

Entwicklung einer neuen Messungsmethode für Schneefeuchtigkeit

Bachelorarbeit
Bachelor für Maschinentechnik | Innovation

Betreuung:
Albert Loichinger

Unterstützung Team IPEK:
(Christian Locher)

FS 2024
Abgabedatum: 2024.

Autor: Peter Kuhn



Abstract

problem

vorstudien

ergebniss von Funktionsmuster

Beschreibung der Abkürzungen

Schneefeuchtigkeit Liquid Water Content, LWC IPEK Institut SLF BA

Inhaltsverzeichnis

1	Einleitung	1
1.1	Lawinen in der Schweiz	1
1.2	Entstehung der Gleitlawine	1
1.3	Endziel des Arbeit	1
1.4	User Story	1
1.5	Anforderungen	1
1.6	Planung der Arbeit	1
2	Liquid Water Content	2
2.1	Physicalische Prinzipien	2
2.2	Kommerzielle Produkte	2
2.3	Publizierte Methoden	2
3	Vorstudie	2
3.1	3M 5559 Water Indikator Tape	2
3.2	Voltcraft	3
3.3	Laser Refraktion, Reflexion	4
3.4	Vibration	6
3.5	Diffusion von Flüssigkeit	6
4	Funktionsmuster	6
4.1	Funktionsweise	6
4.2	Bildverarbeitung	6
4.3	Extrahieren von Informationen aus Bilddaten	8
4.3.1	Anforderungsanalyse	8
4.3.2	Konzeptueller DB Entwurf	9
4.3.3	Logischer DB Entwurf	10
4.3.4	Ansichten für den Analysten	10
4.3.5	Physischer Entwurf	10

4.3.6	Python-Interaktion mit der Datenbank	10
4.3.7	Nächste Schritte	11
4.3.8	Code	11
4.4	Testkriterien	22
4.5	Widerstand gegen Umwelteinflüsse	22
4.6	Montage des Funktionsmusters	22
4.7	Ergebnisse der Versuche	22
4.8	Vergleich der Ergebnisse mit Denometer	22
4.9	Verbesserungsmöglichkeiten des Funktionsmusters	22
5	Ausblick	23
5.1	Presönliche Erfahrung	23
5.2	Fazit	23
5.3	Ausblick	23
6	Literaturverzeichnis	24
7	Erklärung zur Urheberschaft	25
8	Digitaler Anhang	28

1 Einleitung

bachelorarbeit produktentwicklung grundlagenforschung seit 40 jahren forschungsgebiet, da für simulation wichtig. fail ist ein gutes ergebniss

1.1 Lawinen in der Schweiz

jedes jahr 10 Tote. 8 schneebrettlawine. 2 Gleitlawinen.

mit Klimawandel änders sich Gleitlawinen. nicht preventiv mit einer Detonation auslösbar. nicht zeitlich vorhersagbar.

1.2 Entstehung der Gleitlawine

feuchtigkeit sammelt sich zwischen den Eiskristallen an.

feuchtigkeit kommt durch schmelzenden schnee, primär Radiation und sekundär radiation.

Regen auf schnee

feuchtigkeit aus dem Boden. wasserführende schichten.

1.3 Endziel des Arbeit

verringerung des Schadens durch Gleitlawinen

1.4 User Story

Bob sitzt an seinem Computer und shiet eine Warung aufleuchten. Er ruft sofort bei der Ratischen Bahn an und kann den Zug so stoppen vewor er von der Lawine erfasst wird.

1.5 Anforderungen

Die Methode soll einn anzeige haben, die Feststellen kann wann eine Gleitlawine bevorsteht.

Die Methode soll unabhängig von der Dichte des Schness funktionieren.

die methode soll den messbereich des LWC von 1 % bis 7 % abdecken.

die methode soll für einen Hang in der Schweiz einsetzbar sein.

1.6 Planung der Arbeit

Die Arbeit wird in drei Teile aufgegliedert.

in einer Vorstudie werden unterschiedliche physikalische Prinzipien zur Messung des LWC theoretisch und praktisch miteinander verglichen.

Bau des Funktionsmodells. Hier wird ein vielversprechendes physikalisches Prinzip ausgewählt und ein Funktionsmodell gebaut.

Validierung und Dokumentation der Ergebnisse. Doku schreiben.

2 Liquid Water Content

2.1 Physikalische Prinzipien

2.2 Kommerzielle Produkte

2.3 Publierte Methoden

3 Vorstudie

3.1 3M 5559 Water Indikator Tape

Herkunft: Aus dem Elektronikbereich. Zum Beispiel in Handys. Wenn das Tape rot geworden ist, ist Wasser eingedrungen und der Hersteller kann eine Garantieleistung ablehnen.

Funktionsweise: Das papierbasierte Klebeband wird nass. Die rote Farbe auf der Unterseite des Klebebands blutet durch das weiße obere Papier. Die roten Teile zeigen dann permanent Wasser an.

Auswahl von 5559: Der Hersteller 3M hat mehrere Produkte zu Water Indikator. 5559 zeichnet sich durch die dünnere Dicke und somit durch die schnellere Anzeigegeschwindigkeit aus.

5559i ist auf einem transparenten Substrat, was praktisch für die optische Auswertung wäre. Die Produkte sind in Europa nur teilweise erhältlich. 3M verkauft nur Rollen mit 160 m. Zum Testen wurde eine kleine Rolle von einem Elektronikkomponenten-Vertreiber gekauft.

Bei der Recherche zu LWC wurde keine Verwendung von Water Indicator Tapes bemerkt. Somit neuartig.

kostengünstig

Zeitspanne pro Messung weniger als 60 Sek.

Dichte des Schnees muss separat gemessen werden. 5559 zeigt nur das flüssige Wasser in einer Schicht an.

