cechy i deskryptory. Wyniki podzielone na sposób dopasowania punktów kluczowych i ich filtrowanie przedstawiono w tabeli 7.8. Ważnym aspektem jest wpływ operacji morfologicznej na możliwość wykrywania punktów kluczowych przez algorytm SIFT. W przypadku obrazów bez dylacji lub modyfikacji tylko jednego z dwóch obrazów, nie był on w stanie zwrócić nawet jednego punktu kluczowego dla żadnego przypadku testowego. Natomiast, jeśli wykorzystana była dylacja na obu obrazach to był on w stanie podać wystarczającą liczbę punktów kluczowych i estymować położenie szczytu. W tym wypadku, wpływ operacji morfologicznej na poprawę skuteczności klasyfikatora jest bezdyskusyjny, ponieważ pozwala na jakiekolwiek jego działanie.

W przeciwnieństwie do dopasowania szablonu, występowały tutaj ustalenia, dla których brak wykorzystania dylacji dawał częściej lepsze wyniki. Miało to miejsce jednak jedynie dla wskazań przy wykorzystaniu pewnego progu dopasowania cech, a operacja morfologiczna była użyta tylko na jednym z dwóch obrazów. Natomiast, gdy do filtrowania połączonych cech wykorzystany został test współczynników rezultaty były już odmienne, ze wskazaniem na proces dylacji. W pozostałych przypadkach modyfikacja zdjęć dawała lepsze wyniki w większości przypadków. Jeśli dylacja była zastosowana na obu obrazach to średnia dokładność wahała się między 75% a 80% oraz miara F1 w okolicach 90%. Na podobnym poziomie uzyskiwane były wtedy wartości metryk precyzji oraz czułości.

Na podstawie przedstawionych powyżej porównań dotyczących wykorzystania dylacji, można wyciągnąć wnioski, że w wielu przypadkach pozwala ona uzyskać lepsze rezultaty. Dla dopasowania na podstawie szablonu, poprawa wszystkich wskaźników występowała w co najmniej połowie testów, osiągając przy niektórych nawet 90% wzmacniania. W przypadku analizy cech zdarzały się przypadki gdzie lepsze wyniki uzyskiwała algorytm bez wykorzystania dylacji. W ogólności jednak i dla tej metody polepszenie się wyników występowało średnio w ponad 75% przypadków. Wydaje się, że zaproponowana modyfikacja procesu dopasowania obrazów może mieć znaczący wpływ na jego działanie oraz zwiększać jakość klasyfikacji. Jednak, ze względu, że wyniki te nie są jednoznaczne, w pozostałych testach, opisanych w następnym podrozdziale, badane były kombinacje ze wszystkimi rodzajami użycia dylacji.

Tabela 7.8. Wpływ operacji morfologicznej dylacji na algorytm dopasowania na podstawie cech.

	Dokładność	Precyza	Czułość	F1
Algorytm siłowy + Progowanie				
Dylacja na renderze	44/90 (48,9%)	52/90 (57,8%)	82/90 (91,1%)	64/90 (71,1%)
Dylacja na zdjęciu rzeczywistym	32/90 (35,6%)	44/90 (48,9%)	68/90 (75,6%)	44/90 (48,9%)
Dylacja na obu obrazach	178/234 (76,1%)	196/234 (83,8%)	222/234 (94,9%)	210/234 (89,7%)
Algorytm siłowy + Test współczynników				
Dylacja na renderze	176/192 (91,7%)	156/192 (81,3%)	160/192 (83,3%)	152/192 (79,2%)
Dylacja na zdjęciu rzeczywistym	176/192 (91,7%)	156/192 (81,3%)	160/192 (83,3%)	152/192 (79,2%)
Dylacja na obu obrazach	290/384 (75,5%)	348/384 (90,6%)	360/384 (93,8%)	348/384 (90,6%)
FLANN + Progowanie				
Dylacja na renderze	40/90 (44,4%)	46/90 (51,1%)	76/90 (84,4%)	54/90 (60,0%)
Dylacja na zdjęciu rzeczywistym	48/90 (53,3%)	46/90 (51,1%)	72/90 (80,0%)	48/90 (53,3%)
Dylacja na obu obrazach	204/258 (79,1%)	212/258 (82,2%)	250/258 (96,9%)	228/258 (88,4%)
FLANN + Test współczynników				
Dylacja na renderze	170/192 (88,5%)	156/192 (81,3%)	160/192 (83,3%)	150/192 (78,1%)
Dylacja na zdjęciu rzeczywistym	160/192 (83,3%)	160/192 (83,3%)	156/192 (81,3%)	150/192 (78,1%)
Dylacja na obu obrazach	288/384 (75,0%)	348/384 (90,6%)	364/384 (94,8%)	348/384 (90,6%)

7.4.2. Porównanie algorytmów dopasowania obrazów

Skuteczność algorytmów dopasowania obrazów była badana również poprzez porównanie wartości wskaźników jakości uzyskanych przez metodę opartą na wzorach oraz na punktach kluczowych i deskryptorach. Oba algorytmy zostały zgrupowane w całość, niezależnie od użytych parametrów poszczególnych metod. Porównanie odbywało się poprzez zestawienie ze sobą maksymalnych wartości dokładności, precyzyji, czułości i miary F1 uzyskiwanej przez oba klasyfikatory dla współczynnika skalującego o wartościach $K = 0,05$ i $K = 0,01$.

Wyniki przy większej tolerancji błędu przedstawione zostały w tabeli 7.9. Większość zaprezentowanych w niej wyników jest lepszych dla dopasowania obrazów na podstawie szablonu. Najlepszy uzyskany przez nią wynik dokładności wyniósł 86,8%. Natomiast w przypadku dopasowania cech wartość ta była niższa aż o 36,3 p.p., a w liczbie popraw-

7. Porównanie wybranych algorytmów i próby optymalizacji

nych dopasowań o 137. Feature matching uzyskał 100% wartości wskaźników precyzji oraz czułości. Jednak są to mało wartościowe dane. Wynika to z faktu, że dla precyzji zwracał on prawie wszystkie klasyfikacje jako negatywne, a w przypadku czułości jako pozytywne. Podkreśla to też dokładność wynoszącą jedynie 26,5% przy 100% metryki czułości. Dla precyzji identyczny efekt występuje również przy wykorzystaniu korelacji między szablonem, a zdjęciem wejściowym. Mimo to, udało mu się sklasyfikować o kilka więcej prawdziwie pozytywnych niż rozwiązywanie wykorzystujące cechy. Wskaźnik czułości, w przypadku template matching, ma jednak więcej sensu, ponieważ starał się on dokładnie sklasyfikować wszystkie szczyty, co potwierdza dokładność - 82,5%. Algorytm oparty na cechach, gdy uzyskiwał wysoką precyzję to wskaźnik czułości był niski - i na odwrót. Z tego względu przewidywalna była przeciętna wartość miary F1 - 50,5%. Druga z porównywanych metod uzyskała dużo lepszą wartość tej metryki, bo prawie 87%.

Tabela 7.9. Porównanie skuteczności algorytmów dopasowania obrazów na podstawie szablonu oraz na podstawie cech przy dopuszczalnym błędzie $K = 0,05$.

	TP	TN	FP	FN	Dokładność	Precyza	Czułość	F1
Ze względu na dokładność								
Dopasowanie szablonu	162	166	15	35	86,8%	91,5%	82,2%	86,6%
Dopasowanie cech	79	112	140	47	50,5%	36,1%	62,7%	45,8%
Ze względu na precyzję								
Dopasowanie szablonu	12	175	0	191	49,5%	100,0%	5,9%	11,2%
Dopasowanie cech	3	179	0	196	48,1%	100,0%	1,5%	3,0%
Ze względu na czułość								
Dopasowanie szablonu	172	140	41	25	82,5%	80,7%	87,3%	83,9%
Dopasowanie cech	100	0	278	0	26,5%	26,5%	100,0%	41,8%
Ze względu na F1								
Dopasowanie szablonu	162	166	15	35	86,8%	91,5%	82,2%	86,6%
Dopasowanie cech	113	35	201	29	39,2%	36,0%	79,6%	50,0%

Porównanie rezultatów przy współczynniku $K = 0,01$ jeszcze bardziej pogłębiło różnice między dopasowaniem na podstawie wzorca, a porównaniem cech, na korzyść tego pierwszego. Wyniki te przedstawione zostały w tabeli 7.10. Szczególnie duża rozbieżność między tymi metodami widoczna jest przy porównaniu współczynnika F1. W tym przypadku różnica wyniosła prawie 60 p.p., przy zestawieniu najwyższych wartości tej metryki. Dla tego

wskazania bardzo kiepsko wypada dopasowanie cech również pod względem dokładności, ponieważ uzyskuje raptem 17%, przy 70% wartości drugiej metody dla tej miary. Podobnie jak przy większej tolerancji, tak i tutaj, nie ma sensu przykładać uwagi do precyzji i czułości ze względu na charakter klasyfikatorów, które uzyskały te wyniki. Najlepszą dokładność, 48,1%, metoda dopasowania na podstawie cech uzyskała w momencie, gdy klasyfikowała jedynie 3 przypadki jako pozytywne, a pozostałe jako negatywne. Stąd niska wartość miary F1 - 3,0% - tego ustawienia.

