Table of Contents

[Introduction: 3](#_Toc135089319)

[Mathematics: 3](#_Toc135089320)

[Algorithms: 3](#_Toc135089321)

[Software Design: 4](#_Toc135089322)

[Source Code: 5](#_Toc135089323)

[Test Data and Results: 5](#_Toc135089324)

[Graphical User Interface (GUI): 6](#_Toc135089325)

[Presentation of the Algorithm: 6](#_Toc135089326)

[Oral Presentation and Discussion: 7](#_Toc135089327)

**Project Report:**

# Introduction:

The goal of this project is to develop software that utilizes a dynamic time warping (DTW) algorithm for shape matching. The application field chosen for this project is medical imaging, specifically to aid in surgery planning. The software will allow surgeons to match and compare shapes in medical images, enabling them to plan surgical procedures more accurately.

# Mathematics:

Dynamic Time Warping is a technique used to measure the similarity between two temporal sequences that may vary in speed or timing. It aligns two sequences by warping the time axis, allowing for non-linear mappings. The algorithm finds the optimal path through the cost matrix by minimizing the cumulative distance between corresponding points in the sequences.

# Algorithms:

The following steps outline the algorithm for shape matching using dynamic time warping:

Preprocess the input shapes: Convert the input shapes into a suitable representation, such as point clouds or contours. Normalize the shapes to ensure they are of comparable sizes and orientations.

Compute the cost matrix: Construct a cost matrix that represents the pairwise distances between the points of the two shapes. The distance measure can be based on Euclidean distance, Hausdorff distance, or other suitable metrics.

Initialize the accumulated cost matrix: Create an accumulated cost matrix of the same size as the cost matrix. Initialize the first row and column with cumulative distances.

Compute the optimal warping path: Traverse the accumulated cost matrix, calculating the cumulative distance at each point by considering the neighboring cells. Choose the path with the minimum cumulative distance.

Backtrace the optimal path: Starting from the bottom-right cell of the accumulated cost matrix, backtrace the path that leads to the top-left cell. This path represents the optimal alignment between the two shapes.

Measure the similarity: The similarity between the shapes can be quantified by the accumulated distance along the optimal path or by a normalized distance measure.

# Software Design:

The software will be designed with a user-friendly graphical user interface (GUI) to facilitate easy interaction. The main components of the software will include:

File Input: The GUI will allow users to select and load measurement files containing the shapes to be matched.

Shape Preprocessing: The software will preprocess the input shapes, normalizing them and converting them into a suitable representation.

DTW Algorithm: The DTW algorithm will be implemented to compute the optimal warping path between the shapes.

Shape Matching Results: The software will display the matching results both graphically, by plotting the shapes and the optimal alignment, and as text, by providing the similarity measure and other relevant information.

# Source Code:

The source code for the software can be implemented in a programming language such as Python. Here is a simplified example of the code structure:

# Import necessary libraries

# Define functions for shape preprocessing

# Define function for computing the cost matrix

# Define function for computing the accumulated cost matrix

# Define function for backtracing the optimal path

# Define function for measuring similarity

# Define GUI components and event handlers

# Main program loop

# Run the GUI application

# Test Data and Results:

To test the software, a dataset of medical images with known ground truth shapes will be used. The software will compare the matched shapes with the ground truth and compute the similarity measure. Various test cases with different shapes and deformations will be used to evaluate the performance of the algorithm. The results will be recorded, including the similarity scores and any discrepancies between the matched shapes and the ground truth.

# Graphical User Interface (GUI):

The GUI will provide a user-friendly interface for the shape matching software. It will allow users to select a measurement file, visualize the input shapes, and display the matching results. Here is a brief description of the GUI components and their functionalities:

File Selection: The GUI will include a file selection feature where users can choose the measurement file containing the shapes to be matched.

Shape Visualization: Once a file is selected, the GUI will display the input shapes graphically. This can be done using a plot or a graphical representation of the shapes.

