**MSc Dissertation**

**Motion Signal Extraction Framework for the Microsoft Kinect Camera: Point Cloud Registration and its Application as a Motion Correction Metric in PET/CT**

Submitted for the MSc in

Advanced Computer Science

June 18

By

**Alexander C Whitehead**

Word Count: zero

*I, Alexander C Whitehead, confirm that the work presented in this dissertation is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.*

Abstract

Positron emission tomography (PET) is a common medical imaging modality used for acquiring functional images. These scans can take upwards of a few minutes to complete and during this time a subject could move for any number of reasons, including breathing (respiratory motion). Standard practise is to ignore this movement, however, any movement no matter how slight degrades image resolution and introduces motion related artefacts.

Currently a stand-alone, cross-platform framework is under development to provide a facility for image analysis, reconstruction and data processing of a multitude of data formats acquired through the use of PET, computed tomography, single photon emission computed tomography and magnetic resonance imaging scanners. The goal of this project will be the development of modules or libraries that are able to extract motion vector fields or translations using the Microsoft Kinect camera. These modules or libraries should be able to be integrated into the framework mentioned previously, to aid in motion compensated reconstruction on the Sedecal Argus Scanner.

This application will acquire data from a 3D stereo camera creating point cloud representations of a scene and will then calculate and output a vector field or translation that represents the changes in position of these points over time. These vector fields or translations can then be used in motion correction or motion compensated medical imaging reconstruction to aid in the elimination of motion related artefacts.

The development of this project will take place using STIR, image reconstruction toolkit version 3 and will be compatible with an open source, cross-platform application developed by the PET preclinical centre at the University of Hull.

Contents

[1 Introduction 9](#__RefHeading___Toc2501_729452191)

[2 Background Research and Application Specific Technologies 10](#__RefHeading___Toc4036_58740912)

[2.1 Positron Emission Tomography 10](#__RefHeading___Toc4038_58740912)

[2.1.1 Motion Correction 11](#__RefHeading___Toc3603_623086721)

[2.2 Alternative Solutions 11](#__RefHeading___Toc4040_58740912)

[2.2.1 Data Driven Motion Tracking 11](#__RefHeading___Toc4042_58740912)

[2.2.2 Optical Motion Tracking 11](#__RefHeading___Toc4044_58740912)

[2.3 Microsoft Kinect Camera 12](#__RefHeading___Toc4046_58740912)

[2.3.1 Structured Light Sensor 13](#__RefHeading___Toc3605_623086721)

[2.3.2 Time of Flight Sensor 13](#__RefHeading___Toc3607_623086721)

[2.4 Scanner Synchronisation 13](#__RefHeading___Toc2181_4169133381)

[2.4.1 Temporal Synchronisation 13](#__RefHeading___Toc2183_4169133381)

[2.4.2 Spatial Synchronisation 13](#__RefHeading___Toc2185_4169133381)

[2.5 Point Cloud Library 13](#__RefHeading___Toc4048_58740912)

[2.5.1 Cleaning Point Clouds 14](#__RefHeading___Toc4050_58740912)

[2.5.2 Point Cloud Registration 14](#__RefHeading___Toc4052_58740912)

[2.6 Reliant Technologies 14](#__RefHeading___Toc4054_58740912)

[2.6.1 STIR Framework 14](#__RefHeading___Toc4056_58740912)

[2.6.2 SAvVy 14](#__RefHeading___Toc3611_623086721)

[2.6.3 Qt Graphical User Interface 14](#__RefHeading___Toc4058_58740912)

[2.6.4 OpenKinect Libfreenect Driver 15](#__RefHeading___Toc4457_623086721)

[3 Aims, Hypothesis and Objectives 16](#__RefHeading___Toc4062_58740912)

[Objective 1 – Port the STIR framework to Windows 17](#__RefHeading___Toc2187_4169133381)

[Objective 2 – Read data from the Kinect camera 17](#__RefHeading___Toc2189_4169133381)

[Objective 3 – Create stand-alone console application to interface with Kinect 17](#__RefHeading___Toc2191_4169133381)

[Objective 4 – Add ability to save point cloud from Kinect to file or data type 17](#__RefHeading___Toc4066_58740912)

[Objective 5 – Add ability to output multiple point clouds per execution 17](#__RefHeading___Toc4068_58740912)

[Objective 6 – Add ability to timestamp output 18](#__RefHeading___Toc4070_58740912)