Testaufbau: 5559 auf etwas rund 200 g schweres Kleben. Neue Oberfläche von Schnee mit Messer abschneiden/freilegen. 5559 auf Schnee legen und 10, 30, 60, 120 Sek warten. Foto von Klebeband machen. Mit Python rote vs. weiße Fläche berechnen. Oder nur optisch beurteilen.

3.2 Voltcraft

die Graphitsonden, zwischen denen die Spannung aufgebaut und der Widerstand gemessen wird, sind im Messkopf zu gut geschützt. Daher kann keine Messung gemacht werden, wenn die Probe in Schnee gedrückt wird.

Mögliche Lösung: Verlängerung der Graphitproben
mit Stahlplatten

Verbindung des Graphits mit der Platte: kleben oder konstant drücken oder verschrauben.

In Graphit spannend zu arbeiten ist anspruchsvoll und dreckig. Konstant drücken ist fehleranfällig. Kleben: herstellen von leitfähigem Klebstoff:

Test Graphitpulver: 66 % Gewichtsprozent Graphitpulver, 33 % Ergo 7410 Epoxy Klebstoff

Test Aluminiumpulver: 66 % Gewichtsprozent Aluminiumpulver, 33 % Ergo 7410 Epoxy Klebstoff

Ergebnis: nach 24 h, sodass der Ergo 7410 aushärten konnte. Alle Klebestellen sind angeschliffen worden als Oberflächenvorbereitung

Widerstand zwischen Punkt A B 2.6

Widerstand zwischen Punkt A C 0.2

Widerstand zwischen Punkt A D keine Verbindung

Mechanische Stabilität von Test Aluminiumpulver nicht so gut

Ist es möglich auf die Stahlplatte zu verzichten und die Verlängerung mit der Graphit-Epoxy-Mischung zu machen?

Zwischen den beiden Graphitstäben ist eine PAAM-Platte geklebt. Alle offenen Stellen des Epoxy/Graphits sind mit reinem Epoxy überzogen, um Kriechspannungen durch Wasser zu verhindern.

Arbeitsschutz, erklären

Schnee ist Wasser, das vom Boden verdunstet, sich dann in der Atmosphäre an einem Nukleus kondensiert oder resublimiert und dann auf den Boden zurück fällt.

Im Alltag weiß man, dass man mit den Handfön nicht in die Dusche gehen darf, da Wasser elektrisch leitend ist. Diese Schlussfolgerung ist nicht sehr präzise, denn reines Wasser ist nicht leitend, sondern die Ionen (Salze), die im 'normalen' Wasser gelöst sind. Auf sehr geringem Niveau ist auch reines Wasser leitend, da sich spontan $1 \cdot 10^{-7} \text{ M Hydroniumionen (H}_3\text{O}^+) \text{ bilden und den pH-Wert 7 bilden.}$

Die Hypothese ist, dass sowohl die Verunreinigungen durch die Nuklei und die Hydroniumionen genügend Leitfähigkeit bilden, um einen Messwert im $\mu\text{S (Siemens = } 10^{-6} \text{ S)}$ Bereich zu messen.

Im Feldversuch konnte keine Leitfähigkeit gemessen werden.

Eine Erweiterung dieser Messung ist, einen Stoff zum Schnee dazu zu geben, der gut leitfähig ist. Dann wird der Versuchsaufbau mehr in die Richtung ??, wo die

Ausbreitung eines Stoffes im Schnee beobachtet wird. hier wäre diese beobachtung dann über die Leitfähigkeit und nicht wie in ?? optisch.

3.3 Laser Refraktion, Reflexion

Titel: Laser Refraktion und Reflexion

Funktionsweise Mit einem Laser wird der Schnee sowohl durchleuchtet für die Refraktion als auch angeleuchtet für die Reflexion. Flüssiges Wasser bildet wegen seiner Oberflächenspannung konkave linsen auf den Prismen des Eis kristalle. Die grösse und damit die Brennweite ändert sich, je nach dem wieviel Volumen Wasser auf den Eiskristallen ist. Die Effekte der Linsen sollten in der Refraktion sichtbar werden. i

In der Reflexion ändert sich mit änderndem LWC auch die Oberfläche an der das licht gespiegelt wird.

Der TRL für Refraktion und Reflexion ist bei 2.

Beispiele in anderen Sektoren Refraktion wird in der Kristallografie angewant.

und Reflexion wird bei einem auflicht mikroskop immer angeendet.

Die reflexion von wasser an einer Glasscheibe wird genutzt um bei Autos niederschlag auf der windschutzscheibe zu messen.

in beiden Fällen ist hier das TRL 9.

Literatur zu Reflexion Reflexion von IR In 20XX hat Herr XX die Reflexion von IR genutzt um den LWC von Schnee zu bestimmen. Die Ergebnisse wurden im Journal XX veröffentlicht.

benutzte Mittel für den Versuchsaufbau Als Laserquelle wurde ein grüner Bosch Quingo Kreuzlaser genutzt.

Um sowohl die Reflexion als auch die Refraktion gleichzeitig zu sehen, wurde die Schneeprobe auf einen Mikroskopier Objektträger plziert.

die Ergebnisse des Lasers wurden jeweils auf weissem Druckpapier dargestellt. Die Refraktion wird auf dem Papier an der Unterseite der Holzplatte dargestellt.

mit dem Fairphone 3 wurde eine Video aufnahme gemacht, wie sich die Ergebnisse des Lasers verändern.

mit einem Kosmetik Spiegel wurde sowohl die reflexion unten als auch die Refraktion oben gleichzeitig in einem Bild dargestellt.

Um alle Teile in fixen relationen zu halten wurde Stativmaterial genutzt.

In Bild ?? ist die Anordnung der Verschieden Teile auf den Stativmaterial zu sehen.

