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: 3274

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.

Contents

List of Illustrations

List of Tables

List of Abbreviations

2D Two Dimensional
3D Three Dimensional
CT Computed Tomography
GUI Graphical User Interface
MLEM Maximum Likelihood Expectation Maximisation
MR Magnetic Resonance
MRI Magnetic Resonance Imaging
OSEM Ordered Subset Expectation Maximisation
PCL Point Cloud Library
PET Positron Emission Tomography
RGB Red, Green, Blue

1 Introduction

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 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 three dimensional 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 three and will be compatible with an open source, cross-platform application developed by the PET preclinical centre at the University of Hull.

2 Background Research and Application Specific Technologies

2.1 Positron Emission Tomography

PET is a medical imaging modality that is used to acquire functional images, relating to 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.

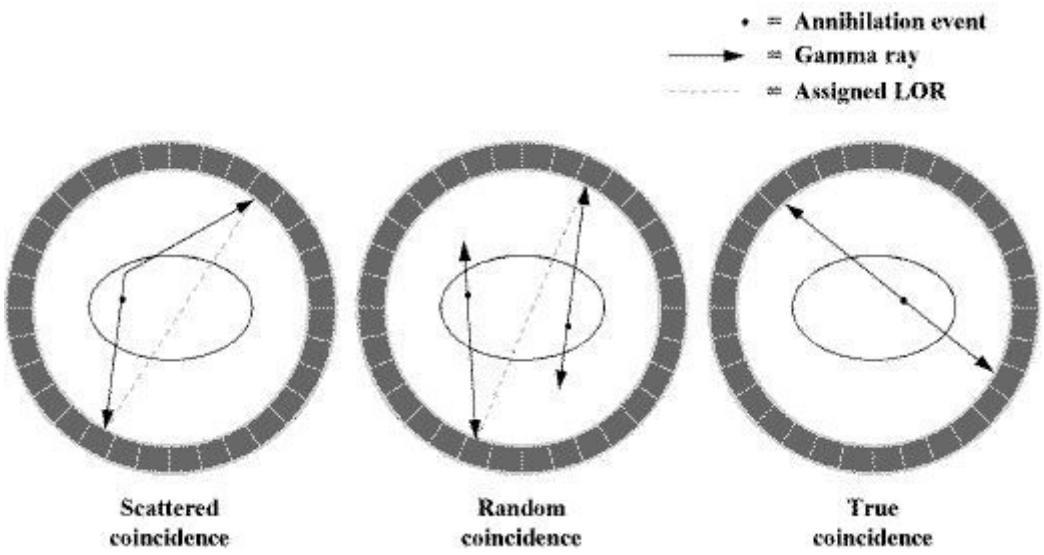

Illustration 1: This illustration shows an example representing the opposing pairs of gamma ray events which are used to determine the location of the positron from the radioactive tracer within the subjects body.

(Nandi, 2013)

Using these pairs of gamma ray events, 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 process. 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 often preformed on the same machine but may require a different bed position as the two parts of the scanner are usually located separately from one another.