[Objective 7 – Add ability to clean point clouds 18](#__RefHeading___Toc4072_58740912)

[Objective 8 – Add ability to register between point clouds 18](#__RefHeading___Toc4074_58740912)

[Objective 9 – Add ability to extract vector field or translation from registered point clouds 18](#__RefHeading___Toc4076_58740912)

[Objective 10 – Add ability to remove extraneous data from point cloud 18](#__RefHeading___Toc4078_58740912)

[Objective 11 – Add ability to synchronise timestamp on point cloud to output from scanner 19](#__RefHeading___Toc4080_58740912)

[Objective 12 – Add ability to translate camera space to scanner space 19](#__RefHeading___Toc4082_58740912)

[Objective 13 – Add ability to apply output from application to output from scanner 19](#__RefHeading___Toc4084_58740912)

[Objective 14 – Port application to Windows 19](#__RefHeading___Toc4086_58740912)

[Objective 15 – Add application to the Windows version of STIR as library 19](#__RefHeading___Toc4088_58740912)

List of Illustrations

[This illustration shows an example representing the opposing pairs of gamma ray events which are used to determine the location of the positron from the radioactive tracer within the subjects body. 10](#Illustration!0|sequence)

[This illustration shows an example of a PET/MR scanner on the left and an example of the output of a PET/MR scanner on the right. 11](#Illustration!1|sequence)

[This illustration shows an example of the Kinect camera version one on the left and an example of the Kinect camera version two on the right. 13](#Illustration!5|sequence)

[This illustration shows a comparison between the Kinect camera version one and the Kinect camera version two. 21](#Illustration!2|sequence)

[This illustration shows the data which was used to generate the time plan Gantt chart. 26](#Illustration!3|sequence)

[This illustration shows the time plan data as represented by a Gantt chart. 27](#Illustration!4|sequence)

List of Tables

[This table shows the list of tasks that must be completed to finish the project. These tasks are then expanded upon by providing a short description of each task and an estimated duration in weeks. 25](#Table!0|sequence)

[This table shows a breakdown of some risks which could occur while working on the project. These risks are then expanded upon by determining the current risk, determining solutions to these risks and then determining the residual risk. 30](#Table!1|sequence)

List of Abbreviations

**2D** Two Dimensional

**3D** Three Dimensional

**CT** Computed Tomography

**MLEM** Maximum Likelihood Expectation Maximisation

**MR** Magnetic Resonance

**MRI** Magnetic Resonance Imaging

**OSEM** Ordered Subset Expectation Maximisation

**PET** Positron Emission Tomography

**RGB** Red, Green, Blue

# Introduction

This is a report that describes the initial research and design stages of the project to develop a motion signal extraction framework for the Microsoft Kinect camera and to then apply the output of this framework to the task of motion correction in positron emission tomographic (PET) scanners.

This report includes sections detailing:

1. A description of the domain that the project exists within and a basic overview and introduction to the concepts that the project intends to address.
2. A walk-through of background research on both the broad topics that the project addresses and application specific technologies and approaches.
3. A specification of the aim, hypothesis and objectives of the project.

The original research proposal for this project can be seen in the appendix below. (Appendix A:)

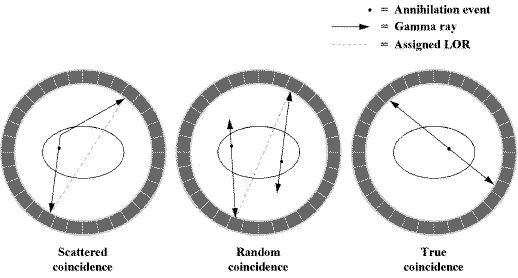
# Background Research and Application Specific Technologies

## Positron Emission Tomography

PET is a medical imaging modality that is used to acquire functional images, images of the internal metabolic processes, of a subjects body. These images are acquired as an aid to the diagnosis of irregularities within the body and are commonly used in the fields of Oncology, Cardiology, Neurology and Psychiatry. (Bertolli, 2018)⁠

In order to operate, a PET scanner attempts to detect opposing pairs of gamma ray events, these gamma ray events originate from the annihilation of a positron within the subjects body, these positrons are introduced into the body using a radioactive biologically active molecule commonly referred to as a tracer.

The illustration below (Illustration 1) (Nandi, 2013) ⁠shows an example representing the opposing pairs of gamma ray events which are used to determine the location of the positron from the radioactive tracer within the subjects body.