Funktionsweise des Versuchsaufbaus Der schnee wird in trockenem Zustand bei -10 grad aus dem Gefrierschrank auf den gekühlten Objektträger gelegt. dann wird beobachtet wie sich die Ergebnisse ändern wenn der schnee an der Raumtemperatur Luft schmilzt. Dieser Schmelzvorgang hat rund 5 min gedauert.

der Laser scheint durch den Objektträger und den Schnee durch. Dann wird das Licht auf den Papier erneut in die Kamera reflektiert.

Die Reflexion geschieht zum einen direkt am Objektträger, als auch danach im Schnee. Dieser Aufbau ist suboptimal, denn die konstante Reflexion des Objektträgers muss aus dem Laser-Ergebnis rausgerechnet werden.

Um Störlicht zu minimieren wurde zuerst eine Einhausung geplant. Der durchgeführte Versuch hat dann aber einfach in einem abgedunkelten Raum stattgefunden.

Messgrößen die Anhäufungen von Licht, und die Intensität wird begutachtet.

V Versuchsergebnisse

Im Bild ?? ist die Reflexion und Refraktion des Objektträgers sichtbar. Diese konstanten Werte müssen von allen Ergebnissen subtrahiert werden.

Aussagekraft der Ergebnisse über den LWC Die Ergebnisse werden direkt von Wasser beeinflusst. Um den Gewichtswert LWC zu erhalten, ist aber die Geometrie der Eiskristalle von externer Entscheidung. Daher ist das Ergebnis nicht direkt mit den LWC überzuführen. Mit der 3D Geometrie der Kristalle wäre die Aussagekraft gut.

Reflexion zum Versuchsaufbau da zwei Techniken gleichzeitig gemessen wurden, war der Versuchsaufbau nicht optimal für beide Messgrößen.

Mit den Ergebnissen der Refraktion bin ich sehr zufrieden.

Verbesserungen des Versuchsaufbaus um besser Reflexionsergebnisse zu bekommen keinen Objektträger nutzte, sondern direkt auf Schnee.

Für eine statische Messung einer Schneeprobe muss die Luft um den Schnee herum gekühlt sein. Ein Ansatz dafür wird im Vorversuch ?? umgesetzt.

Mit dem Laser wird Energie in den Schnee eingebracht, um das Schmelzen und damit Verfälschen des LWC zu minimieren sollte ein möglichst schwacher Laser eingesetzt werden.

Weiterverfolgung der physikalischen Methoden Das Ergebnis der Refraktion zeigt, dass diese Methode umgesetzt werden könnte. Um vergleichbare Werte zu bekommen ist die Kristallgeometrie aber von Bedeutung. Die Messung der Geometrie übersteigt das Ausmass der BA. Um eine Messung durchzuführen muss eine Schneeprobe durchleuchtet werden, um das zu erreichen muss der Schnee physikalisch aus der Schneedecke extrahiert werden. Das ist aufwendig, daher wird die Refraktion nicht weiter verfolgt.

Das Ergebnis der Reflexion ist schwer zu beurteilen. In ?? ist die Reflexion von EM-Wellen bereits untersucht worden. Daher wird die Reflexion nicht weiter untersucht.

3.4 Vibration

Avanode vibriert. Wenn kurz vor Gletschlawine wird der Schnee zur Flüssigkeit. Der Avanode sinkt auf Grund der hohen Dichte und verändert dabei die Position.

VibraNode

Die Form wird von dem AvaNode übernommen. Um eine hohe Formfreiheit und eine hohe Dichte zu erreichen wird der VibraNode aus Ton gebaut. Der ungebrannte Ton wird durch Epoxy Harz und Acryl Farbe vor Wasser geschützt.

Der erste Test hat nicht funktioniert. Ich stand auf dem Schnee, neben dem Virbanode, ich habe rund die vierfache Auflagefläche, aber das 60-fache Gewicht. Das heisst der Schnee war ungeeignet und nicht kurz vor einer Gletschneelawine. Zumindest an der Oberfläche.

Mit dem Virbanode ist es sicher nicht möglich den LWC fest zu stellen. Auch nachdem der Schnee mit Wasser übergossen worden ist, ist der VirbaNode nicht eingesunken.

Ist der LWC die entscheidende Grösse für Gletschneelawinen?

3.5 Diffusion von Flüssigkeit

Mit Handy und Stereoskop Aufbau.

Schnee gekühlt, durch Eisring und Eisunterlage.

Gekühlt ist fast noch besser als perfekt isoliert.

Das obere Abdeckglas wurde weggelassen, da optisch nicht klar genug.

Polarisation von Lichtquelle, oder des reflektierten Lichts hat keinen erkennbaren Effekt auf die Speckeln in der Videoaufnahme.

