AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,
INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ

Symulacja Dyskretna Systemów Złożonych

Symulacja zagrożenia lawinowego w Tatrach Polskich

Autorzy: *Marek Jachym, Wojciech Konieczkowicz*
Kierunek studiów: *Informatyka*
Opiekun pracy: *dr hab. inż. Jarosław Wąs*

Kraków, 2020

Spis treści

1. Przykłady elementów pracy dyplomowej

1.1. Liczba

Pakiet `siunitx` zadba o to, by liczba została poprawnie sformatowana:

$$1\,234\,567\,890{,}098\,765\,432\,1$$

1.2. Rysunek

Pakiet `subcaption` pozwala na umieszczanie w podpisie rysunku odnośników do „podilustracji”:

(a) (b)

Rys. 1.1. Przykład użycia `\subcaption`: (a) litera A, (b) litera B.

1.3. Tabela

Pakiet `threeparttable` umożliwia dodanie do tabeli adnotacji:

Tabela 1.1. Przykład tabeli

Nagłówek[a]
Tekst 1
Tekst 2

[a] Jakiś komentarz...

1.4. Wzory matematyczne

Czasem zachodzi potrzeba wytłumaczenia znaczenia symboli użytych w równaniu. Można to zrobić z użyciem zdefiniowanego na potrzeby niniejszej klasy środowiska `eqwhere`.

$$E = mc^2 \tag{1.1}$$

gdzie

m – masa

c – prędkość światła w próżni

Odległość półpauzy od lewego marginesu należy dobrać pod kątem najdłuższego symbolu (bądź listy symboli) poprzez odpowiednie ustawienie parametru tego środowiska (domyślnie: 2 cm).

2. Wprowadzenie

2.1. Cel

Celem niniejszej pracy jest stworzenie symulacji, która dostarcza w czasie rzeczywistym informacji dotyczących możliwego zagrożenia lawinowego na terenie Tatr w Polsce.

2.2. Opis problemu

Lawiny śnieżne są powszechnie występującym zagrożeniem na całym świecie i stanowią niebezpieczeństwo zarówno dla ludzi, jak i biznesu oraz infrastruktury. Instytucje w różnych państwach alarmują o zagrożeniu lawinowym w podobny sposób, choć można zauważyć, że im większe zagrożenie stanowi zjawisko samoistnego ruchu śniegu, tym bardziej szczegółowe są prognozy.

Głównym problemem przy określaniu ryzyka lawinowego jest złożoność tego zjawiska. Zależy ono zarówno od czynników stałych, do których zalicza się np. ukształtowanie terenu (1. typ uwarunkowań) oraz zmiennych, dotyczących stanu śniegu (2. typ uwarunkowań) oraz warunków meteorologicznych (3. typ uwarunkowań) (Woszczek 2016). Polskie instytucje wydają regularnie tzw. komunikaty lawinowe, są one jednak jedynie ogólnym zapisem zagrożenia - ratownicy zastrzegają, że informacje zawarte w komunikacie stanowią tylko podstawę do samodzielnej oceny.

Przedmiotem poniższej pracy jest więc próba zastosowania technologii, by zapewnić system wspomagający decyzje ekspertów w celu tworzenia jeszcze bardziej precyzyjnych ostrzeżeń. Symulacja sama w sobie nie stanowi profesjonalnego narzędzia, ale zawiera koncepcje i rozwiązania, które można zastosować przy tworzeniu niezawodnego i złożonego narzędzia dla państwowych instytucji.

2.3. Możliwe rozwiązanie

Na świecie modele przewidujące zagrożenie są tworzone w oparciu o metody statystyczne oraz metody uczenia maszynowego takie jak analiza najbliższego sąsiedztwa (ang. nearest neighbor analysis), analiza skupień (ang. cluster analysis), czy drzewa klasyfikacyjne (ang. classification trees) (Joshi, Kumar, Srivastava, Sachdeva, Ganju 2018). Są to jednak rozwiązania stworzone na podstawie wieloletnich

pomiarów (również dotyczących warunków pokrywy śnieżnej), do których nie uzyskano dostępu. Zdecydowano się więc oprzeć na wnioskach autorów publikacji, charakteryzujących najważniejsze czynniki stwarzające ryzyko.