  
Illustration 1: This illustration shows an example representing the opposing pairs of gamma ray events which are used to determine the location of the positron from the radioactive tracer within the subjects body.

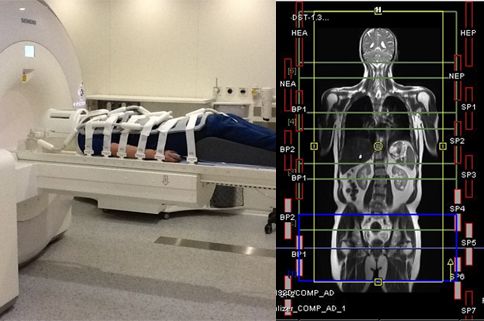
Using the pairs of gamma ray events mentioned above, a three dimensional (3D) image of the concentration of the tracer in specific areas of the subjects body is then reconstructed using algorithms such as maximum likelihood expectation maximisation (MLEM) or ordered subset expectation maximisation (OSEM). (Kadrmas, 2004)

When used for clinical diagnosis these 3D images are usually sliced into separate two dimensional (2D) images.

In order to increase the accuracy of modern PET scanners, additional modalities have been combined with PET to aid in the reconstruction of images. These additional modalities include computed tomography (CT) x-ray scans and magnetic resonance (MR) or magnetic resonance imaging (MRI) scans. (Drzezga *et al.*, 2012)⁠

These additional scans are usually preformed on the same machine but often require a different bed position as the two parts of the scanner are located separately from one another.

The illustration below (Illustration 2) (*New PET MRI scanner is first of its kind in UK*, 2012) shows an example of a PET/MR scanner on the left and an example of the output of a PET/MR scanner on the right.

  
Illustration 2: This illustration shows an example of a PET/MR scanner on the left and an example of the output of a PET/MR scanner on the right.

## Motion Correction

## Alternative Solutions

## Data Driven Motion Tracking

## Optical Motion Tracking

## Microsoft Kinect Camera

The Kinect camera is a family of depth sensing cameras developed for use with the Xbox line of game consoles developed by Microsoft. Each camera consists of, at least, an RGB camera for capturing red, green, blue (RGB) video data, a depth sensor for capturing point cloud data and a multi-array microphone for capturing sound data. Together these devices can, through software, provide 3D motion capture capabilities.

Each version of the Kinect camera uses a different type of depth sensor, however, each type of depth sensor operates using an infrared laser which projects a pattern of points onto a surface. The pattern of points produced by the infrared laser is then used to determine the distance from the camera to each point in space in view of the camera. The data from the depth sensor is then registered to a monochrome image which is captured from an additional CMOS camera to create a 3D scene.

The working depth of the Kinect camera is adjustable within a range of approximately four to eight metres. The minimum average distance the subject would need to away from the Kinect camera is about half a metre, this is to avoid over-saturation of the depth sensor, however, the intensity of the infrared laser can be changed within the firmware of the Kinect camera to allow for scanning of closer objects. (Noonan *et al.*, 2015)

The full specification for both the Kinect camera version one and the Kinect camera version two can be seen in the appendix below. (Appendix B:) (Noonan *et al.*, 2015)⁠

The illustration below (Illustration 3) (Smeenk, 2014)⁠ shows an example of the Kinect camera version one on the left and an example of the Kinect camera version two on the right.

  
Illustration 3: This illustration shows an example of the Kinect camera version one on the left and an example of the Kinect camera version two on the right.

## **Structured Light Sensor**

## **Time of Flight Sensor**

## Scanner Synchronisation

## Temporal Synchronisation

## Spatial Synchronisation

## Point Cloud Library

The Point Cloud Library (PCL) is an open-source library of algorithms for point cloud processing tasks and 3D geometry processing, such as occur in three-dimensional computer vision. The library contains algorithms for feature estimation, surface reconstruction, 3D registration, model fitting, and segmentation.

These algorithms have been used, for example, for perception in robotics to filter outliers from noisy data, stitch 3D point clouds together, segment relevant parts of a scene, extract keypoints and compute descriptors to recognize objects in the world based on their geometric appearance, and create surfaces from point clouds and visualize them.

## Cleaning Point Clouds

## Point Cloud Registration

## Reliant Technologies

## STIR Framework

STIR is Open Source software for use in tomographic imaging. Its aim is to provide a Multi-Platform Object-Oriented framework for all data manipulations in tomographic imaging. Currently, the emphasis is on (iterative) image reconstruction in PET and SPECT, but other application areas and imaging modalities can and might be added.