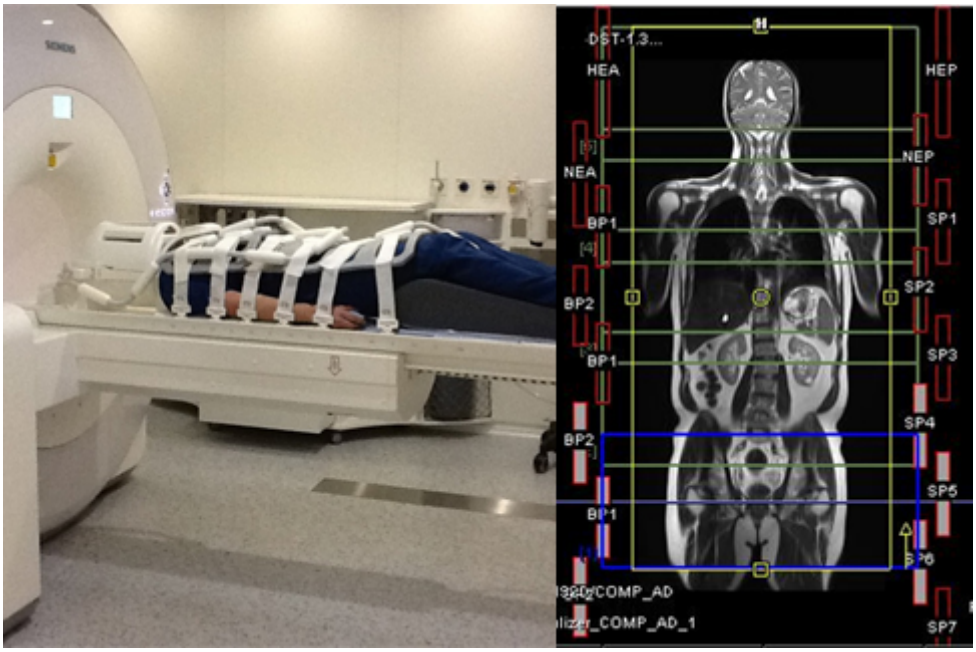

Illustration 2: This illustration shows an example of a PET/MR scanner on the left and an example of the output of a PET/MR scanner on the right.

(*New PET MRI scanner is first of its kind in UK*, 2012)

2.1.1 Motion Correction

A PET scan can take anywhere upwards of 20 minutes to complete, during this time any movement of the subject will result in the loss of resolution or the introduction of motion related artefacts to the final image. Movement which is known to interfere with PET scans includes, semi-involuntary respiratory movement and voluntary muscular movement. (Beyer *et al.*, 2003)

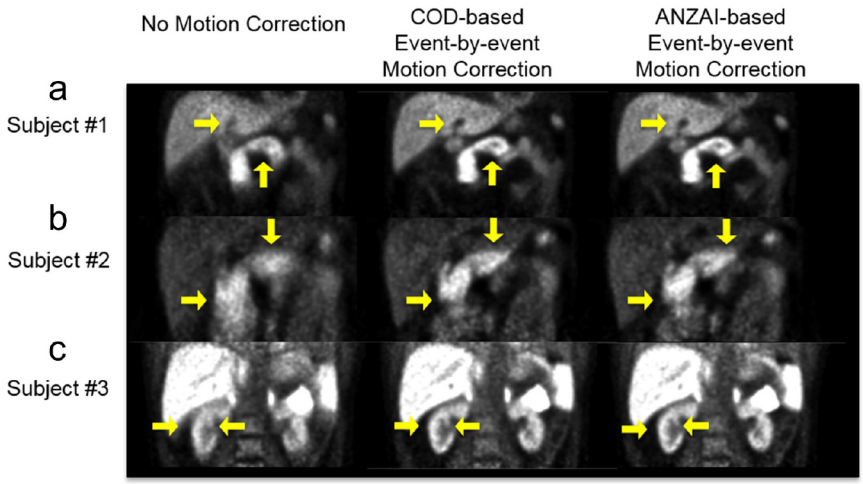

Illustration 3: This illustration shows an example of a PET data which has been affected by motion of the subject on the left, an example of one form of motion correction on the same PET data in the centre and an example of another form of motion correction on the same PET data on the right.
(Ren *et al.*, 2017)

2.2 Alternative Solutions

2.2.1 Data Driven Motion Tracking

One method of motion correction for PET data is to try and determine the motion of the subject from the data of the PET scan itself. To determine the motion of the subject the points at which the subject has either full inhaled or fully exhaled are isolated. These points are then used to gate or bin the data from the PET scanner removing all of the data where the subject is in motion. (Bertolli, 2018)

A disadvantage of this solution is that in order for the image, which is reconstructed from the PET scan data, to exhibit the same characteristics as a normal PET scan the length of the scan must be increased to make up for the missing data.

