Improvements on the Planning and Delivery of Intensity-Modulated Arc Therapy

Grace Tang B.Sc., M.Sc.

Department of Medical Physics and Bioengineering
UCL

A Thesis Presented as Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Radiation Physics

April 22, 2010

Primary Supervisor at UCL:
Dr. Ivan Rosenberg

Secondary Supervisor at UCL:
Professor Robert D Speller

Primary Supervisor at University of Maryland:
Professor Cedric X Yu

Secondary Supervisor at University of Maryland:
Dr. Matthew A Earl

I, Grace Tang, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

To my Grandpa, Grandma, Mum, Dad and little brother

Acknowledgments

It is difficult to begin the acknowledgment list. There are too many people to thank and without whom, this thesis would not be possible.

First, I would like to thank Dr. Rosenberg and Professor Speller for their noble support. It was an unusual situation when I decided to go to the United States for the Ph.D. project but both of them were very understanding and have been very supportive throughout the process. I must also thank them for their guidance and patience while I was in the US, helping me to stay on track.

Secondly, I would like to thank Professor Yu for his generous offer and giving me the opportunity to study in the US. It was very lucky of me to participate in such an exciting and timely research project. He has been a selfless teacher and mentor, and was always willing to share his wisdom and knowledge in medical physics. I feel very grateful for his guidance on the project while he was also very open to my ideas.

I would also like to thank Dr. Earl, who was my secondary supervisor in the US. I have learnt a lot from him and received a lot of help from him. In addition to Dr. Earl, I would like to thank Dr. Shahid Naqvi and Dr. Byong Yong Yi at the University of Maryland who are very knowledgeable medical physicists. I thank them for their patience as I asked them questions 10 times a day, every day, 5 days a week.

There are various components in this project where I have collaborated with other institutions. I would like to specially thank Dr. Daliang Cao at Swedish Cancer Institute, Dr. Shuang Luan at the University of New Mexico and Dr. Chao Wang at the University of Notre Dame for their help on the

IMAT algorithms. I would also like to thank the physicists, dosimetrists and physicians at the Department of Radiation Oncology of the University of Maryland for their help, hospitality and friendship, where some of whom I have spent many late nights working in the department with.

Last but not least, I would like to thank all the love and support from my friends and family, especially my grandparents, my parents and my little brother. I had decided to follow my father's footstep since I was 6 years old. By completing a Ph.D. degree, I hope I am a step closer to becoming a respectable medical physicist like him and help a lot of people.

Abstract

In the past decade, intensity-modulated radiation therapy (IMRT) has taken a significant step towards dose conformality and has now become a standard radiotherapy technique in the clinic. In this era, a rotational IMRT technique called intensity-modulated arc therapy (IMAT) was also proposed to possibly further reduce normal tissue toxicity and compete with conventional IMRT. However, clinical implementation of IMAT had been stagnant primarily due to the lack of mature planning and delivery systems.

In this study, various aspects of treatment planning and delivery of IMAT have been investigated and improved. The dosimetric accuracy and computational efficiency of IMAT planning has been greatly augmented by the use of Monte Carlo technique which is immune to the large number of discrete beams in approximating a continuous rotation as compared with traditional arc calculation methods. An efficient single-arc form of IMAT delivery has also been explored and extended in contrast to the original multi-arc IMAT. Here the clinical feasibility of single-arc IMAT was established by comparing to multi-arc IMAT and conventional IMRT. It was demonstrated that when using multiple arcs, the requirements on aperture shape connectivity incurred fewer constraints on the optimisation so that the plan quality became the best among the three methods studied although the dosimetric differences among them were generally small and considered clinically insignificant. Nevertheless, single-arc IMAT was able to provide a plan quality in between multi-arc IMAT and fixed-field IMRT with a significant delivery efficiency advantage.

Single-arc IMAT may require dose-rate variation for delivery, which is only available with the new treatment machines. To expand the clinical utilisa-

tion, an alternative planning and delivery approach was developed such that single-arc IMAT can be delivered using constant dose-rate with the existing machines, sparing the expensive and time-consuming upgrades.

Clarifications

The works presented in this thesis are based on original ideas. However, there are a few tools and resources which I have used with permission from the original owners to support the investigations. These are:

1. Monte Carlo kernel based convolution/superposition dose engine from Dr. Shahid Naqvi at University of Maryland

2. Continuous intensity-map optimisation from Dr. Daliang Cao at Swedish Cancer Institute

3. K-link IMAT leaf sequencing algorithm from Dr. Shuang Luan at University of New Mexico

4. Arc-modulated radiation therapy algorithm from Dr. Chao Wang at University of Notre Dame

Along with other external resources used, the algorithms stated above are also explicitly identified in the text using footers.

Similarly, the original works and tools developed in this project are differentiated by tagging appendix pointers when they are first introduced in the text. The appendices give the full description of the computer codes of the various tools and algorithms that I have written and developed.

Contents

List of Abbreviations

2, 3, 4D 2, 3, 4-dimensional

3D-CRT 3D-Conformal Radiation Therapy

AM Amplitude Modulation

AMRT Arc-Modulated Radiation Therapy

BEV Beam's Eye View

BOT Beam-On Time

CAT Conformal Arc Therapy

CCC Collapsed Cone Convolution

CDR Constant Dose-Rate

CIMO Continuous Intensity Map Optimisation

CNS Central Nervous System

CPU Computational

CT Computed Tomography

CVSP Convolution/Superposition

DAO Direct Aperture Optimisation

DICOM Digital Imaging and Communications in Medicine

DVH Dose-Volume Histogram

EBRT External Beam Radiation Therapy

EPID Electronic Portal Imaging Device

FM Frequency Modulation

HI Homogeneity Index

HN Head-and-Neck

ICRU International Commission on Radiation Units and Measurements

IGRT Image-Guided Radiation Therapy

IMAT Intensity-Modulated Arc Therapy

IMRT Intensity-Modulated Radiation Therapy

linac Linear Accelerator

MC Monte Carlo

MLC Multileaf Collimators

MOT Mode-Up Time

MRI Magnetic Resonance Imaging

MU Monitor Units

OAR Organ-At-Risk

PBC Pencil Beam Convolution

PET Positron Emission Tomography

QA Quality Assurance

ROI Region of Interests

RTOG Radiation Therapy Oncology Group

SAD Source-to-Axis Distance

SPECT Single Photon Emission Computed Tomography

TERMA Total Energy Released Per Unit Mass

TPS Treatment Planning System

VDR Variable Dose-Rate

VMAT Volumetric-Modulated Arc Therapy

xMOT Extra Mode-Up Time

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Cancer and its treatments

Cancer is a type of disease where the cells undergo abnormal growth that invades and destructs normal healthy tissues, affecting the patient's anatomical functionality. There are two categories of tumour: benign and malignant. Benign tumours are localised tumours which do not spread to other parts of the body and can be easily controlled by removal whereas malignant tumours metastasise and spread to other locations of the body, increasing the challenge of cure. Each year, millions of people in the world die from malignant cancer, affecting all ages, races, and genders.

The type of cancer treatment is usually suggested by the physician based on the site, grade, and staging of the disease. Surgery, chemotherapy, and radiation therapy are the typical treatments for cancer, although other methods are also used, such as hormone therapy, targeted therapy, and immunotherapy. Depending on the location of the tumour, surgery can removal all non-haematological tumours. However, to ensure the residual microscopic disease is under control, surgery is often combined with chemotherapy or radiation therapy.

1.2 Radiation in cancer treatments

Since the discovery of x-rays in 1895, radiation has been used for cancer treatment for more than a century. Radiation can be administered externally or internally. The appropriate choice of the external or internal radiation treatments is justified by the location and stage of the disease. External radiation therapy (EBRT) delivers ionising radiation to the patient from a distance and is most commonly implemented using medical linear accelerators (linacs). Different types of linacs or machines can produce different types of radiation, such as photons, electrons, protons, neutrons, and heavy ions. Typically, a radiation therapy centre is equipped with linacs that deliver photon beams and electron beams at various energy range while some centres provide radiotherapy services with neutrons and charged particles.

For localised treatments, sealed or unsealed radioactive source can be directly placed inside the patient to deliver radiation. The total integral dose to the patient may be reduced as the radiation source does not need to travel from the outer of the patient to the target inside the patient. Such techniques are called brachytherapy. Different from EBRT, brachytherapy may be a more invasive treatment depending on the type of procedures. For example, in a prostate brachtherapy treatment, radioactive seeds are implanted inside the patient under surgical operation. Extra patient care may also be needed for brachytherapy as some treatments require patient isolation and monitor until the radioactivity level of the patient is stabilised and the treatment is completed.

1.2.1 Radiobiology of cancer

The principle of radiation therapy is to kill cancer cells with a lethal dosage of ionising radiation. Precisely, ionising radiation is used to cause DNA damage and subsequently destroy the malignant cancer cells. There are two types of ionising radiation: direct ionising radiation and indirect ionising radiation. Direct ionising radiation refers to the type of radiation that directly ionise or excite the target atoms through Coulomb interactions and deposit energy directly, which leads to a chain of physical and chemical events and reactions that results in biological damage to the target cells. Charged particles such as electron, proton, and alpha particle are directly ionising radiation. Indirectly ionising radiation is uncharged radiation, such as photons and neutrons, which liberate charged particles from the atoms of the material that it interacts with, thereby causing ionisation and biological damage. It is important to note that the biological effects of radiation do not solely occur in the cancer cells but also in the surrounding normal tissues. It is therefore important to maximise the dose to the target for cure rate while minimising the normal tissue toxicity for minimal complications in radiotherapy practices.

1.3 Overview of external radiotherapy in clinics

For treatment service quality and efficiency, a typical radiotherapy department setting consists of imaging and treatment devices such as computed tomography (CT) unit, treatment simulator, treatment planning computer systems and treatment delivery unit. When a patient is diagnosed with cancer and requires radiotherapy, a series of treatment planning procedures will be performed. The treatment planning process begins with the acquisition of the

patient's anatomical data using various imaging modalities such as CT. Based on the 3-dimensional (3D) anatomy information, a patient-specific treatment plan is generated using a computerised treatment planning system (TPS). To verify the allocation of the treatment fields and patient set-up alignment, the patient undergoes another x-ray screening at the treatment simulator where tattoos will be marked on the patient's skin for treatment set-up alignment purposes. Prior to the start of the treatment course, a quality assurance (QA) procedure is performed to verify the dosimetric and deliverability accuracies of the treatment plan.

1.3.1 Patient simulation

Before a treatment plan can be generated, the patient's anatomical data must be acquired. With the current general practice, 3D anatomy data is taken using CT due to the excellent imaging quality, e.g. soft tissue contrast. More importantly, the CT images are essentially the maps of attenuation coefficients of the patient's body. Electron density data can be derived from these attenuation coefficient information, which is needed for inhomogeneous dose calculation of the treatment plans. In addition to CT simulation, other imaging modalities such as magnetic resonance imaging (MRI), positron emission tomography (PET), and single photon emission computed tomography (SPECT) can be used. For clearer delineation of the anatomical structures and special requirements of tissue contrast, datasets from different imaging techniques can be combined or fused together. For example, for brain cases, MRI images can be fused with CT images such that excellent brain tissue contrast can be achieved with MRI while the electron density information is obtained with CT. Furthermore, contrast agents can be used during CT and

MRI acquisitions to enhance the visibility of soft tissues.

1.3.2 Treatment planning

Once the patient CT data are acquired, the images are imported into a TPS for the generation of a treatment plan. Different anatomical structures or regions of interests (ROIs) are defined and delineated so that different tissue volumes can be identified by the TPS and proceed with the appropriate planning procedures such as plan optimisation and plan quality evaluation. For example, the tumour target and the surrounding organs-at-risk (OARs) are contoured so that the planner can designate the appropriate treatment beam directions, sizes, and shapes. The International Commission on Radiation Units and Measurements (ICRU) reports 50 and 62 provide general guidelines to the target and critical organ volumes definitions[ICRU 1993][ICRU 1999]. There are four different definitions of target volume: 1) "The gross tumour volume (GTV) is the gross palpable or visible/demonstrable extent and location of malignant growth". For the curative purpose of radiation therapy, the entire GTV must receive 100% of prescription dose as a minimum requirement. 2) "The clinical target volume (CTV) is the tissue volume that contains a demonstrable GTV and/or sub-clinical microscopic malignant disease, which has to be eliminated. This volume thus has to be treated adequately in order to achieve the aim of therapy, cure or palliation". Typically, CTV can be an expansion of the GTV or it can be a surrounding tissue volume that contains microscopic disease, e.g. positive lymph nodes and lymph nodes at risk. CTV can receive different doses from the GTV, depending on the treatment goal that is determined by a radiation oncologist. 3) As a supplement to ICRU report 50, ICRU report 62 defines the internal target volume (ITV) as an

expanded volume of the CTV with an internal margin. The internal margin accounts for the variations in size and position of the CTV due to anatomical or organ motions such as respiratory motion and abdominal motion. 4) "The planning target volume (PTV) is a geometrical concept, and it is defined to select appropriate beam arrangements, taking into consideration the net effect of all possible geometrical variations, in order to ensure that the prescribed dose is actually absorbed in the CTV". The PTV includes the ITV with an expanded margin that takes into the account of various uncertainties such as patient set-up, ROI delineation, and intrafraction or intratreatment geometrical variations. Typically, the expansion margin of PTV depends on the type of treatment and the use of patient immobilisation devices. Alongside with the GTV, delivering sufficient radiation dose to the PTV is one of the fundamental treatment goals. Usually, the dose of the treatment course is prescribed to the PTV, e.g. at least 95% of the PTV should receive 95% of the prescription dose.

In addition to prescribing the appropriate dosage to the targets, physicians may also specify the maximum dose limits to the surrounding normal tissues or OARs. The planner will then attempt to generate a treatment plan that fulfills these dose requirements. When a conflict occurs between sufficient dose coverage to the target and adequate normal tissue sparing, the physicians will decide which one should take preference for the benefits to the patient and the objectives of the treatment. The plan quality can be evaluated using dose-volume histograms (DVHs) and isodose distributions. While DVH provides a good overview of the treatment plan quantitatively, it does not obtain any spatial information of the dose distribution; whereas isodose distribution offers analysis on the geometrical distribution of the dose quantitatively and

qualitatively.

1.3.3 Treatment verification

After the treatment plan has been generated, various QA procedures must be performed to verify the dosimetric accuracy of the TPS and the deliverability of the linac prior the actual treatment. As part of the clinical requirements, dosimetric verification of treatment plans is important to ensure the dose distributions will be delivered as planned so that errors can be minimised and assure patient safety. In general, QA in radiotherapy includes two separate parts: 1) machine QA and 2) patient-specific QA. Routine QA on the linac is crucial to monitor the stability of the machine performance so that any hardware faults and errors can be detected as early as possible. Typically, machine QA involves several mechanical tests, safety interlock checks, and dosimetry consistency surveys. Mechanical tests may involve inspecting the stability and accuracy of various physical components of the machine such as cross-hair alignment and the coincidence of the light and radiation fields; safety interlock checks may involve verifying the functionality of various safety control systems such as door interlock and emergency switch; dosimetry consistency surveys may involve checking or calibrating various dosimetric parameters such as radiation output and properties of beam modifiers (e.g. output and transmission factors of wedge and multileaf collimators (MLC)). In general practice, machine QA is performed daily, weekly, monthly and annually, with different levels of QA procedures.

Different from machine QA, patient-specific QA must be performed prior the start of the treatment course (note that it may not be necessary to perform machine QA for every patient-specific QA session, e.g. the simple treatments

that allow simple manual calculations). The main purpose of patient-specific QA is to verify the delivery accuracy of the plan, i.e. the MLC sequence (note: a MLC sequence contains the information of the geometrical positions of MLC that define the field shape and the corresponding monitor units (MU), i.e. dose) . The couch angle, gantry angle, and collimator angle are also part of the verification test. Typically, a QA plan is delivered to a QA phantom for dosimetric verification. Detectors such as films, electronic portal imaging device (EPID), and ion chambers can be used to measure dose. With films, EPID, and other array detectors, the dose maps or fluence maps can be measured to compare with the corresponding 2D dose planes that are calculated by the TPS. The analysis of the measurement and the calculation can be performed using QA softwares or in-house computer programs. The measurement from delivery and the calculation from TPS should be in agreement within certain criteria that satisfies the clinic protocol and national standards.

1.3.4 Treatment delivery

During treatment delivery, it is crucial that the patient is set up and positioned correctly such that radiation dose is delivered as planned. Aligning the external body markers such as tattoos with the linac's laser alignment system is a basic patient set-up procedure. However, patient set-up using merely external body markers may not be accurate since there is no fixed relationship between the external body markers and the internal organs, i.e. the patient is not a rigid body. In addition to aligning the patient with the skin markers, bony alignment procedure can be conducted for a more accurate set-up. Fluoroscopy using kilovoltage x-ray beams can be taken to visualise the bone structures of the patient so that it can be aligned with that in the treatment

plan. Nonetheless, tissues and organs can move independently from the rigid bone structures. For soft tissue or target localisation, clinics often use imaging markers to distinct the anatomy volumes in the images. For example, in prostate cases, gold seeds can be surgically implanted into the target so that the location of the tumour can be easily identified in fluoroscopic images, while others use beacon seeds and radiofrequency to track the location of the target. Both of these methods are invasive to patient since surgery is required to implant the fiducial markers. Non-ionising radiation such as ultrasound has been used as an alternative approach for patient set-up, eliminating the use of fiducial markers and the extra radiation dose given to the patient.

These various set-up techniques using different imaging modalities are regarded as image-guided radiation therapy (IGRT). The original concept of IGRT includes patient set-up management prior treatment and target motion control during treatment. Target motion control or the so-called 4D treatment (time is referred to as the 4th dimension in addition to the 3D spatial dimensions) using gating or tracking techniques are still under developments with limited clinical adoptions. Gating or tracking techniques require real-time patient data such that the delivery system can be updated with the correct spatial information of the mobile target and make appropriate adjustments online during delivery. For example, with the real-time respiratory data of the patient, the treatment beam can be "gated" such that the beam is turned on when the target is within the gating window and the beam is turned off when the target moves out of the gating window. Alternatively, the machine can make corrections to the field sizes and shapes of the beam according to the changing positions of the tumour. One of the difficult challenges in such 4D treatment techniques is the accurate collection of the real-time patient data

as well as the communication and adaption of the treatment machine to these data. Currently, most of the real-time tumour location information is reported based on an external surrogate. Thus, it is crucial to obtain an accurate and reliable correlation between the real position of the tumour inside the body and the external surrogate for the purpose of IGRT.

Chapter 2

Rotational radiotherapy techniques

2.1 Rotational therapy

Since the discovery of x-rays in 1895, it has been rapidly adapted for medical applications such as medical imaging and cancer therapy. Before the invention of megavoltage linacs, orthovoltage and cobalt-60 treatment units were mostly used in the clinic. Orthovoltage machines generate x-rays with the energy range of 150 to 500 kV and such kilovoltage beams exhibit fast dose fall-off with depth which is beneficial for treating superficial tumours. On the other hand, cobalt-60 units provide low energy megavoltage gamma rays that can penetrate deeper into the patients but with some skin-sparing effects. However, with only a few finite beam angles, it is difficult to deliver sufficient radiation dose to the deep-seated tumours without excessively irradiating the skin of the patient. Rotational delivery was developed for such scenarios. Radiation is delivered to the target with infinite beam angles in a rotational arc. Because the dose per beam angle is reduced, the skin of the patient can be spared while sufficient dose is delivered to the target. Nonetheless, without sophisticated beam modifying or beam blocking equipments and techniques, rotational therapy is only suitable for the cases where the tumour is symmet-

rically shaped and is centrally located in the body with minimal number of surrounding critical organs. With only rigid lead and cerroband blocks available, limited normal tissue sparing is achieved as the shape of the field cannot not be changed dynamically during the rotational delivery.

2.2 Conformal arc therapy

Conventional rotational therapy gradually faded out as megavoltage linacs were invented and implemented into the clinic. Unlike kilovoltage beam, megavoltage photon beam exhibits slower dose fall-off with a dose build-up region that provides adequate skin-sparing. In addition, the development of computer-controlled MLC enables the clinical feasibility and implementation of an arc therapy technique that was proposed in 1965 by Takahashi [Takahashi 1965] [Cheng *et al* 1989] [Galvin *et al* 1992] [Galvin *et al* 1993b]. The concept of conformal arc therapy (CAT) is to deliver a conformal dose distribution to the tumour as the beam or field shape dynamically conforms to the target at beam's eye view (BEV) at all angles in the delivery arc. Hence, normal tissue sparing is improved as the field size conforms to the target shape only and minimises the volume of the nearby healthy organs exposing in the treatment field. This field conformation technique is also used in the form of static beams, which is known as 3D conformal radiation therapy (3D-CRT).

2.3 Intensity-modulated radiation therapy

The intention of 3D-CRT is to spare the excessive irradiation to the surrounding normal tissues by conforming the treatment field to the shape of the target.

However, such field conformation is based on the 2D projection of the target at BEV and ignores the 3D variations in the target volume. Suppose a 3D target or PTV volume has a concave region that surrounds a nearby OAR as exemplified in 2.1 (a), if a beam enters from below, the BEV of this given beam angle will show discontinued volumes of the target with the OAR in between the PTV as shown in 2.1 (b). In this scenario, the treatment field is required to split into two fields in order to spare the OAR. However, by doing so, the target may be under-dosed especially in the region that is closest to the OAR. Hereby, the 2D field conformation technique is insufficient to optimise the balance between target dose coverage and normal tissue sparing with 3D-CRT.

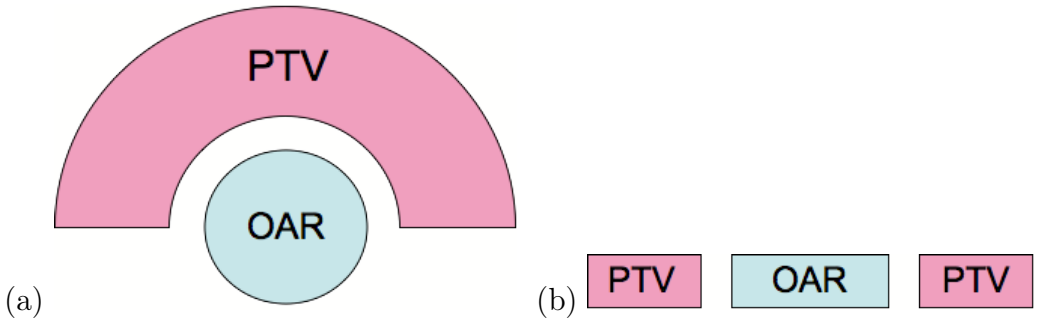


Figure 2.1: (a) Concave target surrounding a nearby OAR.

In 1982, Brahme *et al* discussed the preliminary concept of an inverse planning problem which is regarded as intensity-modulated radiation therapy (IMRT) today [Brahme *et al* 1982]. This seminal article presented a solution that provides uniform dose distribution to a ring-shaped target using a series of non-uniform intensity profiles at different angles in a rotational arc. Instead of requiring the planner to predict a certain dose pattern for each beam directions in the forward planning approach as in 3D-CRT, Brahme *et al* mathematically simulated the dose profiles or intensity profiles needed to achieve a given dose

distribution. Typically, 3D-CRT plans obtain beams with uniform intensity levels although non-uniform intensity levels can be achieved by the field-in-field technique or beam modifiers such as wedge blocks. Nonetheless, these techniques may not be sufficient and efficient to achieve complex non-uniform intensity profiles. Following the publication by Brahme *et al* in 1982, numerous investigations have been carried out for the inverse planning problem in IMRT [Langer & Leong 1987][Brahme 1988][Bortfeld *et al* 1990][Webb 1992] [Cho *et al* 1998][Sauer *et al* 1999]. The inverse planning problem is essentially computerised optimisation of the desired dose distribution. Further details of inverse planning will be discussed in Chapter 3.

Similar to CAT, due to the lack of treatment delivery tools, IMRT was not realised in the clinic until the mid 1990's. The advances in technology and the development of MLC have provided an efficient solution to deliver the optimised non-uniform intensity profiles. Due to the finite size of the MLC, the continuous intensity maps are translated into a set of MLC sequence that consists of the MLC aperture shapes and MU weightings. With the dynamic flexibility of the MLC leaves, complex shapes can be formed and two MLC modes have been used in IMRT: 1) static step-and-shoot MLC and 2) dynamic sliding window [Galvin *et al* 1993a][Bortfeld *et al* 1994][Xia & Verhey 1998] [Convery & Rosenbloom 1992][Spirou & Chui 1994][Stein *et al* 1994]. In static step-and-shoot IMRT, multiple MLC segments or apertures that are different in shapes and dose (or MU weightings) are superimposed at a given angle. The optimised varying intensity profiles at this beam angle is reproduced by stacking these MLC segments that deliver 2D uniform intensity levels. During treatment delivery, each of these MLC segments are delivered one after another at the planned beam angle with the radiation turned off as

the MLC transits from one segment shape to the next. In contrast, dynamic sliding-window IMRT may be a more efficient delivery process as the varying intensity profile is delivered continuously as a time function of the MLC leaf trajectory. Nonetheless, either delivery modes can produce highly modulated intensity profiles.

2.4 Tomotherapy

In 1993, the first hybrid of IMRT and rotational technique had emerged as tomotherapy[Mackie *et al* 1993]. The concept of tomotherapy is to deliver radiation treatment in a fan-beam geometry and the patient is irradiated slice-by-slice with a rotating source. Intensity modulation is obtained using a set of binary collimator that comprises two banks of tungsten leaves. The term "binary" represents the two discrete positions allowed for the collimator leaves: open or close. Intensity modulation is achieved by controlling the opening and closing times of the individual collimator leaves. Two delivery approaches have been developed for tomotherapy: serial tomotherapy and helical tomotherapy [Mackie *et al* 1993][Carol 1995]. Serial tomotherapy was developed by NOMOS Corporation [Carol 1995]. For treatment delivery, a binary collimator system (MIMiC^{TM}, NOMOS Corporation, Sewickley, PA) is added and mounted to the gantry of a linac as demonstrated in figures 2.2 (a) and (b). As the gantry is continuously rotating, the patient is treated one slice at a time (i.e. the slice width is defined by the collimation width of 1-2 cm) with a discrete treatment couch translation indexing from one slice to the next. To avoid unwanted over-dose areas or under-dose areas at the slice abutment regions, the treatment couch is required to move with a high

degree of precision [Carol *et al* 1996][Low & Mutic 1997]. Helical tomotherapy solves this problem by helically irradiating the patient with a dynamic couch movement (see figure 2.2 (c)). The delivery principle is similar to spiral CT and the treatment unit also emulates to a CT scanner as shown in figure 2.2 (d). Radiation is delivered in a fan-beam geometry with a rotating x-ray source as the patient is continuously translating through the gantry. Similar to serial tomotherapy, intensity modulation is achieved by using a binary collimation system in helical tomotherapy.

