Edge Hill University

Department of Computer Science

CHEMICAL MOLECULE 3-D SHAPE MATCHING AND VISUALISATION IN IMMERSIVE VIRTUAL REALITY

A thesis submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy

Supervisory Team:
Doctor Peter Vangorp
Professor Yonghuai Lui
Doctor Ardhendu Behera

Peter Ankomah

Edge Hill University, UK,

September 2021

Dedication

I dedicate this thesis to the Almighty God my creator.
It is also dedicated to my parents; the memory of Samuel Ankomah who encouraged me through this project although unable to see my graduation and Janet Osei who encouraged and supported me with words of life as well as my siblings in many diverse ways.

Acknowledgements

I am heartily grateful to the almighty God for his gift of life. I appreciate my supervisory team, Doctor Peter Vangorp, Professor Yonghuai Lui, and Doctor Ardhendu Behera whose encouragement, guidance and support enabled me to build up awareness of my subject area. Their discussions, clarifications, and autonomy gave me immense confidence and helped my learning throughout the study.

Abstract

Registering shapes is fundamental to many tasks in computing such as fingerprint recognition and motion and image analysis and detection.

Particularly for three dimensional (3-D) shapes, the advantages of visualising data in 3-D such as the ability to represent more details enable improved analysis due to this representation.

For shape matching, the iterative closest point algorithm (ICP) has been the dominant algorithm for such purposes. ICP, however, has several limitations such as its expensive computational cost and its need for a good initial transformation.

Molecules have been represented as 3D point clouds in several tasks such as drug discovery and exploration of the functions of proteins by allowing the 3D point cloud to overlap another molecule fully or partially. Exploring such shapes matching in virtual reality (VR) also provides an immersive visualisation and analysis experience and a way to easily use more intuitive gestures using hand controllers instead of a mouse and keyboard to interact with the shape matching process. This allows a faster-repeated test of the matching as required and enhances the efficient exploration of the 3-D shape.

This thesis aims to develop and evaluate shape matching algorithms based on the standard ICP algorithm and a virtual reality visualisation application for visualising and interacting with the 3-D molecular protein shapes. It proposes three ICP variants developed to improve the registration of 3-D molecular structures. The approaches used in the variants are k-means clustering for partitioning the search space for correspondence, the use of metadata information to enhance meaningful matching and reduce the search space for correspondence, and the use of partial matches to register a subset of the shape and reduce the search space.

The algorithms are evaluated using 3-D molecules under different conditions such as noise levels and mutations to compare their computational speed, convergence properties and the quality of the matches for the 3-D molecular structures.

Molecular structures form the basis of all organisms. Visualisation of molecular structures is important because of its applications such as in drug discovery. This research further presents a demonstration of an exploratory immersive virtual reality application developed in Unity3D for visualising the matching of 3-D molecular structures using the HTC VIVE VR system and the ICP algorithms.

Table of Contents

List of Abbreviations

2

2-D - Two Dimensional 10

3

3-D – Three Dimensional 9

C

CAD - Computer Aided Design 88
CAVE - CAVE Automatic Virtual Environment 83

G

GPU - Graphics Processing Unit 187

H

HMD - Head Mounted Display 81

I

ICP - Iterative Closest Point 9
IVR - Immersive Virtual Reality 82

M

MRI - Magnetic Resonance Imaging 11

P

PDB - Protein Data Bank 189

S

SUS - System Usability Scale 90

U

UI - User Interface 27, 181
UX - User Experience 91

V

VE - Virtual Environment 82
VR - Virtual Reality 9

List of Figures

List of Tables

PART 1: INTRODUCTION

1 CHAPTER 1 INTRODUCTION

1.1 Background

1.1.1 Data and Visualisation

Data is increasingly being generated by every human or machine activity. These masses of data are constantly being mined for insights that can help decision making and improve processes by presenting the data through reports and visualisations.

The exponential growth of this data has resulted in complex, multifaceted data, that poses a challenge to storage, acquisition, and processing. This data complex, termed big data (Keahey, 2013) can consist of data of several types including streaming data such as Twitter posts or high dimensional for instance 3-D molecules.

McKinsey Global Institute (2011) expound on the exponential growth of big data to stem from all fields of life and activity including companies capturing trillions of bytes of information about their customers, suppliers, and operations, and millions of networked sensors being embedded in the physical world in devices such as mobile phones and automobiles as well as multimedia and data from social network websites. This raises the challenge of data visualisation, using techniques and methods for optimally presenting data to users in a way that maximises their understanding to effectively make meaning out of the data (Kirk, 2012).

Keahey (2013) explains that the limitation to visualising data using the common visualisation techniques such as bar charts and scatter plots on a 2-D display, presents challenges such as the need to consider; the complexity of the data in terms of the size and type, the specifications of the display device, how the hardware and software can scale the visualisation in terms of allowing specific features such as real-time drilling down and enlarging the presentation, how the technology used supports collaboration and sharing, and the ability of the display layout to support changing visualisation needs. These limitations place virtual reality visualisation as a major frontier because these challenges can be overcome by the ability to create varied data formats, representations, and collaborative virtual visualisations.

Kirk (2012) proposes a methodological approach to data visualisation projects. The research identified key stages of a data visualisation project to involve; setting a goal and identifying factors affecting the visualisation, having in-depth knowledge about the data to be visualised, evaluating visualisation design options, and deploying the data visualization.

There is no denying the fact that data is important in our current societies and as equally important are technologies to manipulate, process, and make meaning out of this data. In the scope of this research, 3-D molecules are the data to be visualised in the context of shape matching.

1.1.2 Shape Matching

One of the most challenging tasks due to the exponential growth of data discussed in Section 1.1.1 is shape matching. It is a common task in many problems such as matching MRI (magnetic resonance imaging) scans in medical imaging (Nabavi et al., 2000), reconstructing 3-D shapes from partial scans (Levoy et al., 2000) and protein structure similarity studies in chemistry (Axenopoulos et al., 2011). Shape matching aims to compare the extent of similarity or dissimilarity between two or more shapes. 3-D shape representations usually show the shape of the object as a point cloud in a cartesian coordinate system, which exposes the internal structure of the shape and allows algorithms to superimpose them on other objects more easily than with a knowledge of only the surface of the object.

The major approaches to 3-D shape matching are pixel-based matching and feature-based matching (Belongie, Malik and Puzicha, 2002). Pixel-based approaches use colours and their intensities across regions of the shapes to perform a match based on similarities found in these regions whilst feature-based approaches make use of extracted features of the shapes such as the spatial positions of points and other metadata to perform the match. A well-known shape matching algorithm called the Standard ICP algorithm (ICP) (Besl and McKay, 1992) uses the feature-based approach to perform shape matching (Belongie, Malik and Puzicha, 2002) by representing the shape in the form of a point cloud. The ICP algorithm iteratively computes the closest point in the model point cloud for every point in the data point cloud and then determines the transformation that aligns the two shapes. This iterative process results in minimising the distance between the shapes and aligning the two shapes in a partial or full overlap.

There are several other algorithms used for shape matching and these are reviewed in Section (2.3). The algorithm of concern to this research is the Iterative Closest Point (Besl and McKay,

1992) (ICP). In the context of rigid shapes, ICP registers or matches a data shape to a model or reference shape having a set target rotation and position. As part of the matching process, the algorithm finds the closest point in the model shape for each point in the data shape. A transformation is then computed that reduces the distance between the two shapes. These computations are repeated until the error between the data and model shape are below a set threshold.

1.1.3 Virtual Reality

Another challenge from the exponential growth of data discussed in Section 1.1.1 concerns how to derive insights from the data using current visualisation techniques and technologies such as virtual reality (VR).

VR consists of a combination of software such as 3-D (3 Dimensional) scenes and hardware such as head-mounted displays (e.g. HTC VIVE (HTC, 2020)) working together to give the user a feeling of being present in a real environment. Virtual reality can be used to replace the user's sense of their physical environment with an artificial environment of choice (Steptoe, Steed and Slater, 2013) by enhancing the user's experience in the virtual environment using the components or affordance of VR such as navigation and interaction (Ankomah and Vangorp, 2018).

Virtual reality is characterised by several metrics or components. They include presence, immersion, navigation, interaction, knowledge improvements, performance, and usability (Ankomah and Vangorp, 2018). These components relate to the measurable benefits, outcomes, affordances or what the VR system allows the user to do. Research has suggested that increasing the levels of quality of the individual components of VR have different levels of benefit on the user's ability to feel more present in the virtual environment (Cummings and Bailenson, 2016). As compared to ablation studies, the affordances are primary components of the virtual reality experience and cannot be replaced but rather lowered or increased for further research. For instance, taking away the ability to navigate in some VR systems makes them unusable as it prevents the user from moving to different parts of the virtual environment.

The growth of data and the quest to make increasingly optimised decisions from data has fuelled the drive towards analysing data in VR. VR is also seen as an emerging frontier for visualisation which can preserve much of the data structure and intricacies compared with a 2-D screen. VR helps with mimicking an immersive virtual world of the data, allowing enhanced ability to discover patterns especially in fields such as medicine (Lau, Gupta and Sun, 2021), geology (Yi et al., 2019), space science (Roberts et al., 2015) and environments where high dimensional data are the norm for processing. The difficulties in discovering patterns and associations in high dimensional data can be partially overcome through the techniques for visualisation. For instance, 2-D visualisation does not present enough details for a user to discover as much knowledge as 3-D visualisations. Virtual reality also presents an immersive experience that enables the user to discover even more insights because of affordances such as immersion. VR provides an effective way to integrate the strengths of human beings who are good at abstract inference and modern computing machines that possess the capability of computing millions of times per second.

Foo et al. (2009) explain that the use of virtual reality in medical diagnosis followed paths seen by other fields such as engineering, computational chemistry, and military training and simulation. This shows the obvious increase in using VR in mainstream visualisation. Virtual reality is also touted for its benefits in visualising big data (Olshannikova et al., 2015).

The quest for better ways of processing and visualising data and the challenges associated with understanding and getting meaning from data has steered visualisation research and impact into exploring and representing data in an immersive virtual reality environment (IVR), with ways of interacting with the data. The virtual reality environment enhances the user's ability to realise better insights since the user is virtually immersed in the data. The immersion provided by VR makes patterns discoverable and reveals more discrete patterns, especially within high dimensional data. Aspects of VR such as a higher field of view, higher resolutions and 3-D viewing capabilities have been found to aid with the speed and accuracy of perceptions and interactions in virtual environments (Toet, Jansen and Delleman, 2007, Ragan et al., 2015).

Immersive virtual environments are technically clearly able to show data in detail compared to traditional 2-D displays. Moran et al. (2016) reiterate that utilizing emerging technologies of today to create a fully immersive tool that promotes visualisation and interaction can help ease the process of understanding and representing big data. For instance, Calvelo, Piñeiro and Garcia-Fandino(2020) review existing implementations of VR tools for visualising molecules to help with understanding the SARS-CoV-2 (Covid19) molecule. The research show the importance of the various desktop, web, and HMD based tools for studying and getting insights from molecules. Particularly for Zhavoronkov et al., (2020), a VR tool called Nanome was used to provide medicinal chemistry analysis on designing and validating covalent inhibitors (drug molecules that can bind to the target part or site of a virus molecule) on a particular structure of COVID19 (PDB ID: 6W63). The research used a combination of VR and shape matching to show that VR and shape matching algorithms can also facilitate the registration of drug molecules to binding sites. This can potentially help with drug discovery.

In exploring the integration of visualisation toolkits with immersive virtual environments, O'leary et al. (2017) highlight the growing body of evidence that demonstrates the measurable benefits that can be attained when exploring scientific data using immersive interfaces. The research found that the consumerisation of virtual reality and especially its application to non-gaming VR provides an opportunity for scientific visualisations to tap into the affordances of VR (e.g., immersion, knowledge improvements) to gain better insight. The research developed plugin extensions to the Visualisation Toolkit (Schroeder, Martin and Lorensen, 1996) that provide functionalities for loading 3-D objects to be rendered in VR.

This research uses the iterative closest point (ICP) algorithm (Besl and McKay, 1992) and three proposed variants (KmeansICP, SubsetICP+, and TaggedICP (Ankomah et al., 2020)) developed to register 3-D molecular shapes. The algorithms are evaluated in a desktop computer environment. A virtual reality environment is also presented for simulating and visualizing the shape matching.

1.1.4 Molecules and Protein Structures

This research does not primarily focus on the intricacies of molecules, proteins, atoms, and their structures. However, knowledge from aspects of protein structures is relevant for this research as it forms part of the consideration for the development of the shape matching algorithms. This section will attempt to give the reader some basic understanding of how important and complex these molecular entities are concerning their identification and interaction, functions and structural applicability for education, research, drug discovery and disease management and control.

Berman et al., (2000) explains that cells build many complex molecular structures that perform different functions together to ensure the organism they inhabit (multicellular organisms including humans) continues to function. For instance, some cells break down food (cells in amylase enzyme found in saliva breaks down sugars in carbohydrates to glucose), whilst others digest food (pepsin in gastric enzyme, produced in the stomach) used to build new cells to replace dying ones. Other cells help against diseases (cells in blood platelets produced in the bone marrow). Most of these cells form molecular structures that are made up of proteins and nucleic acids and the 3-D structure of the protein molecule shows how these structures are built and how they fold into molecular machines for performing their tasks (fighting diseases and building new cells) (Berman et al. 2000). For purposes of this work, the internal structure of the atom was not considered, but rather the structure of the molecule formed by multiple atoms.

Each molecule is made up of a group of atoms forming a chain and the structure of that chain determines the function of that molecule. A molecular biologist studies the structure of proteins in 3-D to get a deeper insight into structural changes in the molecule caused by mutations, evolutions, or even binding sites of a protein. The knowledge from such a study helps to understand occurrences such as viral mutations (viruses changing their molecular structure to form new variants e.g. variants of the COVID-19), drug resistance (bacterial and viral disease cells helps develop the ability to defeat drugs designed to kill them e.g. resistance of some antibiotics and malarial drugs (Hemlata and Tiwari, 2015)) and disease control at the molecular level.

1.2 Research Problem

The iterative closest point algorithm (Besl and McKay, 1992) has far-reaching uses which has seen interest from researchers to analyse and explore its strengths and limitations (Wang and Zhao, 2017). The limitations of ICP include the requirement for a good initial transformation to ensure the algorithm converges optimally, the high cost of the correspondence search method using a brute force approach to finding the closest point, and the algorithm not converging at large rotation angles (Besl and McKay, 1992).

Specifically for 3D protein shapes, partial or full registration are necessary to study similarities between two molecules, e.g two viruses or part of a vaccine bonding to a viral protein shape. This requires fast and efficient algorithms to match the often thousands of 3D points.

1.3 Motivation

The choice of protein molecules as the dataset for this research was due to its complex high dimensional nature posing inherent visualisation challenges especially with visualising the development, mutations, and interactions of protein molecules. For instance, in the study of the interaction of COVID19, virtual reality has been used to gain more insight into the behaviour of the viral protein which can aid in drug discovery (Calvelo, Piñeiro and Garcia-Fandino, 2020). The 3D shapes of molecules are registered to study their similarity and help with tasks such as development of medicines and vaccines.

For instance, a typical red blood cell contains 280 million molecules (Berman, et al., 2000). In perspective, each molecule is made up of several numbers of atoms forming a chain group. The researcher also has some foundation and interests in computational biology. Even though this research was primarily interested in matching 3-D molecules with a few hundred atoms, the challenges realised will be more prevalent in 3-D datasets with thousands or millions of atoms or points.

Several approaches have been used to detect structural and functional relationships amongst proteins. The earliest algorithms were based on sequence information (Schmitt, Kuhn and Klebe, 2002) such as comparing pairs of amino acids (molecules) in a protein structure (Needleman and Wunsch, 1970). 3-D protein structural comparisons require more sophisticated algorithms to capture, visualise and match the structures based on spatial coordinates of the atoms. 3-D proteins can be represented as rigid objects where the interpoint distances will not change from one view to another. A rigid transformation can be found to align them; however, more reliable matches would be attained if the coordinates are associated with some predefined properties (Schmitt, Kuhn and Klebe, 2002) such as the atom types or sequences.

The presence of outliers (non-uniform points) can also affect alignment with ICP (Chen and Belaton, 2014). Another notable problem is that ICP performs better in some datasets and contexts and worse in others. For instance, in registering terrain map data, Donoso et al. (2017) finds that shape matching algorithms can better make use of the metadata of the shapes to improve the match. Other features could also be extracted to aid in the matching. These suggestions require research and experimentation in shape matching algorithms to evaluate and validate the use of additional features.

No matter what algorithms are used, it is always likely for them to fail due to different complexities of geometry, magnitudes of transformation, sizes of overlap and distributions of outliers. In this case, a human intervention must be involved in one way or another using techniques such as virtual reality to visualise and interact with the data.

Virtual reality, as a new visualisation frontier presents opportunities for new research in 3-D User Interface (3-DUI) developments that help judge whether the registration of given overlapping shapes is a success and provide intervention if necessary. Further research informs the evolution of the technology whilst considering different domains, tasks, and kinds of data for visualisation in virtual reality to be realised. VR can allow humans to intervene in by adjusting the initial transformation and shapes to make sure that the shapes can be brought into high-quality alignment aided by the affordances of virtual reality such as immersion, presence, and navigation.

1.4 Aims and Objectives

The research aims to develop, evaluate and visualise shape matching algorithms based on the Standard ICP algorithm (Besl and McKay, 1992) that explores different techniques for matching 3-D molecules and visualising the matching in virtual reality.

The objectives of the research are:

1. Gain a critical understanding of similarities, differences, and potential areas for improvement in existing shape matching techniques by performing a thorough literature review.

2. Gain a critical understanding of the history, current state, and challenges of virtual reality through a literature review.

3. Develop and evaluate shape matching algorithms based on the Standard ICP algorithm.

4. Develop a virtual reality visualisation software for visualising and interacting with 3-D molecular protein data, allowing exploration of the molecule in immersive virtual reality environments.

1.5 Original Contribution

The research has reviewed shape matching algorithms and virtual reality. The review has shown that data visualisation gains better insight with immersive virtual reality. The novel nature of VR for data visualisation means that applications such as Iterative Closest Point Virtual Reality Application (ICPVR) contributes to advancing data visualisation in virtual reality. This is achieved with the development of an application that allows details of a shape matching process to be observed and interacted with. The research has also developed and evaluated three variants of the ICP algorithm using different techniques to improve the matching process.

Contribution 1: Literature

The research has contributed to existing knowledge on virtual reality by reviewing, identifying, and classifying virtual reality according to identified metrics in "*Virtual Reality: A literature review and metrics-based classification" (Ankomah and Vangorp, 2018)".* The publication is a good knowledge base for virtual reality researchers to understand the core affordances of virtual reality. The article combines applications and use-cases of virtual reality with research agenda for progressing the technology. The reviewed limitations of virtual reality also show researchers what areas of the technology need to be addressed.

The research has also contributed to 3-D shape matching literature and particularly research on ICP algorithms, with an extensive review on shape matching algorithms and the development and publication of a variant of ICP algorithm for matching the 3-D structure of proteins in *"TaggedICP: An Iterative Closest Point Algorithm with Metadata Knowledge for Improved Matching of 3-D Protein Structures" (Ankomah et al., 2020)".*

Contribution 2: Algorithms

The research has developed and evaluated 3 variants of the iterative closest point algorithm.

KmeansICP : An ICP algorithm variant, implementing k-means (Krishna and Narasimha Murty, 1999) partitioning to reduce the correspondence search space and improve its computational efficiency.

SubsetICP+ : An extension of SubsetICP (Chen, Belaton and Pan, 2013) using atom types for partitioning and improving the match quality by selectively skipping some transformations for improved performance and efficiency.

TaggedICP (Ankomah et al., 2020): A variant of the ICP algorithm that uses metadata information or tag information (atom type and neighbourhood information of the point) for partitioning the correspondence search space. This can be used to improve shape registration when considering the use of some metadata. TaggedICP demonstrates how to use metadata information to aid in the matching of 3-D structures for improved robustness.

Contribution 3: Shape Matching Virtual Reality Application

The research has also developed a virtual reality application using Unity3D (Unity Technologies, 2019) and Microsoft C# that allows immersive visualisation, comparison, and evaluation of 3-D shape matching algorithms. The Unity application, Iterative Closest Point Virtual Reality Application (ICPVR), allows a user to use compatible VR headset and hand controllers to easily interact with the 3-D shapes (setting initial transformation) before performing a shape match. ICPVR also allows any shape matching algorithm or 3-D shape to be loaded and visualised in Unity3D with some level of configuration.

ICPVR can help visualise the whole registration process and intervene, when necessary, by providing more accurate initial transformation and adjusting the registration results, so that the algorithm finally converges to the global optimal solution through both qualitative judgement and quantitative measurements. This process has the advantage of combining the strengths of both humans and computers in cognitive judgement and powerful computation.

The development of the virtual reality shape matching environment provides an application that enables other researchers to do further user experimentation on the three-dimensional (3-D) user interface as well as improve the base application with several other features.

1.6 Methodology

1.6.1 Introduction

This research developed three (3) variants of the ICP algorithm using techniques such as k-means clustering which uses the k-means technique to partition the model search space and reduce the computational cost, the subset technique for registering corresponding pairs of subsets from the shapes to be matched and using known information to add a constraint to make the matching more meaningful.

A virtual reality application, called ICPVR was also developed to allow the shape matching process to be visualised, allowing the user to be immersed in viewing the overlaps between the shapes.

Details of the research methodology consisting of the domain analysis, selection of methods and tools, solution development, and evaluation are discussed in the next sections.

1.6.2 Domain Analysis

The research was placed in the right context by highlighting the problem of ICP and extensively reviewing relevant literature on shape matching and virtual reality. This stage was very important as it ensured a better understanding of the related research work, scope, challenges, and limitations. This understanding was very important in achieving the project objectives.

The focus of the research was on shape matching with the Iterative Closest Point Algorithm (Besl and McKay, 1992), variants of ICP, and visualisation in virtual reality. The research was interested in developing different variants of the ICP algorithm that improves the performance and quality of a match over standard-ICP on 3-D molecular structures as well as an application to visualise the matching in immersive virtual reality.

Data visualisation is an important aspect of this research. Particularly, the research was interested in the advanced frontiers of visualisation which has virtual reality as an important platform for creating and presenting visuals that overcome the barriers of traditional visualisations such as display size and interactivity.

3-D user interfaces are constantly being evaluated for their effectiveness in different visualisations, as such, this research's focus on a VR application to match 3-D shapes embodies the core objectives of the research and aligns the analysis in a way to achieve the objectives.

1.6.3 Methods and Tools

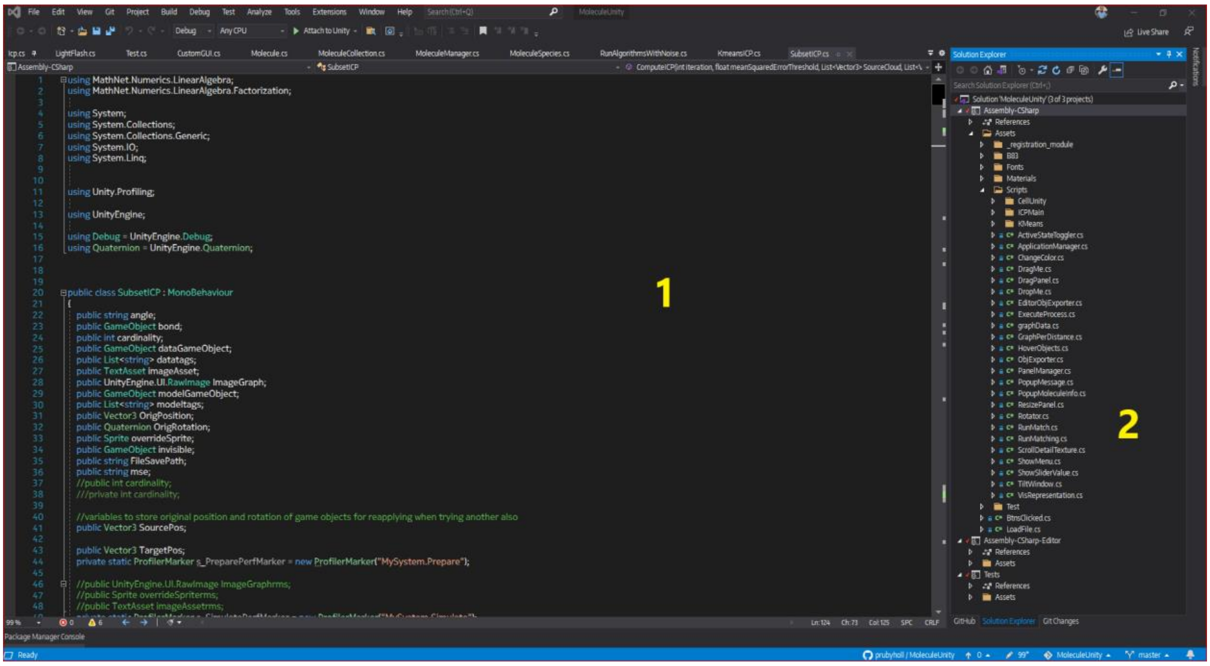
An extensive literature review on 3-D shape matching and virtual reality was performed. Several implementations of ICP were reviewed. Early stages of the research found that enhancements had to be made to the ICP algorithm to improve some steps of the algorithm and to solve the identified problems of shape matching with ICP and achieve the aims and objectives of the research. The options available were to review and adapt an existing implementation of ICP suitable for matching shapes of 3-D protein structures or build an implementation from scratch using the original ICP algorithm (Besl and McKay, 1992).

The research reviewed some ICP implementations developed using C# and C++ so that they can easily be modified and visualised in the Unity3D environment. The approach was largely interested in implementations that can be modified to include changes to different steps of the algorithm. Libpointmatcher (Pomerleau et al., 2013) is an implementation of ICP used in robotics, developed in C++. Libpointmatcher was initially reviewed and tested. It was able to successfully match shapes. The source codes were available, and the research needed to modify the Libpointmatcher source codes and recompile them. Recompiling in C++ always showed errors with multiple dependencies. The codebase was deemed high risk to use, as it was not clear whether it would work properly.

Pointmatcher.net was also reviewed. Pointmatcher.net is an implementation of ICP using the Libpointmatcher binaries but developed in C#. It had a minimal set of filters available and was less extensively tested and optimised. This implementation was also deemed less ideal.

Other smaller libraries were reviewed in quick succession and the best course of action was found to be developing an implementation of ICP from scratch using the original ICP algorithm (Besl and McKay, 1992). This approach meant that the evaluation of the developed algorithms will otherwise not be affected by existing changes which will maintain its simplicity.

Another condition that existing implementation had to meet was to be able to work in Unity3D (Unity Technologies, 2019). Unity3D was selected as the implementation and visualisation environment because amongst other things, it supported the target programming language (C#). Unity3D is a popular game engine. It presents a development environment with a coordinate system and several primitive shapes. The Unity3D application allows scripts to be written in C# to connect to inbuilt APIs that enable interaction with objects by applying transformations to rotate, translate, scale, and reflect objects. The environment also provides a rich toolset for creating 3-D virtual environments with interactable objects.



```
using MathNet.Numerics.LinearAlgebra;
using MathNet.Numerics.LinearAlgebra.Factorization;

using System;
using System.Collections;
using System.Collections.Generic;
using System.IO;
using System.Linq;


using Unity.Profiling;

using UnityEngine;

using Debug = UnityEngine.Debug;
using Quaternion = UnityEngine.Quaternion;



public class SubsetICP : MonoBehaviour
{
    public string angle;
    public GameObject bond;
    public int cardinality;
    public GameObject dataGameObject;
    public List<string> datatags;
    public TextAsset imageAsset;
    public UnityEngine.UI.RawImage ImageGraph;
    public GameObject modelGameObject;
    public List<string> modeltags;
    public Vector3 OrigPosition;
    public Quaternion OrigRotation;
    public Sprite overrideSprite;
    public GameObject invisible;
    public string FileSavePath;
    public string mse;
    //public int cardinality;
    ///private int cardinality;

    //variables to store original position and rotation of game objects for reapplying when trying another also
    public Vector3 SourcePos;

    public Vector3 TargetPos;
    private static ProfilerMarker s_PreparePerfMarker = new ProfilerMarker("MySystem.Prepare");

    //public UnityEngine.UI.RawImage ImageGraphrms;
    //public Sprite overrideSpriterms;
    //public TextAsset imageAssetrms;
```

```
Solution 'MoleculeUnity' (3 of 3 projects)
  Assembly-CSharp
    References
    Assets
      _registration_module
      B83
      Fonts
      Materials
      Scripts
        CellUnity
        ICPMain
        KMeans
        ActiveStateToggler.cs
        ApplicationManager.cs
        ChangeColor.cs
        DragMe.cs
        DragPanel.cs
        DropMe.cs
        EditorObjExporter.cs
        ExecuteProcess.cs
        graphData.cs
        GraphPerDistance.cs
        HoverObjects.cs
        ObjExporter.cs
        PanelManager.cs
        PopupMessage.cs
        PopupMoleculeInfo.cs
        ResizePanel.cs
        Rotator.cs
        RunMatch.cs
        RunMatching.cs
        ScrollDetailTexture.cs
        ShowMenu.cs
        ShowSliderValue.cs
        TiltWindow.cs
        VisRepresentation.cs
      Test
      BtnsClicked.cs
      LoadFile.cs
  Assembly-CSharp-Editor
    References
    Assets
  Tests
    References
    Assets
```

Figure 1.1 Visual Studio 2019 Community code view of the Unity3D project. (1) Code editor showing the currently opened script. (2) Solution view showing the hierarchy of scripts and their folders.

Figure 1.1 shows the Visual Studio 2019 code editor used for developing and debugging the shape matching algorithms, and visualisation and animation scripts. It presents an interface for writing code elements and debugging errors amongst other features.

Based on the limitations identified from the research and testing done with some existing implementations of ICP, the research developed the code from scratch using the original ICP algorithm (Besl and McKay, 1992) publication. The rationale was to allow complete flexibility in experimenting and enhancing different parts of the algorithm. It was also to give an in-depth experience to the researcher compared to the experience realised in implementing an existing codebase. This method was also a better way of objectively measuring the proposed enhancements to be made to the ICP algorithm compared with implementations that would have already enhanced the algorithm in different areas. The idea was not to pick the fastest ICP algorithm out there and improve it, but rather to understand the algorithm and develop several enhancements to the most time-consuming methods for matching and visualising 3-D proteins.

The research also aimed at designing and developing a utility virtual reality application that can be used to visualise a shape matching process using the ICP variant algorithms in virtual reality. This aspect of the research was meant to improve 3-D user interface research and enable the user to interact with the 3-D objects whilst being immersed in the virtual environment as well as investigate how the overall shape matching can be improved with virtual reality visualisation.

1.6.4 Solution Development

The solution development phase considered the domain analysis and methods and tools to come up with an approach to the research objectives and the identified gaps from the literature review.

Three (3) variants of the ICP algorithm were developed that primarily focused on techniques such as reducing the number of corresponding searches by optimising the partitioning and making use of additional metadata available to aid in the matching. These changes were aimed at improving the overall performance of the ICP algorithm in both accuracy and efficiency.

The search for correspondence method of the ICP algorithm (Besl and McKay, 1992) was targeted as a part of the algorithm to improve because it has a high computational cost ($O(N_x N_p)$ for N_p points in the data shape and N_x points in the model shape). This cost increases with the increasing number of points. This is because, for each point in the data shape, the corresponding closest point in the model shape must be found. The high computational cost slows down the algorithm with the increasing number of points in the shapes. This slowdown can largely affect the performance and applicability of the algorithm and especially the real-time interaction in VR when the number of points in the shape increases. This means that optimisations made to the search for correspondence will have a critical impact on the overall performance of the algorithm by reducing the computational cost.

The limitations of ICP motivates the development of its variants. For instance, TaggedICP (Ankomah et al., 2020) uses the metadata (tag) information of the point to improve the matching by finding corresponding points based on matching tag information. This makes the matching meaningful as well as reduces the corresponding search space by limiting the search space using atom type partitions.

The limitations of ICP also motivated the focus on partitioning strategies for the developed variants of the algorithm. All the algorithms made use of some sort of partitioning because it reduces the number of points processed in the corresponding search method. Using only a subset of points in the corresponding search space improves the performance of the algorithm by reducing the time taken for the correspondence search method to complete for each iteration. SubsetICP+, KmeansICP and TaggedICP (Ankomah et al., 2020) partition the points differently to reduce the correspondence search space.

1.6.5 Evaluation

A virtual reality application, Iterative Closest Point Virtual Reality (ICPVR) was developed using Unity3D (Unity Technologies, 2019) and Microsoft C#. The application allows the different ICP algorithms developed as part of the research to be evaluated for a comparison of their improvements over the standard ICP. The visualisation in VR also allows the qualitative evaluation of 3-D user interfaces using the software for testing shape matching algorithms.

The desktop version of ICPVR was used to evaluate the performance of the algorithms and generate performance data on the algorithm such as the details of convergence and the match quality of the shape matching. The data was then used to plot the analysis graphs. The research was particularly interested in the number of iterations each algorithm took to converge at different initial rotation angles as well as the quality of the match in terms of the final distances between corresponding points. It was also interested in the animation of the shape matching process showing the evolution of the shape over the iterations. The registration process was real-time meaning the user instantly gets to see the updated data point cloud after a computed transformation has been applied to it on both desktop and VR. The user is then able to see the shapes transformed after each iteration is complete in a seamless animation.

The research was interested in showing how the convergence and match quality data were able to show whether algorithmic changes made to the developed variants show performance improvements in terms of faster convergence or a better quality of the match. This was done with a combination of convergence graphs, match quality graphs and final matched shapes of the point clouds.

The research was also interested in how the shape matching visualisation will perform in virtual reality and particularly what tweaks or adjustments needed to be done to ensure a smooth visualisation and registration of the structure of 3-D proteins. The VR application provided an environment where interaction with the 3-D protein structures was enhanced compared to the use of a traditional mouse and keyboard. The use of natural hand gestures using the VR hand controllers provided a seamless way of interacting with the 3-D proteins that mimicked the use of natural hands. The research was interested in understanding how the affordances of VR such as immersion provided by the VR application benefits the task of shape matching as well as how the 3-D UI helped with the quick intervention, evaluation, and comparison of the shape matching algorithms.

1.7 Thesis Structure

Each chapter of the thesis was written to focus on its core discussion, remaining as independent as possible from other chapters but still showing a gradual link between them.

Chapter 1: Introduction

The introduction chapter gives the reader the background of the research. It introduces key terminologies and concepts that are central to the research. It explains the research motivation, aims, objectives, methodology, and how the research objectives have been met.

Chapter 2 – Literature Review on Shape matching

The shape matching chapter discusses the concept of shape matching. It discusses, compares and contrasts the different shape matching algorithms before delving into the algorithm of interest to this research, the standard ICP algorithm (Besl and McKay, 1992).

Chapter 3 – Literature review on virtual reality (VR)

The virtual reality chapter evaluates virtual reality. It discusses the history of virtual reality as well as the latest advances in virtual reality technology. The chapter exposes the reader to several applications and use-cases of virtual reality. The research also further explains the metrics or affordance of virtual reality.

Chapter 4 - Proposed ICP Variant - KmeansICP

This chapter discusses KmeansICP. It highlights the design and computational steps involved in the algorithm as well as the computational complexity.

Chapter 5 - Proposed ICP Variant - SubsetICP+

This chapter discusses Subset-ICP+. It highlights the design and its computational steps involved in the algorithm as well as the computational complexity.

Chapter 6 - Proposed ICP Variant - TaggedICP

This chapter discusses TaggedICP. It highlights the computational steps involved in the algorithm as well as the computational complexity.

Chapter 7 - Experimental results and analysis

This chapter details the tests done to compare the ICP algorithm variants. It shows the different match quality and convergence graphs and provides a detailed interpretation of the graphs. It compares whilst providing a critical discussion on the effects of the techniques used in the algorithms.

Chapter 8 - Iterative Closest Point Virtual Reality Application (ICPVR)

This chapter discusses a virtual reality application, ICPVR that allows the developed variants of the ICP algorithms to be visualised in registering 3-D protein datasets.

Chapter 9 - Conclusion and future work

This chapter concludes the thesis by summarising the work done and contrasting it with the aims and objectives of the research. Suggestions for future work are also detailed in this chapter.

1.8 Publication List

The research has published 2 outputs on virtual reality and ICP algorithm respectively.

Virtual reality:

ANKOMAH, P. and VANGORP, P., 2018. Virtual Reality: A literature review and metrics-based classification. Proceedings of EG UK Computer Graphics & Visual Computing, pp. 173-181, 2018

ICP Algorithms and 3-D Shape Matching:

ANKOMAH, P., VANGORP, P., BEHERA, A., and LIU, Y., 2020. TaggedICP: An Iterative Closest Point Algorithm with Metadata Knowledge for Improved Matching of 3-D Protein Structures. Proceedings of the Irish Machine Vision and Image Processing Conference, pp. 37-44, 2020

1.9 Summary

The introduction chapter has detailed the scope of the research. It has highlighted the research's focus on 3-D shape matching and visualisation in immersive virtual reality. The research has developed and evaluated shape matching algorithms based on the Iterative Closest Point Algorithm (Besl and McKay, 1992). It has also developed a virtual reality application for shape matching that can be used for quickly interacting and matching 3-D shapes.

PART 2: LITERATURE REVIEW

2 CHAPTER 2 SHAPE MATCHING

2.1 Introduction

This chapter presents a review of shape matching to advance the user's knowledge in this area. It discusses shape matching especially in the context of matching 3-D shapes. It details some techniques used in shape matching algorithms such as the use of neural networks and additional metadata. It also highlights challenges and possible solutions for the reviewed algorithms. It then focuses on the Iterative closest point algorithm (Besl and McKay, 1992). The strengths and limitations of different variants of the ICP algorithm are also discussed as well as its use in different industries such as Medicine for medical image analysis (Sinko et al., 2018), Education, and other fields. This chapter forms the base knowledge for exploring different techniques in the variants of ICP developed and it attempts to achieve objective two (2), "*Gain a critical understanding of similarities, differences, and potential areas for improvement in existing shape matching techniques by performing a thorough literature review".*

2.2 Background

In the area of 3-D protein structural comparison, Bender et al., (2004) proposed an Atom Environment based approach to molecule similarity searching that combines using atom descriptors such as volume and 3-D geometrical information with a selection of atom features such as fingerprints (e.g. signature of the molecule based on the count of a particular atom), using a distance function (e.g. Manhattan distance), and a Naïve Bayesian Classifier. The approach was evaluated on the MDDR (MDL Drug Report) dataset on a series of similarity search tests, and it showed strong performance in terms of fast retrieval of molecules from the MDDR database.

Similarly, Eckert and Bajorath (2007) reviewed the performance of molecule similarity search-based algorithms. The research discussed approaches such as fingerprints (representing molecular structures and properties using a sequence of bits), 3-D similarity methods (e.g., shape matching algorithms), and descriptor-independent similarity methods (e.g., Histogram comparisons). The research found that molecular similarity searches are important in many forms of designing or identifying compounds and there is a need for novel algorithms to be designed especially for navigating high-dimensional molecules and detecting relationships between them. This is also necessary for drug design.

Zhou, Jiang and Zheng, (2009) explained that similarity matching on the structure of protein molecules is important because the structure preserves more information than the sequence and investigating the structures can help discover insights into the functions of the proteins. The research proposed an angular invariant (not affected by the angle of rotation) feature for the registration of 3-D protein structures. The angular invariant feature vector is composed of the normal of a data point and its k-nearest neighbour. This approach is invariant to translation, rotation, scale, and affine transformations presenting some strengths over the standard ICP algorithm, however, the limitations are not clearly tested, for instance, how the algorithm performs in the presence of outliers and noise.

Another approach to 3-D protein structural comparison involves modelling the structure using a technique called Triangular Spatial Relationship (TSR) (Kondra et al., 2021). TSR uses triangles to represent substructures of the protein shape by selecting a representative atom, computing all possible triangles that can be formed out of the atom, and calculating a key for each triangle. This approach allows shape comparison without superimposing the two shapes but rather comparing the computed keys to detect similarities in substructures between shapes. TSR has advantages over approaches that use atom types, their coordinates, and atomic distances to find similarities between regions of the protein structure by allowing flexibility in the comparisons, for instance, structural changes can alter certain regions of a protein shape. In the atomic distances approach, this will result in a comparison returning dissimilarity between these regions, however, using TSR, these regions (similar but not identical) can be assigned identical keys which goes a long way to show some inherent similarity.

Other similarity tests for 3-D protein structures include the Jaccard/Tanimoto coefficient or statistical significance measuring similarity between two organisms by comparing the ratio of their intersection to their union to identify the presence or absence of biological data (Chung et al., 2019) and (Tabei and Puglisi, 2017).

Registration (matching two shapes) involves matching two shapes so that they can be overlaid on each other. The aim of registration is also to best align a shape or multiple instances of shapes to have them in a partial or fully overlapped position. This can also be seen as finding an optimal transformation that can align two views of an object (Chen and Medioni, 1992).

Usually, registration involves attempts at matching the data shape (reference) and the model shape (template or target). In this project, only rigid 3-D protein shapes were considered. In this case, the underlying transformation that aligns one shape with the other can be modelled as rigid, consisting of a rigid rotation matrix and a translation vector. The interpoint distances between atoms in the overlapping area will not change from one dataset to another. The registration process usually involves processes of computing and applying optimal translation and rotation operations on the data shape repeatedly until it aligns well with the model shape.

Agarwal and Bhowmick (2017) proposed an approach to shape registration that improves on the existing Gravitational Approach (GA) to point cloud registration (Golyanik, Ali and Stricker, 2016). GA assumes that each point in the data point cloud attracts each point in the model point cloud using a gravitational force based on properties of the point and its parent point cloud, for instance, the centre of mass, the mass of the point cloud, and velocity of the previous iteration. The performance of GA was found to decrease with increasing rotation of more than 45 degrees since the main principle of the GA is to minimise the distance between the centre of mass of the shapes. Agarwal and Bhowmick (2017) focused on unifying two approaches (GA approach and the usual objective of registration) by combining the minimisation of the distance between the centre of mass of the two shapes and aligning the two objects in an overlap position. The research found that the improvements to GA were robust against the mass of outliers so that the registration process becomes more dependent on the shape than on the mass of the shape.

2.3 Shape Matching Algorithms

One of the earliest accounts of shape matching was by Fischler and Bolles (1981). The research focused on object location and recognition in 3-D space for robotics and navigation. The research further developed a method called the Random Sample Consensus (RANSAC) for matching a template to experimental data. This method is based on the maximum consensus registration method that solves the minimisation problem by finding the largest set of correspondence that match a generative model (a model of the conditional probability of one model from another model). The method is noted to work well with data containing some errors and thus suited for processes such as automated image analysis with images having some errors. RANSAC works by comparing one model to a small subset of the data at a time whilst increasing this subset to cover large portions of the model or scene.

RANSAC has been applied to the shape registration problem in several pieces of research. Hossein-nejad and Nasri, (2016) explored the use of an improved RANSAC algorithm in the object registration process to eliminate mismatches between corresponding features. The research used the Scale-invariant feature transform (SIFT) method to match features in two images. SIFT can effectively identify features in images with noise, varying light intensities, rotations, and scale changes. SIFT is also known to suffer from large numbers of mismatches in the image registration process and RANSAC had been identified as one of the most applicable methods for removing the mismatches (Fischler and Bolles, 1981). Hossein-nejatd and Nasri (2016) identified that changing the threshold value (the number of matching points needed to assume data and model are a match) of the original RANSAC algorithm resulted in different and often worse matching rates and mismatches. The research showed that setting RANSAC to a threshold of 1 (meaning images are a match when there is a match between one feature in both images) resulted in the algorithm eliminating mismatches as well as some correct matches. An improved RANSAC algorithm (Adaptive RANSAC) is treated as a classification problem by considering the mismatches and correct matches in determining an adaptive threshold value. The research found that the adaptive RANSAC algorithm eliminated a small number of matches alongside the mismatches which is an improvement in using the original RANSAC algorithm for removing mismatches.

RANSAC has also been applied to registering partially overlapping 3-D range images (Chen, Hung, and Cheng, 1999). Chen et al. (1999) extended the RANSAC algorithm by suggesting the data-aligned rigidity-constraint exhaustive search (DARCES). DARCES improves on RANSAC by not requiring any pre-processing nor initial estimates for registering 3-D shapes. It begins the registration process by randomly selecting a primary control point, then the secondary and auxiliary points are selected in such a way that three (3) control points form an acceptable minimal triangle. Other control points are also selected such that they form a bigger triangle around the acceptable minimal triangle. The researchers experimented with capturing four range images (Images from a sensor device that usually represents the distance from a point to a particular scene using pixels) having a total of 4200 points each and belonging to two (2) separate images and registering into a 3D object. The RANSAC-based DARCES was able to register both images in a process that checked all possible data alignments of the datasets without requiring any additional features nor consideration of a good initial estimate. However, DARCES was not found to always be suitable for registering partially overlapping images with more than three (3) control points because of the potential of using control points found outside the overlap region due to the complexities and sizes of a shape.

RANSAC has been extended to other shape registration such as 3-D point cloud plane segmentation using Normal Distribution Transformation of Cells (NDT - assigning a normal distribution to a cell which models the probability of measuring a point). This technique divides a point cloud into planar and non-planar grids for registration (Li et al., 2017). More recently, RANSAC has been improved with a technique that reduces the randomisation of the hypothesis set (Rahman, Li and Yin, 2019).

Faugeras and Hebert (1986) focused on recognising and locating 3-D objects from range data and other 3-D models. The research identified rigid shapes to be best represented with linear primitives and recognition and locating objects in 3-D space to be a major useful task.

Mora et al. (2016) explained that registration methods can be applied to both rigid and non-rigid shapes. Rigid shapes are shapes that do not change their structure or scale in the presence of transformations because the whole shape is affected by the same transformation. Rigid shapes can only be translated to a different place or a different orientation. Non-rigid shapes are however flexible, and their structures and scale can be changed with the application of different transformations.

Rigid transformations such as rotations and translations are applied to rigid shapes with invariant interpoint distances in the overlapping areas of the given shapes whilst non-rigid transformations such as scaling are applied to non-rigid shapes. Non-rigid transformations usually change the structure of the shape applied. For instance, scaling transformations change the size of the objects, making them bigger or smaller. Shear transformations also push objects towards one side. Non-rigid transformations usually consider a larger solution space because of the possibility of deformations due to their non-uniform transformations.

Shape matching algorithms can be classified into coarse and fine-grained. Coarse-grained registration is used to provide a rough and quick estimate of the matching whilst fine-grained is used to iteratively register shapes more accurately (Mora et al., 2016).

It is also noted that different types of registration methods may work in different domains. For instance, when dealing with image data registration, it is suggested that multiple algorithms can be integrated into a system that can decisively adopt a particular mixture of techniques to register them (Guan et al., 2020).

Brown (1992) explained that all registration methods consist of varying implementations of four (4) basic components, feature space, search space, search strategy and a similarity measure. The feature space consists of implementations for extracting information that will be used for the matching. This information (features) includes spatial coordinates of points, orientation features and other available metadata.

The search space is the group of transformations that can be used to align the shapes. For instance, rigid transformations such as translation and rotation can be used to align rigid shapes whilst non-rigid transformations such as scaling can be used to align non-rigid shapes. The search strategy refers to how the next transformation is to be chosen from the applicable space (rigid or non-rigid transformation) including the considerations to be made in choosing the transformation to be tested for the optimal transformation before applying to the shape.

The similarity measure evaluates each test (each series of applying transformations) and determines the extent of the match based on the value of the next transformation. For instance, if the next optimal transformation to align the two shapes is the zero transformation, the results will be deemed to be in a fully overlap position. Another similarity measure is the error metric that computes the misalignment error between the two shapes and deems them a match when that error metric is appropriately small.

Myronenko and Song (2010) also explained that certain qualities are necessary for a workable registration algorithm including the ability to compute the transformation to align the shapes flexibly and easily, the ability to work with high dimensional data sets, and the ability to effectively handle noise and unwanted points. In today's multifaceted data, the ability of registration algorithms to register partial shapes is an added strength.

The ability to compute the transformation is basic to all registration algorithms as transformations are the property required to progressively align the two shapes. Aligning these shapes flexibly and easily allows this process to be easily modified based on changing conditions and often complex requirements. The ability to handle noise and unwanted points makes the shape matching algorithms more robust.

A registration algorithm's ability to work with high dimensional datasets allows it to be used in current registration challenges, often involving complex data such as 3-D medical objects. Also, effectively handling noise, outliers and deformations are common in today's shape registration because of the increasing complexity of the available data, applications and in some instances presence of noise.

With a focus on fine-grained registration algorithms that register shapes more accurately and depending on the transformations required and the complexity and structure of the data, different algorithms are usually considered (Temerinac-Ott, Keuper and Burkhardt, 2010). These algorithms may include robust point matching (Gold et al., 1998), thin plate spline robust point matching(TPS-RPM) (Chui and Rangarajan, 2003), kernel correlation (Tsin and Kanade, 2004), Gaussian mixture model (Jian and Vemuri, 2011), sorting the correspondence space (Assalih, 2013), Bayesian coherent point drift (Hirose, 2020), probability density estimation based Coherent Point Drift (CPD) (Myronenko and Song, 2010) and the Iterative Closest Point (Besl and McKay, 1992).

The robust point matching method (Gold et al., 1998) was designed to work effectively on 2-D or 3-D point sets, rigid or non-rigid shapes, with noise. It is a fast, robust, and flexible algorithm that makes use of a combination of different optimisation techniques such as softassign (Kosowsky and Yuille, 1994, Rangarajan, Chui and Bookstein, 1997) also referred to as “soft correspondence” (each point in one point cloud corresponds somehow to every point in the other set by some weight) which has emerged from neural networks. Robust point matching works by minimizing an analogue objective function (cost function) describing the registration problem. The cost function is constructed for 2-D and 3-D registration problems, and it is structured in a way that uses match matrices to represent correspondence of one set of points to another. Softassign is noted to be slow and complex especially for registering thousands of overlapping 3-D points (Parekh et al., 2015). This limitation was improved for efficiently registering sparse and dense points in a free form shape by using one-way or two-way closest points constraint in conjunction with mean field annealing (Liu, 2006). The one-way refers to the traditional mapping of one point in the data shape to another point in the model shape, whilst two-way constraint correctly assumes that the point in the model point cloud is also a correspondent in the data point cloud. If both sides of the point clouds are considered this way, it results in a complex two-way constraint. The mean field annealing was found to be more computationally effective than the rigid constraints for evaluating possible point matches. One-way constraint matching is often less computationally expensive than two-way constraint matching.

Chui and Rangarajan (2003) proposed a feature-based general-purpose non-rigid registration method, thin-plate spline robust point matching for non-rigid shape registration that can successfully register features whilst rejecting outliers (non-uniform) points appropriately. This method was based on robust point matching (RPM) (Gold et al., 1998) and spline-based deformations (Bookstein, 1989, Wahba and Pennsylvania, 1990). The research developed the thin-plate spline robust point matching algorithm because it can be considered as a natural non-rigid extension of the affine transformation.

Kernel correlation (KC) (Tsin and Kanade, 2004) was an extension of using correlation to match point sets. Correlation involves using some appearance information of the shape to align objects (2 or more shapes with different colour intensities, usually, same shapes taken from different cameras or sensors). The research found that correlation is not very good at matching shapes given only the point sets. Attempts to do this involves treating the structure as binary intensity images which are limited to representing them as 1 (structure present) or 0 (structure absent). This technique means that when noise is present, the images would not match, as such kernel correlation, as a similarity measure seeks to solve this problem by finding the maximum kernel correlation configuration of the two point sets. KC was evaluated and found to easily enable easy convergence of a shape in noisy or noiseless environments compared to standard ICP. This was because of the smoothness provided by weighting the contributions of multiple links between a point and its neighbours.

Jian and Vemuri (2011) proposed an approach to registering point sets with huge amounts of noise and outliers (non-homogeneous points) using the Gaussian mixture (probabilistic way of grouping data points based on their similarities). The approach represents both point sets as a gaussian mixture and the registration was aimed at aligning the mixtures. The approach works by measuring the similarity of two point sets by considering their continuous approximations. Since this approach is based on Gaussians, the computational complexity for rigid transformations is usually O (mn) with n Gaussians and m control points. For non-rigid transformations, this is largely dependent on the number of control points. A dense spacing of control points allows flexible non-rigid deformations to be registered than the sparse spacing of control points.

Shape registration algorithms that represent the shapes as point clouds can be classified into spectral, Iterative, and probabilistic methods (Assalih, 2013). Assalih (2013) developed an approach to matching 2-D sonar images using their 3-D representation called Sorting the Correspondence Space (SCS). The approach involved sorting the correspondence search space and attempting to solve the registration problem for registering SONAR images (using sound waves to measure the distance of objects). The research designed and evaluated a real-time 3-D reconstruction system that used multiple sonar views to get different perspectives of the object. The approach was able to register rigid and non-rigid shapes with large amounts of outliers. The algorithm was also found to be intelligible, robust against outliers, and fast towards issues with estimating 3-D motion from 2-D sonar images such as limited number of detectable features to match, representation in the 3-D motion sequence, and the low signal-to-noise ratio (SNR) of 2-D sonar imagery. The computational complexity of the SCS algorithm was found to be $O\ (N \log\ (N))$.

Myronenko and Song (2010) developed a registration method based on a probabilistic approach (Assalih, 2013) called the Coherent Point Drift (CPD). It considers the matching as a probability estimation problem. One shape represents the Gaussian Mixture Model (GMM) (a complex probability distribution that is a combination or mixture of 3-D Gaussian distributions centred on each point) and another shape represents the data point cloud. The first point cloud is iteratively transformed to maximise the probability that the other point cloud was randomly generated from the GMM.

The GMM is iteratively fit to the data model by maximising the likelihood and this allows the probability of a point being part of any Gaussian to be determined. Every data point in Gaussian has a degree of uncertainty, which is the probability that a point could be generated by the Gaussian cluster. Each Gaussian in the mixture of Gaussians is made up of;

1. The centre or mean of the Gaussian
2. A covariance matrix that represents the width or size of the Gaussian
3. Weight of the Gaussian function

In Point Cloud Registration, the correspondence between point clouds is not explicitly known. This means that the closeness can be found for one point based on the correspondence of a point to another through a probability, with spatially closer points, having a higher correspondence probability than farther points.

Assume two-point clouds, X, with points x1, x2, x3, coloured in red and Y, with points y1, y2, y3, coloured in green. Figure 2.1 shows the known correspondence between the points where we indicate this with dashed lines, however, the algorithm has no prior knowledge of this correspondence and only assumes that spatially closer points are corresponding points.

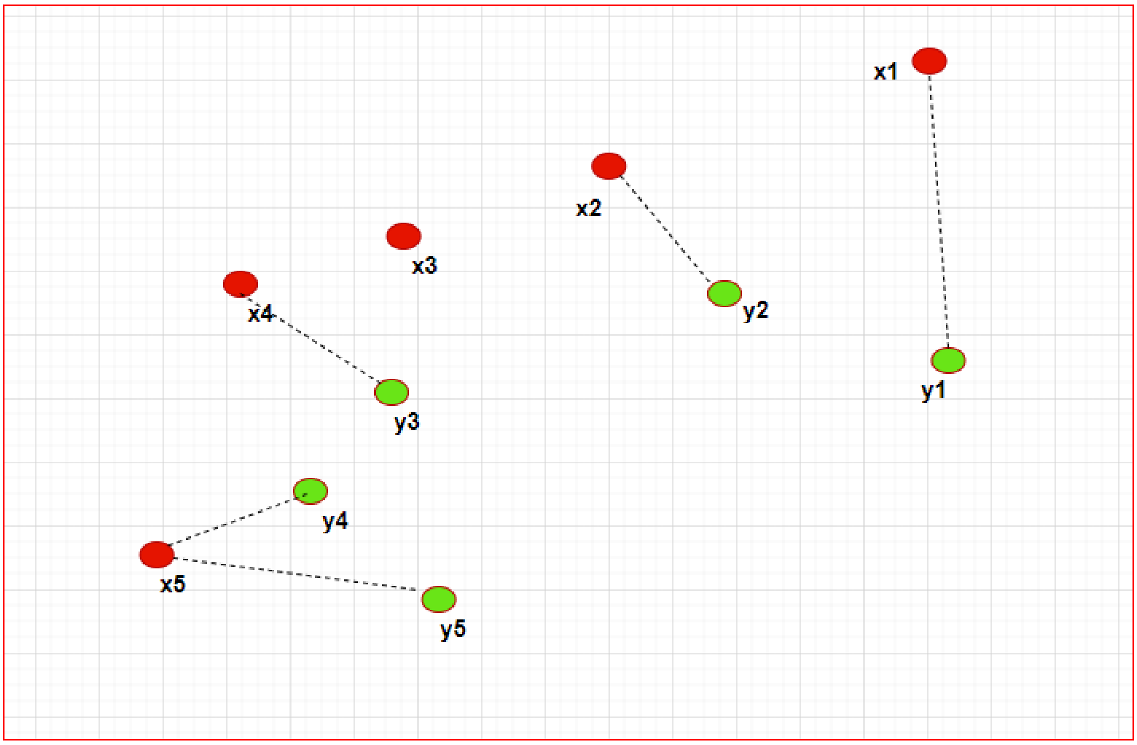
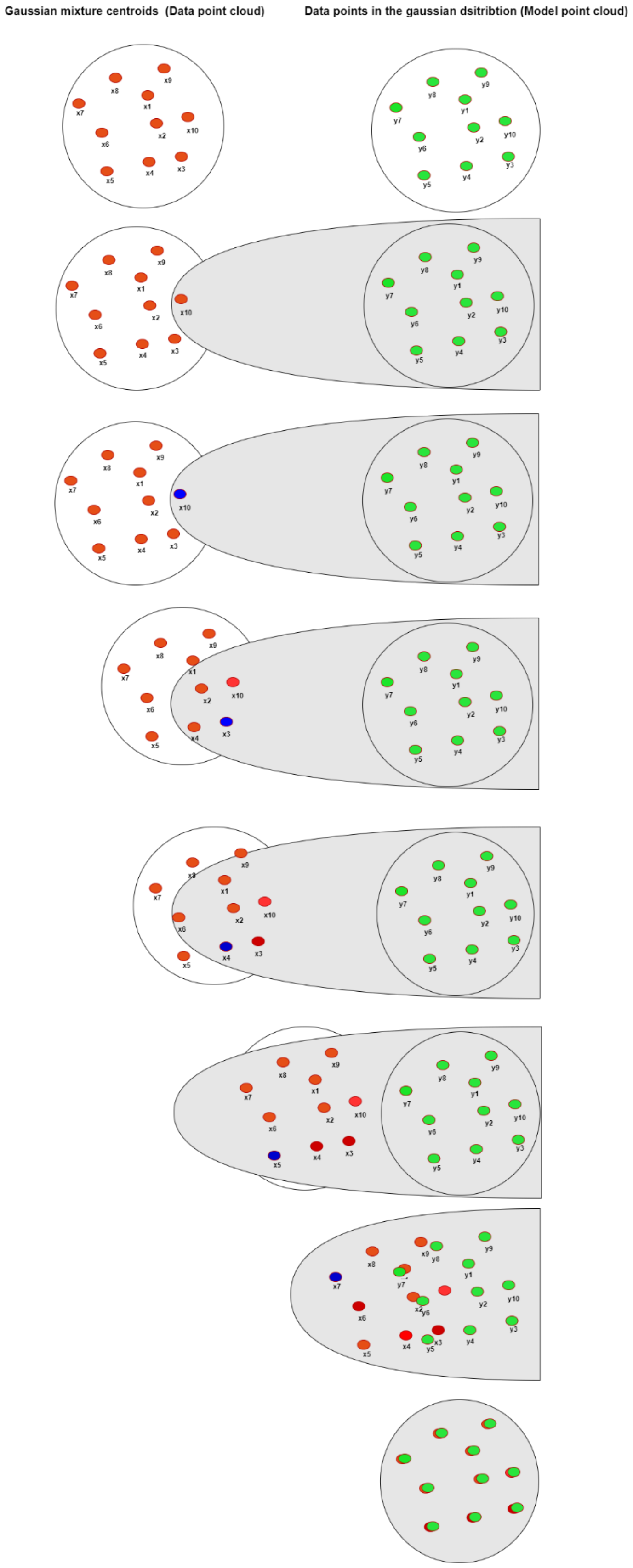


Figure 2.1 A point cloud (x1, x2, x3, x4, and x5 points) and corresponding point cloud with points (y1, y2, y3, y4, and y5 points) showing their point pairs with dotted lines depicting correspondence based on proximity of points. Points y4 and y5 find the same correspondence, x5 and point y3 map to x4 due to the shortest distance between the points and not necessarily the true matches.

Figure 2.1 shows two point clouds, X and Y with points in red and green, showing correspondence between points. Since correspondence is not explicitly known but rather implied in the algorithm based on the spatial distances or proximity between the points, an approximate match can be assigned to a point based on their proximity. This means that there is the possibility of wrong matches (x4 and y3) and multiple points with the same match (y4 and y5)



Gaussian mixture centroids (Data point cloud)
Data points in the gaussian dsitribtion (Model point cloud)
x1
x2
x3
x4
x5
x6
x7
x8
x9
x10
y1
y2
y3
y4
y5
y6
y7
y8
y9
y10

Figure 2.2 The red point cloud represents the data point cloud and also the centroids. The green point cloud represents the model point cloud as well as the data points in the gaussian mixture model. The blue points represent a particular gaussian centroid in focus and all the green points (data points in that gaussian). An Expectation Maximisation process calculates the mean vectors and covariance of each distribution (Gaussian) to update the Gaussian mixture (data point cloud). By

iteratively updating the position of the centroids based on the mean vector and covariance of the data points (data point cloud), the centroid positons in the data point cloud (LHS) move towards the model point cloud (RHS) by applying the computed transformation to all the points in the data point cloud (centroids).

From Figure 2.2, the Gaussian mixture model's centroids are represented as points in the red point cloud (the data point cloud) and the data points in the Gaussian distribution are the points in the model point cloud (green point cloud). Using an expectation maximisation technique to compute the mean vector and covariance matrix of each gaussian and re-updating the centroid locations results in the centroids (data points) to get closer to the model point cloud (Gaussian distribution data points). This process attempts to find the likelihood that the data points (model point cloud) were generated from the data point cloud. CPD uses this process to end up registering the two shapes.

Using the approach of correspondence probability, the coherent point drift treats one point cloud say X, as the GMM and the other point cloud, Y as the point cloud obtained from the GMM. The coherent point drift algorithm uses expectation maximization (EM) to solve the problem of computing correspondences between the moving or changing GMM point cloud (moving because of recalculation of centroids locations) and the fixed-point cloud observed from the GMM. The Expectation maximization enhances the probability that the fixed-point cloud Y was obtained from the GMM by enhancing the probability of points and their correspondence.

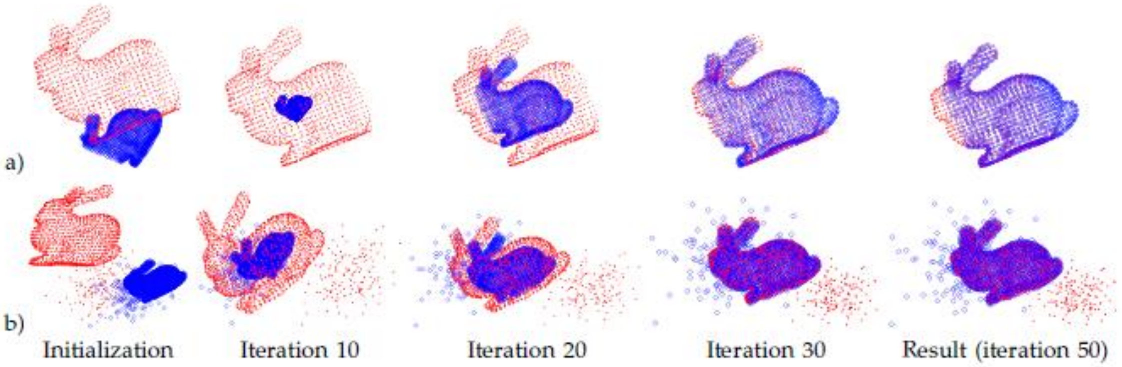


Figure 2.3. Image registration with some level of noise using the Gaussian Mixture Model. a) Image series shows sub-sampled data by cutting off parts of the red and blue bunnies. b) Image series shows the addition of random unwanted points. shapes are registered correctly in the presence of the noise and missing data. Myronenko and Song (2010).

The image registration using the Gaussian Mixture Model in Figure 2.3 Is an extract from Myronenko and Song, (2010). It shows the data shape in the red point cloud with noise and missing points in all instances and the Gaussian mixture model in blue point clouds. Figure

2.3a shows points that were sub-sampled from the bunny data set. This also has the front and back parts of the bunny deleted. Figure 2.3b adds random unwanted points to the point cloud, and this shows as a residue in the point matching as those points were rejected.

The matching attempts to iteratively align the Gaussian Mixture Model to the data shape by transforming the Gaussian Mixture Model in a way that preserves its structure. This is done by moving all the centroids of the Gaussian Mixture Model using the same computed transformation. This process of preserving the structure is based on motion coherence (Yuille and Grzywacz, 1988) that infers that closely related points move in coherence. Using motion coherence implies that rigid and non-rigid transformations should be implemented differently. For rigid shapes, CPD re-calculates the Gaussian Mixture Model centroid locations with a rigid transform to arrive at a closed-form solution (well known or fixed solutions) and for non-rigid shapes, the algorithm regularizes (smoothens the rate of change by adding a displacement function to the positions to ensure uniform movement or change in points according to the motion coherent theory) the displacement field (distance of the centroid movement).

CPD suffers issues such as no proof that the algorithm always converges, certain aspects of the algorithm such as motion coherence (points close to one another tend to move coherently) is not very clear, and the algorithm is sensitive to the rotation of the target shape (Hirose, 2020). To counter these issues, Hirose (2020) extended the Coherent Point Drift method to be based on Bayes (Joyce, 2003) (Bayesian) theorem (the probability of an event occurring based on prior knowledge of conditions associated with that event and not simply a consideration of the population as a whole). The motion coherence aspect of the CPD is replaced with Bayesian inference meaning that it is based on a prior distribution of displacement vectors. The Bayesian CPD enhances the CPD with a guarantee of convergence, ability to perform rigid and non-rigid matching in a single algorithm and flexibility for extension with other concepts such as non-gaussian functions. The CPD achieves a computational complexity of $O\ (MN)$ for rigid point cloud registration and $O\ (M^3)$ for non-rigid point cloud registration where M is the number of points in the gaussian centroids and N is the number of points in the second point cloud.

Many learning based methods have been proposed for the task of shape matching. Litany et al., (2017) proposed an approach called SURFMNet that involves supervised learning of an input set of descriptors to derive a combination of weights based on a type of flexible representation of mapping between shapes (Ovsjanikov et al., 2012). SURFMNet extends FMNet (Litany et al., 2017) using an unsupervised learning model that works best for non-linear transformations of descriptors in order to obtain high-quality mapping between the shapes. The approach does not require knowledge of the shapes and it uses the learning-based approach to transform the given input descriptors whilst improving the properties for mapping the structure of the shapes.

Groueix et al., (2018) proposed a deep-learning approach to shape matching which involves parameterising the surface of the shape into a template and then using a learnt global feature vector to map the transformation into the input surface. The proposed approach, called Shape Deformation Networks, uses a shape deformation network that learns to map a deformed shape to an input shape. The approach works by predicting the transformation that aligns the derived template to the shape. The method was found to be robust.

Another learning-based approach to the shape matching task is the learnable elementary structures (Deprelle et al., 2019). This approach works on the principle of training a network to learn the shapes of elementrary 3D structures which can be used to predict whole or parts of a shape. The approach uses two techniques for learning elementary structures – patch deformation and point translation. Patch deformation starts with a full object and deforms it into a learned structure whilst point translation learning starts from a set of points and an optimisation technique attempts to reconstructer the points into the target object. The limitation of this approach is that it does not generate a continuous surface, but rather a set of points. However, the research identified this approach to be more flexible as the topology of the objects can be changed. The approach was found to improve upon 3D shape generation and matching.

One popular point matching algorithm based on an Iterative approach (Assalih, 2013) is the Iterative Closest Point algorithm (ICP) (Besl and McKay, 1992). It is a well-known shape matching algorithm for registering 3-D shapes including free-forms and curved surfaces. It is based on an iterative process of minimising the distance between two shapes. It is also independent of shape representation as it represents all shapes as point clouds. The algorithm is designed to not require any pre-processing nor make use of features of the points in the cloud except their spatial positions. ICP is fast and modular, allowing aspects of the algorithm to be replaced or improved. For instance, the correspondence search space can be implemented in a way to allow the use of metadata of the points in the matching. Despite these, ICP suffers some limitations such as slowness to converge when registering many thousands of points. This is due to the computational cost of the point-to-point distance calculation that finds the closest point in the model point cloud for every point in the data point cloud. ICP is noted to perform poorly in the presence of outliers.

An ICP-based approach to shape registration that was found to be robust against outliers was the fractional ICP(FICP) (Phillips, Liu and Tomasi, 2007). FICP provides an optimal registration by robustly rejecting likely outliers. Fractional ICP uses the fractional root mean square distance metric to align points that are likely to be inliers than outliers. In an experiment comparing fractional ICP, ICP and TrimmedICP (Chetverikov et al., 2002), using random initial rotation angles 5, 10, 25, and 50 degrees on a random axis, fractional ICP was found to be six(6) to eleven (11) times faster in certain scenarios. Other techniques for outlier detection include rejecting correspondences that have distances above a threshold, points with dissimilar normals or points on the boundary of the geometry, and modelling outliers using sparsity-inducing penalties (Rusinkiewicz and Levoy, 2001). An extensive review of registration methods is outlined in (Brown, 1992, Fonseca and Manjunath, 1996, Zitová and Flusser, 2003, Díez et al., 2015).

For this thesis and the rest of this chapter, the research focuses on the Iterative closest point algorithm by Besl and Mckay (1992). This algorithm is widely known for its simplicity and effectiveness (Mora et al., 2016). It is also the most popular method for fine-grained registration (matching shapes by refining an iterative process from an initial or rough estimated transformation).

2.4 3-D Shape Descriptors and Signatures

Generally, 3-D shape descriptors and signatures are representations used to capture information about the shape of an object to help with registration. They capture the geometrical structure of a 3-D object (Gal, Shamir and Cohen-Or, 2007) and can classify it as a member of a particular category using informative descriptors about the shape (Fang et al., 2015). One of the ways of estimating a good initial transformation for the ICP algorithm is the use of 3-D feature descriptors to capture some knowledge about the shape. Salti, Tombari and Di Stefano, (2014) explain that these approaches can be grouped into Signatures and Histograms. Signatures describe the 3-D surface of the shape considering the neighbourhood of a reference point by defining an invariant local reference frame, whilst Histogram-based methods describe a given point by their structural features that are projected into 2-D histograms. Signatures describe the shape much better than histograms because of their use of spatial representations.

Different techniques have been used to provide a good initial guess or estimate. Attia and Slama, (2017) proposed a *“points projection on the plane”* (PPP) approach. This 3-step approach involves projecting a 3-D shape onto a 2-D plane defined by the Principal Component Analysis (PCA) (Wold, Esbensen and Geladi, 1987). PCA is a technique used to extract major patterns in a matrix defined by, for instance, a group of eigenvectors. PPP allows features or components of the point cloud such as points dispersion to be maintained in the 2-D projection. The second step of PPP computes the projected point cloud by computing the projection of each point using the Euclidean distance between each point and the plane. Finally, the projected point cloud is then used to estimate the transformation and then the point clouds are aligned using ICP. The computational complexity of this approach is $O\ \big(nlog(n)\big)$. This method was found to be robust against large initial transformations in the ICP algorithm.

Rusu et al., (2008) proposed a histogram and signature-based point feature representation that is based on the local neighbourhood information. The approach works by computing the local point feature histograms for all points in the point cloud by encoding the local neighbourhood's geometrical properties using the mean of the curvature at that point. The neighbouring points were identified based on their k closest neighbours. The geometrical properties of the surface result in a unique signature in the feature histograms space. The information was then trained using a machine learning algorithm to derive the classifier. The classifier was used to enable points to be identified based on the surface feature space they belong to so that points belonging to or having similar surface properties will be classed under the same feature space. The research found that the density of points on the surface did not have a major impact on the feature histograms.

A 3-D local feature descriptor known as the Density Curvature Angle (DCA) (He et al., 2021) was proposed to minimise the need for a good initial transformation in 3-D point cloud registration. DCA works by extracting the density of the point cloud using the average distances between k neighbouring points, the curvature of the point cloud, and the normals of the point cloud into a three-feature 3D DCA. DCA was found to be faster when compared with Fast Point Feature Histograms (FPFH)(Rusu et al., 2008), which makes use of the 3-D coordinates of the point and the surface normals to form the feature descriptor, Signature of Histogram of Orientation (SHOT) (Tombari, Salti and Stefano, 2010) which uses both signatures and histograms to create a 3-D descriptor, and Rotational Projection Statistics (RoPS) (Guo et al., 2013) based on rotational projection statistics of the surface, However, DCA was found to take more time to complete when the specified k neighbours are large.