## SAvVy

## Qt Graphical User Interface

Qt is a cross-platform application framework and widget toolkit for creating classic and embedded graphical user interfaces, and applications that run on various software and hardware platforms with little or no change in the underlying codebase, while still being a native application with native capabilities and speed.

## OpenKinect Libfreenect Driver

# Aims, Hypothesis and Objectives

The aim of this project is:

To develop a method of motion correction for medical scanners using the Microsoft Kinect camera which can be integrated into the STIR framework

The hypothesis of this project is that it is possible to extrapolate from a number of point clouds produced by a stereo 3D camera a metric that represents the motion of an object within a medical scanner. It is then possible to use this metric to remove motion related artefacts from and improve the resolution of scan data produced using said medical scanner.

The null hypothesis of this project is that it is not possible to extrapolate from a number of point clouds produced by a stereo 3D camera a metric that represents the motion of an object within a medical scanner. It is then not possible to use this metric to remove motion related artefacts from and improve the resolution of scan data produced using said medical scanner.

To achieve the aim and validate the hypothesis set out above the following objectives will be completed:

1. Port the STIR framework to Windows.
2. Read data from the Kinect Camera using the open source Kinect drivers.
3. Create a stand-alone console application to interface with the Kinect camera.
4. Add the ability to save a point cloud from the Kinect camera to a file or data type.
5. Add the ability to output multiple point clouds per execution of the application.
6. Add the ability to timestamp the point clouds that are output by the Kinect camera.
7. Add the ability to clean the point clouds that are output by the Kinect camera.
8. Add the ability to register between the point clouds that are output by the Kinect camera.
9. Add the ability to extract a vector field or translation between two objects from the registered point clouds that were output from the Kinect camera.
10. Add the ability to remove any extraneous data from the point clouds that were output from the Kinect camera.
11. Add the ability to synchronise the timestamps contained within the point clouds that were output from the Kinect camera to the output from the PET scanner.
12. Add the ability to translate the origin of the space used by the Kinect camera to match the space used by the PET scanner.
13. Add the ability to apply the motion calculated from the output of the Kinect camera to the output of the PET scanner.
14. Port the application to Windows.
15. Add the application to the windows version of the STIR framework as a library.

### Objective 1 – Port the STIR framework to Windows

The application should be able to be run on Windows, in order for that to be the case the STIR framework must be ported to Windows.

### Objective 2 – Read data from the Kinect camera

An application should be created which can read data from the Kinect camera in real time and visualise this data to the user.

The open source Kinect driver should be used for this application, this is because it allows for the development on and distribution to multiple operating systems.

### Objective 3 – Create stand-alone console application to interface with Kinect

A stand-alone console application and library should be created which can either be launched directly as an executable program or can be compiled as a library and then accessed within another application.

To achieve this the application could be written as a library, then a wrapper application could be written to run it independently.

This is advantageous as it allows for more rapid prototyping of ideas without having to rely on a large codebase, which in itself could contain errors, without hindering the ease with which the application could be integrated into a larger framework.

This application should be able to communicate with the Kinect camera and receive point clouds from the Kinect camera.

### Objective 4 – Add ability to save point cloud from Kinect to file or data type

The ability to save the data from one scan of the Kinect camera should be added to the application.

The data should be saved either to a file in storage or to a data type contained within the application.

A class or data type could be written that would be able to contain the data output from the Kinect camera, this class or data type should overload the streaming operators allowing it to be written either automatically to the console output or to a file.

### Objective 5 – Add ability to output multiple point clouds per execution

The ability to save the data of multiple concurrent scans from the Kinect camera should be added to the application.

It is possible either to save each piece of data as they are read from the Kinect camera or to save all pieces of data at once after the scan has been complete.

Each new piece of data should be saved into a list or vector with the type of the class mentioned above, after execution has ended, a method could be called which would stream the entire list or vector either to the console output or to a file.

### Objective 6 – Add ability to timestamp output

The ability to save the time that a scan occurred should be added to the application. This timestamp could be added as a data type to the class mentioned above, the timestamp should also be added to the overloaded streaming operators.

Timestamping the data from the Kinect camera is necessary to synchronize the Kinect camera output to the PET scanner output later.

### Objective 7 – Add ability to clean point clouds

The ability to remove extraneous or outlying points from the data should be added to the application.

The process of cleaning the data can be either an automatic process or a manual one.