4 Funktionsmuster

4.1 Funktionsweise

4.2 Bildverarbeitung

Funktionsweise der Bildverarbeitung

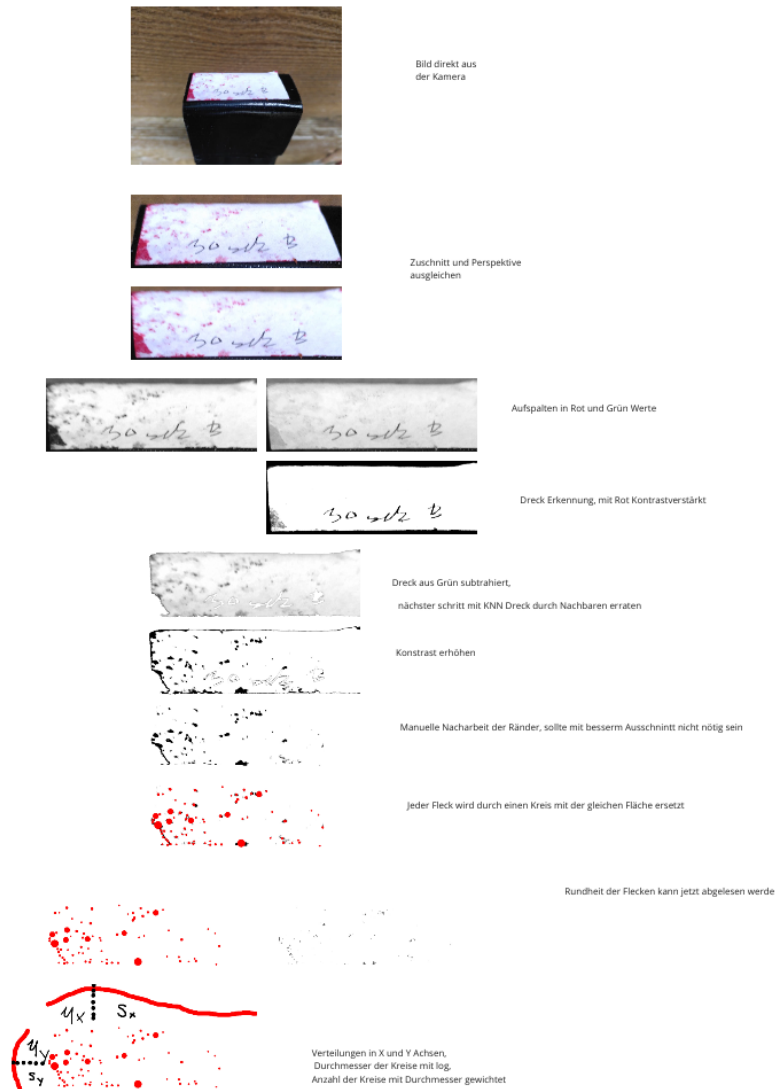


Abbildung 1: Bildverarbeitungskonzept

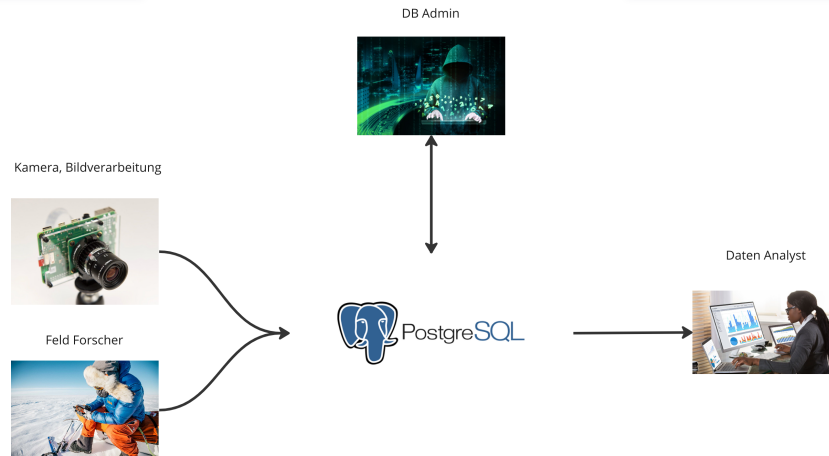


Abbildung 2: Benutzer der Datenbank

4.3 Extrahieren von Informationen aus Bilddaten

Um aus den Bilddaten, die während Feldversuchen gesammelt werden, sinnvolle Erkenntnisse zu gewinnen, ist es entscheidend, die Daten effektiv zu strukturieren. Dies erleichtert die effiziente Speicherung und ermöglicht leistungsstarke Datenabfragefunktionen, wie z. B. das pattern matching, die für eine umfassende Analyse wichtig sind. Hierfür ist der Einsatz eines Datenbanksystems sinnvoll.

Die im Feld gesammelten Daten werden zunächst in der Datenbank gespeichert und zu einem späteren Zeitpunkt analysiert.

Im Folgenden werden die Schritte zur Auslegung der Datenbank dargestellt. Der Code ist in Section 4.3.8

4.3.1 Anforderungsanalyse

Die Anforderungen ergeben sich aus der Funktionsweise des Messaufbaus.

Die Datenbank in dieser Bachelorarbeit wird relativ klein sein, da die Feldversuche zeitintensiv sind. Es wird vermutet, dass maximal 1000 Messungen mit jeweils 3 Taps und je 100 Kreisen durchgeführt werden.

Die Datenbank ist grösser angelegt, als sie für die Vorversuche in der Bachelorarbeit benötigt wird.

Es gibt vier Benutzer die mit der Datenbank interagieren. In der Grafik 2 ist die schematische Darstellung.