Gal, Shamir and Cohen-Or, (2007) proposed a signature-based pose-oblivious (not sensitive to changes in pose of the objects) shape signature that was invariant to uniform scaling and rotations of the points. The approach works by creating a similarity of the probability density function of values of a point on the boundary of the shape which is aimed at finding the local diameter on a sample point on the boundary. This is achieved by random sampling of some point measures such as the distance of a point to the centroid of the shape, which is then used to build the histogram of the values using 64 bins (histogram bars of equal width with height corresponding to measures of data points). The approach was found to be fast for 3-D model comparisons, however, performance was reduced when the models include internal parts instead of just the mesh because the calculation of the diameter function was inaccurate in such instances.

In recent times, deep learning-based methods have been proposed to improve the robustness, efficiency, and performance of shape registration (Huang et al., 2021). The methods have been based on feature learning methods that combine learning-based registration with optimisation-based methods and end-to-end learning-based methods that aim to transform the registration problem into a regression problem. Recent end-to-end based methods are effective in typical registration cases, however, they have also been found to be limited in their accuracy and applicability. (Choy, Dong and Koltun, 2020).

A fast feature-metric point cloud registration framework (Huang, Mei and Zhang, 2020) was proposed that registers shapes based on extracted features. The approach works by iteratively minimising the error between the projected features of the two shapes. It avoids the use of geometry points of the shapes as such there is no correspondence point search involved. The approach consists of an encoder that extracts features from the data and model point cloud. A decoder function is also used to decode features of the point clouds and to obtain the feature-metric projection error which is used to train a neural network in an unsupervised way (The machine learning algorithm discovers patterns in the unlabelled data for its internal classification). Finally, a semi-supervised neural network registers the images without finding some correspondence for each point.

A learning-based local volumetric patch descriptor called 3DMatch (Zeng et al., 2017) was proposed for establishing correspondence between partial 3-D data. The approach works by using a self-supervised (using no labelled data in the training) 3-D convolutional neural network (A type of neural network for learning images that can take an input image and assign importance based on the trained dataset) to learn the mapping from volumetric 3-D patch to a 512-dimensional feature descriptor for the region of the shape. The method was evaluated using the descriptor and RANSAC (Fischler and Bolles, 1981) and results showed improved performance for matching local geometry in new scenes.

The scope of this research involves matching shapes using information about the point such as the coordinates, so it doesn’t develop 3-D descriptors nor use them directly in the developed ICP variants.

Shape Matching Method	Summary	Comparison
Point correspondence	Point matching algorithms that represent the shape as a point cloud and find correspondence between points	Mapping of points, transformations, details of known information used in the matching, efficiency and robustness
Machine learning-based	Uses supervised or unsupervised learning-based methods and a combination of other techniques such as feature extractors	Details and complexity of the learned features.
3-D Shape Descriptors	Various methods for extracting shape representation for use in shape matching to aid with a good initial transformation	Efficiency and robustness of the 3-D shape descriptor.

Table 2.1 Summary classification of the methods reviewed and details on comparison.

2.5 3-D Protein Structure Matching

Schmitt, Kuhn and Klebe, (2002) explained that recent methods of detecting the structural and functional relationships of proteins consider the overall 3-D structure. This is because older methods that just consider the sequence information (similarity-based on positions of consecutive chains of atoms) are lacking because highly similar proteins based on a structure usually have a high similarity based on sequence, however highly similar proteins based on sequence information do not necessarily have high structural similarity. This is because changes such as mutations distort more of the sequence information than the structural information and this has also resulted in recent similarity methods shifting towards comparisons based on the spatial coordinates.

Schmitt, Kuhn and Klebe, (2002) further explained that finding the similarity between 3-D proteins requires a translation and rotation matrix to be found that superimposes one 3-D protein onto another. It was noted that protein matching can be computed purely using the spatial positions of the atoms, however, it is more challenging because of the huge number of atoms available in many proteins. There are also challenges with spatial representation and transformation of the 3-D structures of proteins such as speed of processing, and accuracy of the resulting matching. More reliable solutions in terms of their speed, efficiency and accuracy can be found using methods that associate coordinates with some predefined properties. For instance, the implementation of TaggedICP (Ankomah et al., 2020) uses a similar concept of tag information (a combination of atom type and information of 3 neighbouring atoms of a point) as a property to aid in the registration of 3-D proteins. Schmitt, Kuhn and Klebe, (2002) further developed an algorithm to score 3-D protein shapes based on descriptors developed from the 3-D structure of pockets of protein binding sites (the part of the protein structure where a drug or virus will bind or be attached to). This study noted how the protein binding site's similarity score can help in the design of drugs especially preventing side effects of drugs which are usually caused by the drug binding to a similar protein but not its target.

Similarly, Angaran et al.,(2009) developed a web-based application, called Molecular Local (MolLoc) for recognising similar regions on a molecular surface. The regions for comparison were cavities, binding sites, and functionally relevant residues. The application allows a user to import two PDB files or IDs of two PDBs to be compared. The comparison used two approaches (geometry with atom type, and only geometry). Whilst the geometry and atom type approach iteratively match the atom types in both molecules by considering pairs of atoms with the same atom type, specifically for matching proteins, the geometry only approach matched structures based only on the spatial coordinates and did not consider atom types. Whilst MolLoc compares the surface information of the proteins by allowing the user to select chains in the structure to compare, Schmitt, Kuhn and Klebe, (2002) algorithm use a predefined similarity score based on a descriptor to find the similarity between the structures. MolLoc also allows a better comparison of chains in two proteins whilst Schmitt, Kuhn and Klebe, (2002) algorithm provides a global score that is used to compare parts of proteins. Both algorithms recognise the importance of a geometric comparison of protein structures that considers the spatial positions of the atoms.

Another approach to comparing the regions of protein structural surfaces without using the sequence nor 3-D structural information as part of the comparison was the use of hashed triangles of physicochemical properties (Shulman-Peleg, Nussinov and Wolfson, 2004) (e.g. triangles formed from 3 atom types in the 3-D structure). The hash encodes certain properties of the triangle such as the corresponding atom types and the distances between the atoms. A hash table containing a hash key for each molecule was then created that allowed all triangles for a molecule to be retrieved using the hash key. The triangle information was used to superimpose the triangle on a similar site on the matching protein through an iterative process of calculating all possible transformations to align the triangle nodes with triangle nodes of the matching protein.

Ellingson and Zhang (2012) proposed an Iterative closest point-based algorithm called Triangulation-based Iterative closest point (TIPSA) for protein surface alignment. The research focused TIPSA on the binding sites of proteins (parts of the protein where other proteins in a molecule such as medicines can attach themselves to) and used a combination of nearest neighbour classification of the atoms, information from the 3-D structure of the molecule, and the radius of gyration of the atoms (Lobanov, Bogatyreva and Galzitskaya, 2008) to help predict which molecule can bind to a particular surface. TIPSA avoided the need for initial alignments by finding correspondence between atoms based on their atom types. A set of atoms with their correspondence were then matched to align their surfaces. This atom type-based approach is implemented in TaggedICP and SubsetICP+.

Kirys et al., (2007) proposed an approach to 3-D protein shape matching that first finds correspondence between groups of atoms (subsets) and then match the correspondence based on Fractional ICP (Phillips, Liu and Tomasi, 2007).

In Biomedicine, ICP has been used in several tasks such as registering skulls in medical image analysis (Sinko et al., 2018). Sinko et al. (2018) used ICP, and 3-D SIFT in a process involving the conversion of MRI scans of skulls to 3-D point clouds and an ICP registration. The MRI scans were imported, segmented into features of interest, and then converted to point clouds. The differences in the visualised point clouds were then studied for possible deformities. The research found that using ICP and 3-D SIFT was fast and robust and that new challenging scenarios can be addressed using computer vision technologies, inspired by the latest computer animation technologies. Such technologies include virtual reality.

2.6 Iterative Closest Point

Three of the earliest most popular registration methods were proposed by Besl and McKay (1992), Chen and Medioni (1992), and Zhang (1992). These researches modelled a complete 3-D object acquired through a scanner from different angles and minimised the distance between the shapes using an iterative process of computing transformations. The approach of Chen and Medioni (1992) minimized the distance of the two points along their normal vectors, while Besl and McKay (1992) minimized the point to point distance and has a closed form solution. This approach usually requires some approximation that must be used to get a least squares solution. Zhang (1992) approach minimised the distance between frames in free form.

Using only the geometric information of the points, the ICP algorithm repeatedly computes the distances between each point in the data point cloud to each point in the model point cloud to establish the correspondence between the points. A set of transformations are then computed to best align the two point clouds. This iterative process continues until the specified number of iterations is exhausted or the change in mean squared error over two consecutive iterations is below a set threshold (Donoso, Austin and McAree, 2017, Du et al., 2017).

Chen and Medioni (1992) acquired several range images (3-D images acquired through special sensors that use pixels to indicate depth) from an object in an attempt to register and align these views into one complete object model. The research outlined that a good initial transformation is important as the algorithm works to perform finer matching as compared to coarse matching. Also, because convergence is not always guaranteed, a good starting transformation can ensure convergence.

The ICP algorithm (Besl and McKay, 1992) is designed to always converge at a given level of complexity (reasonable initial transformation and noise). It is designed not to require any additional features of the points to aid in the matching process (Besl and McKay, 1992). For instance, it does not require metadata such as colour or label information for matching similar points. However, other registration methods based on ICP, that make use of feature points have been proposed, including using light and depth information (Weik, 1997), pair-wise registration constraints (Pulli, 1999), surface features (Lee et al., 2011), and feature points (Kang and Park, 2015).

Besl and McKay (1992) used a point to point distance method to find the correspondence between points. This computationally expensive process involves computing the closest point (Euclidean distance) in the model point cloud for each point in the data shape. Assuming shapes P, and Q are to be matched using the point-to-point representation,

Where.

$\boldsymbol{t}$ = Translation vector derived from the computed transformation
R = Rotation matrix derived from the transformation
q_i = Point in Q shape that is closest to p_i given the current transformation specified by R & t
p_i = Point in the transformed P (Rp+$\boldsymbol{t}$) shape

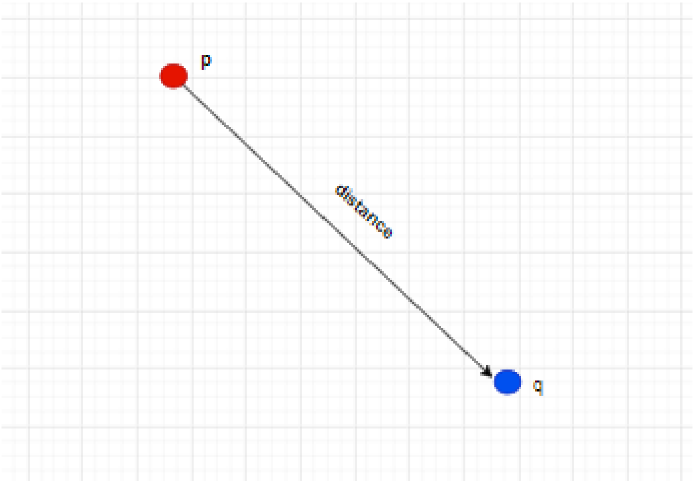


Figure 2.4 Point-to-point distance metric method for finding the distance between two points p and q. The method represents the distance along a straight line joining the two points, transformed point p, (R_p+1) and q.

The point-to-point method in Figure 2.4 shows the distance from point p to point q in the direction of the arrow. The arrow represents where we would like p to end up after applying the next transformation, (R, ***t***). When we have many points, we have many different arrows from which we can estimate that next transformation in the least squares sense. The error for one point-to-point correspondence search after an iteration, ε_{point} is given by.

$$\varepsilon_{point} = \sum_{i} \|Rp_i + t - q_i\|^2$$

This objective function can be minimized for the re-estimation/update of (R, ***t***) with a closed form solution using the quaternion method (Besl and McKay (1992)).

On the other hand, Chen and Medioni (1992) used the point-to-plane method to compute the distance between corresponding points by computing the distance between each point and the plane defined by the corresponding point and its normal vector. The shortest distance from a point to a plane is along a line perpendicular to the plane. Therefore, the shortest distance from point p to the plane is along a line parallel to the normal vector, which is shown as a red arrow line segment. The point to plane method is illustrated below.

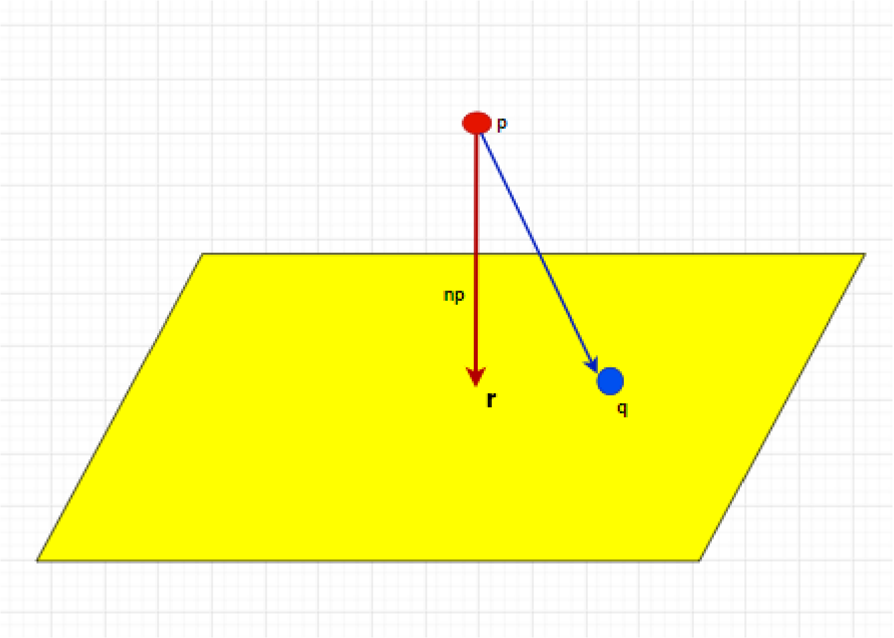


Figure 2.5 Point-to-Plane distance metric method aims at finding the distance from point p to q, in a perpendicular direction that ends up on the plane (yellow surface). np is the direction of p's normal vector. The distance in red is also the projection of the vector from p to q. Assume r is the point where the red line touches the plane, then the shortest distance from p to the plane is the distance between r and p,

In the point-to-plane method, depicted in Figure 2.5, point p (red point) ends up on the plane (yellow surface) but not necessarily at q (blue point). np shows the direction of q's normal vector (perpendicular to the plane). The closest point on the plane to point ***q*** is the distance between the point where the original point normal vector touches the plane (r) and the point on the plane (q). The error is given as.

$$\varepsilon_{plane} = \sum_{i} \left[(Rp_i + t - q_i).n_{q,i} \right]^2$$

The objective function is minimised by selecting control points and computing their surface normals. An iterative process is then repeated that transforms each control point and their normal, finds the intersection between the model surface and the normal line of the transformed control point, computes the tangent plane of the model surface at the transformed control point and then computes the transformation that minimises the error. The objective function is minimized for the re-estimation/update of (R, ***t***) using the least squares method if the transformation change from one iteration to another is relatively small and thus the rotation matrix can be linearized about such changes.

A new symmetrised objective function that achieves the simplicity and computational efficiency of the point to plane method was proposed by Rusinkiewicz (2019). It considers a point and both normals by adding the normals together to create a symmetry, instead of considering one normal in the point to plane method.

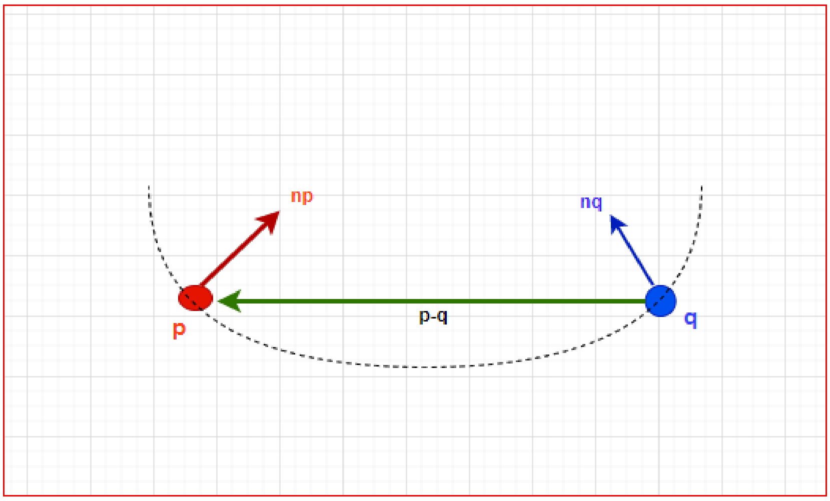


Figure 2.6 Symmetric error metric shows nearby points p and q along a circular arc (dotted lines). The vector between the points p, and q is perpendicular to the sum of their normals (n_p+ n_q). Red arrow represents the direction of point p's normal vector. The Blue arrow represents the direction of point q's normal vector.

In Figure 2.6, the symmetric error metric (Rusinkiewicz, 2019) considers the normals of points p and q. It explains that for any 2 sampled points p and q on a circular arc represented by the dotted lines, the vector between them (***p-q***) is perpendicular to the sum (n_p+n_q) of their normal vectors.

$$\varepsilon_{symm} = \sum_i \left[(Rp_i \ - \ R^{-1}q_i + t).\left(n_{p,i} + n\ _{q,i}\right)\right]^2$$

The Iterative closest point algorithm has been applied to several fields. In molecular biology, ICP has been used to compare the 3-D structures of molecules to study the functions and interactions (Angaran et al., 2009) by representing the molecules as 3-D point clouds (Wallace, Borkakoti and Thornton, 1997, Schmitt, Kuhn and Klebe, 2002, Shulman-Peleg, Nussinov and Wolfson, 2004, Weskamp et al., 2004, Shulman-Peleg et al., 2008), with each point representing an atom.

2.6.1 Algorithm Structure

According to Besl and McKay (1992), assume a data shape P with a number N_p of points and a model shape X with a number N_x of points. Then the ICP algorithm can be described as below.

1. Initialise Iteration to 0, maximum number k_{max} of iterations, set change in MSE threshold, transformation (R, ***t***)
2. Apply the transformation (R, ***t***) to the data point cloud
3. Compute the closest points in the model cloud for each point in the data cloud (Computational cost: $O\left(N_x N_p\right)$)
4. Compute/update the registration transformation (Computational cost: O (N_p))
5. Apply the registration to the data point cloud (Computational cost: O (N_p))
6. Terminate the iteration when the iteration count or MSE threshold is reached.

```
$\mathrm{P}o \longleftarrow P$
$for\ iteration\ k\ \coloneqq\ 0\ to\ k_{max}\ do$
    $closest\ points\ Y_k\ \leftarrow\ CLOSEST\ POINT\ SEARCH\ (P_k, X)$
    $transformation\ M_k, MSE\ d_k\ \leftarrow\ REGISTRATION\ (\ P_o, Y_k)$
    $if$
        $change\ in\ MSE\ \ d_{k-1} - d_k < threshold$
    $then$
        terminate
$return\ \ P_{k+1}, M_k, d_k$
```

Algorithm 1. Iterative Closest Point Algorithm (Besl and McKay, 1992).

Assume the data point set P, with N_p points and the model shape X, with N_x points. The closest point set from P to X is computed as $Y_k = C(\ P_k,\ X)$. The mean square error (MSE) of this correspondence, e_k is computed as below.

$$\frac{1}{N_p}\sum_{i=1}^{N_p}\|y_{ik} - \ p_{ik}\|^2$$

2.6.2 Strengths

Besl and McKay (1992) identified advantages of the ICP matching algorithm to include being independent of shape representation where shapes can be represented in any form such as point sets, line segment sets, curves, and surfaces. Pre-processing of the 3-D data is not required, and the algorithm is designed to only use the objects global features (the entire surface geometry) to perform the match.

ICP is popular for 3-D rigid registration because of its simplicity and effectiveness (Mora et al., 2016). The algorithm can also handle reasonable amounts of evenly distributed noise and its simplicity and extensibility allow it to be used together with other algorithms.

2.6.3 Limitations

This section helps to formulate the research problem. It highlights the specific limitations of the ICP algorithm from Section (2.3) that were of interest to the research and were improved. It also introduces the developed algorithms and their techniques. It helps define the scope of the research and gives the reader a clear understanding of what the research focuses on. This also helps set the stage for the remaining chapters in achieving their objectives.

ICP suffers from some limitations. Besl and McKay (1992) outlined some limitations of the algorithm such as the high-performance cost of the closest point operation (This can be improved with K-d trees or parallel processing). Chen and Belaton (2014) also found out that unwanted points and outliers (points that have significantly worse closest matching points than the others) can affect the convergence to a less accurate solution because ICP considers all points in the correspondence search. ICP also suffers from slow convergence and partial overlaps (Zhang, Yao and Deng, 2021).

It has also been noted that ICP performs well in some contexts and datasets (such as structured and feature-rich datasets) and worse in others (natural, unstructured and information-deprived environments e.g. terrain data) (Donoso, Austin and McAree, 2017). ICP also performs much better when all points in the data shape have true corresponding points in the model shape (full shape matching) than partial overlapping point clouds.

Ning et al. (2018) argued that the three (3) types of outliers in point clouds are sparse outliers (e.g., erroneous measurement points with low local point density in scanned data), isolated outliers (e.g., high point density points in a scanned data) and non-isolated outliers (e.g., outliers attached to the surface of a scanned object).

2.6.3.1 Requirement for a Good Initial Transformation

ICP requires a good initial transformation to converge at a small mean square error (Besl and McKay, 1992). The algorithm may not even converge in certain situations such as large rotations (Besl and McKay, 1992). Larger transformations have been found to decrease the quality and efficiency of the match (Attia and Slama, 2017).

2.6.3.2 High Cost of the Correspondence Search

The correspondence search can be a slow process. With the increasing number of points, the performance of the algorithm can degrade since it involves many such distance calculations. With this process occurring at every iteration, the performance can be greatly affected. In instances of high-density data, it might provide unacceptable performance results (Mora et al., 2016). Besl and McKay (1992) also noted how the performance can be enhanced by using K-d trees (Bentley, 1975, 1990) in the correspondence search method.

Because the search for correspondence does a naive shortest geometric distance calculation (distance of cartesian point coordinates), the matching may not necessarily be meaningful especially point clouds having unwanted points, even though this is how the ICP algorithm is designed. This is because the algorithm computes the closest point in the model point cloud for every point in the data point cloud, meaning that this process ignores true matching points and the error increases with large noise and unwanted points.

The ICP algorithm is computationally expensive, and this cost increases with the increasing number of points in the point clouds. The algorithm computes the spatial distance between each point in the data shape to each point in the model shape in a way to establish the correspondence for each point. The cost of this brute force computation is $O(N_x N_p)$ for a number N_x of points in point cloud X and the number N_p of points in the point cloud P. Besl and McKay (1992), however, suggested the use of binary search data structures such as kd-tree to reduce the computational complexity O (log N_x). Parallel processing can also be used to improve the speed of the correspondence search, otherwise, a shape with millions of points becomes slow to complete this distance calculation.

The processing power (number of Central Processing Units (CPU) cores) available in today's CPUs coupled with Graphical Processing Units (GPUs) also provides an advantage in some regard in terms of parallel processing. Parallel processing can enhance the speed of the distance calculation since the task of computing the distance between two points will not be done sequentially, one point at a time, but rather as many simultaneous calculations in the form of tasks will be processed by the available CPU cores.

2.6.3.3 Convergence at Larger Transformations

The ICP algorithm converges to a less optimal solution when the transformation (rotation and translation) of the data shape is considerably large. It may not even converge with large rotations (Besl and McKay, 1992). This implies that the ICP algorithm requires a good initial transformation to converge at an acceptable solution.

The techniques implemented in the developed ICP variants considered the effects of large transformation angles on the convergence. Techniques using metadata of the points such as colours and labels aid in the matching and the implementation of different partition sizes were expected to help the algorithm converge at an expected minimum even with a large rotation angle.

2.6.3.4 Visualising the Shape Matches in 2-D and Virtual Reality

Visualising the shape match allows the user to not only employ an analysis graph to understand the details and process of matching but also to be able to quickly view and understand shape matching overlaps and quality as well as quickly and interactively adjust initial transformations as required.

Matching shapes in virtual reality extends the usability of the PC based visualisation system with a 3-D user interface providing an immersive user experience. The hand controllers enable the user to use both hands to mimic natural gestures when interacting with 3-D shapes. This allows the transformation to be repeated by changing the initial transformation each time.

The immersion provided by the virtual reality system allows the user to discover more detailed information from the animation of the shape matching process. The user can see and analyse how far the data shape moves away from the model object in an iteration to visually study the error at that iteration.

2.6.4 ICP Variants

The modular nature of the ICP algorithm enables some steps to be improved whilst others are unaffected. This has easily led to the rise in different variants being developed over the years. Many of these variants have focused on improving some steps of the algorithm such as the closest point search or developing specific features for related applications such as improving matching using metadata knowledge of the shapes (Mora et al., 2016).

Rusinkiewicz and Levoy (2001) classified variants of the ICP algorithm according to the stages of the algorithm; Selection, Matching, Weighting, Rejecting, Assigning an error metric, and Minimizing the error metric, and based on the point pairs. This research particularly evaluated how the selection of point pairs affects the convergence speed of the matching. This was done by comparing a baseline ICP variant, Multiview registration method for matching shapes (Puli, 1999) with other methods that handle the selection of points in varying ways (Besl and McKay, 1992, Turk and Levoy, 1994, Masuda, Sakaue and Yokoya, 1996, Weik, 1997).

The other stages of the ICP algorithm and related variants include,

1. Point matching by finding the closest point (Besl and McKay, 1992) which was accelerated using k-d tree, caching, and fast surface point computation (Simon, 1996), some compatibility metric such as metadata colour or tag information(Godin, Rioux and Baribeau, 1994, Ankomah et al., 2020)

2. Weighting based on normals (Chen, Hung, and Cheng, 1999), Gaussian mixture model (Myronenko and Song, 2010)
3. Rejecting pairs (Turk and Levoy, 1994)
4. Assigning an error metric by squared distance from a point to point (Besl and McKay, 1992), point-to-plane (Chen and Medioni, 1992), symmetric error (Rusinkiewicz, 2019)
5. Minimising the error metric.

The Multiview registration method (Pulli, 1999) enhances Chen and Medioni (1992) approach to registering multiple views of an object. Chen and Medioni (1992) proposed registering multiple views of an object by first registering two views into a single base view and then incrementally adding new views to the base view. Pulli (1999) identified issues with Chen and Medioni (1992) approach such as adding new views might end up at the outer or inner layer instead of the middle layer of the view thereby not distributing the errors evenly across. Chen and Medioni (1999) also indicated a limitation of the algorithm not being able to handle complex datasets. Improvements made in the Multiview registration method enable registering datasets that are too large to fit into memory at the same time. The algorithm also obtains a uniform registration for all views to spread the registration errors on all views instead of accumulating them. Also, the pairwise registration is performed once and that is used as a constraint to align the views in the iterative process. The Multiview registration method by Pulli (1999) was found to be robust for scanned data with many surfaces and has been extensively used in production environments (Levoy et al., 2000). The method integrates these features.

1. It uses random sampling for selecting point pairs for matching.
2. Points are matched based on their closeness to the other view that has a normal within 45 degrees of the source normal.
3. The weighting of points is done uniformly
4. The algorithm rejects point pairs containing edge vertices (points on the boundary of one point cloud, found by extending a points normal vector until it intersects with the other point cloud) and the percentage of point pairs with the largest point-to-point distances
5. Uses point to plane error metric
6. Uses the “select-match-minimize” iteration

Comparing the selection of points of the ICP algorithm (Besl and McKay, 1992), Besl and McKay (1992) differ with Pulli (1999) by always using all available points in the dataset for the correspondence search. Points are matched based on the shortest Euclidean distance to a point in the other dataset. Besl and McKay (1992) however suggested that this distance calculation can be improved using K-d trees (Bentley, 1990) as well as closest point caching (Simon, 1996).

Turk and Levoy (1994) employed uniform sub-sampling in their point selection for matching. This method discards pairs of points that are too far apart and eliminates pairs with either point on the mesh boundary to create a sample for selection and matching. This does not require using all the points in the cloud as such the reference point cloud is not required to be a subset of the model data point cloud. This characteristic of the point cloud is normal of range images since the views represent different sides or angles of the object to be registered.

Masuda et al. (1996) proposed a new method for the registration and integration of range images to model the object as a geometric surface. Their point selection method differs in a way that involves random sampling with a different set of points at each iteration. The method works by iterating the randomly sampled points, estimating the motion parameters, and determining the best estimation.

Weik (1997) proposed a registration method for aligning partial surfaces that uses the light intensity (and its variations) and depth information (from the camera) as additional constraints in the form of metadata, for finding the correspondence between point pairs for their improved reliability.

Rusinkiewicz and Levoy (2001) also introduced the normal-space-sampling strategy. This strategy is based on the selected aspects of the reviewed methods (Besl and McKay, 1992, Turk and Levoy, 1994, Masuda, Sakaue and Yokoya, 1996, Weik, 1997). It ensures that the distribution of normals amongst selected points is as large as possible. This method works by grouping points according to the position of the normals in angular space to ensure enough constraints (surface features) are considered in determining the transformation. Uniform sample points are then taken across these groups. This method is noted to have lower computational cost and lower robustness compared to other feature-based methods. Even though both Pulli (1999) and Rusinkiewicz and Levoy (2001) make use of some sort of sampling strategy, the latter differs in that it attempts to group the points and select uniform samples from all groups.

Rusinkiewicz and Levoy’s (2001) classification was augmented with another stage of the ICP algorithm, neighbourhood selection (Donoso, Austin and McAree, 2017). Donoso et al. (2017) evaluated 20,736 ICP variants against their metrics; accuracy, precision, and computational cost. The 20,736 variants were computed from a combination and permutation of 25 methods used in (Donoso, Austin and McAree, 2017) to analyse matching of scanned mining terrain data with 3 new methods analysed by Rusinkiewicz and Levoy, (2001) for matching the same dataset. The research found out that none of the algorithms was simultaneously accurate, precise, and fast to compute. Donoso et al. (2017) had an extensive detail of several ICP methods and their implementation strategy.

Mora et al.(2016) evaluated Chebyshev distance (The greatest of the differences between two points along any coordinate dimension) and Manhattan distance (The distance between two points measured along axes at right angles.) as an alternative to the Euclidean distance. Mora et al.(2016) proposed two distance metrics that reduce the computational cost and increases the average speed by fourteen (14)% using the Manhattan distance metric without negatively impacting the match quality and convergence of the algorithm.

Sparse ICP (Bouaziz, Tagliasacchi and Pauly, 2013) is another variant of the ICP algorithm that uses a technique called sparsity inducing norms to reduce significant amounts of noise and outliers. The technique automatically learns the separation between data and outliers. The approach involves formulating the local alignment problem as maximising the number of zero distances between correspondences. The approach also meant that the least probability norm ℓ_p is used instead of a least squares approach, ℓ_2. This provides a choice between 0 and 1 as a trade-off between efficiency and robustness. SparseICP was extended for facial recognition by using resampling and denoising techniques that improved the registration accuracy (Yu, Da and Guo, 2019).

2.6.5 Use of Additional Metadata

Additional metadata of points in a point cloud can be used in the correspondence search to enhance the algorithm by making the closest point meaningful. For instance, colours, point normals and labels can be introduced in the search for correspondence to make the matching of points more meaningful. Features such as the global orientation of a point in a point cloud can change when transformations are applied to the point cloud and as such, there is the need to factor such features in calculating the mean square error (Schutz, Jost and Hugli, 1998). Invariant features such as local colours and labels of the point will not change with rigid shapes. These invariant features can be used to improve the registration (Godin, Rioux and Baribeau, 1994, Combes and Prima, 2009) by reducing the search space for correspondence (Sharp, Lee and Wehe, 2002).

Schutz et al. (1998) proposed a multi-feature improvement to the traditional ICP algorithm making use of surface colour, geometry, and local shape orientation. Results showed that this improved the range of successful convergence as the multi-feature improvement helped with the matching of corresponding points.

2.6.6 ICP Based on Neural Networks

Advances in machine learning and artificial intelligence systems have seen the application of neural networks to the shape matching problem.

Qi et al. (2017) proposed a hierarchical neural network called PointNet++ as an extension and improvement upon PointNet (Qi, Su, et al., 2017). PointNet++ explores the representation of point clouds as a neural network. The researchers trained a 3-D geometric neural network to perform tasks such as shape matching, shape segmentation, and semantic scene labelling (predicting or naming the shape from a given point cloud) task. The approach taps into the benefits of PointNet such as its ability to abstract sets of local features into higher-level descriptors used to iteratively register a shape through a process of grouping sets of points into overlapping regions, extracting the local features such as geometric and neighbourhood information, grouping these local features into larger units for the processing which results in higher-level features use that describes the whole point cloud. PointNet++ was evaluated on ModelNet CAD model dataset (Wu et al., 2015) and results showed a lower error rate (0.51%) compared to PointNet (0.78%) (Qi, Su, et al., 2017), Multi-layer perception (1.60%) (Simard, Steinkraus and Platt, 2003), Gradient-based learning (Lecun et al., 1998) and comparable with Network in Network (Lin, Chen and Yan, 2014) which uses micro neural networks to extract features (0.47).

Wang and Solomon (2019) used a deep learning approach to develop a learning-based method (Deep Closest Point) for registering two rigid point clouds. The method consists of three (3) parts, a part to map the input cloud to invariant embeddings (A mapping of points in the point cloud to vectors in the neural network) which helps with the matching of pairs of point clouds, and an attention-based (ability to focus on a subset of the inputs) module for predicting soft matching (coarse or rough matching) of pairs of points, and a singular value decomposition part to predict the transformation. The deep closest point was found to outperform ICP (Besl and McKay, 1992) and some ICP variants (Yang et al., 2016). The learned features of Deep Closest Point were found to improve rigid alignment algorithms.

Aoki et al. (2019) developed PointNetLK which is based on PointNet (Qi, Su, et al., 2017). The research explored the idea of interpreting point clouds as an imaging function (producing a fixed dimensional output regardless of samples or ordering of points) to apply image alignment approaches to the point cloud strategy. Their implementation also improved on PointNet (Qi, Su, et al., 2017) to accommodate gradient (dependencies such as similarities and changes between nearby points) estimates through convolution. The convolution involves detecting feature changes between neighbour points and computing a weighted average to represent spatially close points.

Similarly, Qi et al. (2017) improved on PointNet (Qi, Su, et al., 2017) to recognise fine-grained patterns by exploring a hierarchical neural network (PointNet++) that applies PointNet recursively on a nested partition. The partitioning of the points is done by the FPS-Farthest Point algorithm and each partition includes a centroid and a scale. The results showed significant improvements in the accuracy of the match when compared to DeepGM (Luciano and Ben Hamza, 2018).

Yew and Lee (2020) proposed a deep learning-based ICP variant (RPM-Net) whose optimal performance is not based on a good initialization. The research designed a deep learning-based feature extraction network that uses several layers, with different layers extracting different features. This network is then used to compute hybrid features (different types of features such as local geometric information) of each point from its geometric (shape and size) and spatial (location coordinates of the points in Euclidean space) properties. The fusion of these properties was found to improve correspondence for RPM-Net.

2.7 Problem Formulation

The problem is formulated based on the discussed limitations of the ICP algorithm. The limitations of the ICP have shown opportunities to improve the convergence speed, match quality, and present intuitive visualisation.

Three (3) variants of the ICP algorithm were developed to register 3-D protein structures, TaggedICP (Ankomah et al., 2020), KmeansICP, and SubsetICP+. These algorithms have been developed for matching the 3-D structure of protein molecule datasets from the protein data bank (RCSB Protein Data, 2020). TaggedICP makes use of known information of the points (tag information) for matching the point clouds and it partitions the correspondence search space using the atom type. KmeansICP makes use of the popular K-means algorithm for partitioning the search space for correspondence. SubsetICP+ implements a modified version of SubsetICP (Chen, Belaton and Pan, 2013) that creates subsets based on atom type information and rejects some transformations based on their effect on the data point cloud.

The variants discussed in this chapter are all based on the standard ICP algorithm (Besl and McKay, 1992). These algorithms focus on the selection and matching stages of the standard ICP algorithm because of the high computational complexity of that method. Enhancements made to the correspondence search to reduce the computational cost and improve the matching quality goes a long way to reduce the overall computational cost of the algorithm.

The ICP algorithm (Besl and McKay, 1992) is a well-known shape matching algorithm. The limitations of the algorithm have fuelled a desire for researchers to improve it along all of its aspects; Selection, Matching, Weighting, Rejecting, Minimizing the error metric, and Assigning an error metric based on the point pairs (Rusinkiewicz and Levoy, 2001), however, the strengths of the algorithm such as its simplicity (Mora et al., 2016) and extensibility (as a result of its modular design) have equally ensured the ability for researchers to easily implement alternative enhancements for its improved performance. For instance, different distance metrics (point to point, point to plane, and symmetric) can be used in the objective function optimisation. The search for correspondence can also be improved using k-d trees, parallel processing of the search sequence, partitioning of the search space, or caching of points.

The 3-D protein structures are represented as point clouds with atoms representing the points and they are used to match full shapes and partial shapes in the form of mutations of the same molecules. Since the shapes are rigid objects, a rigid transformation is computed that iteratively aligns the shapes using the ICP method.

The improvements in the designed variants can be used to match different 3-D protein structures to compare the speed of the registration, the quality of the match, and to determine the similarity between protein structures. Particularly for TaggedICP, the matching on tag information presents meaningful corresponding point matches that helps the match quality by matching true correspondence because of the constraint presented using the metadata tag information.

2.8 Summary

This chapter has introduced shape matching and its related algorithms. Early uses of shape matching have been discussed. It has provided a critical discussion on the similarities, differences, and potential areas for improvement for some existing shape matching algorithms.

The Standard ICP algorithm has been detailed along with many of its variants. Implementation of several variants of ICP has seen ICP being used to register shapes using different techniques over diverse datasets such as facial recognition (Yu, Da and Guo, 2019), and more recently, techniques based on neural networks (Wang and Solomon, 2019). The chapter has expanded the researcher and the reader's knowledge on many of the techniques of ICP. This has also helped provide a foundation for understanding the developed variants as part of this research.

This chapter has also discussed the main limitations of ICP-based algorithms. The high cost of the correspondence search method, the convergence at larger angles and, shape visualisation and interaction. This has set the stage for the research's focus. These limitations were particularly considered in the design and development of the algorithms by informing which computationally expensive parts of the algorithm needs to be improved. The strategy was to ensure the new ICP variants focused on aspects of the ICP algorithm that benefits optimisation by making use of different approaches such as using metadata for matching (Ankomah et al., 2020), using k-means (Krishna and Narasimha Murty, 1999) for partitioning, and using subsets (Chen, Belaton and Pan, 2013).

The literature review presented an understanding of the high computational cost of the different parts of the ICP algorithm. The knowledge gained helped to reduce the correspondence search space by partitioning the molecules in 3 different ways: by position using a k-means cluster for the KmeansICP algorithm, by atom type for SubsetICP+ and by atom type and neighbourhood information (TaggedICP).

The literature review also helped identify the limitations of visualising 3-D point clouds in virtual reality especially with optimising the number of points rendered per frame. The virtual reality application has a simple interface that allows the user to see important transformation details of the data shape and the model shape. The application allows the user to interact with the shapes and set an initial transformation. The user is also able to visually analyse the overlaps of the shape matches as well as visualise the animation of the shape matching process.

This chapter has achieved objective one (1), *"Gain a critical understanding of similarities, differences, and potential areas for improvement in existing shape matching techniques by performing a thorough literature review".* The thorough literature review and critical discussion presented in the chapter has expanded the reader's knowledge in understanding the similarities, differences, and existing challenges in shape matching algorithms.

3 CHAPTER 3 VIRTUAL REALITY

3.1 Introduction

The research was interested in a way that allows a user to intuitively interact with the shape matching process. This means pausing the visualisation, adjusting the positions of a shape, and continuing with the process, allowing the user to be immered in the shape matching animation and investigate the shape matching overlap using advantages provided by the affordances of virtual reality.

This chapter introduces the reader to virtual reality (VR). It discusses the history of VR and provides a critical discussion on the affordances of virtual reality such as immersion, presence, and navigation.The knowledge gained from this chapter was necessary for the development of the Iterative Closest Point Virtual Reality (ICPVR) application for matching shapes in virtual reality and to achieve objective three (3), "*Gain a critical understanding of the history, current state, and challenges of virtual reality through a literature review*". For instance, the review found that the use of teleportation can increase the overall experience of the user by providing the ability to move in the virtual world within a limited physical space.

3.2 History of Virtual Reality

Virtual reality can be described as a man-made environment presented to the user that makes the user experience a sense of being in the environment. This feeling of realism or being present in a virtual environment is provided by a combination of the 3-D content presented through stereoscopic display devices such as head-mounted displays (HMD) and intuitive navigation such as using hand controllers to interact with the presented content.

VR aims to allow a user to perform an action believing that the action is performed in the real world and this is achieved by mimicking real-world scenarios through the provided virtual environment (Arnaldi, Guitton and Moreau, 2018). Vivid reactions (interaction and feedback from presented content) can create a sense of realism and presence making the user feel that they are inside the presented environment (Shu et al., 2019).

Arnaldi, Guitton and Moreau (2018) explained that the objective of VR is to allow a user to execute tasks virtually as if being executed in the real world and this is achieved by the VR technology mimicking a realistic environment by providing virtually, the same information in the real world. For instance, providing a virtual walk path to the user in VR will allow the user to walk on the path virtually because the brain assimilates the virtual information provided as real.

Virtual reality was noted to have begun by Morton Heilig's multi-sensory simulator that supported VR functionalities such as 3-D display, binaural sound, and interaction. The very first idea of an implementation of VR that included all aspects (components) that we currently associate with VR, except interactivity was credited to Ivan Sutherland in 1965 (Sutherland, 1965, Watson and Luebke, 2005, Kour, 2015). This concept was developed further, gaining a lot of media attention (Marcus and van Dam, 1991) and then this propelled later research and development until presently.

The 1960s through to the 1990s saw this gradual virtual reality development gaining momentum and going through notable developmental phases (Mazuryk and Gervautz, 1996, HTC, 2020, Microsoft, 2020, Oculus, 2020) such as;

The term virtual reality was coined by Jaron Lanier in 1989 (Slater and Sanchez-Vives, 2016). Lanier's description of the possibilities of the virtual world compared to the physical world in its ability to bring imagination to view, creates excitement and potential which is being explored today. Some studies even note the first appearance of the term in the 1960s (Freina and Ott, 2015) which was the era where VR had a lot of media attention (van Dam, Laidlaw and Simpson, 2002). However, the first recognised mechanical device to engage the senses was the Sensorama (Figure 9.1) which did not support interaction (Mazuryk and Gervautz, 1996).

in recent times, virtual reality has been used in new areas of mainstream society which has further pushed its adoption and commercialisation. Increasingly diverse industries are adopting virtual reality and showing its advantages. In the entertainment industry, specifically, live concerts, the Epic Game's Fortnite environment (Marlatt, 2020) hosted a virtual live concert by musician Travis Scott that saw more than twelve (12) million virtual audiences (BBC News, 2020). Although these many virtual audiences are partly a result of the global COVID19 pandemic and restrictions in place in many countries (Buckley, 2020). Similar adoption in other industries proves this trend has come to stay. Price cuts on VR headsets, although not in all categories has helped its adoption, for instance, Facebook cut the price of the Oculus Rift twice in 2017 bringing the price down to $399 from $799 before launching other newer versions of the product.

Other applications of virtual reality include its use in physical education and sports training (Li and Li, 2020) for simulating real life scenarios in improving the user's skills and physical qualities and, improving physical therapy through virtual therapy sessions (Camporesi, Kallmann and Han, 2013).

The proliferation of virtual reality devices and their applications have advanced research into the components or metrics of virtual reality that gives the user the perceived immersion, interaction and navigation, task performance, feeling of presence, knowledge improvements, and usability of the virtual environment (Ankomah and Vangorp, 2018).

3.3 Related Works

Kim et al., (2017) developed a 3-D virtual reality interface for automating the task of matching 3-D and 2-D shapes. The research was aimed at developing a generalised user interface, where any shape matching algorithm can make use of the UI for matching 3-D shapes, however, its design was made specifically for matching biological 3-D shapes such as bone models as compared to ICPVR's UI for selecting different algorithms. As a result, the user interface could only allow users to match 3-D models of human shoulder bone structures by manually rotating and moving the structures. Feedback from participants indicated that the virtual reality interface was more productive than using the 2-D desktop.

Bruder, Steinicke and Nüchter (2014) explored further attempts at 3-D user interfaces. The research developed a point cloud virtual environment by reconstructing a 3-D point cloud of buildings from 2-D range images captured in real-time by a moving robot. The research used an Intelligent Robot for Mapping Applications in 3-D (Irma3D) (Nüchter, Elseberg and Borrmann, 2013) which is a battery-powered lightweight robot capable of taking 2-D and 3-D images of an environment. Irma3D (Nüchter, Elseberg and Borrmann, 2013) also can be controlled remotely to direct its path around the buildings. It can also move autonomously and explore within a set limited area as well as being controlled wirelessly by a user.

As the robot passed around buildings, pairs of 2-D range images captured in real-time from the mobile robot are transmitted over a wireless network connection to the point cloud virtual environment (PCVE). These images were registered with the ICP algorithm implementation found in the 3-D toolkit (3DTK) (The 3-D Toolkit, 2020) and visualised in virtual reality using an Oculus Rift head-mounted display (Oculus, 2020).

Bruder, Steinicke and Nüchter (2014) found several challenges in the point cloud virtual environment. These included the poor performance of rendering millions of 3-D points in virtual reality because the number of points rendered caused the frame rate to drop. The research proposed locking the frame rate to the display's refresh rate and limiting the number of points displayed.

Point clouds have been explored for novel areas such as simulating an unmanned aerial vehicle (UAV) using data from a Ladar (Sensors using light waves to generate topographical data) sensor for validating the selection of a target (Bergé et al., 2016).

The research also found that viewing objects represented as scattered points pose a challenge when users get close to the object as the points tend to appear too far apart to give the impression of a solid object. This is because of the point sampling density of the virtual objects. For instance, a molecule point cloud with atoms representing points as visible atoms composed of a sphere primitive in virtual reality will allow the user to walk through the point cloud and as such the points dissolve into the users' view. However, ICPVR does not suffer from this problem due to the atoms’ textures.

The research also suggested a good interaction design that considers the concentration of points at an area the user selects. This is because, unlike actual 3-D objects where a user can hold a solid but virtual part of the 3-D object, point clouds present gaps or spaces between parts of the 3-D object and an interaction design that considers these gaps might benefit 3-D User interfaces and user interaction with point clouds in virtual reality. The approach of ICPVR allowed the user to interact with the 3-D protein molecules and virtually be able to walk through the molecules because the overlaps of the atoms are points of interest for the user as it provides some visual analysis.

Tredinnick, Broecker and Ponto (2016) pointed out some challenges of point cloud visualisation in virtual reality to include slow rendering performance and lags in interactivity when visualising massive point clouds. The research developed an algorithm to render points in a point cloud over many frames to ensure lag-free interactivity when visualising massive point clouds in VR. The research used a technique that reduces the number of points rendered by a frame by constantly re-projecting a previous frame with the rendered point clouds whilst incorporating some new points. This method ensures that points are rendered seamlessly over many frames allowing large amounts of point cloud data to be rendered with no adverse effects on the user experience such as slow performance or lags in interactivity. However, there is a possibility that, if the viewpoint or the orientation of the point cloud changes significantly from one frame to the next, parts of the point cloud that were previously occluded need to become visible, but the re-projected previous frame will have gaps in those parts causing visual artefacts. This research shows the benefits of rendering large point clouds across multiple frames to reduce the trade-off between interactivity and quality of the virtual experience.

Alexiou, Yang and Ebrahimi (2020) investigated point cloud visualisation in virtual reality. Their research focused on the subjective analysis of the quality of the contents of point clouds such as colour, as well as the user’s experience of interacting with the point clouds in virtual reality.

The research aimed to compare the subjective quality of the environment and 3-D models between passive and interactive evaluation experiments. A Unity3D (Unity Technologies, 2019) based application was developed called PointXR (Alexiou, Yang and Ebrahimi, 2020) that enabled subjective analysis of the quality of the user experience including colour quality and performance. PointXR works by loading and converting the point clouds to 3-D models using the Pcx Point Cloud Importer (Takahashi, 2020). Participants were presented with two versions of the 3-D models for the experiment, original quality and distorted version and results showed that participants preferred close-range and frontal view examination.

ProteinVR (Cassidy et al., 2020) is a web-based molecular visualisation program for visualising 3-D proteins. ProteinVR enables users to interactively visualise 3-D protein structures in virtual reality headsets, mobile devices, and desktop computers. It supports several modes that allow visualisation and interactivity on supported screens and interaction mechanisms. In desktop mode, ProteinVR allows the use of a mouse and keyboard for interaction with the 3-D protein visualisation. In VR mode, ProteinVR can visualise the 3-D structure of proteins in virtual reality using mobile or desktop computers connected with supported VR headsets such as HTC VIVE (HTC, 2020), Oculus Rift (Oculus, 2020), and Google Cardboard (Google, 2016). VR mode also enables a user to navigate around the scene by moving in their physical space or using teleportation to adjust their view of the virtual 3-D protein structure.

Doutreligne et al., (2014) developed a Unity3D based molecular visualisation software called UnityMol. UnityMol enables interactive visualisation of molecular structures in VR, allowing users to investigate and explore a structure represented as hyperballs and helix (Figure 3.3 and Figure 3.4). The application allows the visualisation of molecular events such as molecular bond formation and interactions. This visualisation allows the user to understand the behaviour of molecules and to gather insights leading to a better understanding of the relationships between molecules and how molecular properties influence these relations.

The knowledge gained from the discussed related works informed how ICPVR was designed. It was understood that providing the ability for the user to move freely in the virtual environment especially in a limited physical space required the teleportation feature. The usability of ICPVR was also considered in ensuring the users could interact with the protein molecules. Because there is no limit on the size of the protein molecule (in terms of the number of atoms), large molecules could cause lags during the shape match animation. To address this challenge, the closest point search method, which is a time-consuming method of the ICP algorithm was enhanced inside VR by refreshing the VR display after every set number of correspondence searches which attempts to spread the rendering of the points across multiple frames.

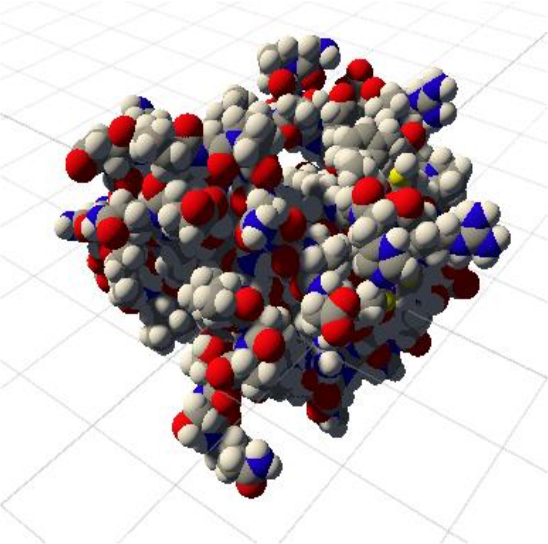


Figure 3.1 3-D representation of a molecule, Unity3D. The image shows the original size with no scaling applied. The 3-D protein was imported from a PDB file. The colours represent different atom types (Red=Oxygen, blue=Nitrogen, yellow=Sulphur, grey=Carbon).

Figure 3.1 shows a 3-D structure of a molecule, represented using the space-filled model (Corey and Pauling, 1953) which consists of using spheres to represent the 3-D structure of the molecule. The space-filled model shows molecules and their associated atoms using the same scale. Atoms are usually represented with different colours, and this allows easy identification of the atoms in the 3-D environment. The bonds are not visible in Figure 3.1 nor used in the research because the research was not concerned with the bonds and none of the algorithms developed specifically utilise the bond structure.

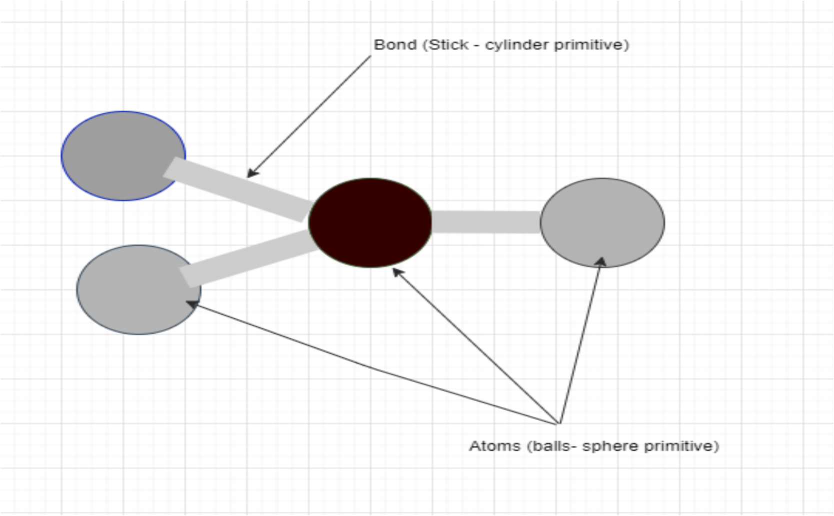

Bond (Stick - cylinder primitive)
Atoms (balls- sphere primitive)

Figure 3.2 Illustration of Ball and Stick model showing atoms and their bonds in a molecule. The atoms are held by bonds to form a sequence. Molecules are made up of many small sequences of atoms and bonds.

The ball-and-stick model (Figure 3.2) can be used to represent the 3-D structure of the molecule, showing atoms represented as balls (sphere primitive), the spatial positions of the atoms as well as the bonds between the atoms represented as sticks (cylinder primitive).

Even though this model hides the bonds between atoms of the molecules as shown in Figure 3.1, the research is not concerned with that because each atom is treated as a point in the point cloud and as such, the bond is not particularly important for this research. Also, the scale of the atoms was uniformly reduced to easily see the overlap of two molecules during a shape match in virtual reality especially when they have been super-imposed. Based on the above, the research represented all 3-D structures with the ball and stick model without the connecting sticks. Molecules and their associated atoms can also be represented using other structures such as the van der Waals (Bondi, 1964), and helix models (Boyle, 2018) shown in Figure 3.3.

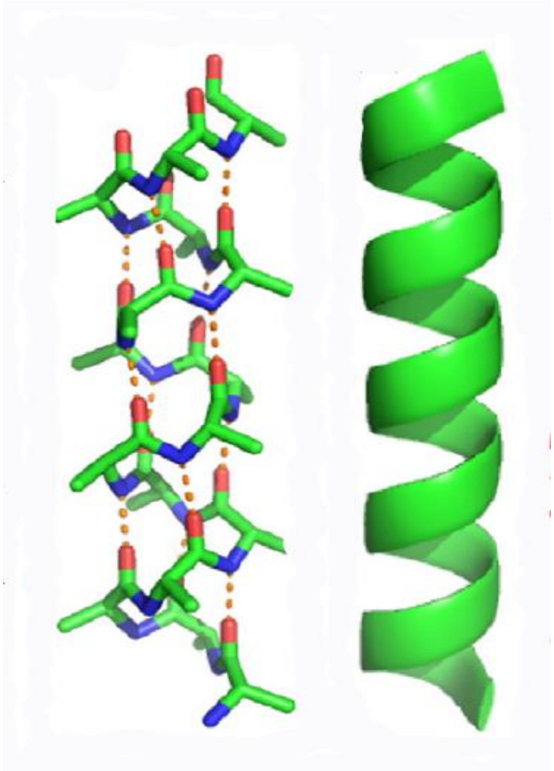


Figure 3.3 Sample helix structure representation of a molecule (LHS) Molecular view of helix structure (RHS) Carton view of a helix structure (Boyle, 2018)

Technological improvements in graphic processing units (GPUs) have provided the necessary advancements in processing capabilities for complex 3-D tasks that otherwise would have suffered poor performance from running on a traditional CPU. GPUs have thousands of processing cores that allows these complex tasks such as molecular visualisations to be efficiently run (Chavent, Vanel, et al., 2011).

The opportunities provided by the advancements in GPU technology have also enabled research and experimentations with complex forms of molecular structural representation such as protein docking (interaction of protein molecules with other molecules), molecular structure evolution (structural changes on a molecule over time due to changes such as mutations) and real-time molecular interaction (modelling the visual changes in the 3-D structures of protein interactions). These new capabilities and visual representations have enhanced user impressions and their subjective analysis of atoms and molecules (Chavent, Lévy, et al., 2011).

Chavent, Vanel, et al. (2011) employed the technical advancements (processing power from thousands of cores of GPUs) in GPUs to research different representations for molecules. The research proposed and demonstrated a molecule successfully rendered with hyper balls (Figure 3.2) and licorice structures (Chavent, Vanel, et al., 2011) in Figure 3.3.

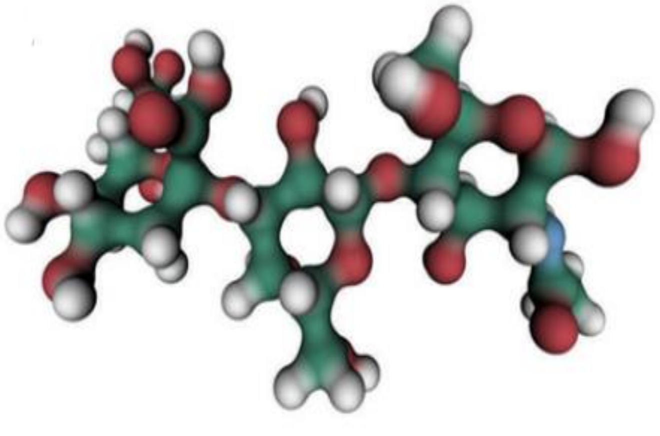


Figure 3.4 Hyperballs representation of a molecule (Chavent, Lévy, et al., 2011) was derived from the ball-and-stick model modified to show atoms bonds as elliptic hyperboloids connecting atoms.

Hyper balls (Chavent, Vanel, et al., 2011) enabled the use of a combination of balls, sticks and hyperboloids (Figure 3.4), which are cone-shaped objects with one or two surfaces (Weisstein, 2020). The use of a mixture of these shapes allows extending the molecular visualisation to visualise complex phenomena such as the rate of changes of molecules due to mutation or protein docking activities over time (Chavent, Vanel, et al., 2011).

3.4 Virtual Reality Metrics

Virtual reality systems can be evaluated and appreciated from different aspects in different metrics as follows: immersion, presence, interaction, knowledge improvement, navigation, performance, and usability. Presence and immersion are the core affordances of VR because they are responsible for blurring the lines between the virtual and the physical world. Navigation and interaction are the actions that can be performed in the virtual world, knowledge improvement is an outcome of a user experiencing the virtual world and performance and usability are evaluation metrics.

The metrics are reviewed using research that evaluated them. Detailed discussions are provided on the aim of the evaluation, the specific VR platforms and tools used, and the results and findings of the research.

3.4.1 Immersion

Lee et al. (2017) explained 'transparent immediacy' as a property of virtual reality which allows users to forget the existence of media and believe that they are immersed in the virtual world. The research further noted how metrics such as interaction can affect the overall measure of immersion, for instance, the poor interaction in a game can affect the overall immersion the user experiences. The research also hypothesised on participating users preferring to use the data gloves as a hand gesture interface more than to use the available HTC VIVE controller in the game experiment. This can be explained by the fact that hand gestures afforded by the hand gloves feel more natural than using controllers.

The term 'immersion' refers to the measurable outputs of VR technology. The more stimuli the system presents to the user's senses to mimic the real world, the more the system is immersive (Slater, 2003). This definition agrees with Cummings and Bailenson (2016) view of immersion being the quality of the virtual environment in terms of its technical provisions such as higher field of regard and supported natural gesture types. This view also supports states of immersion such as full or partial (Kjeldskov, 2001, Marks, Estevez and Connor, 2016) and this view pointed later research to suggest the need for further experimentation on aspects of immersion such as perception and interaction for acceptance (Laha and Bowman, 2012). This is because several other aspects of VR such as interaction, navigation, and usability are the focus of immersion (da Costa and Nedel, 2017). Increasing any of these aspects in terms of making them feel like the real world as much as possible also increases the overall immersion experienced by a user in the VR as such VR experiments prioritise making VR selection, manipulation, and navigation around objects as close to the real-world as possible. Whilst Slater (2003) argued of the separation between immersion and presence (Slater et al., 1996, Spagnolli and Gamberini, 2002) this is rather commonly seen to be loosely used interchangeably in other research (McMahan, 2003).

Research attempting to understand the process of immersion found that as a user interacts with a virtual environment, their perception that the entire environment is within their grip causes the user to enter a flow state (Choi, Kim and Kim, 2007, Zhou and Deng, 2009, Animesh et al., 2011, Cheng, Chieng and Chieng, 2014). Jennett et al. (2008) also described it as extreme immersion, however, some differences exist in the degree of realism. Brown and Cairns (2004) describing the flow state put it as a realisation "*When you stop thinking about the fact that you're playing a computer game and you're just in a computer*".

Stated preference (SP) experiments (experiments that involve using individual respondents' statements about their preference) which in some cases, lacked a sense of realism (because users would be choosing preferences without having experienced their choices) were studied to gain a sense of realism using virtual reality (Farooq, Cherchi and Sobhani, 2018). In a comparative analysis of the choice scenario using virtual reality, Farooq and Cherchi (2018) evaluated the introduction of realism, immersion, and interactivity into the stated preference experiments using HMDs (Head Mounted Displays) and found out that the scenario was more real to the users because of the immersion and interactivity afforded by the HMD (Farooq, Cherchi and Sobhani, 2018). In their experiment, the preference for autonomous vehicle scenarios was increased from forty (40%) to seventy (70%) in their VIRE (Virtual Immersive Reality Environment) setup. Positive results from SP experiments include tenant-mix for shopping centres (Borgers et al., 2010).

Similarly, natural body motions were also employed in the study of overall immersion in medical imaging visualising DT-MRI (Diffusion Tensor Magnetic Resonance Imaging – the process of acquiring information about organic tissues based on their types and the diffusion of water molecules along with them). The research showed that immersion and interactivity realised from using an HMD aided the users understanding of complex geometric models depicting neural structures like neurons (Zhang et al., 2001)

Laha and Bowman (2012) discussed the importance of VR research measuring immersion to compare individual components of immersion rather than evaluating the whole user interface display and this problem may have hindered certain tasks such as analysing volume data visualisation. Their research proposed the need for controlled experiments to address this problem, reiterating that traditional VR experiments lack generality because of confounds involving multiple components of immersion and as such empirical results are needed to back these claims. Laha and Bowman (2012) claims were supported by Cummings and Bailenson (2016).

Cummings and Bailenson (2016) explained that advancing the technology in an immersive environment such as using natural body gestures, using HMDs with a larger field of view, and larger field of regard of the virtual environment results in an increased presence felt by the user because these technological enhancements increased the immersion the user feels. Increased immersions result in a higher feeling of being present in a virtual environment. It is realised that manipulating individual components of immersion has been suggested to be a good way of measuring immersion as these could provide empirical results.

Laha, Bowman and Socha (2014) researched VR analysis of volume data by studying the effects of system fidelity (field of regard, stereoscopy, and head tracking) on tasks in the scientific domain. The research concluded that tasks involving manipulation of spatial objects with isosurface (a 2-D surface that represents points of a constant value in a 3-D scalar field where each position has a value) visualisation using a head-mounted display provided better performance, however, tasks involving 3-D texture-based rendering required a higher field of regard when the levels of other components (stereoscopy and head tracking) were unchanged (Laha and Bowman, 2012). The field of regard refers to the observable space around the user that is filled with VR content whilst the field of view indicates the visible area for the user at an instance without turning their head. 3-D texture-based rendering creates rich, detailed, and often spatially complex graphics and the higher field of regard allows a larger viewing space for the complex object. This controlled experiment was done to evaluate individual components of VR immersion.

HMD based VR systems that have a very high sense of realism face several issues including latency between the movement of the user's head and the resulting change in the image seen in the HMD which is commonly called "Cue conflict" (Marks, Estevez and Connor, 2016). Cue conflict is a mismatch (conflict) between information from our senses that tells us where we are and how we're moving in the world). Cue Conflicts can be caused by latency caused by stalled experiments (Zhang et al., 2001) and this can sometimes cause physiological effects such as motion sickness (Lackner, 1992).

In the medical field, specifically neurosurgery, Pelargos et al.(2017) highlighted the barriers to commercial adoption of VR being the challenge of integrating mobility, vision, immersion, usability, flexibility, and wearability into the medical practice. The research also explained that VR systems should meet a threshold for depth of field, depth of focus, field of view, and image resolution for an immersive experience. Notwithstanding, there have been many pieces of research highlighting the potential of VR's immersion in the medical field due to positive results during tests and experimentation including immersive therapy for treating psychological, psychiatric, medical educational, and self-help problems (Lamson, 2011), simulating medical procedures (Jacobus and Griffin, 1998), and learning performance in VR simulation for robotic laparoscopy (Fiedler et al., 2007).

A study focused on immersive manipulation of molecules using HMD (da Costa and Nedel, 2017) in the field of Chemistry. This research studied the speed at which participants performed tasks concerning dragging movement, selecting, and rotating 3-D molecules in the HMD environment and on desktop screens. The research was aimed at finding the benefits of molecular visualisation and interaction in immersive virtual reality over a desktop screen. Interaction in the VR environment was found to have advantages over interaction using the desktop interface, such as shorter time for users to complete their tasks and with fewer errors. The research was particularly interested in comparing the performance of the desktop and the VR environment in interactions with molecules using as many natural gestures as possible. The research concluded that visualisation using the virtual environment through the HMD increased the user's understanding of the data and structure which would have required more effort using a desktop.

A review of selected research in immersion has highlighted contrasting usage and meanings of the terms immersion and presence. It has also highlighted that a key point of immersion is achieved by the optimal functioning of other VR metrics such as interaction and navigation. This could mean that immersion directly depends on the optimised functioning of these metrics. Research work has also shown an opportunity for immersive environments to enhance the performance and simulation of tasks in many industries.

3.4.2 Presence

Cummings and Bailenson (2016) defined presence as the “psychological experience” of being in a particular virtual environment. Presence describes a user’s sense of being in a VR environment (Zielasko, Weyers, Bellgardt, Pick, Meibner, et al., 2017) and this relies on the adequate selection of specific perceptual cues to activate emotions (Diemer et al., 2015). Passig (2009) argued that the presence provided by VR’s 3-D immersive experience motivates the user to become much more engaged and active in the tasks delivered in a 3-D environment. This conclusion was made in a study of 2-D and 3-D intervention programs for children with learning disabilities. In the experiment, participants could use an HMD with a head tracking sensor and a mouse to navigate, manipulate and interact with animated pictorial scenarios, children using the 3-D environment required less mediation due to the presence experienced as they were immersed in the environment.

In an experiment to measure the presence felt by users in a 3-D immersive environment, the open simulator environment was used with Moodle learning management system to provide virtual cybersecurity training in the form of games such as phishing, identity theft, and password protection scenarios (Xenos et al., 2017). It was concluded that the presence perception of the participants was high. This indicates that the implemented system was performing well. Similarly, the potential of VR in virtual simulated stores (Schnack, Wright and Holdershaw, 2018) was explored through an experiment to compare virtual shopping in immersive 3-D and 2-D desktop displays. The 3-D environment was made up of a stereoscopic HMD and motion-tracked hand controllers. The experiment allowed users to walk through a virtual shopping centre. Their motion was also being tracked whilst interacting with the virtual products. Findings indicated that interactive virtual reality (IVR) technology has the potential to outperform conventional desktop applications concerning telepresence. This finding agrees with previous research that investigated influences of control devices, display type and audio cues on telepresence (Seibert and Shafer, 2017)

With a focus on maximising presence amongst other aspects of VR in the virtual environment, Zielasko et al. (2017) described a deskVR scenario allowing an analyst or researcher to be fully immersed at their desk which can benefit in the gain in productivity when immersed in their data spaces. This finding also agrees with virtual environments increasing task performance (Barfield and Hendrix, 1995, Slater et al., 1996).

Cummings and Bailenson (2016) found out that users felt more present in a virtual environment when there are increased levels of tracking e.g., from joysticks to natural hand gestures, 2-D to 3-D, and lower to wider fields of view. These improvements significantly increase the feeling of immersion as compared to improvements in visual and audio content.

3.4.3 Interaction

Kjeldskov (2001) noted the central issue with human-computer interaction to be the creation of better ways of interacting with computers and explained that, the potential of HCI (Human-Computer Interaction) within VR can spring up interesting insights and new ideas that complement research in the HCI field. This argument is valid because VR has emerged as the current and next frontier of display and interaction technologies as well as agrees with Bowman and Hodges (1999), although this is specifically concerned with VEs (Virtual environments), it highlights interaction design problems due to no consideration or understanding of 3-D interaction tasks and techniques. This was also proven by Ren and O'Neill (2013) further explaining that different input devices have varying impacts on the performance of a task even by the same user. Different devices might have different properties due to intrinsic technicalities such as manufacturer specifics. Also, the design of the input device affects its suitability for certain tasks. Good VR interaction design requires the design process to consider the user's perceptions, capabilities, and anticipated behaviours in the virtual environment (Ren and O'Neill, 2013).

Kjeldskov (2001) argued that the terms 'interaction' and 'interactivity' have become buzzwords and as such have become very vague terms for computer applications. The research found out that, using the word interaction to describe all aspects of interactivity in a virtual environment provided little help in explaining interaction related issues encountered in the experiment. Kjeldskov (2001) proposed 3 terms, moving (ability and constraints of moving around in the virtual world), orientating (the process of a user obtaining a sense of realism in the virtual world), and acting (performing tasks in the VR environment) to better explain interaction. This is because each of the proposed terms (moving, orientating, and acting) are experienced differently based on factors that directly affect them. For instance, using an HMD allows the user to move around the physical world easily whilst the virtual object remains in focus in a fixed position. This also ensures greater flexibility of movement and the field of regard afforded by the HMD allows the user to go around the virtual object creating a higher sense of realism. The ability to rotate the virtual object also increases the sense of realism, however, HMDs can also be constrained with any data cable to a computer and in some cases, their weight can be a distraction, and these can affect moving around in the physical space. CAVEs present a unique consideration for interaction design. Fully immersive CAVEs provide a similar benefit to the HMD without the added constraint of a connecting data cable. Kjeldskov (2001) further proposed the consideration of display type and the design of interaction in virtual environments as this will enhance the quality of interaction methods.

Similar to the earlier discussion on the effects of other features of VR when evaluating a single feature, Wloka (1995) argued that the three features that make up virtual reality interaction are immersion, rich interaction, and presence. Enriching of interaction in a virtual environment can be enhanced using direct manipulation with interaction with objects using multiple input devices with higher degrees of freedom.

Muhanna (2015) explained interactivity as the virtual world giving participants the ability to interact and modify virtual objects. This study proposed a taxonomy of virtual reality to aid researchers alike.

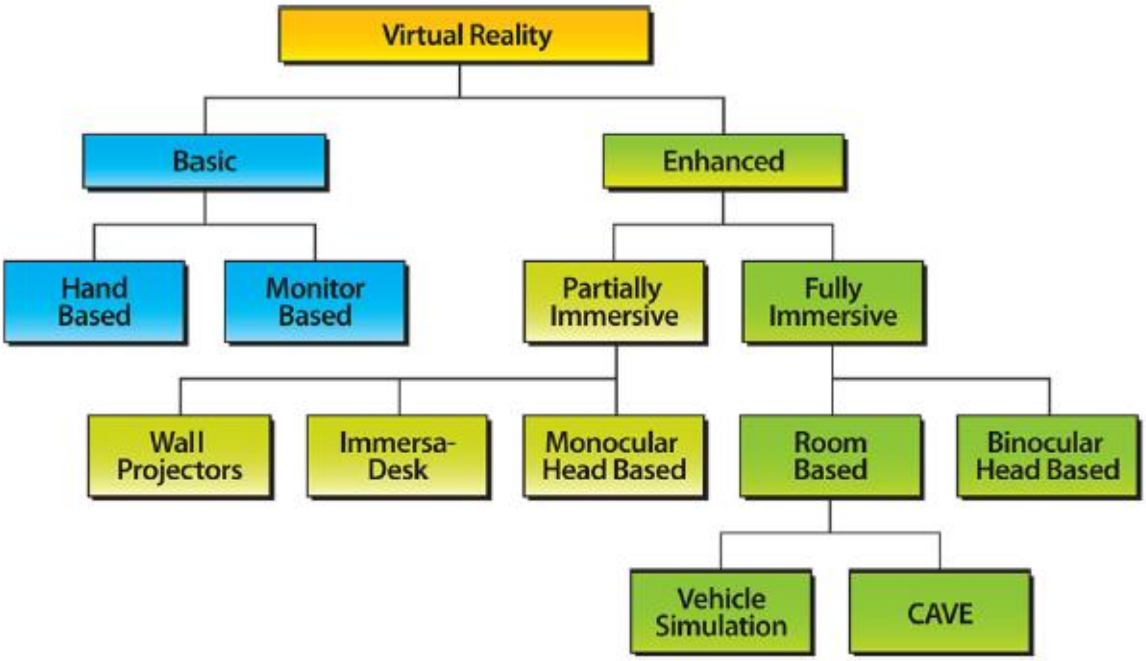


Figure 3.5 Virtual reality taxonomy based on technologies used and immersion levels (Muhunna, 2015).