Another data driven method of motion correction for PET data is to attempt to use a breathing or surrogate signal directly to warp the PET data. This breathing signal can either be acquired through the use of a spirometer or through tracking an object attached to the chest of the subject using an optical camera. (McClelland *et al.*, 2017)

A disadvantage of this solution is that additional hardware and operational steps are required when performing the PET scan.

2.2.2 Optical Motion Tracking

Another method of motion correction for PET data is to try and track the motion of the subject directly using an optical depth sensing camera.
In order to track the motion of the subject, first, the optical depth sensing camera must be temporally and spatially synchronised to the PET scanner, then the motion of the subject can be tracked throughout the the duration of the PET scan by positioning the optical depth sensing camera so that it is in view of the subject within the PET scanner.
The motion data from the optical depth sensing camera is then stored with a timestamp which reflects when the data was acquired, this timestamp can be used to gate or bin the PET data.
In order to remove the motion, vector fields or transformations, reflecting the data which was acquired from the optical depth sensing camera, are then subtracted from the PET data based on the gates or bins determined earlier. (Miranda *et al.*, 2017) (Noonan *et al.*, 2015)

A disadvantage of this solution is that it only allows motion correction based on the surface movement of the subject, as such, this solution is only acceptable when being used to track objects where the internal structure is rigid, like the head. This solution is is not suitable where internal soft tissue can move independently of the surface movement of the subject , like with respiratory motion.

2.3 Microsoft Kinect Camera

The Kinect camera is a family of depth sensing cameras developed for use with the Xbox line of game consoles 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. (Zhang, 2012)

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. This pattern of points 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 can then be 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 be placed 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 adjusted within the firmware of the Kinect camera to allow for the scanning of objects at up to a few centimetres away from the camera. (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)

Illustration 4: 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.

(Smeenk, 2014)

2.3.1 Structured Light Sensor

The Kinect camera version one uses a type of depth camera called a structured light sensor. A structured light camera works by projecting a pattern of alternating vertical lines onto the scene using an infrared laser, this pattern of alternating vertical lines is then read using a camera which is situated a reasonable distance away from the infrared laser but at the same depth as the infrared laser relative to the scene. The camera can then estimate the depth of objects in the scene by measuring the deformation of the pattern of alternating vertical lines. (Khoshelham and Elberink, 2012)

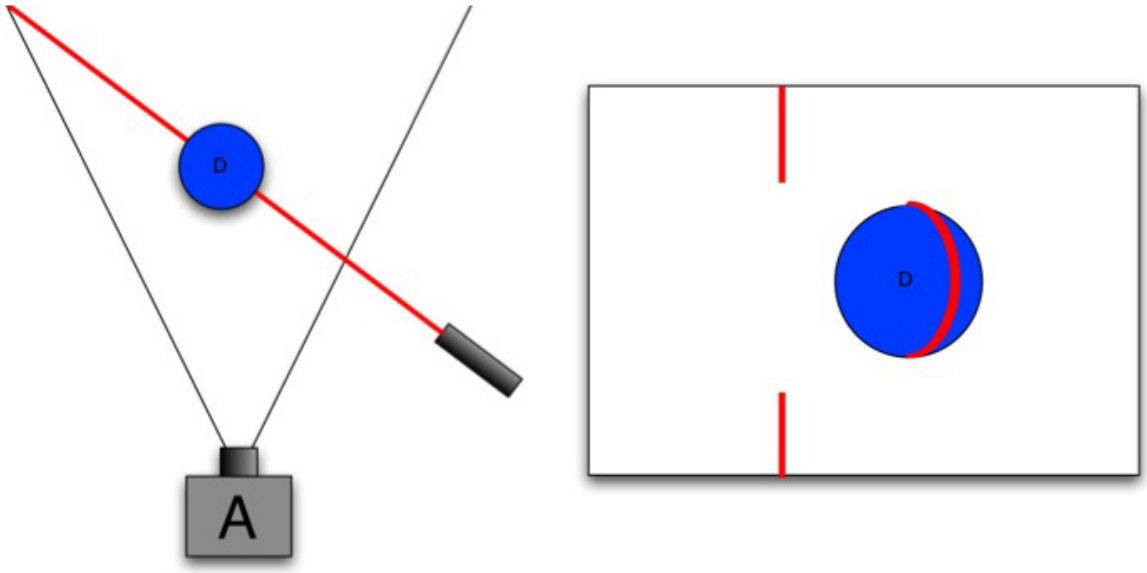

Illustration 5: This illustration shows an example of a scene where an infrared laser is projecting one vertical line onto a sphere, the deformation of this vertical line can then be seen from the perspective of the camera.