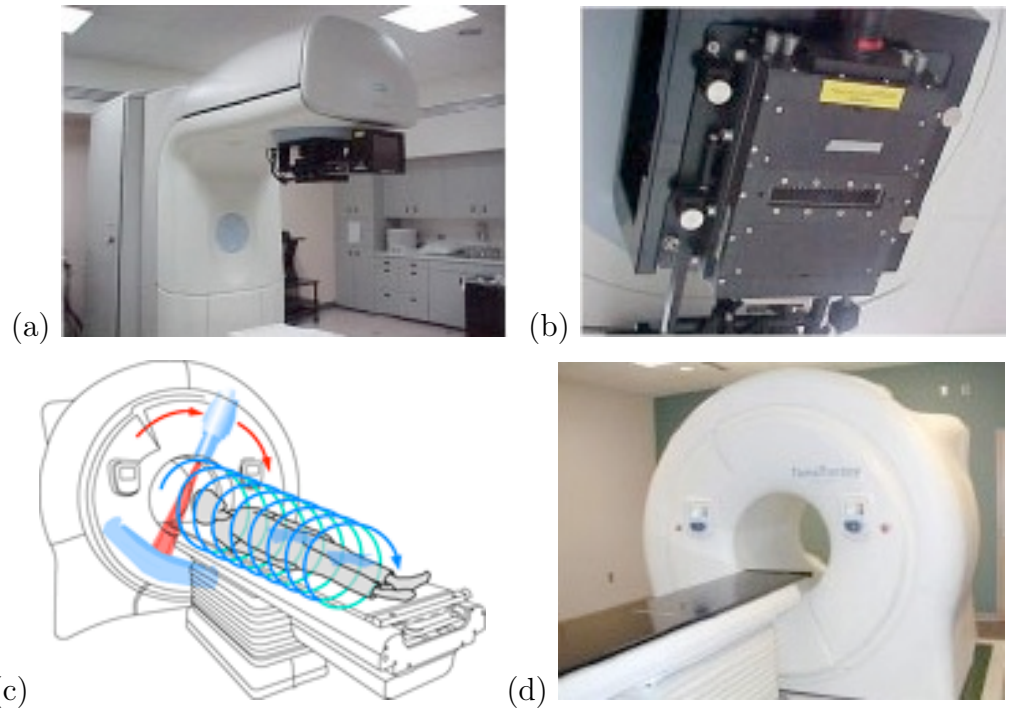


Figure 2.2: (a) A serial tomotherapy unit with the (b) NOMOS MIMic binary collimation system attached to a conventional linac. (c) Schematics of helical tomotherapy delivery. (d) A helical tomotherapy treatment unit.

Because the patient is treated slice-by-slice in a fan-beam geometry, highly conformal dose distribution can be obtained. Despite the superior dose distribution, tomotherapy has several shortcomings: 1) it requires a special machine for treatment delivery; 2) due to the limitation of the machine structure, non-

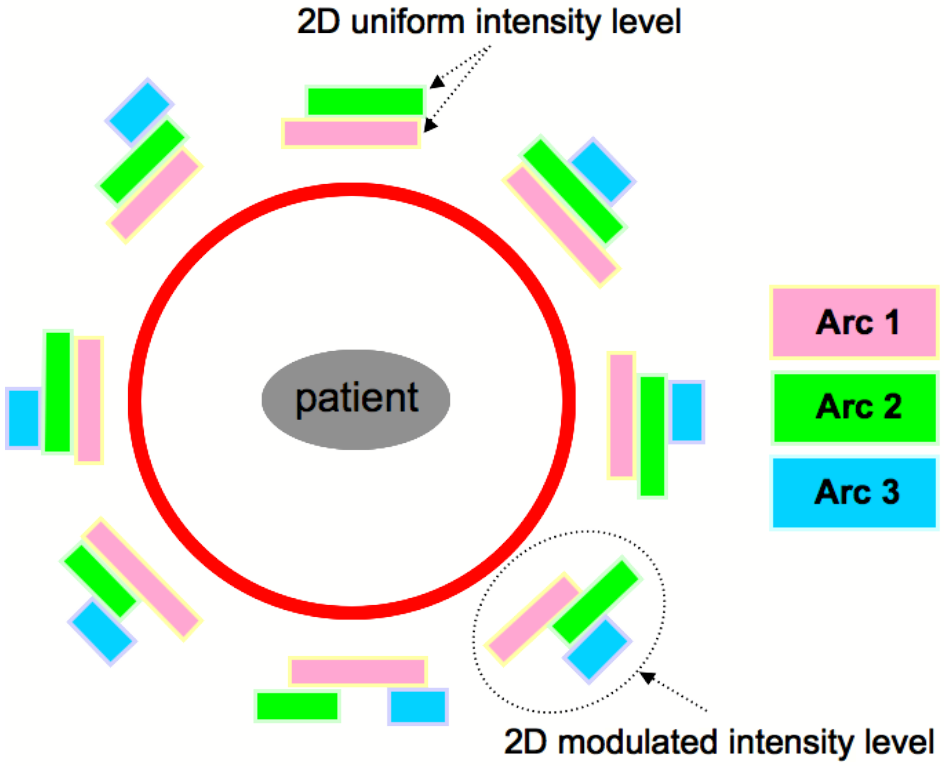
coplanar beams are not possible; 3) radiation penumbra can be large for a given jaw width and can introduce excessive dose to the normal tissues; 4) a treatment plan requires 32 computer processors for parallel processing and may require up to 1 hour of computational (CPU) time for dose calculation, which may add additional burden to the clinic workflow.

While issue (4) might be a technical limitation that can be solved in the near future, issues (1) and (2) are the intrinsic limitations on treatment delivery based on the design of the treatment unit. On the other hand, an alternative delivery approach using dynamic jaws was recently developed to overcome issue (3) [Sterzing *et al* 2010]. In conventional helical tomotherapy where a fixed jaw width is used, treatment delivery often results in broad dose penumbra in the superior and inferior edges of the tumour. Normal tissues in these regions subsequently receive excessive radiation. While narrowing the jaw width renders a sharper radiation penumbra, the overall treatment time will be increased. By dynamically changing the jaw width during treatment delivery, the normal tissues that lie in the superior and inferior of the target can be better shielded with a narrow jaw width, while the jaws will open wide for delivering radiation to the rest of the target efficiently.

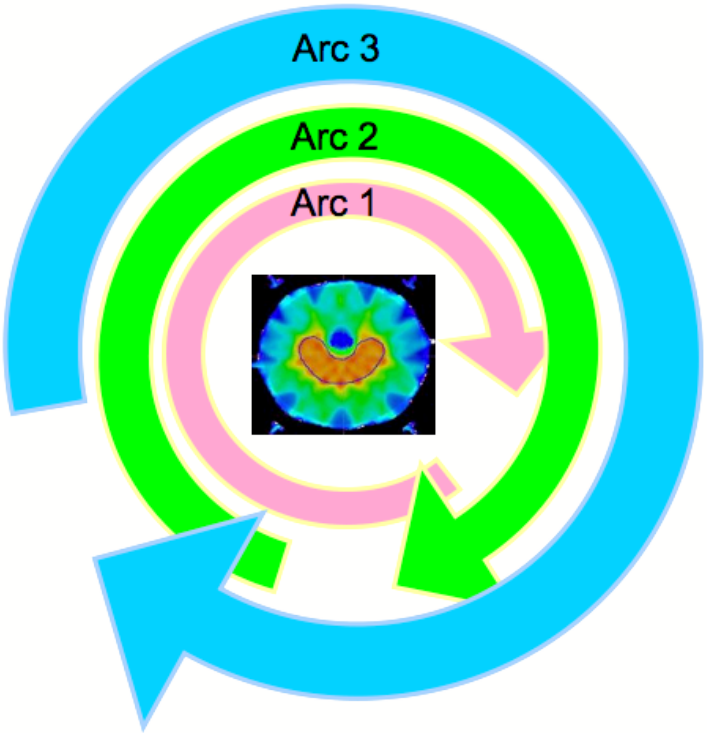
2.5 Intensity-modulated arc therapy

Proposed as an alternative to tomotherapy in 1995 by Yu, intensity-modulated arc therapy (IMAT) is another combination of IMRT and rotational therapy [Yu 1995]. Unlike tomotherapy, radiation is delivered in a cone-beam geometry in IMAT using a general-purpose linac instead of the fan-beam irradiation with a dedicated unit. Figure 2.3 (a) illustrates the principle of IMAT where

the delivery arc is approximated by 8 static beam angles. Each of the rectangular bars at each beam angle represents a 2D uniform intensity level that is defined by one MLC aperture. Intensity modulation is obtained by stacking these MLC apertures or 2D uniform intensity levels at a given beam angle. For delivery, one aperture from each beam angle is grouped into an arc as shown in 2.3 (b). During delivery, radiation is delivered with simultaneous gantry rotation about the patient and the MLC is continuously transitioning from one aperture to the next. In short, intensity modulation is achieved by multiple overlapping arcs. Typically, 5 to 12 arcs are used depending on the complexity of the patient case, which results in a delivery time of 6 to 15 minutes given the maximum gantry speed is 1 minute per 360-degree rotation. Further details in the treatment planning of IMAT is discussed in Chapter 3.



(a)



(b)

Figure 2.3: (a) Simple schematics of IMAT. (b) Delivery of IMAT.

Chapter 3

Treatment planning of IMAT

It is important to note that the treatment planning procedures discussed in this chapter can also be applied for IMRT treatment planning. In fact, most of the treatment planning strategies for IMAT were adopted from that developed for IMRT. The distinct difference and challenge in IMAT treatment planning, however, is the MLC connectivity between the adjacent planning angles. Due to the mechanical limits of the MLC and gantry speed, additional constraints are imposed on IMAT treatment planning process. Equation 3.1 defines the MLC connectivity constraint, i.e. the maximum displacement allowed for the MLC to travel between the adjacent planning beam angles, which is governed by the maximum speed of MLC v_{max}, maximum speed of gantry ω_{max}, and the angular spacing of the static planning beams $\Delta\theta$,

$$d_{max} = \frac{v_{max}\Delta\theta}{\omega_{max}}. \tag{3.1}$$

For simplicity, arc treatment planning often utilises a series of equi-spaced static beams that approximate the continuous delivery arc. The finer the angular spacing or the more number of static beams may increase the arc approximation resolution but the CPU burden will also be increased. An angular spacing of 10° has been used in previous studies and has shown to be an efficient planning parameter [Yu *et al* 2002][Earl *et al* 2003].

3.1 Forward planning approach

To create a treatment plan, the planner is first required to designate the suitable number of beams after the physician has defined and outlined the ROIs on the planning images. At each beam direction, the MLC-shaped aperture is set to conform the target at BEV while the MU weighting of each MLC aperture is manually adjusted. The planner may repeat this step several times before a satisfactory dose distribution is achieved. Such treatment planning approach has been used in 3D-CRT and CAT and is known as forward planning. Since it is a trial and error process, the efficiency of this planning technique may depend on the experience of the individual planner.

In 2002, University of Maryland had explored the clinical feasibility of IMAT using the forward planning approach [Yu *et al* 2002]. The delivery arc of the IMAT plans was approximated by a series of static beams with an even angular spacing of 5° to 10°. Similar to CAT, the MLC aperture was conformed to the target at the BEV at each planning beam angle. Intensity modulation was achieved at each beam angle with 2 to 5 overlapping arcs and each arc was assigned with different dose weightings. Since the speed of gantry rotation and dose rate were remained constant due to mechanical limitations, the MU weightings of all the apertures within the same arc were imposed to remain constant also. For a total of 50 patient case studies of 5 different disease sites (central nervous system (CNS), head-and-neck (HN), thoracic, gastrointestinal, and prostate), IMAT plans had obtained superior dose distributions compared to 3D-CRT. In particular, the normal tissue dose was greatly reduced in IMAT due to the larger number of beams while the dose was spread to a larger volume. With a clinically feasible treatment time

that is similar to 3D-CRT, these forward planned plans had established the clinical value of IMAT.

Similar to the University of Maryland study, Wong *et al* had also explored the clinical feasibility of IMAT using the forward treatment planning approach [Wong *et al* 2002]. In their study, IMAT delivery arcs were designated specifically and separately for the target and the OAR. For example, if the OAR partially overlaps with the target, one arc would be used to cover the entire target where the MLC apertures conform to the whole target volume at BEV with the overlapping OAR; and another arc would be used to cover the partial volume of the target where the MLC apertures conform to the target but exclude the volume that overlaps with the OAR. Although such forward planning strategies had been successfully implemented into the clinic [Bauman *et al* 2004], inverse planning may explore further potential of IMAT by optimising the intensity modulations and MLC sequences for the desired plan quality.

3.2 Inverse planning approach

Different from the forward planning approach, inverse planning minimises the trial-and-error routine in predetermining the MLC aperture shapes and weights. To achieve the desired dose distribution, the planner only needs to determine the number of beams and beam directions (note that some TPS can optimise the beam directions automatically also), and the plan quality wanted by specifying a few dose matrices. These dose matrices are also called optimisation constraints which are the inputs of an objective function [Center 2003]. An objective function is a mathematical formulation that defines and controls

the dose distribution of a given plan during optimisation. Equation 3.2 shows a generalised form of dose-based objective function that is used to define a uniform dose distribution of the target,

$$w \cdot (d_i - d_p)^2, \tag{3.2}$$

where w is the weighting or importance factor of the function, d_i is the current actual dose of the ith voxel, and d_p is the desired or prescription dose. A score is given to the objective function and the goal of optimisation is to minimise the score value until a global minimum is found, i.e. the optimal plan is achieved. For example, a PTV is prescribed with 70 Gy in a given plan. The primary goal is to achieve uniform dose in the PTV and the treatment planner assigns $d_p = 70$ Gy and $w = 100$ in equation 3.2. Note that the value of w is arbitrary and is relative to other sets of objective function, e.g. for a normal tissue that is of lower priority in the same plan, w can be defined as low as 1 (where $1 < w < 100$). Depending on the TPS or optimisation algorithm, w can be a fractional value instead of an integer. After the objective functions are fully defined, the iterative process of optimisation will be started. For each iteration, the dose of each voxel in the PTV i.e. d_i, is evaluated and compared against d_p. If d_i does not reach the planning goal, i.e. 70 Gy, the objective function score would increase in proportion to the absolute differences between d_i and d_p. This process repeats until the objective function score reaches a minimum and the optimal solution is found.

Similarly, a series of objective functions can be assigned to different targets and organ structures to achieve the desired plan quality. For normal tissues, objective functions can be defined to control the maximum dose or

other specific dose-volume limits. For example, equation 3.3 describes an objective function that limits the maximum dose d_{max} of a certain organ, with a weighting factor of w_{max}. The first part of the equation is a least square-difference function that determines the differences between the actual dose and the desired dose, while the second part of the equation is a step function, Θ, that administers penalty to the objective function and increases the score if the ith voxel exceeds the dose limits defined,

$$w_{max} \sum_i (d_i - d_{max})^2 \cdot \Theta(d_i - d_{max}). \tag{3.3}$$

$$\Theta(d_i - d_{max}) = \begin{cases} 1 & \text{if } (d_i - d_{max}) \geq 0 \\ 0 & \text{if } (d_i - d_{max}) \leq 0 \end{cases}$$

In short, the planner indicates the desired endpoint of the planning session with inverse planning. The TPS optimises the MLC delivery sequence that matches closest to these dosimetric specifications. Hence, the planning process may become less labour demanding and planner-experience dependent. This is particularly crucial for the treatment planning of IMRT techniques as the intensity modulation at each beam direction can be a complex function. In addition, the MLC sequence that is required to deliver such intensity modulation or dose distribution can be generated with many possible combinations. It is therefore too labour intensive and clinically inefficient to achieve so with forward planning.

There are two classes of inverse planning: 1) intensity-map based optimisa-

tion and 2) aperture-based optimisation. The intensity-map based optimisation is a 2-step planning approach whereas the aperture-based optimisation is a 1-step approach. Both of these inverse planning methods have been used for IMAT planning and the details of the two optimisation strategies are discussed in the following sections.

3.2.1 Intensity map-based optimisation

In intensity-map based optimisation, two sequential steps are required to generate an IMAT plan: 1) optimisation of a set of ideal intensity maps based on the static planning beams and 2) optimisation of a deliverable MLC sequence that reproduces the ideal intensity maps optimised in step 1. Figure 3.1 illustrates a fluence map of a given beam that is represented by a 2D matrix. Each element of the matrix is a beamlet or pixel of the field and the number in each matrix element indicates the photon fluence weighting or the intensity level, i.e. the intensity of this beam is modulated with 6 different intensity levels. Except for optimisation purposes, this fluence map, however, is meaningless in

1	2	1	1	1	2	0	0
2	1	2	3	3	1	0	0
2	3	4	4	3	2	1	0
3	4	5	5	4	3	2	1
3	6	5	5	5	5	3	1
4	6	6	6	6	5	2	1
2	4	6	6	6	5	2	2
1	3	6	6	6	5	3	1
1	2	5	5	5	5	3	2
1	2	3	4	4	5	3	1
1	2	4	4	3	3	2	1
1	1	2	3	3	2	1	1

Figure 3.1: Fluence map of a given beam.

terms of the actual delivery. For delivery, the fluence map must be translated or segmented into a MLC sequence. Figure 3.2 depicts a simple schematic of MLC segmentation of a modulating intensity profile. The continuous intensity function is approximated by a series of discrete uniform intensity levels where each of these uniform intensity levels is delivered by a MLC aperture. The grey area is the residual intensity after the segmentation. Since these grey area is not included in any apertures and are not being delivered, a loss of plan quality is resulted from this segmentation. Such dosimetric degradation can sometimes cause the overall dose distribution of the plan to fail the clinical acceptance criteria. As an attempt to retrieve the residual plan quality, a segment weight optimisation can be performed where the MLC positions remain unchanged and only the aperture MU weightings are being optimised.

Most of the commercial TPS support this intensity-map based optimisation approach for IMRT, but they are not fully suitable for IMAT planning. While the ideal intensity map optimisation can be used for IMAT planning, the additional MLC connectivity constraint required in IMAT delivery is not a part of the consideration in MLC leaf sequencing in these commercial TPS. Several IMAT sequencing algorithms have been reported, two of which were developed in-house at the University of Maryland: 1) continuous intensity map optimisation (CIMO) and 2) k-link IMAT sequencer [Cao *et al* 2006][Shepard *et al* 2007][Luan *et al* 2008]. Since both of these sequencing algorithms have been used for this project, they will be briefly discussed below.

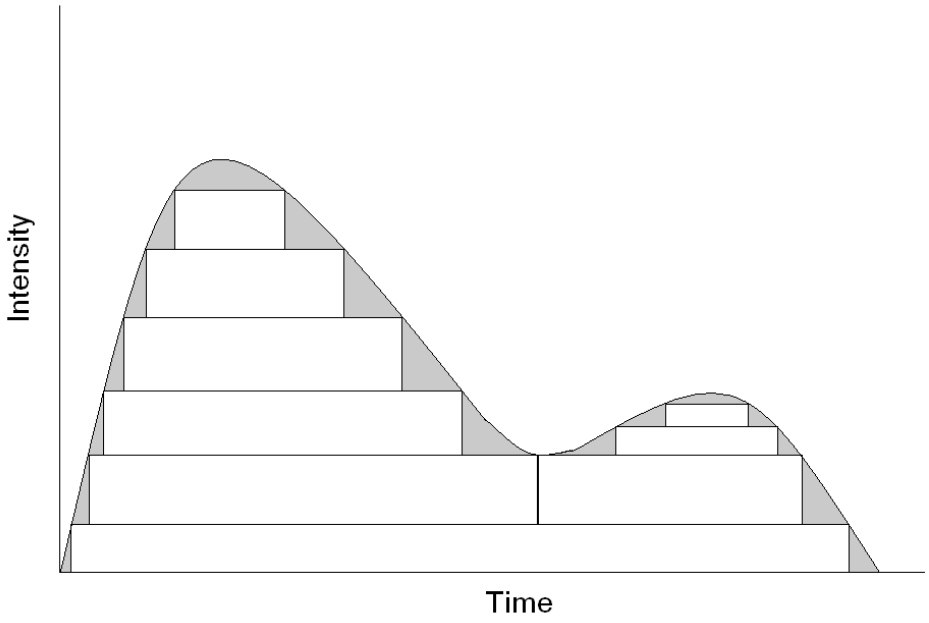


Figure 3.2: Simple schematic of MLC sequencing using the segmental method.

3.2.1.1 Continuous intensity map optimisation

As discussed, most of the leaf sequencing methods segment or approximate the continuous varying intensity function with discrete intensity levels. In contrast, CIMO optimises the MLC sequence against the ideal intensity maps [Cao *et al* 2006][Shepard *et al* 2007]. Based on the maximum number of arcs and the MLC displacement constraint that is specified by the planner, CIMO first generates a set of arbitrary MLC sequence with aperture shapes conforming to the target at BEV and equal MU weightings. The MLC positions and the MU weightings are then optimised simultaneously so that the differences between the resultant intensity maps and the ideal intensity maps is minimised. The optimisation algorithm is based on the simulated annealing technique. However, similar to other simulated annealing algorithms, CIMO randomly selects the changes in leaf positions and aperture weights based on

random seeds while the global minimum solution is not guaranteed.

3.2.1.2 K-link IMAT sequencer

In collaboration with University of Maryland, University of New Mexico and University of Notre Dame, the k-link IMAT sequencer was developed based on the k-link theory from graph algorithms in computer science [Luan *et al* 2008]. With a set of ideal fluence maps, the planner defines k number of arcs where each set of fluence maps (i.e. each planning beam direction) is segmented with k MLC apertures. The same as CIMO, the number of arcs used depends on the plan complexity. By using the k-link shortest path algorithm, each of the k apertures of each planning beam angle is connected to form a delivery arc. The k-link algorithm ensures the MLC to travel the shortest distance between the adjacent planning beams so that the delivery is as smooth as possible. The MLC trajectory of each delivery arc is evaluated to verify if the predefined MLC displacement constraint is complied. The apertures will be adjusted if the MLC constraint is violated but the adjustment will be minimal to sustain plan quality.

3.2.2 Aperture-based optimisation

The 2-step process in intensity-map based optimisation might not be an efficient planning strategy. The translation of a deliverable MLC sequence from the ideal intensity maps may result in unwanted loss of plan quality. Direct aperture optimisation (DAO) uses an 1-step aperture-based optimisation approach where the MLC positions and aperture weights are directly optimised to a dose distribution [Shepard *et al* 2002][Earl *et al* 2003]. Since the MLC aperture shapes and weights are simultaneously optimised, the MLC sequenc-

ing step is eliminated and the planning process becomes more efficient. At the start of a DAO planning session, there is an arbitrary set of MLC apertures assigned to each planning beam angle. The number of MLC apertures at each beam is the number of arcs that is specified by the planner. To initialise the optimisation, the aperture shapes of the MLC segments conform to the target at BEV. An initial dose distribution is calculated based on these arbitrary MLC apertures. The adjustment of MLC positions or weightings during optimisation is reflected in this dose distribution which is evaluated with the objective functions that are defined by the planner. Once a solution is found, DAO outputs a deliverable IMAT MLC sequence without the need of MLC sequencing as in intensity map-based optimisation.

3.3 Dose calculation

To simulate and evaluate the dose distribution to be delivered, a dose calculation is performed for each individual treatment plan for every patient. There are two general categories of dose calculation methods: correction based methods and model based methods [Van Dyk 1999]. Correction based methods are the conventional dose calculation approach where modifications or corrections are applied to depth dose data that was measured in a uniform homogeneous water phantom with a beam incident normal to the flat surface. Corrections are applied to account for non-perpendicular or oblique beam incidence, patient contour irregularity, and patient physical inhomogeneity. Currently, model based dose calculation methods are widely used in the most recent planning systems. Instead of correcting a set of measured data, model based methods model each of the individual beams and propagate through their

paths, taking into the account of the geometric and physical properties of the patient. In general, the dose at a point in the patient can be calculated by the following generalised equation [Papanikolaou *et al* 2004]:

$$D(r) = \int\int\int T(r') \cdot K(r; r') d^3 r' \tag{3.5}$$

Such dose calculation approach is known as the convolution/superposition method (CVSP) [Mackie *et al* 1985][Boyer & Mok 1985][Mohan *et al* 1986] [Ahnesjö *et al* 1987][Mackie *et al* 1987][Murray *et al* 1989][Hoban *et al* 1994]. The dose at a point r in the 3D volume, $D(r)$, is the summation effect of the primary photon energy fluence, $T(r')$, convolving with a dose spread kernel $K(r; r')$ at each point in the irradiated volume (r'). $T(r')$ is the total energy released per unit mass by the primary radiation (i.e. the radiation source) at r', also known as TERMA. The pattern of energy deposition at r (i.e. dose) by the secondary particles released from the primary interaction site r' is described with the dose kernel $K(r; r')$. Typically, TERMA is modeled by the TPS based on some machine-specific measured data, e.g. energy spectrum, beam symmetry and profile, collimator properties such as the transmission factor, and the density information of the patient; while dose kernels or scatter kernels are generated by Monte Carlo (MC) simulations with a homogeneous water phantom [Mackie *et al* 1988].

Note that it is computationally intensive to calculate the integral in equation 3.5 since every point in the volume r' contributes differently to the dose point at r. To reduce the CPU burden, Ahnesjö had investigated the practical implementation of CVSP using the collapsed cone convolution scheme (CCC) [Ahnesjö 1989]. In CCC, the 3D point dose kernels are discretised into a set

of cones which are propagated from the interaction site. Instead of calculating the entire conical kernel element, the cone is collapsed into a line along the cone axis where energy is deposited. Commercial TPS such as Philips Pinnacle^{TM} (Philips Medical, Madison, WI, USA) adopted this calculation scheme into the treatment planning routine.

In cases where the kernel is spatially invariant (relative to the lateral directions of the central axis of the beam), i.e. in a homogeneous medium, the kernel function becomes:

$$K(r; r') = K(r - r') \tag{3.6}$$

Thus, the superposition integral of kernel shrinks into a convolution integral as the dose is now dependent on the relative position of the dose point and the scattering point only, i.e. depth. The dose distribution can then be calculated by merely convolving the energy fluence and the scatter kernel:

$$D(x, y, z) = \int\int T(x', y') \cdot K(x - x', y - y', z) dx'y' \tag{3.7}$$

The kernel function characterises the energy deposition by a small finite sized beam, i.e. a pencil beam. Dose calculation method based on equation 3.7 is often called pencil beam convolution (PBC) and is adopted by the Varian Eclipse^{TM} TPS (Varian Medical Systems, Palo Alto, CA, USA) [Mohan *et al* 1986][Mohan & Chui 1987][Ahnesjö *et al* 1992].

CCC, PBC, or other CVSP dose calculation techniques are based on analytical methods. Various assumptions and approximations are made while adapting these methods for dose computation. The MC method on the other hand, provides the highest accuracy in dose calculation to date

[Roger & Bielajew 1984][Rogers *et al* 1995][Rogers 1991] [Bielajew *et al* 1994] [Ma *et al* 2002]. In mathematics, MC is a stochastic sampling technique that has been used for different operation purposes such as complex physics problems and simulations. For medical physics applications, MC is most commonly used in dose calculation. With the MC method, a large number of particles can be simulated efficiently. By using the ray-tracing technique, each individual particle is simulated and tracked as it is generated in the target and traverses into the calculation volume (a phantom or a patient). The individual interaction event of the particle is simulated and recorded along its path, e.g. interactions with the physical structures in the treatment head of the linac and interactions inside the patient. Because the energy and direction of the particle is known, the type of interaction with various materials can also be simulated (with known physical properties of the interacting material). The transport of the secondary charged particles from the interactions of the source particles is simulated. Energy deposition of these secondary charged particles is then stored as dose. Currently, there are several MC dose calculation code packages available for research purposes and some commercial TPS has incorporated MC calculation for electron beam [Kawrakow *et al* 1996][Baró *et al* 2003][Briesmeister 2000] [Kawrakow *et al* 2004]. However, the simulation time of some MC codes for photon beam is too long for clinical routine as a long time is spent on simulating the secondary charged particle transport although various variance reduction schemes can be used to reduce the simulation time [Kawrakow & Fippel 2000].