Tabela 7.10. Porównanie skuteczności algorytmów dopasowania obrazów na podstawie szablonu oraz na podstawie cech przy dopuszczalnym błędzie $K = 0,01$.

	TP	TN	FP	FN	Dokładność	Precyza	Czułość	F1
Ze względu na dokładność								
Dopasowanie szablonu	129	166	48	35	78,0%	72,9%	78,7%	75,7%
Dopasowanie cech	3	179	0	196	48,1%	100,0%	1,5%	3,0%
Ze względu na precyzję								
Dopasowanie szablonu	12	175	0	191	49,5%	100,0%	5,9%	11,2%
Dopasowanie cech	3	179	0	196	48,1%	100,0%	1,5%	3,0%
Ze względu na czułość								
Dopasowanie szablonu	129	166	48	35	78,0%	72,9%	78,7%	75,7%
Dopasowanie cech	20	14	334	0	9,0%	5,5%	100,0%	10,4%
Ze względu na F1								
Dopasowanie szablonu	129	166	48	35	70,0%	72,9%	78,7%	75,7%
Dopasowanie cech	34	29	285	30	16,7%	10,7%	53,1%	17,8%

Opisane powyżej wyniki dla porównań obu metod w zależności od tolerancji błędu pokazują dużo większą jakość klasyfikatorów opartych na dopasowaniu na podstawie szablonu. Uzyskiwały one lepsze wyniki dokładności i miary F1 w porównaniu do porównania cech. W szczególności uwidocznioło się to przy $K = 0,01$, gdy metody wykorzystujące punkty kluczowe i deskryptory uzyskiwały bardzo niskie wyniki.

Jednak w procesie związanym z tematem pracy sensowne wydaje się faworyzowanie przypadków prawdziwie pozytywnych, a karanie fałszywie pozytywnych. Lepszym wyjściem jest nieoznaczanie widocznych szczytów, niż błędnie podać ich lokalizacje lub wskazać te, które w rzeczywistości są niewidoczne. W teorii takie informacje ma mieć wskaźnik precyzji ($P = \frac{TP}{TP+FP}$). Pewne ustawienia badanych metod potrafiły jednak osią-

gnąć nawet 100% tej miary. Działo się to wtedy, gdy większość predykcji była w klasie negatywnej i tylko kilka (np. 3 TP) prawdziwe pozytywnych. Jednak w takich przypadkach, gdy rozkład klas jest w przybliżeniu równomierny, dokładność wynosi jedynie około 50%. Dla takich predykcji wskaźnik precyzji wydaje się mało informatywny i nie spełnia swojego celu. Ze względu jednak na chęć faworyzowania predykcji prawdziwie pozytywnych, wyszukane zostały takie przypadki testowe, dla których precyzja była największa oraz liczba TP wynosiła co najmniej $0.85 * \max(\text{TP})$ (85% największej liczby TP z danego zbioru). Porównanie wskaźnika precyzji przy takim założeniu pokazano w tabeli 7.11.

Tabela 7.11. Porównanie wartości wskaźnika precyzji przy założeniu minimalnej liczby prawdziwie pozytywnych przypadków dla algorytmów dopasowania obrazów na podstawie szablonu oraz na podstawie cech, gdy dopuszczalny błąd $K = 0,05$ i $K = 0,01$.

	TP	TN	FP	FN	Dokładność	Precyzja	Czułość	F1
$K = 0,05$								
Dopasowanie szablonu	162	166	15	35	86,8%	91,5%	82,2%	86,6%
Dopasowanie cech	100	72	170	36	45,5%	37,0%	73,5%	49,3%
$K = 0,001$								
Dopasowanie szablonu	113	168	26	71	74,3%	81,3%	61,4%	70,0%
Dopasowanie cech	34	29	285	30	16,7%	10,7%	53,1%	17,8%

W przypadku algorytmu dopasowania przy użyciu korelacji dwóch obrazów, gdy współczynnik błędu wynosi 0,05, wynik dla najlepszej precyzji pokrywa się z tym, uzyskanym przy najwyższej dokładności. Dla niego, precyzja wyniosła aż 91,5%, natomiast poprawność wszystkich wskazań jest na poziomie 86,8%. Warto również, ponownie, podkreślić wysokie wartości pozostałych wskaźników - czułość 82,2%, miara F1 86,6%. Przy pięcioprocentowej tolerancji błędu, dopasowanie na podstawie cech, tak jak w wyżej opisanych porównaniach, osiąga dużo gorsze wyniki niż odpowiednik oparty na szablonach. Najlepsza precyzja wyniosła jedynie 37%, przy równie niskiej dokładności - 45,5%. Podobnie miara F1 nie przekroczyła progu 50%. Znaczna różnica jest też w liczbie przypadków prawdziwie pozytywnych - 162 i 100. Mała skuteczność algorytmu porównania punktów kluczowych objawia się także w liczbie fałszywie pozytywnych przypadków - 170. Ta liczba jest wyższa o 155 niż uzyskana przez template matching.

Dla mniejszej tolerancji błędu, $K = 0,01$, rozbieżność między jakością dopasowania na podstawie szablonu, a punktów charakterystycznych była jeszcze większa. Pierwsza z metod uzyskała precyzję ponad 81%, gdy druga niecałe 11%. W przypadku wykorzystania wzorca, całkiem wysoka wartość wystąpiła również dla wskaźnika dokładności, bo prawie 75%. Mimo to, dla tak restrykcyjnego błędu, poprawnie wskazane zostały lokalizacje aż

113 widocznych szczytów i tylko 26 fałszywie pozytywnych. Wynika to z faktu, że przy tych ustawieniach, metoda wolała klasyfikować przypadek jako negatywny niż ocenić błędnie jako pozytywny. Dlatego wynik czułości jest relatywnie niski - 61,4%. Jednak takie podejście o wyborze detekcji jako negatywną, zamiast fałszywie pozytywną, a co za tym idzie zwiększając precyzję, pokrywa się z opisany wyżej preferowanym sposobem klasyfikacji. W przypadku metody opartej na cechach i punktach kluczowych liczba błędnie wskazanych pozytywnych przypadków osiągnęła aż 285, co stanowi 75% wszystkich prób. Skutkiem tego jest bardzo niska precyzja na poziomie 10%. Mimo, że jest to bardzo słaby wynik, to był on najlepszy z uzyskanych przy określonej minimalnej liczbie TP.

Przeprowadzone porównanie pokazało niską skuteczność i jakość klasyfikatora opartego na cechach i punktach kluczowych. Uzyskał on dokładność na poziomie 16% i precyzję około 11% w teście z minimalną liczbą przypadków poprawnie pozytywnych oraz aż 285 fałszywie pozytywnych, co całkowicie dyskwalifikuje możliwość wykorzystania tego rozwiązania. Dopasowanie na podstawie szablonu uzyskało wyniki na dobrym poziomie, nawet dla minimalnej tolerancji błędu. Z powodu odrzucenia feature matching już na tym etapie, ze względu na bardzo niską jakość klasyfikatora dla problemu porównań szczytów, pominięte zostały jakiekolwiek testy złożoności czasowej algorytmów i ich porównanie.

7.4.3. Podsumowanie badania algorytmów dopasowania obrazów

Przeprowadzone badania algorytmów dopasowania obrazów pozwoliły na wyciągnięcie odpowiednich wniosków oraz na wybór optymalnych rozwiązań w tym aspekcie pod kątem procesu identyfikacji szczytów górskich. Zostały przeprowadzone testy wykorzystania dylacji na obrazach binarnych oraz jaki wpływ ma ta modyfikacja na jakość klasyfikatorów. Porównane zostały również ze sobą algorytmy dopasowania obrazów na podstawie wzorca i cech.