W celu jak najdokładniejszego określania ryzyka wykorzystano bardzo precyzyjne informacje dotyczące ukształtowania terenu oraz dane pogodowe (dane dotyczące śniegu nie są ogólnodostępne i ich zmierzenie wymaga specjalistycznej wiedzy oraz narzędzi).

Korzystając z danych topograficznych uproszczono ukształtowanie powierzchni Tatr (każdy z obszarów o powierzchni ponad 2 km^2 reprezentowany jest przy pomocy około 110 punktów) i obliczono odpowiednie cechy. Następnie w połączeniu z cechami dotyczącymi warunków atmosferycznych możliwe stało się określenie ryzyka dla każdego takiego obszaru przy pomocy stworzonego wcześniej drzewa decyzyjnego. Dzięki takiemu podejściu, cały obszar Tatr Polskich podlega obserwacji, a w razie wystąpienia sprzyjających warunków (wykorzystano 2 z 3 istniejących uwarunkowań) w sposób zautomatyzowany wydaje się odpowiednie ostrzeżenia.

3. Stworzony model zjawiska

Niniejszy rozdział opisuje szczegółowo kolejne kroki oraz wykorzystane algorytmy niezbędne do uzyskania efektu końcowego.

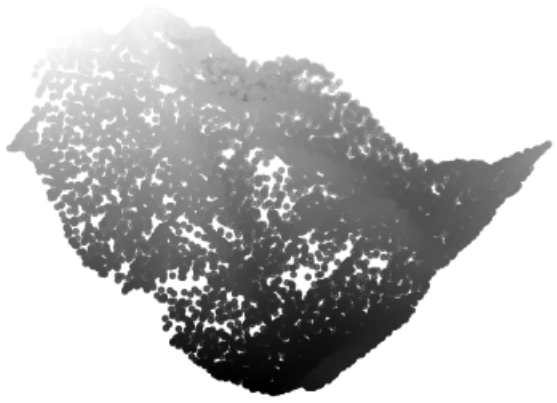
3.1. Opracowanie danych topograficznych

3.1.1. Format .las

Na początkowym etapie pracy otrzymano pliki w formacie .las - każdy z nich reprezentujący ukształtowanie terenu wybranego obszaru.

Format .las został stworzony do przechowywania zbioru punktów w przestrzeni trójwymiarowej (ang. point cloud), otrzymanych przy pomocy metody Lidar, która polega na oświetlaniu wybranych punktów na powierzchni Ziemi laserem i zapisie jego odbicia przy pomocy sensorów. Dzięki tej metodzie powstają mapy o wysokiej rozdzielczości, stosowane w naukach o Ziemi (źródło: angielska wiki).

Otrzymane pliki zawierały średnio około 11 milionów punktów, przy czym każdy z nich reprezentował powierzchnię ponad 2 km^2.



Rys. 3.1. Uproszczona wizualizacja zbioru punktów z pliku .las przy użyciu biblioteki matplotlib

3.1.2. Wybór cech

Jak wspomniano już wcześniej, do oszacowania ryzyka lawinowego konieczne jest posiadanie danych dotyczących cech terenu. Bazując na pracy pani Izabeli Woszczak, skupiono się na obliczeniu taki cech, jak:

– forma terenu (skupiono się na żlebach);

– ekspozycja słoneczna;

– nachylenie powierzchni;

– piętro;

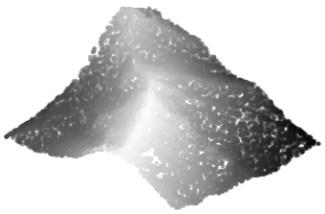
– wysokość.

3.1.3. Triangulacja Delaunaya

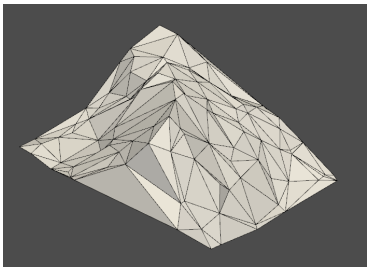
Sam zbiór punktów nie oferuje możliwości łatwego obliczania wyżej wymienionych cech, dlatego zastosowano uproszczenie powierzchni terenu przy pomocy triangulacji Delaunaya.