Figure 3.5 depicts a classification of virtual reality based on the technology and the immersion levels. The enhanced VR is composed of a semi-immersive and a fully immersive experience comprising.

Ciftci, Zhang and Tin (2017) proposed a depth-based recognition framework that recognises mouth gestures to overcome the occlusion (Wu and Popescu, 2017) of the upper half of the display when using HMDs. The framework improved interactive virtual reality applications by providing information-rich facial actions as a way of interaction in VR. This novel approach to VR interaction was tested using a game and results showed the evaluation system correctly classifying 85.9% of the gestures. Wu and Popescu (2017) similarly identified the opportunity of improving VR navigation due to occlusion of parts of scenes by large HMDs by proposing a secondary view that provides multi-perspective to the scene to achieve the dis-occlusion effect. This experiment utilised the Microsoft HoloLens and found out that participants found that secondary perspectives are useful and help efficiently navigate in the virtual environment.

Ni et al. (2011) evaluated the use of freehand menu selection interfaces using tilt and pinch gestures. In their evaluation, they designed a novel menu selection interface called the rapMenu (Ni, McMahan and Bowman, 2008) which is controlled by wrist tilt and several pinch gestures. The rapMenu was found to perform better than a tilt menu (Rahman et al., 2009) in terms of speed and accuracy when the menu had a breadth of 12 or more items, and a rapMenu with better placement of menu items was found to be more usable. A similar technique was proposed earlier for VR (Dachselt and Hübner, 2007).

Coldham and Cook (2017) researched the usability of VR in the context of technology rejection amongst elderly persons. The aim was to understand the factors affecting technology rejection from the perspectives of the elderly. Most of the elderly are not exposed to VR or its related gaming and so the selection of first time VR users, over the age of 65 as participants also provided an opportunity for the experiment to yield accurate results. The setup featured a virtual Google Earth environment allowing the participants to experience the virtual earth using the HTC VIVE (HTC, 2020) and its controllers. The main task involved participants navigating to a known address in the virtual earth environment. The results indicated that many of the participants had little hope for VR, indicating that it did not have many benefits and others expressed the view that VR was for younger people, however, others also saw the potential uses of VR. The research concluded that certain physical aspects of the design of VR components such as data cables from HMDs can limit the experience of the elderly. In recent times, VR devices with no cables have started going mainstream such as the HTC VIVE Pro 2 (HTC, 2021) and The Oculus Quest 2 (Facebook, 2020).

In the field of architecture, VR does not only help in the planning and design of architecture by reducing the cost of the process using virtual prototypes, but it also enhances the user-customer relationship by enabling a new form of visualisation and interaction with the product. By allowing a user to visualise their design prototypes, it further enhances the engagement between user and customer through immersive demonstration of project prototypes (Calado et al., 2013).

Virtual reality has also been found to improve big data analytics. In a study visualising the Twitter feed on MIT campus with a gamepad for interaction, it was concluded that virtual reality can also be used as a data visualisation platform and that a more immersive environment enables interaction (Moran et al., 2015). The research further suggested that the use of 3-D interactions such as hand inputs rather than a gamepad could potentially improve the overall interaction experience. Like many other experiments, the application was built with Unity 3-D and the Oculus Rift DK1 was used as the VR device.

3.4.4 Knowledge Improvement

Eden and Bezer (2011) explored the use of an HMD with a head tracking sensor and a mouse for navigation and tracking. In their study comparing three-dimensions and two-dimensional intervention programs, the research investigated the behavioural aspects of children with intellectual disability. The study used animated 2-D scenarios (Eden and Passig, 2007) and the same scenarios were converted to 3-D and results indicated less mediation (They required less help from the coordinators) involved for users in the 3-D IVR (Interactive virtual environment). This means the users experimenting with the 3-D environment could learn and understand needing less help.

Eden and Bezer (2011) researched interventions needed by children with intellectual disability (ID) in 3-D virtual reality environments and 2-D alternatives (presentation of pictorial images to the students). Intellectual disability refers to a condition that makes it difficult for sufferers to learn and especially understand abstract concepts. Children aged between nine and 21 were grouped based on their levels of ID and were given a means of asking for help during the experiment. The experiment itself was composed of levels of scenarios that the children were expected to re-arrange, and the level of help they requested or received from the instructors to complete the scenarios determined the level of help offered by their instructors. The research concluded that the experiment using the virtual environment required less instructor help compared to the 2-D desktop-based experiment.

Xenos et al. (2017) evaluated knowledge improvements in virtual reality using a virtual environment showing several information security threat scenarios (phishing, identity theft, password security, and social networking-based security). These scenarios were presented to participants and allowed the participants to navigate around the environment while learning about these threats. The environment was designed in a way to evaluate the effectiveness of using virtual reality to learn from previous experience. As such, users were presented with questions as they learned about different security threat scenarios. Based on participant responses, the researchers concluded that virtual reality can be a useful tool for teaching and learning.

Coogan and He (2018) evaluated learning in brain-computer interfaces by testing the performance and learning rates using a VR setup with HTC VIVE (HTC, 2020) and a desktop interface and found no knowledge improvements for users using VR compared to the desktop screen. The research rather found out that users with prior experience of VR performed better in the VR environment. This observation can be due to first-time user adjustments to the HMD and the use of the controllers. The subtle difference here with other experiments include the brain control interface (BIC2000) device that was placed directly on top of the head together with the HTC VIVE. This device was also noted to sometimes interfere with the HTC VIVE. It can fairly be assumed that this might have added extra weight to the already heavy HTC VIVE HMD. Latency issues may have also affected the experience of users as brain control interface signals are routed to the game engine and accepting physical input into a complex mix of interactions.

Several other researchers have found some evidence of knowledge improvements using VR include learning for elderly cohorts (Coldham and Cook, 2018), geography education using a collaborative and immersive environment (Doležal, Chmelik and Liarokapis, 2017), and learning in cell biology for undergraduate students (Johnston et al., 2018).

3.4.5 Navigation

Navigation is one of the core tasks that people perform in virtual environments and spatial knowledge gained in the environment is used to navigate successfully (Sousa Santos et al., 2009). However, contrary conclusions were drawn in Sousa Santos et al. (2009) where it was observed that users performed better in their navigational task on 2-D desktop screens than using an HMD. This could be because of several complaints of users concerning the virtual environment setup including distractions by cables which might have affected their experience and overall performance on the task as well as the weight of the HMD and the brain control interfaces.

Ard et al. (2017) illustrated the superior navigation and conceptual advantages of viewing complex neuroscientific data in an immersive and stereoscopic context. They developed a virtual reality prototype environment called NIVR (Neuro Imaging in Virtual Reality) built with Unity3D (Unity Technologies, 2019). The experiment used the HTC VIVE and allowed users to manipulate MRI scans as well as perform volumetric tasks in the immersive environment. The immersive tractography experience, as well as the neuroscientific near-field experience, were particularly noted to be a first of a kind by encompassing all sorts of neurological data into a single VR viewing program and this was achieved by pre-rendering data to overcome the length of time needed to render the huge frames in neurological data. The exceptional navigation experienced allowed users to manipulate brain segments offering a compelling neuroanatomical experience, or 'digital dissection' of an actual brain. Similar positive results for navigation were concluded in Johnston et al(2018) where 2 virtual reality environments were used to visualise the internals and surface of a cell allowing the user to navigate seamlessly between both environments.

Virtual navigation in 3-D environments was found to be challenging with the standard mouse and keyboard interface (Cockburn and McKenzie, 2002) and thus a recommendation for a new approach that facilitates more intuitive exploration of 3-D visualization to unlock the full potential of areas such as scientific analysis (Steed et al., 2017). Steed et al. (2017) focused on navigation in immersive analytics using neutron scattering and the Oculus Rift HMD. They concluded that using HMD represents a viable path toward unlocking the full potential of large and complex scientific data because the HMD provided a more immersive and enhanced way of interacting with the data and this enhanced the discovery of knowledge in a short time.

Similarly, a mix of results in the study of the usability of virtual reality was identified in Coldham and Cook (2018). The research explored how elderly cohorts could learn navigation of tasks in the virtual environment. Making use of Google Earth-based navigational tasks, it was concluded that users generally expressed their view of the tool being exploratory however many more indicated their frustrations that include difficulties with using triggers and buttons on the hand wands, headset, and tethering issues such as problems with the weight of the headset. This group of users could potentially benefit from wireless HMD displays and full body tracking devices which might have improved their experience.

3.4.6 Performance

The measure of the performance of a virtual reality system was noted to be more beneficial when this is done together with usability studies of the same environment being tested (Toma, Gîrbacia and Antonya, 2012). This approach would have improved research evaluating immersive environments (Cockburn and McKenzie, 2002b, Wang and Li, 2004, Demiralp et al., 2006, Chang et al., 2016).

The performance of a virtual reality system in Computer-Aided Design (CAD) was evaluated using an experiment allowing participants to model and assemble interfaces using a 2-D desktop display and an immersive HMD interface with a Holo-CAVE (Toma, Gîrbacia and Antonya, 2012). Results showed that the immersive environment provided a better awareness of the bounds of the object since the user was immersed in an environment containing the data. The overall immersion was also enhanced with natural interaction using speech. The results indicated that the environment was not more efficient compared to the 2-D desktop environment because it involved physical hand motions using the controllers (Toma, Gîrbacia and Antonya, 2012).

A more conclusive result showing a VR interface outperforming traditional tasks on a 2-D display was detailed in an experiment assessing performance for navigational tasks in a game scenario (Sousa Santos et al., 2009). On the other hand, Coogan and He (2018) found out that user performance did not increase compared to traditional 2-D desktop displays in an experiment using brain control signals to manipulate objects in 3-D and 2-D. Both experiments made use of a VR application built with the Unity game engine and whilst Sousa Santos et al.(2000) used a game-based scenario approach with a mobile-based VR device, and Google Cardboard, Toma, Gîrbacia and Antonya (2012) used a CAVE (Cruz-Neira et al., 1992) for their VR demonstration.

The performance assessment of carotid angiography (The study of the health of veins and blood flow using x-rays and diagnostic imaging tools) was simulated in VR to measure reliability amongst others (Patel et al., 2006). The experiment was to ascertain whether virtual reality could be a valid means of proficiency assessment and improvement in procedural-based medical skills. It was also an opportunity for using VR to explore this complex and high-risk procedure as the demand for learning had risen amongst medical professionals (Gallagher and Cates, 2004). Using a VIST simulator (Vascular Interventional System Trainer – VR simulator system for training in the diagnosis and treatment of diseases related to the blood vessels). Patel et al. (2006) employed the use of seven measures of the participant's operative performance with repeated tests, and it was concluded that the VR simulation procedure enhances the reliability of the assessments during virtual operations on carotid angiography. This result also agrees with similar studies for autism spectrum disorder where the results revealed that virtual environments have the potential to foster improved performance during social communication tasks among the participants (Raj and Lahiri, 2017).

3.4.7 Usability

Toma, Gîrbacia and Antonya (2012) argued that usability is the ability to carry out tasks effectively, efficiently and with satisfaction and that evaluation is the way to measure that usability. Previous research (Bowman, Gabbard and Hix, 2002) had identified issues with usability studies including issues with the reliance on one method such as the difficulty of usability studies based on heuristics (user guidelines) because of no existence of published work on virtual environment user interface design guidelines. Laha and Bowman (2012) had also argued that limitations with current usability studies for VR environments lacked generalizability of results to other VR systems and domains, meaning that in most cases, experiments are designed from the ground up due to the often different datasets required in these environments.

Depth and gestalt cues (How the brain perceives and interprets images) in text label layouts were identified to affect the usability of the immersive environment. Polys, Bowman and North (2011) presented an evaluation methodology for usability that seeks to test proximity and connectedness cues in viewport space layouts. It was concluded that the consistent performance of continuous scaling across tasks suggests that legibility is more important than relative size.

Toma, Gîrbacia and Antonya (2012) designed an experiment to evaluate a VR-CAD system compared to a traditional CAD system that runs on a 2-D display, for modelling as well as assembly user case scenarios. The virtual environment was in a Holo-Cave complete with voice commands and hand motions for a specific task and in each scenario, users could model objects as a measure of task performance to measure the usability of the VR-CAD system. It was concluded that VR presented a physical stress factor because the distance for hand movement in VR was bigger than in desktop.

The system usability standardized questionnaire was used to measure the usability of a virtual 3-D learning environment using the Opensimular tool and augmented with the Moodle learning management environment (Xenos et al., 2017). The focus was on usability aspects such as efficiency, perspicuity, and dependability. Game-based scenarios were presented to student participants revolving around computer security and all students were able to successfully conduct their learning activities within the 3-D virtual environment scenarios. The usability evaluation results, SUS (System Usability Scale) score of 67.3/100 is acceptable and comparable with usability scores achieved at similar systems.

A more complex usability study was performed on a similar game based scenario for practising urinary catheterization (Butt, Kardong-Edgren and Ellertson, 2018) and the evaluation performed using the system usability survey showed results with an overall rating of system usability at a mean of 72.5%. This positive result also agrees with a study into the potential of VR for autism spectrum disorder (Pradeep Raj, Oza and Lahiri, 2017) where a virtual environment was created that allowed autism patients to practise communication with others in a virtual world. The setup allowed the participants to interact with virtual characters

Hou, Dong and Yang (2017) noted the popularity of using eye-tracking to measure usability but hinted at the difficulties in having accurate data because of the combination of wearing a head-mounted display. Research has also pointed out certain issues with measuring the usability of virtual environments such as whether the user experience realised are because of VR or the product being studied. The approach by Hou, Dong and Yang, (2017) made use of electrical signals to measure the usability of game UX by recording the electrical signals emanating from the brain and studying the frequency band and functional connectivity as good indicators to VR game UX.

Bowman, Gabbard and Hix, (2002) evaluated virtual reality research along lines of the usability method evaluated. The research‘s classification of VR usability research (Summative, interview, evaluation, user task analysis, heuristics, and questionnaire) helps broaden adoption, however, Bowman et al. (2002) also identified many issues with current usability studies for the virtual environment including issues with the reliance on one method such as the difficulty of usability studies based on heuristics (user guidelines) because of no existence of published work on virtual environment user interface design guidelines.

More recent usability studies using immersive virtual reality include using game-based learning in education (Ketelhut et al., 2013). This research focused on enhancing science-based assessments by assessing students’ learning using a virtual environment. The general view was that using virtual environments makes the learning of science more practical and meaningful and virtual environments can give a realistic view of the student’s learning and understanding on a particular topic. The research identified four (4) characteristics of good quality science assessments; (1) assessments with questions integrated with real-world scenarios such as giving a user a sprinkler with a tree and a house, expecting the user to sprinkle on or at the root of the tree instead of the house, (2) situating questions in the right context to help the student apply their learning, (3) providing efficient ways of grading, and (4) providing reliable and valid assessments that can help the student’s learning.

Integrating the assessments with such real-world scenario content places the assessments in their right context, allowing the student to apply knowledge of the content for practising. This method of VR learning and assessments also provides valid and reliable grading schemes and assessments. The research used characters that were incorporated into the design of a virtual assessment tool called, the “Sheep Trouble”. “Sheep Trouble” allows students posing as scientists to interact in a virtual world of farmers and their sheep in a local community. The students help the farmers by investigating and solving some scientific problems such as recommendations for sick sheep. An evaluation of the virtual environment found that students actively engaged with the content and assessments, however, a recognised challenge was to ensure that students understood the environment as an assessment and therefore carry on with taking the test than spending time exploring the virtual environment. This challenge can be mitigated by allowing participants to experience similar virtual environments to satisfy their excitement before using the assessment-based VR. Similarly, Özgen, Afacan and Sürer (2021) explained that the challenge with usability in virtual education environments lies in exploring new ways of design learning. The research proposed a design for architectural students that involves using virtual reality to design simple shapes.

Using a combination of user task analysis and heuristics, Toma, Gîrbacia and Antonya, (2012) studied the performance and usability of an immersive VR-Computer Aided Design (CAD) system. Users performed tasks involving modelling and assembling CAD parts. The time users spent performing these tasks were analysed as well as a subjective questionnaire to ascertain users’ overall experience. Results showed that even though the immersive virtual reality environment provided a better sense of the depth of the virtual CAD objects, the performance of the tasks in the virtual environment did not convincingly show improvements over the traditional 2-D desktop-based workspace. Users may have been very used to perform these tasks on a 2-D display. However, this finding points to other usability research into developing experiments that make use of other ways to gesture such as using natural interactions like sight and voice.

Another usability study in the automotive industry is work done by Langley et al.(2016) which found virtual training to be effective in reducing error during task performance when compared to traditional training involving the use of 2-D desktops and other materials. This study also used a subjective method using a questionnaire and an objective-based approach which was expected to make the results more rigorous. The research found that fewer errors were encountered in the virtual training environment compared to the 2-D desktop.

Xenos et al. (2017) employed both an objective and subjective approach to usability studies of their VR application using a game-based virtual environment built with tasks in the information security domain for educational purposes. Participants generally gave positive feedback that the tool helped them gain knowledge whilst a small percentage responded negatively.

Steed et al. (2017) visualised neutron scattering in a 3-D collaborative VR environment. This experiment was to evaluate the efficiency of using the virtual environment. One stage of their usability testing used several material science researchers at scientific workshops, group meetings, and one-on-one demonstration with users less proficient in 3-D visualisation tools and the feedback was generally positive. This approach was specifically aimed at enhancing their prototype 3-D visualisation tool for material sciences.

It can be noticed that there exists the possibility of the user's excitement or otherwise of using the virtual environment for the first time skewing objective and subjective results. A user who has no experience of immersion might generate usability data that contains logs for adjustment. The researcher was very excited during the first time using the CAVE and HMDs that his focus kept hallucinating between controlling the hand devices to adjusting to the weight of the head mount and he had to use them for a while to get used to and focus on analysing the virtual world. This problem can equally be solved by ensuring that participants have experience using any HMDs at least two (2) times before the experiment. This will make them feel familiar with the displays which will then ensure their focus is on the virtual world other than familiarising themselves with the displays during the experiment affecting the data generated. This can be done with a pre-experiment phase where the same experiment is done 3 times by each participant with the aim of the first two logs being adjustment logs.

Some limitations with usability experiments in virtual environments have been identified. Bowman, Gabbard and Hix (2002) identified usability studies that do not involve users but rather modelling of user's interaction such as Goals, Operators, Methods, and Selection (GOMS) and indicated the nonexistence of such models, as well as the low cost of heuristics and performance model evaluation, makes these methods attractive. Further to this, Bowman et al. (2002) indicated the importance of automated analysis which could reduce the need for multiple evaluators during usability studies and allow the automatic collection and possibly analysis of data from many users. It was also suggested that, due to the rapid changing of VEs, using different VEs for the same experiments are encouraged where possible.

3.5 Classification of VR Research Experiments

Table 3.1 classifies VR usability experiments into domains, the metrics being evaluated, the devices used, summary, and findings Sousa Santos et al. (2009). The classification allows an easier and faster overview or summary of advances made in VR research as discussed in this thesis. The table shows the most common devices used for VR experiments. It also shows findings of each experiment, allowing an analysis into how a particular affordance of VR was experienced. Results show there is a relation between the performance of users in immersive VRs using HMDs and desktop-based VR.

Domain	Reference	Metrics	Devices	Experiment	Findings
E-commerce	(Animesh et al., 2011)	Interactivity, presence	PC, stereoscopic display	Virtual shopping in Second Life application	Presence is affected by the density & stability of the experience
	(Cheng, Chieng and Chieng, 2014)	Immersion	Six-axis simulator	Measure virtual experience in the VR environment	Vividness improves experiences
	(Martínez-Navarro et al., 2019)	Presence	HTC Vive	Evaluate a virtual consumer experience	Positive consumer responses were experienced in the virtual e-commerce environment.
	(Morotti, Donatiello and Marfia, 2020)	Usability	HTC Vive	Evaluate a virtual clothing store with vocal commands (Amazon Alexa Voice Assistance)	Most participants rated the experience as very natural.
	(Alzayat and Lee, 2021)	Presence, Interaction, Usability	Oculus Rift	Explore how VR engagement with products	Products such as tools (hammer and pencil) are more for VR environments than others

Domain	Reference	Metrics	Devices	Experiment	Findings
				affects perception of fun and usability	such as clothing and shoes.
Education	(Barfield and Hendrix, 1995)	Presence	Stereoscopic display	Navigate a virtual representation of Stonehenge and search for a rune, inscribed upon the wall of one of Stonehenge's Edifices	Subjective presence was less using an update rate of 5 and 10 Hz when compared to update rates of 20 and 25 Hz.
	(Cockburn and McKenzie, 2002b)	Performance	Desktop	Repeat storage and retrieval exercises for different conditions to access 2-D/3-D cards using spatial memory	The spatial arrangement of information can affect the user's memory of the location of information.
	(Spagnolli and Gamberini, 2002)	Immersion, presence	Unknown	Immersion, Interaction, and movement in the virtual library.	Immersion is determined by both subjective and physical environments.
	(Eden and Passig, 2007)	Knowledge improvement	Generic HMD	Children between the ages of 9 and 12 re-ordered images of	3-D VR enhances the understanding of concepts and should be explored for teaching children.

Domain	Reference	Metrics	Devices	Experiment	Findings
				events in sequence to understand how sequential time is perceived by them in 3-D VR and 2-D	
	(da Costa and Nedel, 2017)	Immersion, interaction	HMD, desktop	Interact with molecules in the HMD	VR is faster and more suitable for molecular interaction tasks.
	(Ciftci, Zhang and Tin, 2017)	Interaction	Samsung Gear VR	Mouth gestures were evaluated as a form of interaction in virtual reality to compensate for the hiding of the upper part of a user's face when using an HMD. Participants controlled a virtual avatar with head and mouth motions in a game scenario.	The mouth gesture recognition system performed well with fewer errors.
	(Coldham and Cook, 2018)	Usability, navigation	HTC VIVE	Senior citizens were made to navigate to an	Acceptance of virtual reality is increased with

Domain	Reference	Metrics	Devices	Experiment	Findings
				address on the google virtual earth application.	the elderly when the task to be performed is useful
	(McFaul and FitzGerald, 2020)	Knowledge improvements and Usability	Google Cardboard Version 2	Law students evaluated a VR environment to practice their presentation skills	There was a mixture of positive and negative responses on the usability and content of the practice application.
	(Marks and Thomas, 2021)	Knowledge improvements	Oculus Rift and HTC Vive	Evaluate the use of VR devices for teaching and learning in a university	VR learning enhanced student's learning outcomes
Games	(Jennett et al., 2008)	Immersion	2-D desktop based VR	Users performed various VR and 2-D tasks from playing games in the Second Life application to clicking randomly appearing square boxes. Users were then presented with a questionnaire to measure how immersed they	Individual-based factors affect how immersed a user feels in a VR environment.

Domain	Reference	Metrics	Devices	Experiment	Findings
				felt undertaking the various tasks	
	(Hou, Dong and Yang, 2017)	Usability	HTC VIVE	Gamers were employed to play several VR games and their electroencephalogram (EEG – brain signals) was measured to find a correlation between the subjective ratings of the game to the signals read from the brains whilst the users are playing the games.	The user’s subjective perspectives of a game are correlated with the brain signals read when the user is playing the game.
Medical	(Zhang et al., 2001)	Immersion	CAVE	3-D models of brains were put in a VR environment and users could view and interact with the virtual objects	immersion and interaction can help a user understand complex neural structures of the brain.

Domain	Reference	Metrics	Devices	Experiment	Findings
	(Demiralp et al., 2006)	Performance	CAVE, 2-D desktop VR	Users performed several tasks in Fishtank VR and the CAVE.	Users performed tasks much faster on the fishtank VR than in the CAVE
	(Eden and Bezer, 2011)	Knowledge improvement	Generic HMD	Children with ID (intelligent disability) performed tasks in 2-D and VR	Children who performed their task in VR encountered fewer errors and needed little help from the instructors
	(Ard et al., 2017)	Navigation, immersion	HTC VIVE	Users were presented with a less complex array-based representation of MRI scans. Users could then place the representation in labelled sockets corresponding to predefined visualisations of the brain. Users also could interact with the visualisation.	VR can enhance the understanding of neuroimaging data through the immersion and navigation provided.
	(Butt, Kardong-Edgren and	Usability	Oculus Rift, desktop	A game-based learning approach was adopted to	User survey feedback indicated a general appreciation and

Domain	Reference	Metrics	Devices	Experiment	Findings
	Ellertson, 2018)			train nurses on urinary catheterization in virtual reality.	acceptance of the virtual training
	(Coogan and He, 2018)	Knowledge improvement, performance	HTC VIVE	The experiment evaluated VR-based BCI (brain-computer interface) for performing tasks	BCI based tasks in virtual reality are not hampered by the feeling of immersion.
Science	(Demiralp et al., 2006)	Performance	CAVE, 2-D desktop-based VR	Users performed various tasks in a CAVE and a desktop-based VR system	Tasks performed in the desktop-based VR system (Fishtank VR) were noticeably faster than in the CAVE
	(Johnston et al., 2018)	Navigation, knowledge improvement	Google Cardboard	Some biology students explored the structure of cells in virtual reality before taking an exam	VR exploration of biological cells improved knowledge
	(Nagao et al., 2017)	Interaction and Immersion	Tiled Wall Display, HMD	Users were immersed in a dataset where they could use hand gestures to interact with the virtual objects to understand the effects of virtual reality on analytics.	Hand gestures provide seamless navigation and integration around virtual objects

Dom ain	Reference	Metrics	Devices	Experiment	Findings
Genera l	(Kjeldskov, 2001)	Interaction	8-sided cube with a cylindrical panoramic display	Full and partial Immersive displays were studied to understand their effects on immersion and presence in the virtual environment	Same interaction techniques such as freehand gestures and voice control have different effects in CAVEs and panoramic displays.
	(Wang and Li, 2004)	Interaction	Desktop VR System	Users were trained in a virtual application for the maintenance of a refinery pump. Users were able to interact with the pump by fixing and unfixing parts.	Virtual reality-based training can enhance the training of maintenance staff.
	(Sousa Santos et al., 2009)	Usability, navigation	HMD and Desktop display	Users explored a maze game, collecting virtual objects. The experiment allowed users to navigate the virtual environment	Usability on the desktop VR was better than on the HMD system.

Domain	Reference	Metrics	Devices	Experiment	Findings
				using an HMD as well as on Desktop VR	
	(Steptoe, Steed and Slater, 2013b)	Interaction, Knowledge improvement	CAVE	Users explored interacting and controlling avatar-like virtual object and their tail	Humans can adjust their sense of positioning in the virtual to control virtual human-like objects.
	(Chang et al., 2016)	Performance	Google Cardboard, Samsung Gear VR, 3Glasses D2, Oculus Rift DK2	Comparative analysis with the physical and virtual world on timing and positioning Accuracy	There exists a trade-off between position and accuracy of VR systems and a combination
	(Wu and Popescu, 2017)	Navigation	Generic VR HMD and Microsoft HoloLens.	Users evaluated using AR and VR to provide multiple perspectives for visualisation	The multi-perspective approach showed different aspects of a view to improving navigation around the virtual environment.

Domain	Reference	Metrics	Devices	Experiment	Findings
	(Zielasko, Weyers, Bellgardt, Pick, Meibner, et al., 2017)	Presence	Generic HMD	Users performed normal desktop-based tasks in virtual reality to study to what extent users' desktop-based tasks can be performed in virtual reality	Immersion helped with performance in virtual reality.

Table 3.1 Virtual reality experiment classification (Sousa Santos et al., 2009, Ankomah and Vangorp, 2018). The classification presents findings based on the domain of the experiment, the metrics and devices tested, and the findings of the experiment.

3.6 Summary

This chapter has discussed virtual reality in detail. It has reviewed virtual reality research in different industries and has shown the benefit of visualisation in scenarios such as health rehabilitation and education.

The metrics discussed were 'presence' and 'immersion' which are the core VR affordances, 'navigation' and 'interaction' being the actions that can be done in the environment, 'knowledge improvement' which is an outcome and 'performance' and 'usability' as evaluation metrics (Ankomah and Vangorp, 2018).

This chapter has achieved objective one (1), "*Gain a critical understanding of the history, current state, and challenges of virtual reality through a literature review".* The knowledge gained from the discussion of the metrics has highlighted the most important metrics to focus on in the ICPVR application (CHAPTER 8 ICP Virtual Reaility Application (ICPVR)). For instance, the knowledge gained helped develop an approach for ICPVR that considered how each metric will impact the user in terms of enhancing the user's experience specifically with matching the 3-D protein shapes and interacting with the virtual world. Similarly, the literature review informed the provision of features such as teleportation and the use of natural gestures such as grabbing and rotating molecules in ICPVR which were aimed at contributing towards enhancing the overall usability and immersion of ICPVR.

Despite various difficulties, issues, and experiences, VR continues to remain one of the major technological topics in entertainment, education, and advertisement, especially with the latest development of computer graphics, image processing, computer vision, machine learning, and advanced GPU hardware.

4 CHAPTER 4 KMEANSICP

4.1 Introduction

This research was interested in improving aspects of the ICP algorithm that can degrade the overall performance of the algorithm in the presence of larger datasets. For instance, the closest point search method of the ICP algorithm does a naïve closest point computation for each point. This will usually impact on the performance when there are millions of points in the shape. One of the approaches to reduce the impact on performance and speed up the matching is to partition the search space in smaller clusters. KMeansICP does this partitioning using the kmeans algorithm (Krishna and Narasimha Murty, 1999). The benefits of the kmeans partitioning over other types of partitioning such as partitioning by atom type is that, kmeans algorithm attempts to provide equal-sized partitions even though this is very much affected by the complexities of the shape such as how evenly distributed the points are.

This chapter discusses the KmeansICP algorithm. It explains the k-means algorithm, the steps in the KmeansICP algorithm, the computational complexity, and the limitations of the algorithm. This chapter and chapters (CHAPTER 5 SUBSETICP+ and CHAPTER 6 TAGGEDICP) aim to achieve objective three (3) of this research, "*Develop shape matching algorithms based on the Standard ICP".*

The k-means algorithm (MacQueen, 1967) partitions a dataset into a specified number k of clusters. The aim is to have each data belonging to the cluster with the closest mean (centroid). The algorithm initialises by randomly creating positions for the centroids. Each iteration involves, assigning all the points to the clusters with the smallest distances and re-locating the centroids of these clusters. The iteration stops when there is no considerable change between the previous cluster means and the next calculated means for all points. This also means that all points would have been in their final cluster and wouldn't be re-assigned to a different cluster.

KmeansICP implements the k-means algorithm (MacQueen, 1967) in the ICP registration as a technique for partitioning. The rationale for using k-means lies in the use of the centroids since the centroids provide a pointer to the closest cluster for each data point cloud.

KmeansICP performs a standard k-means clustering to group the model point clouds. The search for some correspondence then starts by finding the closest centroid to the data point (point-to-centroid). The correspondence search space is then reduced to the clusters of the selected centroids. The correspondence search method of KmeansICP is the same method used in the standard ICP algorithm. The difference is in the reduction of the search space by the k-means clusters.

4.2 Algorithm

According to Saboori, Parsazad and Sadeghi (2010), the k-means algorithm performs the following steps;

1. Select k points at random as cluster centres or centroids
2. Assign data points to their closest clusters according to the Euclidean distance function.
3. Calculate the mean positions of points in each cluster and update its centroids
4. Repeat steps 2, 3 and 4 until the points don’t get re-assigned to different clusters.

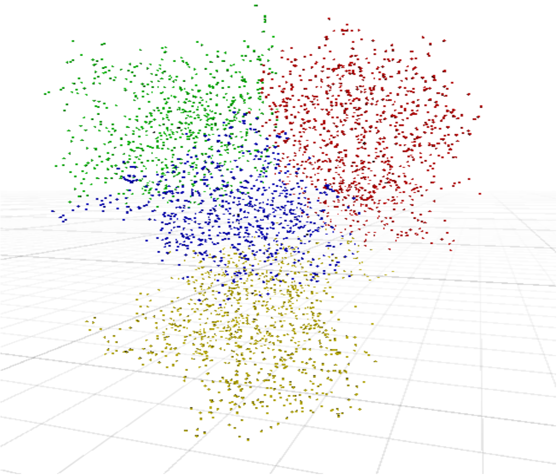


Figure 4.1 Clustering of a 3-D Point cloud into 4 groups (red, green, blue, yellow partitions) using k-means algorithm and Unity3D. KmeansICP uses the k-means algorithm to derive a set number of partitions from the point cloud to reduce the correspondence search space.

KmeansICP uses this method to partition the model point cloud into clusters. The search for correspondence is performed by first finding the closest centroid to the point in the data shape (point-to-cluster), then the cluster belonging to that centroid is searched for the correspondence (point-to-point). KmeansICP uses a point-to-point objective function to minimise the error between the two point clouds. The minimisation is done through an iterative process of computing the least squares quarternion rotation and a translation vector that is applied to the data point cloud.

Assume the search for correspondence for a point $\boldsymbol{p}$ in a point cloud P_x clustered into a number c of clusters, where $c = 1...C_n$. The algorithm finds point p's closest cluster based on the distance between $\boldsymbol{p}$ and its closest centroid of P_x , P_{xc}, the closest point of point $\boldsymbol{p}$ in P_x is the closest point of point $\boldsymbol{p}$ in P_{xc}, because the algorithm does a naïve closest point search in the closest cluster by comparing the distance *d,* of every point in P_{xc} and finding the minimum distance d_{min} . This distance is then appended to the list of closest points, Y_k and this is repeated for every other point in the data shape.

The algorithm statement is

1. Cluster the model point cloudusing the k-means algorithm into a number k of clusters.
2. Initialise Iteration to 0, the maximum number of iterations, set change in MSE threshold, transformation (R, $\boldsymbol{t}$)
3. Apply the transformation (R, $\boldsymbol{t}$) to the data point cloud
4. Compute the closest points in the model point cloud for each point in the data cloud

5. Find the closest cluster to the data point by calculating the distance to each centroid (point-to-centroid)
6. Find the closest point for the data point by searching the cluster space (point-to-point) (Computational cost: $O(N_x N_p)$)
7. Compute/update the registration transformation (Computational cost: (O (N_p)))
8. Apply the registration to the data point cloud (Computational cost: O (N_p))
9. Terminate the iteration when the iteration count or MSE threshold is reached.

The computational complexity of the closest point search method of KmeansICP is O (N_x(C_n+N_m)), where C_n is the number of the clusters in the model point cloud and N_m is the average number of points in each cluster. Since C_n+N_m is usually smaller than $C_n N_m$, it is more efficient than the closest point search method for the standard ICP Closest Point Search.

```
function CLOSEST POINT SEARCH ($P_k$ $X$)
    closest points $Y_k$ $\leftarrow$ $empty\ list$
    Cluster X into $C_n$ clusters with centroids $X_c$
    for all points p in $P_k$ do
        closest cluster C ⟵ CLOSESTCLUSTER (p, $X_c$)
        closest point distance $d_{min}$ $\leftarrow$ ∞
        closest point y ⟵null
        for all points x in C do
            distance d ⟵ DISTANCE (p, C)
            if d < $d_{min}$ then
                closest point ⟵ x
                closest point distance $d_{min}$ $\leftarrow$ $d$
                append closest point x to closest points $Y_k$
   return $Y_k$
```

4.3 Computational Complexity

Assume a data point cloud, P with N_p points and model point cloud X with N_x points, with X clustered into a number C_n of clusters and an average number N_m of points per cluster, the computational complexity will be O ($N_x(C_n + N_m)$). This is based on each point in the data point cloud making a point-to-cluster distance to find the nearest cluster ($N_x C_n$) and then a point-to-point distance to find the corresponding point within the selected cluster ($N_x\ N_m$).

KmeansICP reduces the search space by a factor (fractional size of the clusters, assuming that all clusters are fractions of the whole point cloud) depending on the number of clusters and the complexities (uniformity in cluster sizes) of the shape.

Consider the fractional partition sizes explained in TaggedICP's complexity in Section 6.3. If k-means creates $k = 5$ equal sized partitions in the model point cloud, then the fractional partition sizes will each be ${}^{1}/_{k} = 2$ and the correspondence search space will be reduced by a factor, F= (k+1/kN_x)/N_x=0.2, while k is usually much smaller than N_x.

TaggedICP partitions the search space by the abundance of different atoms in the molecule, whilst KmeansICP partitions the search space by the mean distances of the points.

If the points in the shape are non-uniformly distributed, the efficiency of the clustering for the shape matching is reduced because evenly distributed points will result in equal-size clusters that will evenly reduce the complexity of the point cloud in terms of finding the right correspondence.

Since the points are clustered based on the closest centroid, the resulting correspondence will not always be a true match but an approximated match in some cases. For each cluster, the higher the number of points farther away from the centroid the lower the accuracy of the matches. This is because the mean of the cluster points represented by the centroid is an approximation as such this approximation will be less accurate for points lying close to the boundary of the cluster.

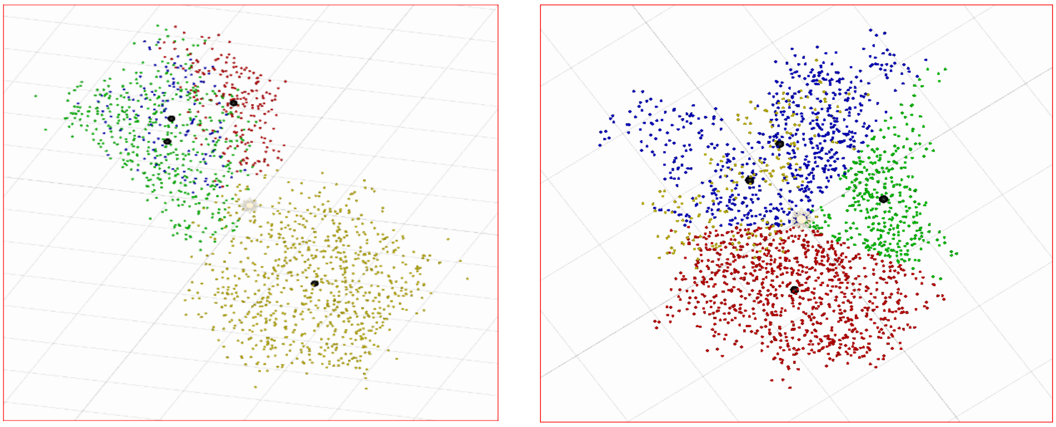


Figure 4.2 k-means clustering of two-point clouds into 4 groups (red, yellow, green, blue clusters with black centroid). (LHS) Clusters show uneven distribution which is based on the shape of the point cloud. (RHS) Clusters show an improved but still uneven distribution of points. KmeansICP depends on the k-means clusters which could result in uneven clusters depending on the shape of the point cloud.

Considering the k-means clustering of two point clouds in Figure 4.2, the uneven shape of the point cloud resulted in the uneven clusters. A similar but improved cluster distribution is seen in the (RHS) image. A detailed visual analysis of the shapes shows overlaps of the points in some regions.

The search for correspondence for a point in the data point cloud starts in the model point cloud with a point-to-cluster match to select the closest (cluster) to the data point. Then a point-to-point correspondence is used to establish the closest point in the established point-to-cluster search space.

There is always the likelihood of much better matches in other clusters in the model point cloud than in the closest cluster to the data point. This is especially likely for points whose matches are farther away from the centroid in the model cloud cluster and of a non-uniformly distributed model point cloud because the centroid itself is a mean based on neighbouring points which is an approximation. The match is first established as a point-to-centroid to select the cluster or partition with the likelihood of a good match and then a point-to-point is used to establish the matched point with computational efficiency.

4.4 Limitations

The clusters created by k-means for KmeansICP can be unevenly distributed based on the shape of the point cloud. This can result in large differences in the number of distance calculations for each selected cluster based on the points in the cluster. The equal-sized partitions give the best distance calculations for the algorithm, meaning that unequal sized partitions with especially large numbers of points will result in many distance calculations rather than an even distribution of distance calculations across all partitions.

Results of this algorithm and a comparison to the other algorithms will be presented in Chapter 7.

5 CHAPTER 5 SUBSETICP+

5.1 Introduction

Another approach to reducing the size of the correspondence search space in an ICP iteration is by using corresponding subsets. For rigid shapes, equal sized corresponding subsets are created based on the unique atom types in the point cloud and the ICP operation registers corresponding subsets in sucession instead of the whole point cloud. The transformation computed from the registration of each subset is then applied to the whole point cloud.

This chapter discusses the SubsetICP+ algorithm. It provides a critical analysis of the design, computational complexity, and limitations. This chapter and chapters (CHAPTER 4 KMEANSICP and CHAPTER 6 TAGGEDICP) aim to achieve objective three (3) of this research, "*Develop shape matching algorithms based on the Standard ICP*".

SubsetICP+ is based on the idea of SubsetICP (Chen, Belaton and Pan, 2013). SubsetICP partitions the model point cloud and data point cloud into an equal number of corresponding subsets. SubsetICP requires that more than one corresponding subset (cardinality) exists otherwise the whole point cloud is considered as one set which will be the same as standard ICP with no benefit. The number of points in each subset must be of sufficient size to do a well-defined ICP. For each iteration, a subset in the data point cloud is registered to a corresponding subset in the model point cloud. The resulting transformation is applied to the data point cloud and the process repeats for all other subsequent subsets until the set maximum iteration is reached. As described above, SubsetICP (Chen, Belaton and Pan, 2013) partitions the point clouds into subsets. SubsetICP+ is simply a SubsetICP where the point clouds are partitioned by atom types.

Consider a dataset of molecules with atom names as metadata. Such a set will have subsets made up of the unique atom names, with each subset storing all atoms of the molecule of the same type (e.g., H, N, O, C and S). Partitioning the model cloud this way enables a faster search for correspondence since the first partitioning serves as a direct pointer to the location of the partition storing the correspondence being sought. Partitioning and caching in a pre-processing stage also ensures this task is not repeated too many times as it is computationally expensive.

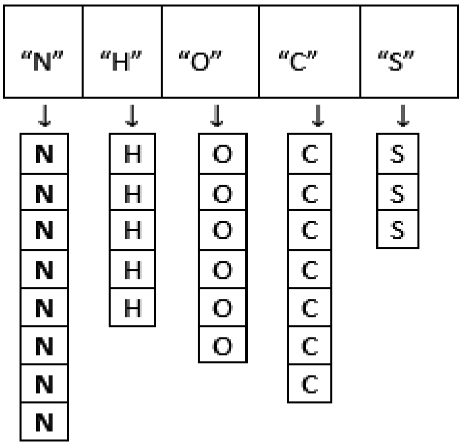


Figure 5.1 SubsetICP+ partitions the search space based on atom types. This technique allows the correspondence search method to only search within atoms of its type, further reducing the time taken to complete the search task.

Figure 5.1 shows the partitioning of SubsetICP+ with unique atom names serving as a pointer to the position of related partition of atoms. SubsetICP+ algorithm performs registration on one partition (e.g., "N" only or "H" only etc) in the data point cloud to its corresponding partition in the model cloud based on the metadata (atom type) knowledge available. TaggedICP' s partitioning is different because it makes use of neighbouring information as part of the correspondence search process whilst SubsetICP+ does not.

Since SubsetICP+ partitions the subsets by atom types, there is the possibility of having corresponding subsets of different sizes (number of atoms in a subset) because the number of atoms is given by their abundance in the 3-D protein dataset. To ensure that corresponding subsets have the same number of atoms during registration, SubsetICP+ could set the size of a subset pair to the size of the smallest corresponding subset. However, this would not be efficient for matching partial shapes and mutations where different counts of the atoms exist, and it would be more efficient to match all available corresponding atom types.

Chen, Belaton and Pan (2013) found out that, one of the limitations of SubsetICP was the unknown optimal cardinality (number of subsets in the point cloud). In the development of

SubsetICP+, it was found that shapes will have one or more optimal cardinalities at which to achieve the best match quality and convergence. The research explained a correlation between low MSE and small cardinality, with a higher cardinality contrasting this observation with a low MSE. This, therefore, shows that cardinality has varying effects on the MSE.

The algorithm's create partition method assumes a point cloud X, with unique atom types U. For each atom type in U, the algorithm resets an empy list L_m and copies all points of type m. The closest point search works in the same was the closest point search for KmeansICP+, by performing a naïve closest point search in the corresponding subset (subset in the model point cloud with same atom type)

5.2 Algorithm

The algorithm statement is

1. Partition the model point cloud into corresponding subsets according to atom types
2. Initialise Iteration to 0, the maximum number of iterations, set change in (R)MSE threshold, set cardinality, transformation (R, ***t***)
3. Apply the transformation (R, ***t***) to the data point cloud
4. Perform a standard ICP operation, registering corresponding subsets in sequence, one subset per iteration and returning a transformation that aligns that subset.
5. Update the registration transformation (Computational cost: (O (N_p)))
6. Apply the registration transformation to the data point cloud (Computational cost: O (N_p))
7. Terminate the iteration when the iteration count or RMSE threshold is reached.

```
function CREATEPARTITIONS (X)
    List of Unique atom types $\mathrm{U} \longleftarrow$ DISTINCTMETADATA (X)
    for all atom types $m$ in $\mathrm{U}$
        List $L_m \longleftarrow$ empty list
```

```
            for all points p in X with atom type m
                append p to $L_M$
    return List of $L_m$,
```

```
function CLOSEST POINT SEARCH ($P_k$ , $X_k$ )
    closest points $Y_k \leftarrow empty\ list$
                $t \leftarrow atom\ type\ of\ subset$
    for all points p in $P_k$ do
        closest point distance $d_{min} \leftarrow \infty$
        closest point y ⟵null

        for all points x in $X_k$ do
            distance d ⟵ DISTANCE (p, x)
            if d < $d_{min}$ then
                closest point ⟵ x
                closest point distance $d_{min} \leftarrow d$
        append closest point y to closest points $Y_k$
return $Y_k$
```

Full SubsetICP+ algorithms

```
Inputs: point cloud X with cardinality m and Y with cardinality n based on atom types on point cloud
Initialise $R_O \leftarrow I, T_o \leftarrow 0, i \leftarrow 1$
X is partitioned into k subsets, $\boldsymbol{X} = X_1 + X_2 + X_3 + \cdots + X_k$
Y is partitioned into k subsets, $\boldsymbol{Y} = Y_1 + Y_2 + Y_3 + \cdots + Y_k$
While i<k and MSE (X, Y)> threshold
    For all points in subset $X_i$ $and$ $Y_i$
        Calculate the correspondences of all points in the subsets
        Compute the transformation $R_i$ $and$ $T_i$
            Then Apply $R_i$ $and$ $T_i$ to point cloud X
        End ICP Iteration
    End Subset Iteration
```

5.3 Computational Complexity

SubsetICP+ partitions the data point cloud and model point cloud into subsets based on the atom types as shown in Figure 5.1. Due to the unequal sizes of the subsets when partitioning by the atom type, the algorithm can choose the cardinality of the subsets for the iteration in two ways; first (1) is to set a global fixed cardinality parameter and then filter the subsets based on a criterion such as spatial distances to a point and second (2) is to dynamically set the subset size to the lowest subset between the corresponding subsets. SubsetICP+ does not implement any of these 2 methods but rather allows the corresponding subsets to retain the sizes based on the abundance of the atom types in the molecule.

Assume a SubsetICP+ with subsets X and Y and cardinality (size) of the subsets N_1 , and N_2 respectively. The complexity of the computation (subsets are registered in sequence, one per iteration instead of the whole point cloud per iteration) will be.

$\mathrm{O}\ (TN_1N_2)$ for each iteration, where $\mathrm{T} = \frac{N_x}{N_1} = \frac{N_y}{N_2}$, and $N_1 \leq N_x, N_2 \leq N_y$ (N_1 and N_2 are the cardinals of subsets X and Y respectively) representing a distance calculation over a fraction of the point clouds based on the subsets.

SubsetICP+ has the same computational complexity as SubsetICP because it reduces the the number of distance calculations to a factor depending on the partitioning: ${N_xN_y}/{(N_1N_2)}$.

SubsetICP+ uses a point-to-point objective function to minimise the error between the two subsets. The minimisation is done through an iterative process of computing the least squares quarternion rotation and a translation vector from the subsets that is applied to the whole data point cloud.

Unlike KmeansICP, SubsetICP+ ensures that the cardinality of the partitions is the same. However, like KmeansICP, the number of distance calculations in SubsetICP+ is reduced by a factor. This factor depends on the number of partitions. A larger number of partitions means smaller distance calculations.

The advantages of SubsetICP+ are that finding the closest points in the subset provides a better estimate of the transformation than using the whole point cloud. Shape information is also preserved because each subset is matched in an iteration, sequentially.

5.4 Limitations

The limitations of Standard ICP applies to SubsetICP+. Even though the search space is reduced to subsets to iteratively match corresponding parts of the point cloud. It is not known how the algorithm will perform with different domain datasets.

Since the optimal cardinality is not known, the algorithm may not always perform as efficiently as it could and determining the optimal cardinality as a pre-processing step can increase the overall time taken to run a match.

Results of this algorithm and a comparison to the other algorithms will be presented in Chapter 7.

6 CHAPTER 6 TAGGEDICP

6.1 Introduction

The research explored another approach to reducing the size of the corresponding search space by making use of known information about the point cloud. This adds a constraint to finding the closest point search and helps make the matching meaningful. Particularly for this research, the concept of

Tagged-ICP involves finding the closest point in the point cloud with the same atom type and 3-nearest neighbours sorted alphabetically. This constraint on the search space reduces the search items to points with the same atom type (metadata tag) and allows probable matches to be searched and found.

This chapter discusses the TaggedICP algorithm. It provides a critical analysis of the design, computational complexity, and limitations. This chapter and chapters (CHAPTER 4 KMEANSICP and CHAPTER 5 SUBSETICP+) aim to achieve objective three (3) of this research, "*Develop shape matching algorithms based on the Standard ICP*".

TaggedICP is based on ICP (Besl and McKay, 1992). It uses metadata information of the points, referred here as "tags" to aid in the matching. The tag information is made up of the atom type, and the 3 nearest neighbours with their atom types ordered alphabetically.

When matching molecules, TaggedICP searches for correspondence by considering a point in the model point cloud with the same 'tag' (atom type and 3 nearest neighbours ordered alphabetically by atom) as the point in the data point cloud, ignoring all other points with different tags. This strategy effectively reduces the correspondence search space by a fraction of the partition count or unique tags available in the dataset. Consider the partitions illustrated in Table 7.1, 2M5T is made up of 5 partitions, though unequal, they all represent smaller subsets or clusters with a unique atom type. This means that TaggedICP reduces the time spent searching for correspondence because the search space is reduced to only atoms of the same type. A similar representation of this partitioning is also illustrated in SubsetICP+ partitions Figure 5.1.

TaggedICP also makes use of neighbourhood information in the correspondence search method by considering a match between the specified number of neighbours when all their metadata under consideration are also a match.

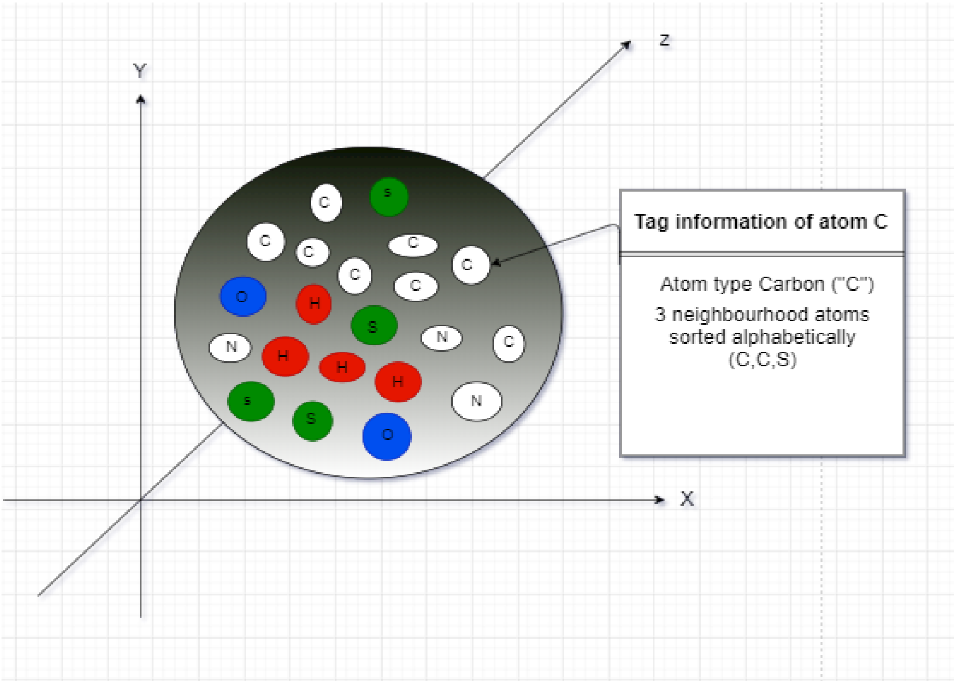
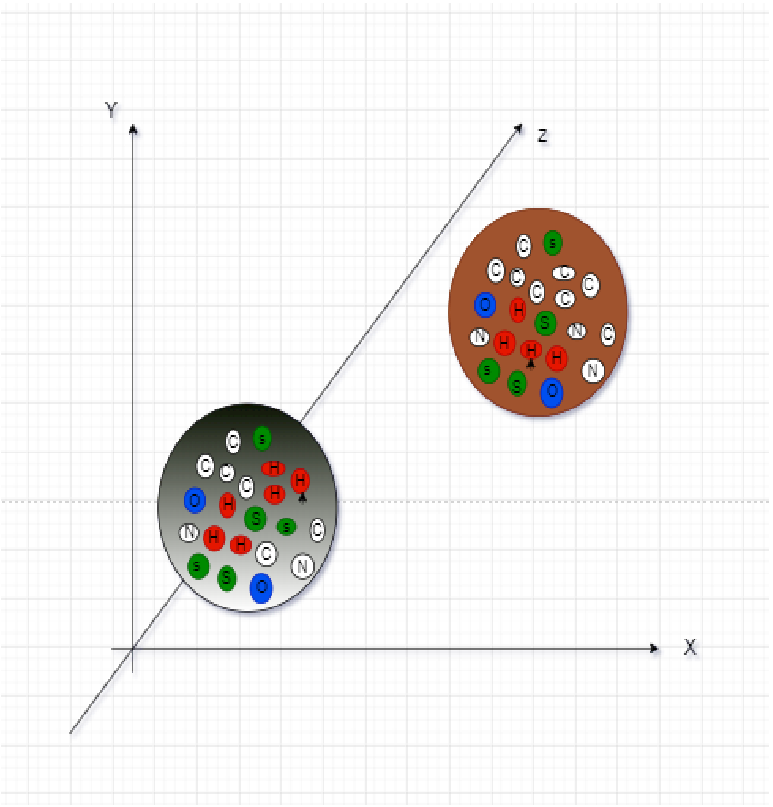

Y
Z
X
Tag information of atom C
Atom type Carbon ("C")
3 neighbourhood atoms sorted alphabetically (C,C,S)
C
S
C
C
C
C
C
C
O
H
S
N
N
C
H
H
H
S
S
O
N

Figure 6.1 Description of colour-coded atoms in a molecule (grey sphere) on an x-y plane. The tag information used in the TaggedICP algorithm is explained for atom C (with an arrow pointing to).

Matching shapes by considering the tag information of a specified number of neighbours further enhances the algorithm by making the matching more meaningful and further reducing the search space. A point in the data shape is considered the closest point of another in the model shape when the tag information of the point matches. This includes a match for the atom types for the specified closest neighbours of the data point in alphabetical order of atom type.

The tag information illustration (Figure 6.1) shows a sample molecule represented as a point cloud of atoms and their tag information made up of the atom type (S=Sulphur, O=Oxygen, N=Nitrogen, C=Carbon) and the 3 closest neighbours of each atom type sorted alphabetically (C, C, S).



Tag information of H (with arrowhead) in Grey molecule
Atom type: H 3 Nearest neighbours sorted alphabetically: H, H, S Atom's XYZ coordinates

Tag information of matching H (with arrowhead) in Brown molecule
Atom type: H 3 Nearest neighbours sorted alphabetically: H, H, S Atom's XYZ coordinates

Figure 6.2 Two molecules representing point clouds, grey and brown molecule, showing their tag information for atoms (with arrowhead). The tag information consists of the atom type, atom types of three (3) nearest neighbours and the atoms relative position within the point cloud

Assuming we want to find the correspondence for atom H in the data shape (tag H in red with arrowhead) in Figure 6.2, in the model point cloud (brown atom). The algorithm first finds the closest 3 neighbours of atom H in the data point cloud and then finds all atoms in the model shape with atom type, H, that have the closest 3 neighbours with the same atom types as the atom types of the 3 nearest neighbours of atom H in the data point cloud ordered alphabetically. If there is more than one atom found as the closest neighbour, the correspondence search returns the model point closest to the data point. If no match is found, the correspondence search returns the closest atom in the search partition (which will be the closest atom of the same type because the search is within the atom type of the data point).

In Figure 6.2, the grey molecule represents the data shape with atom name H (arrowhead) whilst the brown molecule represents a matching atom with name H (with arrowhead) based on neighbourhood information. The atom with the arrowhead (H) in the Grey molecule has 3 closest neighbours (H, H, S) which matches the atom in the brown molecule (H, H, S) based on the tag information of the closest 3 neighbouring atoms.

6.2 Algorithm

The algorithm is similar to the standard ICP algorithm (Besl and McKay, 1992) in Section 2.6.1. However, TaggedICP reduces the correspondence search space by partitioning the model point cloud by atom types. The algorithm searches for correspondence by matching the atom type and the atom types of the 3-nearest neighbouring atoms ordered alphabetically. The matched points must have the same atom types. Thus, for the points that do not have the same atom types, the algorithm returns the closest point in the same atom type partition. It is also possible for these points to have been excluded during the search by filtering, but this is not how the current version of the algorithm is implemented. TaggedICP uses a point-to-point objective function to minimise the error between the two point clouds.

The algorithm statement is.

Pre-processing: partition the model point cloud according to atom types e.g., H, C, S partitions

1. Initialise Iteration to 0, the maximum number of iterations, set change in MSE threshold, transformation (R, t)
2. Apply the transformation (R, t) to the data point cloud
   a. Compute the closest points in the model point cloud for each point in the data cloud by considering the matching of tag information (data point's atom type, its 3-nearest neighbours).
   b. If there is no match for a data point, return the closest atom in the search space (model point atom type partition) (Computational cost: $O\left(N_x N_p\right)$)
3. Compute/update the registration transformation (Computational cost: (O (N_p)))
4. Apply the registration to the data point cloud (Computational cost: O (N_p))
5. Terminate the iteration when the iteration count or RMSE threshold is reached.

The metadata of the point is used to speed up the search for correspondence and improve the registration accuracy. The model point cloud X has been partitioned by the tag and only the partition corresponding to the atom type of p is then further searched and disambiguated.

The specified number of neighbouring points are also searched to match their metadata to the neighbouring points in the data point cloud. If a match based on the tag information is not found, the closest point in the alphabetically sorted atom type partition is returned ignoring the neighbourhood information. TaggedICP might return non-optimal correspondences based on the closest atom with the same type in the partition when there is no match using the neighbour information. This closest point might not be ideal as it doesn't represent the exact match as would have been achieved using the neighbourhood information. This might present less than ideal results in such instances.

The pseudocode presented in Algorithm 2 modifies the original ICP algorithm's correspondence search method to include the use of metadata to represent TaggedICP' s closest point search method. The partitioning based on the atom type was also done in an initialisation stage before the registration process started. This partitioning could be different for specific metadata being used in an implementation of TaggedICP. For instance, registering a point cloud based on colour gradient could partition the point clouds based on a set boundary of RGB values. All other methods of TaggedICP implement ICP's methods directly.

Assume two point clouds P and X. The algorithm finds the three nearest neighbours of each point. Three closest neighbours, p_N for point ***p*** in P and three closest neighbours x_N for point ***x*** in X. The algorithm then does a naïve closest point search for the correspondence of point ***p*** within the points in X with same tag as ***p*** and matching three neighbours sorted alphabetically by atom type.

```
function CLOSEST POINT SEARCH ($P_k$ $X$)
    closest points $Y_k \leftarrow empty\ list$
    For each point p in P, find its three neighbours $p_N$ with sorted tags $t_N$
     For each point x in X, find its three neighbours $x_N$ sorted tags $t'_N$
    for each point p in $P_k$ do
        closest point distance $d_{min} \leftarrow \infty$
        closest point y ⟵null
        Find the tag $t_p$ of p
        for all points x in X with a tag of $t_p$ and the sorted tags $t'_N$ of $t_N$ being the same as $t_N$ do
            distance d ⟵ DISTANCE (p, x)
            if d < $d_{min}$ then
                closest point y ⟵ x
                closest point distance $d_{min} \leftarrow d$
        append closest point y to closest points $Y_k$
    return $Y_k$
```

Algorithm 2 TaggedICP. The point clouds are partitioned according to atom types and sorted alphabetically, and a standard ICP operation is performed whilst using the metadata tag information as a constraint in the correspondence search method.

6.3 Computational Complexity

Based on the standard ICP algorithm (Besl and McKay, 1992), the cost of the correspondence search increases with the increasing number of distance calculations. Given a data point cloud P with N_p points and a model point cloud X with N_x points, the computational complexity of a naive Closest Point Search algorithm is $O(N_P N_x)$. However, it reduces the number of distance calculations by a factor depending on the partitioning. The reduction in the number of distance calculations, reduces the cost of the correspondence search. The objective function uses a point-to-point error metric to minimise the distance between the two point clouds. The minimisation is done through an iterative process of computing the least squares quaternion rotation and a translation vector that is applied to the data point cloud.

Assume a data point cloud with a number N_p of points and a model point cloud with a number N_x of points partitioned by tag, t. The computational cost of the correspondence search is reduced to O ($N_{pt} N_{xt}$).

Given fractional partition sizes $f_{p,t}$ and $f_{x,t}$ for each point cloud, P and X and each tag t such that $\sum_t f_{p,t}$= 1 and $\sum_t f_{x,t}$ =1, the number of distance calculations is reduced to a factor F = $\sum_t f_{p,t}\ f_{x,t}$ ≤ 1 because the distance calculations are within a fraction of the whole point cloud. The distance calculations are reduced because the correspondence search space is within a fraction of the point clouds defined by the number of distinct tags. Factor F is 1 when matching the whole shape.

The factor F would be smallest for a larger number of equal-size partitions because the larger the number of partitions, the smaller the number of points per partition until the tipping point. Consider a point cloud with two equal-sized partitions (fractional sizes 0.5 each) and a second point cloud with 4 partitions (fractional sizes 0.3 for 3 partitions and 0.1 for 1 partition). The distance calculations are reduced to a factor F = 2 (0.5*0.5) = 0.50 when matching the first point cloud with 0.5 fractional partition sizes, whilst the distance calculations will be reduced to F=3(0.3*0.3) + (0.1*0.1) = 0.37 for the second partition when matching using one of the fractional sizes 0.3. Thus, the factor F is smaller (0.37) for larger numbers (3) of equal-sized partitions (second point cloud).

For instance, if 80% of the atoms in a molecule have the same tag information, that partition has a fractional size of 0.8. If the remaining 20% of atoms all have the same other tag, that partition has fractional size of 0.2. This means the distance calculations are reduced to a factor F= (0.8*0.8+0.2*0.2) =0.68.

Consider Table 7.1. When performing a full shape 2M5T matching, the fractional partition sizes of the atoms are C (0.32), N (0.086), O (0.1), S (0.0033), and H (0.48). if partition H is matched, the distance calculation is reduced to F= (0.32*0.32+0.086*0.086+0.1*0.1+0.0033*0.0033+0.48*0.48) =0.35 However, the fractional partition sizes in this research's dataset are given by the abundances of the atoms and their neighbourhoods. 96.5% of the atoms in an organic molecule are made up of only 4 types (Hydrogen (H), Carbon (C), Oxygen (O), and Nitrogen (N)) (Alberts et al., 2002). Roughly 60% of these atoms are hydrogen, 25% are oxygen, 10% are carbon, and the other elements each represent negligible fractions of a molecule (Audesirk, Byers and Audesirk, 2016). The distribution of atoms in a molecule is different in the case of inorganic molecules, for instance, a sample molecule from the earth's crust contains about 1% each of Hydrogen (H), Nitrogen (N), and carbon (C), 55% Oxygen (O), 28% Silicon (Si) and a sizeable fraction for Calcium (Ca), Magnesium (Mg), Sodium (Na), and other atoms (Alberts et al., 2002).

From the percentages of atoms in organic molecules, it can be deduced that if the tag information consisted only of the point's atom type and omitted the neighbourhood information, the computation time would only be reduced by a factor of F= 0.1*0.1+0.55*0.55+0.28*0.28=0.38 in practice, however, TaggedICP uses neighbourhood information when searching for correspondence and this further reduces the factor F. None of the implemented algorithm's computation time is reduced by a factor F=0.38 because the distribution of atoms in the molecular datasets used is not the same as the illustration by Alberts et al. (2002).

The partitions can be made increasingly smaller, however, if corresponding small partitions ignore true or exact matches, it reduces the quality of the correspondence. For instance, consider the same atom type partition. For full shape matching, there exists true correspondences for each data point and reducing the partition size further increases the probability of having fewer true matches because it eliminates some existing matches. However, evenly reducing the atom type partition size can decrease the computational time further if enough points are adhering to the standard ICP principles.

TaggedICP uses the neighbourhood information of atoms in the correspondence search by comparing the tag information of the three (3) nearest neighbouring atoms to both the model point and the data point. Number 3 was selected based on the match quality and convergence tests done in the TaggedICP convergence experiment (Figure 7.5) and the TaggedICP match quality experiment (Figure 7.6). This additional process makes the correspondence search meaningful by increasing the probability that correspondence is a true match.

The cost of the correspondence search in TaggedICP increases with the increasing number of distance calculations inside the individual shape, but not between the shapes. The number of distance calculations is increased with the increasing number of neighbourhood atoms being considered.

6.4 Limitations

Even though TaggedICP uses partitions to reduce the correspondence search space, the use of the k-nearest neighbour constraint increases the time taken for the algorithm to complete the closest point search. This can be mitigated with the help of using k-means clusters to find the k-nearest neighbours when considering the atom as the centroid.

More than one (1) atom cannot be at the same position, however, when more than one neighbouring atom is at the same distance from the reference point but are of a different atom type than what is sought, the correspondence does not return the data point as a match because there is no match between the atom types of the 3-closest neighbours. Instead, the correspondence returns the closest atom in the search space without considering its atom type. In rigid shape matching, such a situation cannot happen in full shape matching, but rather in partial shape matching because full shape matching inherently has true matches for all points.

6.5 Summary

The chapters (CHAPTER 4 KMEANSICP, CHAPTER 5 SUBSETICP+, and CHAPTER 6 TAGGEDICP) have discussed the ICP algorithm variants, their computational complexity, and their limitations. This critical discussion achieves objective three (3), "D*evelop shape matching algorithms based on the Standard ICP".*

All the algorithms described above implement some sort of partitioning. KmeansICP partitions the data point cloud by considering only the spatial distance between the points. This partitioning is useful when searching for neighbourhood points because it shows points closer to others.

The KmeansICP clusters can be used to efficiently limit the search space when neighbourhood points are being considered and searched, however, it is not as effective as the partitions in TaggedICP because the latter partitions are according to known information that is directly used in the correspondence search method. TaggedICP uses the metadata, called tag to reduce the search space and make the search meaningful whilst KmeansICP uses the k-means cluster to reduce the search space. SubsetICP+ also partitions its search space by atom types so the partitions supplied for a closest point search operation are of the same atom type.

The algorithms also differ in how their partitioning is implemented. SubsetICP+ partitions the search space by atom type but using equal-sized partitions for the iterations, TaggedICP partitions by atom type and can match using unequal sized partitions. TaggedICP also uses neighbourhood information to aid in the matching. KmeansICP partitions the search space using K-means clusters. These clusters are approximately equal based on the complexity of the shape (in terms of how evenly distributed the atoms occur in the point cloud).

Partitioning by known information or tag (metadata), as compared to using spatial distance, enables the algorithm to reduce the search space and make the correspondence meaningful by allowing the search to find correspondence from a partition with a higher probability of finding a meaningful match. This partitioning can result in meaningful matches because the search space will only contain points with the same metadata. k-means clusters can be used to further partition the search space in TaggedICP and SubsetICP+ so that in the search space of points with known information, the search method will also use the Euclidean distance function to search in a closer sub-partition to the data point. This will reduce the search even further. Particularly for TaggedICP, k-means can be used to cluster each atom type partition in the model point cloud and then the search for correspondence will search within the cluster (sub-partition) having the closest centroid to the data point. Some of KmeansICP' s disadvantages, such as points farther away from the centroid having less reliable matches, will still apply to this technique, however, it would still reduce the search space from the atom type partition to several clusters in that atom type partition.

Criteria	ICP	TaggedICP	SubsetICP+	KmeansICP
Use of metadata	No	Tag information, (atom type and neighbourhood information) into unequal partitions	atom types into unequal partitions	No
Partitioning Technique	No	Partitioning by atom type. Different size partitions are allowed.	Partitioning by atom type. Points in corresponding subsets are not the same, especially for partial shape matching.	Partitioning by k-means clusters

Transformation filtering	No	No	No	No
Iteration criteria	Iteration registers whole point cloud	Iteration registers whole point cloud	Iteration registers a subset	Iteration registers whole point cloud

Table 6.1 Summary of algorithms. Details of the algorithms with some criteria (use of metadata, partitioning technique, transformation filtering, and iteration criteria) describing their implementation.

Table 6.1 gives a tabular comparison of the algorithms. This allows the reader to understand the algorithms designed. It uses criteria such as “*use of metadata*” and “partitioning technique” to provide an overview of the discussed algorithms.

Part 4: EXPERIMENT RESULTS AND CONCLUSION

7 CHAPTER 7 ICP ALGORITHM EVALUATIONS & ANALYSIS

7.1 Introduction

Three experiments were designed to measure the quality or accuracy of the match in terms of the euclidean distances between final matching points and number of iterations to convergence. The experiments used different levels of simulated noise (0.001, 0.003, 0.005, 0.01, 0.03, 0.05) and initial rotation angles (15, 30, 60, 90, 120, and 150 degrees) to mimic complexities in datasets for a shape match. match the 3D protein structures to evaluate the performance of each algorithm in the presence of the mentioned factors. The datasets used were 2M5T, 2HAX, 2JKG and 2JKF from Protein Data Bank (RCSB Protein Data, 2020). Dataset 2HAX and 2M5T were used for the rotation and noise experiments and 2JKF and 2JKG were used for the noise only experiments. Each dataset (Protein Data Bank file) consists of a flat file with information such as the 3D coordinates of the atoms (points). This file was then loaded in Unity3D to create the 3D representation of the protein shape.

For each experiment, two datasets were matched using a specified criteria (levels of noise and rotation angles). The convergence data was generated after each iteration. This allowed an analysis of the error after each iteration. The data for the quality of the match is obtained after the matching is fully complete. This allowed an analysis of the accuracy of the algorithm because it measures the euclidean distances between each final matching point. Smaller distances between points represents a better match than larger distances between final matching points. Similarly, larger counts of iterations to convergence represents slower algorithms in terms of iterations but not in terms of time in seconds.