Testowanie wpływu dylacji nie przyniosło wyników jednoznacznie potwierdzających przewagę dopasowania obrazów z wykorzystaniem tej techniki względem operacji na surowych zdjęciach. Dla większości przypadków testowych wykorzystanie dylacji wykazywało jednak lepszą klasyfikację pod względem miary dokładności. Istotnym aspektem jest również jej wpływ na działanie deskryptora SIFT przy dopasowaniu na podstawie cech. W tym wypadku dylacja umożliwiła jakiekolwiek działanie temu algorytmowi, ponieważ na wejściowych obrazach nie był on w stanie wskazać żadnych punktów kluczowych. Dodatkowo, najlepsze wyniki precyzji oraz dokładności, dla dopasowania wzorca uzyskane były przy wykorzystaniu tej techniki. Wpływ na template matching oraz statystycznie lepsza skuteczność sprawiają, że wykorzystanie tej operacji morfologicznej w implementacji projektu wydaje się uzasadnione i potrzebne.

Porównanie metod dopasowania obrazów, dla obu wartości współczynnika błędu pokazały bezdyskusyjną przewagę jakości klasyfikatora opartego na korelacji krzyżowej szablonów. Różnica w ich najlepszych wartościach dokładności była na poziomie 30 – 35 punktów procentowych. Jest to duża rozbieżność w tym jaką liczbę poprawnych przypad-

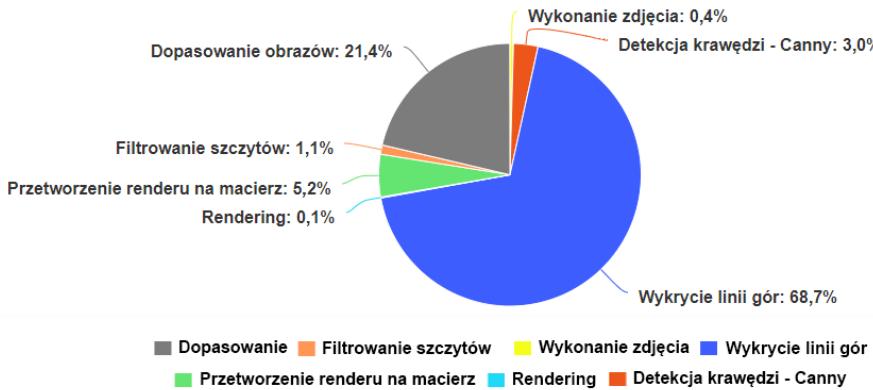
ków oba z nich były w stanie sklasyfikować. Jednak najbardziej wartościowym wyróżnikiem w tej kwestii wydaje się porównanie precyzji przy określonej minimalnej liczbie TP. Wtedy wykorzystanie szablonu przewyższało porównanie cech o 41,3 p.p dokładności oraz 54,5 p.p. precyzji przy $K = 0,05$. Różnice te powiększają się do odpowiednio 57,6 p.p. i 70,6 p.p. przy bardziej restrykcyjnej tolerancji błędu $K = 0,01$. Dopuszczalna mniejsza niedokładność, zgodnie z wcześniejszymi założeniami, ma większe znaczenie w kontekście sprawności i jakości klasyfikatora. W jej przypadku algorytm oparty na cechach całkowicie sobie nie radził uzyskując raptem 16,7% dokładności. Są to wielkości zbyt niskie by można mówić o możliwości wykorzystania tego algorytmu w projekcie rozpoznawania szczytów górskich.

Przedstawione wartości jednoznacznie pokazują różnice skuteczności pomiędzy tymi metodami dla zadania porównania obrazów binarnych gór. Dobre wyniki dopasowania z wykorzystaniem wzorca potwierdzają rozważania teoretyczne o efektywności tego algorytmu przy porównywaniu prostych obrazów o podobnej intensywności. Opisany i uzasadniony w poprzednim podrozdziale brak porównania złożoności czasowej tych rozwiązań sprawił, że wybór opierał się tylko na jakości klasyfikatorów. Z tego powodu za lepszą uznana została metoda porównania oparta na szablonach, wspierana przez proces dylacji.

7.5. Wpływ kolejności przetwarzania pikseli zdjęcia i wykorzystania pamięci podręcznej procesora

Celem wykrycia ewentualnych błędów implementacji poszczególnych algorytmów oraz weryfikacji wyników wykonanych testów przeprowadzono profilowanie procesu identyfikacji szczytów górskich wykonywanego na platformie testowej. Odbyło się to przy symulacji realnych warunków, gdzie widoczne jest około 30 szczytów, a wartości parametrów dopasowania obrazów ustalone zostały zgodnie z najlepszymi wynikami testów przeprowadzonych w poprzednim podrozdziale. Pomiary czasu były przeprowadzone w cyklu 1000 klatek, powtórzone 10 razy. Ich wynik w formie graficznej pokazano na wykresie 7.7. Łatwo zauważyc, że najwięcej czasu system przeznaczał na wyznaczenie najwyższej linii gór przetwarzając obraz binarny z wykrytymi krawędziami. Porównując złożoność tego mechanizmu na przykład do metody Canny, która jest złożonym, wielo-etapowym procesem, można stwierdzić, że tak nie powinno być i hipotetycznie założyć błędą lub nieoptimalną implementację tego etapu.

Z punktu widzenia człowieka oraz tego jak przetwarza on pewne dane, proces wybierania najwyższych krawędzi na danym obrazie binarnym powinien odbywać się kolumna po kolumnie, sprawdzając piksele od góry do dołu. W momencie natrafienia na biały punkt powinien on go wybrać i przejść do następnej kolumny. Takie podejście wydaje się logiczne z takiego punktu widzenia i w ten sposób najpierw został zaimplementowany ten etap. Jednak jest to sposób dobry tylko dla człowieka.



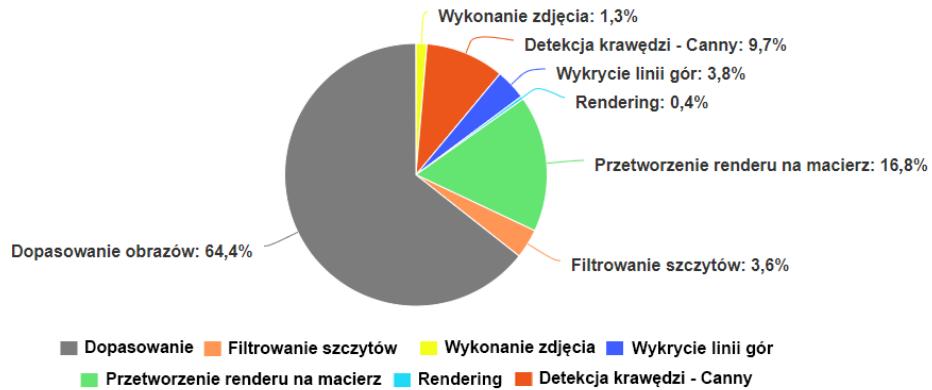
Rysunek 7.7. Profilowanie procesu identyfikacji szczytów górskich przed optymalizacją wykrywania krawędzi najwyższych gór.

Najczęstszą formą reprezentacji zdjęć w pamięci jest zapis rzędami. Oznacza to, że wartości pikseli w danym rzędzie są kolejnymi wpisami pamięci. Taki format występuje również w OpenCV [76]. W tym wypadku dla procesora lepszy jest dostęp do zdjęcia rzędami, ponieważ minimalizuje liczbę zapytań do pamięci i skoków między adresami odpowiednich komórek. Pozwala to również na wykorzystanie pamięci podrzcznej CPU, ze względu na to, że może on pobrać i zapisać w niej pewną liczbę kolejnych pikseli. Nowoczesne procesory wyposażane są w tzw. prefetcher, który automatycznie wykrywa wzorce dostępu do pamięci i z wyprzedzeniem pobiera dane [77]. Dzięki temu nie ma potrzeby ciągłego odpytywania pamięci RAM i przesyłania kolejnych informacji. Dodatkowo, dostęp do pamięci podrzcznej w porównaniu do operacyjnej jest sporo szybszy [78].