The open source Point Cloud Library contains methods and algorithms that have been implemented and tested to clean point clouds.

### Objective 8 – Add ability to register between point clouds

The ability to translate one set of points to another should be added to the application.

The registration process can be either rigid or non-rigid; this means that either the relationship between the points must remain the same for rigid registration or the relationship between the points can change for non-rigid registration.

The open source Point Cloud Library contains methods and algorithms that have been implemented and tested to register point clouds.

### Objective 9 – Add ability to extract vector field or translation from registered point clouds

The ability to extract the transformation from the registered data should be added to the application.

This could either be a vector field for non-rigid registration or a translation if rigid registration is used.

### Objective 10 – Add ability to remove extraneous data from point cloud

The ability to remove structures that are not objects of interest should be added to the application.

In every set of data from the Kinect camera it is likely there will be structures that are of no importance to the functionality of the application, for instance the bezel of the PET scanner, these objects should be removed to cut down on processing power and memory usage.

A smaller data set is also easier to visualise clearly.

A point cloud could be attained before scanning commences, then the difference between the initial point cloud and every other point cloud could be used to remove the bezel of the PET scanner and other extraneous data.

### Objective 11 – Add ability to synchronise timestamp on point cloud to output from scanner

The ability to synchronise the output of the Kinect camera to the output of the PET scanner should be added to the application.

In order for each vector field or translation extracted from the Kinect camera to be relevant it is important to know when this vector field or translation was extracted in relation to the output of the PET scanner, this is to ensure that the correct vector field or translation is applied to the correct data from the PET scanner.

One method to synchronise the output of the Kinect camera to the output of the PET scanner would be to use an Arduino to output a regular pulse of electricity and use this to set the clocks of both devices to the same rate.

### Objective 12 – Add ability to translate camera space to scanner space

The ability to translate the output of the Kinect camera to the output of the PET scanner should be added to the application.

In order for each vector field or translation extracted from the Kinect camera to be relevant the origin of both coordinate systems must be aligned, otherwise the same point in space could be represented in two different locations in both sets of data.

One method to translate the Kinect camera space to the PET scanner space would be to simultaneously scan a radiation point source with both the Kinect camera and the PET scanner and use this to calculate the translation required.

### Objective 13 – Add ability to apply output from application to output from scanner

The ability to translate the output of the PET scanner by the vector field or translation of the data from Kinect camera should be added to the application.

This should apply the motion data gathered from the Kinect camera to the data from the PET scanner, this would effectively remove the motion recorded in the Kinect camera’s data from the data of the PET scanner.

After this step has been applied, the output from PET scanner should appear to have a higher resolution and many motion related artefacts should be eliminated.

### Objective 14 – Port application to Windows

The application should be able to run on a Windows machine.

This is to increase the potential user base for the application.

### Objective 15 – Add application to the Windows version of STIR as library

The applications should be able to be added to the Windows version of the STIR framework as a library and integrated fully into its functionality.

1. Research Proposal

A stand-alone cross-platform framework for image analysis, reconstruction and data processing of a variate of data acquired by PET/SPECT scanners and potentially MRI is under development. The framework links and makes use of external open source libraries in order to handle data formats and perform various complex tasks. The GUI is developed on QT and there is a strong drive to have cross-platform compatibility (especially between Windows and Linux).

The goal will be to provide a responsive/neat/intuitive plug-in able to acquire data from a Microsoft Kinect camera in an easy plug and play fashion and extract from those warp spaces and/or motion signals for motion corrected or compensated medical image reconstruction.

This is a project on scientific programming but many programming challenges have to be addressed. The successful candidate will need to fluent in a C variance, with drive to improve his skills on C++ and Cmake.