Chapter 4

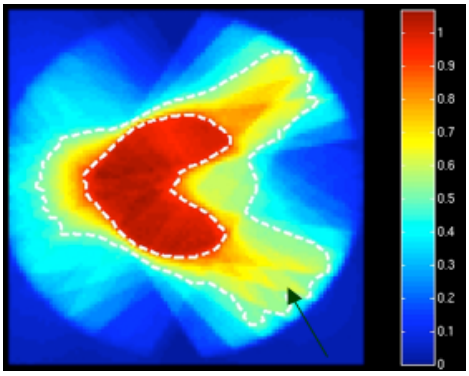
Improving the accuracy and efficiency of dose calculation for treatment planning of intensity-modulated arcs

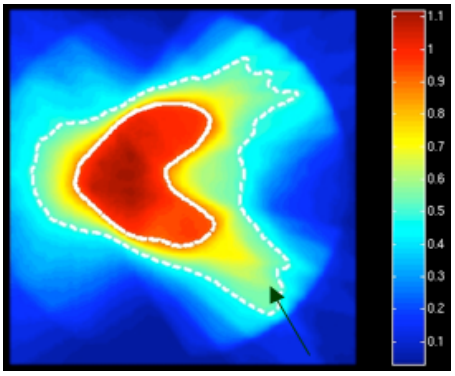
---

4.1 Background and objectives

For the purpose of treatment planning, the continuous delivery arcs of an IMAT plan are often approximated by a series of evenly-spaced static beam angles, which are used for both optimisation and dose calculation. As discussed in Chapter 3, a larger number of static beams may provide a better approximation resolution but CPU burden will also increase. From past efforts, it has been shown that static planning beams with an equi-angular spacing, $\Delta\theta$, of 10° is sufficient for optimising an acceptable treatment plan in clinical feasible times. Nonetheless, this assumption may not be applicable for dose calculation. During the continuous arc delivery, the planned MLC apertures between the adjacent planning angles are linearly interpolated by the MLC controller. Dose distribution that is computed based on these static beams neglects this fact and results in dosimetric errors between treatment

planning and actual delivery. Except from the minor differences in "finger artefacts" in the lower dose regions in static-beam calculation (see figure 4.1), such dosimetric effects may not be significant in CAT as the aperture shapes at the planning beam angles are merely the 2D projections of the target at BEV. However, with intensity modulations in IMAT where the aperture shapes are optimised for each of the static beams, drastic aperture shape variation between the adjacent beams may increase the error in static-beam planning and continuous delivery [Tang *et al* 2008]. Furthermore, the dosimetric differences may exaggerate if the MU weighting of the apertures are allowed to vary between the neighbouring angles.

(a) 

(b) 

Courtesy of M. Earl

Figure 4.1: (a) Finger artefacts on a static-beam calculation for an IMAT plan overestimating the lower dose volume (c.f. yellow region) and (b) are smoothed out in the continuous arc delivery.

Conspicuously, for arc plans with both aperture shape and MU weighting variations, such as IMAT, should be calculated with a large number of interpolated beams in order to minimise any potential dosimetric differences between static-beam planning and continuous arc delivery. Unfortunately, there are no comprehensive TPSs dedicated to and available for IMAT treatment planning at the time of this project while the traditional practice imports the opti-

mised IMAT MLC sequence to a commercial TPS for final dose calculation. Most of the current TPSs use analytical dose calculation algorithms, e.g. the CVSP methods, and the CPU time is dependent on the number of beams. In contrast to the analytical algorithms, MC methods do not suffer from such time-scaling aspect. For a given geometry, the rate of statistical convergence strongly depends on the number of histories used in simulation, not on the number of beams or MLC segments [Fippel *et al* 1999]. Nonetheless, the typical simulation time for MC dose calculation may be too time-consuming for clinical implementation and often requires intensive CPU effort, e.g. clusters of simulation computers. As a solution, an in-house kernel-based MC dose engine was developed in the Department of Radiation Oncology in the University of Maryland to reduce the MC simulation time by replacing the model of secondary particle transport with a point dose kernel [Naqvi *et al* 2003]. In this chapter, dose calculation for IMAT using this existing MC kernel-superposition method was compared with the deterministic CVSP method used in a commercial TPS. The CPU efficiency of both calculation methods were evaluated. The dosimetric effects of dose calculation using the coarsely-spaced planning beams and the finely-spaced interpolated beams were assessed with 10 clinical cases.

4.2 Monte Carlo kernel-based convolution/superposition (MCKS)

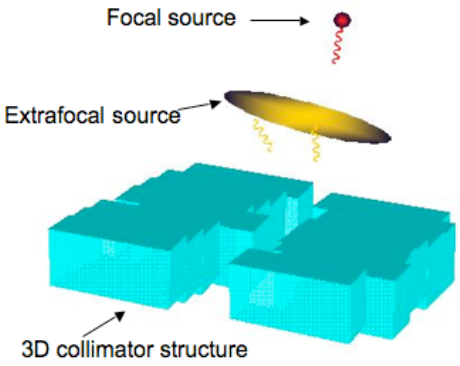
Developed in the Department of Radiation Oncology in the University of Maryland and is currently used clinically as a part of the patient-specific QA procedure, the Monte Carlo kernel-based convolution/superposition (MCKS)

algorithm[1] is different from the traditional full-blown MC dose calculation methods. As described in the previous chapter, conventional MC techniques simulate the individual particle from the instance it is generated at the target and each particle is tracked from the source to the interaction paths in the phantom or patient. With the correct calculation parameters, this detail ray-tracing technique would certainly provide the most accurate dose computation. For such CPU intensive task, a cluster of computers is required. However, even with this large number of computers, the typical simulation may be too long for clinical practices. MCKS reduces the total simulation time by replacing the time-consuming procedure of secondary electron transport with a point dose kernel. Although the details in dose deposition may be slightly compromised, MCKS still performs a full modelling on the primary photon transport in place of a calculation of TERMA that is used by the deterministic polyenergetic CVSP methods. As a result, the simulation time is reduced while maintaining the details in source (i.e. photon) and primary transport modelling. For inhomogeneous phantom calculation, the density scaling method is used for homogeneity correction resulting in a similar accuracy to that found in the CVSP methods [Woo & Cunningham 1990].

A typical MCKS simulation of a photon treatment plan starts by randomly sampling the beam angles and field segments with the probabilities based on the proportional MU weightings, i.e. the higher MU weightings the beam and segment possess, the more likely they will be modelled. A photon is randomly sampled from either the dual-source model or the phase-space data and propagated through the collimator structure, taking into the account of the physical properties such as the MLC/jaws transmission, tongue-and-

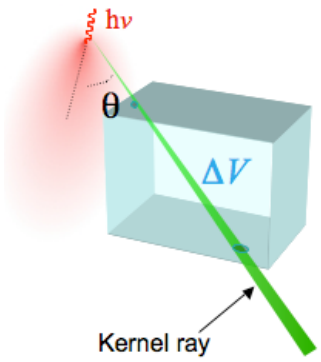
[1]Used with permission from the original developers of MCKS

groove, and curved-end geometry of the MLC leaves (see figure 4.2). The current MCKS codes, however, does not model MLC scatter as a full MC simulation would. After traversing the collimator structures, the photon then interacts somewhere in the simulation phantom (or patient) with a probability based on the energy of the photon and the mass density profile along the photon direction. From the site of interaction, a point dose kernel is issued at a randomly sampled direction. Unlike the deterministic dose engines such as CCC in Pinnacle[3] that use a fix set of kernel ray directions, MCKS randomly samples a ray along the kernel and the energy deposited in the voxels along this kernel ray is stored as dose as illustrated in figure 4.3. The total simulation time of an IMAT treatment plan for a HN case at a voxel size of $2 \times 2 \times 3$ mm^3 is approximately 20 minutes, which is clinically feasible.



Courtesy of S. Naqvi

Figure 4.2: Source modelling of MCKS with a dual-source model.



Courtesy of S. Naqvi

Figure 4.3: A kernel ray propagated from the interaction site and stored the energy deposited in the voxel ΔV along the ray as dose.

4.3 Rediscretising the continuous delivery arc for dose calculation

The ideal method to calculate the dose for an arc treatment plan is to continuously sample the beam angles of the continuous delivery arc. However, it requires a significant change to the original codes of MCKS. Based on a previous study, minimal differences were found using the full continuous sampling calculation method for IMAT [Olteanu *et al* 2006]. An immediate dose calculation method for intensity-modulated arcs is to compute the dose distributions using a large number of interpolated static beams: IMAT plans that were optimised with 36 static beams with $\Delta\theta$ of 10° were interpolated into a series of finely-spaced beams which closely simulate the delivery arc, e.g. with $\Delta\theta$ of 0.5°. The MLC aperture shapes and MU weightings were linearly interpolated between the adjacent planning angles as in the dynamic delivery by a program written in C (see appendix A). Because all the interpolated

beams and MLC segments will also be modelled in the MCKS simulation, this interpolated-beam calculation may result in a more accurate representation of the actual arc delivery. It is important to note that the individual MU weighting assigned to the static beam angles in the treatment plan are delivered in a "sector" in the actual delivery, i.e. the dose intended for a particular beam angle is spread over the angular interval between its successive adjacent beam angle during the continuous rotational delivery as illustrated in figure 4.4. Figure 4.4 (a) shows 3 of the 36 static beams of an IMAT plan starting from the gantry angle of 175° to 185° (anticlockwise) and the optimised MLC sequence that consists of the aperture shapes and weightings is based on the beam angles of [175°, 165°, 155°, ..., 185°]. For an unbiased delivery, a "kick-off" beam is added and placed at half of the $\Delta\theta$ prior the first beam of the optimised sequence and an "ending beam" is added at half of the $\Delta\theta$ beyond the last beam as seen in figure 4.4 (b), i.e. during delivery, the first beam starts from 180° and ends at 180°, with a total arc range of 360°. The planned MU at each static beam is delivered in a uniform delivery sector that spreads from the individual planning angles as shown in table 4.1.

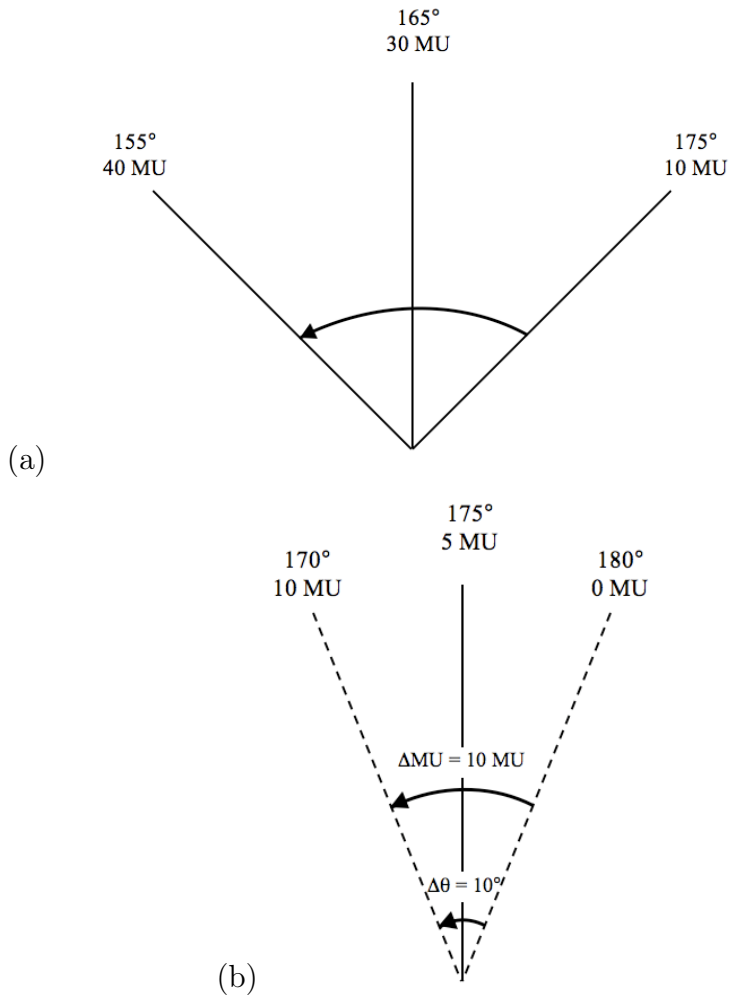


Figure 4.4: (a) An anticlockwise IMAT plan with static beams spaced at every 10° (same IMAT plan example as table 4.1). The MU value dictated at each beam angle is the cumulative MU of the plan. (b) During delivery, the MU is delivered over a "sector" with an angular interval of 10°. A "kick-off" beam is added to the starting beam angle of the plan, i.e., at 175°, in order to enable the MU to be delivered in such "sector".

Table 4.1: Transformation of static beam planning to continuous arc delivery. The MU are delivered over a set of sectors during the arc delivery.

Planning angle	Cumulative MU	Absolute MU	Delivery sector	Cumulative MU	Absolute MU
175°	10	10	180° → 170°	10	10
165°	30	20	170° → 160°	30	20
155°	40	10	160° → 150°	40	10
⋮	⋮	⋮	⋮	⋮	⋮
195°	795	15	200° → 190°	795	15
185°	800	5	190° → 180°	800	5

For an accurate dose calculation, the continuous delivery sequence is rediscretised during the interpolation process. For example, the IMAT sequence from table 4.1 planned with $\Delta\theta = 10°$ was interpolated with an angular separation of 2° as shown in figure 4.5 (a). The interpolation procedure starts from rediscretising the continuous delivery sequence, i.e. the delivery sectors of [180°-170°, 170°-160°, ..., 190°-180°] were rediscretised into a series of static beams with the interpolation spacings of [179°, 177°, 175°, ..., 185°, 183°, 181°], as demonstrated in table 4.2 and figure 4.5. During the rediscretisation, the MU that was intended to be delivered over the angular interval between the adjacent (interpolated) beams, e.g. 180° to 178°, was collapsed into one static beam angle at the mid-interval, i.e. 179°, as seen in figure 4.5 (b). While the aperture shapes were linearly interpolated for these angles, MU weightings that were spread within the individual delivery sectors were re-grouped into a series of absolute values that were then assigned to the interpolated static beams. These interpolated beams were subsequently used in the interpolated-static beam calculation, which intends to minimise the dosimetric differences in static-beam planning and continuous arc delivery.

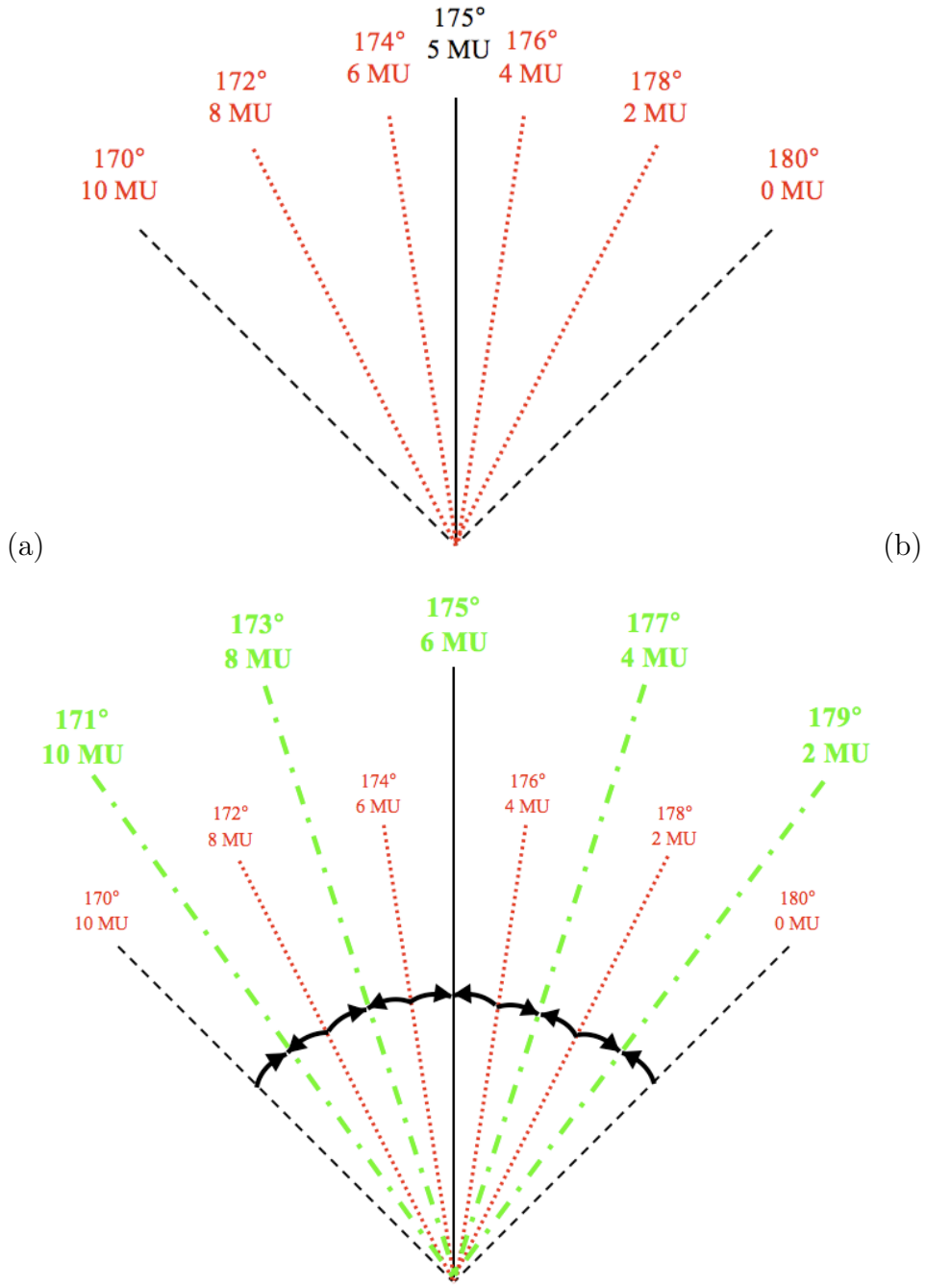


Figure 4.5: (a) Potential interpolated beams (shown in red) of an IMAT plan with an interpolation spacing of 2° and (b) rediscretised for interpolated static-beam dose calculation (final interpolated beams shown in green).

Table 4.2: Example of the rediscretisation procedure. The sequence in table 4.1 is rediscretised with 2° separation for interpolated dose calculation.

Delivery	Cumulative MU	Absolute MU	Rediscretisation	Cumulative MU	Absolute MU
180°					
			179°	2	2
			177°	4	2
↓	10	10	175°	6	2
			173°	8	2
			171°	10	2
170°					
⋮	⋮	⋮	⋮	⋮	⋮
190°					
			189°	796	1
			187°	797	1
↓	800	5	185°	798	1
			183°	799	1
			181°	800	1
180°					

4.4 Statistical comparison of static-beam calculation and interpolated-static beam calculation

Although detailed QA tests has been performed for MCKS, it is necessary to compare it with a well-established commercial TPS for benchmarking the dose calculation accuracy and CPU speed. A 36-beam IMAT plan of a brain case was randomly selected for the comparison between MCKS and Pinnacle^3. Plans in Pinnacle^3 were computed on a Sun Solaris 10 platform that uses the AMD Opteron CPU running at 2.8 GHz, while all MCKS simulations were performed in a MacBook® using the Mac OS X System running at 2.0 GHz.

Since the computer systems used were different for Pinnacle^3 and MCKS, the same plan was computed with MCKS in the Sun Solaris 10 platform to observe any CPU performance differences that may bias the CPU speed test. It was found that the CPU speed of MacBook® was approximately 10% faster than the Sun Solaris platform. For convenience and minimum disturbance to the clinic, all MCKS simulations were performed in the MacBook® with the CPU times adjusted down by 10% to ensure a fair comparison with Pinnacle^3.

4.4.1 Determine the simulation time of MCKS

In general, the endpoint of MC simulations is where 2% statistics is reached. When the statistical uncertainty or standard deviation in dose is within 2% or less, the overall dose distribution does not change significantly and affect the various physical and biological dose indices [Keall *et al* 2000]. The standard deviation, σ_t, with respect to CPU time, t, is defined as:

$$\sigma_t = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(D_{i,t} - D_{i,t\to\infty}\right)^2}, \tag{4.1}$$

where N represents the number of voxels within a certain range of dose, D. To determine the adequate simulation time of MCKS, the time taken to reach near "perfect statistics", $t \to \infty$, was assumed to be 500 minutes. During simulations, a dose file was written and outputted every minute to compare with the "perfect statistics". In figure 4.6, it was shown that MCKS takes approximately 25 minutes to reach 2% statistics in both the high dose region (i.e. 90% of the prescription dose) and low dose region (i.e. 25 - 50% of the prescription dose) in a brain case. The plan was computed again in MCKS for 25 minutes and the dose distribution was compared to that calcu-

lated in Pinnacle^3. It was found that MCKS matched with Pinnacle^3 at 1.4% in the PTV (high dose region), and 1.8% in the normal tissue dose (low dose region) with the presence of the 2% statistical fluctuation, which was in agreement to the extensive benchmarking study previously performed for MCKS [Naqvi *et al* 2003]. After establishing the minimum simulation time of MCKS and verifying its accuracy with Pinnacle^3, the IMAT plan was recalculated in MCKS again using 720 finely interpolated beams. The standard deviation of the dose per voxel was plotted against the CPU time in figure 4.6, showing that the 720-beam calculation (interpolated-static beam calculation) matches the original 36-beam calculation (static-beam calculation), indicating that the CPU time of a MC-based or stochastic-based dose engine is independent of the number of beams involved.

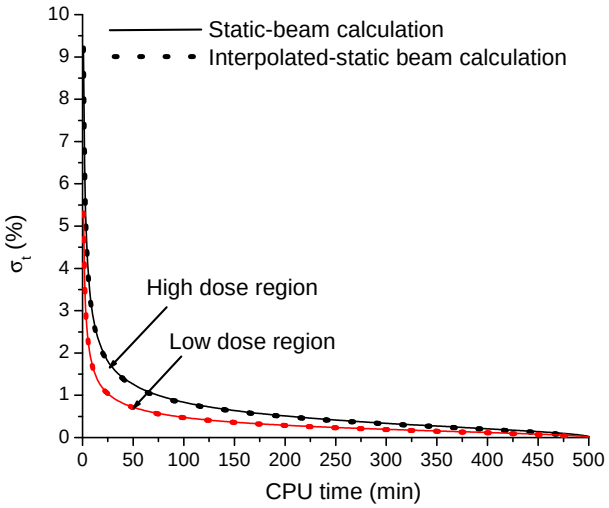


Figure 4.6: Standard deviation (σ_t) of the dose per voxel *vs.* CPU time in the high dose region, i.e. where dose is greater than 90% of the prescription dose; and the standard deviation of the dose per voxel in the low dose region, i.e. where the dose is 25%−50% of the prescription dose. The curves follow Poisson statistics and also illustrate the virtual independency of CPU time on angular spacing.

4.4.2 Influence of voxel size in dose computational time

All MCKS simulations were computed using a voxel size of $2 \times 2 \times 3$ mm^3. Typically, IMRT plans are calculated at a voxel size of $4 \times 4 \times 4$ mm^3 although the use of a finer voxel size of $2 \times 2 \times 3$ mm^3 is desired for a more accurate dose computation. To accommodate different practices, the influence of voxel size in CPU time of both Pinnacle^3 and MCKS was studied using the same brain case. Table 4.3 displays the comparison of CPU times in Pinnacle^3 and MCKS with different voxel sizes, quantified with a time ratio $T(v)$. With larger voxel sizes, Pinnacle^3 is more efficient for dose calculation but MCKS becomes more superior in CPU efficiency for dose calculations using finer voxel sizes.

Table 4.3: Efficiency *vs.* voxel size in MCKS and Pinnacle^3. A 36-field static-beam plan was calculated in Pinnacle^3 and MCKS with different voxel sizes. The endpoint of MCKS simulation is $\sigma_t = 2\%$ for $D > 0.9D_0$, where D_0 is the prescription dose. The ratios $T(v)$ are determined with the CPU times of MCKS and CCC in Pinnacle^3, $T(v) = CPU\_time_{MCKS}/CPU\_time_{Pinnacle}$.

Voxel size (mm^3)	CPU time (min) Pinnacle^3	CPU time (min) MCKS	$T(v)$
$1 \times 1 \times 3$	109	63	0.58
$2 \times 2 \times 3$	20	24	1.20
$3 \times 3 \times 3$	7	12	1.71
$4 \times 4 \times 3$	4	8	2.00
$5 \times 5 \times 3$	2	7	3.50

4.4.3 Dependence of the number of beams in dose calculation

An interpolated-static beam calculation for an IMAT plan may require up to 720 beams. To evaluate the dose calculation efficiency, the CPU time de-

pendency on the number of beams was investigated for both Pinnacle^3 and MCKS. Since there is a limit of the number of beams allowed in Pinnacle^3 and it is not possible to calculate a plan with 720 beams, a 36-beam plan was calculated and the average time taken to calculate per beam $\bar{t}$ was derived. Assuming the interpolated beams have similar field shapes and sizes, the calculation time for each beam was approximated to be constant and the time required to compute n beams in Pinnacle^3 was therefore $n \cdot \bar{t}$. In figure 4.7, it was shown that when there are more than 43 beams in a treatment plan, MCKS is a more efficient dose calculation method, unlike the proportional increase in CPU time to the increasing number of beams found in the analytical dose calculation algorithm in Pinnacle^3. Although this may not be particularly useful for multiple static-field IMRT plan that typically uses only 7 - 9 beams, rotational arc plans consist of larger number of beams would benefit from this efficiency. For example, in the recent single-arc IMAT techniques such as Varian RapidArc^{TM} (Varian Medical Systems, Palo Alto, CA, USA), there are approximately 177 beams involved per arc in a plan. The plan is calculated in the TPS with an analytical dose calculation engine using 4 computer processors, resulting in a CPU time of 10 to 20 minutes per arc, depending on the field size. With the same number of computer processors, MCKS can perform an even faster dose calculation and gain in accuracy at the same time.

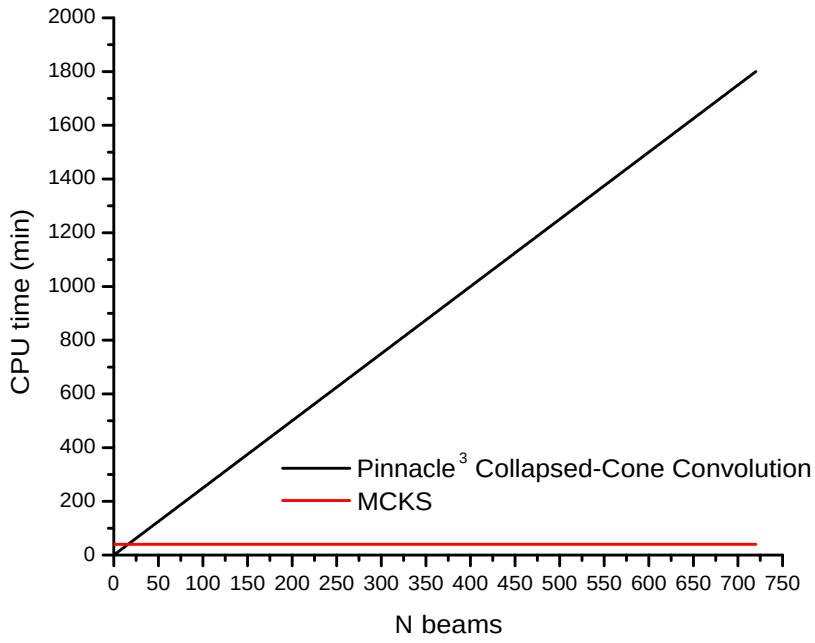


Figure 4.7: Required CPU time as a function of the number of beams using MCKS and collapsed cone convolution in Pinnacle^3 with a $2 \times 2 \times 3$ mm^3 dose grid; the two lines intersect at 43 beams. ($\sigma_t = 0.02$ for high dose region is used as the criteria for MCKS)

4.5 Plan quality comparison of static-beam calculation and interpolated-static beam calculation

For a selection of 10 patient cases (1 lung, 2 brain, 3 prostate, and 4 HN cases), the dose differences between static-beam calculation and interpolated-static beam calculation were investigated. For each patient, the dose distribution calculated using the 36 planning beams was compared with that calculated using 720 interpolated beams.