Takie rozważania teoretyczne potwierdza diagram 7.8. Po zmianie implementacji na analizę obrazu rzędami, wykrycie linii góra stanowi już jedynie 4% czasu całego cyklu. Mimo, że w tym przypadku przetwarzane było całe zdjęcie, a nie tylko niektóre wartości w poszczególnych kolumnach. Możliwe jest dodatkowe zmniejszenie czasu przez zastosowanie flag logicznych celem opuszczenia pętli po znalezieniu wszystkich poszukiwanych pikseli. W przypadku pesymistycznym, gdy wszystkie potrzebne krawędzie będą znajdować się w dolnej części zdjęcia, może pogorszyć to efektywność. Jednak możliwe jest założenie, że potencjalne oprogramowanie stworzone na podstawie opisywanego w niniejszej pracy dyplomowej procesu, będzie wykorzystywane w taki sposób, że krawędzie analizowanych gór będą znajdować się w środkowej części kadru. Dla tak umiejscowionych szczytów udział tego etapu w całym cyklu spada do 3%.

Po zmianie implementacji wykrywania linii góra, najwięcej czasu procesor poświęca na dopasowanie obrazów. Związane to jest z koniecznością wycinania obszaru dla każdego szczytu z obu obrazów binarnych, tworząc za każdym razem nowe macierze. Następnie przeprowadzana jest analiza dla każdego takiego fragmentu. Sumarycznie prowadzi to do dużej ilości obliczeń wykonywanych przez procesor. Z tego powodu prawdopodobnie poprawny jest tak duży procentowy udział tego etapu w cyklu detekcji szczytów.

7. Porównanie wybranych algorytmów i próby optymalizacji



Rysunek 7.8. Profilowanie procesu identyfikacji szczytów górskich po optymalizacji wykrywania krawędzi najwyższych gór poprzez zmianę kolejności analizy pikseli i wykorzystaniem pamięci podręcznej.

Analiza ta pokazuje, że odpowiednie przetwarzanie tablic wielowymiarowych oraz wykorzystanie pamięci podręcznej mogą być kluczowymi czynnikami wpływającymi na złożoność czasową danego zagadnienia. Można też wyciągnąć z tego wnioski, że reprezentacja w pamięci danych dwu i więcej wymiarowych w sposób ciągły pozwala zoptymalizować do nich dostęp i zmniejszyć liczbę potrzebnych zapytań. Zasadne wydaje się w takich przypadkach pisanie kodu w sposób zorientowany na optymalizację nawet kosztem czytelności i refactoringu czy nawet logiczności i zgodności z ludzką intuicją.

8. Weryfikacja końcowa

Uzyskane w trakcie badania algorytmów rezultaty pozwoliły również opracować proces potrafiący identyfikować szczyty górskie na obrazie. Z tego powodu przeprowadzono weryfikację końcową wykorzystanych metod jako całości. Odbyło się to zarówno na statycznych zdjęciach, a także w czasie rzeczywistym na kolejnych klatkach nagrani pozyskiwanych z aparatu urządzenia mobilnego. Poniżej przedstawiono rezultaty takiej weryfikacji, a także wyniki czasowe uzyskiwane podczas działania na żywo - w szczególności liczbę klatek na sekundę.

8.1. Prezentacja działania rozwiązań

Na rysunku 8.1 pokazany został przykładowy wynik otrzymany przy pomocy zaimplementowanego procesu na zdjęciu statycznym. Ze względu na gęstość szczytów na obrazie wejściowym, ograniczona została liczba naniesionych nazw, celem zwiększenia przejrzystości i czytelności. Obok nazw gór, w nawiasach została podana odległość w kilometrach od obserwatora.



Rysunek 8.1. Przykładowy rezultat rozpoznawania szczytów górskich na obrazie.

Celem demonstracyjnym przygotowano również nagrania z naniesionymi nazwami gór zrobione w paśmie Tatr, a także grafiki prezentujące wyniki kolejnych kroków procesu dla wybranych zdjęć. Opisane elementy prezentacyjne oraz kod projektu zostały umieszczone na repozytorium git autora, do którego odnośnik wraz z krótkim opisem zawartości umieszczono w załączniku 1.

8.2. Złożoność czasowa rozwiązania

Założeniem pracy dyplomowej było działanie algorytmów w czasie rzeczywistym. Z tego powodu w trakcie tworzenia nagrania demonstracyjnego zebrane zostały również dane dotyczące czasu trwania procesu rozpoznawania szczytów górskich.

Pozwoliło to sprawdzić liczbę klatek generowanych przez stworzone rozwiązanie w warunkach rzeczywistych. Przeciętnie dopasowywane było około 30 gór, z czego 15 – 20 było rozpoznanych i etykietowanych na każdej pobranej z kamery klatce. Do testu i prezentacji wykorzystane zostało jedynie urządzenie Huawei Mate 50 Pro. Wyniki czasu przetwarzania zostały przedstawione w tabeli 8.1. Badanie wykonane było 10 razy i trwało 1000 klatek każde, a rezultaty uśrednione.

Tabela 8.1. Czas przetwarzania procesu rozpoznawania szczytów górskich na obrazie w warunkach rzeczywistych.

Całkowity czas	Średni czas jednej klatki	Liczba klatek na sekundę
44,9635 s	0,044 s	22,24 FPS

Rozwiązanie w trakcie testu generowało 22,24 klatek na sekundę, co wydaje się wynikiem zadowalającym. Jest to wartość zbliżona do standardu kinowego, w którym filmy na ekranie wyświetlane są z częstotliwością 24 klatek. Z tego powodu uzasadnione jest stwierdzenie, że uzyskany rezultat oznacza wystarczający komfort użytkowania na nowszych urządzeniach mobilnych.

8.3. Podsumowanie weryfikacji końcowej

Przeprowadzona weryfikacja końcowa zaimplementowanego procesu identyfikacji szczytów górskich na obrazie pozwoliła stwierdzić, że uzyskane wyniki badanych i optymalizowanych algorytmów mogą być zastosowane w projektach mających na celu takie działanie. Na podstawie sprawdzonego działania rozwiązania zarówno na statycznych zdjęciach, jak i nagraniu na żywo można uznać, że w wystarczającym stopniu dokładnie opisuje ono widoczne szczyty górskie. Natomiast, rezultaty pomiaru liczby klatek na sekundę oznaczają, że rozwiązanie może zostać sklasyfikowane jako działające w czasie rzeczywistym na badanym urządzeniu.

9. Podsumowanie

9.1. Wykonane prace

W trakcie opracowywania pracy dyplomowej przeprowadzono następujące po sobie etapy tworzenia projektu informatycznego opartego na pracach badawczych i testowaniu algorytmów. Rozpoczęty został od analizy literatury oraz przygotowania teoretycznego. Następnie stworzona została platforma testowa pozwalająca przeprowadzać badania oraz pomiary. Dalszym krokiem były testy różnych metod i algorytmów, na podstawie których wyciągano wnioski i modyfikowano opracowywany proces oraz jego założenia. Na koniec zweryfikowane zostało działanie stworzonego procesu w rzeczywistych warunkach.

Na początku wykonany został przegląd dostępnych i znanych rozwiązań prowadzących do rozpoznawania szczytów górskich na obrazie (rozdz. 2). Na ich podstawie przygotowano wstępny proces zawierający elementy potrzebne do prawidłowej klasyfikacji widocznych na zdjęciach gór. Opisany został on w rozdziale 2.6. Etapami tego procesu były: generowanie trójwymiarowego modelu ziemi na podstawie numerycznych modeli terenu, ustalenie widocznych na nim szczytów, detekcja krawędzi na takiej wizualizacji oraz na rzeczywistym zdjęciu gór, a także dopasowanie obu zdjęć binarnych do siebie.

Przed przystąpieniem do implementacji i badania oraz prób optymalizacji poszczególnych etapów procesu identyfikacji szczytów przygotowano zestaw zdjęć statycznych będących zbiorem testowym (rozdz. 4). Do tego celu wykonano prostą aplikację działającą na urządzeniach mobilnych. Podczas wykonywania zdjęć zapisywała ona również dodatkowe dane, w szczególności geolokalizacyjne. Dzięki nim możliwe było statyczne testowanie algorytmów związanych z pracą dyplomową bez konieczności analizy całych nagrań i fizycznego przebywania w regionach górskich.