1. Kinect Version Comparison
2. Task List

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task Name** | **Description** | **Duration**  **(weeks)** |
| 1 | Research PET Scanners | Conduct background research specifically related to the operation of PET scanners. This research will aid in the writing of the initial report and the design of the application. | 3 |
| 2 | Research Alternative Solutions | Conduct background research specifically related to alternative solutions to the same problem that this project sets out to solve. This research will aid in the writing of the initial report and the design of the application. | 2 |
| 3 | Research Kinect Camera | Conduct background research specifically related to the operation of the Kinect Camera and its API. This research will aid in the writing of the initial report and the design of the application. | 2 |
| 4 | Research Point Cloud Library | Conduct background research related to the open source Point Cloud Library, specifically its ability to perform the cleaning of point clouds and the registration of point clouds. This research will aid in the writing of the initial report and the design of the application. | 2 |
| 5 | Write initial report | Write the initial report; this will involve writing the bulk of the content. | 4 |
| 6 | Edit initial Report | Edit the initial report, this will involve editing the content and ensuring it is presented to the best standard it could be. | 2 |
| 7 | Port STIR framework to Windows | The application should be able to be run on Windows, in order for that to be the case the STIR framework must be ported to Windows. | 2 |
| 8 | Read data from the Kinect | An application should be created which can read data from the Kinect camera in real time and visualise this data to the user. | 1 |
| 9 | Create stand-alone console application to interface with Kinect | A stand-alone console application should be created which can either be launched directly or can also be compiled as a library in order to use it as a part of a larger application. This application should be able to read data from the Kinect camera and communicate with the Kinect camera. | 1 |
| 10 | Add ability to save point cloud from Kinect to file or data type | The ability to save the data from one scan of the Kinect camera should be added to the application. The data should be saved either to a file in storage or to a data type contained within the application. | 1 |
| 11 | Add ability to output multiple point clouds per execution | The ability to save the data of multiple concurrent scans from the Kinect camera should be added to the application. It is possible either to save each piece of data as they are read from the Kinect camera or to save all pieces of data at once after the scan has been complete. | 1 |
| 12 | Add ability to timestamp output | The ability to save the time that a scan occurred should be added to the application. Timestamping the data from the Kinect camera is necessary to synchronize the Kinect camera output to the PET scanner output later. | 1 |
| 13 | Add ability to clean point clouds | The ability to remove extraneous or outlying points from the data should be added to the application. The process of cleaning the data can be either an automatic process or a manual one. | 2 |
| 14 | Add ability to register between point clouds | The ability to translate one set of points to another should be added to the application. At this stage, only the ability to recognise similar structures and match them between different sets of data is necessary. The registration process can be either rigid or non-rigid; this means that either the relationship between the points must remain the same for rigid registration or the relationship between the points can change for non-rigid registration. | 3 |
| 15 | Add ability to extract vector field or translation from registered point clouds | The ability to extract the transformation from the registered data should be added to the application. This could either be a vector field for non-rigid registration or a translation if rigid registration is used. | 3 |
| 16 | Add ability to remove extraneous data from point cloud | The ability to remove structures that are not objects of interest should be added to the application. In every set of data from the Kinect camera it is likely there will be structures that are of no importance to the functionality of the application, for instance the bezel of the PET scanner, these objects should be removed to cut down on processing power and memory usage. A smaller data set is also easier to visualise clearly. | 2 |
| 17 | Add ability to synchronise timestamp on point cloud to output from scanner | The ability to synchronise the output of the Kinect camera to the output of the PET scanner should be added to the application. In order for each vector field or translation extracted from the Kinect camera to be relevant it is important to know when this vector field or translation was extracted in relation to the output of the PET scanner, this is to ensure that the correct vector field or translation is applied to the correct data from the PET scanner. | 2 |
| 18 | Add ability to translate camera space to scanner space | The ability to translate the output of the Kinect camera to the output of the PET scanner should be added to the application. In order for each vector field or translation extracted from the Kinect camera to be relevant the origin of both coordinate systems must be aligned, otherwise the same point in space could be represented in two different locations in both sets of data. | 2 |
| 19 | Add ability to apply output from application to output from scanner | The ability to translate the output of the PET scanner by the vector field or translation of the data from Kinect camera should be added to the application. This should apply the motion data gathered from the Kinect camera to the data from the PET scanner, this would effectively remove the motion recorded in the Kinect camera’s data from the data of the PET scanner. After this step has been applied, the output from PET scanner should appear to have a higher resolution and many motion related artefacts should be eliminated. | 3 |
| 20 | Write final report | Write the final report; this will involve writing the bulk of the content. | 8 |
| 21 | Edit final report | Edit the final report, this will involve editing the content and ensuring it is presented to the best standard it could be. | 3 |
| 22 | Port application to Windows | The application should be able to run on a Windows machine. This is to increase the potential user base for the application but also to ensure that the application is compatible with the STIR framework. | 2 |
| 23 | Add application to the Windows version of STIR as library | The applications should be able to be added to the Windows version of the STIR framework as a library and integrated fully into its functionality. | 2 |

Table 1: This table shows the list of tasks that must be completed to finish the project. These tasks are then expanded upon by providing a short description of each task and an estimated duration in weeks.