Jest to algorytm, który na podstawie zbioru punktów tworzy zbiór trójkątów, gdzie wierzchołki każdego trójkąta stanowią owe punkty. Własnością algorytmu jest, że maksymalizuje on najmniejsze z katów w powstałych trójkątach, unikając tzw. sliver triangles.

TE ZDJĘCIA POWINNY BYĆ DALEJ.



Rys. 3.2. Obszar przedstawiony jako zbiór punktów



Rys. 3.3. Ten samo obszar poddany triangulacji

Plik LaTeXowy jest plikiem tekstowym, który oprócz tekstu zawiera polecenia formatujące ten tekst (analogicznie do języka HTML). Plik składa się z dwóch części:

1. Preambuły – określającej klasę dokumentu oraz zawierającej m.in. polecenia dołączającej dodatkowe pakiety;

2. Części głównej – zawierającej zasadniczą treść dokumentu.

```
\documentclass[a4paper,12pt]{article}      % preambuła
\usepackage[polish]{babel}
```

```
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{times}

\begin{document}                          % część główna

\section{Sztuczne życie}

% treść
% ąśężźćńłóĘŚĄŻŹĆŃÓŁ

\end{document}
```

Nie ma żadnych przeciwskazań do tworzenia dokumentów w LaTeXu w języku polskim. Plik źródłowy jest zwykłym plikiem tekstowym i do jego przygotowania można użyć dowolnego edytora tekstów, a polskie znaki wprowadzać używając prawego klawisza `Alt`. Jeżeli po kompilacji dokumentu polskie znaki nie są wyświetlane poprawnie, to na 95% źle określono sposób kodowania znaków (należy zmienić opcje wykorzystywanych pakietów).

3.2. Opracowanie danych pogodowych

Dane pogodowe potrzebne do symulacji zagrożenia

3.3. Narzędzia

3.4. Przygotowanie dokumentu

Plik źródłowy LaTeXa jest zwykłym plikiem tekstowym. Przygotowując plik źródłowy warto wiedzieć o kilku szczegółach:

– Poszczególne słowa oddzielamy spacjami, przy czym ilość spacji nie ma znaczenia. Po kompilacji wielokrotne spacje i tak będą wyglądały jak pojedyncza spacja. Aby uzyskać *twardą spację*, zamiast znaku spacji należy użyć znaku *tyldy*.

– Znakiem końca akapitu jest pusta linia (ilość pusty linii nie ma znaczenia), a nie znaki przejścia do nowej linii.

– LaTeX sam formatuje tekst. **Nie starajmy się go poprawiać**, chyba, że naprawdę wiemy co robimy.

4. Symulacja zjawiska - implementacja i szczegóły techniczne

4.1. Wybór języka programowania

Językiem który postanowiliśmy wybrać do implementacji symulacji przedstawionego powyżej problemu wybrano język Python w wersji **3.7.x**.

Ze względu na jego prostotę, wydajność i obszerny wybór bibliotek zewnętrznych wybraliśmy zamiast innych, również bardzo wydajnych ale bardziej skomplikowanych języków takich jak C++ czy Java.

4.2. Narzędzia wykorzystywane w trakcie tworzenia projektu

4.2.1. Środowisko programowania

Rys. 4.1. PyCharm Community 2019

Wykorzystano **PyCharm Community 2019.3**, rozbudowane, potężne i wyposażone w dużą ilość dodatkowych narzędzi środowisko stworzone do pracy z językiem Python. Dostępne jest w dwóch wersjach, Professional oraz Community będąca wersją open-source przez co idealnie nadała się do wykorzystania przy naszym projekcie.

4.2.2. Wybrane biblioteki zewnętrzne języka Python

Jak wspomniano wcześniej, Python jest bardzo popularnym językiem przez co posiada wiele bibliotek zewnętrznych.