(Lau, 2013)

2.3.2Time of Flight Sensor

The Kinect camera version two uses a type of depth camera called a time of flight sensor. A time of flight camera works by pulsing an entire scene with an infrared laser, this pulse is then read using a camera. The camera can then estimate the depth of objects in the scene by measuring how long it takes for each pulse to reach each pixel of the camera.

A time of flight sensor works using a very similar method to a surveying device known as a LiDAR scanner. (Lindner, Schiller and Koch, 2010)

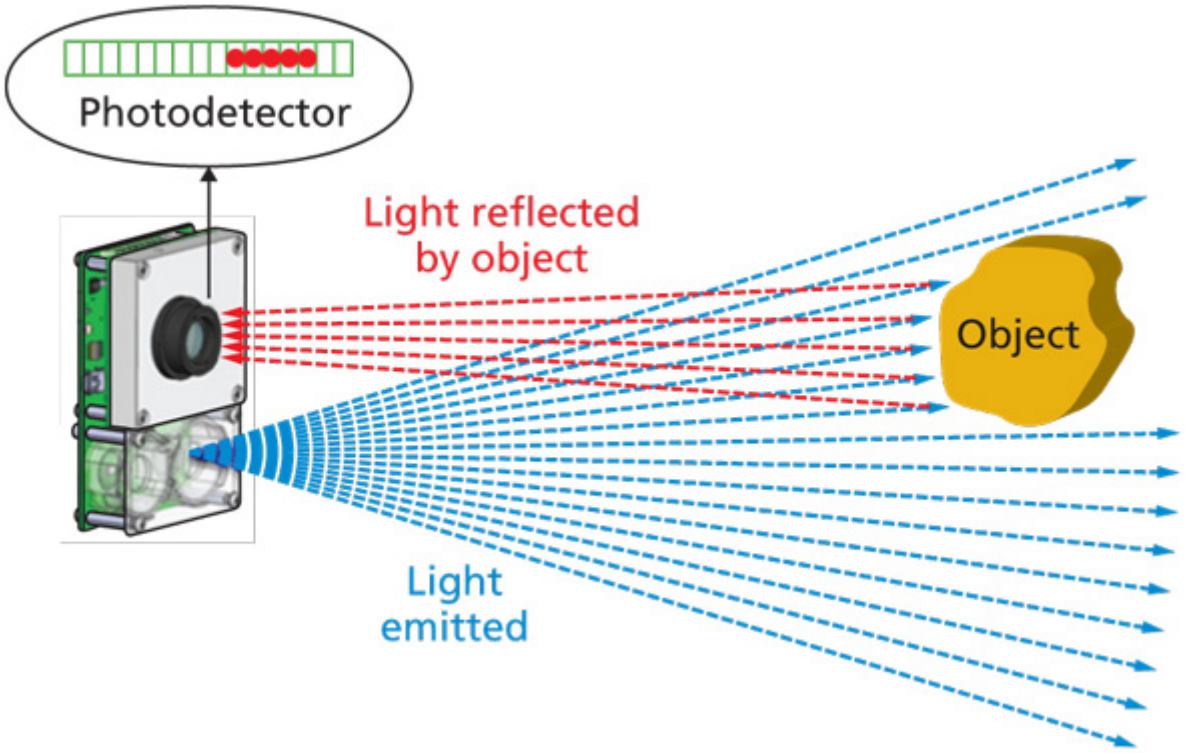

Illustration 6: This illustration shows an example of a scene where a infrared laser is illuminating an object, the reflected light can then be seen returning to the camera.
(Poulin, 2014)

2.4 Point Clouds

A point cloud is an arbitrarily ordered set of Cartesian points in space. The most common use of point clouds is as the standard output of surveying devises such as LiDAR scanners, these scanners measure the surface topology of objects as a number of points.