1. Die Kamera, die die Bilder der Taps macht und auswertet, muss die Auswertungen in die Datenbank schreiben.

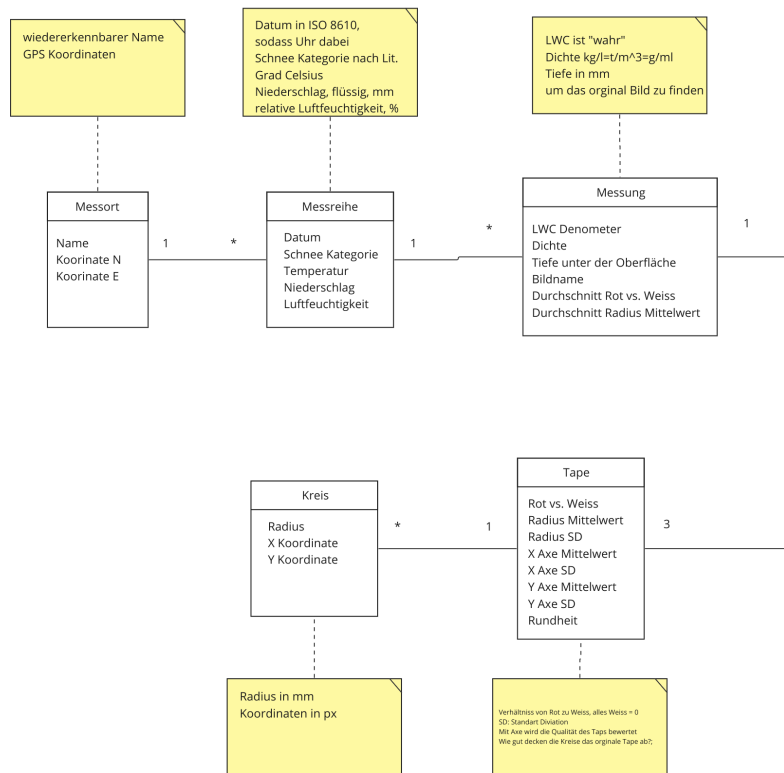


Abbildung 3: UML-Diagramm des konzeptuellen DB-Entwurfs

2. Der Versuchsdurchführende gibt zusätzliche Informationen über den Versuch an, die er ebenfalls in die Datenbank schreiben muss.
3. Der Analyst wird die Daten abfragen und hoffentlich Informationen daraus gewinnen.
4. Der Datenbankadministrator wird im Normalbetrieb nicht benötigt, sollte jedoch berücksichtigt werden.

Die Anforderungen an die Datenbank und ihre Benutzer werden entsprechend den Anforderungen des Messaufbaus und den Bedürfnissen der Benutzer festgelegt.

1

4.3.2 Konzeptueller DB Entwurf

Mit der Unified Modeling Language (UML) wird die Struktur der Datenbank dargestellt. Diese Darstellung ist noch lösungsunabhängig.

4.3.3 Logischer DB Entwurf

Um die Datenbank zu implementieren, wurde PostgreSQL gewählt. Es ist ein Free und Open-Source-System, das neue Features wie zum Beispiel JSON-Datentypen unterstützt.

Der folgende SQL-Code initialisiert die Datenbank: 2

4.3.4 Ansichten für den Analysten

Das Endziel besteht darin, eine Regression aus den Messungen und Taps zu erstellen, um den 'LWC Denoth' zu bestimmen. Für diese Aufgabe sind nur bestimmte Angaben aus der Datenbank erforderlich.

Hier werden zwei Ansichten erstellt: Der erste ist ein minimalistischer Ansatz, mit dem direkt weitergearbeitet werden kann. Die zweite Ansicht dient dazu, genauer zu verstehen, was in der ersten Ansicht dargestellt ist.

Da die Ansichten für den read only Analysten bestimmt sind, muss keine aktualisierbare Ansicht verwendet werden.

3

4.3.5 Physischer Entwurf

Für die Beispieldaten wurden Daten aus der Vorstudie ?? für eine Messung verwendet.

Die Datenbank wird anfangs viele NULL-Werte enthalten, da beispielsweise die Wetterdaten nicht von einer API gefüllt werden. Das ist auch in Ordnung, da die fehlenden Werte mit 0 aufgefüllt werden.

Die Transaktionen sind in dieser Anwendung unproblematisch, da der Benutzer, der die Inserts durchführt (Raspberry, Feldforscher), zu einem früheren Zeitpunkt arbeitet als der Analyst.

Falls die Datenbank von meinem Laptop auf einen Server ausgelagert wird, werden die folgenden Tools zur Sicherheitsprüfung verwendet: www.owasp.org und <http://sqlmap.org/>.

4.3.6 Python-Interaktion mit der Datenbank

Für die Interaktion mit der Datenbank werden verschiedene Python-Skripte verwendet, die je nach Benutzer unterschiedliche Aufgaben erfüllen.

Das folgende Python-Skript ist dazu da Bilder von Taps zu analysieren und die daraus gewonnenen Daten in die Datenbank einzufügen. 5

Das nächste Python-Skript wird interaktiv vom Versuchsleiter verwendet. Zur Zeit ruft das Skript die Bildanalyse auf. 6

4.3.7 Nächste Schritte

Die Python-Programme sollten weiterentwickelt werden, um sämtliche verfügbaren Daten in der Datenbank zu nutzen und um die Funktionalität zu verbessern.

Aktuell läuft die Datenbank mit dem Benutzer Postgres auf einem Laptop. Eine Auslagerung auf einen Server ist derzeit keine Priorität, da dies mit Sicherheitsrisiken verbunden ist. Das Hauptziel dieser Produktentwicklungs Bachelorarbeit besteht darin, das Verhalten des Taps zu verstehen. Sobald dieses Ziel erreicht ist, können weitere Schritte zur Optimierung und Sicherung der Datenbankanfrastruktur unternommen werden.

4.3.8 Code

Listing 1: SQL-Code für die Benutzerinitialisierung

```
-- Prevent default role PUBLIC from creating tables:
REVOKE CREATE ON SCHEMA public FROM PUBLIC;

CREATE USER RaspberryKamera WITH PASSWORD 'abscaaksd.tt33' NOINHERIT;
-- Grant insert and update permissions on specific tables
GRANT INSERT ON TABLE Kreis TO RaspberryKamera;
GRANT INSERT ON TABLE Messung TO RaspberryKamera;
GRANT INSERT, SELECT ON TABLE Tape TO RaspberryKamera;

CREATE USER Feldversuch WITH PASSWORD 'bsacauxiaxbc222/' NOINHERIT;
-- Grant insert permissions on specific tables
GRANT INSERT, SELECT ON TABLE Messung TO Feldversuch;
GRANT INSERT ON TABLE Messreihe TO Feldversuch;
GRANT INSERT ON TABLE Messort TO Feldversuch;

CREATE USER Analyst WITH PASSWORD 'rabgkkaadggg221!' NOINHERIT;
GRANT SELECT ON ALL TABLES IN SCHEMA public TO Analyst;

CREATE USER admin WITH PASSWORD 'sgintyijyj77(';
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO admin WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO admin WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON SCHEMA public TO admin WITH GRANT OPTION;
```