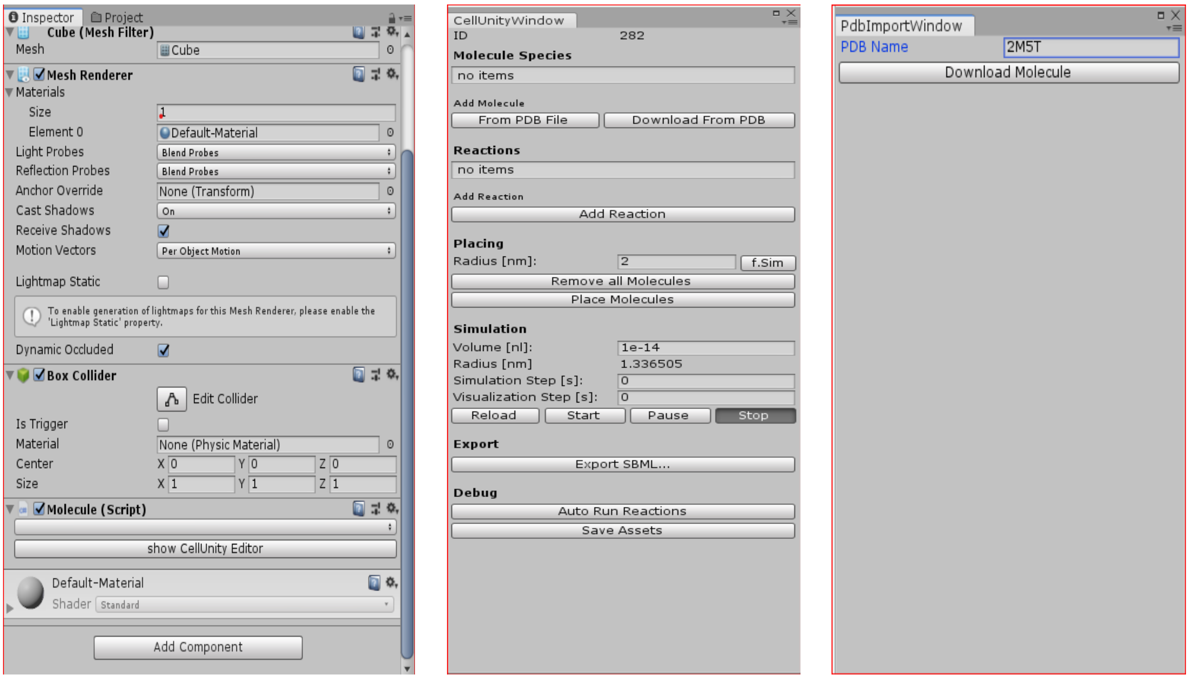
The match quality data are presented as a series of cumulative histograms that represents the quality of the match for each algorithm. Graphs with a sharp vertical rise indicates a better quality of match than graphs with a slower vertical rise across many units on the horizontal axis. The convergence graphs represent the error for each algorithm at every iteration.

The two graphs compare the iteration count at convergence and the distance between final matched shapes in each experiment.

This chapter discusses the experiments performed to evaluate the performance of the ICP variants. It details the setup of the environment, the performance (convergence and match quality) data gathered during the evaluation, and the interpretation and conclusions drawn from the results. This chapter, together with chapters (CHAPTER 4 KMEANSICP, CHAPTER 5 SUBSETICP+, and CHAPTER 6 TAGGEDICP) helps to achieve the overall aim of the research.

7.2 Resources

The ICP algorithms were developed using Microsoft C# (Microsoft, 2020) and the Visual Studio 2019 tool. The graphs were created using Python (Van Rossum and Drake, 2009). The 3-D protein structures were imported from the Protein Data Bank (PBD) file and the desktop visualisations were created with Unity3D (Unity Technologies, 2019). These tools were used to develop the algorithm, visualise the animation, capture the images of the registered shapes, and inspect the shape matching result, and generate the graphs.



Mesh: Cube
Size: 1
Element 0: Default-Material
Light Probes: Blend Probes
Reflection Probes: Blend Probes
Anchor Override: None (Transform)
Cast Shadows: On
Receive Shadows: ☑
Motion Vectors: Per Object Motion
Lightmap Static: □

To enable generation of lightmaps for this Mesh Renderer, please enable the 'Lightmap Static' property.

Dynamic Occluded: ☑
Is Trigger: □
Material: None (Physic Material)
Center: X 0 Y 0 Z 0
Size: X 1 Y 1 Z 1
Default-Material
Shader: Standard

ID 282
Molecule Species: no items
Reactions: no items
Radius [nm]: 2
Volume [nl]: 1e-14
Radius [nm] 1.336505
Simulation Step [s]: 0
Visualization Step [s]: 0

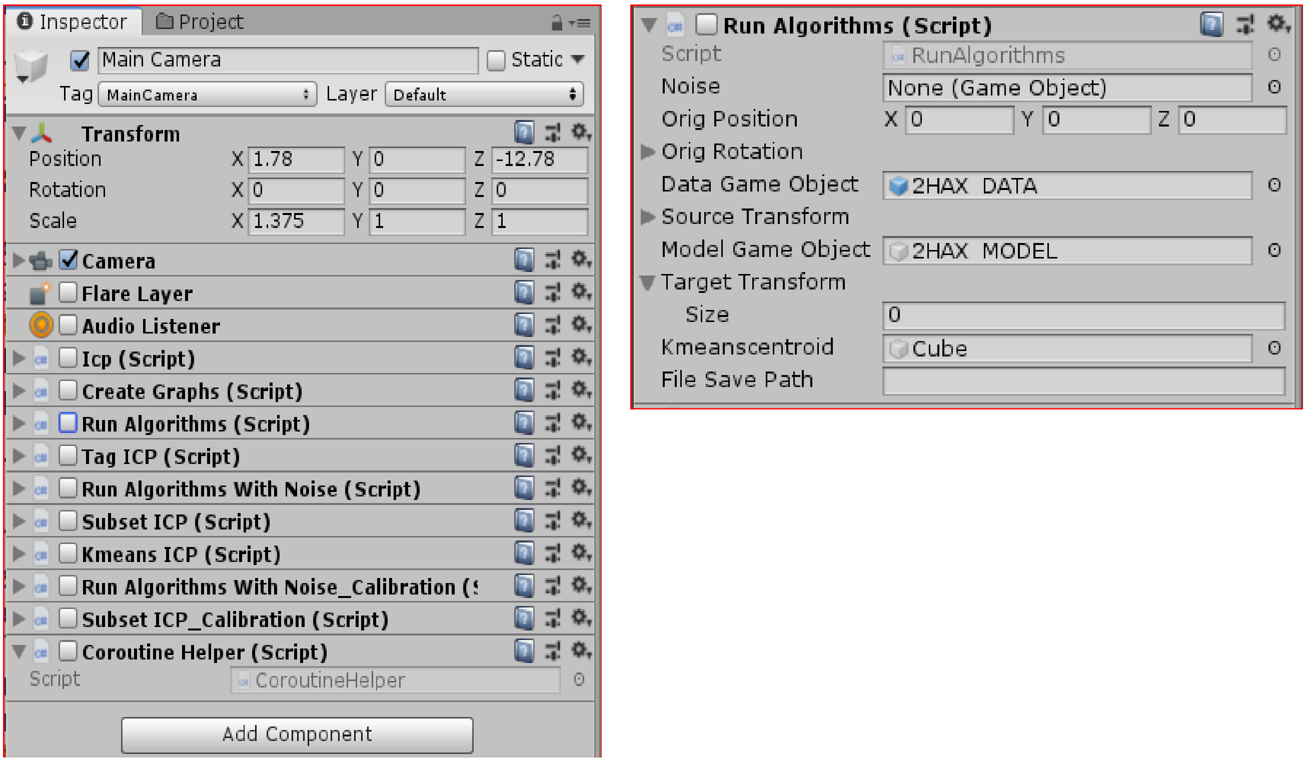
PDB Name: 2M5T

Figure 7.1 Cell Unity (Gehrer, 2015). A unity package for converting Protein Data Bank flat files into 3-D molecule objects. Cell Unity package is added to the Unity project which exposes its PDB Import functionality. (LHS) Inspector view shows the cell unity application for converting PDB files to 3-D structures. (M) Cell Unity Window provides an option for selecting a PDB file from disk or Download a PDB file (from rcsb.org). (RHS) A sample PDB code, 2M5T entered the PDB import window to be download from rcsb.org.

As shown in the Cell Unity package inspector windows (Figure 7.1), to generate the 3-D structure of each protein dataset, the PDB file (Berman et al., 2000) identification was entered into Cell Unity (Gehrer, 2015) and the PDB flat-file was retrieved and automatically generated a 3-D rigid object representation of the protein molecule. The PDB file is a flat file containing information on the 3-D structure of protein molecules such as atom types, sequence information (A group of atoms with a trend in their positions that can repeat in other parts of the molecule), and spatial coordinates of the atoms. This information can be extracted and used to create its 3-D protein structure on a coordinate system in Unity3D with the atoms adhering to the van der Waals radius (Bondi, 1964). The atoms were represented as spheres as required in this research although it can represent the atoms as any primitive type or custom 3D object shape.

7.3 Design

The atoms were represented as spheres with different colours showing their atom types, however, for purposes of showing overlap of points during a visual inspection of the final matches, red colour was used for all atoms in the model point cloud (molecule representing the model point cloud) and blue colour was used for all atoms in the data point cloud (molecule representing the data point cloud). The atoms were also scaled down to 0.034f on all axes to be able to see the overlap of the atoms. The spheres can show the 3-D position of the atoms in the protein molecule. The whole 3-D object represents the molecule whilst the points represent the atoms in the molecule. The generated data and model 3-D shape objects were then assigned to data and model point cloud placeholders in the script as intended for the matching in the Unity application as shown in Figure 7.2.



Inspector | Project

☑ Main Camera □ Static

Tag MainCamera Layer Default

Transform

Position	X	1.78	Y	0	Z	-12.78
Rotation	X	0	Y	0	Z	0
Scale	X	1.375	Y	1	Z	1

- ☑ Camera
- □ Flare Layer
- □ Audio Listener
- □ Icp (Script)
- □ Create Graphs (Script)
- □ Run Algorithms (Script)
- □ Tag ICP (Script)
- □ Run Algorithms With Noise (Script)
- □ Subset ICP (Script)
- □ Kmeans ICP (Script)
- □ Run Algorithms With Noise_Calibration (S
- □ Subset ICP_Calibration (Script)
- □ Coroutine Helper (Script)

Script CoroutineHelper

Add Component

□ Run Algorithms (Script)

Script	RunAlgorithms
Noise	None (Game Object)
Orig Position	X 0 Y 0 Z 0
Orig Rotation	
Data Game Object	2HAX_DATA
Source Transform	
Model Game Object	2HAX_MODEL
Target Transform	
Size	0
Kmeanscentroid	Cube
File Save Path	

Figure 7.2 (LHS) Unity project Inspector with the ICP algorithms added as components (Scripts) and [RHS] details of "Run Algorithms" script showing the model and data point clouds set into their respective placeholders (Data Game Object and Model Game Object).

The Unity project inspector window (Figure 7.2) shows the Unity3D project inspector showing some C# scripts used in the project (LHS). It also shows the Run Algorithms C# script with placeholders for the model and data game object representing the data and model point cloud 3-D protein molecule.

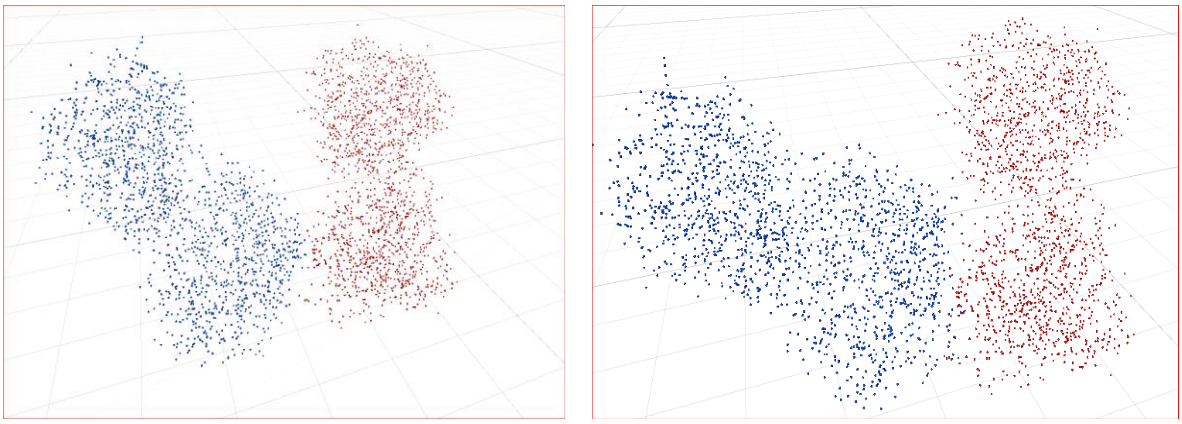


Figure 7.3 2HAX in Unity3D ready for registration with itself (full shape) at (LHS) 60 degrees and (RHS) 90 degrees on the x-axis. Blue represents the data shape and red represents the model shape at the origin. Each point in the point cloud is scaled down to 0.034f on all axes.

2HAX point cloud representation in Unity3D (Figure 7.3) shows 2HAX imported into Unity3D and each point scaled down to 0.034f on all axes. This was done to allow the overlap of the points to be easily observed as scaling down the points makes them sparser and provides some spaces between the points for easier visualisation of the extent of overlap.

Four (4) experiments were designed to.

1. Find the optimal neighbour count for TaggedICP by matching two identical molecules and taking 1-5 neighbours into account in the tag information. (Section 7.5)
2. Test how the initial rotation and partition sizes affect the matching of two identical molecules. (Section 7.6.1)
3. Simulate matching with measurement noise in the 3-D positions of atoms in two otherwise identical molecules and test how the initial rotation affects the matching. (Section 7.6.2)
4. Match two different mutations of the same molecule to compare whether the rotation influences the performance based on the quality of the match from the developed variants. (Section 7.6.3)

The dataset consisted of four different protein structures with PDB (RCSB Protein Data, 2020) identifiers 2HAX, 2M5T, 2JKF, and 2JKG. 2HAX and 2M5T were used for performing full shape matching (2) and noise simulation tests (3), and the TaggedICP neighbourhood tests (1). 2JKF and 2JKG were used in the mutation tests (4). The full shape matching experiment was to evaluate the algorithm's performance with datasets where the data point cloud is the same as the model point cloud. For all tests, the model data shape was positioned at the origin and the data shape was rotated on the x-axis for the indicated angle (15, 30, 60, 90, 120, 150) in degrees of the test.

The research chose protein molecules because of the challenging task of visualising 3-D protein shapes. Many tasks that help save lifes are dependent on protein shape matching such as drug discovery. The research could also help understand how the distribution of atom types affects the shape matching results. For instance, do many partitions improve the matching or otherwise, and how do the different partitioning techniques affect the matching results.

Large initial rotation angles are known to cause difficulties for ICP algorithms (Besl and McKay, 1992). Noise added to the positions of the atoms simulates inaccuracies in protein measurement methods and devices. Mutations are also a common use case for matching the identical functional parts of the 3-D protein structure and highlighting the differences caused by the mutation.

Noise simulation tests were used to evaluate the performance of the algorithms in the presence of different levels of noise. The experiment also studied how the different partitions, based on the distribution of atom types, affect the performance of different algorithms in the presence of noise, in terms of match quality and convergence.

The protein data bank (RCSB Protein Data, 2020) consists of over 167,518 protein structures acquired through various means such as using X-ray crystography and NMR (Nuclear Magnetic Resonance). Since the research's focus is on evaluating the performance of various ICP-based algorithms on matching the 3-D protein structures, it is only concerned with protein matching from the 3-D structural perspective and not with the intricacies of protein sequence matching. Protein sequence matching involves finding the similarity between protein molecules based on the similarity of sequences of atoms (similarly placed positional atoms).

Atom	2HAX	2JKF	2JKG	2M5T
C	848 (52.54)	845(60.14)	860 (59.23)	685 (32.33)
N	213 (13.2)	214 (15.07)	213 (14.67)	183 (8.64)
O	540 (33.46)	359 (25.28)	376 (25.89)	216 (10.1)
P	10 (0.62)	0	0	0
S	3 (0.19)	2 (0.14)	3 (0.20)	7 (0.33)
H	0	0	0	1028 (48.61)
Total	1614	1420	1452	2119

Table 7.1 This table shows the distribution of atoms in PDB 2M5T, 2HAX, 2JKF, and 2JKG. The count and percentages of each atom in the molecule are shown. Atoms have different partition sizes based on their atom types and the algorithms show different performance with the different atoms.

The distribution of atoms in the research dataset (Table 7.1) shows the distribution of atoms and percentages (in parenthesis) in the protein molecules used for the tests. 2HAX contains 1614 atoms, 2JKF contains 1420, 2JKG has 1452 atoms, and 2M5T contains 2119 atoms. Because they each have a different distribution of atoms in their molecule, it is important to show this distribution as it affects the matching of TaggedICP and SubsetICP+ and it also gives an insight into the discussion of the performance of the algorithms.

The Hydrogen(H) atoms are omitted in the PDB file for some of the molecules (2HAX, 2JKF, and 2JKG) because using X-rays for detecting the 3-D structure of the proteins has limitations on reliability and accuracy on detecting Hydrogen (H) atoms (Woińska et al., 2016). PDB (protein data bank) files created using X-ray crystallography usually do not contain the H atoms because the process is difficult to detect H atoms reliably and accurately, however, detecting protein structures with NMR (Nuclear Magnetic Resonance) spectroscopy can detect H atoms more easily. A mixture of molecules having different counts of unique atoms allows the experiment to find the effect of the number and size of partitions on different algorithms and present rigorously.

TaggedICP and SubsetICP+ partition their search space based on atom types. SubsetICP+ partitions the point cloud based on atom types in each subset called the cardinality. The partitioning allows an unequal sized partition of atoms to be registered iteratively.

On the other hand, TaggedICP maintains a unique partition size for the registration which is based solely on the count of the partition's atom type. This can potentially affect the quality of the match and whether the algorithm encounters and recovers from large errors. The results section provides a detailed discussion of the match quality and convergence of the algorithms and an interpretation of how the different aspects of the implementation of the algorithms such as partition size contribute to the results.

The tests will answer, amongst other questions, how the distribution of atoms in the molecule affects aspects of the shape matching. For instance, 2M5T has the highest total number of atoms in our dataset with a Hydrogen atom count at 1028 which is 48.61% of the total number and 2HAX having 848 carbon atoms representing 52.54% of the total atom number. The experiments can find the effect of the sizes of partitions (which is based on the abundance of the individual atom types) on the performance of the shape match.

The experiment also aimed to compare the performance of the algorithms; ICP (Besl and McKay, 1992), TaggedICP (Ankomah et al., 2020), KmeansICP, and SubsetICP+ in terms of their convergence and match quality on the 3-D molecule dataset. For each algorithm, the experiment aims to understand how the reduction in the search space affects the number of iterations at convergence. The use of known metadata and neighbourhood atom information improves the matching for molecules with different atom distributions as it reduces the search space for correspondence and how the distribution of these atoms affects the quality of the match. The correspondence of the points rather than the quantity increases the quality of the match as such reducing the search space by atom types especially in rigid full shape matching enhances the match quality.

The use of neighbourhood information is thought to make the matching meaningful and it will be interesting to understand whether the use of neighbourhood information results in slower convergence. For complex shapes, especially those with large numbers of atoms and non-uniform distribution of points, TaggedICP might return non-optimal correspondences based on the closest atom with the same type in the partition. This occurs when there is no match using the neighbour information. It is noted that TaggedICP partitions the search space according to the atom type to speed up the correspondence search. However, when there is no match based on tag information for a data point, the TaggedICP algorithm reverts to picking the closest point of the same atom type in the model point cloud partition as the correspondence. This closest point might not be ideal as it doesn't represent the exact match as would have been achieved using the neighbourhood information. This might present less than ideal results in such instances.

KmeansICP uses the K-means algorithm (MacQueen, 1967) to cluster the model point cloud, into a number k of clusters, based on each position's proximity to the mean of each cluster's points. To establish some correspondence, the closest cluster to the data point is found (point-to-centroid) and then the cluster is searched for the closest point (point-to-point). The cluster size can affect the performance of the algorithm because the larger the number of clusters present in the model point cloud, the fewer the number of points in each cluster and the faster the search for correspondence. Even though this can make the searches faster due to the fewer numbers of points in each cluster. However, a very large number of clusters cannot be used to get a faster correspondence search because if the number of clusters gets too large, determining the closest cluster starts to take too much time (this tipping point happens at $k=\sqrt{N}$), and the chance of being on the edge of a cluster and getting a worse correspondence increase.

Another technique to increase the probability of accurate matches using k-means is to assign weights to points based on the distance of the points to the centroid and then some points are ignored from the correspondence search in the cluster. However, the current version of the KmeansICP algorithm does not implement any such weight-based points filtering.

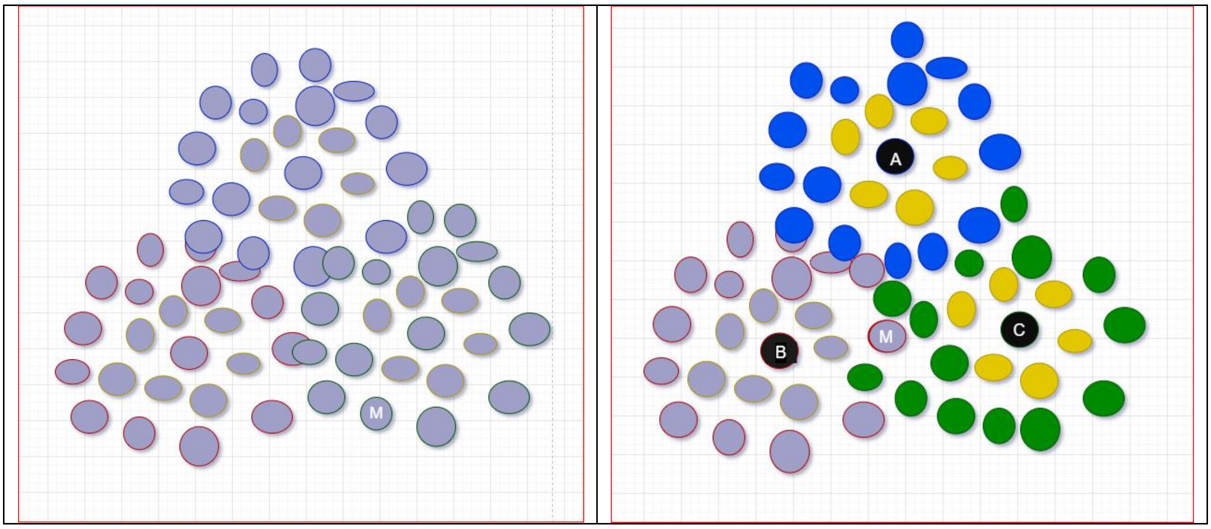


Figure 7.4 (LHS) Data point cloud with atom M (RHS) The correspondence for atom M in the data point cloud is M in the model point cloud. Atom M found its closest correspondence cluster (point-to-centroid to cluster B) and then found the closest point M but it can be observed that atom M in cluster B is closest to cluster C though it belongs to cluster B. This is because k-means uses the mean of the points closest to the centroid which is an approximation.

7.4 Analysis Graphs

The standard ICP algorithm was chosen as the base algorithm because it is simple and modular and allows the measurement of the improvements in these algorithms. The success of these improvements means they can be incorporated into other variants for further experiments on 3-D proteins. Using other variants of ICP such as triangulation-based iterative-closest-point for protein surface alignment (Ellingson and Zhang, 2012) and fractional ICP (Phillips, Liu and Tomasi, 2007) with matching vector pairs in Iterative Protein Alignment (IPA) (Kirys et al., 2007) instead of the standard ICP would have meant that their improvements will also affect the convergence and the match quality of the measurements in this research.

To compare and analyse the performance of the matching of standard ICP with TaggedICP, KmeansICP, and SubsetICP+, the thesis shows two types of graphs: Convergence and Match quality graphs.

The convergence graphs represent the mean squared error of the alignment of the two shapes at the end of each iteration (Figure 7.5). It is noted that the ICP algorithm iteratively finds the transformation that aligns the data shape to the closest points for each iteration so getting the error at the end of each iteration allows a good analysis of the matching process.

Assume the data point set P, with N_p points and the model shape X, with N_x points. The closest point set is computed as $Y_k = C(\, P_k,\ X)$. The mean squared error of this correspondence, e_k is computed as below.

$$e_k = \frac{1}{N_p}\sum_{i=1}^{N_p} ||Y_{ik} - \ P_{ik}||^2$$

The convergence graphs enable an understanding of which algorithms converged faster (iterated less to convergence) in terms of the iteration count at convergence and not the time in seconds. This is because the different algorithms implement different techniques, especially in the closest point search methods so an algorithm matching two shapes in a smaller number of iterations is distinguished from an algorithm that takes considerably less time and converging in many more iterations.

Figure 7.5 shows a convergence graph for two algorithms TaggedICP and ICP with three (3) different angles (30, 60, 90) showing a gradual descent from larger errors to lower errors as the iteration increases.

The convergence graph uses each algorithm’s own closest point search method to find the correspondences that are used for the error calculation. The graph shows the real MSE that would be used by the termination criterion. Unfortunately, some algorithms’ closest point search method imposes additional constraints based on their partitioning, leading to potentially more distant correspondences. This means it may not be fair to compare convergence curves for different algorithms. However, since these curves need to be calculated after each iteration, using a fairer comparison would slow the matching down too much. Each algorithm uses its implementation of the closest point search method, which means TaggedICP’ s implementation is not fair for direct comparison because it uses the most knowledge of the molecule (atom types and neighbourhood information) compared to the other algorithms (ICP, SubsetICP+, and KmeansICP).

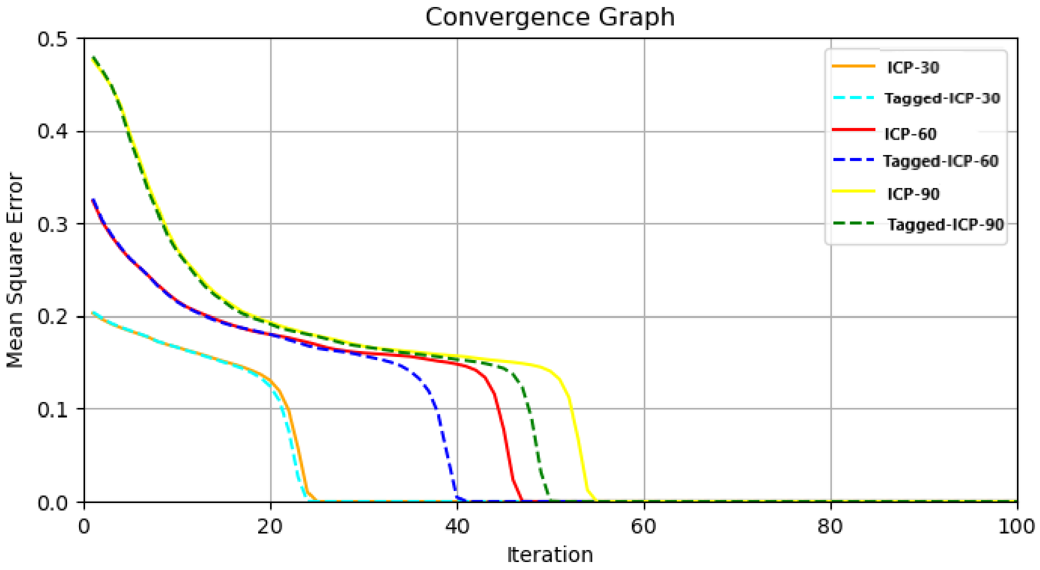


Figure 7.5 The convergence graph shows the number of iterations at which the algorithms converged. It is used in the tests to find out the algorithm that converged faster in terms of the number of iterations as well as the mean square error at convergence. This is important because the convergence characteristics are part of the measure of the performance of the algorithm. The Mean square error is an aggregate of the alignment error across all the points in the cloud. The sudden drop in error vrs iteration is because of the algorithm having a big drop in the error at after that iteration.

The sample convergence results for matching 2HAX (Figure 7.5) shows the convergence graph for ICP and TaggedICP at angle 30, 60, and 90. The graph shows the fastest convergence for both algorithms at angle 30, followed by TaggedICP at angle 60, ICP at 60 and then TaggedICP and ICP respectively at angle 90. It can be noticed that for all angles, TaggedICP converged faster (in terms of fewer iteration counts) than ICP. It can be observed that increasing the initial rotation angles results in higher starting mean squared errors and therefore a good initial transformation is required or algorithmic techniques to not require that for the ICP algorithm to be efficient.

The match quality graphs represent the cumulative percentage of points with their error (distance between final matched points) at or lower than some thresholds on the x-axis position when the shape matching is complete. The graphs were implemented with Matplotlib (Hunter, 2007) library package in Python (van Rossum, 1995). Matplotlib's implementation of a cumulative histogram also draws a vertical line from the 100% mark down to 0% at the x-axis position of the largest error (Figure 7.6) and this appears in some of the match quality graphs. Cumulative histograms cannot decrease like that, but these are correct cumulative histograms, and this is just an artefact of Matplotlib's implementation that the researcher decided to keep because visualisation of the largest error may be useful instead of cutting part of the graph. Since the analysis was interested in every aspect of the representation, the x-axis was set to an appropriate scale to capture relevant parts of the graphs. The larger scales capture the end of the algorithms' line and as such the vertical terminating line.

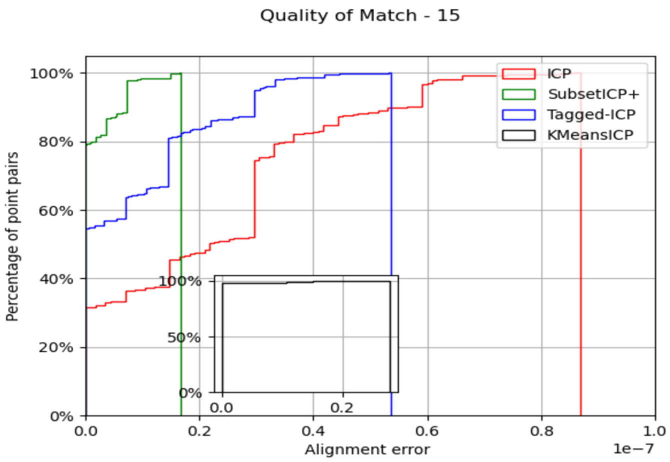


Figure 7.6 Match quality graph with initial rotation angle, 15 degrees. The graph is a cumulative histogram showing the alignment error against percentage of point pairs. It shows how many points have a given alignment error or lower alignment The historgram's terminating line at the largest error is from Python's matplotlib implementation and this was left in the graph to see the full extent of each graph. The cumulative histogram will always end each algorithm's graph at 100%, however this does not necessarily mean bigger errors because the percentage of point pairs is cumulative at a particular error so around 80% of point pairs for SubsetICP+ show a perfect match, however 20% (100-80) show an alignment error that is greater than 0.00.

The sample match quality results for matching 2HAX (Figure 7.6) shows the percentage of points against alignment error and this enables an understanding of the effective match quality based on what percentage of points have a certain amount of alignment error. It is commonly used to evaluate shape matching algorithms (Rusinkiewicz and Levoy, 2001, Du, Xu, et al., 2017). It shows a vertical line from each algorithm's largest error point, the 100% mark, down to 0%. This is due to a behaviour in matplotlib python library.

The alignment error represents the distance between final matching point pairs. A higher cumulative percentage of points (sharply rising graphs and graphs close to the top left) with a lower error implies a better quality of the match. Graphs show a better quality of match when the algorithm rises high up so quickly showing a steep rise. It represents the final positions of the point pairs and as such, it is an objective and more precise representation of the quality of the final matching shapes. The data for this graph is generated when the shape matching was complete by re-running the algorithms own closest point method once and then generating the distance (error) between corresponding matched points.

The two types of graphs (the match quality and the convergence) provide insights into the behaviours of different techniques: how fast and smooth the convergence graphs evolve from one iteration to another and how quick the rise in the match quality's cumulative histogram is used to determine the performance of the algorithm whilst analysing the effects of the partitioning technique on the performance.

7.5 Finding TaggedICP's Optimal Neighbour Count

TaggedICP considers several neighbouring atoms as a constraint in the closest point search to match their atom types for a meaningful match. The research investigated how to arrive at the optimal neighbour count by critically analysing the effect of small and large neighbour counts on the performance of the algorithms and an experiment was designed to find this number. The experiment considered 2HAX and 2M5T molecules because they were the molecules to be used in many of the tests to evaluate the ICP algorithms.

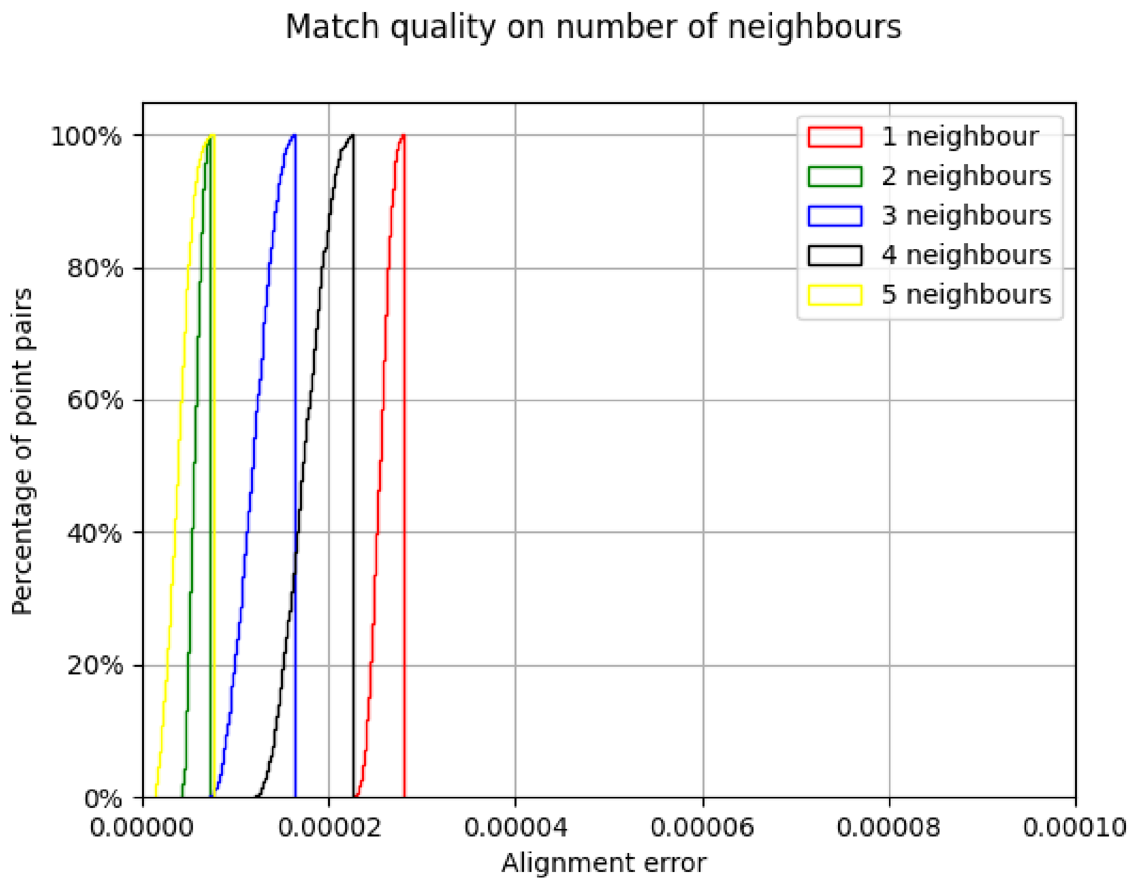


Figure 7.7 Match quality results for registering full shape, 2HAX at rotation angle 15 degrees using the TaggedICP algorithm with different numbers of neighbouring atoms. This is a test to find the optimal number of neighbour atoms when registering 2HAX, as a configuration to be used in the rest of the experiments for 2HAX. The graph shows that using 3 neighbours achieved the best quality of the match.

The optimal neighbour match quality results (Figure 7.7) shows the match quality experiment using 2HAX on 1 to 5 neighbours. It shows TaggedICP had a perfect match on all neighbour counts (1 to 5). This is realised in the scale of the alignment error (e-5).

The separation on the graph is due to floating-point errors. This trend is seen in Figure 7.8 where it is also noticed that TaggedICP converged at similar iteration on all neighbour counts. However, unlike Figure 7.7, Figure 7.8 shows some small differences in the convergence performance for the different neighbour counts. For instance, the neighbour counts seem to have converged at different iterations.

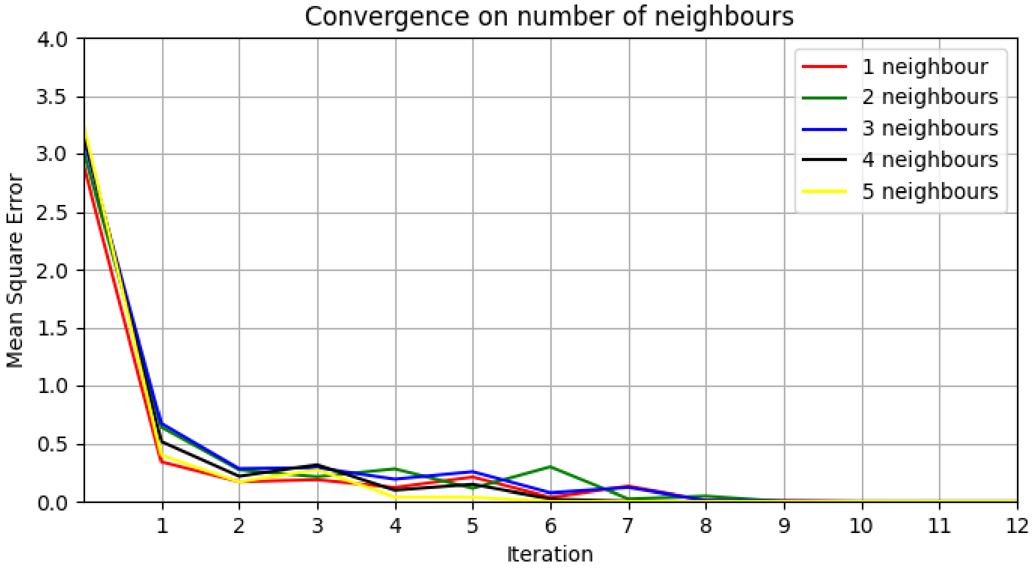


Figure 7.8 Convergence results for registering 2HAX at initial rotation angle 15 degrees on the x-axis using the TaggedICP algorithm with different numbers of neighbouring atoms. This is a test in conjunction with the match quality test above to decide the optimal neighbour of atoms when registering 2HAX. The graph shows that 1 to 5 neighbour(s) achieved a perfect match because all algorithms converged at 0 mean square error. However, the 5-neighbour test was noticed to have converged faster with the 1-neighbour test converging the slowest.

It can be noticed from the results that the convergence graph in the optimal neighbour convergence results (Figure 7.8) shows an inverse relationship with the iteration number at the convergence of the algorithm. The 1-neighbour count started with a small error but converged much later than 5-neighbour that started with a larger error but converged much earlier. In the earlier iterations, this trend looked the opposite of the convergence trend. Considering only Figure 7.8, it can be deduced that the higher the neighbour count, the faster the convergence.

It would be easy to assume that selecting a higher number of neighbour atoms is optimal, however, the match quality graph in Figure 7.7 shows that, for 2HAX, using 1 to 5 neighbours yielded perfect matches. The match quality graphs in Figure 7.7 shows tiny (e-5) alignment errors compared to the distances between atoms in a molecule. This difference in the alignment errors is a coincidence due to floating-point rounding errors.

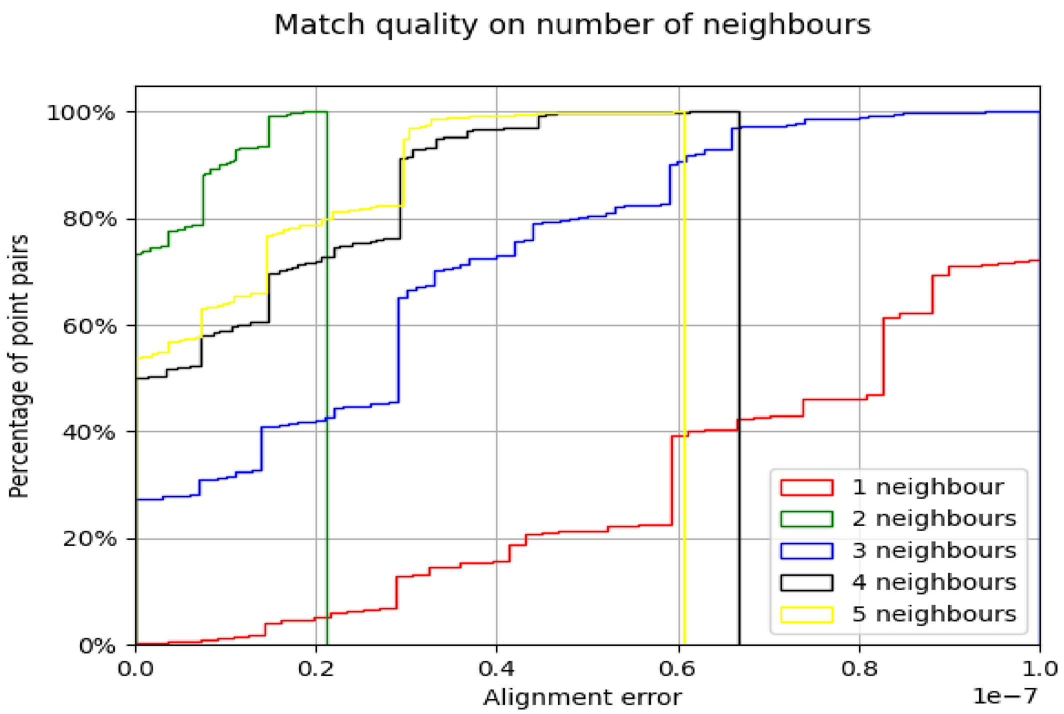


Figure 7.9 Match quality results for registering full shape, 2M5T at an initial angle 15 degrees on its x-axis using the TaggedICP algorithm with different numbers of neighbouring atoms. This is a test to find the optimal number of neighbour atoms when registering 2M5T. The graph shows that using 4-neighbours achieved the best quality of match as the 4-neighbour graph shows more percentage of point pairs at lower alignment errors than the other neighbours.

The optimal neighbour test for registering 2M5T (Figure 7.9) shows match quality for matching 2M5T using 1 to 5-neighbours. It is observed that 2-neighbour count followed by 5-neighbour count show much better results even though there are no substantial differences with the other algorithms considering the scale of the alignment error (e-7). The vertical lines at the end of each cumulative curve are a result of Python’s matplotlib library error.

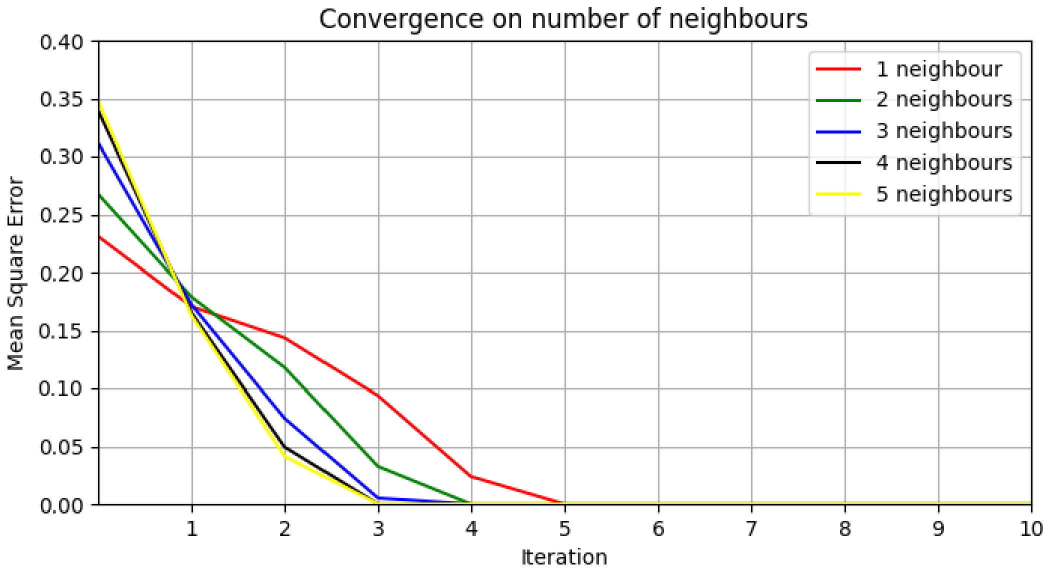


Figure 7.10 Convergence results for registering full shape, 2M5T at initial rotation angle 15 degrees on its x-axis using the TaggedICP algorithm with different numbers of neighbouring atoms. The graph shows that 1 to 5 neighbour(s) achieved a perfect match because all algorithms converged at 0 mean square error. However, the 5-neighbour test was noticed to have converged faster with the 1-neighbour test converging the slowest.

The optimal neighbour test for 2M5T (Figure 7.10) shows the convergence results for matching 2M5T with 1 to 5 neighbours. It can be observed that 5 neighbours converged faster followed by 4-neighbour, 3, 2 and finally 1 neighbour converging the slowest in terms of the number of iterations at convergence. It can also be noticed that all the algorithms converged between 3 and 5 iterations.

The difference between the graphs in Figure 7.9 was too small to regard any as a worse match, however, the convergence graph does not reflect a similar scenario. A good shape match does not necessarily result in faster convergence so Figure 7.10 shows good results for all the neighbour points with a very small difference in the converging values because they all converged to the same error even though at different iterations.

This difference in the two graphs (Figure 7.7 and Figure 7.9) is because of the distribution of atoms in the molecules as well as their sizes (Table 7.1). 2HAX contains 1614 atoms of atom types, C, N, O, and P (4 atoms) whilst 2M5T contains 2119 atoms with the same types of atoms and hydrogen (5 atoms). 2M5T also contains one more partition, Hydrogen (H). 2M5T converged at a higher iteration because it has a partition with a larger number of atoms. Section 6.3 explained that TaggedICP' s number of distance calculations is reduced by a factor, where the factor will be smallest for a larger number of equal-sized partitions. The partitions' details for 2HAX and 2M5T shown in Table 7.1 show that both shapes do not have equal-sized partitions, however, a study of which algorithm is closer to this phenomenon reveals that 2M5T's largest three (3) partitions by percentages are H (48.61), C (32.33), and O (10.1), whilst the largest 3 partitions for 2HAX are C(52.24), O (33.46), and N (13.2). 2HAX has the largest two partitions between the two shapes, with the third partition close to 50% of the second partition whilst 2M5T's third partition is 30% of the second partition. Overall 2HAX has a closer adherence to equal-sized partitions than 2M5T.

The optimal neighbour count can likely be affected by the distribution of atoms (how evenly each atom type is spread in the molecule) in the molecule as well as the size of the molecule. However, for purposes of these experiments and the datasets, all the neighbours yielded substantial results meaning that the choice of neighbours between 1 to 5 does not matter. 3-neighbour yielded slightly better results on one graph (Figure 7.7) so the mean value of 3 neighbours is chosen for the neighbour count in TaggedICP.

Comparison to find the optimal search radius and k-nearest neighbour counts that would yield the lowest classification error for the similarity between atoms evaluated at 1, 3, 4, and 5-nearest neighbour counts, found 1 or 3 neighbours to perform much better than 4, 5 neighbours (Ellingson and Zhang, 2012). The research did not consider 2-nearest neighbours because it was guaranteed to produce identical results to the nearest neighbour. The Triangulation-based Iterative-closest-point for Protein Surface Alignment algorithm (TIPSA) developed for the research, used a combination of radius and atom types to find the maximum number of atoms that can be used in the matching of protein binding sites. The approach first computed the 3-dimensional Delaunay triangulation (Escobar and Montenegro, 1996) to acquire a set of tetrahedra with labelled atoms as vertices. This presented a set of points that was used for the initial alignment. To determine the correspondence for each atom in the data protein, the same atom type is searched in the reference protein with a distance smaller than a set cut off radius value. This approach was to ensure that correspondences do not share the same matches.

This is usually a partial shape match as it matches part of one protein to a binding site (a region on the other structure that the reference protein can dock successfully) or part of another protein. Whilst TaggedICP considered full shape matching of the complete molecule. TIPSA used a fixed threshold distance value that was compared with the distances between the atoms to establish an accepted correspondence based on the distance and the atom within the radius, whilst this research considered the neighbours based on Euclidean distance of closest neighbours to an atom.

TaggedICP considers three (3) neighbouring atoms to register a point cloud based on the results of the tests (Figure 7.7, Figure 7.8, Figure 7.9, and Figure 7.10). It will not have been ideal to set the neighbour count to a small number, e.g., one (1) because the probability of having many atoms in a molecule with just one (1) closest neighbour matching the same atom type with the closest neighbour of another atom in the same molecule could be high. This will reduce the efficiency of finding a meaningful correspondence by relaxing the constraints, especially for large molecules. Increasing the neighbour count provides a stricter constraint which reduces the probability of finding multiple matches satisfying the same condition, but at the same time, this also increases the distance to the found closest matching point and could reduce the time taken for the algorithm to converge. If the point clouds are identical, it is easier to discover correct correspondences, however, when there is noise or mutations, it can impact the performance of the algorithms due to using false matches. Following the experiments, TaggedICP will be used with 3 neighbours.

7.6 Results of Shape Matching

ICP can fail if the initial transformation is not an optimal estimate. The algorithms all start by moving the molecules so their centres of mass coincide, but they can still be rotated. This experiment will test specific initial rotation angles to see whether any of the algorithms can handle large initial rotations better than the standard ICP.

Match quality results are presented in the form of graphs showing the cumulative number of points against their mean squared error values. The match quality graphs that have their lines ending within the visible scale of the graph have a terminating horizontal line. The convergence graphs are presented in the form of mean squared error values against the iteration number. The final matching shapes are shown to be able to see a pictorial view of the complete matches and visually inspect the overlap of the result. With all the visual results, the blue shapes represent the data point cloud, and the red shapes represent the model point cloud.

To show related graphs in a single view, smaller sub-graphs are used for algorithms that are out of scale with the main graph. This usually happens with huge differences in the scale between the main graph and an algorithm's scale. This makes the representation on the same graph inappropriate. The wide differences make details of some algorithms hidden when they are all plotted on the same graph. Using multiple graphs ensures that all the details on the graphs needed for the analysis are visible in the same space.

As discussed in Section 7.3 (Design), four tests were conducted. The full shape registration tests also called same shape matching were designed to match full shapes for both 2HAX and 2M5T. This test represents the most basic of the tests because all the points in the data point cloud had true matches in the model point cloud.

The noisy shape (2HAX and 2M5T) registration tests were designed to test full shape registration in the presence of different levels of noise. Even though all points in the data point cloud had true matches in the model point cloud for the noise tests, the points were displaced by a level of noise represented by its vector magnitude as described in Section 7.6.3.

Another test was the mutation test. Mutation tests were designed to register 2JKF and 2JKG and compare the match quality with the similarity score on the PBD website. 2JKF and 2JKG are known natural mutations so they have some similarities. The test is not to measure the effect of the rotation on the datasets per se, but rather to understand whether the matching with the algorithms is a coincidence or the algorithm's design show advantages or limitations is influencing the match result. Comparison with the structural similarity score is to explore the idea that the structural similarity preserves more information than the sequence information. Meaning that a cumulative percentage of points at an acceptable MSE error greater than the sequence score proves this point.

The implementation was run for 2HAX instead of 2M5T because 2HAX showed a better quality of match using 3 neighbours, so amongst those 2 molecules, we chose the option that we assumed to give us the best possible match using TaggedICP' s closest point search.

7.6.1 Matching Full Shapes with Different Rotation Angles

To begin the shape matching process, the algorithm rotated the data shapes at different initial rotation angles (15, 30, 60, 90, 120, and 150 degrees) around the x-axis, whilst registering the shapes each time. Each algorithm was run for 100 iterations, instead of terminating at convergence or a set mean squared error threshold. This is because it was expected that algorithms will have different convergence errors and setting a global convergence error value might miss some details in the experiment. The choice of 100 iterations was based on past experiences and experimentations.

The size of the point cloud can also be used to estimate the set error threshold however, because of the noisy (error) shape matching and the complexities with the partitions of the algorithms, a standardised large number of iterations was ideal as it captures the convergence and match quality details within the same 100 iteration threshold.

These tests evaluated the algorithms' (ICP, SubsetICP+, TaggedICP, and KmeansICP) match quality and convergence over different initial rotation angles (15, 30, 60, 90, 120, and 150 degrees). The rationale was that ICP can fail if the initial transformation is too large. All the algorithms start by moving the molecules so their centres of mass coincide, but they can still be rotated, so this experiment will test specific initial rotation angles to understand how any of the algorithms compare to each other in terms of the convergence and match quality results and how their techniques influence their performance.

The full same shape test performed a full shape registration test for each of 2HAX and 25MT, meaning that each point in the data shape had a true match in the model shape. The test represents the most basic of all the tests as it does not introduce any level of noise and all the points in the data shape have a true match in the model shape.

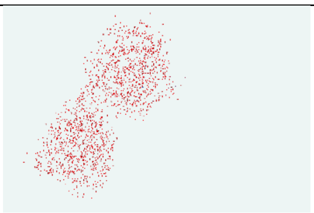
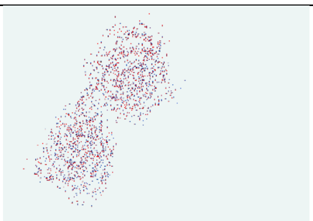
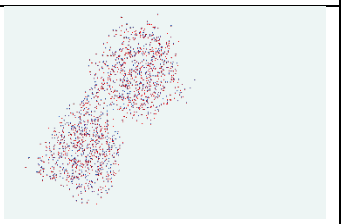
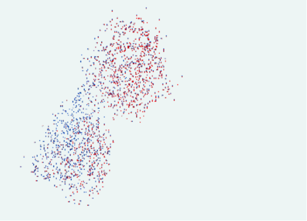
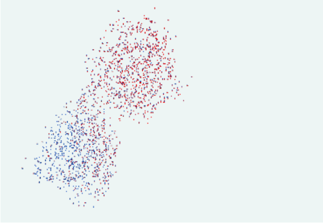
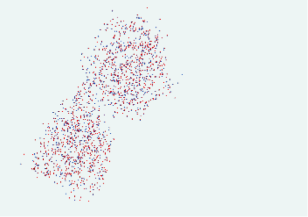
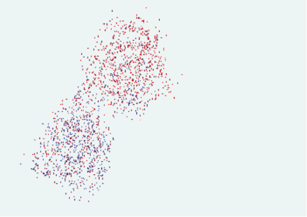
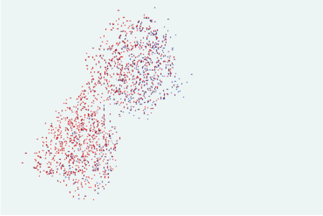
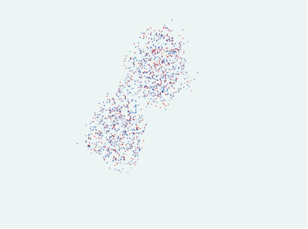
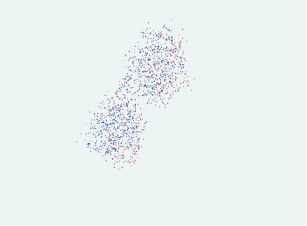
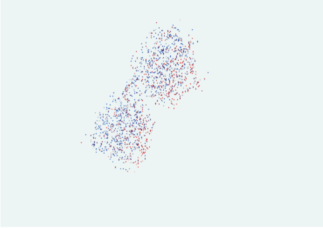
	Angle		
Algorithm	15	60	90
ICP			
TaggedICP			
Kmeans ICP			
SubsetICP +			

Figure 7.11 Final registration results, with initial rotation angles for matching 2HAX (full shape matching subject to varied rotation angles on the x-axis) (blue=data point cloud, red=model point cloud). It can be noticed that ICP at angle 15, TaggedICP at angle 15, and SubsetICP+ at all angles visually show a good match, and the worst results are the ones for KmeansICP. These images are qualitative and are further analysed together with the match quality and convergence graphs.

The results for matching 2HAX (Figure 7.11), visually show shapes with distinctively red points or blue points (perfectly registered shapes) and shapes with a mixture of blue and red points (fewer quality matches). The perfectly registered shapes are in red or blue because of the effect of the small mean square error between corresponding points in the shapes and the shapes captured from just one angle of the 3-D shape. This error is seen to be much larger when there is a clearer distinction between the colours (red and blue) in those shapes. Basically, the larger the error, the clearer it is to identify the individual points visually.

Interpreting the quality of the match purely from the perspective of our human vision shows that shapes with some extent of clearly identified point pairs (point clouds with higher separation of points) such as in KmeansICP at angle 60 or TaggedICP at angle 60 are not as accurate as those matches where there is not much colour separation. For instance, ICP at angle 15 cannot visually distinguish most of the individual point pairs in the overlapping areas representing a much better match than the other algorithms. Many of the shapes show this trend in parts of the point cloud, however KmeansICP and ICP at angle 15 show this trend on the whole point cloud.

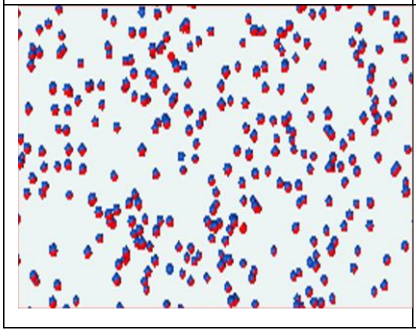
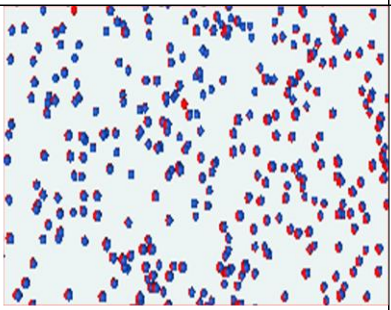
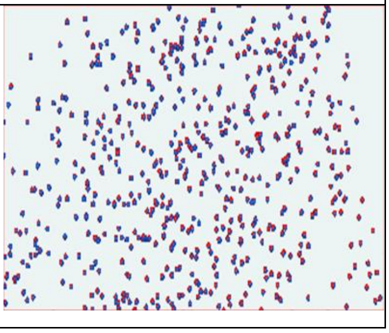
KmeansICP60	TaggedICP60	SubsetICP+60
		

Figure 7.12 Zoomed in overlaps showing a visual analysis of the shapes with more visible matches of point pairs (KmeansICP and TaggedICP) compared to SubsetICP+. This is solely for visual analysis.

The zoomed-in results for matching 2HAX at angle 60 (Figure 7.12) show a zoomed-in version of Figure 7.11 with a selection of the algorithms (KmeansICP, TaggedICP, and SubsetICP+ all at angle 60). Figure 7.12 shows a clear view of the overlaps of the shapes and this enhances the reader’s understanding of how the representation of the matched shapes are meant to show the quality of the shape match, however, a more objective analysis is given in the match quality and convergence graphs.

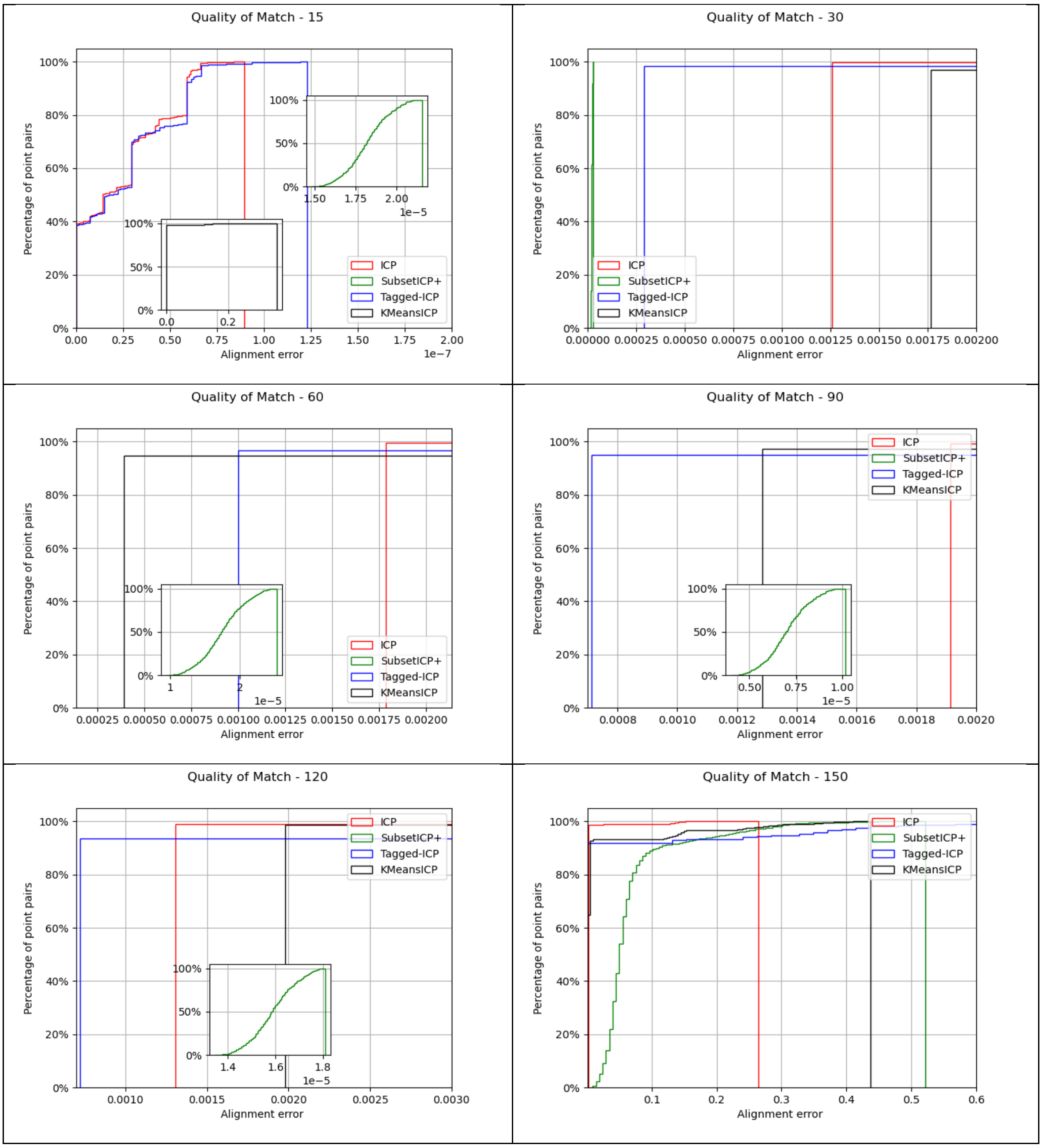


Figure 7.13 Final registration quality results for registering 2HAX with itself. Graphs show results for the percentage of points pairs against alignment error and algorithms that rise quickly are of a better match quality because a higher percentage of point pairs will have a smaller alignment error as the graph rises sharply.

The match quality graphs in Figure 7.13 present a more objective and detailed analysis of the shape registration visual results in Figure 7.11. The quality of the registration is presented in the form of a cumulative percentage of point pairs against their alignment error as discussed in Figure 7.6.

In Figure 7.13, all algorithms are shown to have accurate matches at angle 15. Even though KmeansICP is shown in the sub-graph with increased scale, there were only a few poor matching points (24 of them) with the majority (1599) of the points having perfect matches. As discussed in Section 4.4, the KmeansICP algorithm decides on the correspondence search space by the closest centroid (point-to-centroid), so partitioning is approximate when the shape has not converged because corresponding pairs from a k-means cluster are not necessarily true pairs.

The objective representation of the match quality is also supported by the visual analysis in Figure 7.11 because it is noticed that KmeansICP and TaggedICP show clearer separation of colours at parts of the point cloud, referring to a much lesser quality of match (Figure 7.11) compared to the other algorithms at angle 60. This trend is seen for match quality graphs at angles 30, 60, 90, and 120 where SubsetICP+ shows a comparatively much better quality of the match, with TaggedICP, ICP and KmeansICP following in that order of decreasing quality. However, the trend is not noticed for all the angles such as angles 150 where SubsetICP+ performed the worst.

To understand the performance of SubsetICP+, it is necessary to consider how SubsetICP+ performed much better than all the algorithms in all the tests except angles 150 and 15. This is due to the design of the algorithm. SubsetICP+ matches corresponding subsets of the same atom type. This technique reduces the search space and makes the matching meaningful. The use of corresponding subsets provides a better estimate than when the whole point cloud is matched. Similarly, registering one corresponding subset per iteration is efficient in matching ordered homologous points such as atom types.

Consider the partitioning technique of TaggedICP, KmeansICP and SubsetICP+. It is observed that though TaggedICP uses the 3-neighbour constraint to make the matching meaningful, it reverts to a less optimal match (based on the closest atom in the same atom type partition with no consideration for the 3-nearest neighbours) when the correspondence based on the tag is not found. This reduces the quality of the matches.

For the angles that show SubsetICP+ with a better match (30, 60, 90, 120) than the others, it is noticed that all the other algorithms had approximately similar quality based on the alignment error. This points out that, ICP, TaggedICP and KmeansICP are much similar in terms of their techniques compared to SubsetICP+. However, all algorithms are based on ICP which is the source of their basic similarities. The difference with SubsetICP+ is that it matches one corresponding subset per iteration.

KmeansICP also uses k-means to partition the search space and without the use of any metadata constraint, it is expected that TaggedICP will perform much better. This was true in most of the graphs even though the differences in the quality between kmeansICP and TaggedICP at larger angles was not very significant.

The validity of Figure 7.13 will further be tested in the convergence analysis discussion which will show whether results in the match quality graphs are plausible and find a trend between both match quality and convergence graphs.

In summary, all the algorithms showed a perfect match with a few of them having inaccurate matches with just a few points. This is observed for all the algorithms at all the angles in Figure 7.13. The general observation is that all algorithms had accurate matches at all angles.

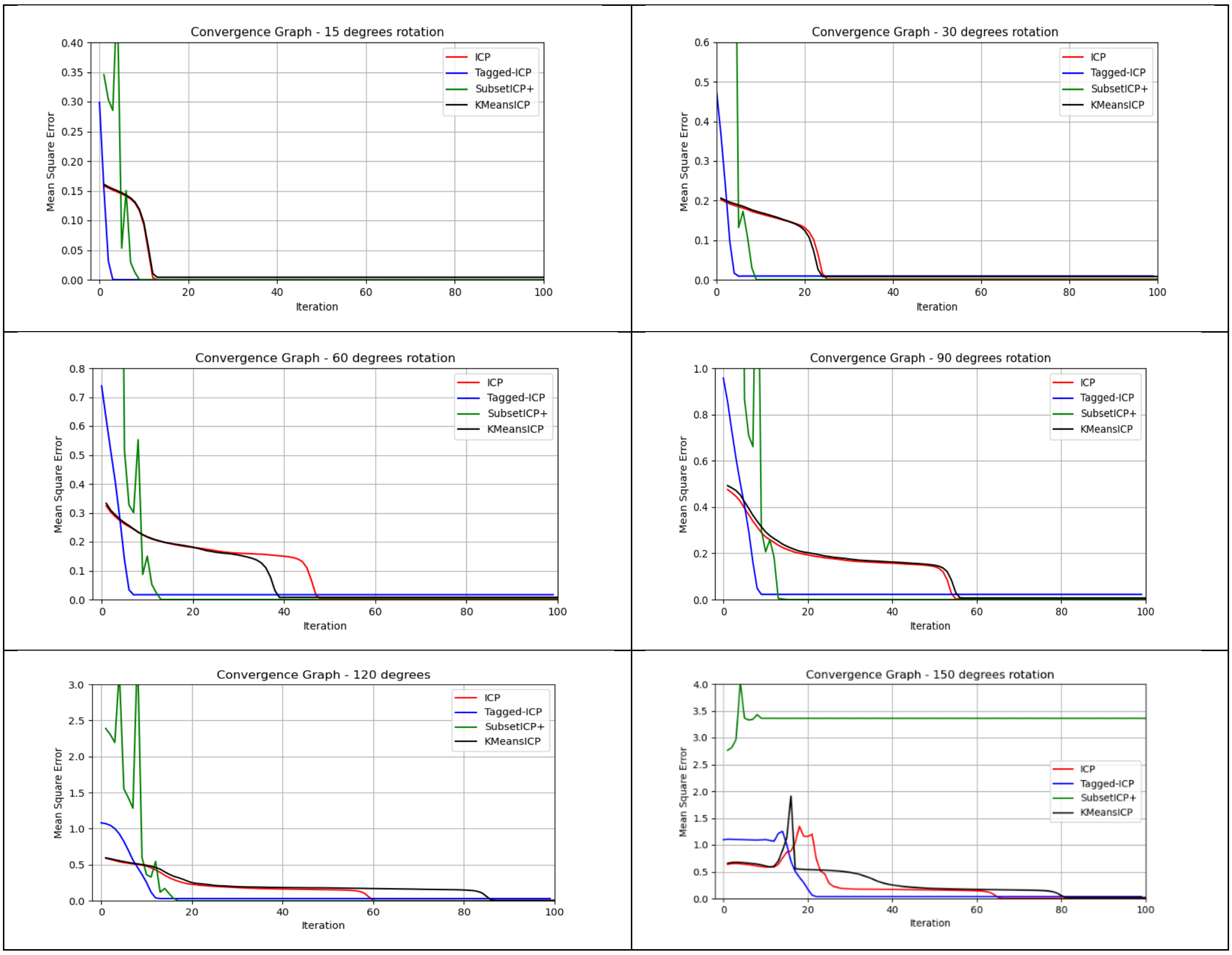


Figure 7.14 Convergence results for matching 2HAX with itself. Graphs show how TaggedICP and SubsetICP+ had a larger starting error but managed to converge faster than other algorithms. The graphs also show TaggedICP and SubsetICP+ converged at almost a constant number of iterations for each angle whilst KmeansICP and ICP show the effect of increasing initial rotation angle on the convergence.

The convergence graphs in Figure 7.14 show that all algorithms converged within the 100-iteration mark. TaggedICP and SubsetICP+ converged at a smaller number of iterations compared to the other algorithms for most of the angles, even though this does not necessarily mean less time in seconds for the convergence. This is because of their use of metadata or known information. Their partitioning by atom types enhances the matching and using neighbourhood information even enhances this further as seen in TaggedICP converging faster for all the angles.

For TaggedICP and SubsetICP+, the faster convergence is because of the use of metadata knowledge to aid with the registration process. Since tag information constraint is constant across the angles, the rotation of the shape is found to have a smaller effect on the iteration at convergence compared to KmeansICP and ICP. It is noticed that SubsetICP+ and TaggedICP+ had all converged within 20 (0 to 20) iterations except at angle 150 for TaggedICP. However, ICP and KmeansICP have a convergence between 10 and 90 iterations. For all angles, it is observed that the iteration at convergence is related to the angle as increasing angles results in increased iteration at convergence.

TaggedICP shows better convergence performance than SubsetICP+ at increasing angles because TaggedICP’ s constraint is at an appropriate level of complexity. It is not too rigid (using a very tight constraint) nor too loose (using insufficiently fewer constraints) to present a lot of false matches. The 3-neighbour tag information allows the matching to be meaningful than SubsetICP+ because, SubsetICP+ doesn’t use metadata in the correspondence search space, but rather just Euclidean distances. Even though the search space is within the same atom type, it is still not as meaningful as TaggedICP because of the use of the 3-neighbour constraint.

In Figure 7.14, KmeansICP and ICP showed a similar convergence trend. It can be observed that, for both algorithms, there was an increase in convergence error with increasing initial rotation angle. At angles 15, 30, and 90, both algorithms converged at approximately the same iteration. This behaviour is expected of the ICP algorithm as large initial rotation angles are noted to result in less accurate matches.

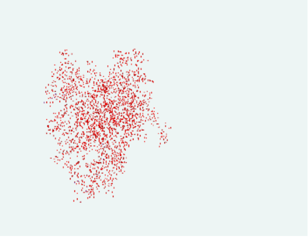
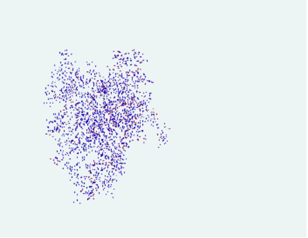
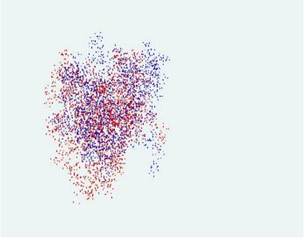
KmeansICP’s correspondence search method is the same as the original ICP implementation as compared to the other algorithms and the correct point matches are less likely to be found, leading the algorithm to converge to a local minimum. As discussed in the KmeansICP algorithm in Section 4.3, although the algorithm partitions the search space to reduce the time spent for the correspondence search, the search space (a k-means cluster) is simply decided by the closest centroid to the point in the data point cloud. This is like the original ICP algorithm that finds the closest point in the model point cloud for each data point although within a limited search space. This limited search space results in kmeansICP converging faster than ICP at larger angles (60,120, and 150) even though not in the case at angle 90 where ICP converged at slightly lesser error than kmeansICP, however, this is not a significant difference.