W rozdziale 2.4 opisano czym są numeryczne modele terenu SRTM. Natomiast, opis generowania trójwymiarowego modelu z wykorzystaniem tych danych za pomocą interfejsu OpenGL został zawarty w rozdziale 5.1. Zbiór SRTM występuje w różnych rozdzielczościach, a wybór odpowiedniej ma znaczenie w kontekście dokładności odwzorowania modelu. Porównanie dwóch rozmiarów SRTM oraz ich wpływ na szczegółowość modelu i czas jego generowania przeprowadzono w rozdziale 7.1. Dodatkowo podjęte zostały próby optymalizacji generowania modelu oraz wykorzystania pamięci urządzenia poprzez odrzucenie niepotrzebnych dla danego zadania wpisów w zbiorze SRTM (rozdz. 7.2).

Proces ustalania widocznych na wygenerowanym modelu szczytach górskich został oparty na danych pochodzących z serwisu GeoNames. Dane te zostały przybliżone w rozdziale 2.5. Dla każdej klatki filmu poddawane są ona procesowi filtracji. Opis tego dwuetapowego cyklu zawarty został w rozdziale 5.2. Wykorzystuje on techniki usuwania niewidocznych powierzchni w procesie renderingu. Są to kolejno test frustum culling (rozdz. 5.2.1) sprawdzający czy dany punkt znajduje się w polu widzenia kamery, a następnie occlusion culling (rozdz. 5.2.2) określający czy jest on zasłaniany przez inne bryły w scenie.

9. Podsumowanie

Jednym z etapów procesu prezentowanego w pracy dyplomowej rozpoznawania szczytów jest rozpoznawanie krawędzi góra. Próby implementacji oraz testowania rozwiązań celem osiągnięcia detekcji wszelkich widocznych pasm górkim doprowadziły do decyzji o uproszczeniu oczekiwanej rezultatu. Przyjął on formę wykrywania jedynie najwyższych krawędzi góra tworzących linię horyzontalną z niebem. Opisane zostało to w rozdziale 5.4.1. Z tego powodu ostatecznie wybrany został algorytm Canny (rozdz. 5.4.2) wspomagany przez metodę wyboru oczekiwanej krawędzi.

Dużo uwagi poświęcono procesowi dopasowania do siebie dwóch obrazów binarnych (rozdz. 5.5) zawierających krawędzie góra - jednego uzyskanego z analizy wyrenderowanego modelu, natomiast drugiego ze zdjęcia rzeczywistego. Przedstawiono w nim analizę literaturoową potencjalnych rozwiązań oraz ich krótki opis. Fragment ten podzielony został na dwa podrozdziały odpowiadające dwóm prezentowanym metodom. Rozdział 5.6 przybliżył algorytm oparty na dopasowaniu wzorca (ang. template matching), w szczególności sposób obliczania podobieństwa dwóch obrazów przy pomocy korelacji krzyżowej lub współczynnika korelacji. Druga z metod, dopasowanie na podstawie cech charakterystycznych (ang. feature matching) opisana została w rozdziale 5.7. W jej przypadku przybliżone zostały wszystkie kolejne etapy wykonywane przez ten algorytm: wykrywanie punktów kluczowych oraz ich deskryptorów (rozdz. 5.7.1), dopasowanie punktów kluczowych z obu obrazów (rozdz. 5.7.2), projekcja położenia punktów kluczowych z wykorzystaniem homografii (rozdz. 5.7.3).

Ze względu na znaczenie prawidłowego dopasowania obrazów zawierających krawędzie danych góra przeprowadzono obszerne testy i analizy opisanych rozwiązań. Badania te zostały zrelacjonowane w rozdziale 7.4. Jednym z testowanych elementów był wpływ operacji morfologicznej dylacji na skuteczność poszczególnych rozwiązań (rozdz. 7.4.1). Kolejnym etapem tych testów było porównanie ze sobą jakości klasyfikatorów opartych na szablonach oraz cechach charakterystycznych. Szczegółowe wyniki oraz wyciągnięte na ich podstawie wnioski zawarte zostały w rozdziale 7.4.2.

Po przeprowadzeniu testów i wyborze optymalnych algorytmów przygotowano całociowy proces potrafiący identyfikować szczyty górskie. Na tak zaimplementowanym cyklu przeprowadzono profilowanie opisane w rozdziale 7.5. Pozwoliło ono zdiagnozować element procesu o kiepskiej wydajności. Udało się go zniwelować poprzez bardziej optymalne pobieranie potrzebnych danych z pamięci oraz wykorzystanie pamięci podręcznej procesora.

Finalnie, udało się przebadać i przetestować różne algorytmy oraz wybrać z spośród nich optymalne, które pozwoliły identyfikować szczyty górskie na obrazie w czasie rzeczywistym realizując założony cel pracy magisterskiej. Działanie stworzonego procesu zostało sprawdzone zarówno na statycznych zdjęciach, dla których znane były parametry lokalizacyjne, jak i poprzez analizę obrazu na żywo z kamery urządzenia symulując warunki rzeczywiste (rozdz. 8).

9.2. Zdobyta wiedza i wyciągnięte wnioski

Poświęcony czas na pracę dyplomową pozwolił na poszerzenie wiedzy z zakresu analizy literatury, projektowania oraz wytwarzania oprogramowania, a także przetwarzania cyfrowego obrazów.

Ze względu na świeżość zagadnienia jakim jest identyfikacja szczytów górskich na zdjęciach, liczba opracowań naukowych jest jeszcze w znacznym stopniu ograniczona. Dostępne źródła pozwoliły jednak zgromadzić wystarczającą wiedzę na temat podejścia do tego problemu oraz poszczególnych jego elementów. Przeanalizowane zostały modyfikacje wprowadzane między różnymi pracami zmieniające możliwości oraz złożoność takich systemów. Finalnie przegląd oraz przyswojony materiał pozwoliły na przygotowanie listy potencjalnych algorytmów oraz stworzenie funkcjonalnego oprogramowania tego typu.

Praca nad projektem pozwoliła ugruntować wiedzę na temat grafiki trójwymiarowej oraz sposobu działania biblioteki OpenGL pozwalającej na jej generowanie. Poznano również techniki optymalizacji procesu renderingu poprzez usuwanie powierzchni niewidocznych. W pracy dyplomowej były wykorzystane do filtrowania niewidocznych wierzchołków modelu odpowiadającym szczytom górkim. Znaczna część pracy magisterskiej poświęcona była zagadnieniu dopasowania obrazów. Dzięki temu poznano techniki rejestracji oraz wyrównywania zdjęć, które pozwalają porównywać ze sobą obrazy za pomocą szablonów (metoda template matching) czy cech charakterystycznych (metoda feature matching). Zapoznano się również z algorytmami obliczającymi odległości między dwoma punktami geograficznymi.

Mimo, że główną dziedziną niniejszej pracy dyplomowej jest informatyka to w jej ramach poruszane są również zagadnienia związane z geografią czy geoinformatyką. Z tego względu udało się poszerzyć zasób informacji z nimi powiązanymi. Poznano wiele nazw gór, ich położenie czy wysokość nad poziomem morza, w szczególności występujące w polskich pasmach górskich. Ale również aspekty dotyczące siatki geograficznej oraz jej rozmiarem w zależności od położenia. Poznano także numeryczne zbiory danych dotyczące topograficznej mapy Ziemi oraz graficzne sposoby ich interpretacji. Zapoznano się również z kilkoma metodami pozwalającymi estymować odległość między dwoma punktami geograficznymi.

Zagadnienie rozpoznawania szczytów górskich przy pomocy algorytmów opartych czysto na metodach przetwarzania obrazów pozwala wyciągnąć wnioski, że mimo dynamicznie rozwijającej się gałęzi sztucznej inteligencji do podobnych celów, to niektóre problemy dalej mogą być rozwiązywane podejściem klasycznym. Nawet, jeśli takie podejście może być związane z większą złożonością, w szczególności w aspektach dobierania prawidłowego schematu i algorytmów. Osiągnięta możliwość rozpoznawania szczytów górskich w czasie rzeczywistym na urządzeniach mobilnych pokazuje, że dzisiejsza technologia sprzętowa jest już na wystarczającym poziomie by móc obsłużyć zadania tego typu.

9.3. Możliwość rozwoju i dalszych badań

Opracowany w ramach pracy dyplomowej proces na podstawie przeprowadzonych badań i prób optymalizacji algorytmów pozostawia możliwość dalszego rozwoju oraz usprawnień. Stworzona do celów testowych platforma może być punktem wyjścia do dalszych prac badawczych. Dzięki swojej modułowej architekturze możliwe jest dodatnie kolejnych metod do procesu, wykorzystanie sieci neuronowych czy lepsze dostrojenie parametrów opisanych wyżej algorytmów. Następnie dla takich zmian w łatwy sposób mogą być przeprowadzone testy. Dodatkowe widoki, poszerzające użytkownictwo platformy testowej pozwalają na żywo sprawdzać wpływ różnych parametrów, na przykład progowania algorytmu Canny, co może pomóc w wyborze optymalnych wartości takich danych i badać ich znaczenie w zależności od różnych czynników zewnętrznych.