Die pseudozufällige Passwörter sind nicht optimal, besser wäre $SELECT gen_random_uuid()$;

Listing 2: SQL-Code für die DBinitialisierung

```
CREATE DATABASE TapeMessungenBAKuhn;

CREATE TABLE MessOrt (
    id SERIAL PRIMARY KEY,
```

```
nameOrt VARCHAR(255) NOT NULL,
koordinateN FLOAT,
koordinateE FLOAT
);

CREATE TABLE MessReihe (
    id SERIAL PRIMARY KEY,
    datum TIMESTAMP WITH TIME ZONE NOT NULL,
    schneeKategorie VARCHAR(255),
    temperatur FLOAT,
    niederschlag INT,
    luftfeuchtigkeit FLOAT,
    -- Foreign Key reference to MessOrt
    messOrt_id INT REFERENCES MessOrt(id)
);

CREATE TABLE Messung (
    id SERIAL PRIMARY KEY,
    lwcDenothMeter FLOAT,
    dichte FLOAT,
    tiefeUnterSchnee INT NOT NULL,
    bildname VARCHAR(255) NOT NULL,
    -- Foreign Key reference to MessReihe
    messReihe_id INT REFERENCES MessReihe(id)
);

CREATE TABLE Tape (
    id SERIAL PRIMARY KEY,
    rotVsWeiss FLOAT NOT NULL,
    radiusMittelwert FLOAT NOT NULL,
    radiusSD FLOAT NOT NULL,
    xAxeMittelwert FLOAT NOT NULL,
    xAxeSD FLOAT NOT NULL,
    yAxesMittelwert FLOAT NOT NULL,
    yAxeSD FLOAT NOT NULL,
    rundheit FLOAT NOT NULL,
    -- Foreign Key reference to Messung
    messung_id INT REFERENCES Messung(id)
);

CREATE TABLE Kreis (
    id SERIAL PRIMARY KEY,
    Radius FLOAT NOT NULL,
    xKooridnate INT NOT NULL,
    yKooridnate INT NOT NULL,
    -- Foreign Key reference to Tape
    tape_id INT REFERENCES Tape(id)
);
```

Listing 3: SQL-Code für die Views

```
CREATE VIEW Minimal_Messung_Tape_View AS
SELECT m.lwcDenothMeter, t.rotVsWeiss, t.radiusMittelwert
FROM Messung m
JOIN Tape t ON m.id = t.messung_id;
```

```
CREATE VIEW Full_Measurement_View AS
SELECT mo.id AS messort_id, mo.nameOrt, mo.koordinateN, mo.koordinateE,
       mr.id AS messreihe_id, mr.datum, mr.schneeKategorie, mr.temperatur, mr.niederschlag,
       m.id AS messung_id, m.lwcDenothMeter, m.dichte, m.tiefeUnterSchnee, m.bildname,
       t.id AS tape_id, t.rotVsWeiss, t.radiusMittelwert, t.radiusSD, t.xAxeMitte
FROM MessOrt mo
JOIN MessReihe mr ON mo.id = mr.messOrt_id
JOIN Messung m ON mr.id = m.messReihe_id
JOIN Tape t ON m.id = t.messung_id;
```

Listing 4: SQL-Code für Beispiel Daten

```
INSERT INTO MessOrt (nameOrt, koordinateN, koordinateE)
VALUES ('Rothenthurm', 47.1, 8.683333);
```

```
INSERT INTO MessReihe (datum, schneeKategorie, temperatur, niederschlag, luftfeuchtigkeit)
VALUES ('2024-03-10T15:02:08', 'schnee_beregnete', 6, 2, 100, 1);
```

```
INSERT INTO Messung (lwcDenothMeter, dichte, tiefeUnterSchnee, bildname, messReihe_id)
VALUES (NULL, NULL, 30, 'bild1.jpg', 1);
```

Listing 5: Bilderkennung und verarbeitung

```
#!/usr/bin/python
```

```
import psycopg2
from config import config
import cv2
import numpy as np
import pandas as pd
```

```
def do_image(image_name, messung_id): # Accept tape_id as a parameter
    """
    Process an image to detect circles, calculate statistics, and insert data into the database.

    Args:
        image_name (str): The filename of the image to process.
        messung_id (int): The ID of the Messung associated with the circles.
```



```
Returns:
    float: The mean radius of the detected circles.
"""

df = process_image(image_name)
mean_radius = perform_statistics(df, messung_id)

tape_id = get_last_tape_id()
for index, row in df.iterrows():
    insert_data_kreis(row['Radius'], row['X-coordinate'], row['Y-coordinate'])
# Pass tape_id to insert_data
#print(df)
return mean_radius

# Function to perform statistics on a DataFrame
def perform_statistics(df, messung_id):
    """
    Calculate statistics on a DataFrame containing circle data and insert them into the database.

    Args:
        df (pandas.DataFrame): DataFrame containing circle data.
        messung_id (int): The ID of the Messung associated with the circle data.

    Returns:
        float: The mean radius of the detected circles.
    """

    # Calculate mean and standard deviation
    mean_radius = df['Radius'].mean()
    mean_x_coordinate = df['X-coordinate'].mean()
    mean_y_coordinate = df['Y-coordinate'].mean()
    std_radius = df['Radius'].std()
    std_x_coordinate = df['X-coordinate'].std()
    std_y_coordinate = df['Y-coordinate'].std()

    # Insert statistics into the database
    insert_data_tape(mean_radius, mean_x_coordinate, mean_y_coordinate, std_radius)

    return mean_radius

def insert_data_tape(mean_radius, mean_x_coordinate, mean_y_coordinate, std_radius):
    """
    Insert statistics into the database.
    """
```

Args:

mean_radius (float): Mean radius of detected circles.
mean_x_coordinate (float): Mean x-coordinate of detected circles.
mean_y_coordinate (float): Mean y-coordinate of detected circles.
std_radius (float): Standard deviation of radius of detected circles.
std_x_coordinate (float): Standard deviation of x-coordinate of detected circles.
std_y_coordinate (float): Standard deviation of y-coordinate of detected circles.
messung_id (int): The ID of the Messung associated with the statistic.