4.5.1 IMAT planning

All IMAT treatment plans were generated using the 2-step intensity map-based planning approach. The arc range of all the plans was 360° and 36 equi-spaced static beams were used for optimisation. A set of ideal intensity maps were optimised using these static gantry angles in the Pinnacle^3 TPS utilising the P^{3}IMRT module. These 36 sets of ideal fluence maps were then exported and transferred to CIMO[2] for MLC sequencing with a MLC displacement constraint applied. The maximum distance allowed for MLC to travel between the adjacent planning beam angles was 5 cm, based on the maximum speed of MLC of 3 cm/s and the maximum mechanical rotational speed of the gantry of 6°/s (see equation 3.1). The resultant MLC sequence that consists of the aperture shapes and MU weightings were transferred to the beam-interpolation program before a final dose calculation was performed in MCKS. Based on the observation that MCKS simulation time is independent of the number of beams, a fine interpolation spacing of 0.5° was used. For each patient, two sets of calculation were conducted based on: (1) 36 planning beam angles and (2) 720 interpolated beam angles. The two sets of calculation were subsequently called the 36-beam calculation and 720-beam calculation to represent the static-beam optimisation and simulate the continuous delivery, respectively. All doses were computed using a dose grid of 2 × 2 × 3 mm^3 and the endpoint of the simulations was where 2% statistics was achieved. The computed dose distributions were imported back to Pinnacle^3 for plan evaluation. Because the output dose files of MCKS were in ASCII format, a conversion of these files into *.img files was necessary in order to enable the import into the TPS (see appendix B).

[2]Used with permission from the original developers of CIMO.

4.5.2 Dose differences between 36-beam calculation and 720-beam calculation

All patient plans were evaluated based on DVHs. A summary of the comparisons between the 36-beam calculation and the 720-beam calculation is tabulated in table 4.4. The plans were evaluated and compared using several clinically used plan quality indices such as D_{95}, which represents the dose received by at least 95% of the ROI and is usually served as an indicator of target coverage. For normal tissue dose, V_x is used to reflect the volume of ROI receiving x Gy of dose. The dose differences are defined by,

$$\frac{(D_{720-beam} - D_{36-beam})}{D_{36-beam}} \times 100\%. \tag{4.2}$$

From the DVH comparisons, most discrepancies shown were negative differences, indicating that the dose computed in the static-beam treatment plans (represented by the 36-beam calculation) was overestimated compared to that in the continuous delivery simulation (represented by the 720-beam calculation). Large differences up to 17% were observed in the high dose region, i.e. in the targets (GTV, CTV, and PTV). Such differences were particularly significant in complex HN cases. Except from the reduced dose computed to the targets, the dose to normal tissues were significantly lower in the 720-beam calculations, up to almost 100% of differences were seen, e.g. the $V_{71.4}$ of rectum in all three prostate cases was at least 95% different between the two sets of calculation. The large discrepancies in the low dose regions between the calculations can be explained by the smoothing of the finger artefacts in the interpolated-static beam calculation, as shown in figure 4.8.

Although the dose to critical organs were overestimated in planning which

is beneficial for normal tissue sparing, the target coverage was largely degraded. These large discrepancies between static-beam optimisation and interpolated-beam final dose calculation may degrade the treatment quality to patients, which may affect the clinical outcomes. To account for such discrepancies, a similar interpolated static-beam calculation approach can be adopted during optimisation. At a certain iteration of the optimisation process, a dose calculation using a series of finely interpolated beams can be performed to simulate the final dose calculation. The objective function will then be updated and renew the optimisation such that the algorithm optimises on the "actual" dose distribution. Although the optimisation time may be lengthened by this procedure, the potential differences between static-beam planning and continuous arc delivery can be minimised. Future work plans may involve repeating this study with a larger number of patient cases of the same disease site in order to yield a more quantitative/statistically-robust conclusion of such dose differences and the subsequent effects on treatment quality and patient outcome.

Table 4.4: Dosimetric differences in static-beam calculation and interpolated-static beam calculation for all 10 patient case studies. The interpolated-static beam plans were normalised at the mean dose of the PTV to the corresponding static-beam plans. Dose differences are defined as $[(D_{720-beam}\text{-}D_{36-beam})/D_{36-beam}] \times 100$ (%).

Case	**ROI**	**Dose metric**	**Dose differences (%)**
Lung	GTV	D_{95}	-1.8

Continued on next page

Table 4.4 – continued from previous page

Case	ROI	Dose metric	Dose differences (%)
	PTV	D_{95}	-10.1
	Left Lung	V_{20}	-10.7
	Right Lung	V_{20}	-9.9
	Heart	Max	-11.8
	Spinal Cord	Max	0.6
Brain 1	GTV	D_{95}	-1.0
	PTV	D_{95}	-7.2
	Left Eye	Max	-6.6
	Right Eye	Max	-6.8
	Left Optic Nerve	Max	-9.4
	Brainstem	Max	-1.5
	Spinal Cord	Max	-8.7
Brain 2	PTV	D_{95}	-3.9
	Left Eye	Max	-6.5
	Right Eye	Max	-2.0
	Left Optic Nerve	Max	-0.5
	Right Optic Nerve	Max	-3.6
	Brainstem	Max	-7.2
Prostate 1	GTV	D_{95}	-0.9
	PTV	D_{95}	-6.4
	Bladder	$V_{66.7}$	-27.4
	Rectum	$V_{71.4}$	-94.2
Prostate 2	GTV	D_{95}	-0.5

Continued on next page

Table 4.4 – continued from previous page

Case	ROI	Dose metric	Dose differences (%)
	PTV	D_{95}	-6.5
	Bladder	$V_{66.7}$	-32.2
	Rectum	$V_{71.4}$	-99.2
Prostate 3	GTV	D_{95}	-1.2
	PTV	D_{95}	-7.4
	Bladder	$V_{66.7}$	-22.2
	Rectum	$V_{71.4}$	-99.9
HN 1	CTV 1	D_{95}	-4.0
	CTV 2	D_{95}	-9.1
	CTV 3	D_{95}	-15.0
	Left Parotid	Mean	-16.5
	Right Parotid	Mean	-10.7
	Spinal Cord	Max	3.1
HN 2	GTV	D_{95}	0.7
	PTV	D_{95}	-6.4
	Optic Chiasm	Max	-13.8
	Left Parotid	Mean	-10.8
	Right Parotid	Mean	-9.3
	Right Optic Nerve	Max	-13.5
	Brainstem	Max	-7.5
	Spinal Cord	Max	-13.2
HN 3	GTV	D_{95}	-4.4
	PTV	D_{95}	-16.9

Continued on next page

Table 4.4 – continued from previous page

Case	ROI	Dose metric	Dose differences (%)
	Left Parotid	Mean	-19.1
	Right Parotid	Mean	-18.2
	Optic Chiasm	Max	-25.5
HN 4	GTV	D_{95}	-1.1
	PTV	D_{95}	-6.7
	Vocal Cord	Max	-5.0
	Spinal Cord	Max	-3.7

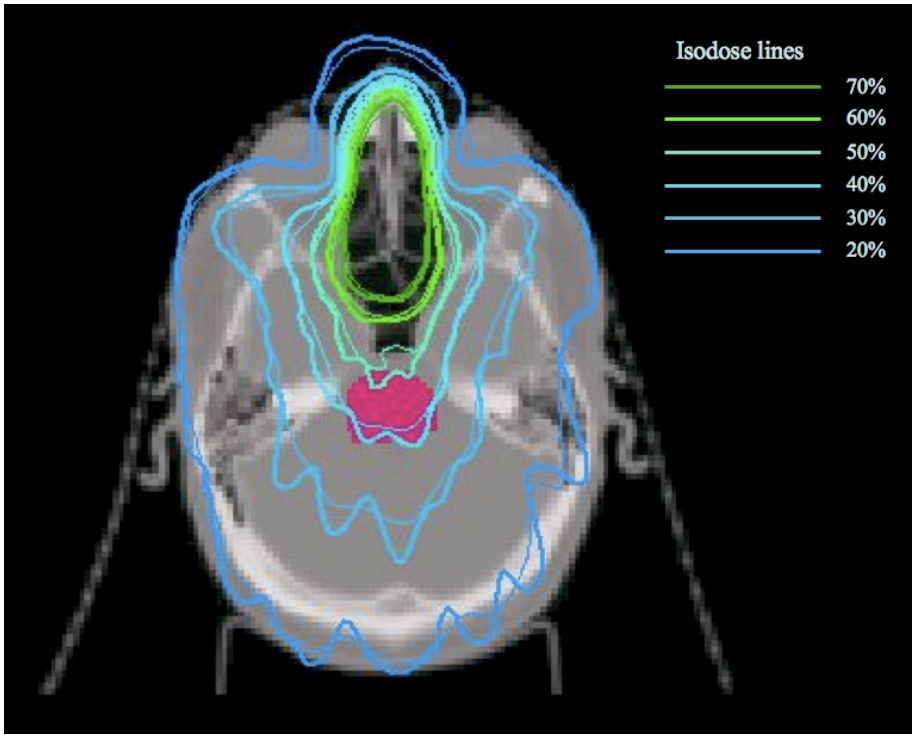


Figure 4.8: A MCKS-calculated isodose distribution comparison of a brain case illustrating that the finger artefacts in the low dose region of the static-beam calculation (thick lines) are smoothed out in the interpolated-static beam calculation (thin lines). Apparent dosimetric discrepancies at the brain-stem (pink region) and other lower dose regions are also observed.

4.5.3 Influence of aperture shape variation in dose calculation

From the plan comparisons, some large dose discrepancies were observed. All the plans were generated with a fairly relaxed MLC displacement constraint of 5 cm, i.e. the MLC leaves had the freedom to displace up to 5 cm between the adjacent planning angles within an angular interval of 10°. As a result, large aperture shape variations can occur between the neighbouring angles. Clearly, the differences between static-beam calculation and interpolated-static beam calculation were induced by the additional interpolated MLC apertures. However, the correlation between aperture shapes and dose calculation accuracy has yet been established. Note that the "dose calculation accuracy" in this context is described as the differences between static-beam optimisation and interpolated-beam final dose calculation of intensity-modulated arc plans, not the statistical accuracy of the computation of dose distributions. To investigate the effects of aperture shape variation on the accuracy of static-beam calculation, 2 HN cases were selected from the 10 patient cases where another IMAT plans were generated using a more stringent MLC constraint of 3 cm. Two sets of calculation were also performed for these new IMAT plans using the 36 planning beams and 720 interpolated beams.

Table 4.5 summarises the comparisons of the 36-beam and 720-beam calculations of the two sets of plans with 5 cm and 3 cm MLC constraints for the two HN cases. It was shown that with a larger aperture shape variation (i.e. the plan with 5 cm MLC constraint), larger differences could be seen between the static-beam optimisation and the interpolated-beam final dose calculation; whereas for the plan with smaller aperture shape variation (i.e.

the plan with 3 cm MLC constraint), the plan quality degradation in delivery was significantly less especially in the target doses. The reduced discrepancies found in the plan that was generated with a MLC constraint of 3 cm can be explained by the similarity of the interpolated apertures. The decrease in MLC constraint implicitly restricted the aperture shape variation as the MLC was not allowed to travel more than 3 cm per 10 degrees of angular range. The smaller the aperture shape variation it was between the neighbouring planning angles, the less different of the aperture shapes it was for the additional interpolated beams. As a result, the dose calculation based on these additional interpolated MLC apertures would be similar to the dose calculation that was based on the original static planning beams. On the other hand, the doses to normal tissues were still largely different between the static-beam calculation and interpolated-static beam calculation for both sets of plan. This may be an effect of the smoothing of the "finger artefacts" in the delivery, which was mentioned in the previous section.

Table 4.5: A comparison for HN cases 1 and 3 utilising 36 beams and 720 beams for dose calculation. The HN cases were planned with two different sets of MLC constraint (d_{max}) of 5 cm and 3 cm. Percentage dose differences are defined as $[(D_{720-beam}$ - $D_{36-beam})/D_{36-beam}] \times 100$ (%).

Case	ROI	Dose metric	Dose differences (%)	
			d_{max} = 5 cm	d_{max} = 3 cm
HN 1	CTV 1	D_{95}	-4.0	-1.5
	CTV 2	D_{95}	-9.1	0.0
	CTV 3	D_{95}	-15.0	-2.9
	Left Parotid	Mean	-16.5	-5.1
	Right Parotid	Mean	-10.7	0.1
	Spinal Cord	Max	3.1	-8.9
HN 3	CTV	D_{95}	-4.4	0.1
	PTV	D_{95}	-16.9	-2.7
	Left Parotid	Mean	-19.1	-10.6
	Right Parotid	Mean	-18.2	-9.8
	Optic Chiasm	Max	-25.5	-14.2

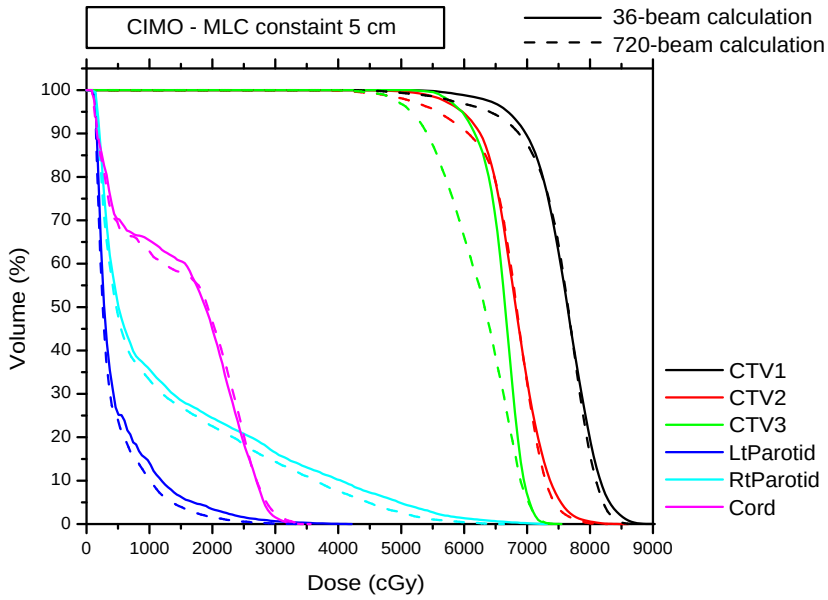
4.5.4 Influence of MU weighting variation in dose calculation

During optimisation, the algorithm optimises both the aperture shapes and the aperture MU weightings. Therefore, the MU weighting also varies between the adjacent planning beam angles. As it has been shown that aperture shape variation affects the accuracy of static-beam planning, the influence of MU weighting variation in dose calculation was also investigated. Because it is not possible to restrict the MU weightings in CIMO, the k-link leaf sequencing algorithm[3] was used as a cross-reference. Although the k-link IMAT leaf sequencer could neither restrict the MU weightings during sequencing, it was designed to maintain minimum MU weighting fluctuation between the neighbouring planning beams. For the same two HN cases used in section 4.5.3,

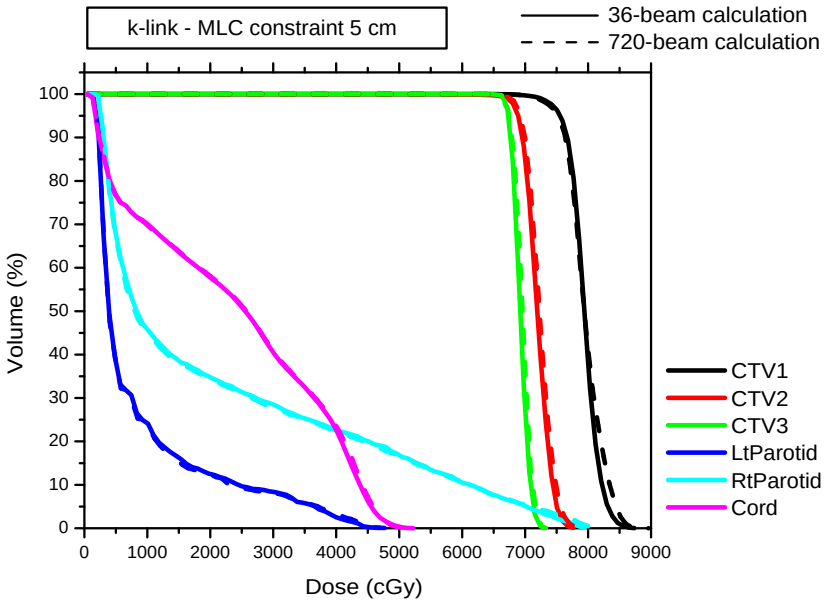
[3]Used with permission from the original developers of the k-link IMAT leaf sequencer.

4 sets of IMAT plans were generated using CIMO and the k-link IMAT sequencer with the MLC displacement constraints of 3 cm and 5 cm for each case. The resultant plans were calculated with the 36 static planning beams and the interpolated 720 beams in MCKS, the same procedure as in the previous sections.

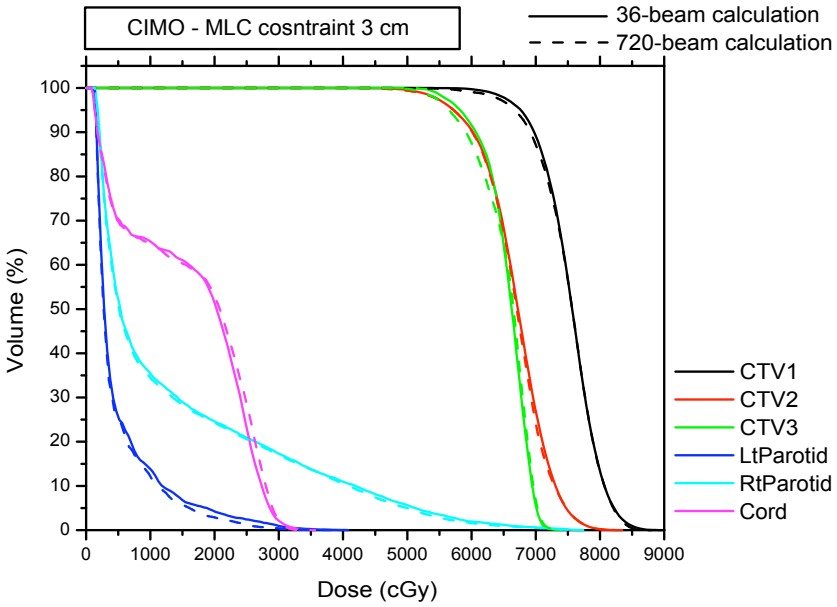
The DVH comparisons between the two sets of calculation for the 4 different sets of plan are shown in figures 4.9 and 4.10. All k-link plans showed minimal differences between the 36-beam calculation and 720-beam calculation. With two sets of different MLC constraints, the k-link sequencer produced plans with minimal aperture shape variation while obtaining a comparable plan quality compared to that in CIMO (see figure 4.11). On the other hand, large MU weighting variations of the apertures also affect the dose calculation accuracy as seen in the CIMO cases. Larger MU weighting variation adds additional differences in the properties of the original static planning beams and the interpolated beams used for final dose calculation. Hence, such larger MU fluctuation introduces extra discrepancies between static beam optimisation and interpolated-beam final dose calculation. Compared to the k-link sequencer plans, CIMO obtained plans with more fluctuating MU profiles within the arcs as shown in figure 4.12. The MU fluctuation was particularly rigorous in the CIMO plans, where the absolute MU differences between the planning intervals (i.e. 10°) were up to 27 MU in HN case 1 and up to 31 MU in HN case 3. These large differences of MU could create a "dynamic wedge effect" during delivery, which was not predicted by the static-beam calculation during optimisation.


CIMO - MLC constaint 5 cm
36-beam calculation
720-beam calculation
Volume (%)
Dose (cGy)
CTV1
CTV2
CTV3
LtParotid
RtParotid
Cord

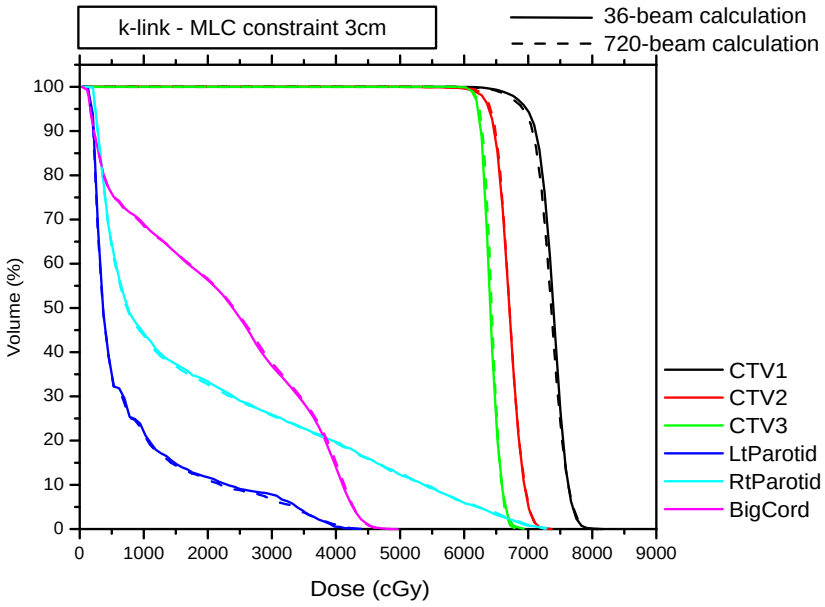
(a)


k-link - MLC constraint 5 cm
36-beam calculation
720-beam calculation
Volume (%)
Dose (cGy)
CTV1
CTV2
CTV3
LtParotid
RtParotid
Cord

(b)

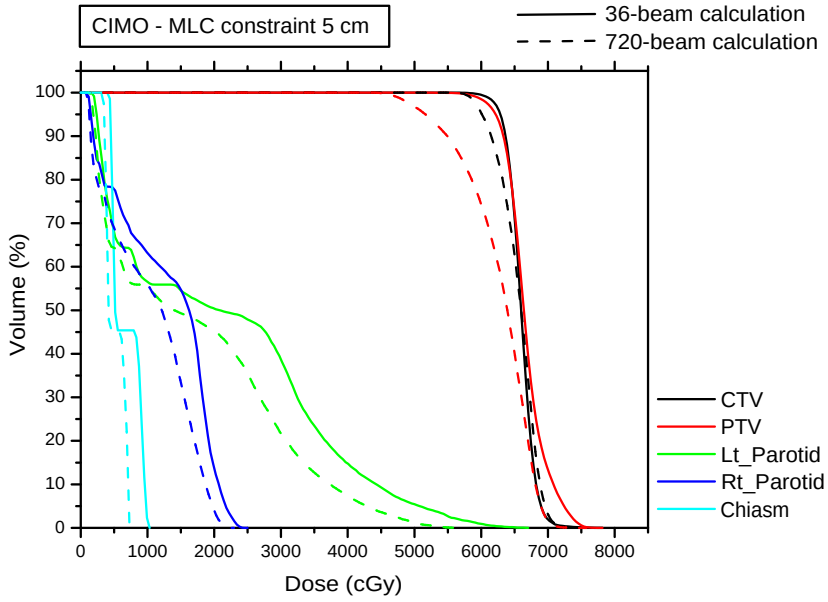


(c)

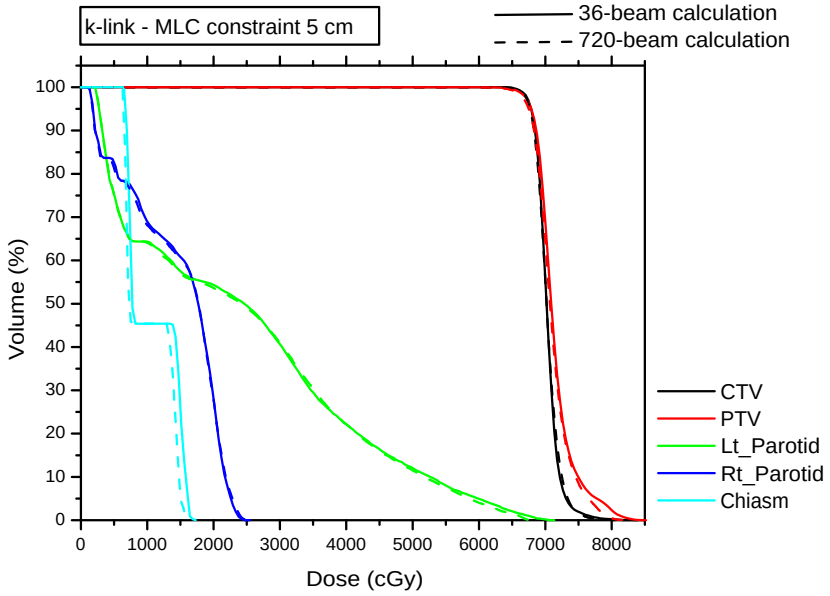


(d)

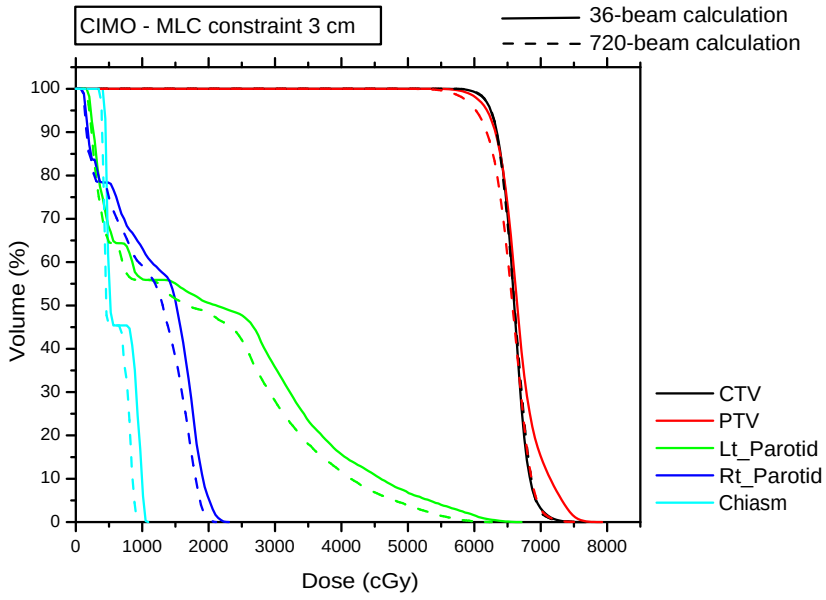
Figure 4.9: DVH comparisons of the 36-beam calculation and 720-beam calculation for the plans generated with a MLC constraint of 5 cm from (a) CIMO and (b) k-link leaf sequencer, and the plans generated with a MLC constraint of 3 cm from (c) CIMO and (d) k-link leaf sequencer for HN case 1.