Do implementacji symulacji naszego problemu wykorzystaliśmy następujące biblioteki zewnętrzne:

1. **PyQt5** - oferująca zestaw narzędzi do budowy graficznego interfejsu użytkownika
2. **pyowm** - oferująca zestaw narzędzi do eksploatacji API Open Weather Map
3. **laspy** - zestaw funkcji umożliwiająca wykonywanie wielu operacji na plikach .las
4. **paramiko** - moduł umożliwiający komunikację SSH z poziomu skryptu w Pythonie
5. **pyvista** - biblioteka zawierająca implementację algorytmu triangulacji Delaunaya

4.2.3. Usługi chmurowe Google Cloud Platform i dane pogodowe

Kluczowym, z punktu widzenia naszego projektu, okazało się znalezienie rozwiązania problemu ciągłej aktualizacji danych pogodowych. Z powodu iż dane muszą być nieustannie aktualizowane co 8 godzin, jedna z naszych maszyn roboczych musiałaby pracować bez przerwy. Nie jest to w żadnym stopniu rozwiązanie optymalne. Zdecydowaliśmy się skorzystać z darmowego okresu próbnego oferowanego na platformie **Google Cloud Platform**. Oferta Google zawarła w sobie możliwość stworzenia instancji maszyny wirtualnej operującej na systemie **Debian 10 Buster** co otworzyło nam drogę do rozwiązania problemu ciągłej aktualizacji danych pogodowych.

Dane pogodowe pobierane są co 8 godzin, o godzinach 00:00, 08:00 i 16:00 przy pomocy skryptu **weather_conditions.py** który korzysta z biblioteki **pyowm**

weather_conditions.py

```
#!/usr/bin/python2.7
import pyowm
import os
import time
import tarfile
#ZMIENNA path REPREZENTUJE ODPOWIEDNIE ŚCIEŻKI DO PLIKÓW
def get_weather_conditions(map_name):
# UTWÓRZ ODPOWIEDNI FOLDER JEŻELI NIE ISTNIEJE
if os.path.isdir(path):
        pass
else:
        path = path
os.mkdir(path)
# WSPÓŁRZĘDNE ODPOWIEDNIEGO OBSZARU I CZAS POBRANIA DANYCH
coords = extract_coords(map_name)
czas_pomiaru = str(time.ctime()).split("␣")
# POBRANIE DANYCH POGODOWYCH
owm = pyowm.OWM("7202a85833f71127c0a0b4fefc86ea2a")
observation = owm.weather_around_coords(float(coords["N_lat"]), float(coords["W_long"]))

# USUWANIE NAJSTARSZEGO POMIARU I ZASTĄPIENIE GO NOWYM
data = os.listdir(path)
data.sort()
if len(data) >= 6:
        os.remove(path)
# STWÓRZ PLIK ZAWIERAJĄCY AKTUALNE DANE POGODOWE
filename = str(observation[0].get_weather().get_reference_time('date'))[0:10]
data = open(path, "w+")
data.write(str(observation[0].get_weather().get_temperature('celsius')['temp']) + "\n")
data.write(str(observation[0].get_weather().get_snow()) + "\n")
data.write(str(observation[0].get_weather().get_wind('meters_sec')) + "\n")
data.write(str(observation[0].get_weather().get_rain()) + "\n")

with open(path) as file:
        file.readline()
        for line in file:
                get_weather_conditions(line.rstrip()[0:16])
```

4.2.3.1. Harmonogram pobrania danych

Za odpowiedni czas pobrania danych odpowiada narzędzie uniksopodobnych systemów operacyjnych **cron**. Jest to realizowane za pomocą odpowietnich wpisów w pliku **crontab**

```
0 6 * * * /usr/bin/python /home/marekjachym99/lavalanche/weather_conditions.py

0 14 * * * /usr/bin/python /home/marekjachym99/lavalanche/weather_conditions.py

0 22 * * * /usr/bin/python /home/marekjachym99/lavalanche/weather_conditions.py
```

Bibliografia

[1] A. Diller. *LaTeX wiersz po wierszu*. Gliwice: Wydawnictwo Helion, 2000.

[2] L. Lamport. *LaTeX system przygotowywania dokumentów*. Kraków: Wydawnictwo Ariel, 1992.

[3] M. Szpyrka. *On Line Alvis Manual*. `http://fm.ia.agh.edu.pl/alvis:manual`. AGH University of Science and Technology. 2011.