The convergence graphs show that SubsetICP+ had an oscillating line at the earlier iterations. This is because of the differences in the errors between subsequent subsets at the beginning of the iteration. Since each iteration returns a transformation based on registering an atom type-based subset, which is applied to the whole point cloud, this error is applied to the global shape, so the shape registration process regularises from a larger error to smaller errors.

In conclusion, it can be understood from matching full same shape, 2HAX that the partitioning technique influences the convergence by reducing the number of iterations. The match quality test in Figure 7.13 showed increasingly less quality of match as the initial rotation angles increased. However, this was not always the case for SubsetICP+.

SubsetICP+ and TaggedICP showed a better convergence than the other algorithms since both algorithms partition by known information. It can be concluded that the partitioning technique influences the match quality of the algorithm and the convergence in that the more known information is used in the matching, the better the match quality and the faster the convergence in terms of iterations.

Like 2HAX, a full shape matching was performed using 2M5T. This was to ascertain whether the same results in 2HAX will be found and further critique the influence of partition techniques on match quality and convergence. 2M5T contains Hydrogen atoms unlike other shapes in our experiment and this will result in one larger partition (H atom count is 1028). SubsetICP+ and TaggedICP are indeed only for molecules, or similar 3-D structures but KmeansICP could apply to any point clouds.

	Rotation Angle		
Algorithm	15	60	90
ICP			

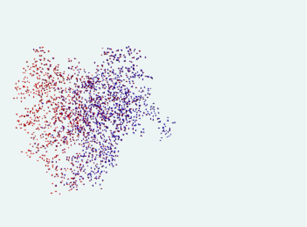
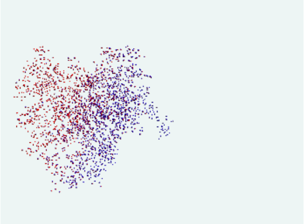
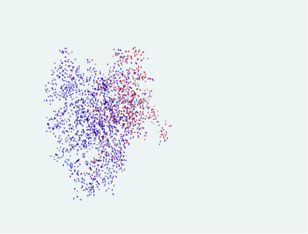
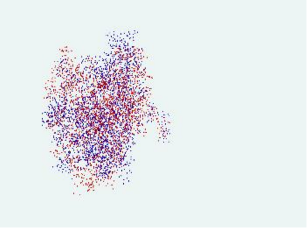
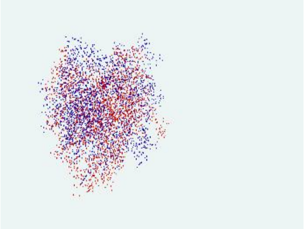
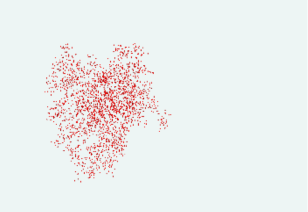
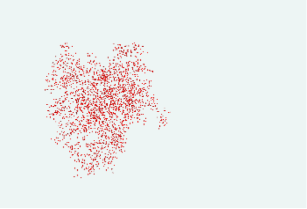
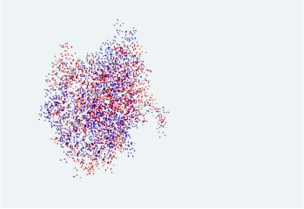
TaggedICP			
Kmeans ICP			
SubsetICP+			

Figure 7.15 Registration results for matching 2M5T with itself at different initial rotation angles (blue = data point cloud, red= model point cloud. ICP at angle 15 and SubsetICP+ at all angles visually show a good match, whilst other algorithms show a mixture of good and bad registration)

The registration results for matching 2M5T (Figure 7.15), visually show shapes with distinctively red points or blue (perfectly registered shapes) and shapes with a mixture of red and blue points (comparatively lesser quality match). A detailed description of the visual analysis using the colours is already discussed in Figure 7.11.

From Figure 7.15, SubsetICP+ shows a much better quality of match at angles 15, and 60, whilst ICP and TaggedICP show a much better quality of match at angles 15 than any other angles of their test. For angle 60, SubsetICP+ shows better quality than all the other algorithms as shown in the zoomed-in Figure 7.16. Comparing these zoomed-in images with the images in Figure 7.15 allows an understanding of the match quality of the other angles. For instance, a mapping of the patterns observed in Figure 7.16 can be used to further analyse other images in Figure 7.15.

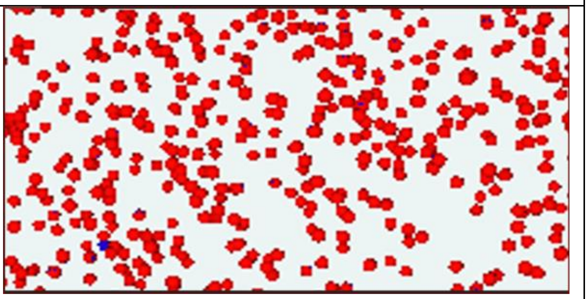
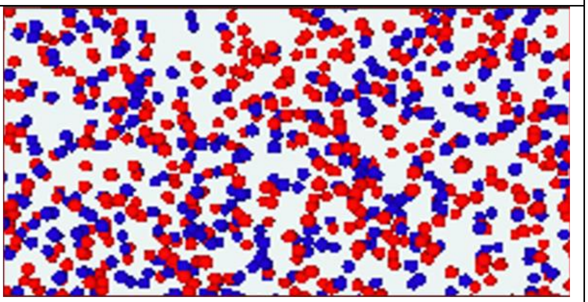
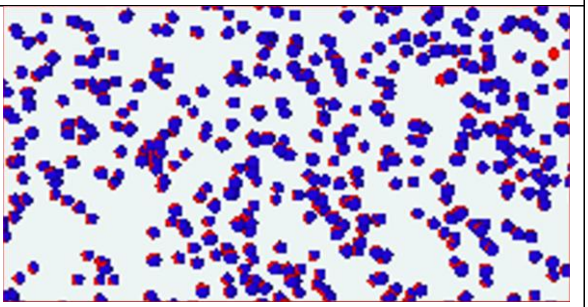
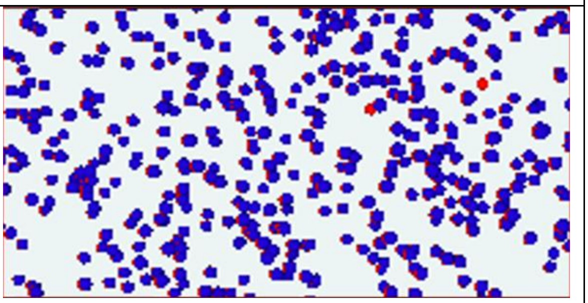
	Angle at 60
SubsetICP+	
KmeansICP	
TaggedICP	
ICP	

Figure 7.16 Sample zoomed-in images of final matches showing the extent of overlaps of point pairs. SubsetICP+, TaggedICP, and ICP show a better quality of match because there is a small colour separation due to the overlaps.

In the zoomed-in images for registering 2M5T at angle 60 (Figure 7.16), the patterns show ICP, SubsetICP+, and TaggedICP with no large separation of colours except for two points. This represents a better quality of match because it shows that the distances between corresponding points are minimal. The other algorithms show varying levels of atom colour separation. The slight separations represent the level of overlap of the part of the point cloud.

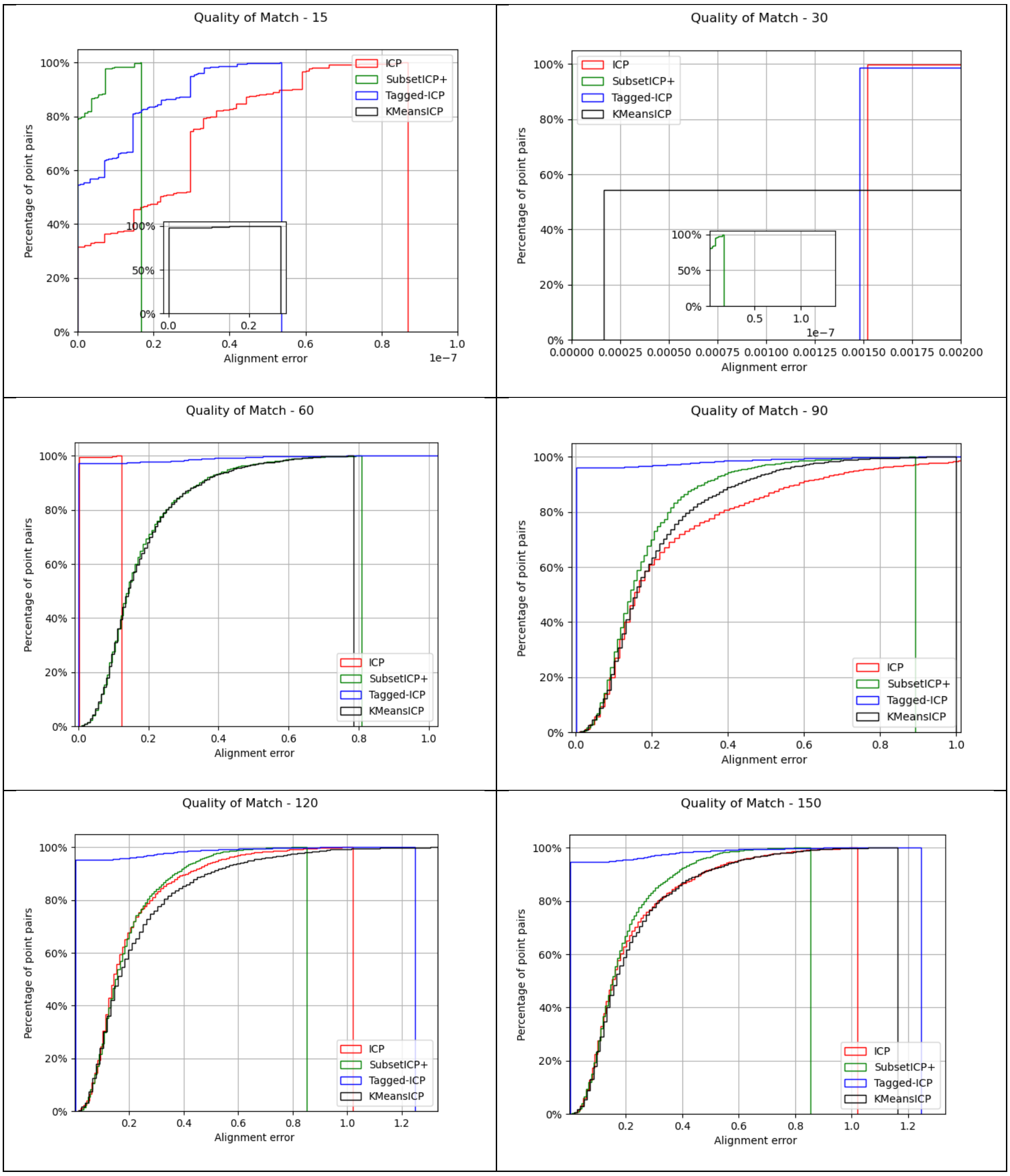


Figure 7.17 Match quality results for matching full shape 2M5T. Graphs show the percentage of corresponding point pairs against the alignment error. SubsetICP+ and TaggedICP are noticed to have a much better match quality compared to the other algorithms.

Figure 7.17 represents the test matching full shape 2M5T. This is an objective measure of the visual images in Figure 7.15. It is observed that all algorithms performed much better at smaller angles especially for SubsetICP+ that showed a better match quality at angle 30.

It is also noticed how TaggedICP performed consistently better across different initial rotation angles especially angles 60 to 150. This was not observed when matching 2HAX (Figure 7.13), rather SubsetICP+ performed better at angles 60 to 120. This shows that for matching full shapes, a larger partition size aids matching with TaggedICP because of the fractional partition sizes, however, this is not the case for SubsetICP+ since a larger partition will present larger errors which will be applied to the whole point cloud.

it is also observed at angles 60 to 150, that SubsetICP+, KmeansICP and ICP had similar performance with a noticeable trend in their curve. For kmeansICP and ICP, their large similarities are responsible for this trend, However with SubsetICP+, it was expected that the algorithm will mimic the better match quality shown in matching 2HAX (Figure 7.13), however, it was rather seen to perform at par with the other algorithms except at angle 30.

To consider the partitioning and the distribution of atoms in 2M5T on the effect of the performance of SubsetICP+, Table 7.1 is revisited. It is realised that 2M5T's largest partition is H representing close to half the size of the point cloud. Matching smaller subsets prevents the algorithm from being trapped in a local minimum and as such a partition of that size is not ideal as it presents large errors that are applied to the whole point cloud, however, with 2HAX, SubsetICP+ equally matched a partition that was 52.54% of the point cloud as shown in Table 7.1, therefore the claim about SubsetICP+ performing below expectation due to the size of the H partition cannot be sustained. It is not clear about the inter-atom distances between the H atoms or how the distribution of each atom affects the matching, however, the complexity (how evenly distributed points are in a point cloud) of a shape contributes to the quality of the matching process.

Figure 7.17 shows that KmeansICP and ICP have similar line curve because of the similarities of their closest point search method. Both algorithms perform a standard ICP closest point search. Although KmeansICP reduces the search space due to the k-means clusters, the efficiency with the k-means clusters finding a meaningful match is not as efficient as the partitioning in TaggedICP or SubsetICP+.

From Figure 7.17, it can be concluded that metadata information helps the match quality of a shape matching process. Although this was not a consistent observation throughout the tests, e.g., at angle 60, ICP and TaggedICP performed better than the other algorithms instead of TaggedICP performing better than the rest, however, the general case at larger angles (90, 120, 150) shows better performance for TaggedICP because of the use of metadata knowledge.

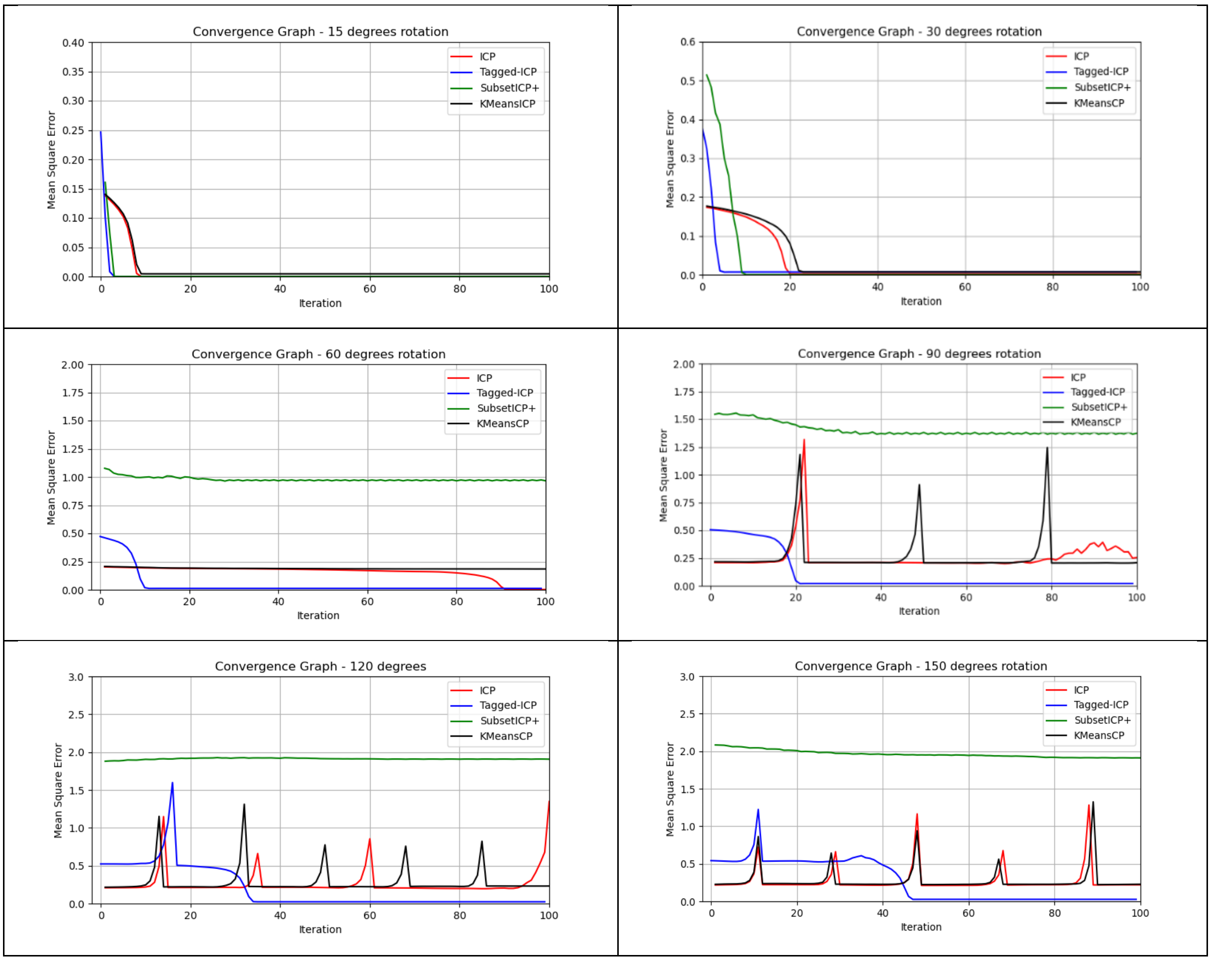


Figure 7.18 Convergence results for full shape matching of 2M5T. The graphs show the consistency of KmeansICP and TaggedICP algorithms. All algorithms seem to converge at roughly 10 iterations at angle 15 degrees with this figure increasing as the initial rotation angle increases. SubsetICP+ and TaggedICP consistently performed much better than the other algorithms. At larger angles, SubsetICP+ is seen to not have converged. As discussed in section 2.6.3.3, ICP and KmeansICP converged to a less optimal solution because of the large initial rotation angle. The algorithm is seen to oscillate because it converged at a non-optimal position. The algorithm detects the diversion and re-enters the point cloud.

Figure 7.18 shows convergence results for matching full shape 2M5T. All algorithms are shown to converge at increasingly larger iterations as the angle increases. However, TaggedICP showed convergence at consistently lower iterations even as the angle increased. KmeansICP and ICP are shown to have a better convergence at lower angles and then both oscillate later at larger angles (90, 120, 150). SubsetICP+ is seen to not have converged at angles 60, 90, 120, and 150.

TaggedICP performed much better than the other algorithms due to the use of the metadata tag information used for the partitioning. It can be noticed from Figure 7.18 that for angles 120 and 150, TaggedICP had a similar oscillating trend at one iteration and was able to recover due to the enhanced closest point search due to the tag information (atom type and 3-nearest neighbour information). The performance of these two algorithms (ICP and KmeansICP) agrees with the match quality graphs in Figure 7.17. The performance of the algorithms in registering 2M5T has a convergence and match quality result that validates the shape registration visual analysis in Figure 7.15.

Even though 2M5T has one larger atom (1028 H atoms), the algorithms were all seen to have performed poorly compared to matching 2HAX at the same larger angles (90, 120, 150). For instance, the performance of the algorithms for 2M5T convergence at angle 90 is much worse than the same performance for 2HAX at angle 90 in Figure 7.14.

To establish a relationship between the size and number of partitions of the shape on the performance of the shape registration for a particular algorithm. The registration results show that larger mean squared errors were encountered in the 2M5T as compared to 2HAX. Particularly for angles 120 and 150 degrees. However, these results were more consistent than not across all tests and angles to determine that the size and number of partitions affect the registration results.

From these tests, it is concluded that larger rotation angles had a large effect on convergence and match quality. The use of atom type partitions imposes a constraint that improves the matching by reducing the correspondence search space and makes the correspondence meaningful or at least close to that.

It is also observed that the match quality shows a relationship with the convergence for 2M5T. It is found that where the match has a higher quality, a convergence at lower iteration is usually seen. However, for 2HAX, it is rather noticed that the use of neighbourhood information played a role in the convergence as the match quality was not always related to the convergence. In matching 2HAX, TaggedICP converged at a smaller number of iterations at all angles, however, it did not have the better match quality for all angles.

7.6.2 Matching Shapes with Levels of Simulated Noise

To simulate noise in the point clouds and study the performance of the shape matching in the presence of different levels of noise in the dataset, the iniial rotations for the data shape for each test was set to (0, 0, 0, 0). The positions of the atoms in the data point cloud were translated by a vector computed from the radius of the point cloud and the noise factor. This is given in the equation below;

$$\overrightarrow{noise} = Random.InsideUnitSphere * radius * noise\ factor$$

Where;

$Random.InsideUnitSphere$ represents a Unity3D (Unity Technologies, 2019) function that returns a random position within the bounds of a unit sphere (Sphere with centre (0,0,0) and radius 1).

$radius$ is the radius of the point cloud represented as a Unity3D game object?

$noise\ factor$ is the noise size used to determine the extent to which to move the position of the atom (0.001, 0.003, 0.005, 0.01, 0.03, 0.05). For instance, a noise factor of 0.001 means the atoms will be moved (random position return by the noise factor) by a maximum of 0.1% of the molecule's radius and 0.05 will have atoms being moved in the direction of the resulting vector by a maximum of 5% of the molecule's radius.

The noise experiments simulated measurement noise by changing the positions of the atoms slightly to mimic measurement noise added by a position measuring device whilst scanning or acquiring the molecule. There are several sources of noise when generating the 3-D structure of a molecule. For instance, using X-rays to scan the molecule presents noises such as random noises that are almost always part of the data due to the emission, absorption and scattering nature of x-rays to and from tissues to acquire the structure of the proteins (Borek, Minor and Otwinowski, 2003, Ryan et al., 2019). Other sources of errors include errors based on the methods used in the process to acquire the molecular 3-D structures such as assumptions of the parameters of certain measuring devices such as X-ray generation devices (e.g. goniometers) and unwanted data caused by interference such as x-ray excitation during the scanning or acquiring process (Ryan et al., 2019).

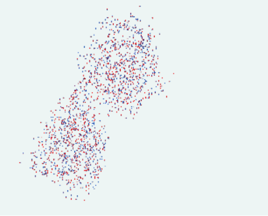
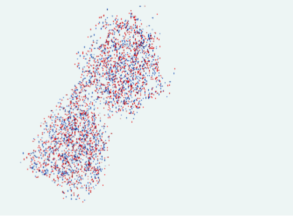
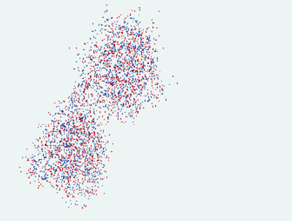
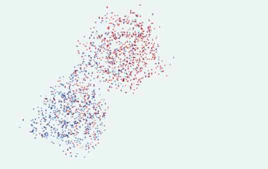
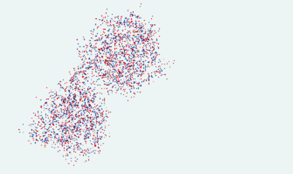
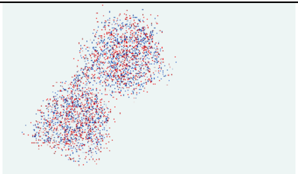
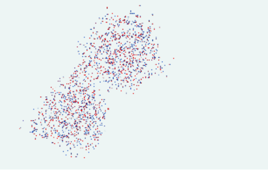
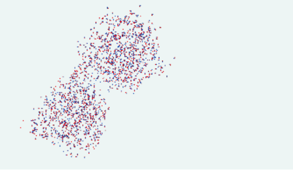
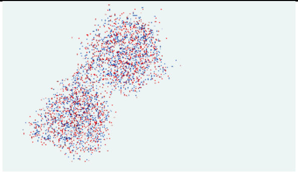
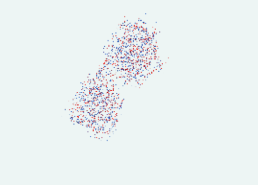
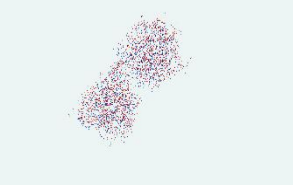
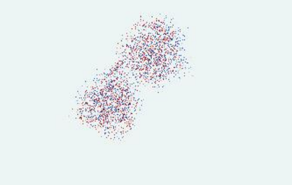
	Amount of noise		
Algorithm	0.001	0.03	0.05
ICP			
TaggedICP			
KmeansICP			
SubsetICP+			

Figure 7.19 Visual results for registering 2HAX with simulated noise and 0 rotation angle on all axes (Blue = data point cloud, Red= model point cloud) for error levels 0.001, 0.03, and 0.05. All images show a level of error. With TaggedICP 0.001 showing a better quality than the other algorithms.

Results for registering 2HAX with simulated noise at 0-degree rotation angle on all axes (Figure 7.19) shows the results of matching by all the algorithms in the presence of different levels of noise (0.001, 0.03, and 0.05). The visual analysis of the final registered shapes shows a similar colour tint for most shapes except TaggedICP at error 0.001, with a better visual match quality than the other shapes. The clearer the separation of colours within a section of the point cloud, the larger the distances in the point pairs and the worse the match quality.

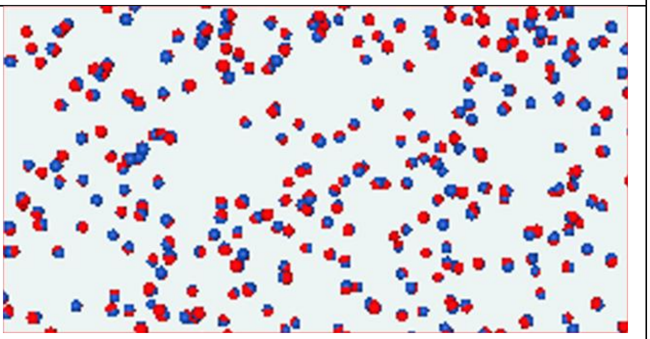
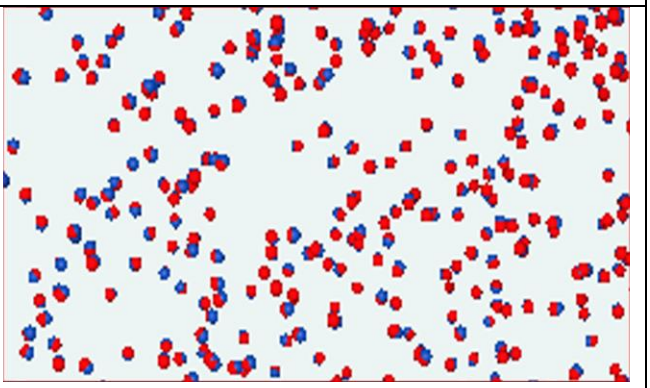
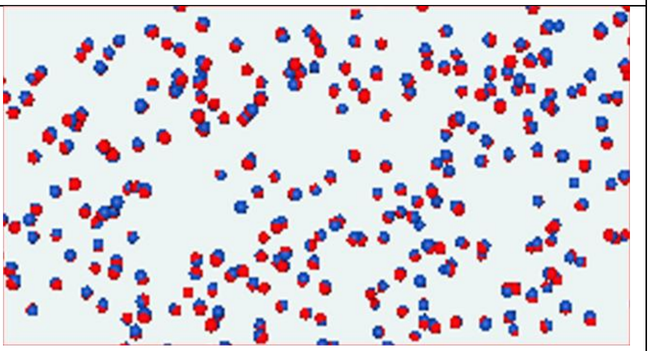
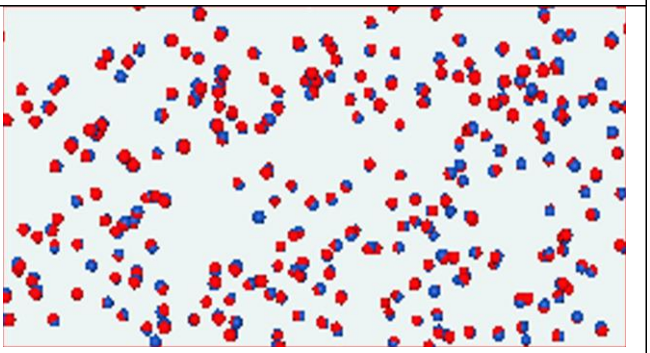
	Noise level 0.001
ICP	
TaggedICP	
KmeansICP	
SubsetICP+	

Figure 7.20 Zoomed-in upper part of the molecule for matching 2HAX with 0.001 level of noise. TaggedICP shows a slightly different colour separation than the other algorithms with more red points to the right section of the image. The red point clouds are more vivid but not in a sense of separation from the blue point clouds. Other images show the uniform colour distribution of red and blue.

Zoomed-in results for registering 2HAX at noise level 0.001 (Figure 7.20) shows zoomed-in matching results for 2HAX, noise level 0.001. The level of overlap of some of the points reflecting the quality of match at those sections of the shape can be seen. For TaggedICP, the distribution of red and blue point clouds is vivid, where red points are denser at the right-hand side of the image and blue points are denser at the left-hand side of the image. The colour separation gives an idea of the quality of match around the region because it helps to show the overlaps. Even though the images were taken at just one angle, they help provide a good visual analysis of the shape match.

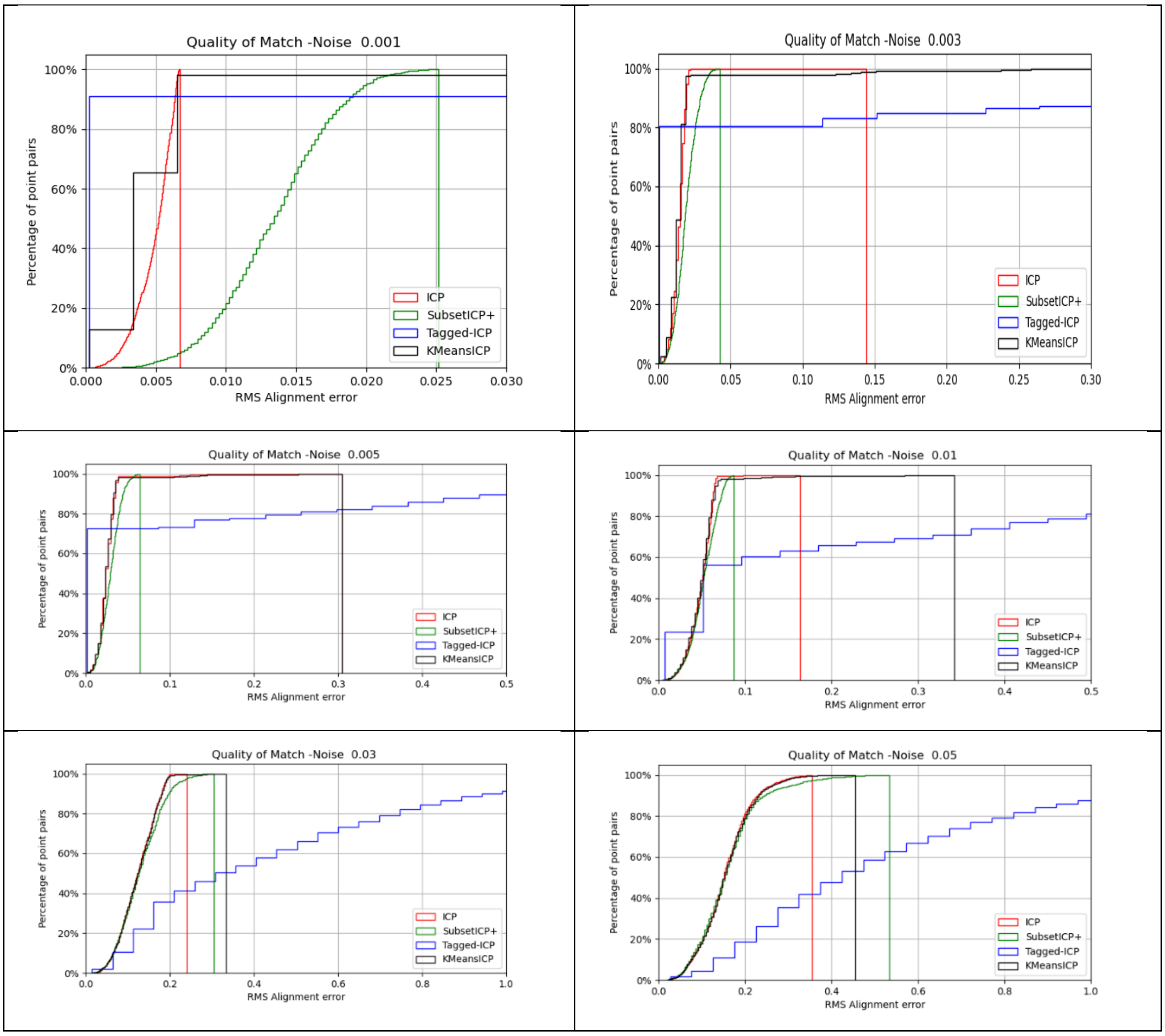


Figure 7.21 Match quality results for matching full shape, 2HAX with simulated noise at 0 initial rotation on both axes. The noise levels from 0.001f to 0.05f indicates a measure of how far each point was shifted to simulate the addition of noise.

The match quality results in Figure 7.21 represent results from full shape matching of 2HAX with simulated noise at 0-degree initial rotation on both axes. It is observed that all alignment errors in Figure 7.21 have relatively larger errors than the results shown in the registration without noise, for instance in matching full shape 2HAX (Figure 7.13). This shift from smaller errors is due to the addition of simulated noise as explained in the matching shapes with simulated noise section (7.6.2). The level of noise determines the distance displaced by a point (atom) in the point cloud (molecule). The noise level changed the relative positions of the atoms and their neighbourhood points that are expected to lead to less accurate matching.

For smaller noise levels (0.001 to 0.005) all algorithms are seen to have a much better quality of match than the larger noise levels. TaggedICP is seen to have a much better match quality than all the other algorithms for noise levels 0.001, 0.003, and 0.005 and the worse performance for noise levels 0.03 and 0.05. This shows that TaggedICP is sensitive to noise with increasing noise levels reducing the match quality. SubsetICP+, KmeansICP and ICP consistently maintained a similar match quality for all noise levels.

It is also noted that for larger error levels (0.01, 0.03, 0.05), all algorithms except TaggedICP maintained a similar level of match quality. It is also observed that the rise in TaggedICP' s cumulative curve decreased continuously as the error level increased. Horizontal parts of the line graph represent alignment errors that were not encountered in the algorithm. For instance, consider TaggedICP at error 0.01, it is observed that a vertical line rises to a little above 20% and then another horizontal line before another vertical line rising from 20% to just below 60%. A higher vertical line represents many points at the same or close error position and a horizontal line represents few points at a closer alignment error. This is further depicted in the zoomed-in image in Figure 7.22.

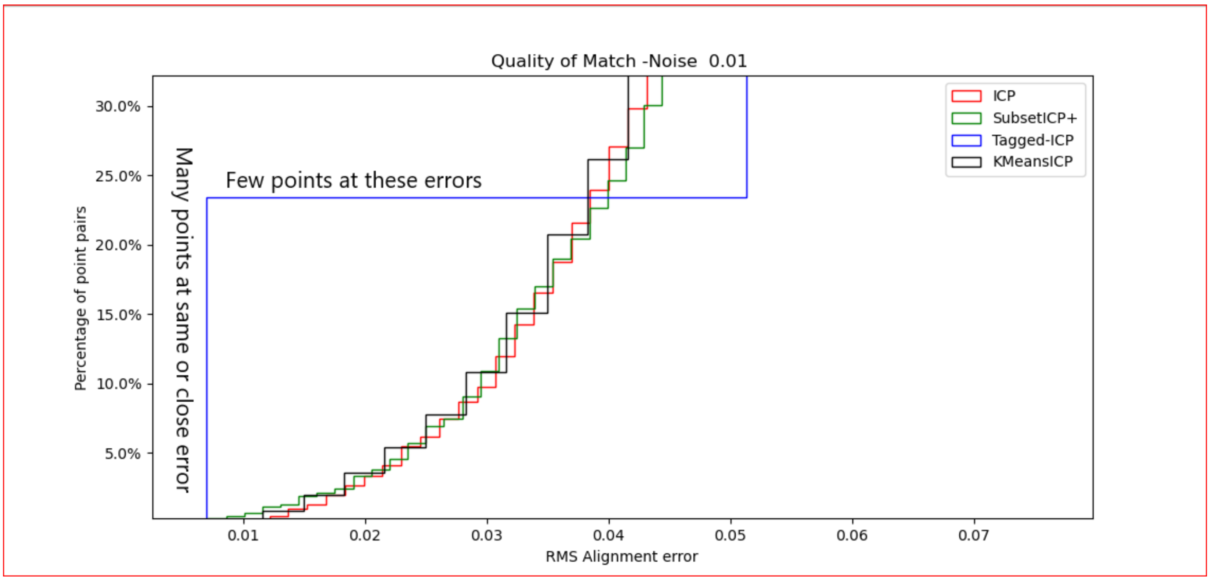


Figure 7.22 Sample zoomed in quality of match with noise 0.01 showing a horizontal line representing a small number of points at that same alignment error and a vertical line showing many points at similar or close error levels from approximately 0.006f to 0.06f error. This graph represents the cumulative histogram with alignment error as such all algorithms will eventually get to the 100% mark, however, it can be deduced from the graph what percentage of point pairs for each algorithm had a particular alignment error representing the quality of the match. For instance, the graph shows

that roughly 23% of point pairs for Tagged-ICP had a better match (less than 0.01 error) compared to the other algorithms. This also means that the average error reduces with many matching points.

It can be deduced from the match quality graphs in Figure 7.21 that TaggedICP had a much better, but gradually decreasing quality of match on all error levels. This trend was also observed in SubsetICP+, KmeansICP, and ICP, however, their error margins were much smaller than TaggedICP. It was expected that all algorithms will decrease in their match quality with increased noise levels as shown by the performance of TaggedICP. TaggedICP had a much worse performance at larger error levels (0.01, 0.03, and 0.05) compared to the other algorithms because of false matches resulting in less ideal correspondence, however, a comparison with the noise experiments in 2M5T should confirm whether this behaviour was peculiar to 2HAX or not.

Even though TaggedICP partitions by atom type and uses neighbourhood information to make the matching meaningful, the presence of these increasing levels of noise resulted in less meaningful matches because TaggedICP reverts to the closest atom in the data point in the search partition without considering the neighbourhood information when no match, using the tag information is found. These correspondences will be less ideal compared to matches based on atom type and neighbourhood information and result in TaggedICP’s less accurate performance.

In most of the graphs, it is noticed that SubsetICP+, ICP, and KmeansICP show a similar trend of performance for most noise levels. Their values show a better quality of match than TaggedICP for larger errors because their curves rise higher than TaggedICP in most places. Even though SubsetICP+ also partitions its search space based on atom type, it was observed to have performed at par with KmeansICP and ICP. This is because the presence of noise adds a layer of difficulty to the registration process, so after partitioning by atom types displaced by levels of noise, SubsetICP+ still performs a standard ICP iteration on each subset. This results in the similarity between the algorithms that have the same closest point search method.

Even though registering subsets per iteration in SubsetICP+ was expected to show smaller errors and as such possibly shows a better quality of match than ICP and KmeansICP, every transformation computed from a subset iteration is applied to the whole point cloud. Although the partitions in SubsetICP+ are based on atom types, the correspondence search method does not use this metadata in the matching. This makes the correspondence search method of all three algorithms the same (ICP, KmeansICP, and SubsetICP+), though with k-means partitions (KmeansICP), atom type-based partitions (SubsetICP+), and no partition (ICP).

It can be deduced from these results that the partitioning technique has a relationship with the performance of the algorithms in the presence of noise levels even though this observation was not consistent. TaggedICP uses the tag information and showed a better match quality at smaller noise levels that gradually decreased at higher error levels.

It can also be deduced that the closest point search method has a relationship with the match quality because KmeansICP and SubsetICP+ use different partitioning techniques, k-means clusters and atom type based, however, they both performed at par with ICP. At higher error levels, KmeansICP, SubsetICP+ and ICP were found to have a consistent performance. This is because, the initial transformations were constant across noise levels, meaning that these algorithms would have the same or similar closest point set thus resulting in the same or similar error across iterations.

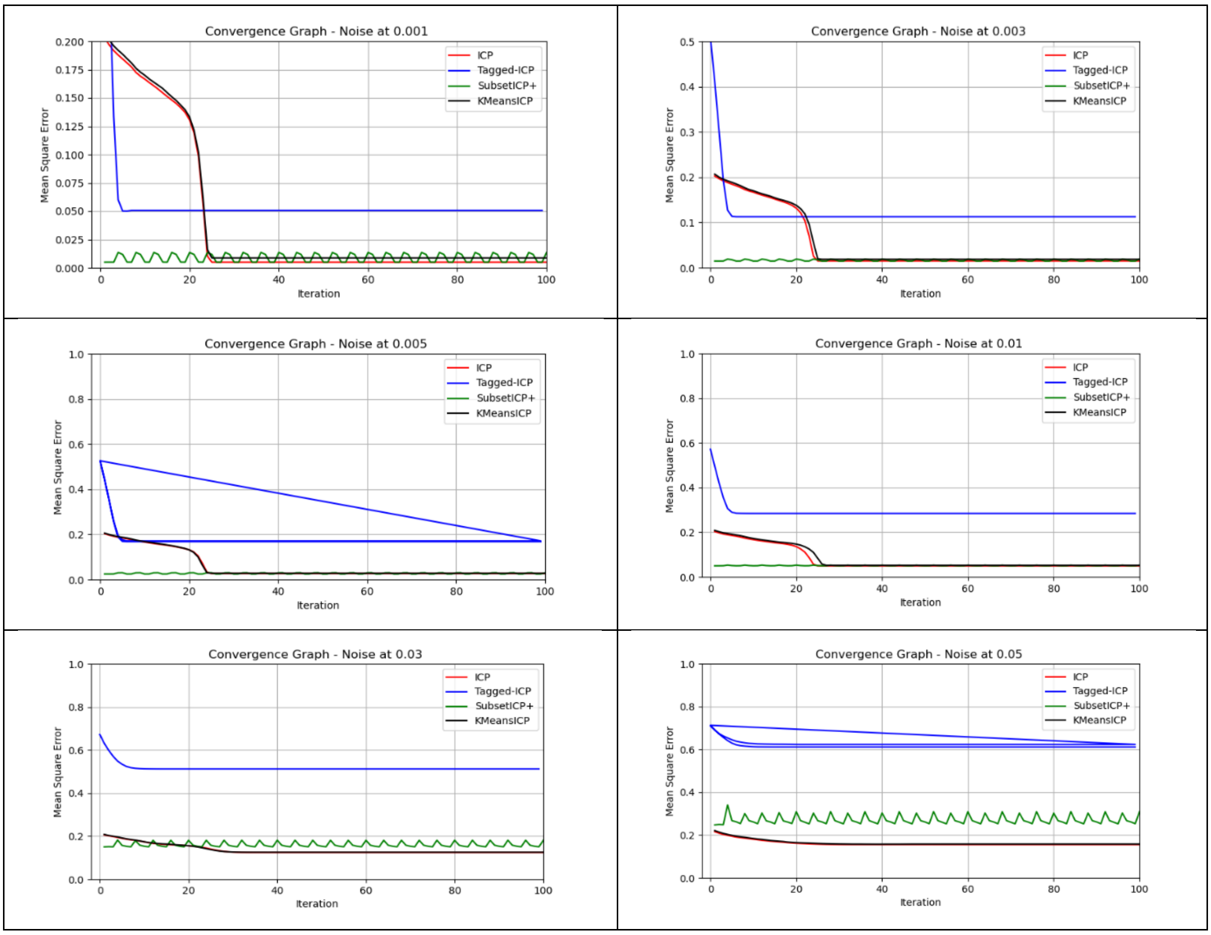


Figure 7.23 Convergence results for full shape matching of 2HAX with simulated noise at 0-degree rotation on all axes. The addition of some noise is shown to have increased the convergence error for all algorithms. The convergence error seems to increase as the error level increases. However, the algorithms showed different errors across the iterations.

Figure 7.23 shows the convergence results for full shape registration, 2HAX with simulated noise at 0-degree rotation on all axes. It can be observed from the graphs that most of the algorithms converged at a higher error and iteration when the noise level increased. However, this observation is not consistent throughout the experiments. Each algorithm converged at a close number of iterations in all error levels and all the algorithms recovered from a higher error except SubsetICP+ that converged on the first iteration for all noise levels. SubsetICP+ converged with a lower error at smaller noise levels and higher error at higher noise levels. The algorithm was also noticed to have oscillated after convergence. This is because matching individual subsets per iteration and applying the transformation to that subset affects the whole point cloud and a different subset is recomputed and reregistered in the next iteration using a different atom type-based partition that also has a different subset size. This irregularity results in a seeming distortion and fix phenomena seen as an oscillation in the graph. If the whole point cloud was registered in each iteration, it would have a global and consistent error that increases or decreases gradually.

TaggedICP had the worst convergence error which is because of the design of the correspondence search method and it also agrees with the match quality in Figure 7.21 as it showed the worst match quality for especially large angles (0.01, 0.03, and 0.05). This also shows that too many constraints should not be used for tentative correspondence search when the underlying transformation is not accurate enough, which may exclude some good matches and help the algorithm to traverse the local minima.

KmeansICP and ICP had a similar convergence performance with their graphs almost identical. This was also observed in the match quality results (Figure 7.21) where both algorithms and SubsetICP+ had a similar graph because of similarities in their correspondence search space.

In the 2HAX convergence graph (Figure 7.23), It can be deduced that the match quality has a relation with the convergence because it shows less accurate match quality reflecting in larger convergence errors. TaggedICP is noticed to have converged at a much larger error than the other algorithms with smaller noise levels because the match quality shows some level of much better matches. It is noticed that for the smaller noise levels TaggedICP shows better matches up to 90% of the points (noise 0.001), 80% (noise 0.003) and with this figure decreasing as the noise level increases, whilst the other algorithms hit 100% mark in the visible scale of the graph.

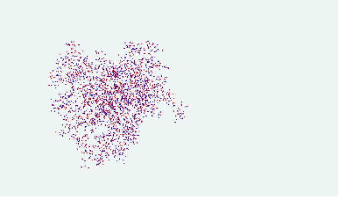
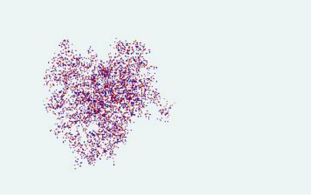
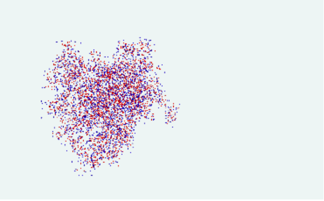
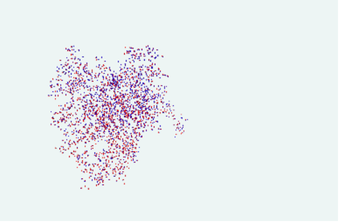
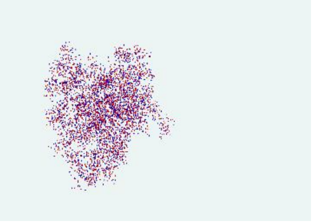
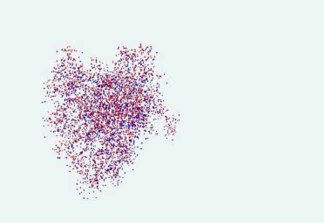
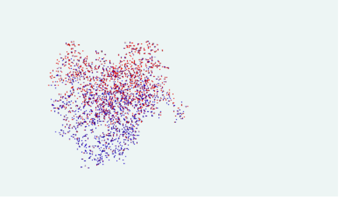
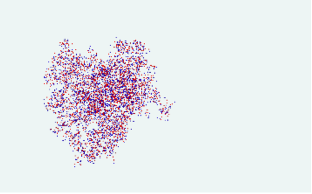
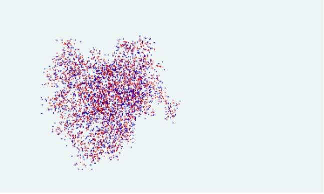
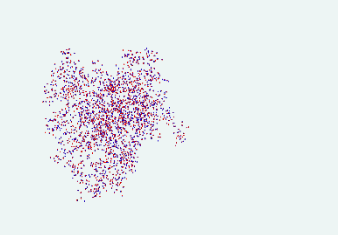
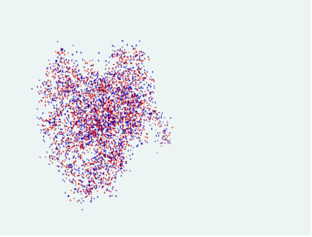
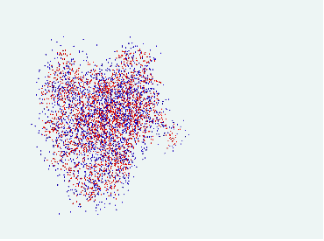
Algorith m	Amount of noise		
	0.001	0.03	0.05
ICP			
TaggedI CP			
Kmeans ICP			
SubsetI CP+			

Figure 7.24 Visual representation of matching full shape, 2M5T with simulated noise at 0-degree rotation on all axes. The separation of colours gives an idea of the quality of the match. The clearer the separation of colours, the larger the distances between final correspondence points and the worse the match quality. Most images show a similar level of quality which can be validated with the match quality and convergence graphs.

Figure 7.24 represents the results of full shape matching of 2M5T with simulated levels of noise at 0-degree rotation. The results of the final registered shapes show a similar colour tint for most images, except KmeansICP at 0.01, which shows a much better level of separation between the blue point cloud and the red point cloud.

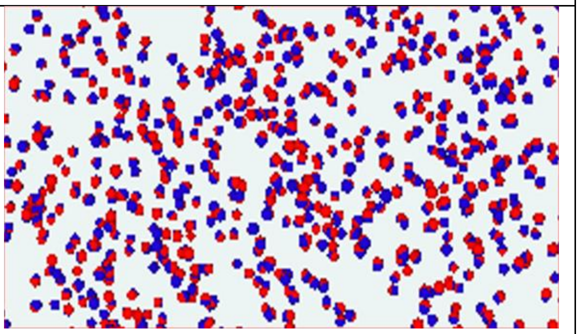
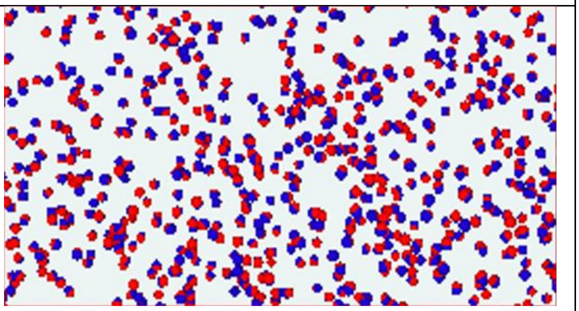
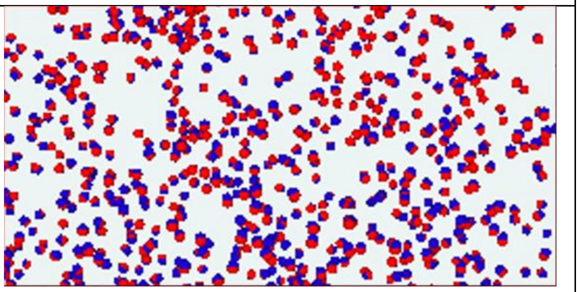
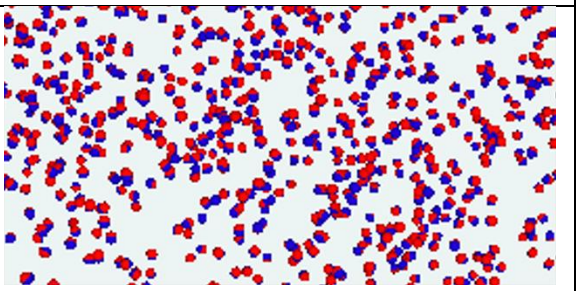
Algorithm	Visual match at error 0.001
ICP	
TaggedICP	
KmeansICP	
SubsetICP+	

Figure 7.25 Zoomed in visual results for matching 2M5T at error 0.001. The separation of colours shows the extent of the match quality. Points farther away from their correspondence shows poorer quality in those regions.

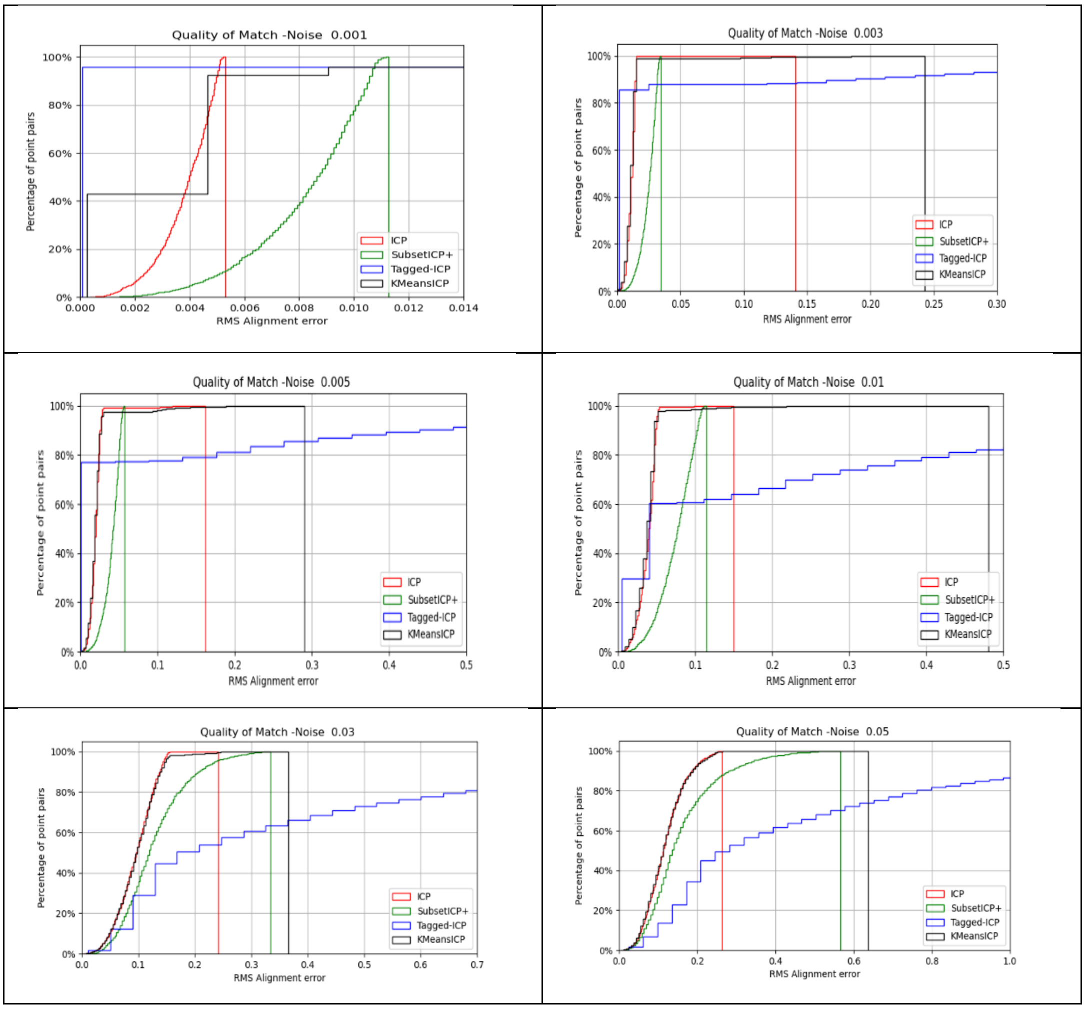


Figure 7.26 Match quality results for registering full shape 2M5T with simulated noise. TaggedICP shows a good quality of match for smaller noise levels compared to the other algorithms even though this was not for all the points. This observation is vice versa for larger noise levels such as 0.03 and 0.05.

The full shape match quality results of 2M5T with simulated noise at 0-degree initial rotation (Figure 7.26) shows results for a test done to evaluate the algorithms in registering 2M5T with levels of noise (0.001, 0.003, 0.005, 0.01, 0.03, 0.05).

From the match quality results of 2M5T (Figure 7.26), it can be observed that TaggedICP showed a better match quality than the other algorithms with this quality decreasing as the noise level increased. This effect was seen in the quality of match for noise levels 0.01, 0.03, and 0.05 compared to the other algorithms. It can also be observed that KmeansICP and ICP showed an almost identical line representing a similar quality of match across the different noise levels.

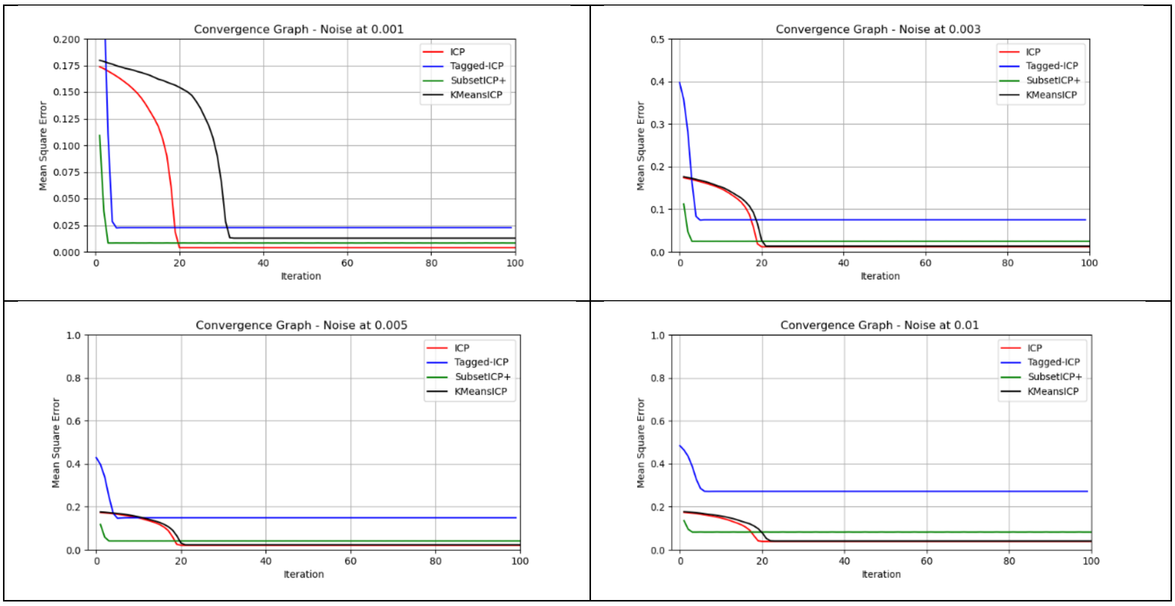
It can be observed that the match quality for 2M5T (Figure 7.26) is like the match quality results for matching 2HAX with simulated noise in (Figure 7.21). This is also observed and discussed in Figure 7.21, along with the poor performance of TaggedICP at larger error levels.

Compared to the match quality results for 2HAX (Figure 7.21), the match quality for 2M5T (Figure 7.26) shows SubsetICP+ performing much worse than KmeansICP and ICP.

The similarities in the algorithm's performance with the two different shapes (2HAX and 2M5T) validate the consistency of the match quality algorithms and the experiments, however, a detailed observation of the graphs reveals the slight differences that also validates the fact that the shapes have a different distribution of atom types and total atom numbers as shown in Table 7.1 and their partition techniques influence their match quality.

A detailed observation of the quality of match 0.001 for 2M5T (Figure 7.26) shows KmeansICP's first rise to be slightly above 40% whilst in the quality of match for 2HAX (Figure 7.21), that figure is below 20%. Since both tests were under the same condition and KmeansICP does not use atom type-based partitions, the result indicates the influence of the distribution of points (how evenly spread the points are in the cloud) and the size of the subset as a factor in the match quality of the shape registration.

It is also observed that the match quality for 2M5T (Figure 7.26) and Figure 7.21 had similarities in each of the graphs. TaggedICP showed a better match quality for smaller noise levels, but gradually decreasing the quality of match on larger noise levels: 0.01, 0.03, and 0.05.



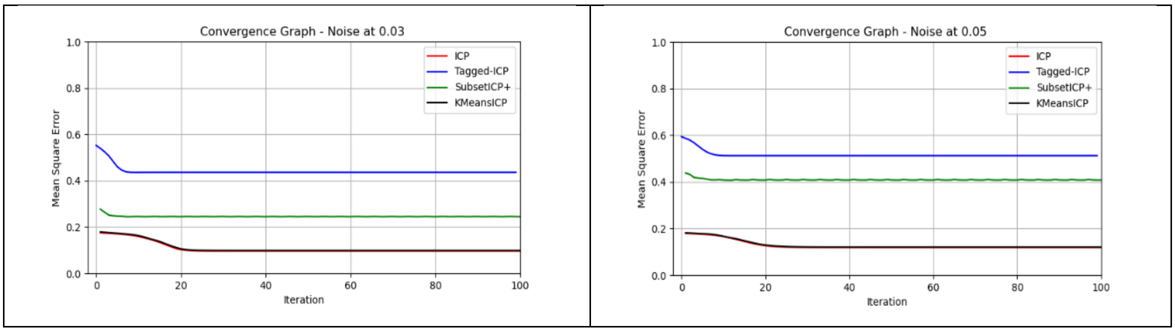


Figure 7.27 Convergence results for matching 2M5T with simulated noise. TaggedICP and SubsetICP+ showed a faster convergence but at a larger error than KmeansICP and ICP. KmeansICP and ICP also show similar performance.

From the convergence results for full shape matching 2M5T with simulated noise at 0-degree rotation (Figure 7.27), it can be observed that, for all error levels, SubsetICP+, ICP, and KmeansICP converged at lower errors than TaggedICP. This directly reflects a similar performance in the match quality tests in Figure 7.26. It can also be observed that ICP, followed by KmeansICP and SubsetICP+ converged at the smallest error. However, SubsetICP+ is seen to have converged at constant iteration count, even though at smaller error in small noise levels (0.001, 0.003, 0.005, and 0.01) and larger errors at larger noise levels (0.03 and 0.05).

TaggedICP' s large error with increasing noise levels in the convergence results (Figure 7.7Figure 7.27) is because of the TaggedICP algorithm reverting to a less optimal closest point when there is none found whilst matching with the tag information. Since the underlying algorithm with TaggedICP is still ICP, reverting to this less optimal closest point based on just the closest point in the atom-type partition becomes even less optimal than ICP's closest point method because the choice would have been affected by a constraint partition.

TaggedICP and SubsetICP+ are noticed to have converged faster in all the tests, from lower convergence errors at lower noise levels to higher convergence errors with increasing levels of noise. This trend is expected as the levels of noise increase, however, TaggedICP also converged at the largest errors for all the noise levels. This is because of the tag information constraint presenting challenges with registration at these levels of noise.

The convergence results of 2M5T in Figure 7.27 also shows KmeansICP and ICP having a very similar trend of their convergence for noise levels 0.003, 0.005, and 0.01, 0.03, and 0.05 including converging at almost identical iteration count. This is because both algorithms use the same closest point search method. Even though KmeansICP partitions the search space based on k-means clusters and as such there was the expectation that KmeansICP might converge faster, the similarity of their convergence shows that the KmeansICP's partitioning did not help improve the convergence for 2M5T. In fact, at the smallest error of 0.001, KmeansICP showed a larger error at convergence. This is because KmeansICP can return less accurate matches from the boundary of a cluster, especially boundaries that intersect other clusters, returning potentially suboptimal correspondences.

It is noticed that the convergence results for 2HAX at lower noise levels were slightly at lower errors than 2M5T (Figure 7.23 and Figure 7.27) and particularly for SubsetICP+, matching with 2HAX showed oscillations for five (5) out of the six (6) tests in Figure 7.23, however, no such behaviour was seen in the convergence matching for 2M5T (Figure 7.27). This is because the distribution of atom types slightly affects the performance of the algorithm. The probability of finding a better match in a subset of the same atom type is higher than the probability of finding a better match in a large subset of the same atom type. Since SubsetICP+ convergence performance is mainly because of the quality of the closest points instead of their quantity, a small subset with the same atom type increases the likelihood of finding a close to exact match within that limited subset.

According to Table 7.1, 2M5T has the largest subset size of 1028, for atom H, but not in terms of percentages as 2HAX has the largest size in terms of percentage of atom type, C making up 52.54% of the atoms, however, if the size of the subset is considered, then 2M5T is the likely shape to return the most inaccurate matches especially when matching on the subset H.

7.6.3 Matching Shapes with Mutations 2JKF and 2JKG

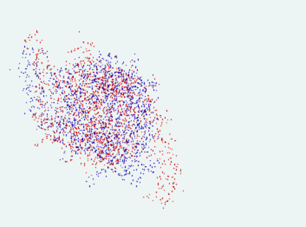
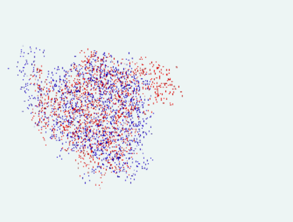
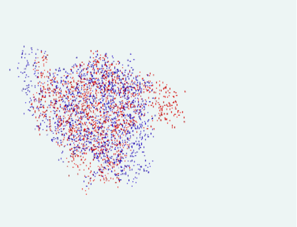
Protein similarity studies consider the sequence information (the sequence of characters in the amino acid chains in the protein), to determine the similarity between proteins and their inferred common ancestry (homology) (Pearson, 2013). Two proteins are homologous when their sequence information is similar. This similarity is the number of positional matching characters in the amino acid chain or strand (the primary structures of protein).

Similar proteins based on their amino acid sequence matches do not necessarily mean their 3-D structures are alike to the same extent. This is because a mutation can distort the sequence and the structures. However, since the sequence information is at a more primary lower level, comparing proteins using their structures is more important because the structures preserve more information than the sequence during changes such as mutations. This means that the match quality results from the algorithms should show much better quality than the quality of a similar protein comparison that considers only the sequence information and not the 3-D structure. The focus of the research is not on the similarity matching but the performance of the developed variants.

It is often noted that the structure of proteins informs their functions and proteins with similar shapes can have common ancestry and similar functions (Kinoshita and Nakamura, 2005). However, the similarity of a protein is also based on several factors such as the aspects of the protein the algorithm considers, for instance, the sequence of amino acids or the 3-D structure of the protein. Even though in many cases, the similarity of the protein sequence can be used to find similarly shaped proteins, similarity based on sequence information does not necessarily map to the similarity of the protein structures because these are at different structural levels of a protein. Comparing the structure of proteins is more important than comparing the sequence information because the sequence information is the primary structure at the most basic level, which can change easily with even small mutations whilst the structure is resilient to more subtle changes (Srivastava et al., 2016).

Even though in many cases, the similarity of the protein sequence can be used to find similarly shaped proteins, similarity based on sequence information does not necessarily map to the similarity of the protein structures because these are at different structural levels of a protein. Also comparing the structure of proteins is more important than comparing the sequence information because the sequence information is the primary structure at the most basic level, which can change easily with even small mutations whilst the structure is resilient to more subtle changes (Srivastava et al., 2016).

Since 2JKF and 2JKG are mutations, the initial rotations of the data shapes are not to study the effect of the angles on the match quality of convergence but are to understand whether a result is a coincidence of the initial rotation.

	Rotation Angle		
Algorithm	15	60	90
ICP			

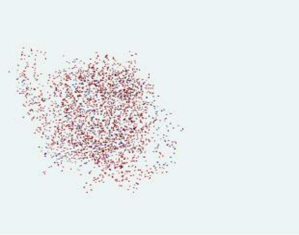
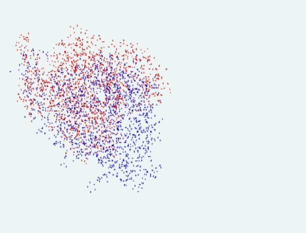
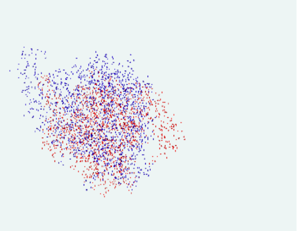
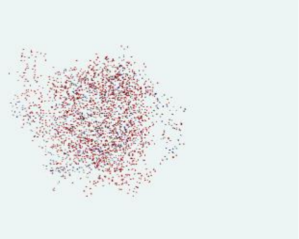
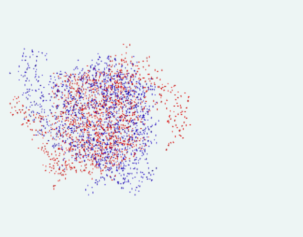
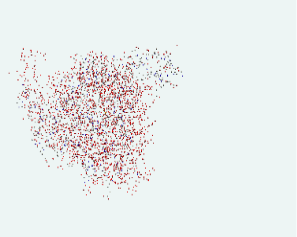
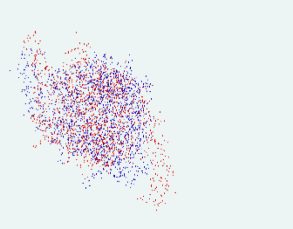
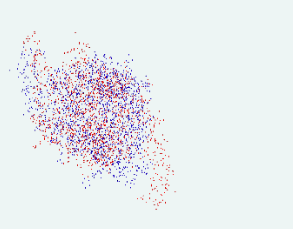
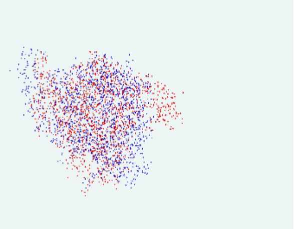
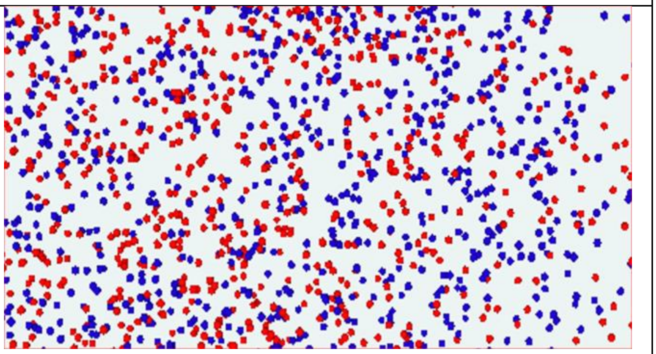
TaggedICP			
KmeansICP			
SubsetICP+			

Figure 7.28 results of matching 2JKG (red model point cloud) and 2JKF (blue data point cloud). The separation of colours indicates the match quality. For instance, ICP at angle 15 is of a better match quality than ICP at angle 60 because there is a clear separation of the colours at angle 60.

The match quality results in Figure 7.28 shows visual results of matching 2JKG and 2JKF. It is noted that ICP at angle 15 and SubsetICP+ at angle 15 showed a better visual match quality than all the other results. This is seen in how easy it is to see the separation of colours in the point clouds. The other results show various levels of colour separation with SubsetICP+ at angle 60 having a much worse result. A closer look at some of the results clearly shows partial overlaps, for instance, the results for all algorithms at angle 60.

	Angle 60
ICP	

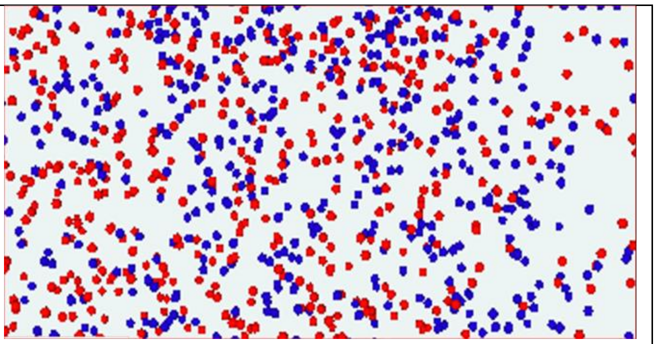
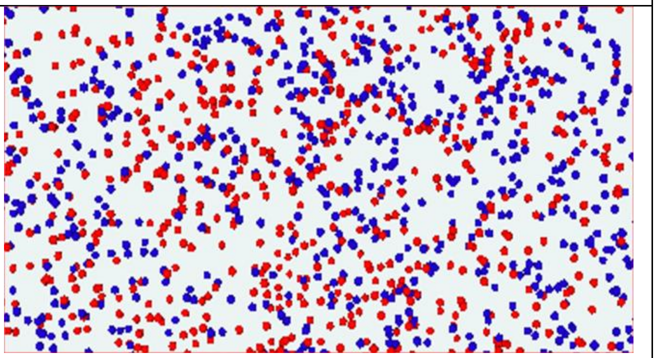
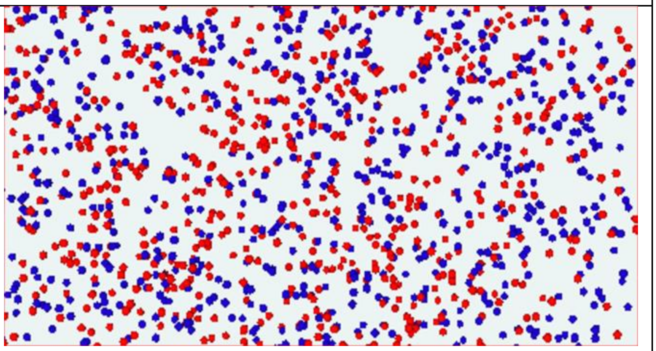
TaggedICP	
KmeansICP	
SubsetICP+	

Figure 7.29. Zoomed images of the shape matching results. The zoomed-in images were taken at the centre of the shape match and as such does not show details such as those captured at the ends of the shape match. The colour separation is very similar and does not show large differences for analysis.