CIMO - MLC constraint 5 cm
36-beam calculation
720-beam calculation
Volume (%)
Dose (cGy)
CTV
PTV
Lt_Parotid
Rt_Parotid
Chiasm

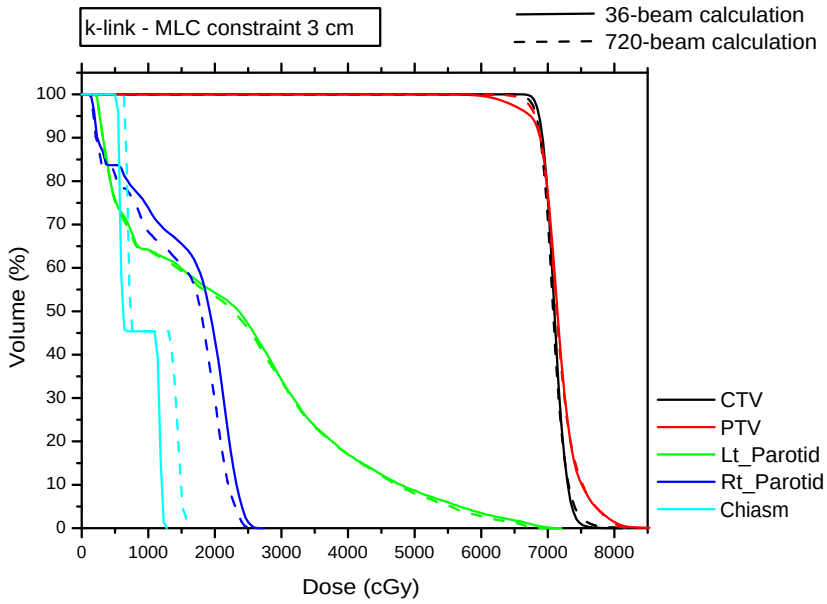
(a)


k-link - MLC constraint 5 cm
36-beam calculation
720-beam calculation
Volume (%)
Dose (cGy)
CTV
PTV
Lt_Parotid
Rt_Parotid
Chiasm

(b)



(c)



(d)

Figure 4.10: DVH comparisons of the 36-beam calculation and 720-beam calculation for the plans generated with a MLC constraint of 5 cm from (a) CIMO and (b) k-link leaf sequencer, and the plans generated with a MLC constraint of 3 cm from (c) CIMO and (d) k-link leaf sequencer for HN case 3.

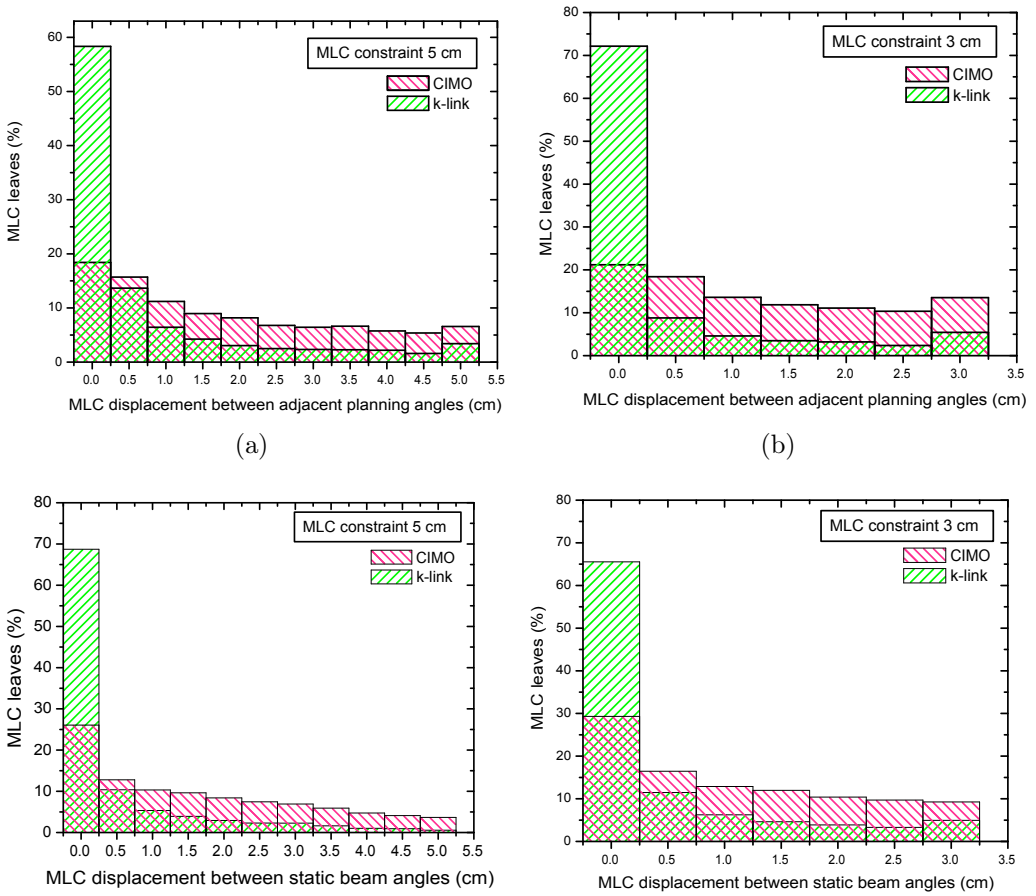


Figure 4.11: Comparison of the characteristics of CIMO and k-link IMAT sequencer. Distribution of MLC travel in the 10-degree intervals through the 360-degree arc for a plan with 5 cm MLC constraint is displayed in a) for HN case 1 and c) for HN case 3; and for a plan with 3 cm MLC constraint is displayed in b) for HN case 1 and d) for HN case 3.

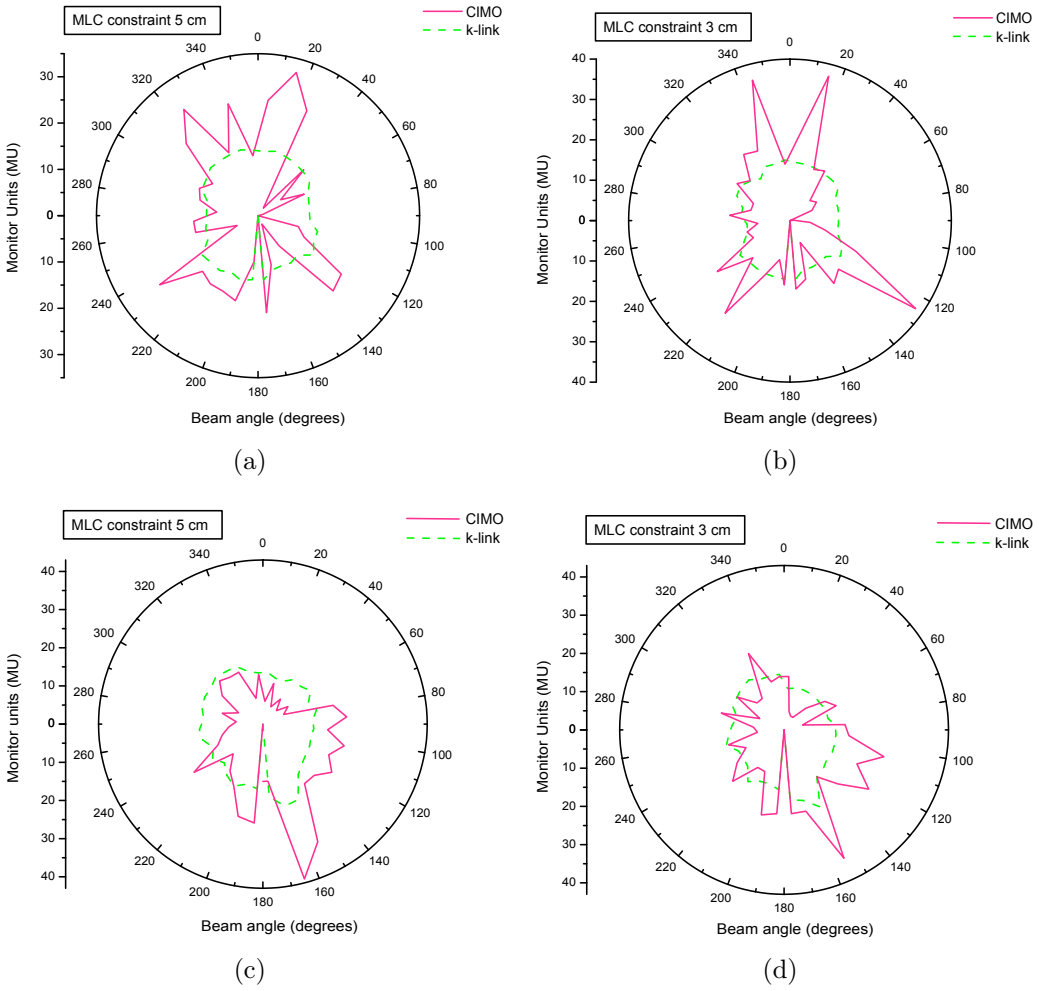


Figure 4.12: Comparison of the characteristics of CIMO and k-link IMAT sequencer. The absolute MU of the 10-degree spaced static planning beams is plotted on a circular scale which represents the beam angle of the actual delivery arc. The origin of the graph corresponds to the isocentre of the plan and the radius corresponds to the magnitude of the MU of the beams increasing from the origin. The CIMO plans and k-link plans are compared for a) HN case 1 with 5 cm MLC constraint, b) HN case 1 with 3 cm MLC constraint, c) HN case 3 with 5 cm MLC constraint, and d) HN case 3 with 3 cm MLC constraint.

4.5.5 Summary

The purpose of treatment planning is to simulate the actual radiation treatment delivery. In order to deliver a precise and accurate treatment to the patient, the simulation or planning accuracy is crucial. For IMAT, the treatment planning accuracy is limited by the static-beam planning approach where the continuous delivery arc is approximated by a series of static beam angles. Dose distributions calculated using these static planning beams might be different from the actual continuous arc delivery. To minimise these discrepancies between planning and delivery, dose should be computed with a series of finely-spaced interpolated beams. With the use of MC dose engine, IMAT plans can be computed accurately and efficiently since the CPU time of MC dose calculation methods is independent of the number of beams. Based on 10 patient case studies, it was found that the differences between static-beam planning and interpolated-beam final dose calculation was affected by the complexity and variation of the MLC aperture shapes and weightings in the plan. Although the number of cases studied was limited to provide any statistical conclusions, these preliminary results have shown that dose errors can be significant and can affect patient outcomes if these effects are not taken into the account during planning. For treatment plans with large aperture shape variation and rigorous MU fluctuation between the adjacent planning angles, dose distribution should be calculated with a large number of finely interpolated beams in order to simulate the actual delivery accurately.

Chapter 5

Improving the delivery efficiency of IMAT

5.1 Background and objectives

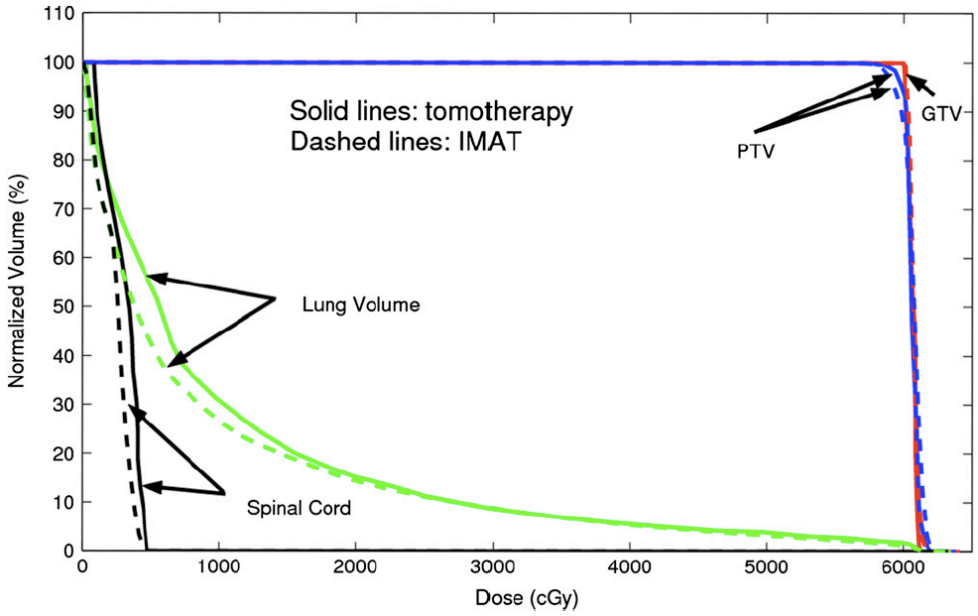
For the two prominent rotational IMRT techniques, a previous study has found that IMAT is capable of delivering a comparable dose distribution to helical tomotherapy, which is known for its high dose conformality [Cao *et al* 2007]. In the plan comparison study of helical tomotherapy and IMAT by Cao *et al*, IMAT plans were compared with tomotherapy plans for several different disease sites. In figure 5.1, the DVH comparisons show that IMAT can achieve very similar dose coverage to the targets while delivering much lower dose to the critical organs compared to tomotherapy. Although this study only presented a small population of clinical cases and no definite conclusions can be drawn, the study demonstrated the dosimetric capability of IMAT. Furthermore, while tomotherapy requires a special machine for treatment delivery, IMAT can be delivered using a general-purpose linac. Because of the restriction of the design of the tomotherapy machine, non-coplanar arcs cannot be used in tomotherapy. With a non-zero couch angle, non-coplanar arcs can be achieved in IMAT, where they are useful for certain plans such as brain cases (e.g. vertex arcs can be used to minimise irradiation to the optical organs).

Although the superior dosimetric quality of IMAT has been supported by several plan comparison studies, IMAT has not been realised for clinical routines since its first proposal in 1995 [Ma *et al* 2001][Yu *et al* 2002] [Cao *et al* 2007][Tang *et al* 2010]. The incomplete development of treatment planning of IMAT was one of the major limitations. However, recently, tremendous efforts have been spent on developing treatment planning tools for IMAT. For example, several robust IMAT leaf sequencing algorithms have been developed in several different institutions while others have successfully adapted the existing IMRT optimisation algorithms for IMAT planning [Wong *et al* 2002][Earl *et al* 2003][Shepard *et al* 2007][Luan *et al* 2008] [Gladwish *et al* 2007]. In addition, the traditional misconception of requiring a large amount of beams for accurate arc dose calculation has been resolved and an efficient dose calculation method using MC dose engine has been proposed as described in the previous chapter [Tang *et al* 2008]. With different optimisation and dose calculation tools developed and refined, IMAT has transitioned from a theoretical concept to a practical radiotherapy technique with a robust treatment planning scheme. Although these algorithms are currently available for research use only, they suffice for generating clinical acceptable IMAT treatment plans that are deliverable.

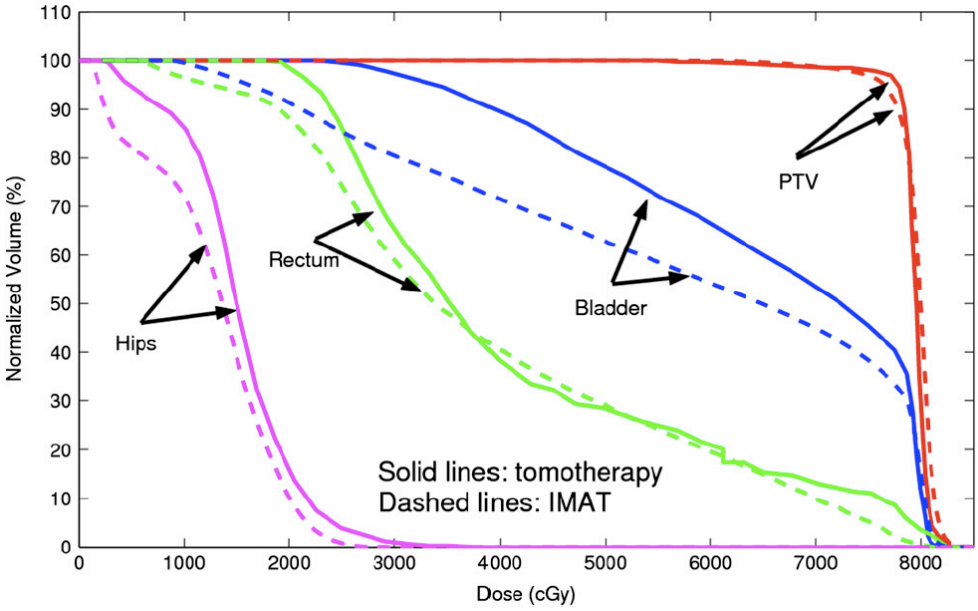
The other limitation for IMAT to become clinically applicable was the unavailability of delivery systems. In order to deliver an IMAT plan with modulating MLC aperture shapes and MU weightings, a linac must be equipped with dynamic control of the MLC positions and dose rates as the gantry is rotating. However, while existing linacs are able to control dynamic motions of MLC, continuous dose-rate variation was not possible during gantry rotation. In 2007, Varian Medical Systems enabled IMAT delivery with their

latest existing linacs for which the MLC and dose rates can be dynamically modulated with simultaneous gantry motion.

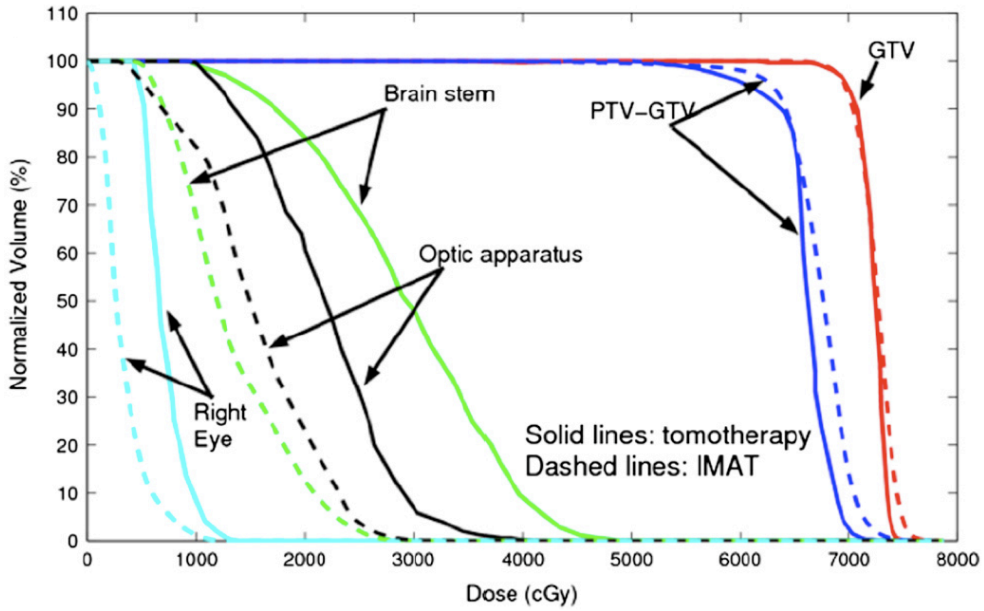
With the advances in technology, the availability of treatment planning system and delivery unit now fulfills the basic technical requirements for clinical use, while the plan quality of IMAT has been validated in several different studies comparing to the current standards such as tomotherapy and multiple static-field IMRT. Despite the comparable dose distributions, the delivery time of IMAT is similar to multiple static-field IMRT. Depending on the complexity of the case, a typical IMAT plan obtains 5 to 12 multiple overlapping arcs. Given the maximum gantry speed is 1 minute per 360-degree rotation (i.e. 1 minute per full rotational arc), it takes approximately 6 to 15 minutes to deliver IMAT. Note that the IMAT delivery time is proportional to the number of arcs involved in a plan with some residual time for machine calibration checks between each arc. With the same patient set-up procedure, the treatment delivery time or beam-on time (BOT) of multi-arc IMAT is similar to IMRT. These close similarities in plan quality and delivery efficiency to IMRT and tomotherapy do not suffice IMAT to bring a new impact to the clinic. In order to exploit the full potential of IMAT, an alternative delivery mode was developed to improve the efficiency of treatment delivery.


Solid lines: tomotherapy
Dashed lines: IMAT
PTV
GTV
Lung Volume
Spinal Cord
Normalized Volume (%)
Dose (cGy)
0
10
20
30
40
50
60
70
80
90
100
110
0
1000
2000
3000
4000
5000
6000

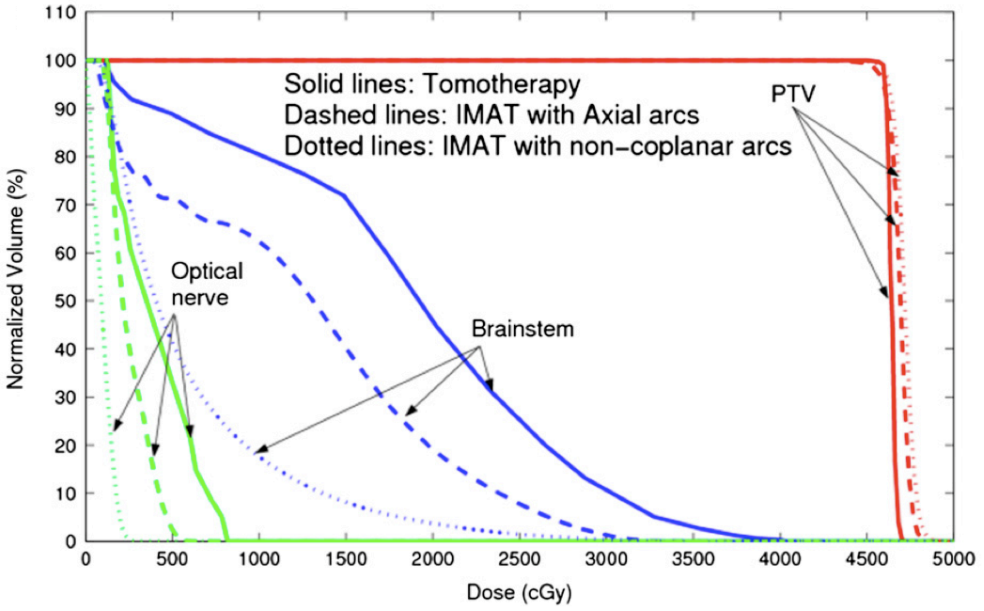
(a)


PTV
Rectum
Bladder
Hips
Solid lines: tomotherapy
Dashed lines: IMAT
Normalized Volume (%)
Dose (cGy)
0
10
20
30
40
50
60
70
80
90
100
110
0
1000
2000
3000
4000
5000
6000
7000
8000

(b)



(c)



(d)

Courtesy of D. Cao

Figure 5.1: DVH comparisons of helical tomotherapy and IMAT plans for (a) a lung case, (b) a prostate case, (c) a HN case, and (d) a brain case.

5.2 The development of single-arc IMAT

5.2.1 Converting multi-arc IMAT into single-arc IMAT

Hypothetically, if a beam is delivered at a small finite angle away from the original planning angle, the dosimetric effect induced by the angular difference should be minimal [Tang *et al* 2007]. To test this hypothesis, three multi-arc IMAT plans (1 brain, 1 HN, and 1 prostate cases) were converted into single-arc IMAT plans. The multiple overlapping MLC apertures at the planning beam angles were rearranged and placed in between the planning beam intervals such that all the MLC apertures were delivered in a single gantry sweep. To illustrate the conversion method, a 5-arc IMAT plan that was planned with 36 beams was used for demonstration. Figure 5.3 (a) shows 5 MLC apertures stacking at a particular beam angle in the original multi-arc IMAT plans. To "compress" these stacked apertures into neighbouring apertures, the geometric centre of the apertures were first determined. Because the individual overlapping MLC shapes at the static beam angles were planned to be delivered with different arcs and the shapes were not connected for a single-arc delivery, these apertures were rearranged in the order of the geometric centroid positions such that the MLC transitions during delivery could be as smooth as possible. The geometric centre of the 2D MLC apertures $Q_x$is defined by,

$$Q_x = \bar{y}_T \cdot A_T = \sum_{i=1}^{60} \bar{y}_i \cdot (A_e)_i, \tag{5.1}$$

$$(A_e)_i = [(d_A)_i - (d_B)_i] \cdot w_i, \tag{5.2}$$

$$A_T = \sum_{i=1}^{60} (A_e)_i. \tag{5.3}$$

The 2D MLC apertures are shaped using up to 60 MLC leaf pairs (Varian Millennium^{TM} 120, Varian Medical Systems, Palo Alto, CA). To calculate the area of the 2D MLC segment A_T, the elementary area that is defined by each opposed leaf pair is summed up. Each elementary area $(A_e)_i$ of leaf pair i is defined by the product of the distance between the opposed leaf pair, i.e. distance between the leaf in bank A and bank B (d_A and d_B), and the width of the leaves w. For the 60 leaf pairs of the Varian Millennium^{TM} 120 MLC model, w of each of the leaves in the first 10 leaf pairs is 10 mm; 5 mm for the next 40 leaf pairs; and 10 mm for the last 10 leaf pairs (see figure 5.2).

Once the geometric centres were determined, the stacked apertures were dislocated from the original beam angle and were spread evenly around this planning angle over the beam intervals in the order of the geometric centroid positions as shown in figure 5.3 (b). The "spreading" angular spacing of these relocated apertures $\Delta\theta_s$ is defined by the static planning beam interval $\Delta\theta$ and the number of arcs n in the plan (i.e. the number of MLC apertures at each planning beam):

$$\Delta\theta_s = \Delta\theta/n. \tag{5.4}$$

Thus, for the original 5-arc IMAT plan that was planned with 36 static beams with $\Delta\theta = 10°$, $\Delta\theta_s$ for the spread MLC apertures in the converted single-arc plan is 2°. Given that the apertures displayed in figure 5.3 (c) were rearranged from the leftmost centroid position (aperture number 3) to the rightmost centroid position (aperture number 5) and assume that the rotational direction of the single-arc is anticlockwise, the stacked apertures of

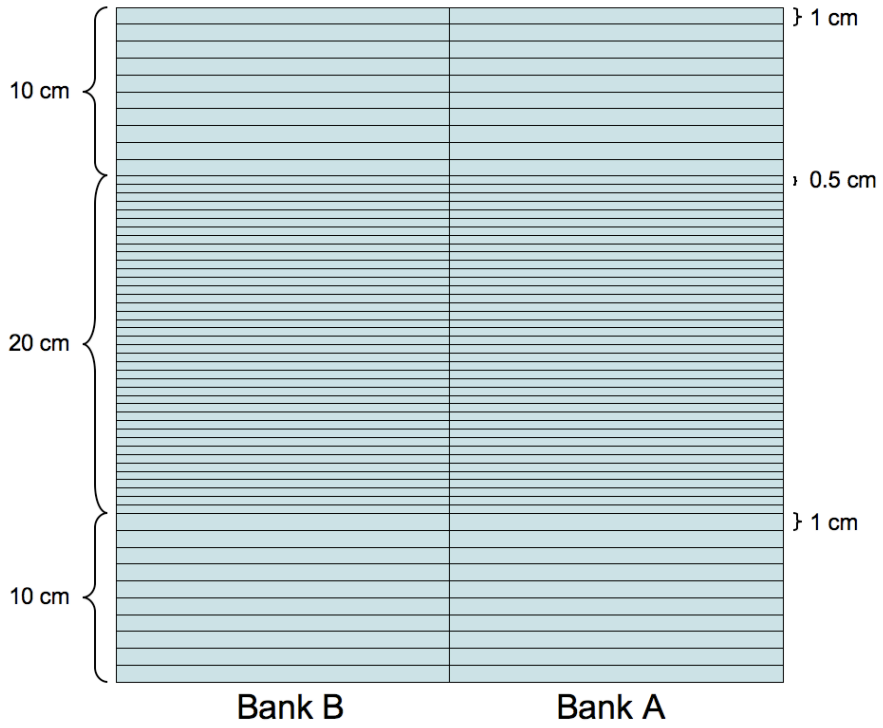


Figure 5.2: An illustration of the Varian Millennium^{TM} 120 MLC model (not to scale). It comprises of two banks of MLC leaves, with 60 leaves in each bank. The top 10 and bottom 10 leaf pairs have a leaf width of 1 cm, and the 40 leaf pairs in the middle have a leaf width of 0.5 cm.

the next adjacent beam should be arranged in the order of the rightmost to the leftmost of centroid positions. In short, the apertures were rearranged and relocated with alternating order of the geometric centres between the adjacent planning beams for a smooth delivery. The resultant single-arc IMAT plan now contains 36×5 beam directions, each beam consists of one MLC aperture only.