Konsekwencją decyzji związanej z detekcją krawędzi gór ograniczającej się jedynie do tych, które tworzą linię z niebem był brak możliwości analizy widoczności szczytów będących poniżej tej granicy. Stworzenie rozwiązania umożliwiającego w skutecznny sposób rozpoznawać pozostałe kontury pozwoliłoby lepiej dopasowywać wzorzec widocznych szczytów oraz analizować też niższe wierzchołki. Dobrym pomysłem mogłoby się okazać wykorzystanie do tego celu sztucznej inteligencji, która w ostatnich latach zyskuje popularność w wielu obszarach przetwarzania cyfrowego obrazu.

Możliwe byłoby również poszerzenie możliwości oprogramowania wykorzystującego opracowany proces tak aby mógł rozpoznawać szczyty górskie jedynie na podstawie lokalizacji, bez potrzeby posiadania informacji o kątach odchylenia urządzenia. Wiążałoby się to prawdopodobnie z potrzebą generowania panoramy w postaci sfery 360° , dla której poszukiwano by wzorca szczytu widocznego na zdjęciu. Dzięki temu, klasyfikacja gór byłaby również możliwa na zdjęciach opisanych tylko meta-tagiem o geolokalizacji, na przykład udostępnianych w internecie czy mediach społecznościowych.

Dalsze badania i testowanie przedstawionych w powyższych rozdziałach algorytmów, w szczególności różnych ustawień hiperparametrów może prowadzić do poprawienia ich rezultatów oraz końcowego wyniku rozpoznawania szczytów górskich. Warto również zaimplementować inne niż opisane metody, które mogą okazać się lepszymi w danym zastosowaniu.

Bibliografia

- [1] L. Baboud, M. Cadik, E. Eisemann i H.-P. Seidel, "Automatic photo-to-terrain alignment for the annotation of mountain pictures", lip. 2011, s. 41–48. DOI: 10.1109/CVPR.2011.5995727.
- [2] R. Fedorov, "Mountain Peak Detection in Online Social Media", sierp. 2015.
- [3] R. Fedorov, P. Fraternali i M. Tagliasacchi, "Mountain Peak Identification in Visual Content Based on Coarse Digital Elevation Models", list. 2014, s. 7–11. DOI: 10.1145/2661821.2661825.
- [4] R. Fedorov, D. Frajberg i P. Fraternali, "A Framework for Outdoor Mobile Augmented Reality and Its Application to Mountain Peak Detection", czer. 2016, s. 281–301, ISBN: 978-3-319-40620-6. DOI: 10.1007/978-3-319-40621-3_21.
- [5] G. Baatz, O. Saurer, K. Köser i M. Pollefeys, "Large Scale Visual Geo-Localization of Images in Mountainous Terrain", t. 7573, paź. 2012, s. 517–530, ISBN: 978-3-642-33708-6. DOI: 10.1007/978-3-642-33709-3_37.
- [6] L. Porzi, S. R. Buló, P. Valigi, O. Lanz i E. Ricci, "Learning Contours for Automatic Annotations of Mountains Pictures on a Smartphone", w *Proceedings of the International Conference on Distributed Smart Cameras*, ser. ICDSC '14, New York, NY, USA: Association for Computing Machinery, 2014, ISBN: 9781450329255. DOI: 10.1145/2659021.2659046.
- [7] M. B. Kursa, "Random ferns method implementation for the general-purpose machine learning", *CoRR*, t. abs/1202.1121, 2012.
- [8] P. Lis, *Po co mi rozszerzona rzeczywistość? - Platforma Przemysłu Przyszłości*, Dostęp zdalny (03.03.2019): <https://przemyslprzyszlosci.gov.pl/po-co-mi-rozszerzona-rzeczywistosc/>, 2020.
- [9] D. Georgiev, *10 Pokemon Go Usage and Revenue Statistics Worth Knowing in 2023*, Dostęp zdalny (03.03.2019): <https://techjury.net/blog/pokemon-go-usage-statistics>, 2023.
- [10] I. I. S. B.V., *Ikea App Page - IKEA*, Dostęp zdalny (03.03.2019): <https://www.ikea.com/au/en/customer-service/mobile-apps/say-hej-to-ikea-place-pub1f8af050>, 2023.
- [11] "PeakFinder App", w PeakFinder GmbH, Eglistrasse 8, 8004 Zurich, Switzerland.
- [12] I. e. B. Politecnico di Milano Dipartimento di Elettronica, *PeakLens*, Dostęp zdalny (20.01.2023): <https://www.peaklens.com/>.
- [13] U. Deusdle, *Generate a panorama*, 2018. adr.: https://www.udeuschle.de/panoramas/makepanoramas_en.htm (term. wiz. 10.10.2022).
- [14] P. GmbH, *PeakFinder*, 2023. adr.: <https://www.peakfinder.com/> (term. wiz. 10.10.2022).
- [15] G. U. G. i Kartografii, *Numeryczny model terenu (NMT) - Geoportal Krajowy*, Dostęp zdalny (06.03.2019): <https://www.geoportal.gov.pl/dane/numeryczny-model-terenu>, 2023.

9. Bibliografia

- [16] A. AEROSPACE, *DEM, DSM & DTM: Digital Elevation Model - Why It's Important* - AEVEX, Dostęp zdalny (06.03.2019): <https://geodetics.com/dem-dsm-dtm-digital-elevation-models/>.
- [17] C. I. o. T. Ramirez Eric NASA - Jet Propulsion Laboratory, *Shuttle Radar Topography Mission*, Dostęp zdalny (13.04.2023): <https://www2.jpl.nasa.gov/srtm/missionoverview.html>. adr.: <https://www2.jpl.nasa.gov/srtm/missionoverview.html>.
- [18] VisualizationSoftware.com, *3DEM Visualization Software*, Dostęp zdalny (20.03.2023): <https://www.visualizationsoftware.com/3dem/>, 2013.
- [19] M. W. GeoNames, *GeoNames*, Dostęp zdalny (25.03.2023): <https://www.geonames.org/>.
- [20] CC, *Creative Commons*, Dostęp zdalny (13.06.2023): <https://creativecommons.org/licenses/?lang=pl>.
- [21] T. K. G. Inc., *OpenGL - The Industry Standard for High Performance Graphics*. adr.: <https://www.opengl.org/> (term. wiz. 30. 10. 2021).
- [22] K. G. Inc., *OpenGL ES Overview*, Dostęp zdalny (10.06.2023): <https://www.khronos.org/api/opengles>, 2023.
- [23] K. G. Inc., *EGL Overview*, Dostęp zdalny (10.06.2023): <https://www.khronos.org/egl>, 2023.
- [24] OpenCV. "Home: OpenCV". (), adr.: <https://opencv.org/> (term. wiz. 30. 10. 2021).
- [25] Mutligps, *Co to jest system GPS, kiedy powstaje i gdzie jest stosowany*. Dostęp zdalny (03.03.2019): <https://www.multigps.pl/gps.html>, 2023.
- [26] *Build location-aware apps | Android Development*, Dostęp zdalny (03.03.2019): <https://developer.android.com/training/location>, 2023.
- [27] *3.2: Working with sensor-based orientation*, Dostęp zdalny (03.03.2019): <https://google-developer-training.github.io/android-developer-advanced-course-practicals/unit-1-expand-the-user-experience/lesson-3-sensors/3-2-p-working-with-sensor-based-orientation/3-2-p-working-with-sensor-based-orientation.html>, 2023.
- [28] *WGS 84 - WGS84 - World Geodetic System 1984, used in GPS*, Dostęp zdalny (12.01.2023): <https://epsg.io/4326>, 2022.
- [29] S. Jonathan, *LearnOpenGL - Frustum Culling*, Dostęp zdalny (14.04.2023): <https://learnopengl.com/Guest-Articles/2021/Scene/Frustum-Culling>, 2021.
- [30] opengl-tutorial, *Tutorial 3 : Matrices*, Dostęp zdalny (14.04.2023): <https://www.opengl-tutorial.org/beginners-tutorials/tutorial-3-matrices/>.
- [31] NVidia, *Chapter 29. Efficient Occlusion Culling*, Dostęp zdalny (14.04.2023): <https://developer.nvidia.com/gpugems/gpugems/part-v-performance-and-practicalities/chapter-29-efficient-occlusion-culling>, 2004.
- [32] T. K. G. Inc., *Query Object*, Dostęp zdalny (14.04.2023): https://www.khronos.org/opengl/wiki/Query_Object, 2019.