Returns:

None

```
"""
sql = """INSERT INTO tape (radiusmittelwert, xaxemittelwert, yaxesmitte
VALUES (%s, %s, %s, %s, %s, %s);"""
conn = None
try:
    # Read database configuration
    params = config()
    # Connect to the PostgreSQL database
    conn = psycopg2.connect(**params)
    # Create a new cursor
    cur = conn.cursor()
    # Execute the INSERT statement
    cur.execute(sql, (mean_radius, mean_x_coordinate, mean_y_coordinate,
    # Commit the changes to the database
    conn.commit()
    print("Statistics inserted into the database.")
    # Close communication with the database
    cur.close()
except (Exception, psycopg2.DatabaseError) as error:
    print(error)
finally:
    if conn is not None:
        conn.close()

# Function to retrieve the last inserted tape_id
def get_last_tape_id():
    """
    Retrieve the ID of the last inserted tape from the database.

    Returns:
        int: The ID of the last inserted tape.
    """
```

```
        # Read database configuration
params = config()
        # Connect to the PostgreSQL database
conn = psycopg2.connect(**params)
        # Create a new cursor
cur = conn.cursor()

cur.execute("SELECT id FROM tape")
last_tape_id = cur.fetchone()
if last_tape_id:
    return last_tape_id[0]
else:
    return 1

def show_image_progsess(df, image, contours, radii_list, x_coords_list, y_c
    """
    Display the processed image with circles and contours.

    Args:
        df (pandas.DataFrame): DataFrame containing circle data.
        image (numpy.ndarray): Original image.
        contours (list): List of contours detected in the image.
        radii_list (list): List of radii of detected circles.
        x_coords_list (list): List of x-coordinates of detected circles.
        y_coords_list (list): List of y-coordinates of detected circles.

    Returns:
        None
    """
    # Display DataFrame
    print(df)

    # Display the original image
    cv2.imshow('Original Image', cv2.imread(image_name))

    # Display the image with contours
    image_with_contours = cv2.cvtColor(image, cv2.COLOR_GRAY2BGR)
    for contour in contours:
        cv2.drawContours(image_with_contours, [contour], 0, (0, 255, 0), 2)
    cv2.imshow('Image with Contours', image_with_contours)
```

```
# Display the image with contours and circles
# Create a copy of the original image for drawing circles
image_with_circles = cv2.cvtColor(image, cv2.COLOR_GRAY2BGR)

for x, y, r in zip(x_coords_list, y_coords_list, radii_list):
    cv2.circle(image_with_circles, (x, y), r, (0, 0, 255), cv2.FILLED)
cv2.imshow('Image_with_Circles', image_with_circles)

cv2.waitKey(0)
cv2.destroyAllWindows()

def insert_data_kreis(radius, x_coordinate, y_coordinate, tape_id):
    """
    Insert circle data into the database.

    Args:
        radius (int): Radius of the circle.
        x_coordinate (int): X-coordinate of the circle.
        y_coordinate (int): Y-coordinate of the circle.
        tape_id (int): The ID of the tape associated with the circle.

    Returns:
        None
    """
    sql = """INSERT INTO kreis (radius, xkooridnate, ykooridnate, tape_id)
    conn = None
    try:
        # Read database configuration
        params = config()
        # Connect to the PostgreSQL database
        conn = psycopg2.connect(**params)
        # Create a new cursor
        cur = conn.cursor()
        # Convert NumPy integers to Python integers
        radius = int(radius)
        x_coord = int(x_coordinate)
        y_coord = int(y_coordinate)
        # Execute the INSERT statement
        cur.execute(sql, (radius, x_coord, y_coord, tape_id))
        # Commit the changes to the database
        conn.commit()
        # Close communication with the database
```

```
        cur.close()
    except (Exception, psycopg2.DatabaseError) as error:
        print(error)
    finally:
        if conn is not None:
            conn.close()

def process_image(image_name):
    """
    Process an image to detect circles and return a DataFrame containing circle data.

    Args:
        image_name (str): The filename of the image to process.

    Returns:
        pandas.DataFrame: DataFrame containing circle data.
    """
    # Load the image
    image = cv2.imread(image_name, cv2.IMREAD_GRAYSCALE)

    # Invert the image (since blobs are black on a white background)
    image = cv2.bitwise_not(image)

    # Find contours
    contours, _ = cv2.findContours(image, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    # Lists to store radius, x-coordinate, and y-coordinate
    radii_list = []
    x_coords_list = []
    y_coords_list = []

    # Iterate through each contour
    for contour in contours:
        # Calculate the center of gravity (centroid) and area of the contour
        M = cv2.moments(contour)
        if M["m00"] != 0:
            cx = int(M["m10"] / M["m00"])
            cy = int(M["m01"] / M["m00"])
            area = cv2.contourArea(contour)

            # Calculate the radius of the circle using the area
            radius = int(np.sqrt(area / np.pi))
```

```
        # Append radius, x-coordinate, and y-coordinate to respective lists
        radii_list.append(radius)
        x_coords_list.append(cx)
        y_coords_list.append(cy)

    # Create DataFrame
    data = {'Radius': radii_list, 'X-coordinate': x_coords_list, 'Y-coordinate': y_coords_list}
    df = pd.DataFrame(data)

    if __name__ == '__main__':
        show_image_progress(df, image, contours, radii_list, x_coords_list, y_coords_list)

    return df

if __name__ == '__main__':
    image_name = 'bild1.png'

    do_image(image_name)
```