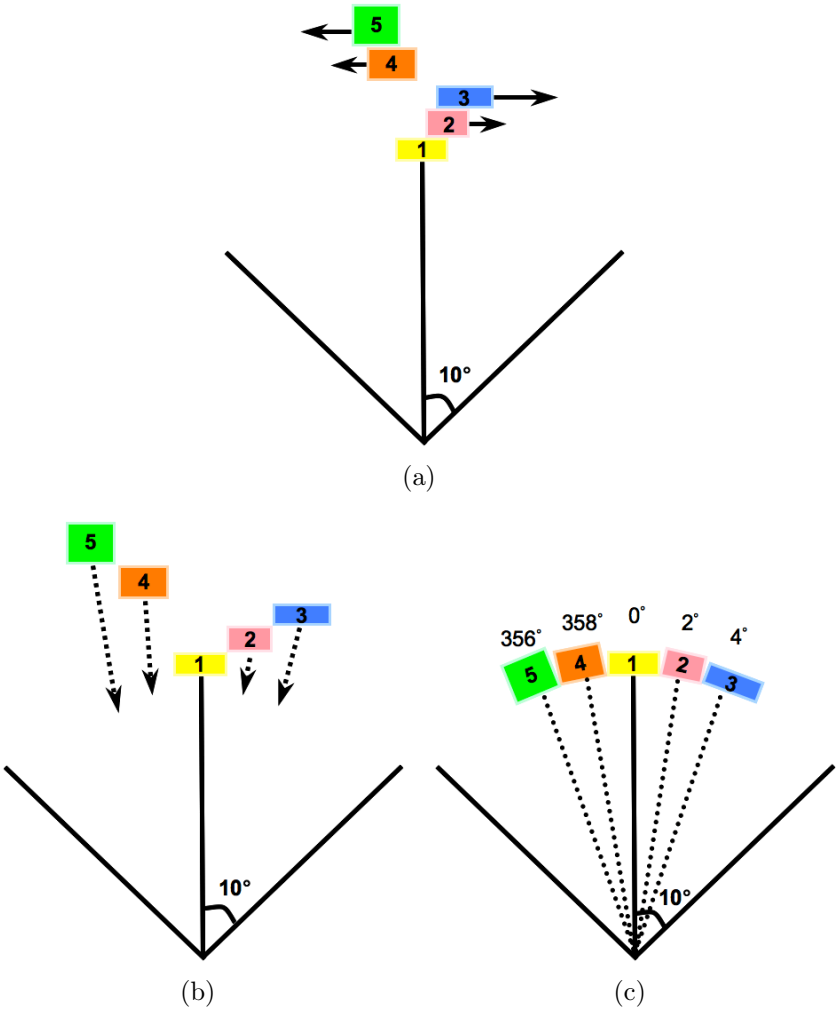


Figure 5.3: Schematics of a multi-arc IMAT plan converted into a single-arc IMAT plan. (a) The geometric centroid positions were first determined for the stacked MLC apertures in the multi-arc IMAT plan. (b) The stacked apertures were shifted from the original planning angle and rearranged according to the order of the geometric centres. (c) The ordered apertures were spread within the planning beam intervals resulting in a single-arc IMAT plan.

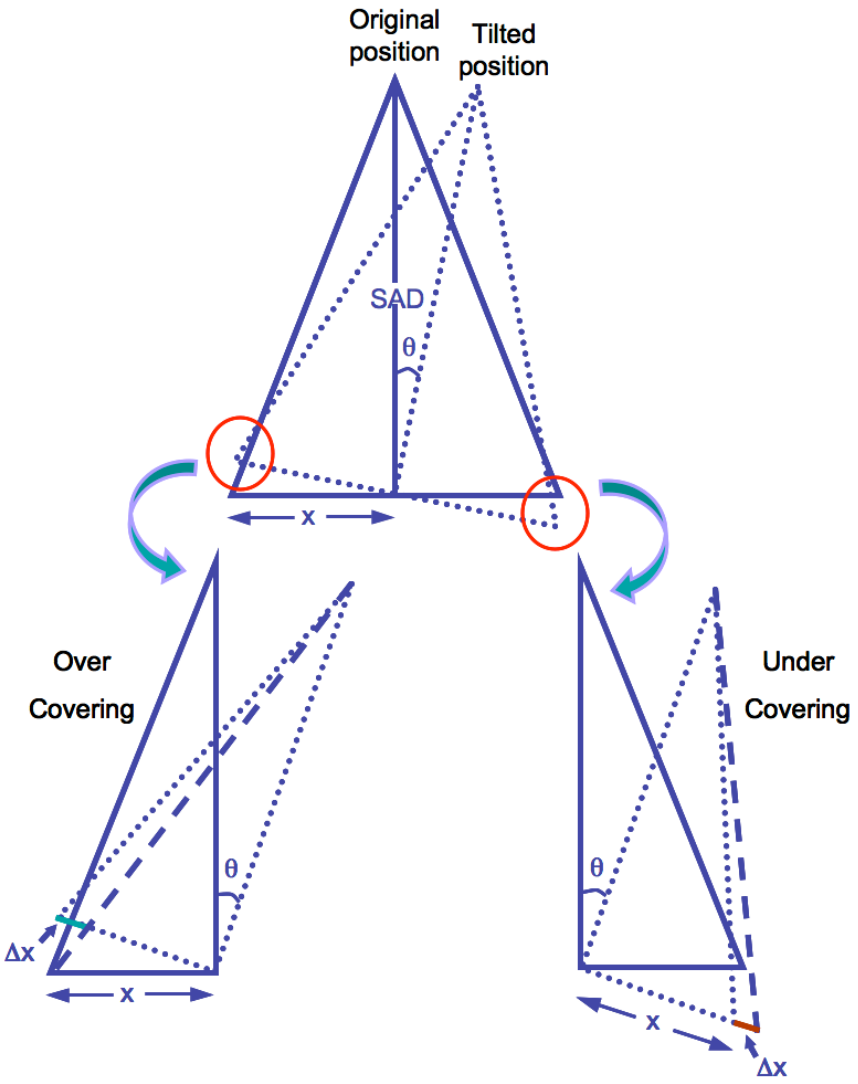


Figure 5.4: An illustration of the geometric error of the apertures after the single-arc conversion. As the aperture is tilted by θ to the right-hand-side of the original position, the left side of the aperture field over-covers the target and the right side of the field under-covers the target.

As the apertures were shifted or tilted from the original planning angles to different angular positions during the single-arc conversion, some of these apertures may under-cover or over-cover the target. When the apertures are under-covering the target, radiation delivered to the target is reduced; and when the apertures are over-covering the target, excessive dose is delivered and

may possibly deliver extra dose out of the target to the surrounding normal tissues. As shown in figure 5.4, when the MLC aperture is shifted from its original position by an angular difference of θ, the apparent field size to the target will be extended by Δx on one side and will be shrunk by Δx on the other side, compared to what was originally planned. In order to compensate for this geometric error, a correction was applied to the shifted apertures using equations 5.5 and 5.6, where x is the width between the isocentre to the edge of the individual MLC leaf and SAD is the source-to-axis distance, which is 100 cm by default for isocentric treatments such as IMAT. To correct for the side of aperture that is over-covering the target, equation 5.5 was applied to all the MLC leaves on this side of the field and to correct for the under-covering of the target, equation 5.6 was applied to the MLC leaves on the other side of the field.

$$x - \Delta x = SAD \cdot \frac{x \cdot sin(\frac{\pi}{2} - \theta)}{SAD + x \cdot cos(\frac{\pi}{2} - \theta)} = SAD \cdot \frac{x \cdot cos\theta}{SAD + x \cdot sin\theta} \qquad (5.5)$$

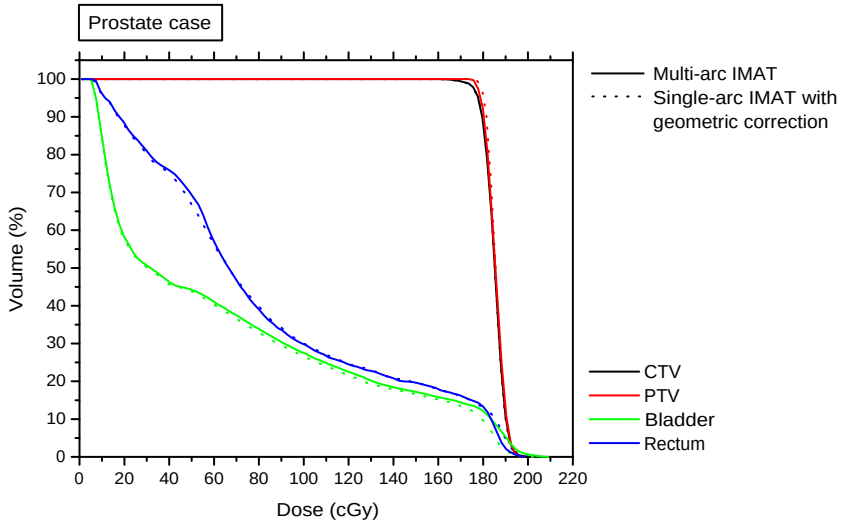
$$x + \Delta x = SAD \cdot \frac{x \cdot sin(\frac{\pi}{2} - \theta)}{SAD - x \cdot cos(\frac{\pi}{2} - \theta)} = SAD \cdot \frac{x \cdot cos\theta}{SAD - x \cdot sin\theta} \qquad (5.6)$$

For 3 test patient cases, IMAT plans were generated using the intensity-map optimisation planning approach. A set of ideal intensity maps were first optimised for 36 static planning beams in Pinnacle^3 using the P^{3}IMRT module, followed by a MLC leaf sequencing process using the k-link IMAT sequencer. The resultant multi-arc IMAT plans were transferred to a program that was written in C for the conversion into single-arc plans using the conversion method mentioned above (see appendixes C and D). A final dose

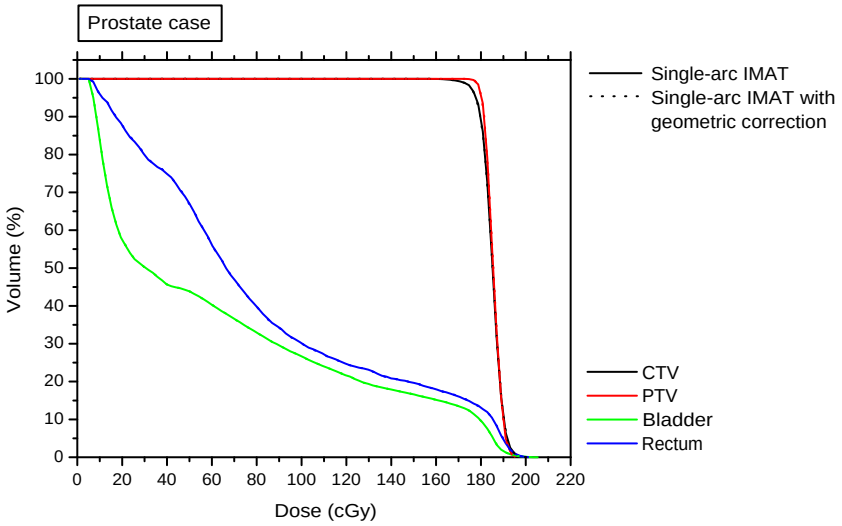
calculation of the converted single-arc IMAT plans and the original multi-arc IMAT plans were performed with MCKS. A dose grid of $2 \times 2 \times 3$ mm^3 was used and the endpoint of all simulations was where 2% statistics was achieved. To accurately simulate the actual delivery, all plans were calculated using a large number of interpolated beams with an interpolation spacing of 0.5°.

The DVH comparisons between multi-arc IMAT plan and the corresponding converted single-arc IMAT plans of the 3 cases are shown in figures 5.5 - 5.7. The 5-arc prostate plan was converted into a 180-beam single-arc plan; the 11-arc brain case was converted into a 396-beam single-arc plan; and the 12-arc HN plan was converted into a 432-beam single-arc plan. For all cases, the multi-arc IMAT plan and the single-arc IMAT plan were almost identical, indicating that the relocation of the MLC segments during the single-arc conversion process induce minimal dosimetric effects. This was also observed in the comparisons between single-arc plan with geometric corrections and single-arc plan without geometric corrections in figures 5.5 (b), 5.6 (b), and 5.7 (b). These results have shown that multi-arc IMAT plans can be delivered with only one single gantry rotation without compromising the plan quality, greatly improving the delivery efficiency.

In spite of the encouraging results found in this feasibility study, the single-arc conversion method used was not the ideal approach for single-arc IMAT planning. To optimise directly the plan quality of single-arc IMAT, an algorithm called arc-modulated radiation therapy (AMRT) was developed in the University of Notre Dame and University of Maryland [Wang *et al* 2008].

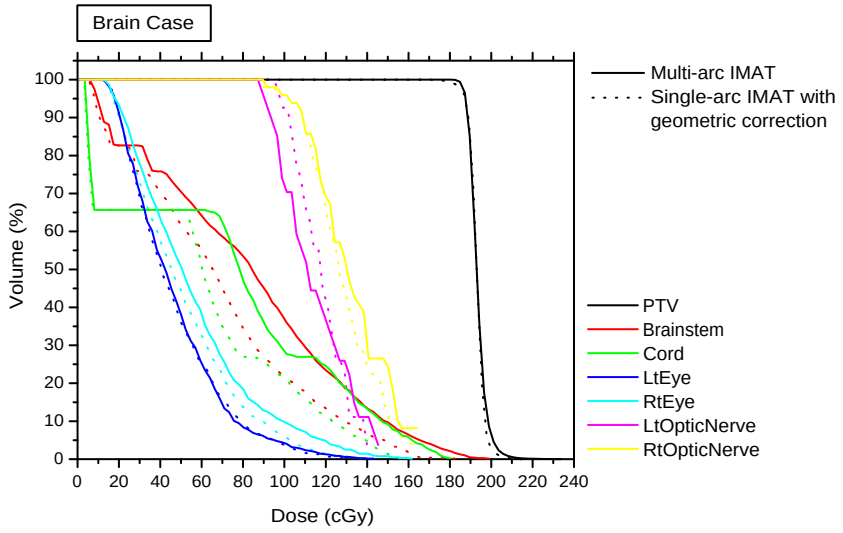


(a)

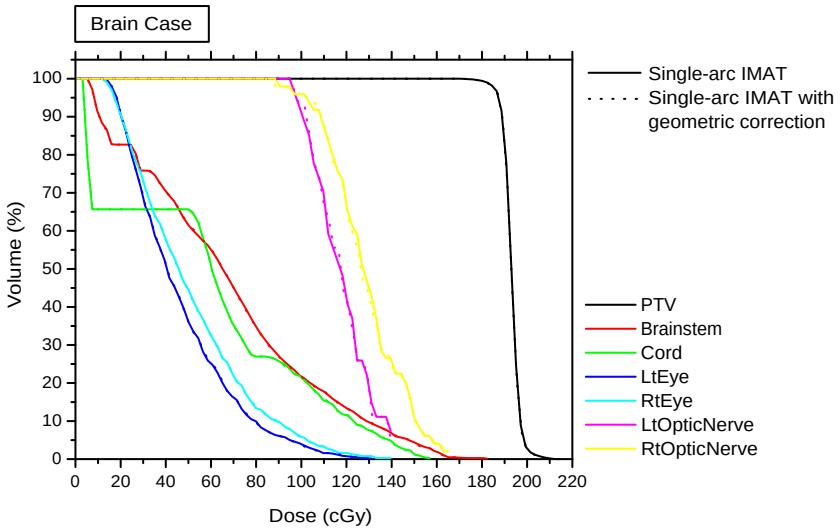


(b)

Figure 5.5: DVH comparisons between (a) the multi-arc IMAT plan the corresponding single-arc IMAT plan with geometric correction and, (b) the single-arc IMAT plans with and without geometric correction for the prostate case.

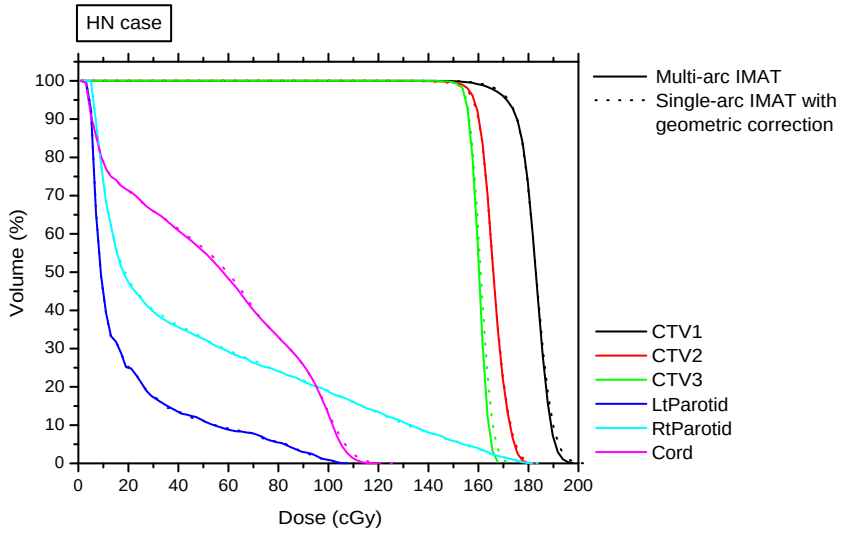


(a)

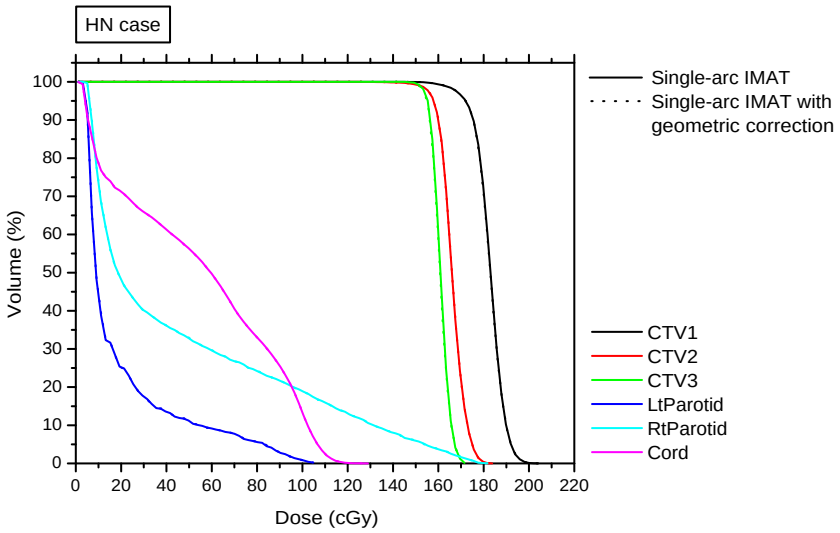


(b)

Figure 5.6: DVH comparisons between (a) the multi-arc IMAT plan the corresponding single-arc IMAT plan with geometric correction and, (b) the single-arc IMAT plans with and without geometric correction for the brain case.



(a)



(b)

Figure 5.7: DVH comparisons between (a) the multi-arc IMAT plan the corresponding single-arc IMAT plan with geometric correction and, (b) the single-arc IMAT plans with and without geometric correction for the HN case.

5.2.2 Single-arc IMAT at University of Maryland: Arc-Modulated Radiation Therapy

From the previous section, it was found that intensity modulations can be delivered using multiple overlapping arcs while complex dose distributions can also be delivered in a single-arc form. Although there is no intensity modulation in a single-arc plan as there is only one MLC aperture per beam direction, the dose delivered with closely-spaced beams overlap and accumulate to a complex dose distribution from the target's point of view (inside the patient). Because only one single gantry rotation is involved in the entire treatment plan, the delivery time can be greatly reduced.

AMRT[1] is an optimisation algorithm developed at the University of Notre Dame and University of Maryland for the purpose of single-arc IMAT planning [Wang *et al* 2008]. The planning strategy of AMRT follows the intensity-map optimisation approach where AMRT is responsible for the MLC leaf sequencing procedure. A typical AMRT planning process starts with the optimisation of a set of ideal intensity maps using K static beams in a commercial TPS such as Pinnacle[3]. These K sets of ideal intensity distributions are then transferred to the AMRT algorithm so that a set of MLC sequence along the delivery arc with the total arc length of $K \cdot \theta$ degrees is generated, where θ is the angular spacing between the static beams. For each set of the K intensity maps, all possible leaf openings that can reproduce the optimised fluence are generated. For every leaf pair opening, the optimal deliverable leaf trajectory between the adjacent planning angles that reproduces the closest intensity profile that was optimised in the initial optimisation is found by the coupled

[1]Work done in this project contributed to the quality control of the AMRT development, while the algorithm codes were designed in the University of Notre Dame by Dr. Chao Wang.

path planning algorithm used in computer science. By using the shortest path algorithm borrowed from graph theory, the optimal set of leaf trajectory with the minimal differences to the ideal intensity maps is determined. A single-arc sequence is formed by combining the leaf trajectories of all leaf pairs and a MLC displacement constraint is applied. MLC positions may be adjusted to fulfill the MLC constraint requirement but a segment weight optimisation can be performed to retrieve any loss of plan quality. The resultant output of the AMRT optimisation is a single-arc IMAT MLC sequence where a set of apertures is spread over a delivery arc. Figure 5.8 exemplifies a simple schematics of such single-arc IMAT sequence. Different from multi-arc IMAT where there are multiple stacked MLC apertures at each planning angle and are delivered by multiple overlapping arcs, there is only one MLC aperture per planning angle in a single-arc IMAT plan. The MLC shapes dynamically change from one angle to the next as the gantry is continuously rotating about the patient. Because the MU weightings of these evenly-spaced beams or apertures are optimised (i.e. variable), dose-rate variation is required for delivery.

The total leaf sequencing time for AMRT is less than 3 minutes (note that the AMRT algorithm is a stand-alone program and is not incorporated into and streamlined with a commercial TPS) but the initial fluence optimisation may take up to 1 hour using a commercial TPS. To date, the current AMRT algorithm is not enabled with collimator angle optimisation and the collimator angle is first defined during the initial intensity map optimisation (collimator angle can be non-zero).

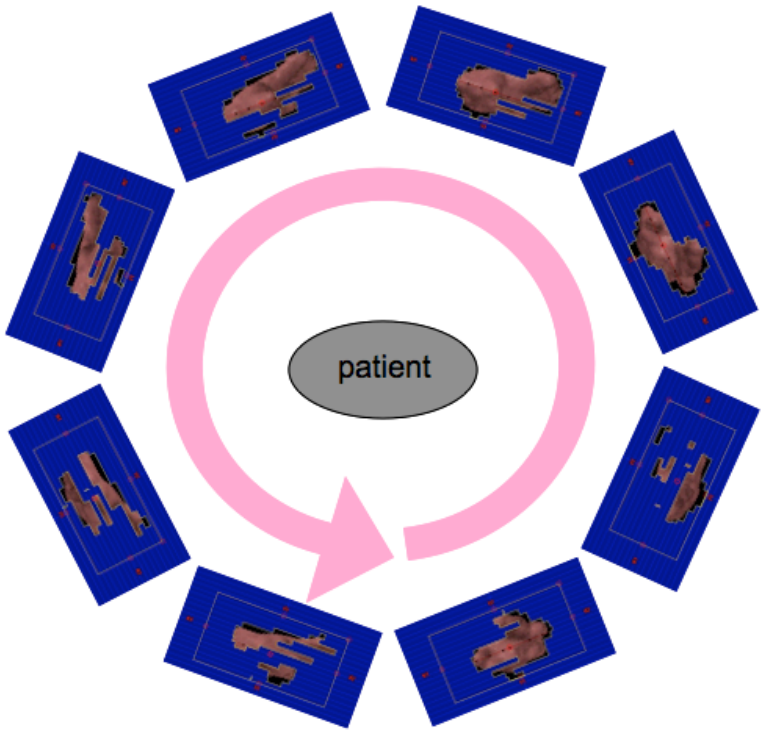


Figure 5.8: An example display of the MLC sequence of an AMRT plan. The pink colour region is the target.

5.3 Summary

A derivative of the IMAT technique is investigated by a proof of principle study. By collapsing the stacked apertures in a multi-arc IMAT plan into a single-arc IMAT plan, it was shown that the plan quality of a multi-arc IMAT plan can be delivered with only one gantry rotation. Single-arc IMAT primarily aims at improving delivery efficiency while providing a high quality dose distribution. With a dedicated optimisation algorithm, AMRT, the potential of single-arc IMAT can be fully explored.

Chapter 6

Clinical feasibility of single-arc IMAT: plan comparison of AMRT, IMAT, and IMRT

6.1 Background and objectives

To investigate the clinical feasibility of single-arc IMAT, the plan quality and delivery times of AMRT were compared with multi-arc IMAT and multiple static-field IMRT for 12 past patient cases that were previously treated with IMRT [Tang *et al* 2010]. The cases were selected to present a variety of complex cases for different disease sites, including 3 HN, 3 brain, 3 lung, and 3 prostate cases. Apart from the prostate cases, all cases were at different disease stages, with different avoidance structures, and different prescription doses which represent different clinical scenarios. An IMRT, an IMAT, and an AMRT plans were created for each case. All plans were generated to clinically-accepted standard that was approved by a physician. To ensure a fair comparison, fundamental planning parameters such as isocentre, prescription dose, planning objectives, and optimisation constraints were kept the same for all three sets of plans.

6.2 Treatment planning

The IMRT, IMAT, and AMRT plans were generated using the 4-step intensity-map optimisation approach. First, the ideal intensity maps were optimised using the same commercial TPS, followed by a MLC sequence conversion. After leaf sequencing, a segment weight optimisation was conducted as an attempt to retrieve any loss of plan quality during MLC segmentation. A final dose calculation was performed using MCKS where the IMAT and AMRT plans were computed using a large number of interpolated beams to closely simulate the actual delivery.

6.2.1 Optimisation of ideal intensity distributions

IMRT plans that conformed to clinical standards were first generated for all cases as a benchmark. A set of ideal fluence maps were optimised using 7 - 9 beams in Pinnacle^3 utilising the P^{3}IMRT module. For a fair comparison, the same set of optimisation constraints were used to generate the ideal intensity distributions for the IMAT and AMRT plans using 36 static beam angles. The choice of optimisation constraints for the avoidance structures or critical organs were referred to the clinical guidelines at the University of Maryland and Radiation Therapy Oncology Group (RTOG) protocols for HN (05-22), CNS (05-13), lung (06-17), and prostate (04-15), while the prescription dose to the tumour was based on the actual clinical treatment (`www.rtog.org`). The prescription dose and dose constraints are tabulated in tables 6.1 and 6.2, respectively.

All cases were planned to fulfill the minimum clinical acceptance criteria for target coverage, i.e. $V_{95\%}$ of PTV $> 95\%$, where 95% of the volume of

Table 6.1: A list of prescription dose of the patient cases.