2.4.1 Point Cloud Library

The point cloud library (PCL) (*PointCloudLibrary/pcl: Point Cloud Library (PCL)*, 2018) is an open source framework facilitating the processing of point clouds. Processing that the PCL is capable of includes, the removal of extraneous or outlying points, the registration of one point cloud to another, the segmentation of one point cloud into two or more parts and mesh reconstruction from point clouds.

The point cloud library is supported by quite a large proportion of the technology industry including, Nvidia and Toyota. (*PCL Developers Blog*, 2015)

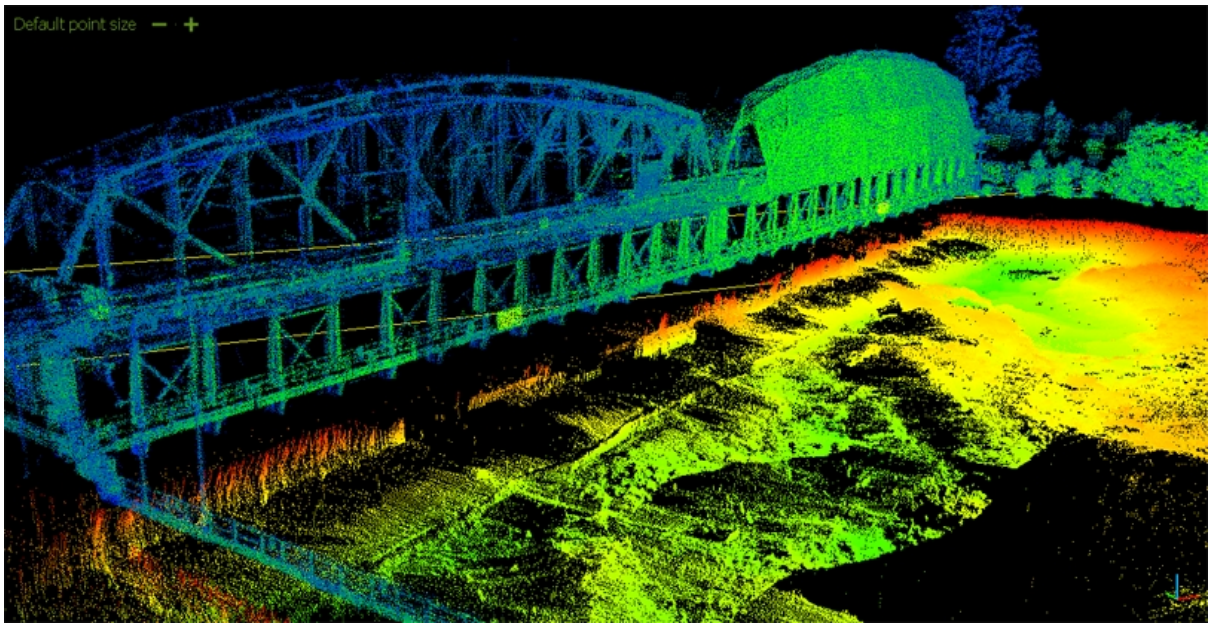

Illustration 7: This illustration shows an example of a point cloud, this point cloud shows a scan of a bridge crossing a river.
(*Point Cloud Processing / Data Management | H2H Associates*, 2018)

2.5 Reliant Technologies

2.5.1 OpenKinect Libfreenect Driver

OpenKinect Libfreenect (*OpenKinect/libfreenect: Drivers and libraries for the Xbox Kinect device on Windows, Linux, and OS X*, 2018) and OpenKinect Libfreenect2 (*OpenKinect/libfreenect2: Open source drivers for the Kinect for Windows v2 device*, 2018) are open source drivers for the Kinect camera version one and Kinect camera version two respectively.