The zoomed-in images for registering 2JKF and 2JKG shows a similar colour separation for the algorithms. TaggedICP's points seem to be denser than the other algorithms.

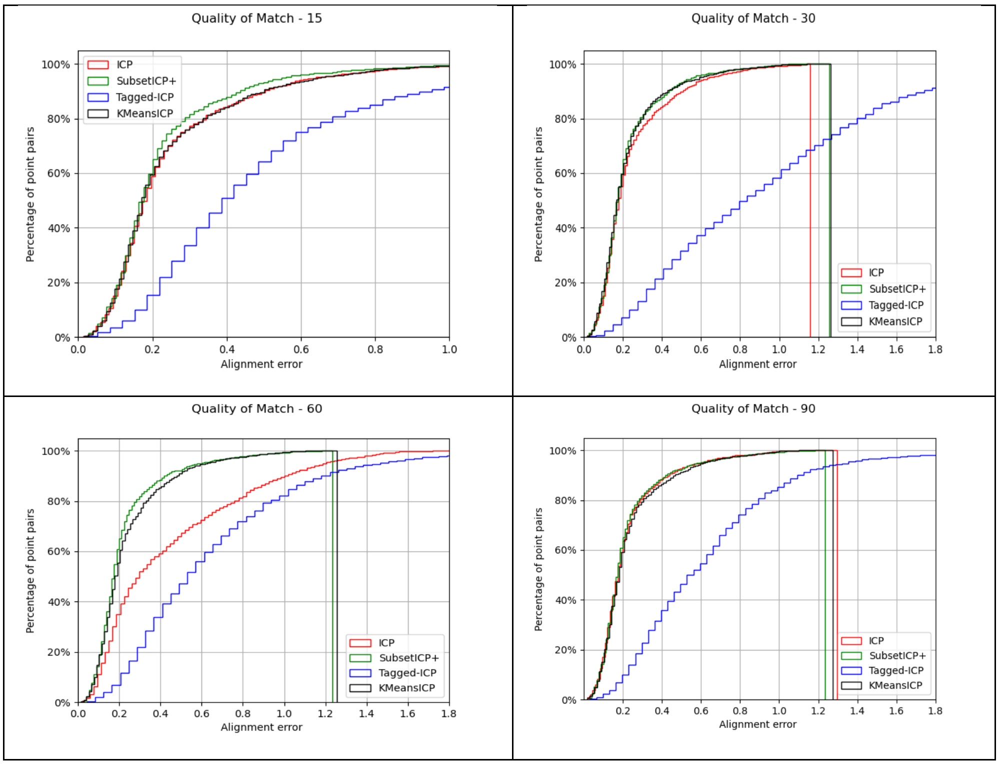


Figure 7.30 Match quality results for 2JKG and 2JK showing cumulative percentage of point pairs with alignment error. TaggedICP performed the worst in the series of graphs whilst other algorithms had a varying performance.

The match quality results for registering 2JKG and 2JKF (Figure 7.30) represent a test done to match 2JKG and 2JKF. The registration of the two shapes represents attempts to match naturally occurring mutations. Unlike manually shifting the positions of points to simulate the addition of noise for the noise experiments, 2JKF and 2JKG represent the same base molecule with some evolution. That means that the positions and types of atoms in the molecule had naturally changed. Even though the 3-D shape looks similar in some respects, there doesn't exist a true match for all the atoms because the shapes are not structurally the same.

It can be observed from the match quality results for matching 2JKG and 2JKF (Figure 7.30) that the algorithms had a relatively closer match quality except for TaggedICP. KmeansICP, ICP, and SubsetICP+ show a similar trend at angles 15, 30, and 90. TaggedICP is also seen as having the least quality of match at all the angles.

KmeansICP, SubsetICP+, and ICP had a similar trend in the graph because both algorithms use the same closest point search method that finds the closest point to a data point by doing a point-to-point distance calculation. Even though KmeansICP partitions its search space based on k-means clusters to reduce the distance calculations, its performance with ICP and SubsetICP shows no enhancement derived for matching 2JKG and 2JKF.

In the match quality graph (Figure 7.30), SubsetICP+ and TaggedICP were expected to show a higher quality of match than the rest of the algorithms because of the use of atom type-based subsets and the metadata tag information in the matching. However, though SubsetICP+ showed performance at par with KmeansICP and ICP, TaggedICP lagged at a much worse quality of the match.

TaggedICP performed worst because the algorithm reverts to a closest but less optimal correspondence when it does not find a match using the tag information and the mutations of the molecules make it more challenging to find the true matches. TaggedICP' s constraint is beneficial when matching especially full shapes. ICP and KmeansICP showed a familiar trend of having similar performance because of the similarity in their correspondence search methods.

All the algorithms are variants of ICP and share the same weakness in theory. Every point in the data shape needs a good match in the model shape because that's how the algorithm is structured, so in the presence of noise, the correspondence search returned from TaggedICP can sometimes be much worse than would have been returned from the Standard ICP's closest point search method.

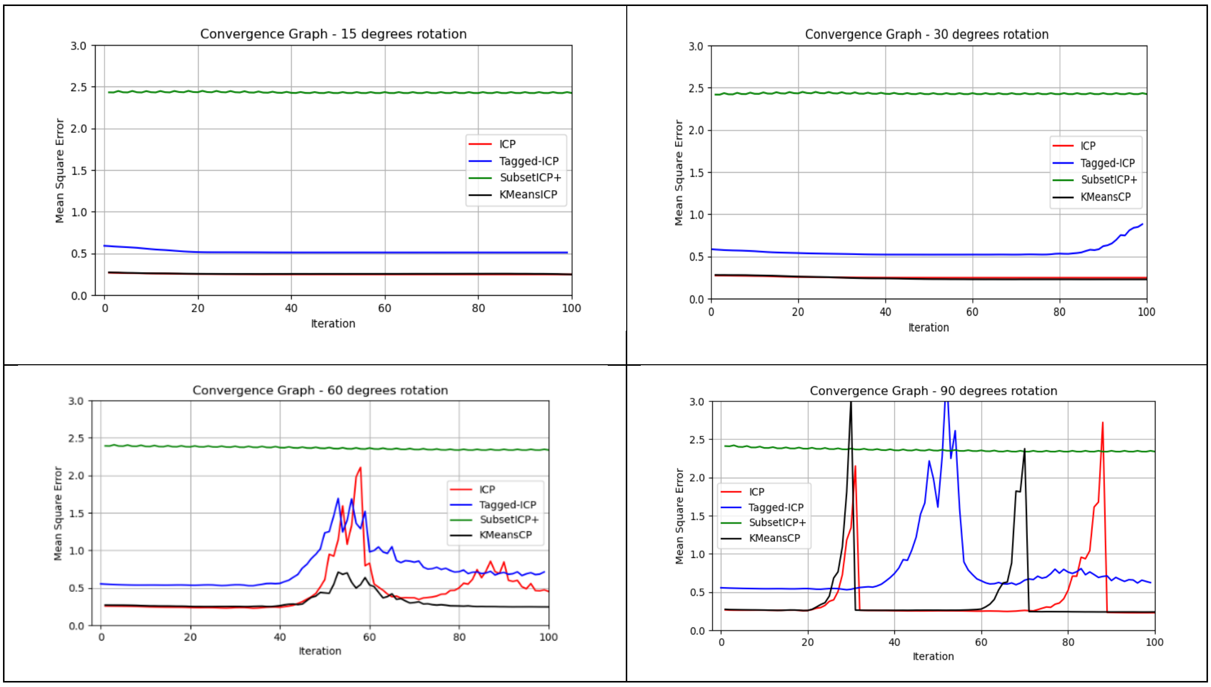


Figure 7.31 Convergence results for 2JKF and 2JKG. The algorithms converged at a higher error and in some instances, there was no convergence. The poor performance of the algorithms is because of the dissimilarity between 2JKF and 2JKG.

The convergence results for matching 2JKF and 2JKG (Figure 7.31) shows that most of the algorithms converged at a higher error and iteration when the noise level increased. However, this observation is not consistent with some of the algorithms. For instance, ICP, KmeansICP, and TaggedICP show an oscillating behaviour at angles 60 and 90 and SubsetICP+ shows the worse convergence with larger errors at all angles.

KmeansICP and ICP are observed to have similar results based on the similarities of their graphs. This has also been observed in many of the previous tests which are because of the similarities in their correspondence search method, where correct matches were not distinguished from the wrong ones and the wrong matches were not rejected for the underlying transformation estimation, which is the essential task for shape registration.

Average errors for sample experiments								
	Noise levels				No noise with initial rotation angle			
DataSet	0.001	0.01	0.03	0.05	15	30	60	90
2HAX	0.018	0.112	0.237	0.304	1.04E-3	0.0053	0.0082	0.0099
2M5T	0.01	0.119	0.216	0.246	1.13E-03	0.004	0.051	0.122
2JKF-2JKG (Mutation)	Noise experiment not run because datasets are naturally occurring mutations serving as noises				1.17E09	0.411	0.967	0.346

Figure 7.32 Average convergence error for sample experiments

The statistic average errors for each dataset (*Figure 7.32*) across different experiments. The final convergence error for each algorithm is considered as an individual case and the average error found for each rotation angle in the noise, mutation, and non-noise experiments shows the average error increasing with increasing levels of noise and the average error increasing with increasing initial rotation angle. The initial rotation angle is the angle in degrees on the x-axis. ICP has an inherent limitation on performance with increasing noise or initial rotation angle and that is responsible for the large average errors with increasing angles and noises.

7.7 Summary

This chapter has evaluated the performance of some variants of the ICP algorithm (KmeansICP, TaggedICP, and SubsetICP+) with conditions such as different levels of noise, and different initial rotation angles.

Most of the algorithms performed well in the noiseless experiments (section 7.6.1) even though their performance in terms of match quality was seen to dwindle and iteration count and error at convergence were seen to increase with the initial rotation angle increased.

Similarly, a trend was observed for all experiments with noise (section 7.6.2). All algorithms were observed to show much worse match quality and convergence with the noise experiments than the noiseless experiments (section 7.6.1). This is because of the increased complexity of the shape registration task when noise is added to data.

Particularly for the individual algorithms, SubsetICP+ showed a match quality performance that was in part with ICP and convergence at fewer iterations in low to medium levels of noise in several experiments including matching with different rotation angles (Figure 7.13). However, it sometimes converged at larger errors. SubsetICP+ uses the subset technique of matching atom type-based subsets, one per iteration. The atom type-based subsets enable the search space to be reduced, making the convergence faster in terms of using fewer iterations.

TaggedICP uses the metadata of the points to aid in the matching. It showed a match quality performance that was much better than the other algorithms in smaller noise levels as well as faster convergence than all the other algorithms in all noiseless experiments. With the noise experiments, TaggedICP sometimes converged the fastest, however, with a larger error. The algorithm decreased in performance because it returns a potential sub-optimal closest point when a match is not found using the tag information.

KmeansICP and ICP were noticed to have a similar graph in some of the experiments including convergence with 2HAX (Figure 7.14) and convergence with 2M5T (Figure 7.18). This is because of the similarities of their correspondence search method. KmeansICP uses ICP's correspondence search method to find a match after determining the search partition using the closest k-means cluster.

In conclusion, this chapter has evaluated the developed algorithms in Chapters 4.CHAPTER 4 KMEANSICP, 5.CHAPTER 5 SUBSETICP+, and 6.CHAPTER 6 TAGGEDICP, on four (4) protein data sets, 2HAX, 2M5T, 2JKG, and 2JKF. The experiments have discussed and tested the performance of the algorithms in different scenarios, full shape matching, noise matching,

matching mutations and using TaggedICP' s closest point search method. The match quality and convergence graphs have been discussed and critically analysed to provide the reader with the details of the performance of the different algorithms. This chapter helps to achieve the aim of the research by evaluating the algorithms developed in chapters (CHAPTER 4 KMEANSICP, CHAPTER 5 SUBSETICP+ and CHAPTER 6 TAGGEDICP).

PART 5: ICP SHAPE MATCHING IN VIRTUAL REALITY

8 CHAPTER 8 ICP Virtual Reaility Application (ICPVR)

8.1 Introduction

This chapter discusses, the design, development, and evaluation of the Iterative Closest Point Virtual Reality Application (ICPVR). It is aimed at achieving objective four (4), "*Develop a virtual reality visualisation software for visualising and interacting with molecular protein data, allowing exploration of the molecule in immersive virtual reality environments*". It uses the knowledge gained from Sections (CHAPTER 3 VIRTUAL REALITY, CHAPTER 4 KMEANSICP, CHAPTER 5 SUBSETICP+, and CHAPTER 6 TAGGEDICP) to focus on aspects of virtual reality metrics such as navigation and usability. The application presents the user with a simple interface of buttons representing the ICP algorithms as well as the ability to see a visualisation of the shape matches after every iteration in succession as an animation.

The objective of VR is to allow a user to act virtually, whilst thinking the action is being performed in the real world (Arnaldi, Guitton and Moreau, 2018). The usefulness of VR has seen it become very popular and used for many important tasks such as virtual design and construction of buildings and vehicles, training with virtual flight simulators and teaching and learning. A detailed history, a review of the components of virtual reality, and various VR experiments were discussed in CHAPTER 3 VIRTUAL REALITY.

Shape registration is a challenging task and visualising the shape registration process enhances a user's visual analysis of the shape match. For instance, an immersive visualisation of the shape getting trapped and recovering from the local minima problem allows the user to gather insights on this problem. ICPVR presents an immersive virtual reality environment that enables the user to discover insights about the registration process. It allows the shape overlap to be visually inspected for better analysis. For instance, consider a molecule point cloud with colour coded atoms and a registration process that is optimised to match using the metadata of the points and which continuously show a much worse overlap for a region of the molecule. An immersive visual analysis of the final shape match can gather insights into the areas of the shape with a worse match based on an immersive visual analysis of the overlap.

ICPVR visualises the shape using the algorithms; ICP (Besl and McKay, 1992), TaggedICP (Ankomah et al., 2020), KmeansICP, and SubsetICP+ algorithms. ICPVR is aimed at being a simple virtual reality 3-D utility application that allows a shape matching algorithm to be set up in the application with some configuration. ICPVR is used to visualise the registration and serves as a quick prototype for the user to gain insights from the registration process. The current version allows the user to add the 3-D point clouds and algorithms from the Unity Editor.

8.2 Motivation

The shape registration using ICP is always likely to converge to a local minimum because of the local iterative procedure the algorithm implements and especially without a good initial transformation. VR can be used to help visualize the resulting shape match by analysing the shape overlap and gathering insights into the worse affected areas of the match by the extent of the overlap.

ICPVR helps shape matching by providing visual feedback on the registration quality, allowing the user to adjust the shapes or transformations as necessary. This is almost always required as the registration quality varies from one shape to another, affected by the complicated geometry, large magnitude of the transformation, varied size of overlaps, and simplified mechanism in matching points.

The initial transformation of the data shape can be adjusted whenever necessary and there is the possibility of doing this whilst the shape matching process is ongoing, however, it is not clear the measurable effects of this real time adjusting on the quality of the match.

The ICPVR application serves as a system’s research project output that brings out new ideas for other systems to be built on top of it. Even though point cloud processing in virtual reality is not unique, to the best of the researcher’s knowledge, there is no existing research that evaluates shape matching algorithms in an immersive virtual environment.

8.3 Hardware and Software Setup

The software was developed using C# and the Unity3D game engine (Unity Technologies, 2019) software. The hardware consisted of the HTC VIVE head-mounted display (HMD) (HTC, 2020). This was connected to an Intel 64-bit computer with a 2.2 GHz Core i7-8750H processor, 16GB RAM (random access memory), and a GeForce GTX 1070 graphics card.

ICPVR's 3-D user interface places buttons on a menu in the VR environment. The buttons represent the implemented variants of the ICP algorithm. The user can click any of the buttons to run the corresponding ICP variant to match the two point clouds. The use of HMD enables the user to visualise all angles of the point clouds by moving their head.

Two (2) point clouds derived from the same protein molecule PDB ID: 2HAX was retrieved from the Protein Data Bank (PDB) (Berman et al., 2000) using CellUnity (Gehrer, 2015).

8.4 Design & Development

ICPVR was designed and developed using Unity3D (Unity Technologies, 2019) and C#. Unity3D provides an integrated GUI development environment for creating the virtual environment. ICPVR is made up of a base plane with the ability for the user to teleport around, and a panel object housing buttons that map to the various ICP algorithms.

Early in the development phase, the metrics of virtual reality were considered, and the development was done to ensure maximum immersion, interaction, knowledge improvement, navigation, performance, and usability. For instance, the prototype was kept simple with a base plane (coloured blue) that is wide enough to fit the view of the user. Disruptive objects were not placed in the virtual environment to ensure that the user focuses on the shape registration processes, interaction with the 3-D shapes and noticing the wall of transformations. The interaction was also enhanced by allowing the user to use a pointing device such as a joystick to interact with objects in the virtual world. For instance, the HTC VIVE hand controllers could be used to touch, rotate, and move virtual objects. The performance of the virtual environment was considered along lines of reducing the click response time and reducing lag. The correspondence search method of the ICP algorithms which usually takes more time to process was updated by returning control to the display after 50 correspondence searches. This prevents the long-running task from freezing the display with lags but rather continues to render a responsive virtual environment to the user. In considering the usability of the virtual environment, the presentation of buttons mapped to algorithms was a way to present a simple usable interface for the user. The ability to teleport in the virtual environment was also provided to allow the user to easily move around the virtual space whilst considering their limited physical space and enhance usability.

ICPVR shows the user a real-time animation of the shape matching process in the virtual reality environment. The real-time animation was achieved by applying the computed transformation to the data shape in each iteration which automatically updates the data point through re-rendering in the virtual environment. This repeated process creates visible animation. This process can be computationally expensive and can result in a lagged virtual environment.

To improve the performance of the animation and prevent lagging in the VR environment, the shape matching process implemented Unity3D (Unity Technologies, 2019) Coroutine feature that allows flexibility in returning control to the display. This is achieved by asynchronously running the ICP algorithm and returning control to the display (refreshing the display to show the latest update of the positions of the atoms) at set intervals. This process ensures that the user sees the point positions from the most recent completed iteration and with the current head and hand tracking. This flexibility means that the animation is rendered to the display over several frames. The user is then able to see, after each iteration, the data shape rotating and moving closer to the model shape as computed transformations are applied to the data shape to bring it closer to the model shape.

The point clouds are represented as sphere primitives and different colours distinguish the data point cloud from the model point cloud (yellow data point cloud and red model point cloud). This is useful to facilitate visual analysis of the final shape registration as the user looks at the extent of the overlap of the atoms based on their distinct colours. The shape matching process runs 100 iterations, and the user sees feedback on the final transformation after the matching is complete.

The current representation of the molecules as point clouds also makes the user able to walk into the point clouds in the virtual environment. This allows a user to easily visualise points that are closer to the centre of the point cloud as the user virtually moves into the molecular structure. The ability to walk into the molecule allows the user to study the overlaps on atoms that are closer to the centre of the molecules (the molecules are volumetric) to visually see wrong matches and decide on how the data point can be moved as well as how to adjust the initial transformation. Allowing the point clouds to dissolve when the user gets closer was found to be a limitation when viewing point clouds of walls because the expectation would have rather been dense point clouds looking like a solid object (Bruder, Steinicke and Nüchter, 2014). On the other hand, the opposite of this is needed in ICPVR.

Unity3D (Unity Technologies, 2020) also allows the 3-D objects to prevent other objects from passing through them by using a Collider object around the 3-D objects to detect collisions with other objects such as the user’s virtual character. Detecting collisions can be handled with scripts to prevent the moving object from advancing its position or other appropriate actions such as pushing the molecule away but that was not the requirement.

ICPVR did not implement collision detection because the research was interested in allowing the user to fully explore all aspects of the molecule and virtual world including walking into the molecules and viewing the overlaps of atoms closer to the centre of the molecule (the molecules are volumetric). This meant that the atom primitives could overlap with each other. Collision detection can also be implemented to give the user some more details on the registration including real-time data on points being matched based on the detected collisions between the corresponding points.

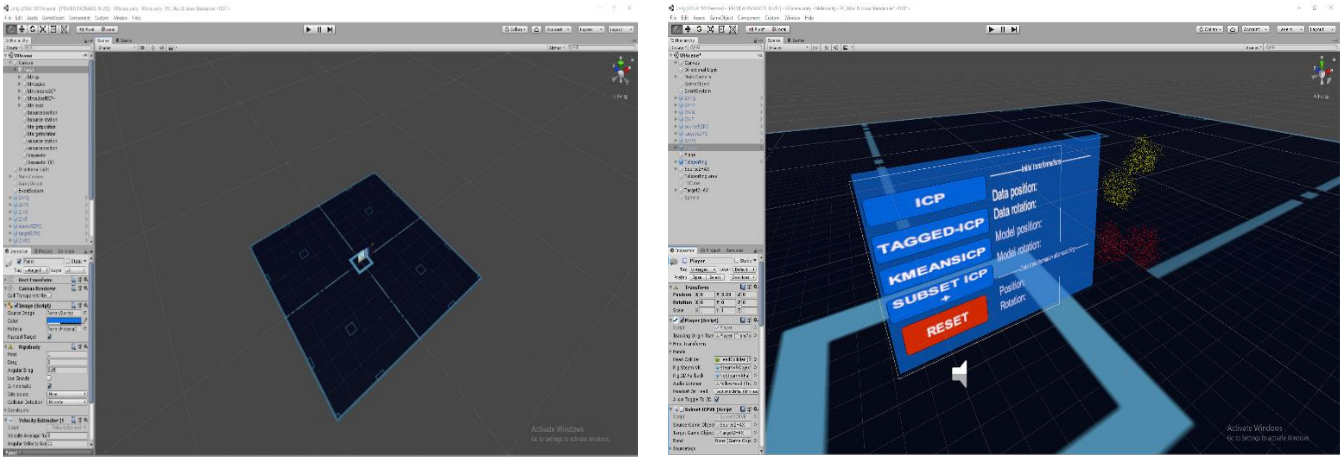


Figure 8.1 (LHS) Unity3D Integrated development environment with the base plane and (RHS) panel object hosting the ICP algorithm buttons and placeholder texts for the transformations.

The SteamVR Unity3D package by HTC (HTC, 2020) was used to provide the utilities that allow configuration and binding the HTC hardware controllers and VIVE headset to the virtual environment. For instance, the StreamVR package allows the configuration of the controller buttons to specific or generic actions in the virtual environment. This provision allows the input to function correctly, such as manipulating objects and teleporting with the hand controllers.

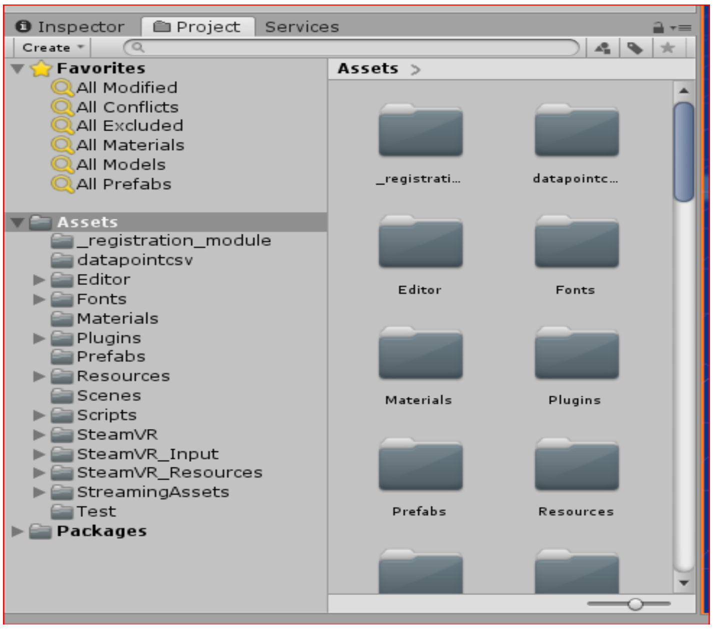
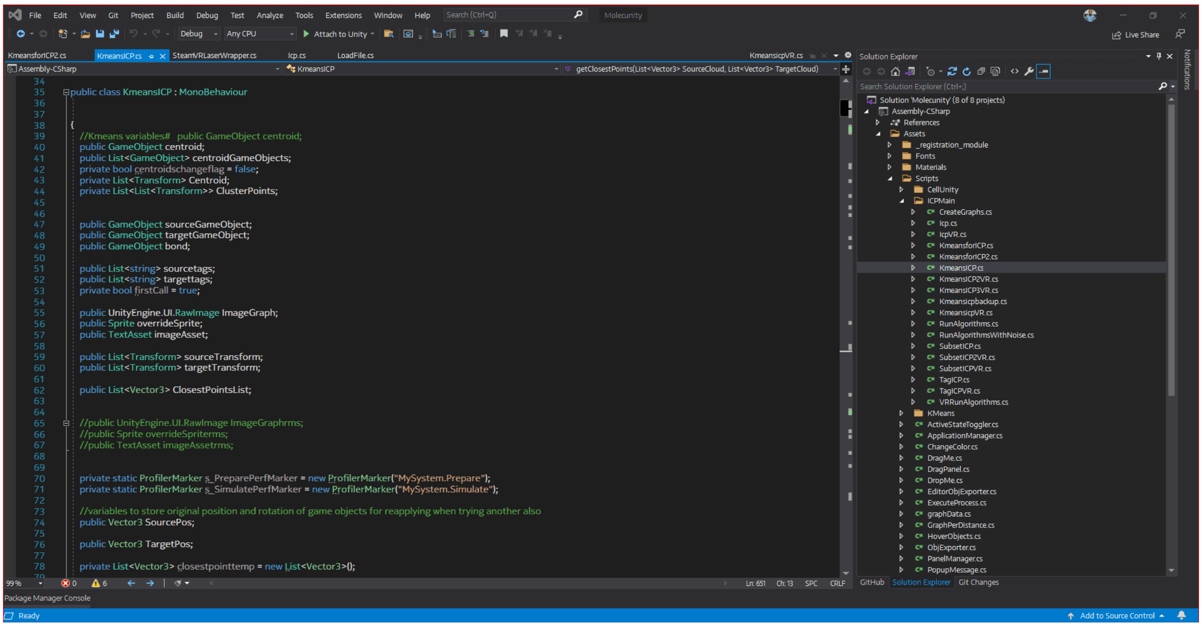


Figure 8.2 ICPVR Assets shows the SteamVR packages that allow binding functionalities of the virtual environment to the HTC VIVE headset and controllers.

Figure 8.2 shows ICPVR asset folders inside Unity3D. These folders contain class files, internal and third-party libraries. For instance, the SteamVR packages are the library files from Steam that adds the functionality of teleportation in the virtual environment as well as interaction with the molecular structures.



```
public class KmeansICP : MonoBehaviour


{
    //Kmeans variables#   public GameObject centroid;
    public GameObject centroid;
    public List<GameObject> centroidGameObjects;
    private bool centroidschangeflag = false;
    private List<Transform> Centroid;
    private List<List<Transform>> ClusterPoints;


    public GameObject sourceGameObject;
    public GameObject targetGameObject;
    public GameObject bond;

    public List<string> sourcetags;
    public List<string> targettags;
    private bool firstCall = true;

    public UnityEngine.UI.RawImage ImageGraph;
    public Sprite overrideSprite;
    public TextAsset imageAsset;

    public List<Transform> sourceTransform;
    public List<Transform> targetTransform;

    public List<Vector3> ClosestPointsList;


    //public UnityEngine.UI.RawImage ImageGraphrms;
    //public Sprite overrideSpriterms;
    //public TextAsset imageAssetrms;


    private static ProfilerMarker s_PreparePerfMarker = new ProfilerMarker("MySystem.Prepare");
    private static ProfilerMarker s_SimulatePerfMarker = new ProfilerMarker("MySystem.Simulate");

    //variables to store original position and rotation of game objects for reapplying when trying another also
    public Vector3 SourcePos;

    public Vector3 TargetPos;

    private List<Vector3> closestpointtemp = new List<Vector3>();
```

Figure 8.3 Visual Studio 2019 with code editing view for (LHS) KmeansICP C# codes and (RHS) Solution wide view showing project classes including the shape matching algorithms.

Figure 8.3 shows the Visual Studio 2019 code editor for developing the shape matching algorithms. The LHS shows a view of the KmeansICP algorithm whilst the RHS shows the tree folder structure of the whole project.

ICPVR was designed to visualise the registration process using the same algorithms used in the experiments in Chapter 7. It was aimed that ICPVR will be able to easily plug in a shape registration algorithm with very minimal modification and be able to visualise in virtual reality.

To make the registration process real time and for the user to see the real-time animation, like the registration process on the desktop, the algorithm was adjusted to update its state to the VR display right after the computed transformation has been applied to the data shape. Updating its state to the display results in the user constantly seeing the animation of the model shape. ICPVR can also be extended to show other features such as pop-ups of 3-D shape descriptions and details of each iteration. These were not in the scope of the initial phase of development as the target was a quick working prototype application.

Another consideration during the development and initial testing was the performance of the algorithms in the presence of many thousands of points. In some instances, the display was seen to freeze whilst the background processes ongoing operations. Further debugging revealed that the correspondence search algorithm takes the most time in running as such the correspondence search method was updated to return control to the display after a specified number of searches. For instance, returning to the display after every 50 correspondence prevents the display from freezing, even though the algorithm will take the same time to complete the process.

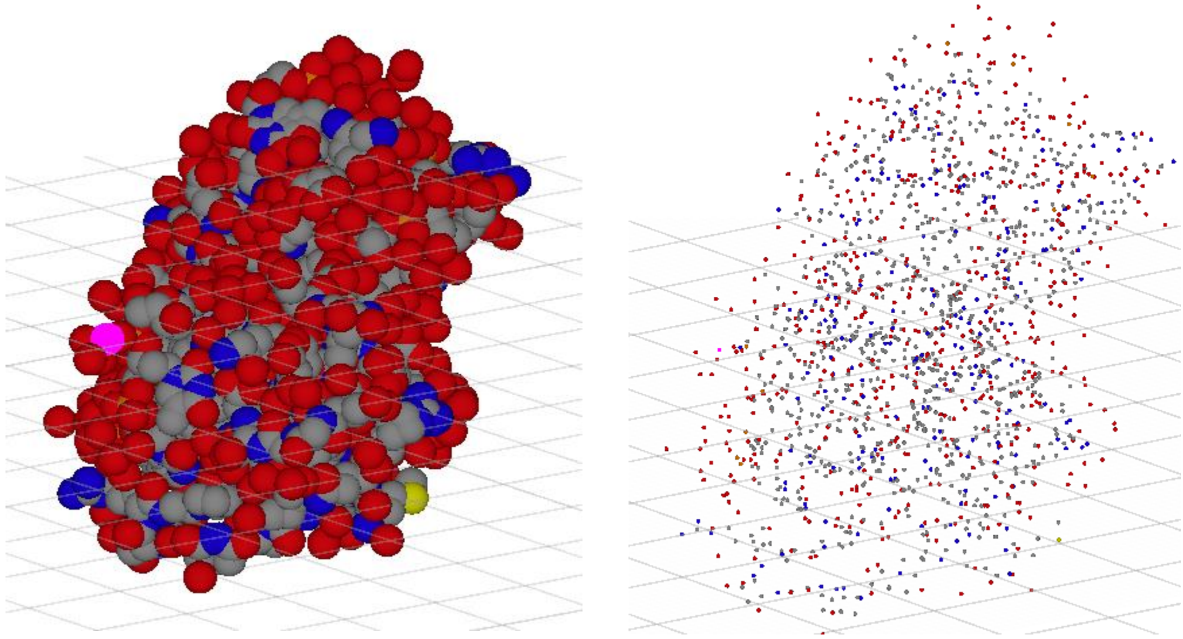


Figure 8.4 PDB ID 2HAX, with 1614 atoms (RHS) Image showing different colour-coded atoms with original scaling as imported from the PDB file and (LHS) images with colour coded atoms and scaled-down atom sizes. Red atoms represent Oxygen (O), blue atoms are Nitrogen (N), and grey atoms are Carbon (C), yellow represents Sulphur (S) and pink represents phosphorus (P).

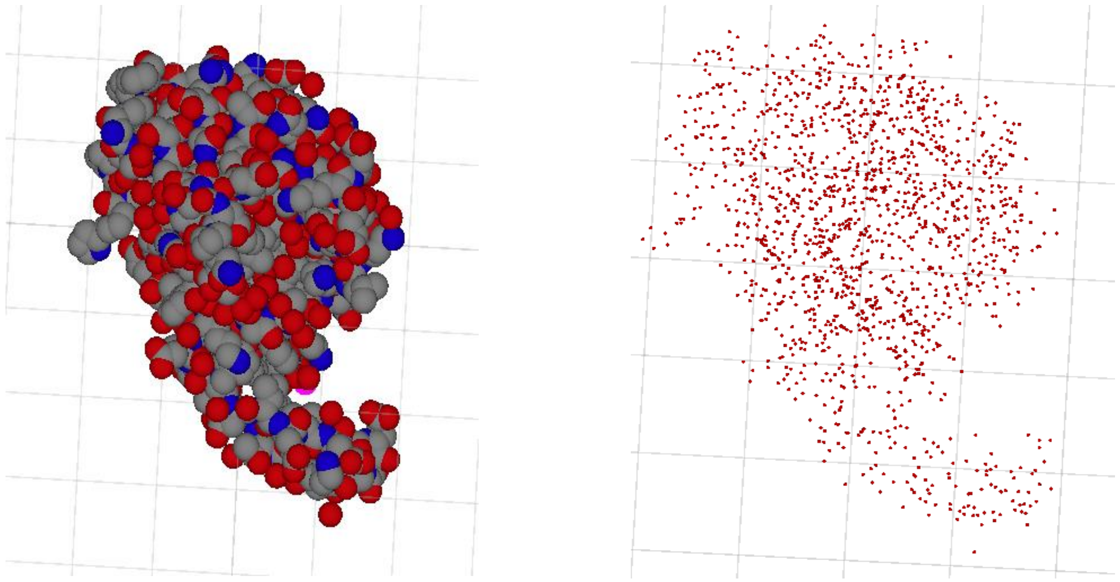


Figure 8.5 PDB ID 2JKG with 1452 (LHS) Image showing different colour-coded atoms with original scaling and (RHS) shapes with colour coded atoms and scaled-down atom sizes. Red atoms represent Oxygen (O), blue atoms are Nitrogen (N), and grey atoms are Carbon (C).

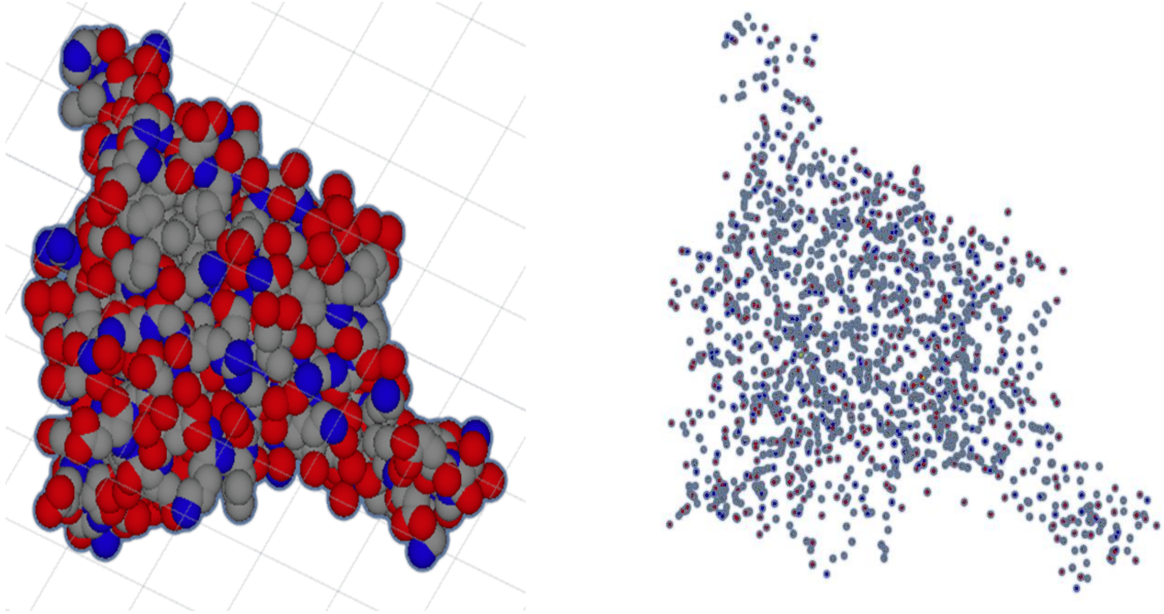


Figure 8.6 PDB ID: 2JKF with 1420 atoms (LHS) shows colour coded atoms and the original scale of the atoms and molecule. (RHS) shows molecule scaled down for visualisation. Red atoms represent Oxygen (O), blue atoms are Nitrogen (N), and grey atoms are Carbon (C)

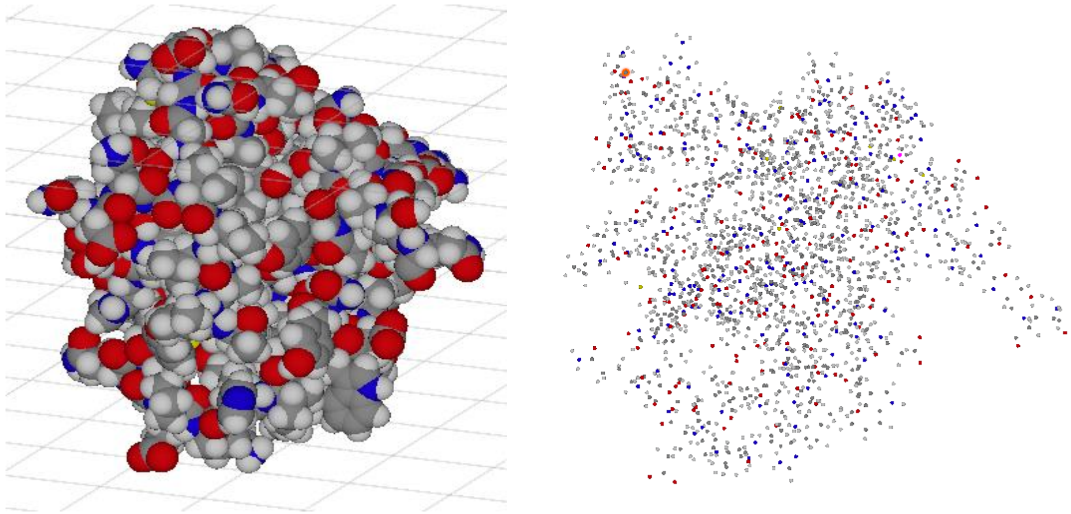


Figure 8.7 PDB ID: 2M5T for same shape registration. 2M5T is used for testing a very basic registration process by matching with itself. (LHS) 2M5T with original scaling and colour coded atoms. (RHS) 2M5T with scaled-down atoms for registration. Red atoms represent Oxygen (O), blue atoms are Nitrogen (N), and grey atoms are Carbon (C)

8.5 Evaluation

8.5.1 Overview

As discussed earlier, ICPVR is a virtual reality application that enables shape matching or registration and performing visual analysis on the quality of the shape match using the overlap of the shapes. ICPVR also allows a comparison of the initial and final transformations of the data point cloud. This can be used to ascertain the quality of the shape match when matching full same molecule shapes. ICPVR currently presents the user with two protein molecules, with their atoms represented as spheres, just like in the desktop version. However, the development environment allows the user to place and match any other 3-D shape.

ICPVR also allows the user to set or modify the initial transformations of the data and model point clouds by using the hand controllers to rotate and move the point clouds as desired. ICPVR's interface presents buttons that map to the different ICP algorithms. The interface also shows the initial transformation for both point clouds and the final transformation values for the data point cloud. The position of the point cloud is represented as a 3-D point (float x, float y, float z), whilst the rotation of the point cloud is represented as a quaternion (float x, float y, float z, float w).

The design of the user interface shows placeholders for the relative rotation and relative position of the model point cloud as well as the data point clouds before and after matching relative positions. This position is usually referred to as the position of the object. The model point cloud's position and rotation on the user interface are necessary because it allows the user to compare the position and rotation of the data point cloud after matching with the same positions for the static model point cloud.

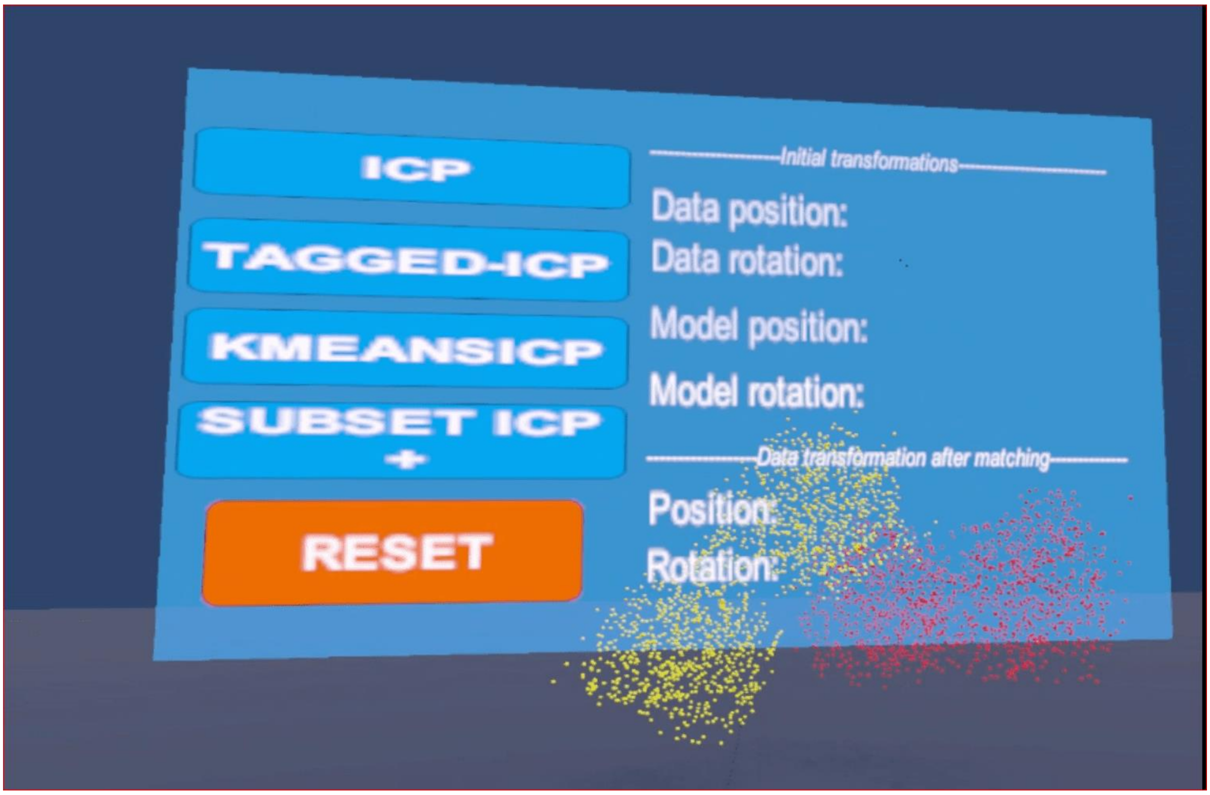


Figure 8.8 ICPVR home screen user interface with a button for each ICP variant and placeholders for the initial and final relative transformations of the data and model point cloud. The "RESET" button is used to re-initialise the data and model point cloud to default positions. The interface also shows red and yellow point clouds.

Figure 8.8 shows the user interface of ICPVR in virtual reality with sample point clouds red and yellow objects. The scaling of the point clouds was done with a consideration of how the user will interact with the point clouds. Making the shapes big will result in problems with interaction such as the user unselecting an object but because of the large size, the user needs to move their hands or pointer farther away for the system to register the unselect action on that virtual object.

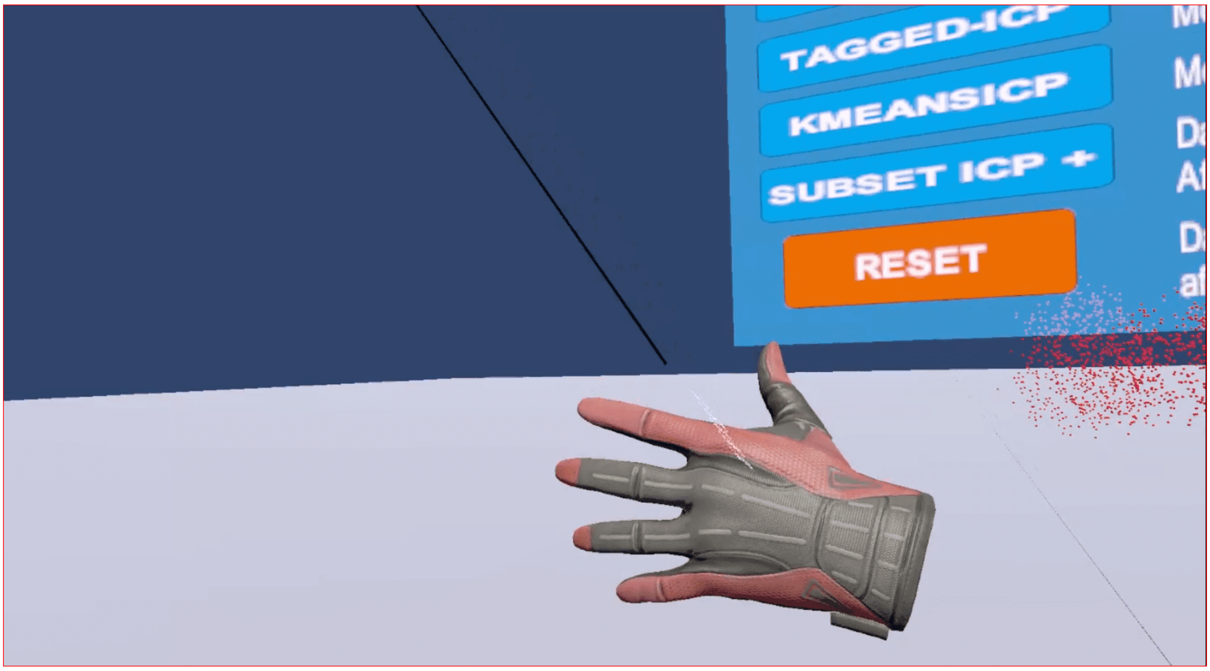


Figure 8.9 Default virtual hand for touching and interacting with the virtual shapes. The virtual hands provide the user with the ability to use the hardware controllers to manipulate objects whilst animating to give the user some sense of using their real hands.

Figure 8.9 shows a virtual hand that mimics real hands when the user interacts with objects in the virtual world. The black line represents the pointer used for clicking buttons and selecting virtual objects. The virtual hand object can also be replaced with any 3-D object. However, using a hand-like object contributes to the overall immersion experienced by the user because the user's sense of using their own hands is increased whilst viewing actions of their virtual hands.

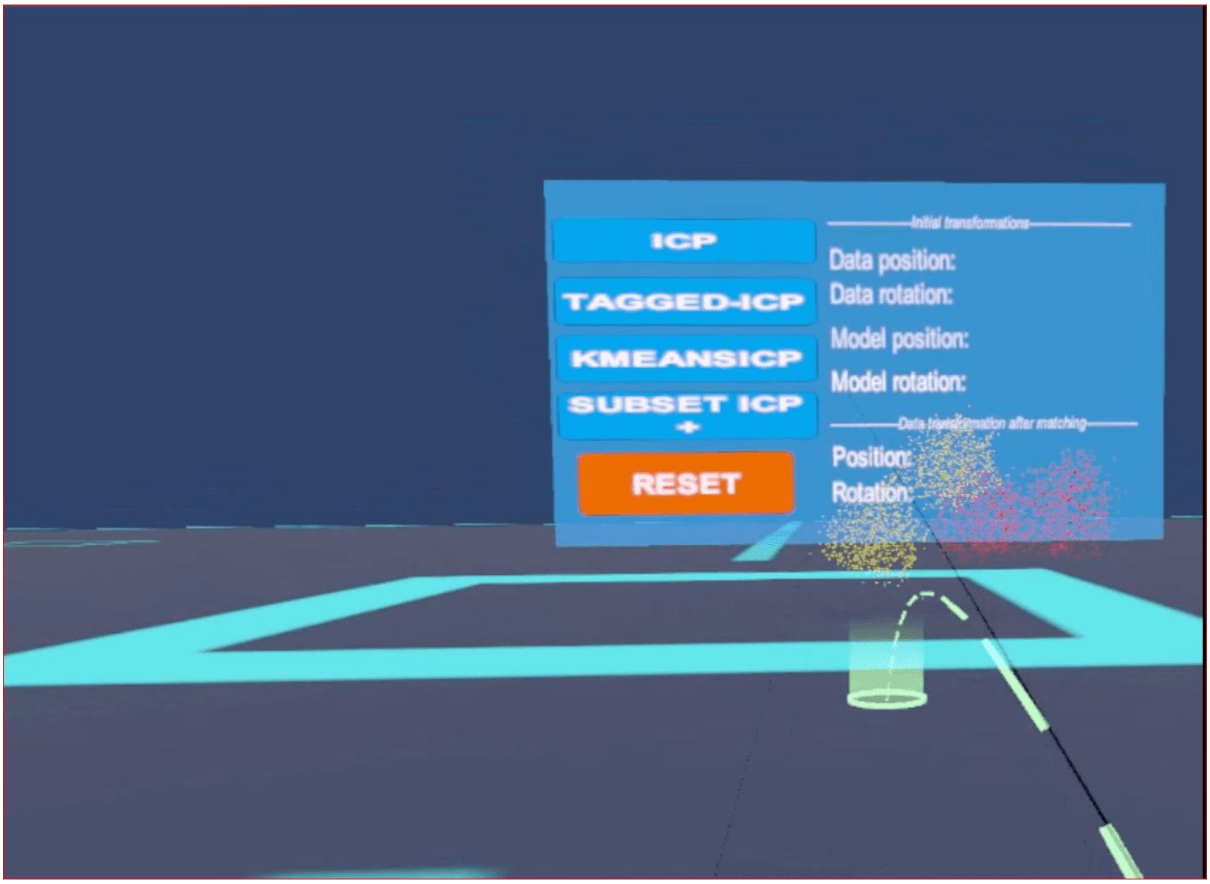


Figure 8.10 ICPVR showing the teleportation feature. The target teleport position is represented by the cylindrical object. The light green spaced line connecting the teleport target is the pointer from the position of the user. The black point is the default VR pointer for clicking and selecting objects. Teleportation helps the user to move around the virtual environment within the limited space of the user's physical environment.

Figure 8.10 shows the ICPVR user interface with the user ready to teleport to a new location. The cylindrical object represents the new location the user wants to virtually move to whilst the dashed line shows an animation from the current position of the user to the target location.

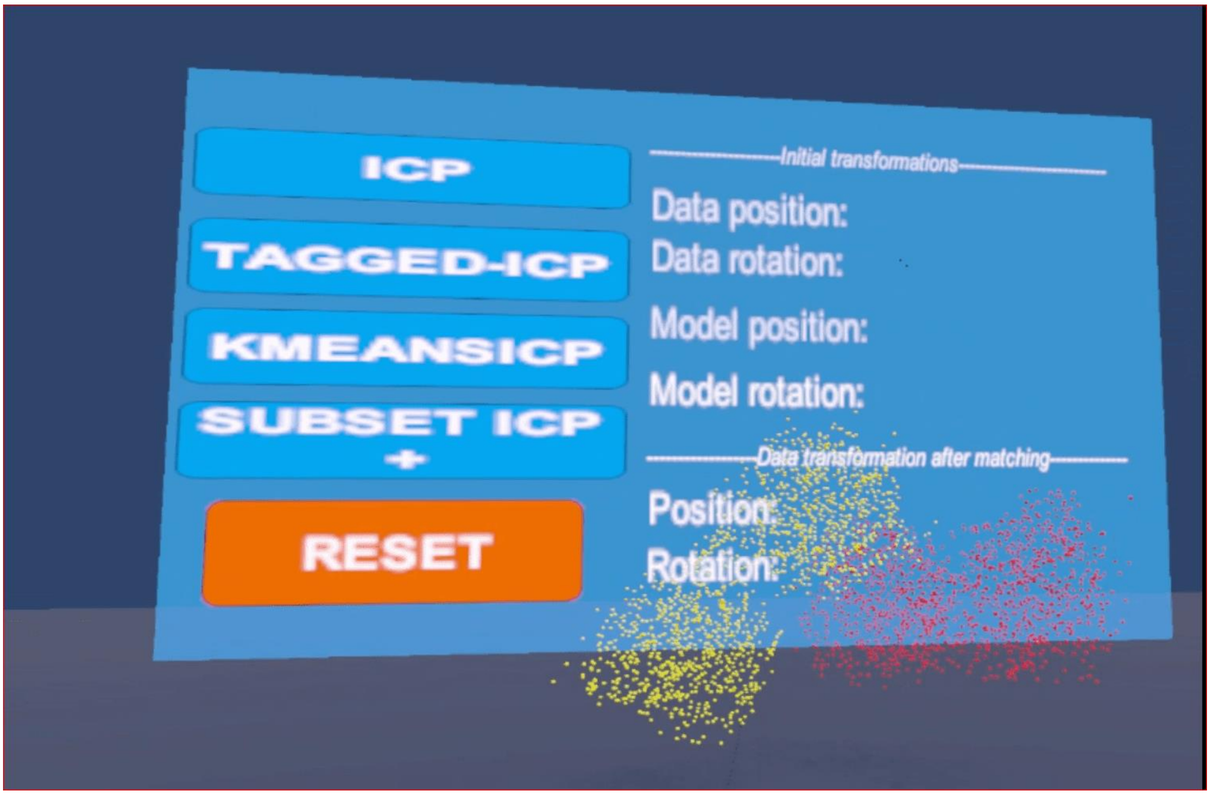


Figure 8.11 ICPVR showing sample red and yellow point clouds used for the experiment. Any other 3-D rigid object can be imported and used for matching and the appropriate metadata can be used in place of the atom types.

Figure 8.11 shows the two (2) default point clouds, red and yellow 3-D objects in ICPVR. Any other 3-D object can be used instead of the default shapes. To start the shape matching process, the user points the virtual laser pointer using the hand controller or compatible joystick to the desired algorithm button on the interface representing Standard ICP, TaggedICP, KmeansICP, and SubsetICP+. The user then selects the algorithm by pressing the trigger button. This action updates the VR interface with the initial transformation values of the point clouds and starts the shape matching process using the selected algorithm and the point clouds in the virtual environment. ICPVR does not allow the user to pause the animation or the registration process, however, this is planned for future updates, where the user can pause the registration process and change the relative position and rotation of the data shape.

The algorithm is visualised, iteration by iteration. The visualisation is updated with the new position and rotation of the data shape once the new computed transformation has been applied to it. These periodic updates allow the user to see the molecules getting closer to a perfect match.

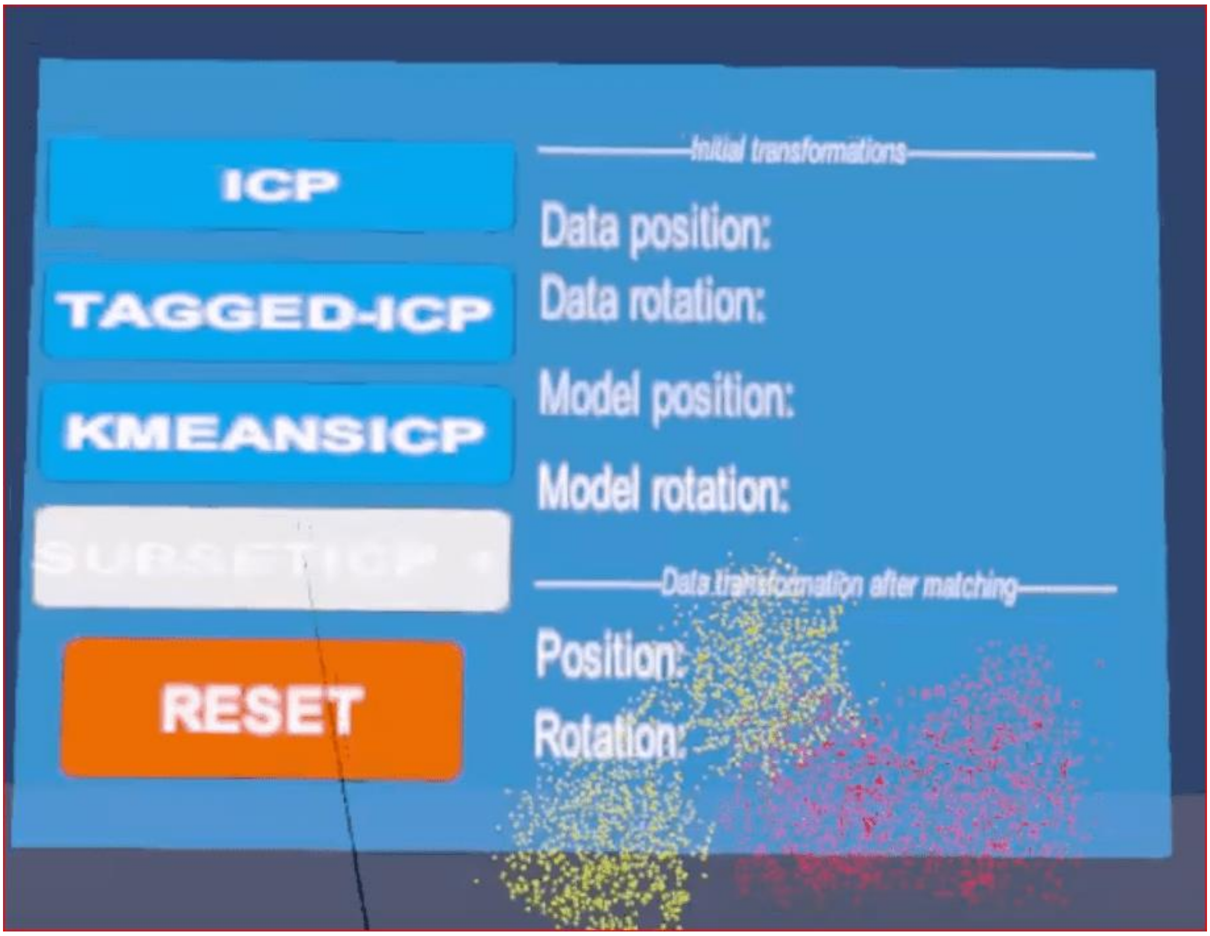
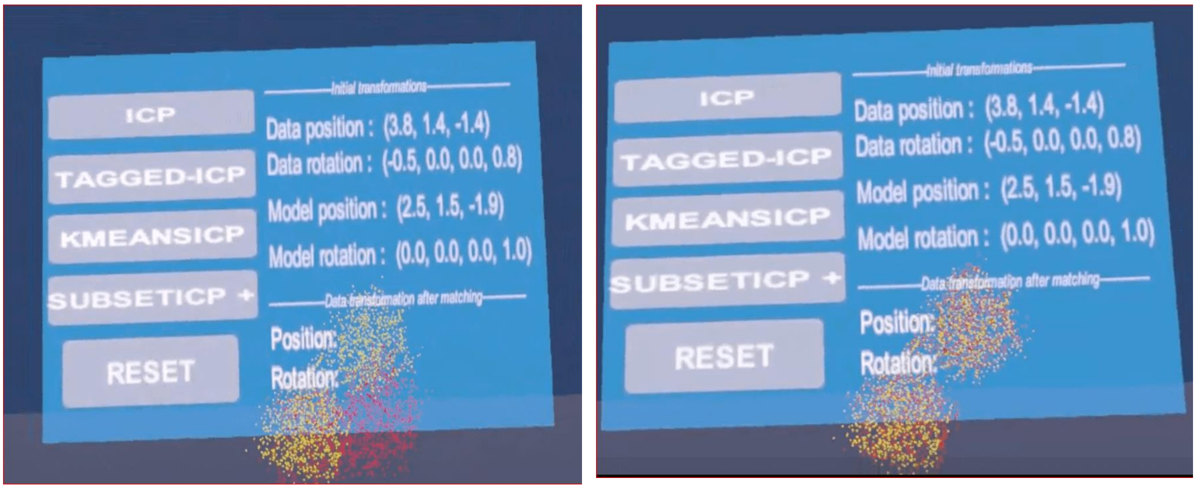


Figure 8.12 ICPVR UI with initial transformation (upper right part of the white text) and the registration process ongoing. As discussed earlier, the application shows the initial transformation of the data and model point cloud and then shows the final transformation after the registration process is complete

Figure 8.12 shows the user selecting the SubsetICP+ button. The SubsetICP+ button shows a hover colour, visible when the point has focused on it. Clicking the button then starts the shape matching animation process using the yellow data point cloud and the red model point cloud.



Initial transformations

Data position : (3.8, 1.4, -1.4)

Data rotation : (-0.5, 0.0, 0.0, 0.8)

Model position : (2.5, 1.5, -1.9)

Model rotation : (0.0, 0.0, 0.0, 1.0)

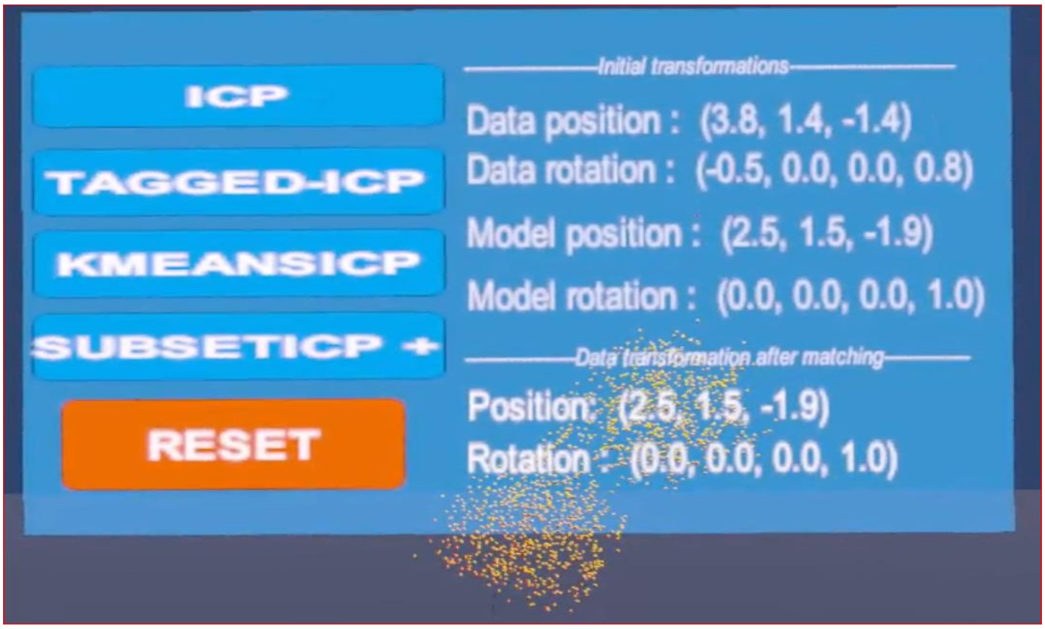
Data transformation after matching

Position:

Rotation:

Figure 8.13 Visual analysis of the shape matching process at different iteration. (LHS) represents an earlier iteration whilst (RHS) represents a later iteration.

Figure 8.13 shows various stages of the shape matching animation process. Once the animation starts, all the buttons are greyed out to prevent users from initiating new actions. The LHS image represents a snapshot earlier in the registration process whilst the RHS image represents a snapshot much later in the registration process.



Initial transformations

Data position : (3.8, 1.4, -1.4)

Data rotation : (-0.5, 0.0, 0.0, 0.8)

Model position : (2.5, 1.5, -1.9)

Model rotation : (0.0, 0.0, 0.0, 1.0)

Data transformation after matching

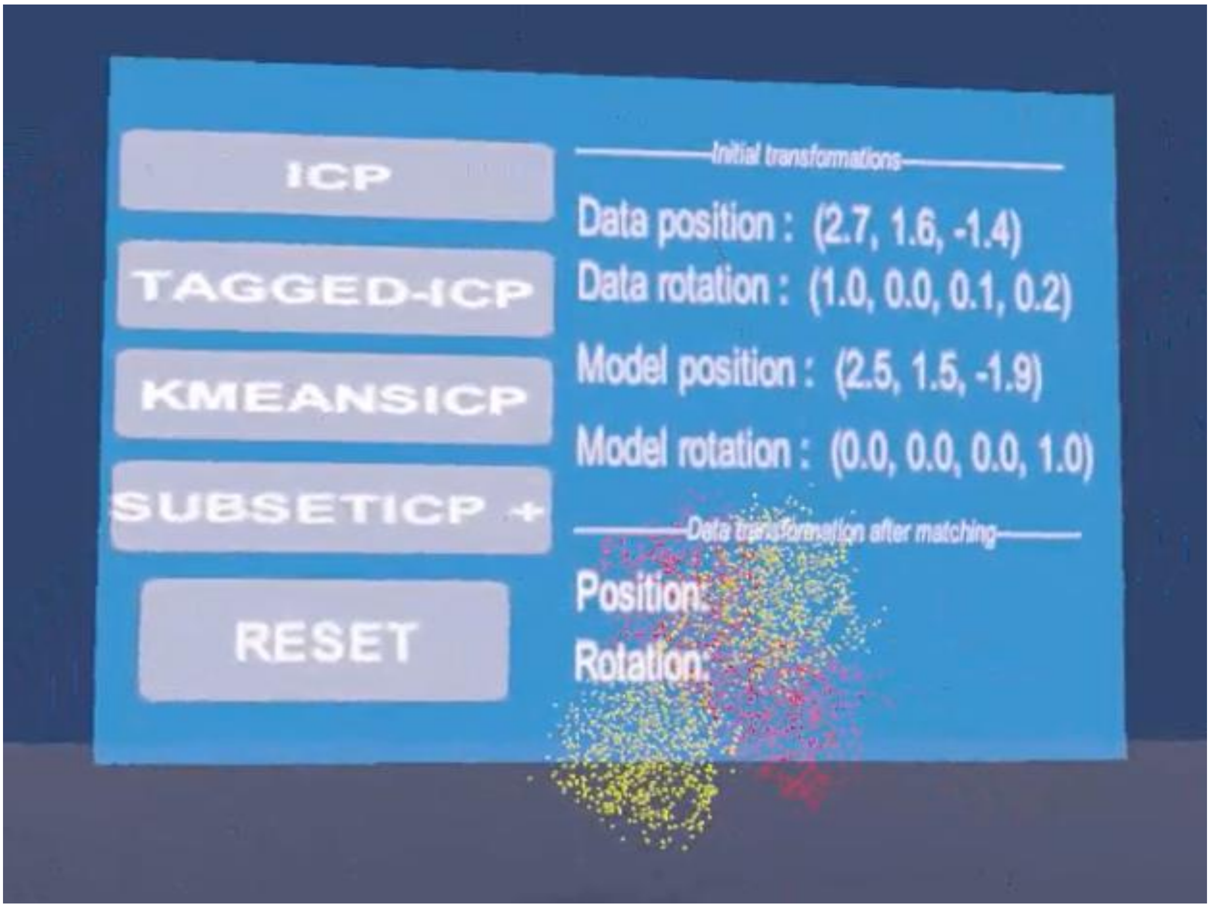
Position: (2.5, 1.5, -1.9)

Rotation : (0.0, 0.0, 0.0, 1.0)

Figure 8.14 ICPVR with details of the initial and final transformation after registering a shape. The image shows a good match with the position of the data point cloud, the same as the model point cloud.

After the shape matching process is done, the VR menu is updated with the position and rotation angle of the data shape. This allows a comparison of the model point clouds angle and position with the matched data point clouds angle and position.

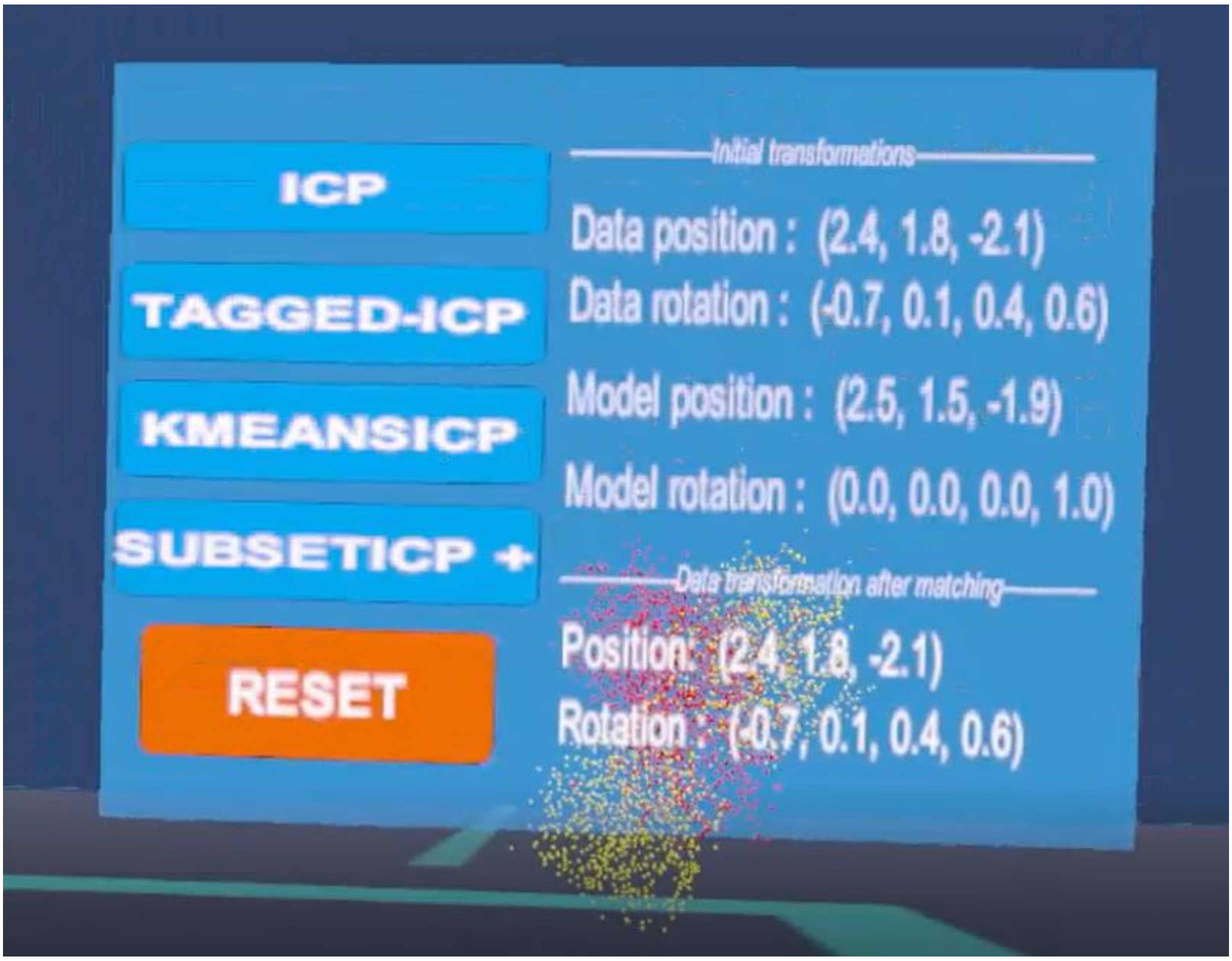
Another way of viewing the resulting match is by studying the overlap of the two shapes closely in the VR environment. For complex shapes, the user can detect patterns in the match such as the consistent quality of match for certain regions of the point cloud by repeated tests of setting different initial transformations and visually checking the point cloud overlaps. The interface can provide feedback on the registration quality and allow the user to adjust the shapes/transformations as necessary (Figure 8.15 - Figure 8.18).



Data position : (2.7, 1.6, -1.4)
Data rotation : (1.0, 0.0, 0.1, 0.2)
Model position : (2.5, 1.5, -1.9)
Model rotation : (0.0, 0.0, 0.0, 1.0)

Position:
Rotation:

Figure 8.15 Two point clouds (Yellow model point cloud and red data point cloud) with their initial transformations and the ICP algorithm running.