Listing 6: Bilderkennung und verarbeitung

```
#!/usr/bin/python

import psycopg2
from config import config
import cv2
import numpy as np
import pandas as pd
from imageToCircle3 import do_image
# Function to retrieve the last inserted tape_id
def get_last_messung_id(cur):

    """
    Retrieve the last inserted messung_id from the database.

    Args:
        cur (psycopg2.cursor): Cursor object for database interaction.

    Returns:
        int: The last inserted messung_id incremented by 1.
    """
```

```
cur.execute("SELECT id FROM messung")
last_messung_id = cur.fetchone()
if last_messung_id:
    return last_messung_id[0] + 1
else:
    return 1
def insert_data(messreihe_id):
    """
    Insert a new messung entry into the database.

    Args:
        messreihe_id (int): The ID of the messreihe associated with the mes

    Returns:
        None
    """
    sql = """INSERT INTO messung (messreihe_id)
            VALUES (%s);"""
    conn = None
    try:
        # Read database configuration
        params = config()
        # Connect to the PostgreSQL database
        conn = psycopg2.connect(**params)
        # Create a new cursor
        cur = conn.cursor()
        # Execute the INSERT statement
        cur.execute(sql, (messreihe_id))
        # Commit the changes to the database
        conn.commit()
        print("Messung inserted into the database.")
        # Close communication with the database
        cur.close()
    except (Exception, psycopg2.DatabaseError) as error:
        print(error)
    finally:
        if conn is not None:
            conn.close()

# Main function
def main():
    """
    Main function to execute the program.
```

```
Prompts the user for a messungReihe ID input, processes a list of image
"""
# Prompt user for messung_id input
messung_id = input("Enter MessungReihe ID: ")

# List of image names
image_names = ['bild1.png', 'bild2.png', 'bild3.png']

try:
    # Read database configuration
    params = config()
    # Connect to the PostgreSQL database
    conn = psycopg2.connect(**params)
    # Create a new cursor
    cur = conn.cursor()

    # Retrieve the last inserted tape_id
    messung_id = get_last_messung_id(cur)
    print("Next MessungReihe ID: ", messung_id)

    avg_mean_radius = 0
    # Process each image
    for idx, image_name in enumerate(image_names):
        # Process the image
        avg_mean_radius += do_image(image_name, messung_id)
    avg_mean_radius = avg_mean_radius / 3

    # Commit the transaction
    conn.commit()

except (Exception, psycopg2.DatabaseError) as error:
    print(error)
finally:
    # Close cursor and connection
    if cur is not None:
        cur.close()
    if conn is not None:
        conn.close()
```



```
# Entry point of the program  
if __name__ == "__main__":  
    main()
```

4.4 Testkriterien

4.5 Widerstand gegen Umwelteinflüsse

4.6 Montage des Funktionsmusters

4.7 Ergebnisse der Versuche

4.8 Vergleich der Ergebnisse mit Denometer

4.9 Verbesserungsmöglichkeiten des Funktionsmusters

5 Ausblick

5.1 Persönliche Erfahrung

5.2 Fazit

5.3 Ausblick

methode weiter verfolgen, good stuff

6 Literaturverzeichnis

7 Erklärung zur Urheberschaft

Ich erkläre hiermit, dass ich die vorliegende Arbeit ohne Hilfe Dritter angefertigt habe. Ich habe nur die Hilfsmittel benutzt, die ich angegeben habe. Gedanken, die ich aus fremden Quellen direkt oder indirekt übernommen habe, sind kenntlich gemacht. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

KI-Einsatz ohne Kennzeichnungspflicht

Ich bin mir bewusst, dass die Nutzung maschinell generierter Texte keine Garantie für die Qualität von Inhalten und Text gewährleistet. Ich versichere daher, dass ich mich textgenerierender KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Ich verantworte die Übernahme jeglicher von mir verwendeter maschinell generierter Textpassagen vollumfänglich selbst. Ich versichere, dass ich keine KI-Schreibwerkzeuge verwendet habe, deren Nutzung der Prüfer / die Prüferin explizit schriftlich ausgeschlossen hat.

Ort/Datum: Rapperswil, 2024

Unterschrift:

Peter Kuhn

Abbildungsverzeichnis

1	Bildverarbeitungskonzept	7
2	Benutzer der Datenbank	8
3	UML-Diagramm des konzeptuellen DB-Entwurfs	9

Tabellenverzeichnis

8 Digitaler Anhang

Lebenslauf

Personalien

Peter Kuhn
Webergasse 16
8640 Rapperswil
078 707 12 46 (Mobil)
043 268 55 87 (Festnetz)
peter.jo.kuhn@gmail.com



Bildung

1998.06.17	geboren
2005 - 2011	Primarschule
2011 - 2013	Langzeit Gymnasium Kantonsschule Zürcher Oberland
2013 - 2017	Kurzzeit Gymnasium Math. Naturwiss. Gym. Rämibühl
2017 - 2018	Zivildienst
2018 - 2020	Mathematik Studium ETH Zürich
2021 - jetzt	Maschinenentechnik und Innovation Studium an der OST

Maturarbeit

Sprachen

- Deutsch (Muttersprache)
- Englisch (sehr gut schriftlich und mündlich)
- Italienisch (gut mündlich)

Programmiersprachen

C++, Java, Python,
JavaScript, Bash, Matlab, html/css, Mysql, Exel

Fähigkeiten

Führerausweis Kat. B
Aktives Mitglied von Velove,
einer von Studenten geleiteten Velo Werkstatt

Sport

Mountainbike
Rennvelo
Schwimmen