This application will make use of the OpenKinect Libfreenect and OpenKinect Libfreenect2 drivers as a way to control the motor and accelerometer of the Kinect cameras and as a way to retrieve depth and RGB information from the Kinect cameras.

2.5.2 Qt Graphical User Interface

Qt graphical user interface (GUI) (*qt/qtbase: Qt Base (Core, Gui, Widgets, Network, ...)*, 2018) is an open source framework for creating cross platform native GUIs in C++ applications.

This application will make use of Qt GUI as a way of providing a stand alone GUI for its wrapper application when being run as a stand-alone application.

2.5.3STIR Framework

STIR (*UCL/STIR: Software for Tomographic Image Reconstruction*, 2018) is an open source framework for the reconstruction of images from PET data developed at University College London by the Institute of Nuclear Medicine.

STIR may make use of this application as a library to provide motion correction facilities for PET data.

2.5.4SAvVy

SavVy (*NikEfth/SAvVy: SAvVy - Stir dAta Viewer*, 2018) is an open source application for viewing sinograms developed using the Qt GUI for its front-end and the STIR framework as its back-end, SavVy was developed at the University of Hull by Dr Nikolas Efthymiou.

SavVy may make use of this application as a plug-in to provide motion correction facilities for PET data.

3 Aims, Hypothesis and Objectives

The aim of this project is:

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

The hypothesis for 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 imaging 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 imaging 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 and 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 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 PCL 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 PCL 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 acquired 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 transformation extracted from the Kinect camera to be relevant it is important to know when this vector field or transformation was extracted in relation to the output of the PET scanner, this is to ensure that the

correct vector field or transformation 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 transformation 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 transformation 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 transformation 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.

Appendix A: 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.

Appendix B: Kinect Camera Version Comparison

Scanner	Scanner Type	Minimum Range (mm)	Maximum Range (mm)	Horizontal Resolution (px)	Vertical Resolution (px)	Scanning Rate (Hz)	Field of View (deg)
Kinect V1	Structured Light	400	4000	640	480	30	60
Kinect V2	Time of Flight	500	8000	512	424	30	70

Illustration 8: This illustration shows a comparison between the Kinect camera version one and the Kinect camera version two.

Appendix C: 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.

Appendix D: Time Plan Table

Task Name	Start Date	Duration	End Date
Research PET Scanners	31/05/18	25	24/06/18
Research Alternative Solutions	11/06/18	14	24/06/18
Research Kinect Camera	11/06/18	14	24/06/18
Research Point Cloud Library	11/06/18	14	24/06/18
Write initial report	31/05/18	26	25/06/18
Edit initial Report	11/06/18	15	25/06/18
Port STIR framework to Windows	11/06/18	14	24/06/18
Read data from the Kinect	11/06/18	7	17/06/18
Create stand-alone console application to interface with Kinect	25/06/18	7	01/07/18
Add ability to save point cloud from Kinect to file or data type	25/06/18	7	01/07/18
Add ability to output multiple point clouds per execution	25/06/18	7	01/07/18
Add ability to timestamp output	25/06/18	7	01/07/18
Add ability to clean point clouds	02/07/18	14	15/07/18
Add ability to register between point clouds	02/07/18	21	22/07/18
Add ability to extract vector field or translation from registered point clouds	09/07/18	21	29/07/18
Add ability to remove extraneous data from point cloud	16/07/18	14	29/07/18
Add ability to synchronise timestamp on point cloud to output from scanner	13/08/18	14	26/08/18
Add ability to translate camera space to scanner space	13/08/18	14	26/08/18
Add ability to apply output from application to output from scanner	20/08/18	21	09/09/18
Write final report	30/07/18	54	21/09/18
Edit final report	27/08/18	26	21/09/18
Port application to Windows	03/09/18	13	15/09/18
Add application to the Windows version of STIR as library	03/09/18	13	15/09/18

Illustration 9: This illustration shows the data which was used to generate the time plan Gantt chart.