1. Time Plan Table
2. Time Plan Gantt Chart
3. Risk Assessment

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Risk** | **Current Risk** | | | **How to Avoid** | **How to Recover** | **Residual Risk** | | |
| Severity  (L/M/H) | Likelihood  (L/M/H) | Significance  (Severity, Likelihood) | Severity  (L/M/H) | Likelihood  (L/M/H) | Significance  (Severity, Likelihood) |
| Data loss | H | M | HM | Keep backups | Reinstate from backups | L | M | LM |
| Loss of backups | H | L | HL | Keep multiple backups on multiple different forms of media in multiple different locations | Use alternate backup | L | L | LL |
| Underrate workload | H | M | HM | Regularly review progress against Time Plan | Invest more time into work or reduce objectives | H | L | HL |
| Critical error in deliverable | H | M | HM | Perform adequate background research | Thoroughly test and debug code | H | L | HL |
| Skill risk | M | M | MM | Perform adequate training or seek out specialists | Invest more time into background research | L | L | LL |
| Scope creep | M | H | MH | Fully define objectives | Fully define current objectives and do not change the objectives again | M | L | ML |
| Inefficient program speed | H | L | HL | Invest time in testing and debugging code | Optimise code or remove slow code | M | L | ML |
| Medical emergency | H | L | HL | Care for developers health including regular periods of rest | Comment code regularly so that it is well understood | M | L | ML |
| Injury while using application | H | M | HM | Ensure proper training is in place before the application is used | Medical attention should be available to anyone injured while the application is in use | H | L | HL |
| Damage to equipment while the application is in use | H | M | HM | Test application compatibility with different working situations | Repair damage to equipment | H | L | HL |
| Incorrect conclusions drawn because of use of application | H | H | HH | Never solely use the application to draw conclusions | Use other sources of data to aid in drawing conclusion | M | M | MM |

Table 2: This table shows a breakdown of some risks which could occur while working on the project. These risks are then expanded upon by determining the current risk, determining solutions to these risks and then determining the residual risk.

1. Ethics Report

If your project uses other people (‘participants’) for the collection of information (typically in getting comments about a system or a system design, getting information about how a system could be used, or evaluating a working system) then you need to read through the checklist in Section A below before completing the declaration in Section B.

If your project does **not** make use of other people then you can skip Section A and directly complete the declaration in Section B by marking box ‘1’ with an X.

Section A

**1. Participants will not be exposed to any risks greater than those encountered in their normal working life.**

Researchers have a responsibility to protect participants from physical and mental harm during the investigation. The risk of harm must be no greater than in ordinary life. Areas of potential risk that require ethical approval include, but are not limited to, investigations that occur outside usual laboratory areas, or that require participant mobility (e.g. walking, running, use of public transport), unusual or repetitive activity or movement, that use sensory deprivation (e.g. ear plugs or blindfolds), bright or flashing lights, loud or disorienting noises, smell, taste, vibration, or force feedback

**2. The experimental materials will be paper-based, or comprised software running on standard hardware.**

*Participants should not be exposed to any risks associated with the use of non-standard equipment: anything other than pen-and-paper, standard PCs, mobile phones, and PDAs is considered non-standard.*

**3. All participants will explicitly state that they agree to take part, and that their data could be used in the project.**

*If the results of the evaluation are likely to be used beyond the term of the project (for example, the software is to be deployed, or the data is to be published), then signed consent is necessary. Each participant should sign a separate consent form.*

*Otherwise, verbal consent is sufficient, and should be explicitly requested in the introductory script.*

**4. No incentives will be offered to the participants.**

*The payment of participants must not be used to induce them to risk harm beyond that which they risk without payment in their normal lifestyle.*

**5. No information about the evaluation or materials will intentionally be withheld from the participants.**

*Withholding information or misleading participants is unacceptable if participants are likely to object or show unease when debriefed.*

**6. No participant will be under the age of 16.**

*Parental consent is required for participants under the age of 16.*

**7. No participant will have an impairment that may limit his or her understanding or communication.**

*Additional consent is required for participants with impairments.*

**8. Neither I nor my supervisor is in a position of authority or influence over any of the participants.**

*A position of authority or influence over any participant must not be allowed to pressurise participants to take part in, or remain in, any experiment.*

**9. All participants will be informed that they can withdraw at any time.**

*All participants have the right to withdraw at any time during the investigation. They should be told this in the introductory script.*

**10. All participants will be informed of my contact details.**

*All participants must be able to contact the investigator after the investigation. They should be given the details of both student and module coordinator or supervisor as part of the debriefing.*

**11. The evaluation will be discussed with all the participants at the end of the session, and all participants will have the opportunity to ask questions.**

*The student must provide the participants with sufficient information in the debriefing to enable them to understand the nature of the investigation.*

**12. All the data collected from the participants will be stored in an anonymous form.**

*All participant data (hard copy and soft-copy) should be stored securely, and in anonymous form.*

If your evaluation does comply with all the twelve points above, please mark box ‘2’ in Section B.