Initial transformations

Data position : (2.4, 1.8, -2.1)

Data rotation : (-0.7, 0.1, 0.4, 0.6)

Model position : (2.5, 1.5, -1.9)

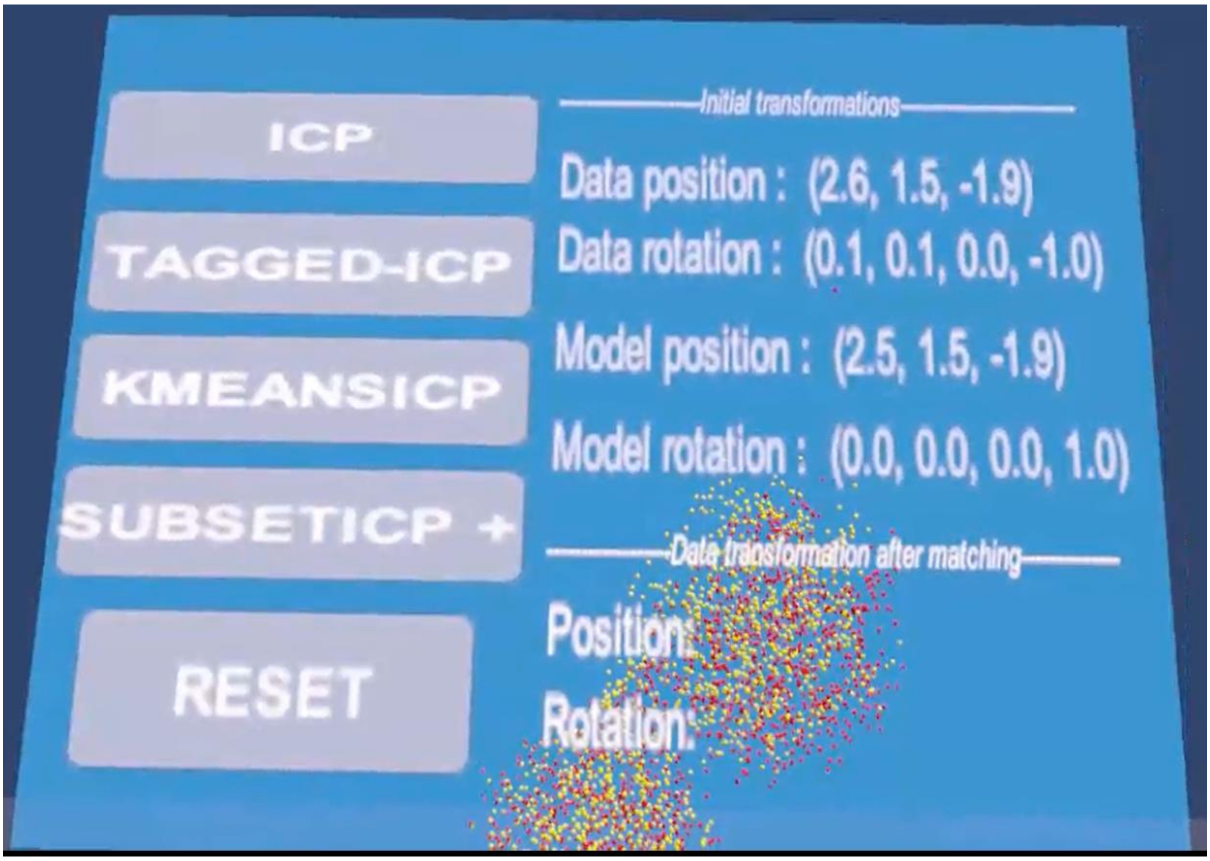
Model rotation : (0.0, 0.0, 0.0, 1.0)

Data transformation after matching

Position: (2.4, 1.8, -2.1)

Rotation : (-0.7, 0.1, 0.4, 0.6)

Figure 8.16 The two point clouds in the previous figure complete their matching. A visual analysis of the matched shapes and a comparison of the initial and final matching transformations, especially for the rotation shows a poor match that can be improved.



Initial transformations

Data position : (2.6, 1.5, -1.9)

Data rotation : (0.1, 0.1, 0.0, -1.0)

Model position : (2.5, 1.5, -1.9)

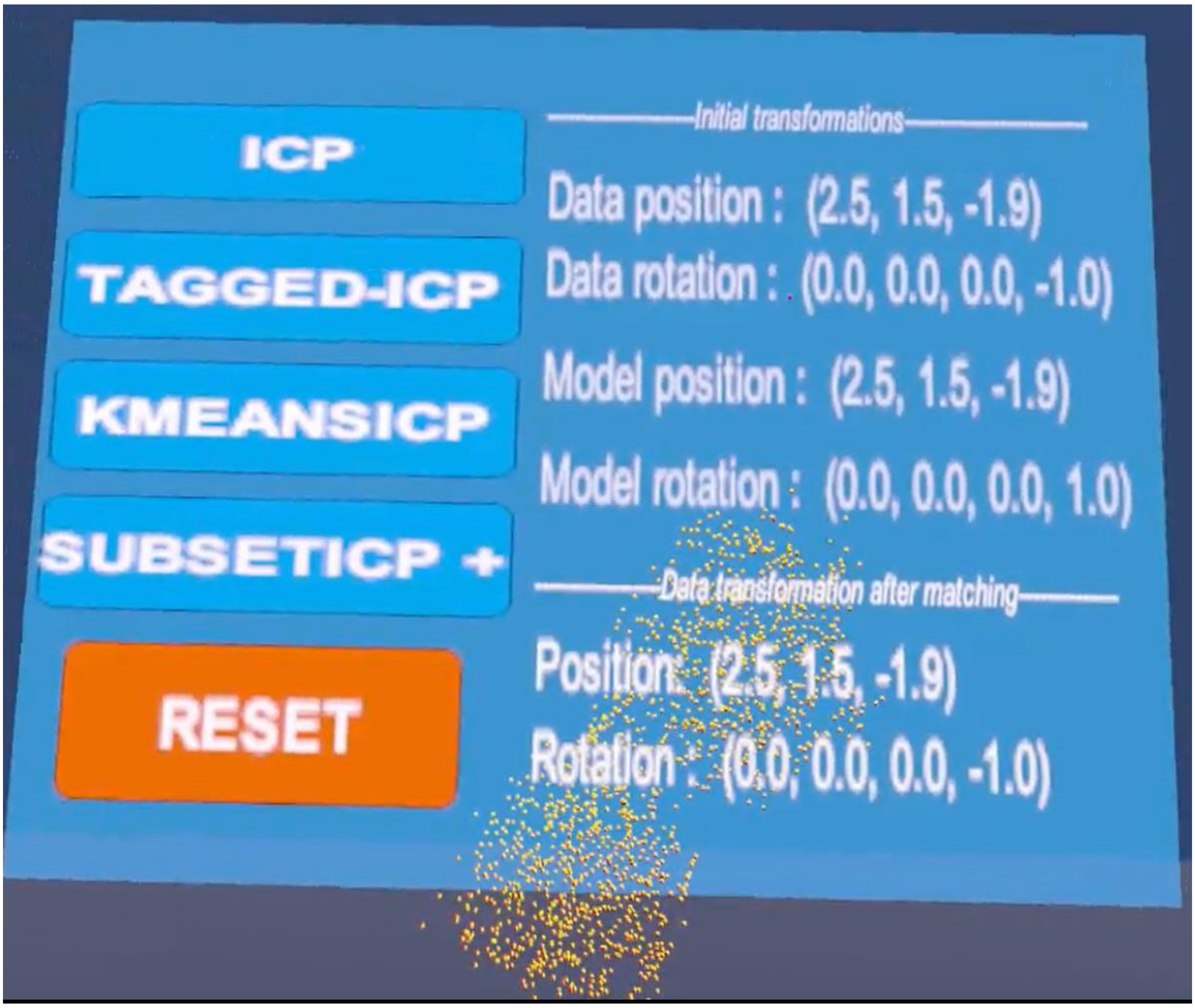
Model rotation : (0.0, 0.0, 0.0, 1.0)

Data transformation after matching

Position:

Rotation:

Figure 8.17 The data point cloud in the previous figure is adjusted as shown in the previous and current data transformation after matching and the current initial data transformation. The adjustment is expected to improve the matching.



Initial transformations

Data position : (2.5, 1.5, -1.9)

Data rotation : (0.0, 0.0, 0.0, -1.0)

Model position : (2.5, 1.5, -1.9)

Model rotation : (0.0, 0.0, 0.0, 1.0)

Data transformation after matching

Position: (2.5, 1.5, -1.9)

Rotation : (0.0, 0.0, 0.0, -1.0)

Figure 8.18 The matching of the two point clouds in previous figures shows a much better match when the data shape was adjusted after failure.

Figure 8.15 shows the registration of two point clouds, the yellow model, and red data point clouds. The completed matching in Figure 8.16 shows the relative (x, y, z) positions of the point clouds as the same (2.4, 1.8, -2.1) whilst the rotation (x, y, z, w) for the data point cloud (-0.7, 0.1, 0.4, 0.6) and model point cloud (0.0, 0.0, 0.0, 0.1) shows some difference resulting in some alignment errors.

In Figure 8.17, the data point cloud’s transformation was adjusted to a relative position of (2.6, 1.5, -1.9) and rotation of (0.1, 0.1, 0.0, -1.0) and then the matching was restarted with these transformations. This resulted in a perfect match shown in Figure 8.18 with the same relative position (2.5, 1.5, -1.9) and rotation (0.0, 0.0,0.0, -1.0) values for both data and model point cloud .

Adjusting the transformation is almost always required as the registration quality varies from one angle or position to another, affected by the complicated geometry, large magnitude of the transformation, varied size of overlaps, and simplified mechanism in matching points.

This virtual reality environment for shape matching allows adjusting of transformations by enabling the user to use the hand controllers to move and rotate the shapes before matching shapes as shown in Figure 8.9 and Figure 8.19. This means that users can quickly and repeatedly set different transformations before matching until the desired overlap outcome is achieved or the effects of different initial transformations are realised.

Users are also able to teleport to new locations in the virtual world using the HTC VIVE (HTC, 2020) controllers. This allows flexibility for the users to adjust the virtual environment to make it spacious and easy for navigation considering the users' physical spaces.

8.5.2 Immersive Visualisation

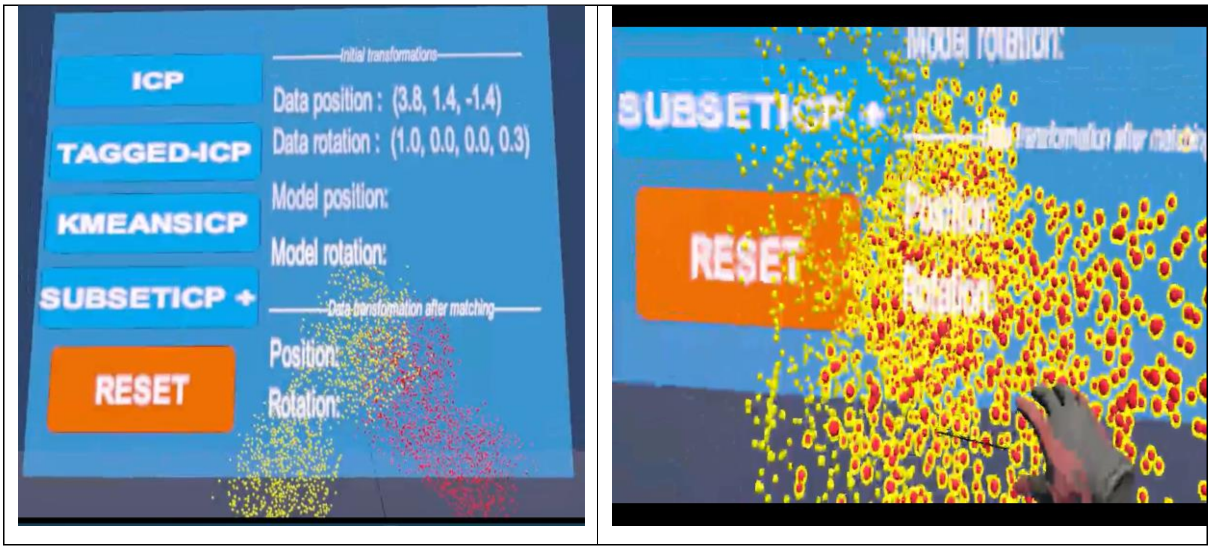
The ICPVR application allows a user to immerse themselves in the shape matching process. The user gains a better understanding of the iterative closest point’s process of computing and applying a transformation whilst noticing the shape’s translation and rotation animation. It also allows the user to have a better visual inspection of the shape overlaps after the matching is complete.

During earlier testing of the application, some lags were experienced in the animation which was due to the long-running time for the closest point search algorithm which freezes the display. Knowledge gained from the review of related works helped distribute the closest point set method across several frames to ensure a smooth animation. Specifically, the display was refreshed after every 50 correspondence searches.

The animation is affected by the algorithm and the frame rate. This is based on the frame rate of the display. The shape’s translation and rotation animation enable the user to understand how the distance between some shapes gets minimised smoothly until convergence whilst others have an oscillating nature as observed during testing.

8.5.3 Interaction

ICPVR allows users to interact with the 3-D molecules using the HTC VIVE (HTC, 2020) hand controllers. The user can move the data and model shape objects as shown in Figure 8.19 to the desired relative position and rotation. ICPVR shows the relative position and rotation of the data and model shape on the user interface when the algorithm button is clicked to start the shape matching process. It is easy to rotate the protein molecule using the HTC VIVE's hand controllers, however, the transformation is not currently shown in real-time, so the user can see the current transformation when the algorithm button is clicked. Showing the transformation information in real-time together with other updates are planned for a future update. User testing with participants for feedback on how to improve the interaction is also planned for a future version.



Initial transformations

Data position : (3.8, 1.4, -1.4)

Data rotation : (1.0, 0.0, 0.0, 0.3)

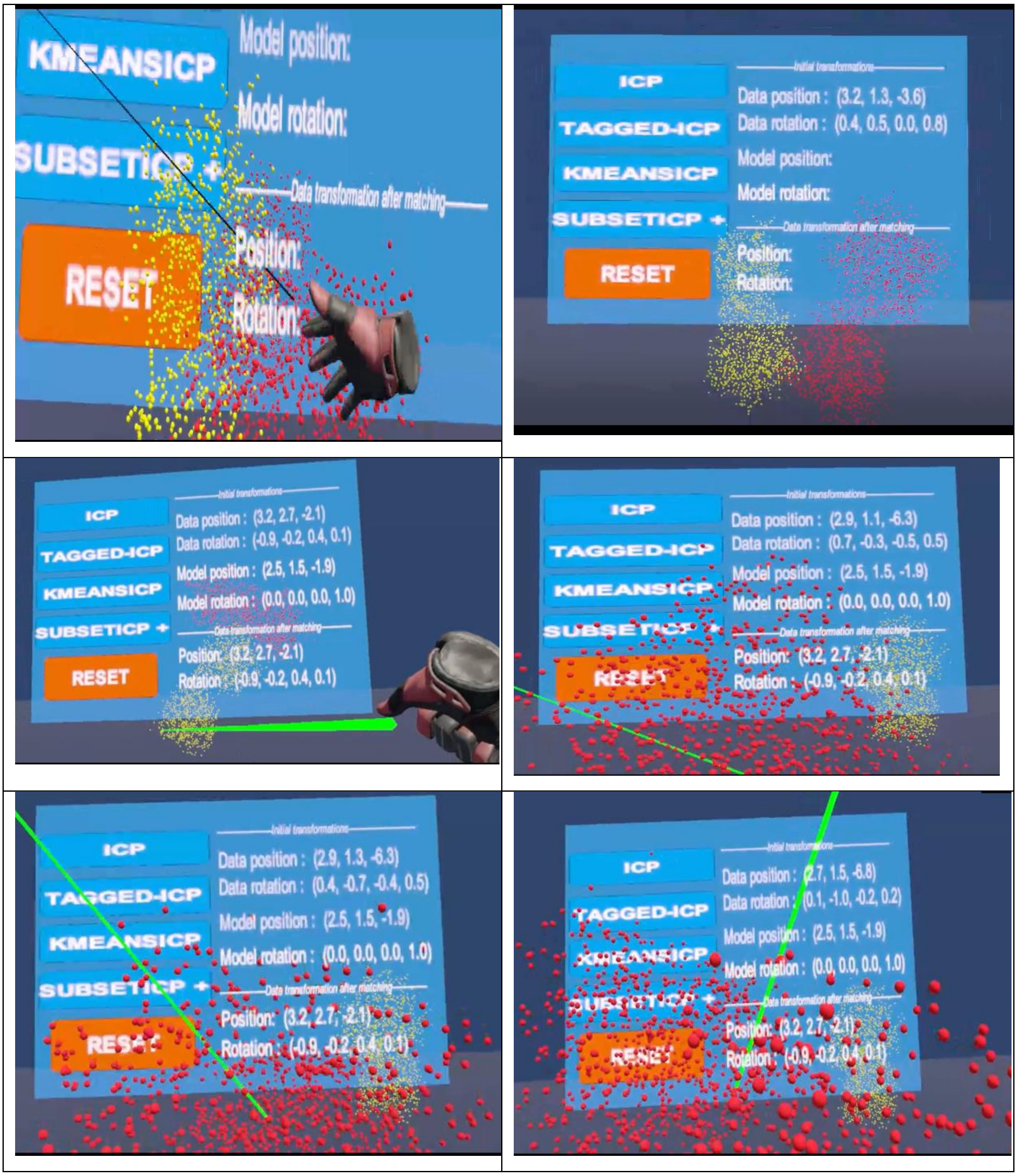
Model position:

Model rotation:

Data transformation after matching

Position:

Rotation:



Model position:
Model rotation:
Position:
Rotation:

Data position : (3.2, 1.3, -3.6)
Data rotation : (0.4, 0.5, 0.0, 0.8)
Model position:
Model rotation:
Position:
Rotation:

Data position : (3.2, 2.7, -2.1)
Data rotation : (-0.9, -0.2, 0.4, 0.1)
Model position : (2.5, 1.5, -1.9)
Model rotation : (0.0, 0.0, 0.0, 1.0)
Position: (3.2, 2.7, -2.1)
Rotation : (-0.9, -0.2, 0.4, 0.1)

Data position : (2.9, 1.1, -6.3)
Data rotation : (0.7, -0.3, -0.5, 0.5)
Model position : (2.5, 1.5, -1.9)
Model rotation : (0.0, 0.0, 0.0, 1.0)
Position: (3.2, 2.7, -2.1)
Rotation : (-0.9, -0.2, 0.4, 0.1)

Data position : (2.9, 1.3, -6.3)
Data rotation : (0.4, -0.7, -0.4, 0.5)
Model position : (2.5, 1.5, -1.9)
Model rotation : (0.0, 0.0, 0.0, 1.0)
Position: (3.2, 2.7, -2.1)
Rotation : (-0.9, -0.2, 0.4, 0.1)

Data position : (2.7, 1.5, -6.8)
Data rotation : (0.1, -1.0, -0.2, 0.2)
Model position : (2.5, 1.5, -1.9)
Model rotation : (0.0, 0.0, 0.0, 1.0)
Position: (3.2, 2.7, -2.1)
Rotation : (-0.9, -0.2, 0.4, 0.1)

Figure 8.19 A user interacting with the data point cloud (in red), using the HTC Vive hand controller and pointing, selecting, rotating and moving the data point cloud.

The visibility of the shape's transformation can also be enhanced with tooltips on top of the shape to dynamically show the current rotation and position as the user manipulates the object. However, this is not implemented in the current version of ICPVR.

Several studies have evaluated the use of hand controllers for interacting with 3-D objects in virtual reality. One such study evaluated the usability of the HTC VIVE (HTC, 2020) hand controllers and hand gestures using data gloves on task performance (Lee et al., 2017, 2017). The researchers experimented with participants running a VR game with an HTC VIVE controller and then the data gloves. The participants were then asked to fill a questionnaire after each test. The experiment concluded on data gloves as interaction preference by the users. The choice of acceptance of a particular interaction medium is affected by the responsiveness, understandability, user pre-emptiveness, and accuracy of using that interaction medium. It can be deduced that data gloves presented a more natural interaction medium than hand controllers.

It has also been noted that spatial perception in a virtual reality environment is enhanced by how natural the interaction is (Seibert and Shafer, 2017). For instance, using two hands for interaction in the virtual environment because of the use of two controllers, mimics human natural usage of hands which increases the users feeling of immersion. It can be easier to locate and interact with two objects at the same time using hand controllers in both hands (Khambadkar and Folmer, 2013), however, other research has shown users prefer using one hand in VR interactions (Nanjappan et al., 2018). The differences are because of the specific tasks the users performed. For instance, users preferred using only one hand for manipulating 3-D objects in VR environments (selection, rotation and moving tasks).

In the case of ICPVR, the two 3-D objects were easier to interact with, using two VR hand controllers. The evaluation felt natural to interact with both data shape and model shape with both hands using the two (2) VR controllers at the same time. The user interface was easy to manipulate. The placement of the model point cloud and the animation of the shape match was easily able to be viewed. The teleportation shown in Figure 8.2, allowed an enhanced interaction by giving the user the ability to move around the virtual world whilst remaining physically static especially useful for users with limited physical space. The user was able to successfully place the point clouds at desired positions in the virtual world and clicked an algorithm to observe the animated shape match as shown in the matching of two point clouds in Figure 8.17 and Figure 8.19.

8.5.4 Usability

ICPVR was found to be useful for evaluating shape matching algorithms with visualisations because all the desired tasks were able to be performed in the virtual environment. The shapes were interactable, details of the initial transformations were shown even though this can be enhanced with real-time information on the transformation whilst the user interacts with the shapes.

It was easy to teleport around the virtual environment especially going closer or farther to the virtual shapes to better see the shape matching animation as required. After the matching, A visual inspection of the shape overlap was able to reveal the regions of the shape with a much better match than the others.

8.5.5 Knowledge Improvements

The more the system was used to visualise protein structures, the more visual insight was realised. For instance, in the desktop version, some instances of the ICP iteration were seen to recover from local minima, however, in virtual reality, this was more vivid due to the immersion.

Using the virtual environment increased understanding of the shape matching process and allowed new ideas to be envisaged such as showing real time information during the shape matching process such as data of worse transformations.

8.5.6 Towards Accurate Initial Transformation

The use of hand controllers in the VR simulation, allows the user to interact with the point clouds more easily as it closely mimics natural interaction than using a desktop computer's mouse and keyboard. Using the hand controllers makes the interaction feel more natural and accurate because the user is not limited to the 2-D interface from the PC monitor.

The enhanced interaction and higher field of view (The visible area of the VR content for the user at an instance without turning their head) afforded by the virtual environment (ICPVR) enables the user to see the point clouds in more detail and as such be able to set a much-desired transformation based on their visual analysis and perception as shown in (Figure 8.17, Figure 8.19, Figure 8.18)

8.6 Limitations of ICPVR

ICPVR shows the user a real-time animation of the shape matching process in the virtual reality environment. This real time animation was achieved by applying the computed transformation to the data shape in each iteration and automatically updating the point cloud in the virtual environment. This repeated process creates visible animation. This process can be computationally expensive and can result in a lagged virtual environment.

To improve the performance of the animation and prevent lagging in the VR environment, the shape matching process implemented Unity3D (Unity Technologies, 2019) Coroutine feature that allows flexibility in returning control to the display. This is achieved by asynchronously running the ICP algorithm and returning control to the display (refreshing the display to show the latest update of the positions of the atoms) after each computed transformation is applied to the data shape. This process ensures that the user sees the point positions from the most recent completed iteration. This flexibility means that the animation is rendered to the display over several frames. The user is then able to see, after each iteration, the data shape rotating and moving closer to the model shape as computed transformations are applied to the data shape bringing it closer to the model shape. The closeness is judged solely by the user's perception. However, ICPVR can be enhanced to show how close the shapes are using real time display of the distance between the shapes.

The current representation of the molecules as point clouds also makes the user able to walk into the point clouds in the virtual environment. This allows a user to easily visualise points that are closer to the centre of the point cloud as the user virtually moves into the molecular structure. The ability to walk into the molecule allows the user to study the overlaps on atoms that are closer to the centre of the molecules (the molecule is volumetric) to visually see wrong matches and decide on how the data point can be moved as well as how to adjust the initial transformation. Bruder, Steinicke and Nüchter, (2014) found sparse point clouds generated from the robot unsuitable because they wanted to be able to visualise the 3-D buildings with the walls seen as solid objects which are only possible with many dense point clouds. ICPVR rather requires the point cloud to be sparse enough so the visual overlaps of the shapes can be analysed after the matching is complete.

Collision detection can be implemented to give the user more details on the registration including real time data on the points being matched based on the detected collisions between the corresponding points.

Even though the focus of ICPVR was to present a simple 3-D UI, many of the parameters in the ICP algorithm were not exposed in the UI. For instance, the user cannot set the number of iterations nor the different algorithm-specific parameters, such as subset size, k-means cluster size.

Even with the limitations discussed, ICP usually converges to a local minimum and the visualization in VR can help traverse the local minimum and thus make sure that it eventually converges to the global minimum. This is critical for the detection of the mutations of molecules for drug design for example where accurate information is critical. It has the advantage of combining the strengths of human abstract inference and the computational power of the modern computer and hardware.

8.7 Summary

This chapter has discussed a virtual reality application, ICPVR that can be used for evaluation, testing and comparison of 3-D shape matching algorithms. ICPVR allows simulation of the performance of 3 ICP variants (KmeansICP, SubsetICP, and TaggedICP) on 3-D protein molecules in virtual reality. The chapter further discussed related works, highlighting existing virtual reality applications used for matching 3-D structures. Particularly for ICPVR, the chapter discussed ways a user can interact with the virtual 3-D protein structures and how easy ICPVR can be used for quick comparisons on ICP variants.

The research evaluated ICPVR in terms of interaction, immersive visualisation, knowledge improvements, and usability. The application highlighted the button receiving focus from the pointer when the user pointed using the VIVE controller. Even though the algorithm names are written on the buttons, this made it easier to distinguish between them and especially for the user to see the currently selected button before clicking. The application also allowed the pointing, selecting, rotating, and moving of the protein molecules (Figure 8.19). Manipulating the protein molecules was necessary for adjusting the initial transformations as shown in (Figure 8.15, Figure 8.16, Figure 8.17, and Figure 8.18). The teleportation feature (Figure 8.10) was also helpful for the researcher to move in the virtual environment whilst remaining in the limited space. This was necessary especially for the visual analysis of the molecules. The background was made less distractive with bright colours to ensure maximum immersion by the user in focusing fully on the shape matching process. The visual analysis of the shape matching animation improved the researcher’s knowledge and resulted in more ideas for future work.

This chapter has achieved objective four (4), “*Develop a virtual reality visualisation software for visualising and interacting with 3-D molecular protein data, allowing exploration of the molecule in immersive virtual reality environments*”. This has been achieved by developing and evaluating ICPVR as well as presenting an analysis of its strengths and limitations.

CHAPTER 9 CONCLUSIONS AND FUTURE WORK

9.1 Introduction

The research sought to achieve the aims and objectives set out in Section 1.4. This chapter discusses how the project meets all the objectives. The chapter further discusses the future work and how various parts of this research can be improved and extended. A personal reflection section also discusses the researcher's experience and skills gained whilst working on this project.

9.2 Summary and Discussion

The research proposed to study the domain of 3-D shape matching in virtual reality. The standard ICP (Besl and McKay, 1992) algorithm was studied as the base algorithm of choice to utilise for the shape matching process. It was found that ICP suffers from poor quality of match and convergence with large initial rotation angles and computationally expensive methods such as the correspondence search method. There is also the need for using novel visualisation techniques to understand the shape matching process.

These challenges presented opportunities for developing several variants of ICP to improve aspects of the algorithm for matching 3-D protein datasets. The research developed, evaluated, and critically discussed three (3) variants of the ICP algorithm for matching 3-D protein datasets (KmeansICP, SubsetICP+, and TaggedICP). It also developed a virtual reality shape matching application, ICPVR for visualisation of the shape matching in virtual reality. The prototype application has provided a means of visualising shape matches in virtual reality with the ability for the user to see the animation of the process and interacting with the shapes.

The research has investigated and reviewed virtual reality. It has outlined the history of virtual reality in a progressively developing manner. This has allowed an understanding and appreciation of the evolution of virtual reality for the reader to understand areas of its application and possibly the next phases of its development as well as the current applications of virtual reality. The metrics (the affordances) or what the system offers the user of VR have also been researched and discussed (immersion, presence, navigation, knowledge improvements, and usability). The approach considered several experiments using virtual reality and identified the specific affordances evaluated as well as the findings. Key concepts in virtual reality were also discussed such as how all the metrics of VR help to increase the user's feeling of immersion, and this contributes to the overall experience of a user. This review has contributed to advancing knowledge with a peer-reviewed publication titled *"Virtual Reality: A literature review and metrics-based classification"*. The virtual reality investigation and publication meets objective (2) of this research; "*Gain a critical understanding of the history, current state, and challenges of virtual reality through a literature review*".

In chapter 2, the research extensively identified and reviewed existing shape matching algorithms. The algorithms were discussed along lines of common properties of shape matching algorithms; feature identification, feature matching, transformation model estimation, and image transformation (Zitová and Flusser, 2003). It was also noted that all registration algorithms consist of varying implementations of some components namely, feature space, search space, search strategy, and similarity measure. The discussion highlighted the different techniques used in the algorithms, and their advantages and disadvantages. The discussion also explained how the various techniques used in shape matching algorithms allow a robust registration process in the presence of different levels of noise. It also highlighted current novel approaches to shape registration tasks, training and using machine learning algorithms to predict transformations and output of some methods of the registration process. The discussion meets the objective (1); "*Gain a critical understanding of similarities, differences, and potential areas for improvement in existing shape matching techniques by performing a thorough literature review".*

A system's research approach has been used to develop ICPVR (Iterative Closest Point Virtual Reality) which is a virtual reality environment for visualising shape matching. ICPVR allows a shape matching algorithm to be added to the virtual environment with some modifications to allow amongst other things, placeholders for the data and model shapes with some amount of modification to the shape matching code. ICPVR enables the user to interact with the 3-D objects as well as navigate in the virtual environment. The purpose of enabling these metrics for the virtual environment was to increase the levels of the user's overall experience. A user being immersed in the shape matching process from setting initial rotation angles of the 3-D objects to visualising the shape matching animation to the final shape matching, allows the user to discover patterns in the shape match process. Such insights can be used to further improve the shape matching process. Optimising the visualisation in virtual reality also advances the study, evaluation, and usability of virtual reality environments for performing tasks. This is applicable in environments such as scientific visualisations, learning or task-based health rehabilitation. ICPVR is basic and can be enhanced further to provide improvements such as better responsiveness during the VR simulation, improved aesthetics, improved automated ways of benchmarking or testing new algorithms and easier integration with new shape matching algorithms. The ICPVR development meets the objective; “Develop a virtual reality visualisation software for visualising and interacting with 3-D molecular protein data, allowing data exploration on a desktop and in immersive virtual reality environments” The ICPVR application provided the user with a means of using the hand controllers to interact with the virtual proteins. The ICPVR, its background literature and evaluation meet objective four (4): *“Develop a virtual reality visualisation software for visualising and interacting with 3-D molecular protein data, allowing exploration of the molecule in immersive virtual reality environments”.*

The research further developed and evaluated three (3) variants of the ICP algorithm, TaggedICP; An ICP variant that uses metadata knowledge to improve the registration, KmeansICP; An ICP based algorithm that partitions the correspondence search space based on k-means clustering (Krishna and Narasimha Murty, 1999), and SubsetICP+; An ICP algorithm based on SubsetICP (Chen, Belaton and Pan, 2013) that uses atom types for partitioning. The research was interested in comparing the developed algorithms and understanding how the different partition types and algorithm implementation affects shape registration with different levels of error and initial rotation angles. The experiments evaluated the different partitioning techniques and considered the effects of the size of the partitions, the complexity of the shapes, and the type of partitions (k-means clustering-based or atom types based) on the match quality and convergence of the different ICP algorithms.

TaggedICP uses metadata, called the tag to aid with the registration of the point cloud. The tag information consists of the atom type of the data point and its 3 closest neighbouring atoms. The tag information provides constraints that allow meaningful matches to be found especially for full same shape matching that has existing true correspondences. The algorithm was found to have a high match quality and performed well in the presence of small to moderate noises. However, the algorithm was found to suffer in the presence of large noises. SubsetICP+ performs registration by registering corresponding subsets of the two-point clouds. Although SubsetICP+ doesn’t make use of metadata, it was designed to register shapes by performing a complete ICP registration on each corresponding. SubsetICP+ was found to converge faster in many instances and the rejection of some transformation provided a good match quality and convergence in many instances with noise. KmeansICP used the K-means algorithm to cluster the search space to reduce the time taken for the search of correspondence.

The research found out that shape matching using metadata or some known information about the point improves the quality of the match. However, this was not found to be true in all cases. Additionally, considering the neighbouring information of the points in the search for correspondence can further help increase the match quality further at a cost of the number of neighbours. This means that many neighbourhood points being considered provides a higher level of constraint and can make the algorithm slow to converge and less accurate in the presence of noise. A very small number of neighbourhood points being considered can also make the results less accurate. This required the use of 3 neighbour points as a trade-off between the quality of the match and the speed of convergence. The development and evaluation of the developed variants achieve the objective (3); "*Develop and evaluate shape matching algorithms based on the Standard ICP algorithm"*

The development and evaluation of the algorithms along with their discussions achieves the research objective; "Develop, evaluate and visualise shape matching algorithms based on the Standard ICP algorithm (Besl and McKay, 1992) that explores different techniques for matching 3-D molecules and visualising the matching in virtual reality".

8.8 Future Work

Future work on ICPVR can improve the UI design of ICPVR and implement other 3-D shape matching algorithms. A comparison framework can be built to automatically test all algorithms and visualise the results in VR. ICPVR can also be enhanced to gather objective user experience feedback through the implementation of task scenarios. This feedback can help improve the visualisation and interaction in ICPVR. The search for correspondence methods in the ICP variants can utilise CPU parallel processing and GPU to speed up the process.

The algorithms can also be improved by implementing certain techniques such as rejection of false matches, weighing of the reliability of matches, using symmetric ICP to establish point matches in two directions, instead of one direction from data shape to model shape, global optimization of various parameters inside the proposed ICP variants, enhanced visualization of molecules in complex shapes, instead of point primitives, and the ability for ICPVR to provide immediate feedback on the quality of the shape matching in VR and help the user to decide when to stop.

Other interesting extensions of the ICP algorithm that can be pursued include registration of partial overlapping shapes, using point-to-plane distance-based objective functions, and extraction of local features to help refine the matches.

Non-ICP based shape registration methods such as (Horache, Deschaud and Goulette, 2021) end-to-end learning local multi-view descriptors for 3D point clouds (Li et al., 2020), deep learning based two-stage method for computing local descriptors for point cloud registration (Poiesi and Boscaini, 2021), and spinNet (Ao et al., 2021) have demonstrated state-of-the-art results on datasets such as 3DMatch benchmark and (Zeng et al., 2017) ETH (Pomerleau et al., 2012) for 3D point cloud registration. These methods show a great potential for the use of machine learning models in real world 3D point registration problems.

MS-SVConv (Horache, Deschaud and Goulette, 2021) is developed for registering challenging datasets that are largely different. The approach combines MS-SVConv which doesn't require all points in the point cloud to be considered in the registration process and Unsupervised transfer learning with data generation (UDGE) that creates two partial point clouds from a single point cloud. These two techniques combine to create an unsupervised deep learning model that works on unknown datasets to achieve a model that generalises for registering dissimilar datasets such as points clouds from different sensors or different environments. SpinNet (Ao et al., 2021) also generalises for unseen scenarios to aid with matching point clouds from different sensors, however unlike MS-SVConv (Horache, Deschaud and Goulette, 2021) spinNet leverages point-based and 3D cylindrical convolutional natural layers to provide a rotation invariant neural feature learner model for 3D surface matching. The local multi-view approach to shape descriptors (Li et al., 2020) differs from the voxel based approach (Horache, Deschaud and Goulette, 2021) in a way that allows it to present more information of the 3D geometry. Poiesi and Boscaini,(2021) presents a deep learning-based model to compute the generalisable and distinctive (GeDi) local descriptors for point cloud registration. The approach transforms sets of neighbouring points into a feature descriptor that helps the registration process achieve rotation-invariance.

Future work on non-ICP based methods can investigate registering deformable objects using deep learning approaches as well as creating new 3D feature descriptor and extractor models.

8.9 Reflections on my Research Journey

My PhD journey has been a great experience. Gradually adding to my knowledge in the domain of this thesis has been exciting and challenging at the same time. I have been fortunate to have supervisors who have systemically helped and encourage me to learn and do the right thing.

Through the journey, I have accepted decisions that were sometimes challenging, and in some instances, past understandings questioned, and newer insights explored. At the start of this journey, I was not fully aware of the magnitude of knowledge that I was going to develop and the amount of hard work that gaining such knowledge would entail. From the onset of my research, my supervisor will gently prompt me to read texts that I would not have naturally read or referred to. My supervisory team did point me to many helpful resources, and I could observe that care was taken not to overly guide me too much, but to try to make me decide on the path to follow as a way of developing my independent thinking further.

I have developed my writing, critical thinking, and presentation skills, whilst writing articles and presenting at conferences. I have also learnt the formal processes of research that are systematic to manage risks and result in maximum knowledge gain for the researcher as well as the validity of the output. I have also realised how critical these skills are for my further research and further growth.

REFERENCES

AGARWAL, S. and BHOWMICK, B., 2017. 3D point cloud registration with shape constraint. In: *2017 IEEE International Conference on Image Processing (ICIP)*. Presented at the 2017 IEEE International Conference on Image Processing (ICIP). pp. 2199–2203.

ALBERTS, B., JOHNSON, A., LEWIS, J., RAFF, M., ROBERTS, K., and WALTER, P., 2002. The Chemical Components of a Cell. *Molecular Biology of the Cell. 4th edition* [online]. Available from: https://www.ncbi.nlm.nih.gov/books/NBK26883/ [Accessed 29 Jun 2020].

ALEXIOU, E., YANG, N., and EBRAHIMI, T., 2020. PointXR: A Toolbox for Visualization and Subjective Evaluation of Point Clouds in Virtual Reality. In: *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)* [online]. Presented at the 2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX). pp. 1–6. Available from: 10.1109/QoMEX48832.2020.9123121.

ALZAYAT, A. and LEE, S.H. (Mark), 2021. Virtual products as an extension of my body: Exploring hedonic and utilitarian shopping value in a virtual reality retail environment. *Journal of Business Research*. 130, pp. 348–363.

ANGARAN, S., BOCK, M.E., GARUTTI, C., and GUERRA, C., 2009. MolLoc: a web tool for the local structural alignment of molecular surfaces. *Nucleic Acids Res*. 37 (suppl_2), pp. W565–W570.

ANIMESH, PINSONNEAULT, YANG, and OH, 2011a. An Odyssey into Virtual Worlds: Exploring the Impacts of Technological and Spatial Environments on Intention to Purchase Virtual Products. *MIS Quarterly*. 35 (3), p. 789.

ANIMESH, PINSONNEAULT, YANG, and OH, 2011b. An Odyssey into Virtual Worlds: Exploring the Impacts of Technological and Spatial Environments on Intention to Purchase Virtual Products. *MIS Quarterly*. 35 (3), p. 789.

ANKOMAH, P. and VANGORP, P., 2018a. Virtual Reality: A literature review and metrics-based classification. *Computer Graphics & Visual Computing* [online]. Available from: https://repository.edgehill.ac.uk/10615/1/paper1009_CRC.pdf.

ANKOMAH, P. and VANGORP, P., 2018b. Virtual Reality: A literature review and metrics-based classification. *Computer Graphics & Visual Computing* [online]. Available from: https://repository.edgehill.ac.uk/10615/1/paper1009_CRC.pdf.

ANKOMAH, P., VANGORP, P., BEHERA, A., and LIU, Y., 2020a. Tagged-ICP: An Iterative Closest Point Algorithm with Metadata Knowledge for Improved Matching of 3D Protein Structures. In: [online]. Presented at the Irish Machine Vision and Image Processing conference. Available from: https://research.edgehill.ac.uk/en/publications/tagged-icp-an-iterative-closest-point-algorithm-with-metadata-kno-2 [Accessed 11 Jul 2020].

ANKOMAH, P., VANGORP, P., BEHERA, A., and LIU, Y., 2020b. Tagged-ICP: An Iterative Closest Point Algorithm with Metadata Knowledge for Improved Matching of 3D Protein Structures. In: [online]. Presented at the Irish Machine Vision and Image Processing conference. Available from: https://research.edgehill.ac.uk/en/publications/tagged-icp-an-iterative-closest-point-algorithm-with-metadata-kno-2 [Accessed 11 Jul 2020].

AO, S., HU, Q., YANG, B., MARKHAM, A., and GUO, Y., 2021. SpinNet: Learning a General Surface Descriptor for 3D Point Cloud Registration. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* [online]. Presented at the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA: IEEE. pp. 11748–11757. Available from: https://ieeexplore.ieee.org/document/9577271/ [Accessed 14 Mar 2022].

AOKI, Y., GOFORTH, H., SRIVATSAN, R.A., and LUCEY, S., 2019. PointNetLK: Robust & Efficient Point Cloud Registration using PointNet. *arXiv:1903.05711 [cs]* [online]. Available from: http://arxiv.org/abs/1903.05711 [Accessed 9 Feb 2020].

ARD, T., KRUM, D.M., PHAN, T., DUNCAN, D., ESSEX, R., BOLAS, M., and TOGA, A., 2017. NIVR: Neuro imaging in virtual reality. In: *Proceedings - IEEE Virtual Reality* [online]. IEEE. pp. 465–466. Available from: http://ieeexplore.ieee.org/document/7892381/.

ARNALDI, B., GUITTON, P., and MOREAU, G., 2018a. *Virtual Reality and Augmented Reality: Myths and Realities* [online]. Newark, UNITED STATES: John Wiley & Sons, Incorporated. Available from: http://ebookcentral.proquest.com/lib/edgehill/detail.action?docID=5323668 [Accessed 24 May 2020].

ARNALDI, B., GUITTON, P., and MOREAU, G., 2018b. *Virtual Reality and Augmented Reality*. John Wiley & Sons, Incorporated.

ASSALIH, H., 2013. *3D Reconstruction and Motion Estimation Using Forward Looking Sonar*.

ATTIA, M. and SLAMA, Y., 2017. Efficient Initial Guess Determination Based on 3D Point Cloud Projection for ICP Algorithms. In: *2017 International Conference on High Performance Computing Simulation (HPCS)*. pp. 807–814.

AUDESIRK, T., BYERS, B.E., and AUDESIRK, G., 2016. *Biology: Life on Earth with Physiology*. 11th edition.

AXENOPOULOS, A., DARAS, P., PAPADOPOULOS, G., and HOUSTIS, E.N., 2011. A shape descriptor for fast complementarity matching in molecular docking. *IEEE/ACM Trans Comput Biol Bioinform*. 8 (6), pp. 1441–1457.

AZUMA, R., BAILLOT, Y., BEHRINGER, R., FEINER, S., JULIER, S., and MACINTYRE, B., 2001. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*. 21 (6), pp. 34–47.

AZUMA, R.T., 1997. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*. 6 (4), pp. 355–385.

BARFIELD, W. and HENDRIX, C., 1995. The effect of update rate on the sense of presence within virtual environments. *Virtual Reality*. 1 (1), pp. 3–15.

BBC News, 2020. Fortnite's Travis Scott virtual concert watched by millions. *BBC News* [online]. 24 April. Available from: https://www.bbc.com/news/technology-52410647 [Accessed 31 Jul 2021].

BELONGIE, S., MALIK, J., and PUZICHA, J., 2002. Shape Matching and Object Recognition Using Shape Contexts. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*. 24 (24), p. 14.

BENDER, A., MUSSA, H.Y., GLEN, R.C., and REILING, S., 2004. Similarity searching of chemical databases using atom environment descriptors (MOLPRINT 2D): evaluation of performance. *Journal of Chemical Information and Computer Sciences*. 44 (5), pp. 1708–1718.

BENTLEY, J.L., 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM*. 18 (9), pp. 509–517.

BENTLEY, J.L., 1990. K-d trees for semidynamic point sets. In: *Proceedings of the sixth annual symposium on Computational geometry* [online]. Berkley, California, USA: Association for Computing Machinery. pp. 187–197. Available from: 10/b7rcvf [Accessed 24 Apr 2020].

BERGÉ, L.-P., AOUF, N., DUVAL, T., and COPPIN, G., 2016. Generation and VR visualization of 3D point clouds for drone target validation assisted by an operator. In: *2016 8th Computer Science and Electronic Engineering (CEEC)*. Presented at the 2016 8th Computer Science and Electronic Engineering (CEEC). pp. 66–70.

BERMAN, H.M., WESTBROOK, J., FENG, Z., GILLILAND, G., BHAT, T.N., WEISSIG, H., SHINDYALOV, I.N., and BOURNE, P.E., 2000a. The Protein Data Bank. *Nucleic Acids Res*. 28 (1), pp. 235–242.

BERMAN, H.M., WESTBROOK, J., FENG, Z., GILLILAND, G., BHAT, T.N., WEISSIG, H., SHINDYALOV, I.N., and BOURNE, P.E., 2000b. The Protein Data Bank. *Nucleic Acids Res*. 28 (1), pp. 235–242.

BESL, P.J. and MCKAY, N.D., 1992a. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 14 (2), pp. 239–256.

BESL, P.J. and MCKAY, N.D., 1992b. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 14 (2), pp. 239–256.

BESL, P.J. and MCKAY, N.D., 1992c. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 14 (2), pp. 239–256.

BONDI, A., 1964. van der Waals Volumes and Radii. *The Journal of Physical Chemistry*. 68 (3), pp. 441–451.

BOOKSTEIN, F.L., 1989. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 11 (6), pp. 567–585.

BOREK, D., MINOR, W., and OTWINOWSKI, Z., 2003. Measurement errors and their consequences in protein crystallography. *Acta Crystallographica Section D: Biological Crystallography*. 59 (11), pp. 2031–2038.

BORGERS, A., BROUWER, M., KUNEN, T., JESSURUN JORAN, J., and JANSSEN INGRID, I., 2010. A virtual reality tool to measure shoppers' tenant mix preferences. *Computers, Environment and Urban Systems*. 34 (5), pp. 377–388.

BOUAZIZ, S., TAGLIASACCHI, A., and PAULY, M., 2013. Sparse Iterative Closest Point. *Computer Graphics Forum*. 32 (5), pp. 113–123.

BOWMAN, D.A., GABBARD, J.L., and HIX, D., 2002. A Survey of Usability Evaluation in Virtual Environments: Classification and Comparison of Methods. *Presence: Teleoperators and Virtual Environments*. 11 (4), pp. 404–424.

BOWMAN, D.A. and HODGES, L.F., 1999. Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments. *Journal of Visual Languages and Computing*. 10 (1), pp. 37–53.

BOYLE, A.L., 2018. Applications of de novo designed peptides. In: *Peptide Applications in Biomedicine, Biotechnology and Bioengineering* [online]. Elsevier. pp. 51–86. Available from: https://linkinghub.elsevier.com/retrieve/pii/B978008100736500003X [Accessed 7 Jul 2021].

BROOKS, F.P.Jr., 1987. Walkthrough—a dynamic graphics system for simulating virtual buildings. In: *Proceedings of the workshop on Interactive 3D graphics* [online]. pp. 9–21. Available from: http://portal.acm.org/citation.cfm?id=319120.319122.

BROWN, E. and CAIRNS, P., 2004. A grounded investigation of game immersion. In: *Extended abstracts of the 2004 conference on Human factors and computing systems - CHI '04* [online]. p. 1297. Available from: http://portal.acm.org/citation.cfm?doid=985921.986048.

BROWN, L.G., 1992. A survey of image registration techniques. *ACM Comput. Surv.* 24 (4), pp. 325–376.

BRUDER, G., STEINICKE, F., and NÜCHTER, A., 2014. Poster: Immersive point cloud virtual environments. In: *2014 IEEE Symposium on 3D User Interfaces (3DUI)* [online]. Presented at the 2014 IEEE Symposium on 3D User Interfaces (3DUI). pp. 161–162. Available from: 10.1109/3DUI.2014.6798870.

BRYSON, S. and LEVIT, C., 1992. The virtual wind tunnel. *IEEE Computer Graphics and Applications*. 12 (4), pp. 25–34.

BUCKLEY, R., 2020. Pandemic Travel Restrictions Provide a Test of Net Ecological Effects of Ecotourism and New Research Opportunities. *Journal of Travel Research*. p. 0047287520947812.

BUTT, A.L., KARDONG-EDGREN, S., and ELLERTSON, A., 2018a. Using Game-Based Virtual Reality with Haptics for Skill Acquisition. *Clinical Simulation in Nursing*. 16, pp. 25–32.

BUTT, A.L., KARDONG-EDGREN, S., and ELLERTSON, A., 2018b. Using Game-Based Virtual Reality with Haptics for Skill Acquisition. *Clinical Simulation in Nursing*. 16, pp. 25–32.

CALADO, A.V.S., SOARES, M.M., CAMPOS, F., and CORREIA, W., 2013. Virtual reality applied to the study of the interaction between the user and the built space: A literature review. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* [online]. pp. 345–351. Available from: https://pdfs.semanticscholar.org/03a0/36b19edb1bdd6716dbd663f086ce3cbf7590.pdf.

CALVELO, M., PIÑEIRO, Á., and GARCIA-FANDINO, R., 2020. An immersive journey to the molecular structure of SARS-CoV-2: Virtual reality in COVID-19. *Computational and Structural Biotechnology Journal*. 18, pp. 2621–2628.

CAMPORESI, C., KALLMANN, M., and HAN, J.J., 2013. VR solutions for improving physical therapy. In: *2013 IEEE Virtual Reality (VR)*. Presented at the 2013 IEEE Virtual Reality (VR). pp. 77–78.

CARMIGNIANI, J., FURHT, B., ANISETTI, M., CERAVOLO, P., DAMIANI, E., and IVKOVIC, M., 2011. Augmented reality technologies, systems and applications. *Multimedia Tools and Applications*. 51 (1), pp. 341–377.

CASSIDY, K.C., ŠEFČÍK, J., RAGHAV, Y., CHANG, A., and DURRANT, J.D., 2020. ProteinVR: Web-based molecular visualization in virtual reality. *PLOS Computational Biology*. 16 (3), p. e1007747.

CHANG, C.-M., HSU, C.-H., HSU, C.-F., and CHEN, K.-T., 2016a. Performance Measurements of Virtual Reality Systems. In: *Proc.$\backslash$ ACM on Multimedia Conference*. pp. 655–659.

CHANG, C.-M., HSU, C.-H., HSU, C.-F., and CHEN, K.-T., 2016b. Performance Measurements of Virtual Reality Systems. *Proceedings of the 2016 ACM on Multimedia Conference - MM '16*. pp. 655–659.

CHAVENT, M., LÉVY, B., KRONE, M., BIDMON, K., NOMINÉ, J.-P., ERTL, T., and BAADEN, M., 2011. GPU-powered tools boost molecular visualization. *Brief Bioinform*. 12 (6), pp. 689–701.

CHAVENT, M., VANEL, A., TEK, A., LEVY, B., ROBERT, S., RAFFIN, B., and BAADEN, M., 2011. GPU-accelerated atom and dynamic bond visualization using hyperballs: A unified algorithm for balls, sticks, and hyperboloids. *Journal of computational chemistry*. 32 (13), pp. 2924–2935.

CHEN, J. and BELATON, B., 2014a. An Improved Iterative Closest Point Algorithm for Rigid Point Registration. In: X. WANG, W. PEDRYCZ, P. CHAN and Q. HE, eds. *Machine Learning and Cybernetics* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg. pp. 255–263. Available from: http://link.springer.com/10.1007/978-3-662-45652-1_26 [Accessed 16 Dec 2019].

CHEN, J. and BELATON, B., 2014b. An Improved Iterative Closest Point Algorithm for Rigid Point Registration. In: X. WANG, W. PEDRYCZ, P. CHAN and Q. HE, eds. *Machine Learning and Cybernetics* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg. pp. 255–263. Available from: http://link.springer.com/10.1007/978-3-662-45652-1_26 [Accessed 16 Dec 2019].

CHEN, J., BELATON, B., and PAN, Z., 2013a. A Robust Subset-ICP Method for Point Set Registration. In: D. HUTCHISON, T. KANADE, J. KITTLER, J.M. KLEINBERG, F. MATTERN, J.C. MITCHELL, M. NAOR, O. NIERSTRASZ, C. PANDU RANGAN, B. STEFFEN, M. SUDAN, D. TERZOPOULOS, D. TYGAR, M.Y. VARDI, G. WEIKUM, H.B. ZAMAN, P. ROBINSON, P. OLIVIER, T.K. SHIH and S. VELASTIN, eds. *Advances in Visual Informatics* [online]. Cham: Springer International Publishing. pp. 59–69. Available from: http://link.springer.com/10.1007/978-3-319-02958-0_6 [Accessed 21 Feb 2020].

CHEN, J., BELATON, B., and PAN, Z., 2013b. A Robust Subset-ICP Method for Point Set Registration. In: D. HUTCHISON, T. KANADE, J. KITTLER, J.M. KLEINBERG, F. MATTERN, J.C. MITCHELL, M. NAOR, O. NIERSTRASZ, C. PANDU RANGAN, B. STEFFEN, M. SUDAN, D. TERZOPOULOS, D. TYGAR, M.Y. VARDI, G. WEIKUM, H.B. ZAMAN, P. ROBINSON, P. OLIVIER, T.K. SHIH and S. VELASTIN, eds. *Advances in Visual Informatics* [online]. Cham: Springer International Publishing. pp. 59–69. Available from: http://link.springer.com/10.1007/978-3-319-02958-0_6 [Accessed 21 Feb 2020].

CHEN, Y. and MEDIONI, G., 1992. Object modelling by registration of multiple range images. *Image and Vision Computing*. 10 (3), pp. 145–155.

CHENG, L.K., CHIENG, M.H., and CHIENG, W.H., 2014. Measuring virtual experience in a three-dimensional virtual reality interactive simulator environment: A structural equation modeling approach. *Virtual Reality*. 18 (3), pp. 173–188.

CHETVERIKOV, D., SVIRKO, D., STEPANOV, D., and KRSEK, P., 2002. The Trimmed Iterative Closest Point algorithm. In: *Object recognition supported by user interaction for service robots*. Presented at the Object recognition supported by user interaction for service robots. pp. 545–548 vol.3.

CHOI, D.H., KIM, J., and KIM, S.H., 2007. ERP training with a web-based electronic learning system: The flow theory perspective. *International Journal of Human Computer Studies*. 65 (3), pp. 223–243.

CHOY, C., DONG, W., and KOLTUN, V., 2020. Deep Global Registration. In: [online]. Presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2514–2523. Available from: https://openaccess.thecvf.com/content_CVPR_2020/html/Choy_Deep_Global_Registration_CVPR_2020_paper.html [Accessed 27 Jul 2021].

CHUI, H. and RANGARAJAN, A., 2003. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*. 89 (2–3), pp. 114–141.

CHUNG, N.C., MIASOJEDOW, B., STARTEK, M., and GAMBIN, A., 2019. Jaccard/Tanimoto similarity test and estimation methods. *arXiv:1903.11372 [stat]* [online]. Available from: http://arxiv.org/abs/1903.11372 [Accessed 5 Nov 2019].

CHU-SONG CHEN, YI-PING HUNG, and JEN-BO CHENG, 1999. RANSAC-based DARCES: a new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 21 (11), pp. 1229–1234.

CIFTCI, U., ZHANG, X., and TIN, L., 2017. Partially occluded facial action recognition and interaction in virtual reality applications. In: *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. pp. 715–720.

CLARK, D., DEMMEL, J., HONG, J., LAFFERRIERE, G., SALKIND, L., and TAN, X., 1989. VPL DataGlove. In: *NASA Conference on Space Telerobotics*.

COCKBURN, A. and MCKENZIE, B., 2002a. Evaluating the effectiveness of spatial memory in 2D and 3D physical and virtual environments. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* [online]. Minneapolis, Minnesota, USA: Association for Computing Machinery. pp. 203–210. Available from: https://doi.org/10.1145/503376.503413 [Accessed 10 Jun 2020].

COCKBURN, A. and MCKENZIE, B., 2002b. Evaluating the Effectiveness of Spatial Memory in 2D and 3D Physical and Virtual Environments. *Conference on Human Factors and Computing*

Systems. Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves. (4), pp. 203–210.

COLDHAM, G. and COOK, D.M., 2017. VR usability from elderly cohorts: Preparatory challenges in overcoming technology rejection. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2017-September, pp. 131–135.

COLDHAM, G. and COOK, D.M., 2018. VR usability from elderly cohorts: Preparatory challenges in overcoming technology rejection. *2017 National Information Technology Conference, NITC 2017*. 2017-September, pp. 131–135.

COMBES, B. and PRIMA, S., 2009. Prior affinity measures on matches for ICP-like nonlinear registration of free-form surfaces. In: *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. pp. 370–373.

CONNER, B.D., SNIBBE, S.S., HERNDON, K.P., ROBBINS, D.C., ZELEZNIK, R.C., and VAN DAM, A., 1992. Three-dimensional widgets. In: *Proceedings of the 1992 symposium on Interactive 3D graphics - SI3D '92* [online]. pp. 183–188. Available from: http://portal.acm.org/citation.cfm?doid=147156.147199.

COOGAN, C.G. and HE, B., 2018. Brain-Computer Interface Control in a Virtual Reality Environment and Applications for the Internet of Things. *IEEE Access*. 6 (c), pp. 10840–10849.

COREY, R.B. and PAULING, L., 1953. Molecular Models of Amino Acids, Peptides, and Proteins. *Review of Scientific Instruments*. 24 (8), pp. 621–627.

DA COSTA, L.A.L.F. and NEDEL, L.P., 2017a. An Immersive Visualization Study on Molecules Manipulation. In: *Proc.$\backslash$ IEEE Symposium on Virtual and Augmented Reality*.

DA COSTA, L.A.L.F. and NEDEL, L.P., 2017b. An Immersive Visualization Study on Molecules Manipulation. In: *Proc. IEEE Symposium on Virtual and Augmented Reality*.

CRUZ-NEIRA, C., SANDIN, D.J., DEFANTI, T.A., KENYON, R.V., and HART, J.C., 1992. The CAVE: audio visual experience automatic virtual environment. *Communications of the ACM*. 35 (6), pp. 64–72.

CUMMINGS, J.J. and BAILENSON, J.N., 2016a. How Immersive Is Enough? A Meta-Analysis of the Effect of Immersive Technology on User Presence. *Media Psychology*. 19 (2), pp. 272–309.

CUMMINGS, J.J. and BAILENSON, J.N., 2016b. How Immersive Is Enough? {A} Meta-Analysis of the Effect of Immersive Technology on User Presence. *Media Psychology*. 19 (2), pp. 272–309.

DACHSELT, R. and HÜBNER, A., 2007. Three-dimensional menus: A survey and taxonomy. *Computers and Graphics (Pergamon)*. 31 (1), pp. 53–65.

VAN DAM, A., LAIDLAW, D.H., and SIMPSON, R.M., 2002. Experiments in Immersive Virtual Reality for Scientific Visualization. *Computers & Graphics*. 26 (4), pp. 535–555.

DEMIRALP, Ç., JACKSON, C.D., KARELITZ, D.B., ZHANG, S., and LAIDLAW, D.H., 2006a. {CAVE} and fishtank virtual-reality displays: A qualitative and quantitative comparison. *IEEE Transactions on Visualization and Computer Graphics*. 12 (3), pp. 323–330.

DEMIRALP, Ç., JACKSON, C.D., KARELITZ, D.B., ZHANG, S., and LAIDLAW, D.H., 2006b. CAVE and fishtank virtual-reality displays: A qualitative and quantitative comparison. *IEEE Transactions on Visualization and Computer Graphics*. 12 (3), pp. 323–330.

DEPRELLE, T., GROUEIX, T., FISHER, M., KIM, V.G., RUSSELL, B.C., and AUBRY, M., 2019. Learning elementary structures for 3D shape generation and matching. *arXiv:1908.04725 [cs]* [online]. Available from: http://arxiv.org/abs/1908.04725 [Accessed 22 Jan 2022].

DESAI, P.R., DESAI, P.N., AJMERA, K.D., and MEHTA, K., 2014. A Review Paper on Oculus Rift-A Virtual Reality Headset. *arXiv:1408.1173 [cs]* [online]. Available from: http://arxiv.org/abs/1408.1173 [Accessed 25 May 2020].

DEY, A., BILLINGHURST, M., LINDEMAN, R.W., and SWAN, J.E.I., 2018. A Systematic Review of 10 Years of Augmented Reality Usability Studies: 2005 to 2014. *Frontiers in Robotics and AI* [online]. 5. Available from: https://www.frontiersin.org/articles/10.3389/frobt.2018.00037/full [Accessed 17 Jun 2020].

DIEMER, J., ALPERS, G.W., PEPERKORN, H.M., SHIBAN, Y., and MÜHLBERGER, A., 2015. The impact of perception and presence on emotional reactions: a review of research in virtual reality. *Frontiers in Psychology*. 6, p. 26.

DÍEZ, Y., ROURE, F., LLADÓ, X., and SALVI, J., 2015. A Qualitative Review on 3D Coarse Registration Methods. *ACM Comput. Surv.* 47 (3), p. 45:1-45:36.

DOLEŽAL, M., CHMELIK, J., and LIAROKAPIS, F., 2017. An immersive virtual environment for collaborative geovisualization. In: *Proc.$\backslash$ Virtual Worlds and Games for Serious Applications*. pp. 272–275.

DONOSO, F.A., AUSTIN, K.J., and MCAREE, P.R., 2017a. Three new Iterative Closest Point variant-methods that improve scan matching for surface mining terrain. *Robotics and Autonomous Systems*. 95, pp. 117–128.

DONOSO, F.A., AUSTIN, K.J., and MCAREE, P.R., 2017b. How do ICP variants perform when used for scan matching terrain point clouds? *Robotics and Autonomous Systems*. 87, pp. 147–161.

DOUTRELIGNE, S., CRAGNOLINI, T., PASQUALI, S., DERREUMAUX, P., and BAADEN, M., 2014. UnityMol: Interactive scientific visualization for integrative biology. In: *IEEE Symposium on Large Data Analysis and Visualization 2014, LDAV 2014 - Proceedings* [online]. IEEE. pp. 109–110. Available from: http://ieeexplore.ieee.org/document/7013213/.

DU, S., CUI, W., ZHANG, X., WU, L., and XIONG, L., 2017. Precise isotropic scaling iterative closest point algorithm based on corner points for shape registration. In: *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* [online]. Presented at the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 1811–1815. Available from: 10.1109/SMC.2017.8122879.

DU, S., XU, Y., WAN, T., HU, H., ZHANG, S., XU, G., and ZHANG, X., 2017. Robust iterative closest point algorithm based on global reference point for rotation invariant registration. *PLOS ONE*. 12 (11), p. e0188039.

ECKERT, H. and BAJORATH, J., 2007. Molecular similarity analysis in virtual screening: foundations, limitations and novel approaches. *Drug Discovery Today*. 12 (5), pp. 225–233.

EDEN, S. and BEZER, M., 2011a. Three-dimensions vs. two-dimensions intervention programs: The effect on the mediation level and behavioural aspects of children with intellectual disability. *European Journal of Special Needs Education*. 26 (3), pp. 337–353.

EDEN, S. and BEZER, M., 2011b. Three-dimensions vs. two-dimensions intervention programs: the effect on the mediation level and behavioural aspects of children with intellectual disability. *European Journal of Special Needs Education*. 26 (3), pp. 337–353.

EDEN, S. and PASSIG, D., 2007. Three-Dimensionality As an Effective Mode of Representation for Expressing Sequential Time Perception. *J. Educational Computing Research*. 36 (1), pp. 51–63.

ELLINGSON, L. and ZHANG, J., 2012. Protein Surface Matching by Combining Local and Global Geometric Information. *PLOS ONE*. 7 (7), p. e40540.

ESCOBAR, J.M. and MONTENEGRO, R., 1996. Several aspects of three-dimensional Delaunay triangulation. *Advances in Engineering Software*. 27 (1), pp. 27–39.

FACEBOOK, 2020. *Oculus Quest 2: Our Most Advanced New All-in-one VR Headset | Oculus* [online]. [online]. Available from: https://www.oculus.com/quest-2/?locale=en_GB [Accessed 4 Aug 2021].

FANG, Y., JIN XIE, GUOXIAN DAI, MENG WANG, FAN ZHU, TIANTIAN XU, and WONG, E., 2015. 3D deep shape descriptor. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* [online]. Presented at the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA: IEEE. pp. 2319–2328. Available from: http://ieeexplore.ieee.org/document/7298845/ [Accessed 26 Jul 2021].

FAROOQ, B., CHERCHI, E., and SOBHANI, A., 2018. Virtual Immersive Reality for Stated Preference Travel Behavior Experiments: A Case Study of Autonomous Vehicles on Urban Roads. *Transportation Research Record*. 2672 (50), pp. 35–45.

FAUGERAS, O.D. and HEBERT, M., 1986. The Representation, Recognition, and Locating of 3-D Objects. *The International Journal of Robotics Research*. 5 (3), pp. 27–52.

FEBRETTI, A., NISHIMOTO, A., THIGPEN, T., TALANDIS, J., LONG, L., PIRTLE, J.D., PETERKA, T., VERLO, A., BROWN, M., PLEPYS, D., SANDIN, D., RENAMBOT, L., JOHNSON, A., and LEIGH, J., 2013. CAVE2: a hybrid reality environment for immersive simulation and information analysis. In: M. DOLINSKY and I.E. MCDOWALL, eds. [online]. Presented at the IS&T/SPIE Electronic Imaging, Burlingame, California, USA. p. 864903. Available from: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2005484 [Accessed 25 May 2020].

FIEDLER, M.J., CHEN, S.-J., JUDKINS, T.N., OLEYNIKOV, D., and STERGIOU, N., 2007. Virtual Reality for Robotic Laparoscopic Surgical Training. In: J.D. WESTWOOD, R.S. HALUCK, H.M. HOFFMAN, G.T. MOGEL, R. PHILLIPS, R.A. ROBB and K.G. VOSBURGH, eds. *Medicine Meets Virtual Reality 15*. pp. 127–129.

FISCHLER, M.A. and BOLLES, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*. 24 (6), pp. 381–395.

FISHER, S.S., MCGREEVY, M., HUMPHRIES, J., and ROBINETT, W., 1987. Virtual environment display system. In: *Proceedings of the 1986 workshop on Interactive 3D graphics* [online]. New York, NY, USA: Association for Computing Machinery. pp. 77–87. Available from: https://doi.org/10.1145/319120.319127 [Accessed 25 May 2021].

FONSECA, L.M.G. and MANJUNATH, B.S., 1996. Registration Techniques for Multisensor Remotely Sensed Imagery. p. 8.

FOO, J.-L., LOBE, T., and WINER, E., 2009. A Virtual Reality Environment for Patient Data Visualization and Endoscopic Surgical Planning. *Journal of Laparoendoscopic & Advanced Surgical Techniques*. 19 (s1), pp. s211–s217.

FREINA, L. and OTT, M., 2015. A Literature Review on Immersive Virtual Reality in Education: State of the Art and Perspectives. In: *International Scientific Conference on eLearning and Software for Education*. pp. 133–141.

GAL, R., SHAMIR, A., and COHEN-OR, D., 2007. Pose-Oblivious Shape Signature. *IEEE Transactions on Visualization and Computer Graphics*. 13 (2), pp. 261–271.

GALLAGHER, A.G. and CATES, C.U., 2004. Approval of virtual reality training for carotid stenting: What this means for procedural-based medicine. *Journal of the American Medical Association*. 292 (24), pp. 3024–3026.

GEHRER, D., 2015. CellUnity - an Interactive Tool for Illustrative Visualization of Molecular Reactions. In: *Proceedings of CESCG 2015: The 19th Central European Seminar on Computer Graphics (non-peer-reviewed)*. p. 7.

GODIN, G., RIOUX, M., and BARIBEAU, R., 1994a. Three-dimensional registration using range and intensity information. In: *Videometrics III* [online]. International Society for Optics and Photonics. pp. 279–290. Available from: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/2350/0000/Three-dimensional-registration-using-range-and-intensity-information/10.1117/12.189139.short [Accessed 14 Jan 2020].

GODIN, G., RIOUX, M., and BARIBEAU, R., 1994b. Three-dimensional registration using range and intensity information. In: *Videometrics III* [online]. International Society for Optics and Photonics. pp. 279–290. Available from: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/2350/0000/Three-dimensional-registration-using-range-and-intensity-information/10.1117/12.189139.short [Accessed 14 Jan 2020].

GOLD, S., RANGARAJAN, A., LU, C.-P., PAPPU, S., and MJOLSNESS, E., 1998. New algorithms for 2D and 3D point matching: pose estimation and correspondence. *Pattern Recognition*. 31 (8), pp. 1019–1031.

GOLYANIK, V., ALI, S.A., and STRICKER, D., 2016. Gravitational Approach for Point Set Registration. In: [online]. Presented at the Proceedings of the IEEE Conference on Computer Vision and

Pattern Recognition. pp. 5802–5810. Available from: https://openaccess.thecvf.com/content_cvpr_2016/html/Golyanik_Gravitational_Approach_for_CVPR_2016_paper.html [Accessed 20 Jul 2021].

GOOGLE, 2016. *Google Cardboard – Google VR* [online]. Available from: https://vr.google.com/cardboard/.

GROUEIX, T., FISHER, M., KIM, V.G., RUSSELL, B.C., and AUBRY, M., 2018. 3D-CODED : 3D Correspondences by Deep Deformation. *arXiv:1806.05228 [cs]* [online]. Available from: http://arxiv.org/abs/1806.05228 [Accessed 22 Jan 2022].

GUAN, H., SU, Y., HU, T., WANG, R., MA, Q., YANG, Q., SUN, X., LI, Y., JIN, S., ZHANG, J., MA, Q., LIU, M., WU, F., and GUO, Q., 2020. A Novel Framework to Automatically Fuse Multiplatform LiDAR Data in Forest Environments Based on Tree Locations. *IEEE Transactions on Geoscience and Remote Sensing*. 58 (3), pp. 2165–2177.

GUO, Y., SOHEL, F.A., BENNAMOUN, M., WAN, J., and LU, M., 2013. RoPS: A local feature descriptor for 3D rigid objects based on rotational projection statistics. In: *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*. Presented at the 2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA). pp. 1–6.

HE, Y., YANG, J., HOU, X., PANG, S., and CHEN, J., 2021. ICP registration with DCA descriptor for 3D point clouds. *Optics Express*. 29 (13), p. 20423.

HEILIG, M.L., 1992. EL Cine del Futuro: The Cinema of the Future. *Presence: Teleoperators and Virtual Environments*. 1 (3), pp. 279–294.

HEMLATA and TIWARI, A., 2015. Applications of bioinformatics tools to combat the antibiotic resistance. In: *2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI)* [online]. Presented at the 2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI), Faridabad, India: IEEE. pp. 96–98. Available from: http://ieeexplore.ieee.org/document/7489545/ [Accessed 19 Jun 2021].

HILLMANN, C., 2019. Comparing the Gear VR, Oculus Go, and Oculus Quest. In: C. HILLMANN, ed. *Unreal for Mobile and Standalone VR: Create Professional VR Apps Without Coding* [online]. Berkeley, CA: Apress. pp. 141–167. Available from: https://doi.org/10.1007/978-1-4842-4360-2_5 [Accessed 31 Jul 2021].

HIROSE, O., 2020. A Bayesian Formulation of Coherent Point Drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. pp. 1–1.

HORACHE, S., DESCHAUD, J.-E., and GOULETTE, F., 2021. 3D Point Cloud Registration with Multi-Scale Architecture and Unsupervised Transfer Learning. *arXiv:2103.14533 [cs]* [online]. Available from: http://arxiv.org/abs/2103.14533 [Accessed 14 Mar 2022].

HOU, G., DONG, H., and YANG, Y., 2017a. Developing a Virtual Reality Game User Experience Test Method Based on {EEG} Signals. In: *Proc.$\backslash$ Enterprise Systems*. pp. 227–231.

HOU, G., DONG, H., and YANG, Y., 2017b. Developing a Virtual Reality Game User Experience Test Method Based on EEG Signals. In: *2017 5th International Conference on Enterprise Systems (ES)* [online]. Presented at the 2017 5th International Conference on Enterprise Systems (ES). pp. 227–231. Available from: 10/ggzt89.

HTC, 2020. *VIVE^{TM} | Discover Virtual Reality Beyond Imagination* [online]. [online]. Available from: https://www.vive.com/uk/ [Accessed 25 May 2020].

HTC, 2020. *VIVE^{TM} | Discover Virtual Reality Beyond Imagination* [online]. [online]. Available from: https://www.vive.com/uk/ [Accessed 25 May 2020].

HTC, 2021. VIVE Pro 2 Overview | VIVE United Kingdom. *VIVE Pro 2 Headset* [online]. Available from: https://www.vive.com/uk/product/vive-pro2/overview/www.vive.com/uk/product/vive-pro2/overview/ [Accessed 31 Jul 2021].