Case	Diagnosis	Prescription dose
HN 1	T4N0 squamous cell of the left maxilla and recurrent T1N0 squamous cell of the retromolar fossa	59.4 Gy
HN 2	T4N0 squamous cell of the left nasal cavity and maxillary sinus	70.2 Gy
HN 3	T2N1 squamous cell of the right floor of mouth	72 Gy
Brain 1	Recurrent astrocytoma in the left frontoparietal lobe	60 Gy
Brain 2	Right sphenoid wing meningioma	54 Gy
Brain 3	Grade III anaplastic oligoastrocytoma	59.4 Gy
Lung 1	Recurrent squamous cancer in the right lung	70.2 Gy
Lung 2	T2N0 right upper lobe NSCLC	72 Gy
Lung 3	T3N3 NSCLC	66.6 Gy
Prostate 1 - 3	Low risk adenocarcinoma	78 Gy

the PTV should receive at least 95% of the prescription dose. While ensuring the target received adequate dosage, the dose to the avoidance organs were kept under the limits. However, for some cases, the maximum dose limits were unavoidably exceeded due to the proximity or overlapping of the target volume.

6.2.2 Leaf sequencing

For IMRT, the ideal intensity maps were converted into a set of step-and-shoot MLC sequence in Pinnacle^3; for IMAT, multi-arc IMAT MLC sequences were optimised using the k-link IMAT leaf sequencer; and for AMRT, single-arc MLC sequences were generated using the AMRT algorithm. Different from IMRT, MLC sequencing for IMAT and AMRT is more complicated as the algorithm needs to consider the MLC connectivity between the adjacent aper-

Table 6.2: Dose constraints for the critical organs.

Organ	Measure	Limit
Parotid glands	Mean dose	< 26 Gy
Larynx	Max dose	< 45 Gy
Brainstem	Max dose	< 60 Gy
Spinal cord	Max dose	< 50 Gy
Optic chiasm	Max dose	< 54 Gy
Optic nerves	Max dose	< 45 Gy
Retina	Max dose	< 50 Gy
Brachial plexus	Max dose	< 60 Gy
Heart	V_{67}	< 45 Gy
Bilateral lungs	V_{20}	$< 37\%$
Bladder	V_{65}	$\leq 50\%$
	V_{75}	$\leq 25\%$
Rectum	V_{60}	$\leq 50\%$
	V_{70}	$\leq 25\%$

tures for deliverability. Based on equation 3.1, a MLC displacement constraint of 3 cm per planning beam interval (10°) was used for IMAT, but for AMRT, the choice of MLC constraint was more complicated.

In AMRT, because the number of beams increases, the MLC displacement constraint limits the treatment delivery time, i.e. the larger the MLC constraint, the longer the delivery time it is. The relationship among the MLC displacement constraint and delivery time is described in equations 6.1 - 6.3. The minimum delivery time T_{min} is governed by the fastest or maximum gantry speed ω_{max} and the arc length of the plan $K \cdot \theta$, where the maximum gantry speed is regulated by the maximum mechanical MLC speed v_{max} and the maximum MLC displacement between the adjacent aperture, i.e. MLC displacement constraint Δd. Given that all AMRT plans consist of full rotational arc (i.e. $K \cdot \theta = 360°$), the delivery time is only affected by the MLC displacement constraint of the plan. For the purpose of improving delivery efficiency with AMRT, MLC displacement constraints of 1 cmdeg^{-}1

to 2.5 cmdeg^{-}1 were used, resulting in delivery times of 2 to 5 minutes (note that v_{max} of MLC was defined as 3 cms^{-1}). In addition to the selection of MLC constraint, the number of beams or apertures was also one of the AMRT planning parameters. As an effort to maintain a fair comparison, the number of apertures of the AMRT plans were the same as that in the corresponding IMAT plans.

$$\omega_{max}(^{\circ}\mathrm{s}^{-1}) = \frac{v_{max}(\mathrm{cms}^{-1})}{\Delta d(\mathrm{cmdeg}^{-1})} \tag{6.1}$$

$$T_{min}(s) = \frac{K \cdot \theta(\mathrm{deg})}{\omega_{max}(\mathrm{degs}^{-1})} \tag{6.2}$$

$$T_{min}(s) = \frac{K \cdot \theta(\mathrm{deg}) \times \Delta d(\mathrm{cmdeg}^{-1})}{v_{max}(\mathrm{cms}^{-1})} \tag{6.3}$$

6.2.3 Segment weight optimisation

As an attempt to achieve the optimal plan quality, segment weight optimisation was performed for all plans. Different from the optimisation of ideal intensity maps and MLC sequences, segment weight optimisation only focuses on the MU weightings of the MLC apertures. The optimisation constraints used were the same set used in the initial optimisation of the ideal fluence maps.

Because IMAT and AMRT plans were not generated in Pinnacle^3, a conversion of the MLC sequences was necessary in order to enable the import of the plans. The IMAT and AMRT MLC sequences that contains the MLC shapes and MU weightings were in ASCII format. To import the plans into Pinnacle^3 for segment weight optimisation, these ASCII files were converted

into unix scripts (that were unique for the Pinnacle^3 system) using a program written in C (see appendix E). For IMAT, the MLC sequence is similar to a 36-beam IMRT MLC sequence and each beam contains k apertures, where k is the number of arcs; but for AMRT, the MLC sequence consists of K beams, where $180 < K < 432$. Because the number of beams in the AMRT plans exceeded the limit of Pinnacle^3, a conversion was required in order to import the plans for segment weight optimisation. Contrary to the conversion method of multi-arc IMAT to single-arc IMAT described in section 5.2.1, AMRT plans were converted to multi-arc IMAT plans by stacking the single-arc apertures into multiple overlapping apertures using a program written in C (see appendix F). The single-arc apertures in the AMRT plans were stacked on to 36 evenly spaced beam angles, resulting in a 36-beam AMRT plan with $K/36$ apertures per beam. Since it was proved in the previous chapter that the dosimetric effects are minimal when beams are delivered $\leq 5^\circ$ away from the original planning positions, all single-arc AMRT plans were converted to multi-arc AMRT plans for plan import to Pinnacle^3. After segment weight optimisation, the multi-arc AMRT sequences were processed and the apertures were repositioned back to the original beam angles to single-arc MLC sequences for final dose calculation.

6.2.4 Dose calculation

The dose distributions of all plans were computed using MCKS with a dose grid of $2 \times 2 \times 3$ mm^3 on a single CPU running at 2.0 GHz in a Mac OS X system. The endpoint of all simulations was where 2% statistics was achieved. For IMAT and AMRT plans, interpolated-static beam calculations were performed where the plans were calculated using a large number of interpolated

beams with an interpolation spacing of 0.5° and 0.1°, respectively. To evaluate the plans, the computed doses were imported back to Pinnacle[3].

6.3 Plan evaluation

For the ease of comparison, IMRT, IMAT, and AMRT plans of each case were normalised such that the three sets of plan achieved the same mean dose to the PTV. The quality of the plans were analysed using several clinically-used dose-volume metrics. For GTV, CTV or PTV, $V_{95\%}$ was used to quantify target coverage, measuring the volume of target that receives at least 95% of the prescription dose. The dose uniformity within the target was estimated by the homogeneity index (HI), which was defined as the ratio of the difference between the dose received by 5% (D_5) and 95% (D_{95}) of the target volume, and the mean dose (D_{mean}) received by the target: :

$$HI = \frac{D_5 - D_{95}}{D_{mean}}, \tag{6.4}$$

As indicated by equation 6.4, the lower the HI value, the more uniform the dose distribution. For normal tissues, V_x was used to assess the volume of avoidance structure receiving x Gy of dose, while the mean dose and maximum dose values were also used to evaluate normal tissue dose.

6.4 Delivery times

The delivery times of the plans were estimated for comparison. The delivery times were estimated as the total time taken to deliver the plans from the radiation beam was first turned on until the plan was fully delivered and the

machine was turned off. Patient set-up time was not taken into the account of the estimation, assuming the set-up procedure was the same for all three treatment modalities. For IMRT, the delivery was estimated by the total time required for the gantry to transit between planned beam angles Θ, the total time required for the MLC to transit between the MLC segments j, and the residual machine mode-up time (MOT):

$$t_{beam}(s) \simeq \sum_{i=1}^{n-1} |\Theta_i - \Theta_{i+1}| \cdot 6^{\circ}\mathrm{s}^{-1}, \tag{6.5}$$

$$t_{MLC}(s) \simeq \sum_{i=1}^{n} [(j_i - 1) \cdot 1 \text{ s}], \tag{6.6}$$

$$MOT(s) \simeq (n-1) \times 30 \text{ s}, \tag{6.7}$$

$$T_{IMRT}(s) = \frac{\text{Total MU (MU)}}{\text{Dose Rate (MUs}^{-1})} + t_{beam} + t_{MLC} + MOT. \tag{6.8}$$

Based on the maximum gantry speed of $6^{\circ}\mathrm{s}^{-1}$, the total time required for the gantry to transition from one beam to the next is described in equation 6.5 , where n is the total number of beams in the plan. The time required for the MLC to transform from one aperture shape to the next is accounted by equation 6.6, where j_i is the number of apertures of beam i. Based on the observation of 30 IMRT treatments, the average time taken for a Varian 21 EX linac equipped with Millennium 120 MLC to change aperture shapes within a beam was approximately 1 second. Apart from the time taken for the gantry and MLC to proceed to the planned positions, the machine would perform calibration checks prior the delivery of each treatment beams and

that an average MOT of 30 seconds was observed. Summing up t_{beam}, t_{MLC} and MOT, the total delivery of IMRT also depends on the number of MU in the plan and the dose rate used in treatment. For this study, a constant dose rate of 400 MUmin^{-1} was used to estimate the IMRT delivery times, since this dose rate was used in typical IMRT treatments in the University of Maryland.

The delivery time estimation for IMAT and AMRT was less complicated. For both IMAT and AMRT, dose-rate variation is required for delivery since the MU weightings can vary between the adjacent apertures. However, this does not directly affect the total delivery time. For IMAT, the estimated delivery time T_{IMAT} is proportional to the number of arcs k (note that all arcs of the IMAT plans have the range of 360° and therefore the minimum time for the gantry to rotate per IMAT arc is 360°/6°s^{-1} = 60 s):

$$T_{IMAT}(s) = k \cdot 60(s) + (k-1) \cdot MOT(s). \tag{6.9}$$

For AMRT, the delivery time was estimated using equation 6.3. Given that all AMRT plans contain 360° single-arcs and the maximum MLC speed was set to 3 cms^{-1}, the estimated delivery T_{AMRT} was:

$$T_{AMRT}(s) = \frac{360^\circ}{3 \text{ cms}^{-1}} \cdot \Delta d(\text{cmdeg}^{-1}) = 120 \text{ s} \cdot \Delta d(\text{cmdeg}^{-1}). \tag{6.10}$$

Note that there was no MOT for AMRT delivery since there was only one continuous rotational arc involved in each of the plans.

6.5 Plan comparison: HN cases

For all 3 HN cases, IMRT, IMAT, and AMRT achieved comparable dose distributions as shown in the DVH comparisons in figures 6.1 and plan summaries in table 6.3. The planning parameters and properties such as the number of beams used are listed in table 6.4.

For HN case 1, IMAT achieved the best target coverage with 99.7% in CTV and 98.2% in PTV, while AMRT and IMRT obtained similar target doses with 97.1% and 97.7% in CTV, and 95.9% and 95.6% in PTV, respectively. IMAT also delivered the most homogeneous dose to the PTV with a HI of 0.085, compared to 0.119 in AMRT and 0.114 in IMRT. Although AMRT obtained similar PTV dose homogeneity to IMRT, the dose conformality to the target was better than IMRT as shown in figure 6.2. The normal tissue doses were similar between the three sets of plans but with slightly higher dose to the right parotid in the AMRT plan.

In HN case 2, the tumour was located in the nasopharyngeal region with the right optic nerve and optic chiasm in close proximity, increasing the challenge of treatment planning. The left optic nerve adjoined the PTV and therefore it was compromised for full dose coverage to the target. In this case, IMRT obtained the best target coverage but with slightly higher dose delivered to the right parotid, optic chiasm, and the right optic nerve. Nonetheless, IMRT achieved the most homogeneous dose distribution, followed by IMAT and AMRT. With similar target coverage to the corresponding IMAT plan, AMRT delivered substantially higher dose to the brainstem with 48.1 Gy, compared to 43.2 Gy in IMAT and 40.7 Gy in IMRT, while the dose to other normal tissues was similar to the two other sets of plan.

HN case 3 was a less complicated HN case where there were fewer critical organs in proximity to the PTV except for the larynx and spinal cord. All three plans obtained similar target coverage and normal tissue dose. 96.3% of the CTV was covered with 95% of prescription dose in the IMRT plan, 96.1% in IMAT, and 96.7% in AMRT. For PTV, AMRT delivered slightly higher coverage with 96.9% in $V_{95\%}$ when this was 95.0% in IMRT and 96.7% in IMAT. Although the dose homogeneity in IMRT was slightly lower compared to the other two plans, the dose to the larynx was the lowest in the IMRT plan.

Overall, IMAT provided the best plan quality out of the three sets of plans for the HN cases. AMRT obtained medial dose distributions compared to IMAT and was comparable to that in IMRT. However, the delivery time of IMAT may take at least 15 minutes on average which was the slowest compared to AMRT and IMRT. While AMRT could achieve dose distributions that rival IMRT, the average delivery time was under 4.8 minutes and IMRT may take 8.7 minutes on average (see table 6.5).

Table 6.3: Plan quality summary of the HN cases.

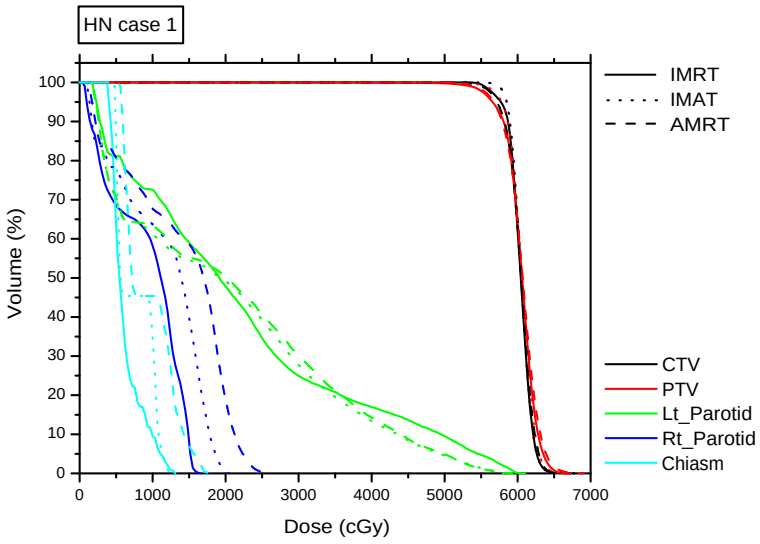
Case	ROI	Dose Index	IMRT	IMAT	AMRT
HN 1	CTV	$V_{95\%}$	97.7%	99.7%	97.1%
	PTV	$V_{95\%}$	95.6%	98.2%	95.9%
		HI	0.114	0.085	0.119
	Left Parotid	Mean	20.3 Gy	20.0 Gy	20.5 Gy
	Right Parotid	Mean	9.0 Gy	11.4 Gy	13.8 Gy
HN 2	GTV	$V_{95\%}$	98.2%	95.9%	96.8%
	PTV	$V_{95\%}$	97.8%	96.2%	97.0%
		HI	0.139	0.151	0.156
	Right Parotid	Mean	22.9 Gy	20.6 Gy	20.5 Gy
	Brainstem	Max	44.1 Gy	46.1 Gy	52.2 Gy
		1 cc	40.7 Gy	43.2 Gy	48.1 Gy
	Optic Chiasm	Max	53.4 Gy	50.2 Gy	52.5 Gy
	Right Optic Nerve	Max	49.9 Gy	44.7 Gy	47.9 Gy
HN 3	CTV	$V_{95\%}$	96.3%	96.1%	96.7%
	PTV	$V_{95\%}$	95.0%	96.7%	96.9%
		HI	0.106	0.093	0.090
	Larynx	Mean	34.0 Gy	36.0 Gy	36.7 Gy
	Spinal Cord	Max	44.2 Gy	45.3 Gy	46.9 Gy
		1 cc	39.1 Gy	39.4 Gy	40.1 Gy

Table 6.4: Plan properties of the HN cases.

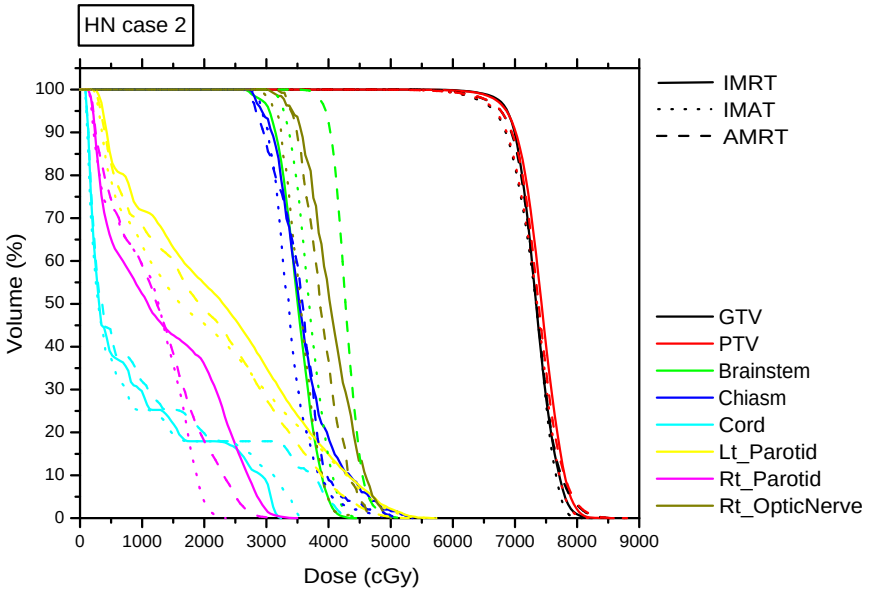
Case	Plan	# beams	# arcs	# apertures	Monitor Units
HN 1	IMRT	7	-	74	672
	IMAT	36	11	396	525
	AMRT	396	1	396	645
HN 2	IMRT	7	-	78	628
	IMAT	36	11	396	754
	AMRT	396	1	396	1038
HN 3	IMRT	9	-	102	530
	IMAT	36	9	324	478
	AMRT	324	1	324	609

Table 6.5: Estimated delivery times of the HN cases.

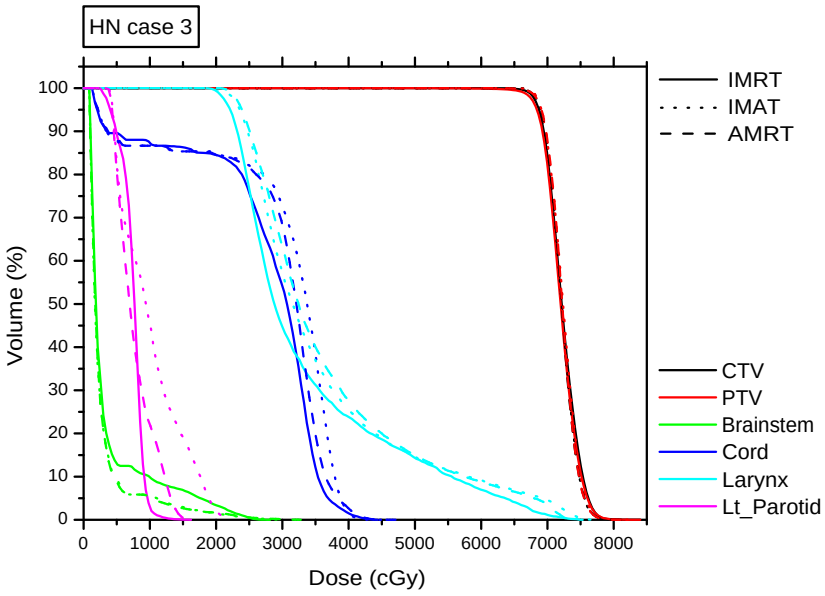
Case	Plan	Estimated Delivery Time (min)
HN 1	IMRT	9.72
	IMAT	16.0
	AMRT	5.50
HN 2	IMRT	7.88
	IMAT	16.0
	AMRT	5.50
HN 3	IMRT	8.58
	IMAT	13.0
	AMRT	3.60



(a)

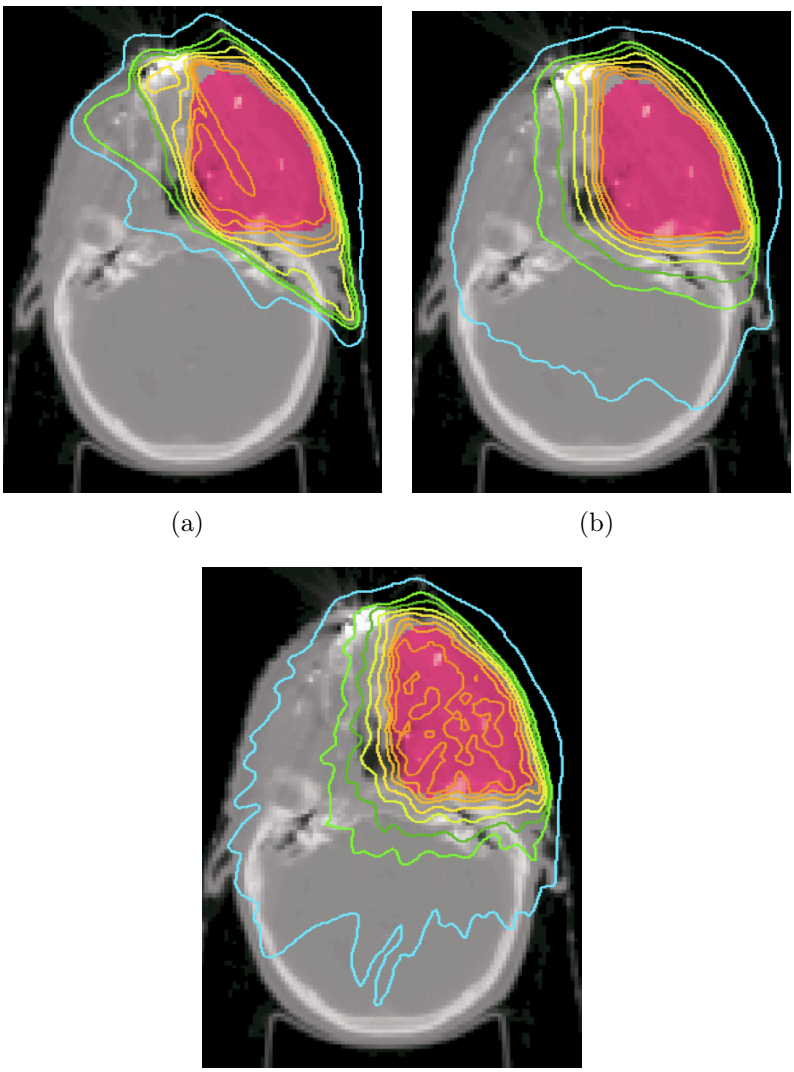


(b)



(c)

Figure 6.1: DVH comparisons between the IMRT, IMAT, and AMRT plans for (a) HN case 1, (b) HN case 2, and (c) HN case 3.



(a) (b)

(c)

Figure 6.2: Isodose distributions of HN case 1 for (a) the IMRT plan, (b) the IMAT plan, and (c) the AMRT plan.

6.6 Plan comparison: brain cases

Similar to the HN cases, all 3 treatment modalities provided comparable dose distributions for all three brain cases. The dose summary of the plans and the plan properties are tabulated in tables 6.6 and 6.7, respectively.

For brain case 1, the brainstem and the left optic nerve were in close proximity of the CTV and therefore the maximum doses for these two organs were slightly compromised in order to cover the target with an adequate amount of prescription dose. With 12 arcs, IMAT was able to obtain the most superior target coverage and target dose homogeneity while delivering the lowest dose to most of the OARs. Although AMRT did not achieve a dose distribution that was as homogeneous as IMAT, it provided a similar plan quality to IMRT but with slightly higher doses to the brainstem, optic chiasm, and left optic nerve.

Similar to brain case 1, the PTV of brain case 2 was also close to several critical organs such as the optic nerves and brainstem. Because a significant portion of the right optic nerve was overlapped with the PTV, the right optic nerve was inevitably compromised and was not regarded as an OAR. Although the left optic nerve did not overlap with the PTV, it adjoined the PTV and therefore it received a higher dose than the maximum limit. With 94.2% of PTV coverage, IMRT almost failed to produce a clinical acceptable plan while this was 99.8% in IMAT and 98.3% in AMRT. Both IMAT and AMRT were able to reduce the total dose to the left optic nerve and left eye, but the total dose to the right eye was substantially increased compared to IMRT. With a biased beam configuration to avoid direct irradiation to the critical organs (as an attempt to minimise the toxicity), the right eye was better-spared in the

IMRT plans. Nonetheless, both IMAT and AMRT provided more conformal dose distributions than IMRT as it can be seen in figure 6.4.

Compared to the previous two cases, the PTV in brain case 3 was located at the superior portion of the skull and was of moderate distance away from the critical organs such as the optic nerves. With a biased beam setting, the 5-beam IMRT plan successfully avoided excessive irradiation to both of the optic nerves but resulted in a slightly less PTV coverage compared to IMAT and AMRT. AMRT obtained similar target coverage to IMAT but with a slightly less homogeneous dose distributions. AMRT also delivered the highest dose to the left eye and both optic nerves, followed by IMAT. However, the maximum dose to the right eye was greatly reduced to 40.9 Gy and 39.4 Gy in IMAT and AMRT, respectively, compared to 46.1 Gy in IMRT.

Although the three sets of treatment plan of all brain cases were generally comparable, IMAT provided slightly better plan quality while AMRT achieved similar dose distributions to IMRT. Although IMAT obtained superior dose distributions, the average estimated delivery time was 13.5 minutes, which was almost 4 times longer than that in AMRT. As shown in table 6.8, the average delivery time of an AMRT plan was 3.4 minutes. Compared to the average delivery time of 5.2 minutes for IMRT, the delivery efficiency of AMRT was only gained by 2 minutes for the brain cases.

Table 6.6: Plan quality summary of the brain cases.