Appendix E: Time Plan Gantt Chart

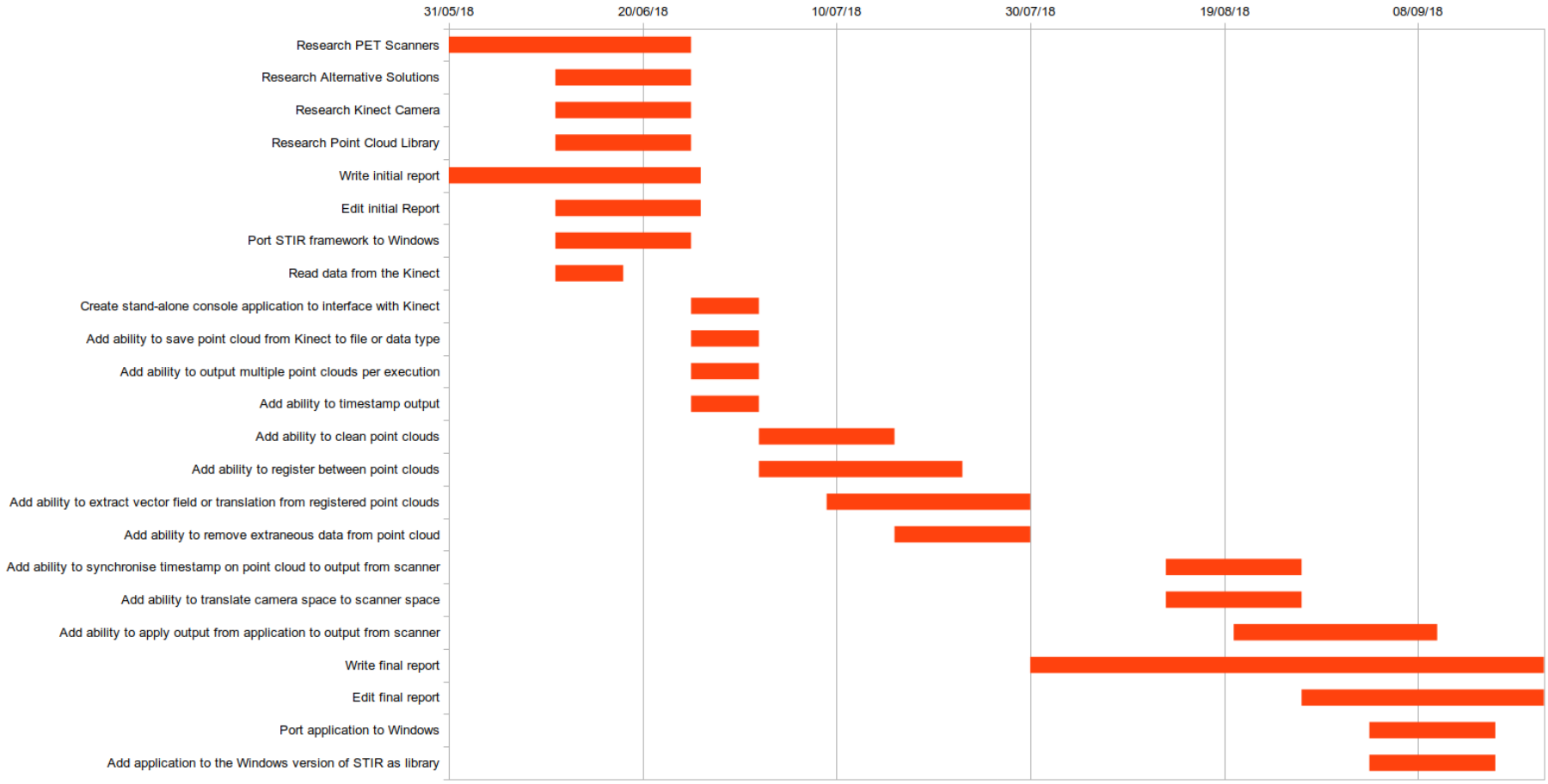

Illustration 10: This illustration shows the time plan data as represented by a Gantt chart.