- [33] WikiPedia, *Promień Ziemi*, 2022. adr.: https://pl.wikipedia.org/wiki/Promień_Ziemi (term. wiz. 15.07.2023).
- [34] *Calculate distance and bearing between two Latitude/Longitude points*, Dostęp zdalny (29.03.2023): <https://www.movable-type.co.uk/scripts/latlong.html>.
- [35] D. Prasetya, P. Nguyen, R. Faizullin, I. Iswanto i E. Armay, "Resolving the shortest path problem using the haversine algorithm", *Journal of Critical Reviews*, t. 7, nr. 1, s. 62–64, sty. 2020, ISSN: 2394-5125. DOI: 10.22159/jcr.07.01.11.
- [36] T. Vincenty, "DIRECT AND INVERSE SOLUTIONS OF GEODESICS ON THE ELLIPSOID WITH APPLICATION OF NESTED EQUATIONS", *Survey Review*, t. 23, nr. 176, s. 88–93, 1975. DOI: 10.1179/sre.1975.23.176.88.
- [37] J. Canny, "A Computational Approach to Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, t. PAMI-8, nr. 6, s. 679–698, 1986. DOI: 10.1109/TPAMI.1986.4767851.
- [38] J. M. Blackledge, "Chapter 16 - Segmentation and Edge Detection", w *Digital Image Processing*, ser. Woodhead Publishing Series in Electronic and Optical Materials, J. M. Blackledge, red., Woodhead Publishing, 2005, s. 487–511, ISBN: 978-1-898563-49-5. DOI: <https://doi.org/10.1533/9780857099464.4.486>. adr.: <https://www.sciencedirect.com/science/article/pii/B9781898563495500161>.
- [39] OpenCV, *OpenCV: Canny Edge Detection*, Dostęp zdalny (13.06.2023): https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html.
- [40] w3.org, *Gaussian Blur*, Dostęp zdalny (11.06.2023): <https://www.w3.org/Talks/2012/0125-HTML-Tehran/Gaussian.xhtml>.
- [41] O. Vincent i O. Folorunso, "A Descriptive Algorithm for Sobel Image Edge Detection", sty. 2009. DOI: 10.28945/3351.
- [42] J. Hosang, R. Benenson i B. Schiele, *Learning non-maximum suppression*, 2017. arXiv: 1705.02950 [cs.CV].
- [43] G. Bebis, "Wykłady w ramach CS491E/791E: Computer Vision (Spring 2004)", Computer Science Department, Computer Science & Engineering, University of Nevada, Reno, 2004.
- [44] OpenCV, *OpenCV: Template Matching*, 2023. adr.: https://docs.opencv.org/3.4/de/da9/tutorial_template_matching.html (term. wiz. 10.06.2023).
- [45] OpenCV, *OpenCV: Template Matching*, 2023. adr.: https://docs.opencv.org/3.4/d4/dc6/tutorial_py_template_matching.html (term. wiz. 10.06.2023).
- [46] A. Vision, *Template Matching*, 2012. adr.: https://docs.adaptive-vision.com/4.7/studio/machine_vision_guide/TemplateMatching.html (term. wiz. 10.06.2023).
- [47] B. Woodham, J. Little, F. Tung i K. Grauman, "Computer Vision. Lecture 6: Template Matching", The University of British Columbia.
- [48] R. Caubalej, "Image Processing - Template Matching", 2021.

9. Bibliografia

- [49] OpenCV, *OpenCV: Image Registration*, 2023. adr.: https://docs.opencv.org/4.x/db/d61/group__reg.html (term. wiz. 05. 08. 2023).
- [50] OpenCV, *OpenCV: Understanding Features*, 2023. adr.: https://docs.opencv.org/4.x/df/d54/tutorial_py_features_meaning.html (term. wiz. 26. 06. 2023).
- [51] A. Lieberman, *Computer Vision Project - Project 2: Local Feature Matching*. adr.: https://www.cc.gatech.edu/classes/AY2016/cs4476_fall/results/proj2/html/alieberman3/index.html#motivation (term. wiz. 26. 06. 2023).
- [52] R. Szeliski. 2006.
- [53] GeeksforGeeks, *Image Registration using OpenCV*. adr.: <https://www.geeksforgeeks.org/image-registration-using-opencv-python/> (term. wiz. 05. 08. 2023).
- [54] A. Rosebrock, *Image alignment and registration with OpenCV*, 2020. adr.: <https://pyimagesearch.com/2020/08/31/image-alignment-and-registration-with-opencv/> (term. wiz. 05. 08. 2023).
- [55] A. Naskar, *Practical Image Registration and Alignment with OpenCV and Python*, 2023. adr.: <https://thinkinfi.com/image-alignment-and-registration-with-opencv/> (term. wiz. 05. 08. 2023).
- [56] K. Mikolajczyk i C. Schmid, "An Affine Invariant Interest Point Detector", t. 1, mar. 2002, ISBN: 978-3-540-43745-1. DOI: 10.1007/3-540-47969-4_9.
- [57] OpenCV, *CS4670/5670: Project 2: Feature Detection and Matching*, 2012. adr.: <https://www.cs.cornell.edu/courses/cs4670/2012fa/projects/p2/project2.html> (term. wiz. 26. 06. 2023).
- [58] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *Int. J. Comput. Vision*, t. 60, nr. 2, s. 91–110, list. 2004, ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000029664.99615.94. adr.: <http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94>.
- [59] OpenCV, *OpenCV: Introduction to SIFT (Scale-Invariant Feature Transform)*, 2023. adr.: https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html (term. wiz. 21. 06. 2023).
- [60] E. Rublee, V. Rabaud, K. Konolige i G. Bradski, "ORB: An efficient alternative to SIFT or SURF", w *2011 International Conference on Computer Vision*, 2011, s. 2564–2571. DOI: 10.1109/ICCV.2011.6126544.
- [61] OpenCV, *OpenCV: ORB (Oriented FAST and Rotated BRIEF)*, 2023. adr.: https://docs.opencv.org/4.x/d1/d89/tutorial_py_orb.html (term. wiz. 21. 06. 2023).
- [62] E. Rosten i T. Drummond, "Machine Learning for High-Speed Corner Detection", t. 3951, lip. 2006, ISBN: 978-3-540-33832-1. DOI: 10.1007/11744023_34.
- [63] M. Calonder, V. Lepetit, C. Strecha i P. Fua, "BRIEF: Binary Robust Independent Elementary Features", t. 6314, wrz. 2010, s. 778–792, ISBN: 978-3-642-15560-4. DOI: 10.1007/978-3-642-15561-1_56.
- [64] OpenCV, *OpenCV: Feature Matching*, 2023. adr.: https://docs.opencv.org/4.x/dc/dc3/tutorial_py_matcher.html (term. wiz. 26. 06. 2023).