If your evaluation does not comply with one or more of the twelve points above, please mark box ‘3’ in Section B unless you **know** that your supervisor already has ethical approval for the project (in which case mark box ‘4’). If you are unsure mark box ‘3’.

*[Adapted from Department of Computing Science University of Glasgow Ethics checklist form for 3rd/4th/5th year, MSc IT/CS/ACS projects 2007]*

Section B

|  |  |
| --- | --- |
| Student Name | Alexander C Whitehead |
| Project Title | Motion Signal Extraction Framework for the Microsoft Kinect Camera: Point Cloud Registration and its Application as a Motion Correction Metric in PET/CT |
| Supervisors Names | Dr D Parker  Dr N Efthymiou |

|  |  |  |
| --- | --- | --- |
| This is a declaration that the ethical concerns for above project have been considered (in particular with regards to the 12 point checklist above) with the following outcome: | | Please mark only ONE box with an X |
| 1 | This project does not involve other people in the collection of information and therefore does not require an ethical review | X |
| 2 | This project complies with the **entire** twelve point ethical checklist and therefore does not require ethical review. |  |
| 3 | This project does not comply with **all** of the twelve points above and therefore does require ethical review and the completion and submission of an ethical approval form. |  |
| 4 | This project does not comply with **all** of the twelve points above, however the supervisor already has ethical approval for this research |  |

If you have marked box ‘3’ you will be expected to apply for ethical approval. Further advice is available from both your project supervisor and the Department’s Ethical Officer, as well as by reading and completing [this form](https://canvas.hull.ac.uk/files/317681/download?download_frd=1).

References

Bertolli, O. (2018) *Data-Driven methods for respiratory signal detection in Positron Emission Tomography*. University College London. Available at: https://liveuclac-my.sharepoint.com/personal/rmhathi\_ucl\_ac\_uk/\_layouts/15/onedrive.aspx?id=%252Fpersonal%252Frmhathi%255Fucl%255Fac%255Fuk%252FDocuments%252FINM%252FINM%252Dshared%252Fpublications%252Fthesis%252FBertolli%252D2018%252DPhD%252Dthesis%252Ep (Accessed: 21 June 2018).

Drzezga, A. *et al.* (2012) ‘First clinical experience with integrated whole-body PET/MR: comparison to PET/CT in patients with oncologic diagnoses.’, *Journal of nuclear medicine : official publication, Society of Nuclear Medicine*. Society of Nuclear Medicine, 53(6), pp. 845–55. doi: 10.2967/jnumed.111.098608.

Kadrmas, D. J. (2004) ‘LOR-OSEM: statistical PET reconstruction from raw line-of-response histograms’, *Physics in Medicine and Biology*. IOP Publishing, 49(20), pp. 4731–4744. doi: 10.1088/0031-9155/49/20/005.

Nandi, A. (2013) *How do PET scans differentiate depth along a cross-section? - Quora*. Available at: https://www.quora.com/How-do-PET-scans-differentiate-depth-along-a-cross-section (Accessed: 22 June 2018).

*New PET MRI scanner is first of its kind in UK* (2012). Available at: http://www.uclh.nhs.uk/Research/BRC/News/Pages/NewPETMRIscannerisfirstofitskindinUK.aspx (Accessed: 22 June 2018).

Noonan, P. J. *et al.* (2015) ‘Repurposing the Microsoft Kinect for Windows v2 for external head motion tracking for brain PET’, *Physics in Medicine and Biology*. IOP Publishing, 60(22), pp. 8753–8766. doi: 10.1088/0031-9155/60/22/8753.

Smeenk, R. (2014) *Kinect V1 and Kinect V2 fields of view compared - Roland Smeenk*. Available at: https://smeenk.com/kinect-field-of-view-comparison/ (Accessed: 23 June 2018).

⁠