HUANG, X., MEI, G., and ZHANG, J., 2020. Feature-Metric Registration: A Fast Semi-Supervised Approach for Robust Point Cloud Registration Without Correspondences. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* [online]. Presented at the

2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA: IEEE. pp. 11363–11371. Available from: https://ieeexplore.ieee.org/document/9156354/ [Accessed 27 Jul 2021].

HUANG, X., MEI, G., ZHANG, J., and ABBAS, R., 2021. A comprehensive survey on point cloud registration. *arXiv:2103.02690 [cs]* [online]. Available from: http://arxiv.org/abs/2103.02690 [Accessed 24 Jul 2021].

HUNTER, J.D., 2007. Matplotlib: A 2D Graphics Environment. *Computing in Science Engineering*. 9 (3), pp. 90–95.

JACOBUS, C. and GRIFFIN, J., 1998. Method and system for simulating medical procedures including virtual reality and control method and system for use therein. *US Patent 5,769,640*. p. 12.

JENNETT, C., COX, A.L., CAIRNS, P., DHOPAREE, S., EPPS, A., TIJS, T., and WALTON, A., 2008. Measuring and defining the experience of immersion in games. *International Journal of Human Computer Studies*. 66 (9), pp. 641–661.

JIAN, B. and VEMURI, B.C., 2011. Robust Point Set Registration Using Gaussian Mixture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 33 (8), pp. 1633–1645.

JOHNSTON, A.P.R., RAE, J., ARIOTTI, N., BAILEY, B., LILJA, A., WEBB, R.R.I., FERGUSON, C., MAHER, S., DAVIS, T.P., WEBB, R.R.I., MCGHEE, J., PARTON, R.G., LIJA, A., WEBB, R.R.I., FERGUSON, C., MAHER, S., DAVIS, T.P., WEBB, R.R.I., MCGHEE, J., and PARTON, R.G., 2018. Journey to the centre of the cell: Virtual reality immersion into scientific data. *Traffic*. 19 (2), pp. 105–110.

JOYCE, J., 2003. Bayes' Theorem. [online]. Available from: https://plato.stanford.edu/archives/spr2019/entries/bayes-theorem/ [Accessed 3 May 2020].

KANG, J.M. and PARK, J.B., 2015. Satellite and street maps matching method using iterative Closest Point. In: *2015 15th International Conference on Control, Automation and Systems (ICCAS)*. pp. 684–687.

KEAHEY, T.A., 2013. Using visualization to understand big data. *IBM Business Analytics Advanced Visualisation* [online]. Available from: https://pdfs.semanticscholar.org/3add/2d3d3ecf7641a4cc611deecb68403ed11b33.pdf.

KETELHUT, D., NELSON, B., SCHIFTER, C., and KIM, Y., 2013. Improving Science Assessments by Situating Them in a Virtual Environment. *Education Sciences*. 3 (2), pp. 172–192.

KHAMBADKAR, V. and FOLMER, E., 2013. GIST: a gestural interface for remote nonvisual spatial perception. In: *Proceedings of the 26th annual ACM symposium on User interface software and technology* [online]. New York, NY, USA: Association for Computing Machinery. pp. 301–310. Available from: https://doi.org/10.1145/2501988.2502047 [Accessed 9 Sep 2020].

KIM, K., LAWRENCE, R.L., KYLLONEN, N., LUDEWIG, P.M., ELLINGSON, A.M., and KEEFE, D.F., 2017. Anatomical 2D/3D shape-matching in virtual reality: A user interface for quantifying joint kinematics with radiographic imaging. In: *2017 IEEE Symposium on 3D User Interfaces (3DUI)*. pp. 243–244.

KINOSHITA, K. and NAKAMURA, H., 2005. Identification of the ligand binding sites on the molecular surface of proteins. *Protein Sci*. 14 (3), pp. 711–718.

KIRK, A., 2012. *Data Visualization: a successful design process* [online]. Olton, UNITED KINGDOM: Packt Publishing, Limited. Available from: http://ebookcentral.proquest.com/lib/edgehill/detail.action?docID=1108349 [Accessed 15 Sep 2020].

KIRYS, T.V., FERANCHUK, S.I., TUZIKOV, A.V., and ROCHA, J., 2007. Iterative protein alignment algorithm (IPA). In: *Proc. of the 3-rd Moscow conference on computational molecular biology. øe*. Citeseer. pp. 145–147.

KJELDSKOV, J., 2001. Interaction : Full and Partial Immersive Virtual Reality Displays. *Iris24*. pp. 587–600.

KOCIAN, D.F., 1977. *A Visually-Coupled Airborne Systems Simulator (VCASS) - An Approach to Visual Simulation* [online]. AIR FORCE AEROSPACE MEDICAL RESEARCH LAB WRIGHT-PATTERSON

AFB OH. Available from: https://apps.dtic.mil/sti/citations/ADA039999 [Accessed 25 May 2021].

KONDRA, S., SARKAR, T., RAGHAVAN, V., and XU, W., 2021. Development of a TSR-Based Method for Protein 3-D Structural Comparison With Its Applications to Protein Classification and Motif Discovery. *Frontiers in Chemistry*. 8, p. 602291.

KOSOWSKY, J.J. and YUILLE, A.L., 1994. The invisible hand algorithm: Solving the assignment problem with statistical physics. *Neural Networks*. 7 (3), pp. 477–490.

KOUR, A., 2015. A survey on virtual reality.pdf. *International Journal of Scientific and Research Publications*. 5 (4), pp. 1–8.

KRISHNA, K. and NARASIMHA MURTY, M., 1999. Genetic K-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*. 29 (3), pp. 433–439.

KRUEGER, M.W., GIONFRIDDO, T., and HINRICHSEN, K., 1985. VIDEOPLACE—an artificial reality. In: *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '85* [online]. pp. 35–40. Available from: http://portal.acm.org/citation.cfm?doid=317456.317463.

LACKNER, J.R., 1992. *Simulator sickness* [online]. Available from: http://asa.scitation.org/doi/10.1121/1.404501.

LAHA, B. and BOWMAN, D.A., 2012a. Identifying the Benefits of Immersion in Virtual Reality for Volume Data Visualization. *Workshop Immersive Visualization Revisited – Challenges and Opportunities at IEEE VR 2012*. D (March), pp. 1–2.

LAHA, B. and BOWMAN, D.A., 2012b. Identifying the Benefits of Immersion in Virtual Reality for Volume Data Visualization. *Workshop Immersive Visualization Revisited – Challenges and Opportunities at IEEE VR 2012*. D (March), pp. 1–2.

LAHA, B., BOWMAN, D.A., and SOCHA, J.J., 2014. Effects of VR system fidelity on analyzing isosurface visualization of volume datasets. *IEEE Transactions on Visualization and Computer Graphics*. 20 (4), pp. 513–522.

LAMSON, R.J., 2011. *Virtual Therapy: prevention and treatment of psychiatric conditions by immersion in virtual reality environments*. Dr. Ralph J. Lamson.

LANGLEY, A., LAWSON, G., HERMAWATI, S., CRUZ, M.D., APOLD, J., ARLT, F., and MURA, K., 2016. Establishing the Usability of a Virtual Training System for Assembly Operations within the Automotive Industry. 26 (6), pp. 667–679.

LAU, I., GUPTA, A., and SUN, Z., 2021. Clinical Value of Virtual Reality versus 3D Printing in Congenital Heart Disease. *Biomolecules*. 11 (6), p. 884.

LECUN, Y., BOTTOU, L., BENGIO, Y., and HAFFNER, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 86 (11), pp. 2278–2324.

LEE, H., LEE, K., KO, M., KANG, J., JOO, I., MOON, H., and KIM, K.-M., 2011. 3D breast registration for PET-CT and MR based on surface matching. In: *2011 IEEE Nuclear Science Symposium Conference Record*. pp. 3121–3124.

LEE, S., PARK, K., LEE, J., and KIM, K., 2017. User Study of VR Basic Controller and Data Glove as Hand Gesture Inputs in VR Games. *Proceedings - 2017 International Symposium on Ubiquitous Virtual Reality, ISUVR 2017*. pp. 1–3.

LEVOY, M., RUSINKIEWICZ, S., GINZTON, M., GINSBERG, J., PULLI, K., KOLLER, D., ANDERSON, S., SHADE, J., CURLESS, B., PEREIRA, L., DAVIS, J., and FULK, D., 2000. The Digital Michelangelo Project: 3D Scanning of Large Statues. p. 14.

LEVOY, M., RUSINKIEWICZ, S., GINZTON, M., GINSBERG, J., PULLI, K., KOLLER, D., ANDERSON, S., SHADE, J., CURLESS, B., PEREIRA, L., DAVIS, J., and FULK, D., n.d. The Digital Michelangelo Project: 3D Scanning of Large Statues. p. 14.

LI, C. and LI, Y., 2020. Feasibility Analysis of VR Technology in Physical Education and Sports Training. *IEEE Access*. pp. 1–1.

LI, L., YANG, F., ZHU, H., LI, D., LI, Y., and TANG, L., 2017. An Improved RANSAC for 3D Point Cloud Plane Segmentation Based on Normal Distribution Transformation Cells. *Remote Sensing*. 9 (5), p. 433.

LI, L., ZHU, S., FU, H., TAN, P., and TAI, C.-L., 2020. End-to-End Learning Local Multi-view Descriptors for 3D Point Clouds. *arXiv:2003.05855 [cs]* [online]. Available from: http://arxiv.org/abs/2003.05855 [Accessed 14 Mar 2022].

LIN, M., CHEN, Q., and YAN, S., 2014. Network In Network. *arXiv:1312.4400 [cs]* [online]. Available from: http://arxiv.org/abs/1312.4400 [Accessed 13 Aug 2021].

LITANY, O., REMEZ, T., RODOLÀ, E., BRONSTEIN, A., and BRONSTEIN, M., 2017. Deep Functional Maps: Structured Prediction for Dense Shape Correspondence. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Presented at the 2017 IEEE International Conference on Computer Vision (ICCV). pp. 5660–5668.

LIU, Y., 2006. Automatic registration of overlapping 3D point clouds using closest points. *Image and Vision Computing*. 24 (7), pp. 762–781.

LOBANOV, M.Yu., BOGATYREVA, N.S., and GALZITSKAYA, O.V., 2008. Radius of gyration as an indicator of protein structure compactness. *Molecular Biology*. 42 (4), pp. 623–628.

LUCIA, B., VETTER, M.A., and MOROZ, O., 2021. The Rhetoric of Google Lens: A Postsymbolic Look at Locative Media. *Rhetoric Review*. 40 (1), pp. 75–89.

LUCIANO, L. and BEN HAMZA, A., 2018. Deep learning with geodesic moments for 3D shape classification. *Pattern Recognition Letters*. 105, pp. 182–190.

MACQUEEN, J., 1967. Some methods for classification and analysis of multivariate observations. In: [online]. Presented at the Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, The Regents of the University of California. Available from: https://projecteuclid.org/euclid.bsmsp/1200512992 [Accessed 5 Jun 2020].

MARCUS, A. and VAN DAM, A., 1991. User-Interface Developments for the Nineties. *Computer*. 24 (9), pp. 49–57.

MARKS, B. and THOMAS, J., 2021. Adoption of virtual reality technology in higher education: An evaluation of five teaching semesters in a purpose-designed laboratory. *Education and Information Technologies* [online]. Available from: https://link.springer.com/10.1007/s10639-021-10653-6 [Accessed 14 Aug 2021].

MARKS, S., ESTEVEZ, J.E., and CONNOR, A.M., 2016a. Towards the Holodeck: Fully Immersive Virtual Reality Visualisation of Scientific and Engineering Data. *Proceedings of the 29th International Conference on Image and Vision Computing New Zealand - IVCNZ '14*. pp. 42–47.

MARKS, S., ESTEVEZ, J.E., and CONNOR, A.M., 2016b. Towards the Holodeck: Fully Immersive Virtual Reality Visualisation of Scientific and Engineering Data. [online]. Available from: http://arxiv.org/abs/1604.05797%0Ahttp://dx.doi.org/10.1145/2683405.2683424.

MARLATT, R., 2020. Capitalizing on the Craze of Fortnite: Toward a Conceptual Framework for Understanding How Gamers Construct Communities of Practice. *Journal of Education*. 200 (1), pp. 3–11.

MARTÍNEZ-NAVARRO, J., BIGNÉ, E., GUIXERES, J., ALCAÑIZ, M., and TORRECILLA, C., 2019. The influence of virtual reality in e-commerce. *Journal of Business Research*. 100, pp. 475–482.

MASUDA, T., SAKAUE, K., and YOKOYA, N., 1996. Registration and integration of multiple range images for 3-D model construction. In: *Proceedings of 13th International Conference on Pattern Recognition*. pp. 879–883 vol.1.

MAZURYK, T. and GERVAUTZ, M., 1996a. *Virtual Reality: History, Applications, Technology and Future*. Vienna University of Technology.

MAZURYK, T. and GERVAUTZ, M., 1996b. *Virtual Reality - History, Applications, Technology and Future*.

MCFAUL, H. and FITZGERALD, E., 2020. A realist evaluation of student use of a virtual reality smartphone application in undergraduate legal education. *British Journal of Educational Technology*. 51 (2), pp. 572–589.

MCMAHAN, A., 2003. Immersion, engagement and presence. *The video game theory reader*. 67, p. 86.

MICROSOFT, 2020. *Microsoft HoloLens | Mixed Reality Technology for Business* [online]. [online]. Available from: https://www.microsoft.com/en-us/hololens [Accessed 25 May 2020].

MICROSOFT, 2020. *Language Independence and Language-Independent Components* [online]. [online]. Available from: https://docs.microsoft.com/en-us/dotnet/standard/language-independence-and-language-independent-components [Accessed 18 Jul 2020].

MORA, H., MORA-PASCUAL, J.M., GARCÍA-GARCÍA, A., and MARTÍNEZ-GONZÁLEZ, P., 2016. Computational Analysis of Distance Operators for the Iterative Closest Point Algorithm. *PLOS ONE*. 11 (10), p. e0164694.

MORAN, A., GADEPALLY, V., HUBBELL, M., and KEPNER, J., 2015. Improving Big Data Visual Analytics with Interactive Virtual Reality. In: *Proc High Performance Extreme Computing*.

MOROTTI, E., DONATIELLO, L., and MARFIA, G., 2020. Fostering fashion retail experiences through virtual reality and voice assistants. In: *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. Presented at the 2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW). pp. 338–342.

MUHANNA, M.A., 2015. Virtual reality and the {CAVE}: Taxonomy, interaction challenges and research directions. *Journal of King Saud University - Computer and Information Sciences*. 27 (3), pp. 344–361.

MYRONENKO, A. and SONG, X., 2010. Point Set Registration: Coherent Point Drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 32 (12), pp. 2262–2275.

NABAVI, A., MAMISCH, C.T., GERING, D.T., KACHER, D.F., PERGOLIZZI, R.S., WELLS, W.M., KIKINIS, R., BLACK, P.M., and JOLESZ, F.A., 2000. Image-guided therapy and intraoperative MRI in neurosurgery. *Minimally Invasive Therapy & Allied Technologies*. 9 (3–4), pp. 277–286.

NAGAO, K., YE, Y., WANG, C., FUJISHIRO, I., and MA, K.L., 2017. Enabling interactive scientific data visualization and analysis with see-through hmds and a large tiled display. In: *2016 Workshop on Immersive Analytics, IA 2016* [online]. IEEE. pp. 1–6. Available from: http://ieeexplore.ieee.org/document/7932374/.

NANJAPPAN, V., LIANG, H.-N., LU, F., PAPANGELIS, K., YUE, Y., and MAN, K.L., 2018. User-elicited dual-hand interactions for manipulating 3D objects in virtual reality environments. *Human-centric Computing and Information Sciences*. 8 (1), p. 31.

NEEDLEMAN, S.B. and WUNSCH, C.D., 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*. 48 (3), pp. 443–453.

NI, T., BOWMAN, D.A., NORTH, C., and MCMAHAN, R.P., 2011. Design and evaluation of freehand menu selection interfaces using tilt and pinch gestures. *International Journal of Human-Computer Studies*. 69 (9), pp. 551–562.

NI, T., MCMAHAN, R.P., and BOWMAN, D.A., 2008. Tech-note: rapMenu: Remote menu selection using freehand gestural input. In: *3DUI - IEEE Symposium on 3D User Interfaces 2008*. pp. 55–58.

NÜCHTER, A., ELSEBERG, J., and BORRMANN, D., 2013. Irma3D — An Intelligent Robot for Mapping Applications*. *IFAC Proceedings Volumes*. 46 (29), pp. 119–124.

OCULUS, 2020. *Oculus Rift | Oculus* [online]. [online]. Available from: https://www.oculus.com/rift/?locale=en_GB#oui-csl-rift-games=mages-tale [Accessed 25 May 2020].

OCULUS, 2020. *Oculus Rift | Oculus* [online]. [online]. Available from: https://www.oculus.com/rift/?locale=en_GB#oui-csl-rift-games=mages-tale [Accessed 25 May 2020].

O'LEARY, P., JHAVERI, S., CHAUDHARY, A., SHERMAN, W., MARTIN, K., LONIE, D., WHITING, E., MONEY, J., and MCKENZIE, S., 2017. Enhancements to VTK enabling scientific visualization in immersive environments. In: *Proceedings - IEEE Virtual Reality* [online]. IEEE. pp. 186–194. Available from: http://ieeexplore.ieee.org/document/7892246/.

OVSJANIKOV, M., BEN-CHEN, M., SOLOMON, J., BUTSCHER, A., and GUIBAS, L., 2012. Functional Maps: A Flexible Representation of Maps Between Shapes. p. 11.

ÖZGEN, D.S., AFACAN, Y., and SÜRER, E., 2021. Usability of virtual reality for basic design education: a comparative study with paper-based design. *International Journal of Technology and Design Education*. 31 (2), pp. 357–377.

PAREKH, P.M., KATSELIS, D., BECK, C.L., and SALAPAKA, S.M., 2015. Deterministic annealing for clustering: Tutorial and computational aspects. In: *2015 American Control Conference (ACC)* [online]. Presented at the 2015 American Control Conference (ACC). pp. 2906–2911. Available from: 10/ggtjmt.

PASSIG, D., 2009. Improving the Sequential Time Perception of Teenagers with Mild to Moderate Mental Retardation with 3D Immersive Virtual Reality (IVR). *Journal of Educational Computing Research*. 40 (3), pp. 263–280.

PATEL, A.D., GALLAGHER, A.G., NICHOLSON, W.J., and CATES, C.U., 2006. Learning Curves and Reliability Measures for Virtual Reality Simulation in the Performance Assessment of Carotid Angiography. *Journal of the American College of Cardiology*. 47 (9), pp. 1796–1802.

PEARSON, W.R., 2013. An Introduction to Sequence Similarity (“Homology”) Searching. *Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.]* [online]. 0 3. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3820096/ [Accessed 22 Jul 2020].

PELARGOS, P.E., NAGASAWA, D.T., LAGMAN, C., TENN, S., DEMOS, J.V., LEE, S.J., BUI, T.T., BARNETTE, N.E., BHATT, N.S., UNG, N., BARI, A., MARTIN, N.A., and YANG, I., 2017. *Utilizing virtual and augmented reality for educational and clinical enhancements in neurosurgery* [online]. Churchill Livingstone. Available from: https://www.sciencedirect.com/science/article/pii/S0967586816303162.

PHILLIPS, J.M., LIU, R., and TOMASI, C., 2007. Outlier Robust ICP for Minimizing Fractional RMSD. In: *Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007)* [online]. Montreal, QC, Canada: IEEE. pp. 427–434. Available from: http://ieeexplore.ieee.org/document/4296784/ [Accessed 24 Jun 2019].

POIESI, F. and BOSCAINI, D., 2021. Generalisable and distinctive 3D local deep descriptors for point cloud registration. *arXiv:2105.10382 [cs]* [online]. Available from: http://arxiv.org/abs/2105.10382 [Accessed 14 Mar 2022].

POLYS, N.F., BOWMAN, D.A., and NORTH, C., 2011. The role of Depth and Gestalt cues in information-rich virtual environments. *International Journal of Human Computer Studies*. 69 (1–2), pp. 30–51.

POMERLEAU, F., COLAS, F., SIEGWART, R., and MAGNENAT, S., 2013. Comparing ICP variants on real-world data sets. *Autonomous Robots*. 34 (3), pp. 133–148.

POMERLEAU, F., LIU, M., COLAS, F., and SIEGWART, R., 2012. Challenging data sets for point cloud registration algorithms. *The International Journal of Robotics Research*. 31 (14), pp. 1705–1711.

PRADEEP RAJ, K.B., OZA, P., and LAHIRI, U., 2017. Gaze-sensitive Virtual Reality based Social Communication Platform for Individuals with Autism. *IEEE Transactions on Affective Computing*.

PULLI, K., 1999. Multiview registration for large data sets. In: *Second International Conference on 3-D Digital Imaging and Modeling (Cat. No.PR00062)* [online]. Presented at the Second International Conference on 3-D Digital Imaging and Modeling (Cat. No.PR00062). pp. 160–168. Available from: 10/ddk4bh.

PUTRINO, D., WONG, Y.T., WEISS, A., and PESARAN, B., 2015. A training platform for many-dimensional prosthetic devices using a virtual reality environment. *Journal of neuroscience methods*. 244, pp. 68–77.

QI, C.R., SU, H., MO, K., and GUIBAS, L.J., 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *arXiv:1612.00593 [cs]* [online]. Available from: http://arxiv.org/abs/1612.00593 [Accessed 25 Apr 2020].

QI, C.R., YI, L., SU, H., and GUIBAS, L.J., 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In: I. GUYON, U.V. LUXBURG, S. BENGIO, H. WALLACH, R. FERGUS, S. VISHWANATHAN and R. GARNETT, eds. *Advances in Neural Information Processing Systems 30* [online]. Curran Associates, Inc. pp. 5099–5108. Available from: http://papers.nips.cc/paper/7095-pointnet-deep-hierarchical-feature-learning-on-point-sets-in-a-metric-space.pdf [Accessed 25 Apr 2020].

RAGAN, E.D., BOWMAN, D.A., KOPPER, R., STINSON, C., SCERBO, S., and MCMAHAN, R.P., 2015. Effects of field of view and visual complexity on virtual reality training effectiveness for a visual scanning task. *IEEE Transactions on Visualization and Computer Graphics*. 21 (7), pp. 794–807.

RAHMAN, M., GUSTAFSON, S., IRANI, P., and SUBRAMANIAN, S., 2009. Tilt techniques: investigating the dexterity of wrist-based input. In: *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09* [online]. New York, New York, USA: ACM Press. p. 1943. Available from: http://portal.acm.org/citation.cfm?id=1518701.1518997%5Cnhttp://portal.acm.org/citation.cfm?doid=1518701.1518997%5Cnhttp://dl.acm.org/citation.cfm?doid=1518701.1518997.

RAHMAN, M., LI, X., and YIN, X., 2019. DL-RANSAC: An Improved RANSAC with Modified Sampling Strategy Based on the Likelihood. In: *2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC)*. Presented at the 2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC). pp. 463–468.

RAJ, K.B.P. and LAHIRI, U., 2017. Virtual reality based social communication platform: Implications on performance and eye gaze. *8th International Conference on Computing, Communications and Networking Technologies, ICCCNT 2017*.

RANGARAJAN, A., CHUI, H., and BOOKSTEIN, F.L., 1997. The softassign Procrustes matching algorithm. In: J. DUNCAN and G. GINDI, eds. *Information Processing in Medical Imaging*. Berlin, Heidelberg: Springer. pp. 29–42.

RCSB PROTEIN DATA, 2020. *RCSB PDB* [online]. [online]. Available from: https://www.rcsb.org/search?request=%7B%22query%22%3A%7B%22type%22%3A%22group%22%2C%22logical_operator%22%3A%22and%22%2C%22nodes%22%3A%5B%7B%22type%22%3A%22terminal%22%2C%22service%22%3A%22structure%22%2C%22parameters%22%3A%7B%22value%22%3A%7B%22entry_id%22%3A%222JKG%22%2C%22assembly_id%22%3A%221%22%7D%2C%22operator%22%3A%22strict_shape_match%22%7D%2C%22node_id%22%3A0%7D%5D%2C%22label%22%3A%22query-builder%22%7D%2C%22return_type%22%3A%22assembly%22%2C%22request_options%22%3A%7B%22pager%22%3A%7B%22start%22%3A0%2C%22rows%22%3A100%7D%2C%22scoring_strategy%22%3A%22combined%22%2C%22sort%22%3A%5B%7B%22sort_by%22%3A%22score%22%2C%22direction%22%3A%22desc%22%7D%5D%7D%2C%22request_info%22%3A%7B%22src%22%3A%22ui%22%2C%22query_id%22%3A%225e86be7eeb70248a4db9136c70ce39be%22%7D%7D [Accessed 22 Jul 2020].

REN, G. and O'NEILL, E., 2013. 3D selection with freehand gesture. *Computers and Graphics (Pergamon)*. 37 (3), pp. 101–120.

ROBERTS, D.J., GARCIA, A.S., DODIYA, J., WOLFF, R., FAIRCHILD, A.J., and FERNANDO, T., 2015. Collaborative telepresence workspaces for space operation and science. In: *2015 IEEE Virtual Reality (VR)*. Presented at the 2015 IEEE Virtual Reality (VR). pp. 275–276.

VAN ROSSUM, G., 1995. Python tutorial. [online]. Available from: https://ir.cwi.nl/pub/5007 [Accessed 23 Jul 2020].

RUSINKIEWICZ, S., 2019. A symmetric objective function for ICP. *ACM Transactions on Graphics*. 38 (4), pp. 1–7.

RUSINKIEWICZ, S. and LEVOY, M., 2001a. Efficient variants of the ICP algorithm. In: *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*. pp. 145–152.

RUSINKIEWICZ, S. and LEVOY, M., 2001b. Efficient variants of the ICP algorithm. In: *Proceedings Third International Conference on 3-D Digital Imaging and Modeling* [online]. Presented at the Proceedings Third International Conference on 3-D Digital Imaging and Modeling. pp. 145–152. Available from: 10/b4jsst.

RUSU, R.B., MARTON, Z.C., BLODOW, N., and BEETZ, M., 2008. Learning informative point classes for the acquisition of object model maps. In: *Robotics and Vision 2008 10th International Conference on Control, Automation*. Presented at the Robotics and Vision 2008 10th International Conference on Control, Automation. pp. 643–650.

RYAN, S.G., BUTLER, M.N., ADEYEMI, S.S., KALBER, T., PATRICK, P.S., ZAW THIN, M., HARRISON, I.F., STUCKEY, D.J., PULE, M., and LYTHGOE, M.F., 2019. Imaging of X-Ray-Excited Emissions from Quantum Dots and Biological Tissue in Whole Mouse. *Scientific Reports*. 9 (1), p. 19223.

SABOORI, E., PARSAZAD, S., and SADEGHI, A., 2010. Improving the K-means algorithm using improved downhill simplex search. In: *2010 2nd International Conference on Software Technology and Engineering* [online]. Presented at the 2010 2nd International Conference on Software Technology and Engineering. pp. V2-350-V2-354. Available from: 10/bc22s3.

SALTI, S., TOMBARI, F., and DI STEFANO, L., 2014. SHOT: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*. 125, pp. 251–264.

SAMSUNG, 2020. Samsung Gear VR with Controller. *The Official Samsung Galaxy Site* [online]. Available from: http://www.samsung.com/global/galaxy/gear-vr/ [Accessed 26 May 2020].

SCHMITT, S., KUHN, D., and KLEBE, G., 2002a. A New Method to Detect Related Function Among Proteins Independent of Sequence and Fold Homology. *Journal of Molecular Biology*. 323 (2), pp. 387–406.

SCHMITT, S., KUHN, D., and KLEBE, G., 2002b. A New Method to Detect Related Function Among Proteins Independent of Sequence and Fold Homology. *Journal of Molecular Biology*. 323 (2), pp. 387–406.

SCHNACK, A., WRIGHT, M.J., and HOLDERSHAW, J.L., 2018. Immersive virtual reality technology in a three-dimensional virtual simulated store: Investigating telepresence and usability. *Food Research International*.

SCHROEDER, W.J., MARTIN, K.M., and LORENSEN, W.E., 1996. The design and implementation of an object-oriented toolkit for 3D graphics and visualization. In: *Proceedings of Seventh Annual IEEE Visualization '96*. Presented at the Proceedings of Seventh Annual IEEE Visualization '96. pp. 93–100.

SCHUTZ, C., JOST, T., and HUGLI, H., 1998. Multi-feature matching algorithm for free-form 3D surface registration. In: *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170)* [online]. Presented at the Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170). pp. 982–984 vol.2. Available from: 10.1109/ICPR.1998.711852.

SEIBERT, J. and SHAFER, D.M., 2017. Control mapping in virtual reality: effects on spatial presence and controller naturalness. *Virtual Reality*. 22 (1), pp. 79–88.

SHARP, G.C., LEE, S.W., and WEHE, D.K., 2002. ICP registration using invariant features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 24 (1), pp. 90–102.

SHU, Y., HUANG, Y.-Z., CHANG, S.-H., and CHEN, M.-Y., 2019. Do virtual reality head-mounted displays make a difference? A comparison of presence and self-efficacy between head-mounted displays and desktop computer-facilitated virtual environments. *Virtual Reality*. 23 (4), pp. 437–446.

SHULMAN-PELEG, A., NUSSINOV, R., and WOLFSON, H.J., 2004. Recognition of Functional Sites in Protein Structures. *Journal of Molecular Biology*. 339 (3), pp. 607–633.

SHULMAN-PELEG, A., SHATSKY, M., NUSSINOV, R., and WOLFSON, H.J., 2008. MultiBind and MAPPIS: webservers for multiple alignment of protein 3D-binding sites and their interactions. *Nucleic Acids Res*. 36 (suppl_2), pp. W260–W264.

SIMARD, P., STEINKRAUS, D., and PLATT, J., 2003. Best Practices for Convolutional Neural Networks.

SIMON, D.A., 1996. Fast and Accurate Shape-Based Registration. p. 216.

SINKO, M., KAMENCAY, P., HUDEC, R., and BENCO, M., 2018a. 3D registration of the point cloud data using ICP algorithm in medical image analysis. In: *2018 ELEKTRO*. Presented at the 2018 ELEKTRO. pp. 1–6.

SINKO, M., KAMENCAY, P., HUDEC, R., and BENCO, M., 2018b. 3D registration of the point cloud data using ICP algorithm in medical image analysis. In: *2018 ELEKTRO*. pp. 1–6.

SLATER, M., 2003. A Note on Presence Terminology. *Emotion*. 3, pp. 1–5.

SLATER, M., LINAKIS, V., USOH, M., and KOOPER, R., 1996. Immersion, presence, and performance in virtual environments: an experiment with tri-dimensional chess. *Proceedings of the 3rd ACM Symposium on Virtual Reality Software and Technology (VRST 1996), Hong Kong, China*. (JUNE), pp. 163–172.

SLATER, M. and SANCHEZ-VIVES, M.V., 2016. Enhancing Our Lives with Immersive Virtual Reality. *Frontiers in Robotics and AI*. 3, p. 74.

SOUSA SANTOS, B., DIAS, P., PIMENTEL, A., BAGGERMAN, J.-W., FERREIRA, C., SILVA, S., and MADEIRA, J., 2009. Head-mounted display versus desktop for 3D navigation in virtual reality: a user study. *Multimedia Tools and Applications*. 41 (1), p. 161.

SPAGNOLLI, A. and GAMBERINI, L., 2002. Immersion/Emersion: Presence in hybrid environments 1 Visibility of the technical medium and emersion. *Presence 2002: Fifth Annual International Workshop*. pp. 1–14.

SRIVASTAVA, S., LAL, S.B., MISHRA, D.C., ANGADI, U.B., CHATURVEDI, K.K., RAI, S.N., and RAI, A., 2016. An efficient algorithm for protein structure comparison using elastic shape analysis. *Algorithms for Molecular Biology* [online]. 11 (1). Available from: http://link.gale.com/apps/doc/A465000599/AONE?u=edge&sid=zotero&xid=5941e4af [Accessed 12 Aug 2020].

STEED, C.A., DANIEL, J., DROUHARD, M., HAHN, S., and PROFFEN, T., 2017. Immersive visual analytics for transformative neutron scattering science. In: *2016 Workshop on Immersive Analytics, IA 2016*. IEEE. pp. 38–43.

STEPTOE, W., STEED, A., and SLATER, M., 2013a. Human Tails: Ownership and Control of Extended Humanoid Avatars. *IEEE Transactions on Visualization and Computer Graphics*. 19 (4), pp. 583–590.

STEPTOE, W., STEED, A., and SLATER, M., 2013b. Human tails: Ownership and control of extended humanoid avatars. *IEEE Transactions on Visualization and Computer Graphics*. 19 (4), pp. 583–590.

SUTHERLAND, I.E., 1965. The Ultimate Display. In: *Proceedings of the IFIP Congress*. pp. 506–508.

TABEI, Y. and PUGLISI, S.J., 2017. Scalable Similarity Search for Molecular Descriptors. *arXiv:1611.10045 [cs]* [online]. Available from: http://arxiv.org/abs/1611.10045 [Accessed 27 Oct 2019].

THE 3D TOOLKIT, 2020. *3DTK - The 3D Toolkit* [online]. [online]. Available from: http://slam6d.sourceforge.net/ [Accessed 3 Aug 2020].

TOET, A., JANSEN, S.E.M., and DELLEMAN, N.J., 2007. Effects of field-of-view restrictions on speed and accuracy of manoeuvring. *Perceptual and Motor Skills*. 105 (3 Pt 2), pp. 1245–1256.

TOKUSHO, Y. and FEINER, S., 2009. Prototyping an Outdoor Mobile Augmented Reality Street View Application. In: *in Int'l Symp. Mixed and Augmented Reality (ISMAR*.

TOMA, M.I., GÎRBACIA, F., and ANTONYA, C., 2012. A comparative evaluation of human interaction for design and assembly of 3D CAD models in desktop and immersive environments. *International Journal on Interactive Design and Manufacturing*. 6 (3), pp. 179–193.

TOMBARI, F., SALTI, S., and STEFANO, L.D., 2010. Unique signatures of histograms for local surface description. In: *In Proc.of the European Conf. onComputer Vision(ECCV*.

TREDINNICK, R., BROECKER, M., and PONTO, K., 2016. Progressive feedback point cloud rendering for virtual reality display. In: *2016 IEEE Virtual Reality (VR)* [online]. Presented at the 2016 IEEE Virtual Reality (VR). pp. 301–302. Available from: 10.1109/VR.2016.7504773.

TSIN, Y. and KANADE, T., 2004. A Correlation-Based Approach to Robust Point Set Registration. In: T. PAJDLA and J. MATAS, eds. *Computer Vision - ECCV 2004* [online]. Berlin, Heidelberg: Springer. pp. 558–569. Available from: 10.1007/978-3-540-24672-5_44.

TURK, G. and LEVOY, M., 1994. Zippered polygon meshes from range images. In: *Proceedings of the 21st annual conference on Computer graphics and interactive techniques* [online]. New York, NY, USA: Association for Computing Machinery. pp. 311–318. Available from: https://doi.org/10.1145/192161.192241 [Accessed 22 Apr 2020].

UNITY TECHNOLOGIES, 2019a. *Unity* [online]. Available from: https://Unity.com [Accessed 18 Jan 2020].

UNITY TECHNOLOGIES, 2019b. *Unity* [online]. Available from: https://Unity.com [Accessed 18 Jan 2020].

UNITY TECHNOLOGIES, 2020. *Unity - Scripting API: Collider* [online]. [online]. Available from: https://docs.unity3d.com/ScriptReference/Collider.html [Accessed 4 Aug 2020].

UNIVERSITY OF MICHIGAN, 2021. UM-VRL: Virtual Prototyping of a Sailing Yacht. *Virtual Prototyping of a Sailing Yacht* [online]. Available from: http://www.umich.edu/~vrl/project/yacht/ [Accessed 27 Jun 2021].

USC SCHOOL OF CINEMATIC ARTS, 2021. USC Hugh M. Hefner Moving Image Archive » Morton Heilig: Photos. *MORTON HEILIG : THE FATHER OF VIRTUAL REALITY* [online]. Available from: https://www.uschefnerarchive.com/morton-heilig-photos/ [Accessed 27 Jun 2021].

VAN ROSSUM, G. and DRAKE, F.L., 2009. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace.

WAHBA, G. and PENNSYLVANIA, P., 1990. *Spline Models for Observational Data. Society for Industrial and Applied Mathematics*.

WALLACE, A.C., BORKAKOTI, N., and THORNTON, J.M., 1997. Tess: A geometric hashing algorithm for deriving 3D coordinate templates for searching structural databases. Application to enzyme active sites. *Protein Science*. 6 (11), pp. 2308–2323.

WANG, F. and ZHAO, Z., 2017. A survey of iterative closest point algorithm. In: *2017 Chinese Automation Congress (CAC)*. pp. 4395–4399.

WANG, Q.H. and LI, J.R., 2004. A desktop VR prototype for industrial training applications. *Virtual Reality*. 7 (3–4), pp. 187–197.

WANG, Y. and SOLOMON, J.M., 2019. Deep Closest Point: Learning Representations for Point Cloud Registration. *arXiv:1905.03304 [cs]* [online]. Available from: http://arxiv.org/abs/1905.03304 [Accessed 7 Feb 2020].

WATSON, B. and LUEBKE, D., 2005. The ultimate display: Where will all the pixels come from? In: *Computer* [online]. pp. 54–61. Available from: http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.136.3720 http://ieeexplore.ieee.org/document/1492267/.

WEIK, S., 1997. Registration of 3-D partial surface models using luminance and depth information. In: *Proceedings. International Conference on Recent Advances in 3-D Digital Imaging and Modeling (Cat. No.97TB100134)* [online]. Presented at the Proceedings. International Conference on Recent Advances in 3-D Digital Imaging and Modeling (Cat. No.97TB100134). pp. 93–100. Available from: 10/cbtgmn.

WEISSTEIN, E.W., 2020. Hyperboloid. *https://mathworld.wolfram.com/Hyperboloid.html* [online]. Available from: https://mathworld.wolfram.com/Hyperboloid.html [Accessed 8 Aug 2020].

WESKAMP, N., KUHN, D., HÜLLERMEIER, E., and KLEBE, G., 2004. Efficient similarity search in protein structure databases by k-clique hashing. *Bioinformatics*. 20 (10), pp. 1522–1526.

WLOKA, M.M., 1995. Interacting with virtual reality. In: J. RIX, S. HAAS and J. TEIXEIRA, eds. *Virtual Prototyping: Virtual environments and the product design process*. pp. 199–212.

WOIŃSKA, M., GRABOWSKY, S., DOMINIAK, P.M., WOŹNIAK, K., and JAYATILAKA, D., 2016. Hydrogen atoms can be located accurately and precisely by x-ray crystallography. *Science Advances* [online]. 2 (5). Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4928899/ [Accessed 12 Aug 2020].

WOLD, S., ESBENSEN, K., and GELADI, P., 1987. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*. 2 (1), pp. 37–52.

WU, M.L. and POPESCU, V., 2017. *Efficient VR and AR Navigation through Multiperspective Occlusion Management*.

WU, Z., SONG, S., KHOSLA, A., YU, F., ZHANG, L., TANG, X., and XIAO, J., 2015. 3D ShapeNets: A Deep Representation for Volumetric Shapes. In: [online]. pp. 1912–1920. Available from: https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Wu_3D_ShapeNets_A_2015_CVPR_paper.html [Accessed 9 Feb 2020].

XENOS, M., MARATOU, V., NTOKAS, I., METTOURIS, C., and PAPADOPOULOS, G.A., 2017. Game-based learning using a 3D virtual world in computer engineering education. In: *IEEE Global Engineering Education Conference, EDUCON*. IEEE. pp. 1078–1083.

YANG, J., LI, H., CAMPBELL, D., and JIA, Y., 2016. Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 38 (11), pp. 2241–2254.

YEW, Z.J. and LEE, G.H., 2020. RPM-Net: Robust Point Matching using Learned Features. [online]. Available from: https://www.arxiv-vanity.com/papers/2003.13479/ [Accessed 24 Apr 2020].

YI, D.-C., CHEN, Y.-S., HAN, P.-H., WANG, H.-C., and HUNG, Y.-P., 2019. Archaeological Excavation Simulation for Interaction in Virtual Reality. In: *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. Presented at the 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). pp. 1249–1250.

YU, Y., DA, F., and GUO, Y., 2019. Sparse ICP With Resampling and Denoising for 3D Face Verification. *IEEE Transactions on Information Forensics and Security*. 14 (7), pp. 1917–1927.

YUILLE, A.L. and GRZYWACZ, N.M., 1988. The Motion Coherence Theory. In: *[1988 Proceedings] Second International Conference on Computer Vision* [online]. Presented at the [1988 Proceedings] Second International Conference on Computer Vision. pp. 344–353. Available from: 10/ddq9gg.

ZENG, A., SONG, S., NIESSNER, M., FISHER, M., XIAO, J., and FUNKHOUSER, T., 2017. 3DMatch: Learning Local Geometric Descriptors From RGB-D Reconstructions. In: [online]. Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1802–1811. Available from: https://openaccess.thecvf.com/content_cvpr_2017/html/Zeng_3DMatch_Learning_Local_CVPR_2017_paper.html [Accessed 27 Jul 2021].

ZHANG, J., YAO, Y., and DENG, B., 2021. Fast and Robust Iterative Closest Point. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. pp. 1–1.

ZHANG, S., DEMIRALP, C., KEEFE, D.F.F., DASILVA, M., LAIDLAW, D.H.H., GREENBERG, B.D.D., BASSER, P.J.J., PIERPAOLI, C., CHIOCCA, E.A.A., DEISBOECK, T.S.S., DEMIRALP, Ç., KEEFE, D.F.F., DASILVA, M., LAIDLAW, D.H.H., GREENBERG, B.D.D., BASSER, P.J.J., PIERPAOLI, C., CHIOCCA, E.A.A., and DEISBOECK, T.S.S., 2001. An immersive virtual environment for DT-MRI volume visualization applications: a case study. *Proceedings Visualization, 2001. VIS '01.* Vi, pp. 437–584.

ZHANG, Z., 1992. Iterative point matching for registration of free-form curves. report. [online]. INRIA. Available from: https://hal.inria.fr/inria-00074899 [Accessed 19 Jun 2021].

ZHAVORONKOV, A., ZAGRIBELNYY, B., ZHEBRAK, A., ALADINSKIY, V., TERENTIEV, V., VANHAELEN, Q., BEZRUKOV, D.S., POLYKOVSKIY, D., SHAYAKHMETOV, R., FILIMONOV, A., BISHOP, M., MCCLOSKEY, S., LEIJA, E., BRIGHT, D., FUNAKAWA, K., LIN, Y.-C., HUANG, S.-H., LIAO, H.-J., ALIPER, A., and IVANENKOV, Y., 2020. *Potential Non-Covalent SARS-CoV-2 3C-like Protease Inhibitors Designed Using Generative Deep Learning Approaches and Reviewed by Human Medicinal Chemist in Virtual Reality* [online]. preprint. Available from: https://chemrxiv.org/articles/Potential_Non-Covalent_SARS-CoV-2_3C-like_Protease_Inhibitors_Designed_Using_Generative_Deep_Learning_Approaches_and_Reviewed_by_Human_Medicinal_Chemist_in_Virtual_Reality/12301457 [Accessed 19 Jun 2021].

ZHOU, N.N. and DENG, Y.L., 2009. Virtual reality: A state-of-the-art survey. *International Journal of Automation and Computing*. 6 (4), pp. 319–325.

ZHOU, Y., JIANG, J., and ZHENG, H., 2009. 3D Protein Structure Similarity Comparison Using a Angular-Invariant Feature Registration Method. In: *2009 3rd International Conference on Bioinformatics and Biomedical Engineering*. Presented at the 2009 3rd International Conference on Bioinformatics and Biomedical Engineering. pp. 1–4.

ZIELASKO, D., WEYERS, B., BELLGARDT, M., PICK, S., MEIBNER, A., VIERJAHN, T., and KUHLEN, T.W., 2017. Remain seated: towards fully-immersive desktop VR. In: *2017 IEEE 3rd Workshop on Everyday Virtual Reality (WEVR)* [online]. Presented at the 2017 IEEE 3rd Workshop on Everyday Virtual Reality (WEVR). pp. 1–6. Available from: 10/ggx7gp.

ZIELASKO, D., WEYERS, B., BELLGARDT, M., PICK, S., MEI\S SNER, A., VIERJAHN, T., and KUHLEN, T.W., 2017. Remain seated: Towards fully-immersive desktop {VR}. In: *Proc.$\backslash$ Everyday Virtual Reality*. pp. 1–6.

ZITOVÁ, B. and FLUSSER, J., 2003. Image registration methods: a survey. *Image and Vision Computing*. 21 (11), pp. 977–1000.

9 Appendix

Sensorama – In the years 1960 - 1962, Morton Heilig created a virtual reality system (Figure 9.1) that played a pre-recorded film in colour. The user’s senses were stimulated appropriately when they encountered virtual environments with sound generated by the connected stereophonic sound system, smell generated by odour emitters, wind generated by a fan and vibration by the motion chair the user sits on. This was the first approach to creating a VR system even though it was not interactive. (Mazuryk and Gervautz, 1996).

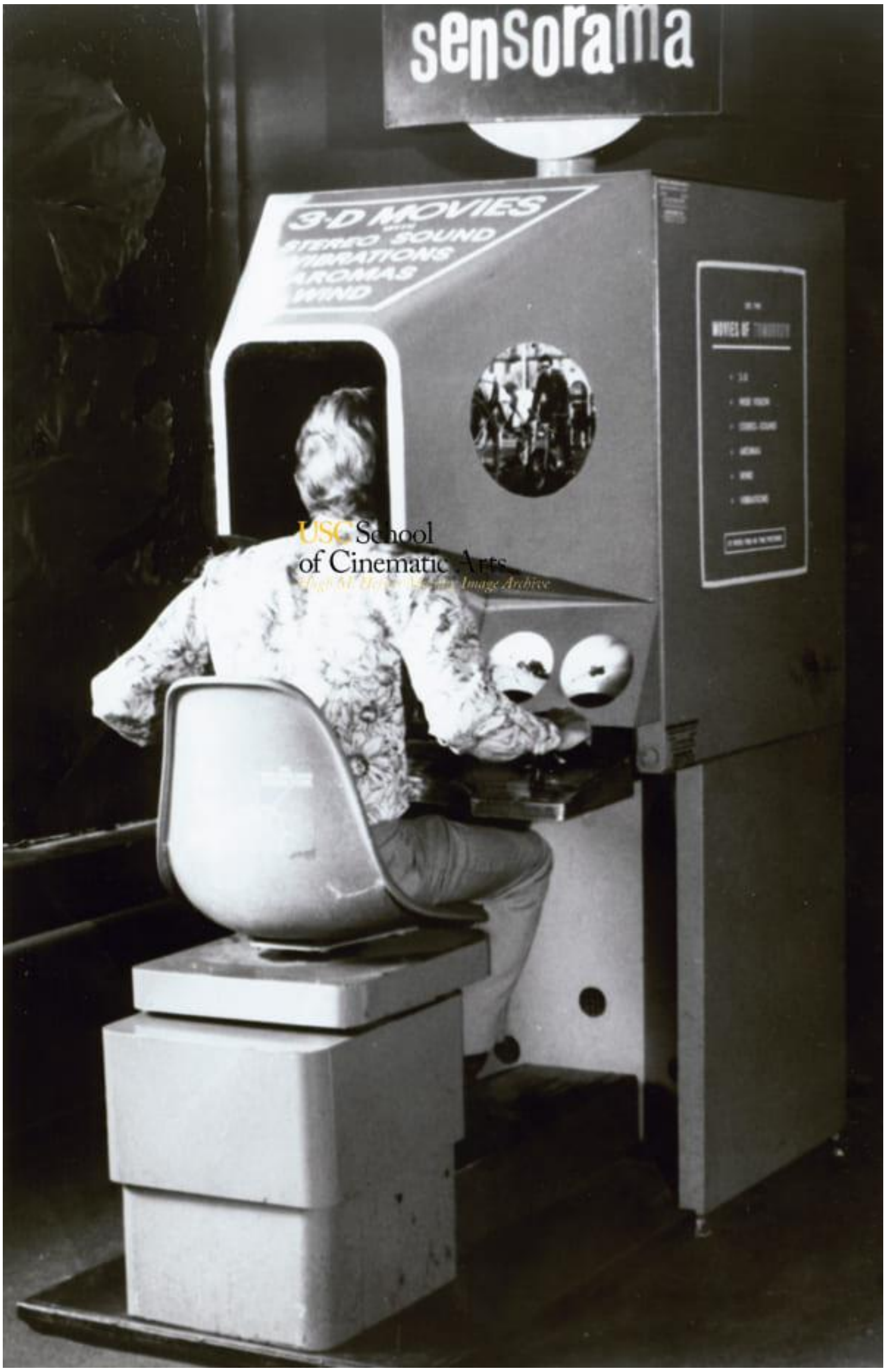


Figure 9.1 Morton Heilig's Sensorama. (Heilig, 1992). The early virtual reality machine allowed users to experience some interaction with the movie that was being watched. The aim was to add a sense of taste, smell, and feel to the hearing and vision (USC School of Cinematic Arts, 2021)

The Ultimate Display – in 1965, Ivan Sutherland proposed a concept of a virtual world that included interaction, force-feedback, sound, smell, and taste (Figure 9.2) (Mazuryk and Gervautz, 1996). Ivan Sutherland then created the first hardware-based VR device, which was a heavy head-mounted display that hanged from the ceiling whilst in use. As a result of its looks and weight, It derived its name, 'The Sword of Damocles' from the sword above the throne of the Greek character Damocles. (Mazuryk and Gervautz, 1996). The Sword of Damocles featured head tracking and a stereo view (presenting two horizontally shifted views of a scene to the users' eyes) that was updated correctly according to the user's head position and orientation. (Mazuryk and Gervautz, 1996).

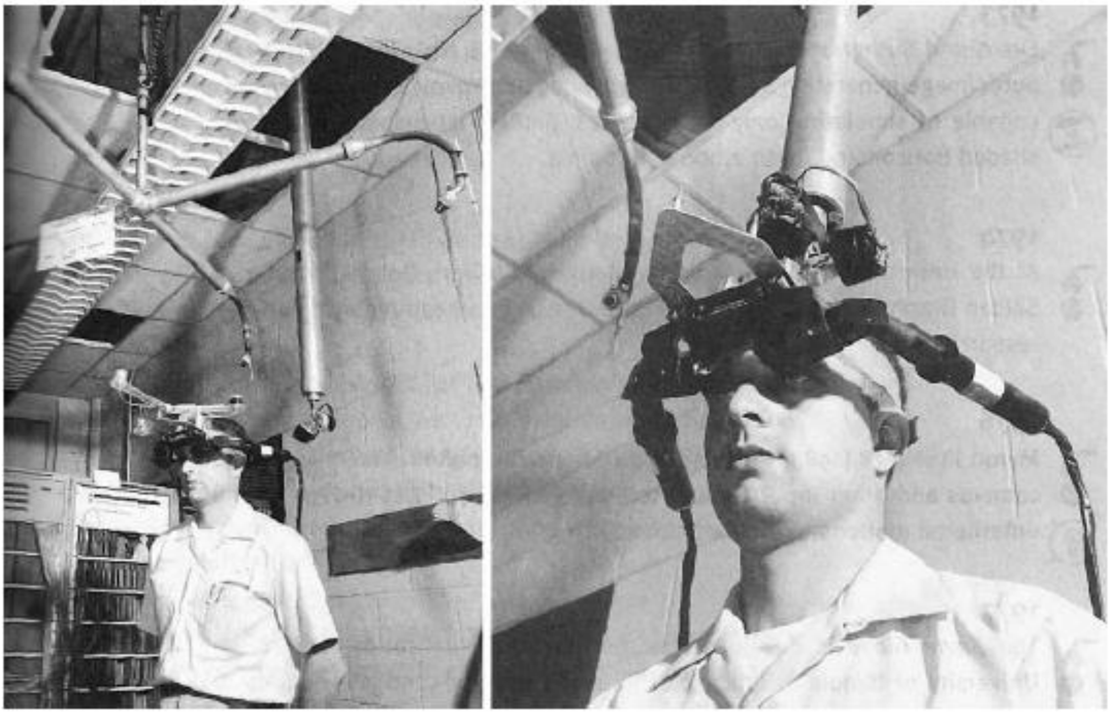


Figure 9.2 The Sword of Damocles (Sutherland, 1968). The heavy mounted setup from the ceiling connecting the Head Mounted Display (Black part facing the eyes) featured head tracking and stereo views that adjusted based on the user's head and orientation.

GROPE – the prototype of a force-feedback system (Figure 9.3) (using motorised components such as a hand controller to simulate reactions to the user such as vibrations from touching virtual objects) built at the University of North Carolina (UNC) in 1971. (Mazuryk and Gervautz, 1996).

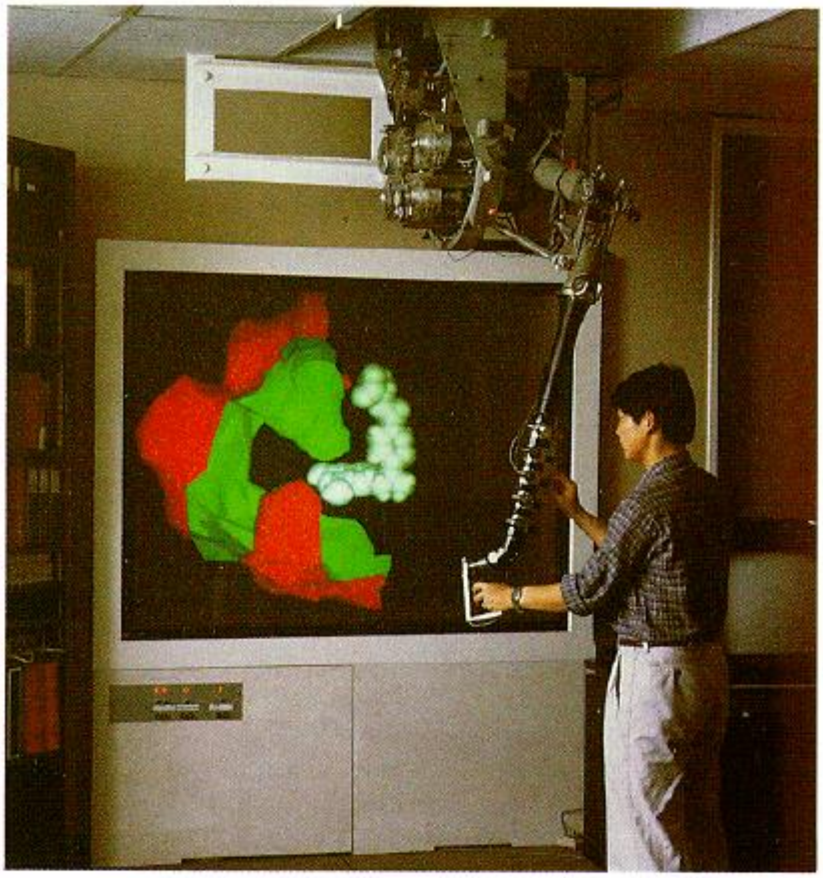


Figure 9.3 prototype force-feedback system (Brooks et al., 1990). Motorised components are used to simulate feedback such as vibrations. It was developed to provide a means for interaction with protein molecules and the discovery of drugs.

VIDEOPLACE (Figure 9.4) – This was originally designed and developed as an environment enabling persons in different geographical locations to experience the same video as an advancement to human-computer interactions. It allowed the silhouettes of users to be captured from a camera and projected onto a large screen by a projector in real time. Users could see their images on the large screen and could move to see the projected silhouettes update.

Other graphic objects were added to the screen and this enabled interaction between the users, using their silhouettes to interact with the other objects on the screen or amongst other user's silhouettes. (Krueger, Gionfriddo and Hinrichsen, 1985, Mazuryk and Gervautz, 1996)

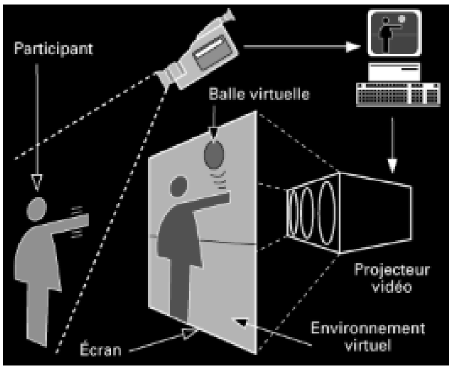


Figure 9.4 VIDEOPLACE. (Krueger, Gionfriddo and Hinrichsen, 1985). The setup captures the silhouettes of users from mounted cameras and displays them as avatars on the screen. Other objects can be captured and placed on the screen as well, allowing the users to use their avatar to control those objects.

VCASS – The Visually Coupled Air-borne System (Figure 3.5) was a flight simulator developed by Thomas Furness that featured a head-mounted display that enhanced the outside view of the fighter pilot view with optimal flight path information. (Kocian, 1977, Mazuryk and Gervautz, 1996).

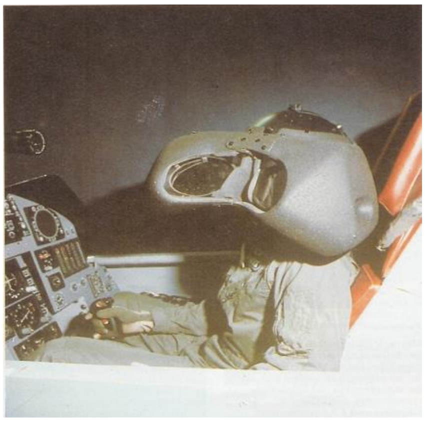


Figure 9.5 The Visually Coupled Airborne Systems Simulator (VCASS). (Kocian, 1977). The simulator was developed for the United States Airforce to present augmented information about the flight in the head-mounted display. This was to visually couple multiple information into a single view that is easily seen by the pilot in different head rotations.

VIVED – Virtual environment visual display (Figure 9.6) was built in 1984 at NASA Ames Research Centre. The system consisted of a stereoscopic HMD and other components built from customising commercially available components. (Fisher et al., 1987, Mazuryk and

Gervautz, 1996).

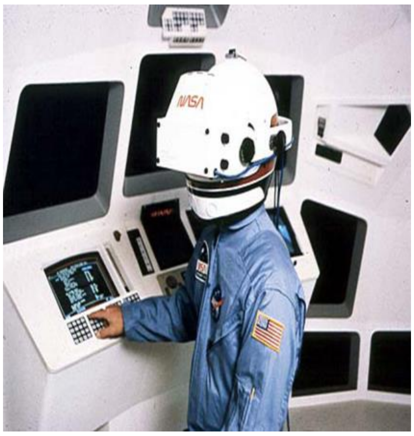


Figure 9.6 Virtual Environment Display System. The system consisted of a head-mounted display that allowed the user to interact with their complex tasks and increase their situational awareness with the ability to see such statuses in the head-mounted display.

VPL DataGlove– VPL developed the DataGlove and the first commercially available VR device, the Eyephone (Figure 9.7). (Clark et al., 1989, Mazuryk and Gervautz, 1996)

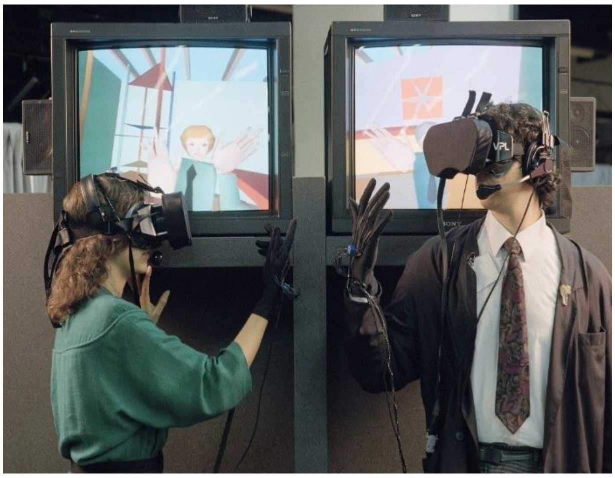


Figure 9.7 The Eyephone HMD and the DataGlove hand tracking (Clark et al., 1989). The system was composed of an HMD and a DataGlove hand tracking device that was original designed to control the Utah and MIT dextrous hand using a VPL DataGlove.

BOOM (Binocular Omni Oriented Monitors) (Figure 9.8) – This consisted of a small box that had two CRT monitors inside and holes for the user to look through. The setup allowed the user to keep the box by the eyes and navigate the virtual world whilst a mechanical arm provided the position and orientation of the box. The Boom was commercialised in 1989 by Fake Space Labs.

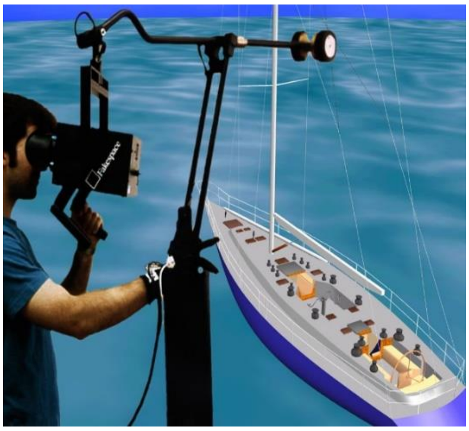


Figure 9.8 The Binocular Omni Oriented Monitors (BOOM) is a head display device developed by FakeSpace Labs. The optical part of the device is in a box that is connected to a multi-link arm. The user can see the virtual world by looking through the box. The user can also adjust the box to see other parts of the virtual world. (University of Michigan, 2021)

UNC Walkthrough project (Figure 9.9) – In the later part of the 1980s, the University of North Carolina developed a virtual architecture allowing users to experience it by walking through. The system was enhanced using VR devices such as several HMDs, optical trackers and the Pixel-Plane graphics engine. (Brooks, 1987, Mazuryk and Gervautz, 1996).

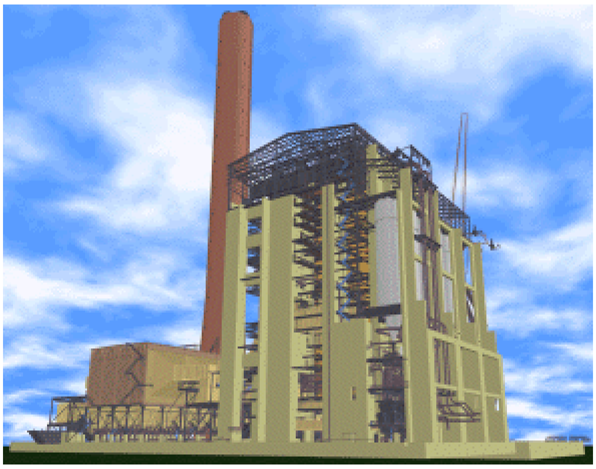


Figure 9.9 The Walkthrough Project (Brooks, 1987) was a virtual environment for architects and their clients to virtually explore and refine design specifications before the physical building is developed. It allowed quick prototypes for building designs to be made. [Image source: http://www.cs.unc.edu/~walk/] [Accessed: 16/06/2020]

Virtual Wind Tunnel –The Virtual Wind Tunnel (Figure 9.10) was an application that allowed the study of flow fields (the spread of the density and velocity of fluid over space and time). The virtual wind tunnel used the BOOM display and the DataGloves for interaction. (Bryson and Levit, 1992, Mazuryk and Gervautz, 1996).

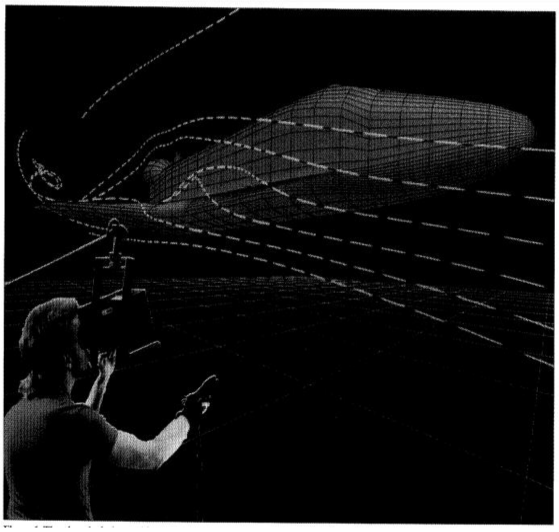


Figure 9.10. The Virtual wind tunnel (Bryson and Levit, 1992) enables the exploration of numerically generated 3-D unsteady velocity vector fields using techniques such as tufts, streamlines, and particle paths. The user views the visualisation by looking through the holes in the BOOM device and manipulates the objects using the hand controllers to generate some motion around different objects to study the flow field.

CAVE – The CAVE Automated Virtual Reality Environment (Figure 9.11) is a virtual reality system, presented in 1992. It projects stereoscopic images onto multiple walls to create a 3-D environment. The user views the environment through the required 3-D shutter glasses. The multiple walls provide a wider field of regard to the user.

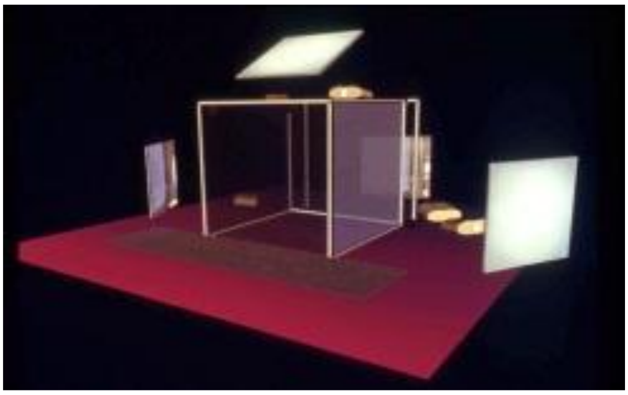


Figure 9.11. CAVE Automatic Virtual Environment (Febretti et al., 2013) provides a virtual reality experience in room-sized hardware. CAVE consists of several cameras that project different angles of the virtual 3-D scene onto walls that allows the user to experience a room-sized viewing space using special wearable glasses.

Current Generation of VR HMDs - The current generation of VR HMDs are advanced in terms of their tracking such as combining natural hand gesture recognition with the traditional controllers. They include the HTC VIVE (HTC, 2020), Facebook's Oculus Rift (Desai et al., 2014, Oculus, 2020), and Samsung Gear VR (Hillmann, 2019, Samsung, 2020).

Wireless HMDs which eliminate the need for cables connecting the HMDs to a PC are going mainstream with manufacturers, for instance, the untethered HTC VIVE Pro 2 (HTC, 2021). The HTC VIVE consists of a wearable head-mounted display and hand controllers (Figure 9.12). It has 2 base station mounts that help to calibrate or map the physical environment for use in the virtual space. The controller buttons allow mapping to various functions such as teleporting, grabbing, and throwing.

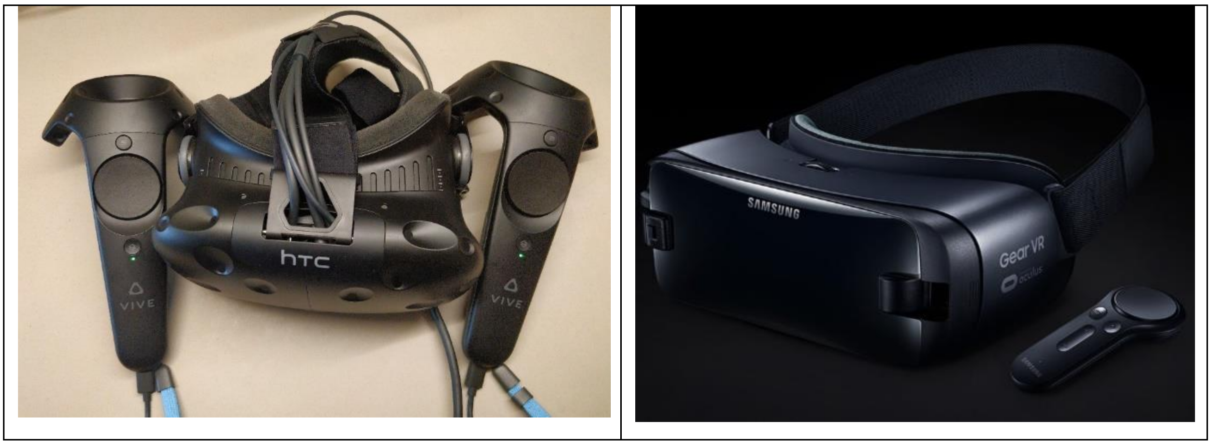


Figure 9.12 (LHS) HTC VIVE Head-mounted display together with its hand controllers. The setup utilises two base stations that help calibrate the physical environment and provide the space for the virtual environment. (RHS) Samsung Gear VR.

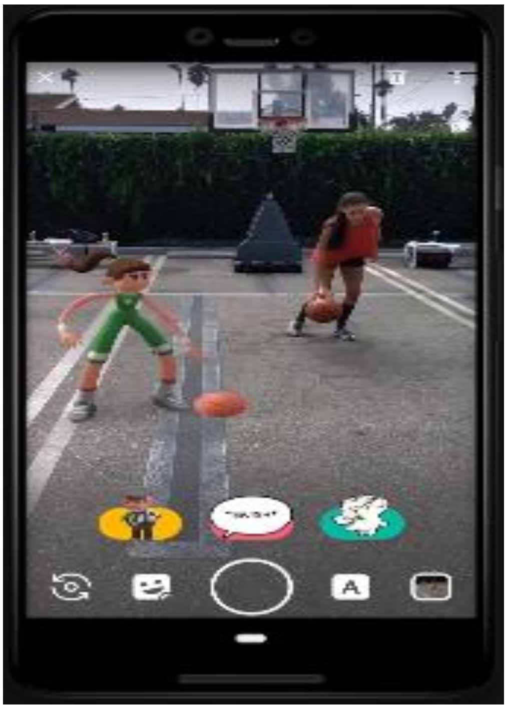
Augmented and Mixed Reality (AR/MR) – A technology that presents a virtual world that enriches, rather than replaces the real world. It can also be defined as a real-time direct or indirect view of the physical environment that has been enhanced by adding virtual objects (Carmigniani et al., 2011). AR & MR are characterised by immersing a user in an interactive 3-D scene that combines the real world with the virtual world (Azuma, 1997). The augmented reality experience allows a user to view an environment consisting of the real world with virtual objects superimposed on it. This is usually viewed through a compatible AR or MR device, for instance, Microsoft HoloLens, and smartphones (Mazuryk and Gervautz, 1996).

An early example of mixed reality was the VCASS which augmented and enriched a fighter pilot's view with added information e.g., optimal flight path and other details from the flight dashboard. This enrichment of the real world with virtual information advanced research in augmented reality (Mazuryk and Gervautz, 1996).

Mixed reality also engages the user in a 3-D world that combines the real world and the virtual world, however, whilst augmented reality overlays the virtual objects on top of the real-world object as can be seen in Figure 9.13, mixed reality anchors virtual objects to the objects in the real world.

In recent times, some of the most used augmented reality applications include Google lens (Lucia, Vetter and Moroz, 2021) for searching the internet using images directly captured from a device camera and the Google Maps Street View (Tokusho and Feiner, 2009) feature in Google maps that overlays information content such as directions on the maps.

Dey et al. (2018) undertook an extensive review of augmented reality research between 2005 and 2014.



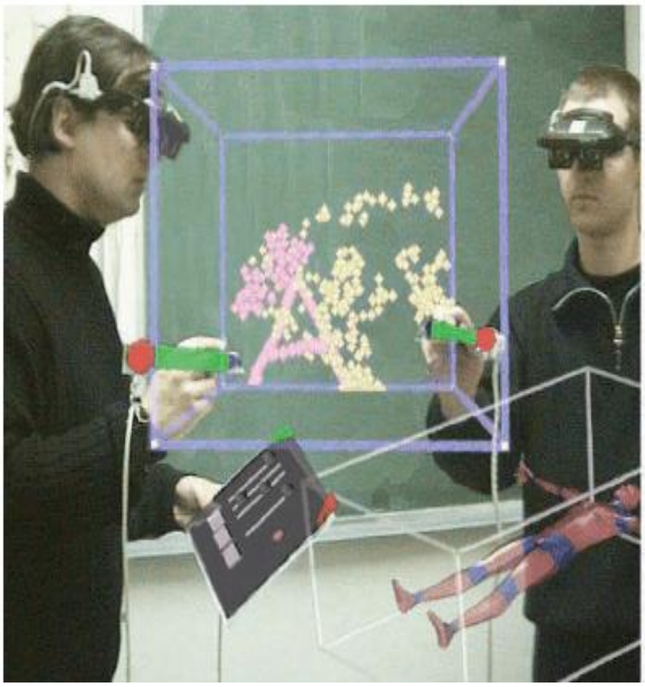


Figure 9.13. (LHS) A mobile-based Augmented reality with an environment enriched with a 3-D character (source: https://arvr.google.com/ar/) (RHS) users interacting with virtual objects in mixed reality using a mixed reality HMD (Azuma et al., 2001). The mixed reality environments allow users to collaboratively work on a task such as building 3-D prototypes.