- [65] T. V. Haavardsholm, *Lecture 4.2 Feature matching*. adr.: https://www.uio.no/studier/emner/matnat/its/TEK5030/v19/lect/lecture_4_2_feature_matching.pdf (term. wiz. 26.06.2023).
- [66] M. Muja i D. G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration", w *International Conference on Computer Vision Theory and Applications*, 2009. adr.: <https://api.semanticscholar.org/CorpusID:7317448>.
- [67] M. Muja i D. G. Lowe, "FLANN - Fast Library for Approximate Nearest Neighbors. User Manual.", 2009.
- [68] Aids, *Introduction to Image Processing - Part 7: Homography*, 2021. adr.: <https://perez-aids.medium.com/introduction-to-image-processing-part-7-homography-4af34ce9f93d> (term. wiz. 30.06.2023).
- [69] OpenCV, *OpenCV: Basic concepts of the homography explained with code*, 2023. adr.: https://docs.opencv.org/4.x/d9/dab/tutorial_homography.html (term. wiz. 30.06.2023).
- [70] M. Maulion, *Homography Transform - Image Processing*, 2021. adr.: <https://mattmaulion.medium.com/homography-transform-image-processing-eddbcb8e4ff7> (term. wiz. 30.06.2023).
- [71] OpenCV, *OpenCV: Feature Matching + Homography to find Objects*, 2023. adr.: https://docs.opencv.org/4.x/d1/de0/tutorial_py_feature_homography.html (term. wiz. 30.06.2023).
- [72] M. A. Fischler i R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", w *Readings in Computer Vision*, M. A. Fischler i O. Firschein, red., San Francisco (CA): Morgan Kaufmann, 1987, s. 726–740, ISBN: 978-0-08-051581-6. DOI: <https://doi.org/10.1016/B978-0-08-051581-6.50070-2>. adr.: <https://www.sciencedirect.com/science/article/pii/B9780080515816500702>.
- [73] O. Chum i J. Matas, "Matching with PROSAC - progressive sample consensus", w *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, t. 1, 2005, 220–226 vol. 1. DOI: [10.1109/CVPR.2005.221](https://doi.org/10.1109/CVPR.2005.221).
- [74] S. K. Agrawal, *Metrics to Evaluate your Classification Model to take the right decisions*, 2021. adr.: <https://towardsdatascience.com/understanding-morphological-image-processing-and-its-operations-7bcf1ed11756> (term. wiz. 15.04.2023).
- [75] P. Chhikara, *Understading Morphological Image Processing and Its Operations*, 2022. adr.: <https://towardsdatascience.com/understanding-morphological-image-processing-and-its-operations-7bcf1ed11756> (term. wiz. 12.04.2023).
- [76] OpenCV, *OpenCV: cv::Mat Class Reference*. adr.: https://docs.opencv.org/3.4/d3/d63/classcv_1_1Mat.html (term. wiz. 01.08.2023).
- [77] B. Andrist i V. Sehr, "Wyosce wydajny C++. Opanuj sztukę optymalizowania działania kodu.", w Helion S.A., 2023, rozd. 4. Struktury danych, s. 107–111.
- [78] S. Slotin, "Algorithms for Modern Hardware.", w 2023, rozd. 4.1 Memory Hierarchy.

Spis rysunków

2.1 Rzut pionowy danych SRTM 49-50N, 20-21E wygenerowany z wykorzystaniem oprogramowania 3DEM.	15
2.2 Teoretyczne etapy procesu identyfikacji szczytów górskich na obrazie.	17
4.1 Przykładowe zdjęcie gór i terenu zebrane w paśmie górkim Tatr.	22
5.1 Przepływ danych i sterowania procesu identyfikacji szczytów górskich z uwzględnieniem sensorów i urządzeń.	24
5.2 Wizualizacja tworzenia siatki trójkątów między punktami ze zbioru SRTM. .	25
5.3 Przykład wizualizacji terenu z wykorzystaniem interfejsu OpenGL i danych SRTM.	26
5.4 Wizualizacja frustum culling. Ilustracja stożka widoczności kamery przy projekcji perspektywicznej. Odwzorowane zostaną tylko obiekty znajdujące się w przestrzeni stożka ściegego - zacieniona część. Czerwone koła będące poza obszarem widzenia kamery zostaną odrzucone.	27
5.5 Wizualizacja Occlusion Culling. Czerwone koło jest przesłaniane przez zielony prostokąt, więc zostaje usunięte z procesu renderingu.	28
5.6 Przykładowy rezultat identyfikacji szczytów górskich na trójwymiarowym modelu terenu.	29
5.7 Przykład wstępnie oczekiwanej rezultatu detekcji krawędzi gór.	31
5.8 Przykład uproszczonej detekcji krawędzi gór.	32
5.9 Przykład mapy podobieństwa uzyskanej wykorzystując algorytm dopasowania wzorca przy porównaniu obrazów binarnych krawędzi gór.	35
5.10 Przykład części wykrytych punktów charakterystycznych wraz z ich kierunkiem i rozmiarem na obrazie binarnym krawędzi gór.	38
5.11 Przykład dopasowanych punktów kluczowych dwóch obrazów binarnych krawędzi gór.	40
5.12 Przykład transformacji homograficznej na podstawie dopasowanych punktów kluczowych celem oznaczenia położenia szukanego szablonu.	41
7.1 Porównanie generowanego terenu w zależności od rozdzielczości danych. Górnny wyrenderowany model prezentuje dokładność 30 m, dolny natomiast 90 m.	46
7.2 Wizualizacja usuwania niepotrzebnych danych SRTM.	48
7.3 Przykłady nałożenia artefaktów zasłaniających część szczytów w wybranych przypadkach testowych.	52
7.4 Przykład obliczonego podobieństwa dla poszczególnych przypadków przy użyciu algorytmu dopasowania szablonu. Na ich podstawie ustalany był próg wartości dla danych ustawień parametrów.	54
7.5 Dopuszczalny błąd dopasowania w zależności od wartości współczynnika tolerancji K dla wybranego szczytu.	56

7.6 Przykład wpływu dylacji na wyniki algorytmu dopasowania obrazów na podstawie szablonów przy przykładowych ustawieniach parametrów. Górnny wykres przedstawia wyniki bez wykorzystania dylacji, dolny natomiast z wykorzystaniem dylacji.	57
7.7 Profilowanie procesu identyfikacji szczytów górskich przed optymalizacją wykrywania krawędzi najwyższych gór.	67
7.8 Profilowanie procesu identyfikacji szczytów górskich po optymalizacji wykrywania krawędzi najwyższych gór poprzez zmianę kolejności analizy pikseli i wykorzystaniem pamięci podręcznej.	68
8.1 Przykładowy rezultat rozpoznawania szczytów górskich na obrazie.	69

Spis tabel

4.1 Zebrane dane dotyczące przykładowego zdjęcia w paśmie górskim Tatr.	23
7.1 Porównanie czasu generowania wierzchołków i trójkątów w zależności od rozdzielczości danych SRTM.	47
7.2 Porównanie pamięci zajmowanej przez wierzchołki i trójkąty w zależności od rozdzielczości danych SRTM.	47
7.3 Porównanie czasu generowania wierzchołków i trójkątów przy uproszczeniu i braku uproszczenia danych.	49
7.4 Porównanie dokładności wybranych algorytmów obliczających odległość między dwoma punktami geograficznymi.	50
7.5 Porównanie czasu wykonania wybranych algorytmów obliczających odległość między dwoma punktami geograficznymi.	51
7.6 Porównanie skuteczności algorytmów dopasowania obrazów na podstawie szablonu oraz na podstawie cech przy przykładowych ustawieniach parametrów tych metod z wykorzystaniem operacji dylacji oraz bez.	58
7.7 Wpływ operacji morfologicznej dylacji na algorytm dopasowania na podstawie szablonu.	60
7.8 Wpływ operacji morfologicznej dylacji na algorytm dopasowania na podstawie cech.	61
7.9 Porównanie skuteczności algorytmów dopasowania obrazów na podstawie szablonu oraz na podstawie cech przy dopuszczalnym błędzie $K = 0,05$	62
7.10 Porównanie skuteczności algorytmów dopasowania obrazów na podstawie szablonu oraz na podstawie cech przy dopuszczalnym błędzie $K = 0,01$	63
7.11 Porównanie wartości wskaźnika precyzyji przy założeniu minimalnej liczby prawdziwie pozytywnych przypadków dla algorytmów dopasowania obrazów na podstawie szablonu oraz na podstawie cech, gdy dopuszczalny błąd $K = 0,05$ i $K = 0,01$	64
8.1 Czas przetwarzania procesu rozpoznawania szczytów górskich na obrazie w warunkach rzeczywistych.	70

Spis załączników

- | | |
|--|----|
| 1. Repozytorium pracy dyplomowej | 84 |
|--|----|

Załącznik 1. **Repozytorium pracy dyplomowej**

Rozwiążanie projektowe, materiały prezentacyjne oraz wyniki testów zostały umieszczone w repozytorium git autora pracy dyplomowej w serwisie GitHub pod adresem:
<https://github.com/madamskip1/Mountain-Peak-Detection>

W szczególności istotne foldery:

- */projekt/* - kod aplikacji
- */praca_dyplomowa/* - praca dyplomowa w formacie LaTeX oraz skompilowana do postaci dokumentu pdf
- */materiały_prezentacyjne/* - materiał prezentacyjne, w szczególności nagranie pokazujące działanie projektu w czasie rzeczywistym
- */testy_wyniki/* - wyniki poszczególnych testów w formacie csv