Appendix F: 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.

Appendix G: 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.

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).

Beyer, T. *et al.* (2003) 'Dual-modality PET/CT imaging: the effect of respiratory motion on combined image quality in clinical oncology', *European Journal of Nuclear Medicine and Molecular Imaging*. Springer-Verlag, 30(4), pp. 588–596. doi: 10.1007/s00259-002-1097-6.

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.

Khoshelham, K. and Elberink, S. O. (2012) 'Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications', *Sensors*. Molecular Diversity Preservation International, 12(2), pp. 1437–1454. doi: 10.3390/s120201437.

Lau, D. (2013) *Gamasutra: Daniel Lau's Blog - The Science Behind Kinects or Kinect 1.0 versus 2.0*. Available at: https://www.gamasutra.com/blogs/DanielLau/20131127/205820/The_Science_Behind_Kinects_or_Kinect_10_versus_20.php (Accessed: 24 June 2018).

Lindner, M., Schiller, I. and Koch, R. (2010) 'Time-of-Flight sensor calibration for accurate range sensing', *Computer Vision and Image Understanding*. Academic Press, 114(12), pp. 1318–1328. doi: 10.1016/J.CVIU.2009.11.002.

McClelland, J. R. *et al.* (2017) 'A generalized framework unifying image registration and respiratory motion models and incorporating image reconstruction, for partial image data or full images', *Physics in Medicine and Biology*. IOP Publishing, 62(11), pp. 4273–4292. doi: 10.1088/1361-6560/aa6070.

Miranda, A. *et al.* (2017) 'Markerless rat head motion tracking using structured light for brain PET imaging of unrestrained awake small animals', *Physics in Medicine and Biology*. IOP Publishing, 62(5), pp. 1744–1758. doi: 10.1088/1361-6560/aa5a46.

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).

NikEfth/SAvVy: SAvVy - Stir dAta Viewer (2018). Available at: https://github.com/NikEfth/SAvVy (Accessed: 25 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.

OpenKinect/libfreenect: Drivers and libraries for the Xbox Kinect device on Windows, Linux, and OS X (2018). Available at: https://github.com/OpenKinect/libfreenect (Accessed: 25 June 2018).

OpenKinect/libfreenect2: Open source drivers for the Kinect for Windows v2 device (2018). Available at: https://github.com/OpenKinect/libfreenect2 (Accessed: 25 June 2018).

PCL Developers Blog (2015). Available at: http://pointclouds.org/blog/ (Accessed: 25 June 2018).

Point Cloud Processing / Data Management | H2H Associates (2018). Available at: http://h2hassociates.com/services/point-cloud-processing-data-management/ (Accessed: 23 June 2018).

PointCloudLibrary/pcl: Point Cloud Library (PCL) (2018). Available at: https://github.com/PointCloudLibrary/pcl (Accessed: 25 June 2018).

Poulin, M. (2014) *LEDs and sensing technology enable new ranging applications (MAGAZINE) - LEDs*. Available at: https://www.ledsmagazine.com/articles/print/volume-11/issue-6/features/developer-forum/leds-and-sensing-technology-enable-new-ranging-applications.html (Accessed: 24 June 2018).

qt/qtbase: Qt Base (Core, Gui, Widgets, Network, ...) (2018). Available at: https://github.com/qt/qtbase (Accessed: 25 June 2018).

Ren, S. *et al.* (2017) 'Data-driven event-by-event respiratory motion correction using TOF PET list-mode centroid of distribution', *Physics in Medicine and Biology*. IOP Publishing, 62(12), pp. 4741–4755. doi: 10.1088/1361-6560/aa700c.

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).

UCL/STIR: Software for Tomographic Image Reconstruction (2018). Available at: https://github.com/UCL/STIR (Accessed: 25 June 2018).

Zhang, Z. (2012) 'Microsoft Kinect Sensor and Its Effect', *IEEE Multimedia*, 19(2), pp. 4–10. doi: 10.1109/MMUL.2012.24.