Result Display: The GUI will show the matching results both graphically and as text. The graphical representation can include the input shapes, the optimal alignment, and any other relevant visualizations. The text display will provide the similarity measure and additional information about the matching process.

Controls and Options: The GUI can include buttons or dropdown menus for additional controls and options. For example, users might have the option to select different distance metrics, adjust parameters for shape normalization, or choose different visualization settings.

# Presentation of the Algorithm:

A presentation of the algorithm can be created using PowerPoint or other similar tools. The presentation should include the following key points:

Introduction: Briefly explain the purpose of the project and the application field chosen (medical imaging for surgery planning).

Dynamic Time Warping: Provide an overview of the dynamic time warping algorithm, explaining its ability to measure similarity between sequences and its applicability to shape matching.

Algorithm Steps: Present the steps of the algorithm in a clear and concise manner. Use diagrams or flowcharts to illustrate the process.

Cost Matrix and Optimal Path: Describe the construction of the cost matrix and the determination of the optimal warping path. Explain how the path represents the alignment between the shapes.

Shape Matching Results: Discuss how the similarity between shapes can be quantified using the accumulated distance or a normalized distance measure.

Software Demonstration: Showcase the graphical user interface of the software and demonstrate its functionality by loading sample measurement files, visualizing shapes, and displaying matching results.

Performance and Future Enhancements: Mention any performance considerations, such as runtime complexity, and discuss possible improvements or extensions to the algorithm.

# Oral Presentation and Discussion:

Prepare a short oral presentation to present the project results and participate in a discussion. The presentation should cover the following points:

Introduction: Start with a brief introduction, explaining the purpose of the project and the chosen application field.

Algorithm Overview: Provide a high-level overview of the dynamic time warping algorithm for shape matching. Highlight its advantages and suitability for medical imaging.

Software Demonstration: Conduct a live demonstration of the software, showcasing its user interface, loading measurement files, and displaying matching results. Explain the steps involved in shape matching using the software.

Test Results: Present the results obtained from testing the software on various datasets. Discuss the similarity scores, visual comparisons, and any limitations or challenges encountered during the testing phase.

Future Directions: Share potential future enhancements or applications of the software. Discuss how it can be integrated into existing surgical planning systems and its potential impact on improving surgical procedures.

Discussion: Engage the audience in a discussion about the project, addressing any questions, feedback, or suggestions they may have.

By following these steps, you will be able to successfully complete Task T1 and deliver a comprehensive project report, a functional GUI, a presentation explaining the algorithm, and present your project results effectively.

Rough Contents:

Introduction

1.1 Project Overview

1.2 Objective

1.3 Scope

Mathematics

2.1 Dynamic Time Warping Algorithm

2.2 Distance Metrics

2.3 Similarity Measures

Algorithms

3.1 Shape Preprocessing

3.2 Construction of Cost Matrix

3.3 Computation of Accumulated Cost Matrix

3.4 Optimal Warping Path Determination

3.5 Similarity Measurement

Software Design

4.1 GUI Design and Components

4.2 File Input and Shape Visualization

4.3 DTW Algorithm Implementation

4.4 Result Display and User Interaction

4.5 Additional Controls and Options

Source Code

5.1 Overview of the Source Code Structure

5.2 Functions and Modules

Test Data and Results

6.1 Dataset Description

6.2 Testing Methodology

6.3 Performance Evaluation

6.4 Discussion of Results

Presentation of the Algorithm

7.1 PowerPoint Slides or Visual Representation

7.2 Algorithm Overview

7.3 Steps and Diagrams

Oral Presentation and Discussion

8.1 Introduction

8.2 Algorithm Overview

8.3 Software Demonstration

8.4 Test Results

8.5 Future Directions

8.6 Discussion

Conclusion

9.1 Summary of Achievements

9.2 Lessons Learned

9.3 Final Remarks

References

Appendices

11.1 Sample Measurement Files

11.2 Additional Figures or Graphs