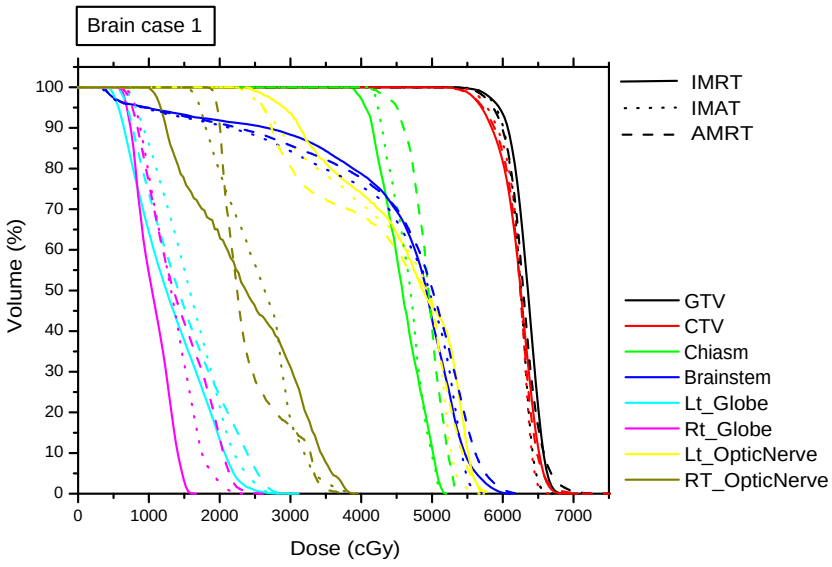
Case	ROI	Dose Index	IMRT	IMAT	AMRT
Brain 1	GTV	$V_{95\%}$	98.9%	99.9%	98.3%
	CTV	$V_{95\%}$	95.0%	97.0%	95.7%
		HI	0.140	0.108	0.136
	Brainstem	Max	61.4 Gy	58.1 Gy	62.3 Gy
		1 cc	58.5 Gy	55.8 Gy	59.5 Gy
	Optic Chiasm	Max	51.9 Gy	51.6 Gy	53.5 Gy
	Left Optic Nerve	Max	57.4 Gy	56.9 Gy	58.3 Gy
	Right Optic Nerve	Max	39.0 Gy	39.8 Gy	38.1 Gy
Brain 2	GTV	$V_{95\%}$	98.7%	100%	99.9%
	PTV	$V_{95\%}$	94.2 %	99.8%	98.3%
		HI	0.090	0.059	0.080
	Brainstem	Max	55.0 Gy	55.0 Gy	54.5 Gy
		1 cc	53.0 Gy	53.5 Gy	52.7 Gy
	Left Optic Nerve	Max	55.2 Gy	54.6 Gy	53.3 Gy
	Left Eye	Max	36.0 Gy	28.8 Gy	28.5 Gy
	Right Eye	Max	28.8 Gy	36.6 Gy	36.8 Gy
Brain 3	PTV	$V_{95\%}$	98.8%	99.2%	99.8%
		HI	0.050	0.043	0.074
	Brainstem	Max	58.8 Gy	59.1 Gy	60.5 Gy
		1 cc	54.6 Gy	53.9 Gy	53.5 Gy
	Left Optic Nerve	Max	43.6 Gy	48.1 Gy	50.2 Gy
	Right Optic Nerve	Max	42.8 Gy	44.5 Gy	49.1 Gy
	Left Eye	Max	39.1 Gy	38.2 Gy	41.2 Gy
	Right Eye	Max	46.1 Gy	40.9 Gy	39.4 Gy

Table 6.7: Plan properties of the brain cases.

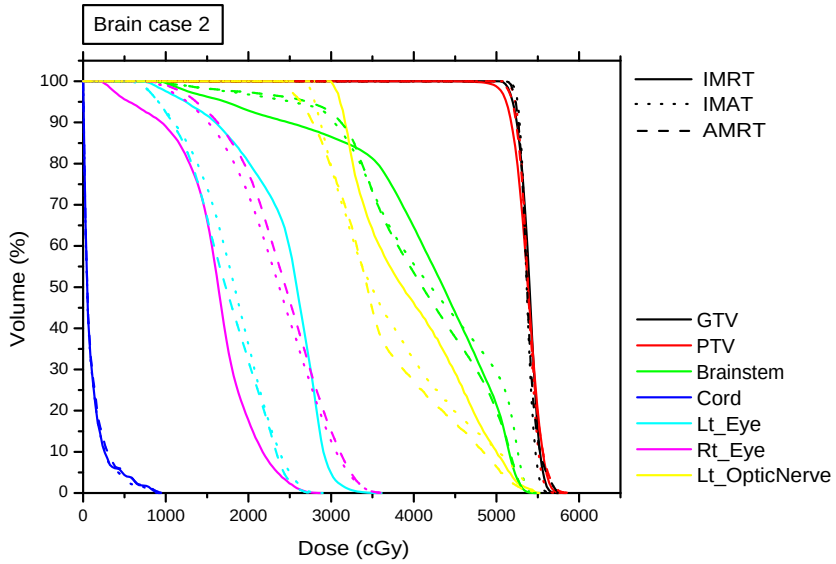
Case	Plan	# beams	# arcs	# apertures	Monitor Units
Brain 1	IMRT	7	-	105	752
	IMAT	36	12	432	681
	AMRT	432	1	432	871
Brain 2	IMRT	7	-	65	393
	IMAT	36	7	252	506
	AMRT	252	1	252	627
Brain 3	IMRT	5	-	55	415
	IMAT	36	9	324	474
	AMRT	324	1	324	654

Table 6.8: Estimated delivery times of the brain cases.

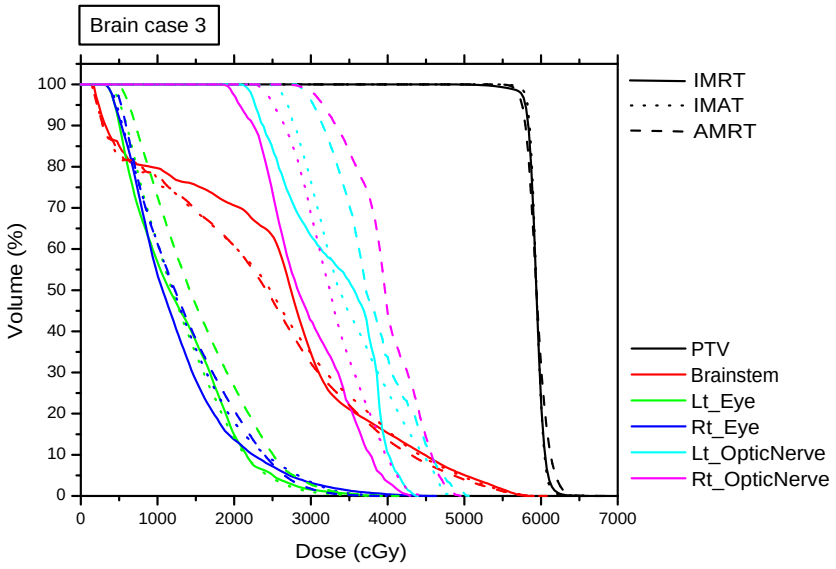
Case	Plan	Estimated Delivery Time (min)
Brain 1	IMRT	6.90
	IMAT	17.5
	AMRT	4.80
Brain 2	IMRT	4.55
	IMAT	10.0
	AMRT	1.75
Brain 3	IMRT	4.06
	IMAT	13.0
	AMRT	3.60



(a)

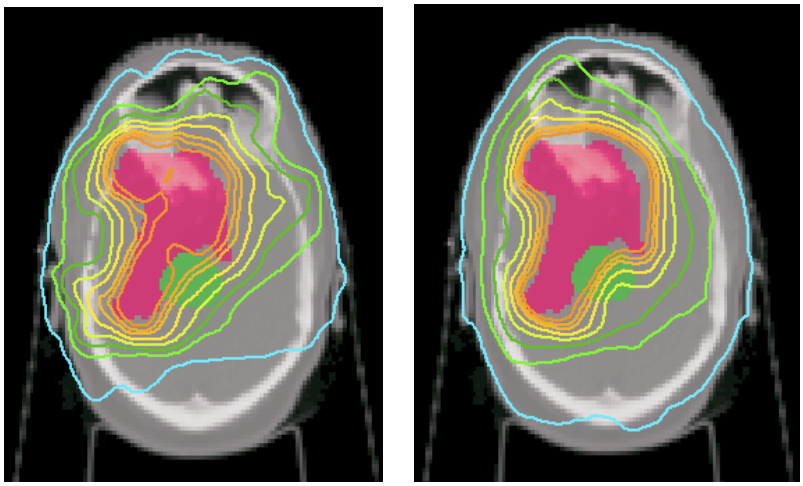


(b)



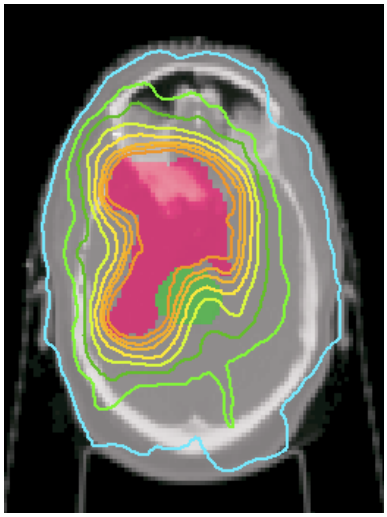
(c)

Figure 6.3: DVH comparisons between the IMRT, IMAT, and AMRT plans for (a) brain case 1, (b) brain case 2, and (c) brain case 3.



(a)

(b)



(c)

Figure 6.4: Isodose distributions of brain case 2 for (a) the IMRT plan, (b) the IMAT plan, and (c) the AMRT plan.

6.7 Plan comparison: lung cases

A summary of results of the lung cases are tabulated in table 6.9 and the DVH comparisons are illustrated in figure 6.6. Table 6.10 lists the plan parameters such as the total number of MU and MLC apertures for reference. For the lung cases, IMRT had the advantage of utilising a biased beam arrangement where the preferred angles could be chosen to introduce minimum irradiation to the healthy normal tissues. This was particularly useful for cases where a tumour was located in the middle of the lung volume in either side of the lungs (see figure 6.5).

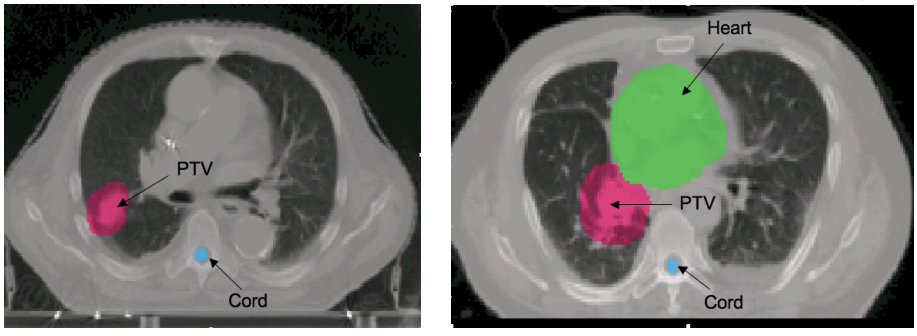
In lung case 1, 7 beams at 140°, 180°, 210°, 250°, 280°, 340°, and 10° were used in the IMRT plan to minimise irradiation to the healthy left lung. Although IMAT and AMRT used full 360-degree arcs and AMRT slightly increased the dose to the left lung, the V_{20} was only 5.3%. However, IMRT obtained the best plan out of the three sets of plan where it achieved the most homogeneous dose to the target and the lowest dose to the lung volumes. AMRT delivered slightly lower dose to the target than both IMAT and IMRT as seen in the DVH comparison in figure 6.6 (a) but it was still a clinically acceptable plan.

The same as in lung case 1, IMRT took the advantage of having the freedom to select preferred beam angles and reduced the dose delivered to the healthy left lung in lung case 2. However, this results in higher irradiation to the right lung as all the beams were located at the right side of the patient delivering the prescribed radiation dose to the target. Although the full rotational arc configuration in IMAT and AMRT spread the dose to a larger volume around the patient, it reduced the dose to the right lung where all the beams were

concentrated at that side of the patient in the IMRT plan. Nonetheless, the major shortfall of the IMAT and AMRT plan was introducing greater dose to the spinal cord, with a maximum dose of 39.7 Gy and 40.9 Gy, respectively, where this was only 18.9 Gy in IMRT.

Different from the previous two cases, lung case 3 was a more complicated case where the target volume included the lymph nodes located in the mediastinum. Because the PTV included a few discontinued volumes, it was difficult to achieve homogeneous target coverage. As seen in figure 6.7, both IMAT and AMRT outperformed IMRT in terms of target coverage and dose homogeneity with the lowest dose to the lung volumes. However, AMRT delivered excessive dose to the spinal cord which exceeded the maximum limit.

For lung cases, AMRT sufficed to be an efficient radiation treatment. Overall, AMRT could rival both IMAT and IMRT as it propelled a balance between plan quality and treatment efficiency. On average, AMRT reduced the delivery time to under 4 minutes while IMRT was 2.5 times slower and IMAT was 3 times longer, as shown in table 6.11.



(a) (b)

Figure 6.5: A sample of the transverse slice at the isocentre-plane of (a) lung case 1 and (b) lung case 2.

Table 6.9: Plan quality summary of the lung cases.

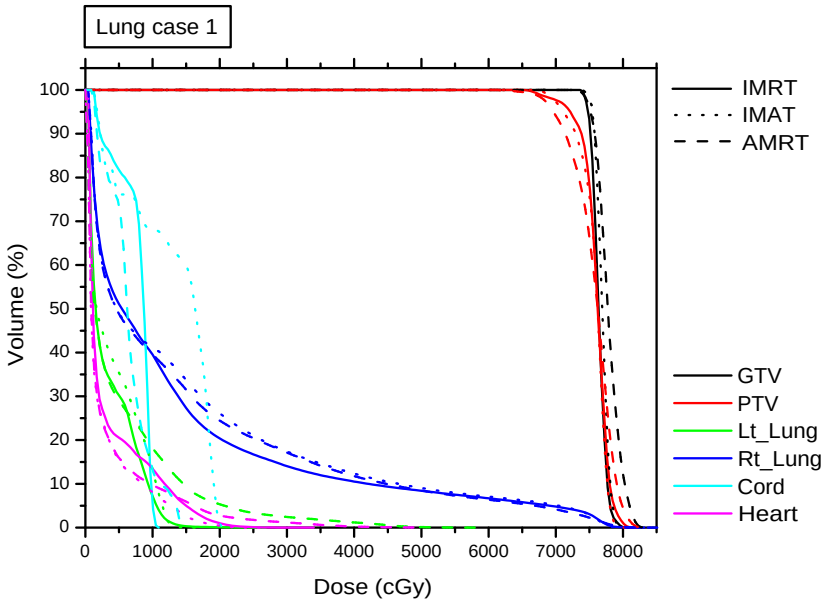
Case	ROI	Dose Index	IMRT	IMAT	AMRT
Lung 1	GTV	$V_{95\%}$	100%	100%	100%
	PTV	$V_{95\%}$	98.5%	99.3%	97.5%
		HI	0.086	0.091	0.135
	Left Lung	V_{20}	0.2%	0.0%	5.3%
	Right Lung	V_{20}	20.3%	26.2%	24.5%
Lung 2	GTV	$V_{95\%}$	100%	100%	100%
	PTV	$V_{95\%}$	97.8%	95.8%	96.2%
		HI	0.088	0.087	0.093
	Left Lung	V_{20}	0.3%	7.9%	10.9%
	Right Lung	V_{20}	34.7%	27.7%	29.1%
	Spinal Cord	Max	18.1 Gy	39.7 Gy	40.9 Gy
		1 cc	13.9 Gy	34.1 Gy	31.6 Gy
	Heart	V_{40}	8.0%	10.1%	10.4%
Lung 3	GTV	$V_{95\%}$	97.9%	100%	99.8%
	PTV	$V_{95\%}$	95.0%	99.8%	99.2%
		HI	0.202	0.103	0.146
	Left Lung	V_{20}	43.1%	36.1%	37.1%
	Right Lung	V_{20}	24.0%	18.2%	19.0%
	Spinal Cord	Max	53.7 Gy	51.9 Gy	64.6 Gy
		1 cc	50.4 Gy	48.8 Gy	55.5 Gy
	Heart	V_{40}	0.5%	0.6%	0.7%

Table 6.10: Plan properties of the lung cases.

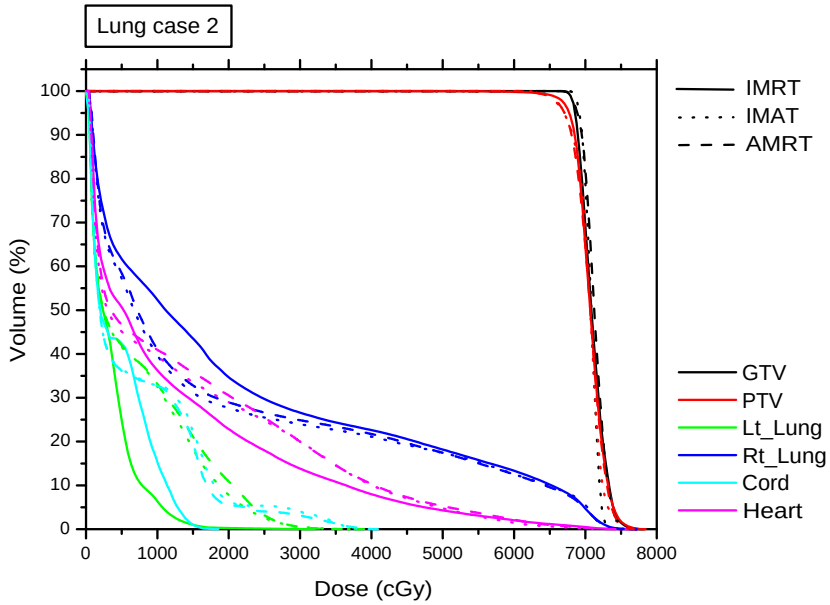
Case	Plan	# beams	# arcs	# apertures	Monitor Units
Lung 1	IMRT	7	-	53	347
	IMAT	36	5	180	547
	AMRT	180	1	180	665
Lung 2	IMRT	7	-	62	556
	IMAT	36	9	324	862
	AMRT	324	1	324	919
Lung 3	IMRT	7	-	104	1195
	IMAT	36	12	432	907
	AMRT	432	1	432	1325

Table 6.11: Estimated delivery times of the lung cases.

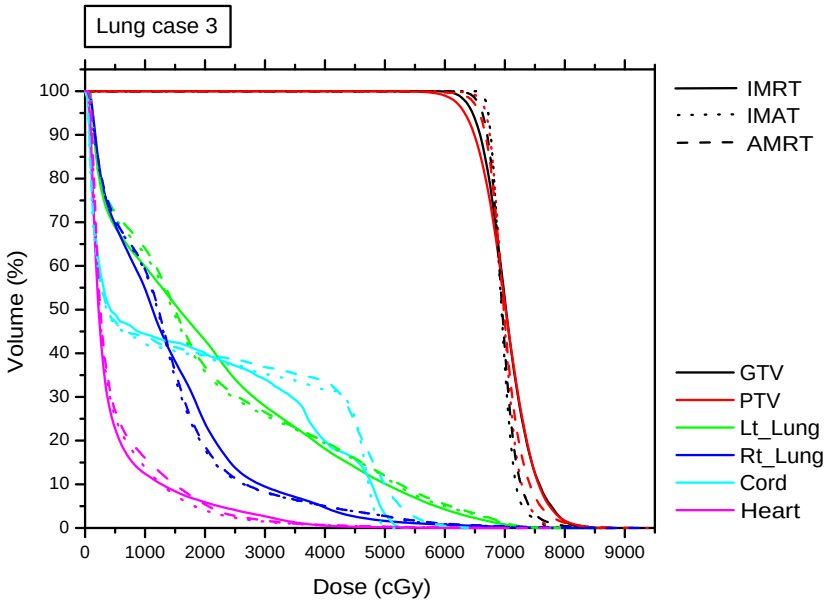
Case	Plan	Estimated Delivery Time (min)
Lung 1	IMRT	4.26
	IMAT	7.00
	AMRT	2.50
Lung 2	IMRT	6.91
	IMAT	13.0
	AMRT	3.60
Lung 3	IMRT	18.9
	IMAT	17.5
	AMRT	6.00



(a)

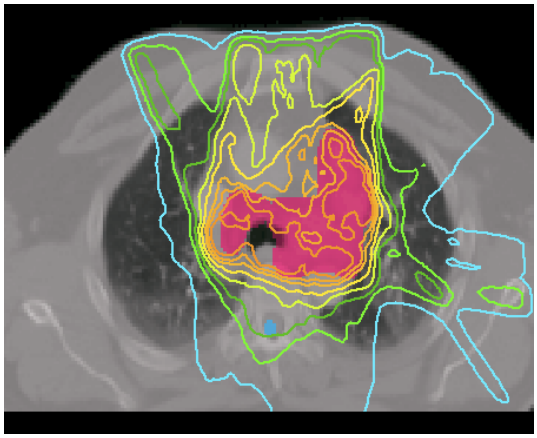


(b)

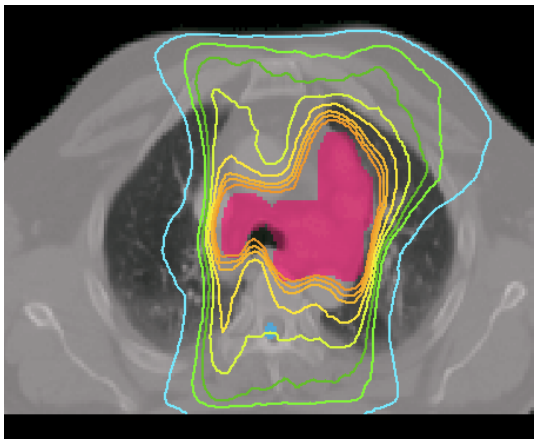


(c)

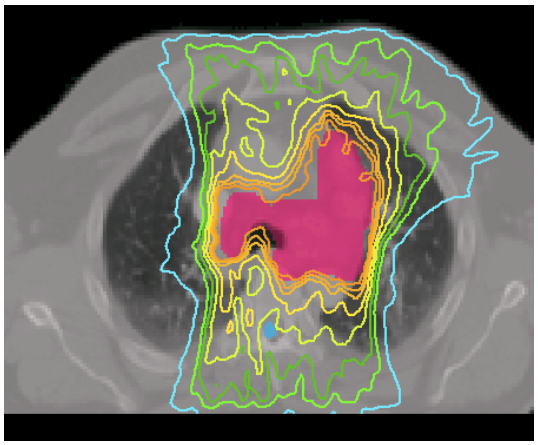
Figure 6.6: DVH comparisons between the IMRT, IMAT, and AMRT plans for (a) lung case 1, (b) lung case 2, and (c) lung case 3.



(a)



(b)



(c)

Figure 6.7: Isodose distributions of lung case 3 for (a) the IMRT plan, (b) the IMAT plan, and (c) the AMRT plan.

6.8 Plan comparison: prostate cases

Because all 3 prostate cases were very similar, they will be discussed as a group below. The plan summaries are tabulated in table 6.12 and the plan properties can be found in table 6.13.

In general, IMRT, IMAT, and AMRT provided very similar plan quality in terms of target coverage and normal tissue sparing in all the prostate cases. Due to the simplicity of the target (prostate only, no nodal volumes involved), only 5 arcs were used in IMAT and there were 180 beams in each of the AMRT plans. For IMAT, 100% target coverage was achieved in all cases and it has also provided the most uniform target dose distribution compared to IMRT and AMRT.

The dose delivered to bladder and rectum were also very similar between the three sets of plan but AMRT showed more effective normal tissue sparing while delivering slightly lower target dose compared to IMRT and IMAT. For example in prostate cases 2 and 3, AMRT delivered lower doses to the bladder and rectum, especially in the higher dose regions (see V_{75} of bladder and V_{70} of rectum in table 6.12). Overall, the total dose to the bladder was similar between the three techniques but AMRT was able to further reduce the total dose to the rectum.

For all prostate cases, AMRT only required 2.5 minutes for delivery on average while IMRT and IMAT required 4 minutes and 7 minutes, respectively (see table 6.14). Although AMRT showed some minor degradation in target coverage, the plans still fulfilled the clinical standard, proving AMRT is an efficient radiation treatment technique. However, unlike the previous three disease sites, IMAT did not outperform IMRT in terms of plan quality. In

addition, IMAT may take the longest time to deliver, making it the most inefficient treatment modality for prostate cases.

Table 6.12: Plan quality summary of the prostate cases.

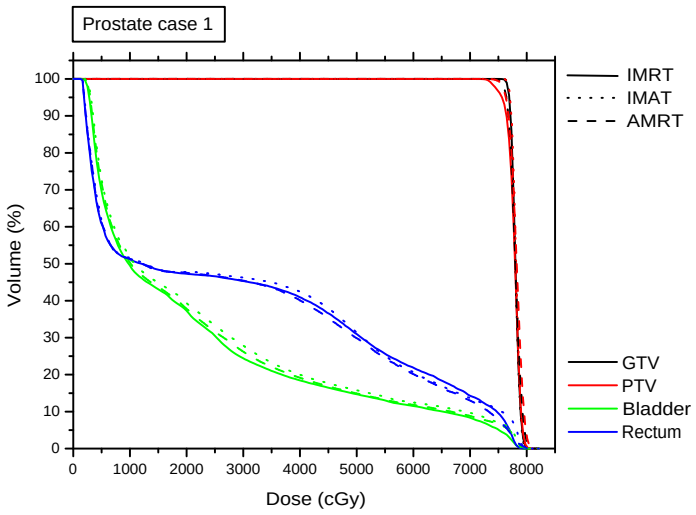
Case	ROI	Dose Index	IMRT	IMAT	AMRT
Prostate 1	GTV	$V_{95\%}$	100%	100%	100%
	PTV	$V_{95\%}$	98.3%	99.9%	100%
		HI	0.047	0.028	0.046
	Bladder	V_{65}	10.1%	11.3%	10.5%
		V_{75}	5.1%	7.1%	6.7%
	Rectum	V_{60}	21.9%	20.6%	20.2%
		V_{70}	14.4%	13.7%	12.9%
Prostate 2	GTV	$V_{95\%}$	100%	100%	99.9%
	PTV	$V_{95\%}$	100%	100%	97.7%
		HI	0.037	0.034	0.063
	Bladder	V_{65}	26.5%	27.1%	26.5%
		V_{75}	17.0%	19.7%	15.4%
	Rectum	V_{60}	22.5%	24.1%	20.6%
		V_{70}	15.6%	15.7%	13.8%
Prostate 3	GTV	$V_{95\%}$	100%	100%	100%
	PTV	$V_{95\%}$	100%	100%	97.3%
		HI	0.036	0.034	0.065
	Bladder	V_{65}	30.7%	31.5%	29.6%
		V_{75}	18.4%	19.8%	11.6%
	Rectum	V_{60}	18.5%	21.3%	16.6%
		V_{70}	8.9%	9.4%	7.9%

Table 6.13: Plan properties of the prostate cases.

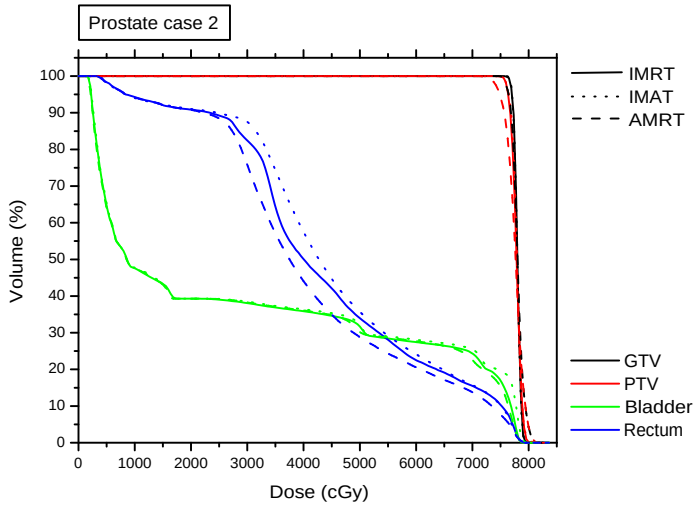
Case	Plan	# beams	# arcs	# apertures	Monitor Units
Prostate 1	IMRT	7	-	34	387
	IMAT	36	5	180	420
	AMRT	180	1	180	517
Prostate 2	IMRT	7	-	22	293
	IMAT	36	5	180	362
	AMRT	180	1	180	444
Prostate 3	IMRT	7	-	28	338
	IMAT	36	5	180	394
	AMRT	180	1	180	546

Table 6.14: Estimated delivery times of the prostate cases.

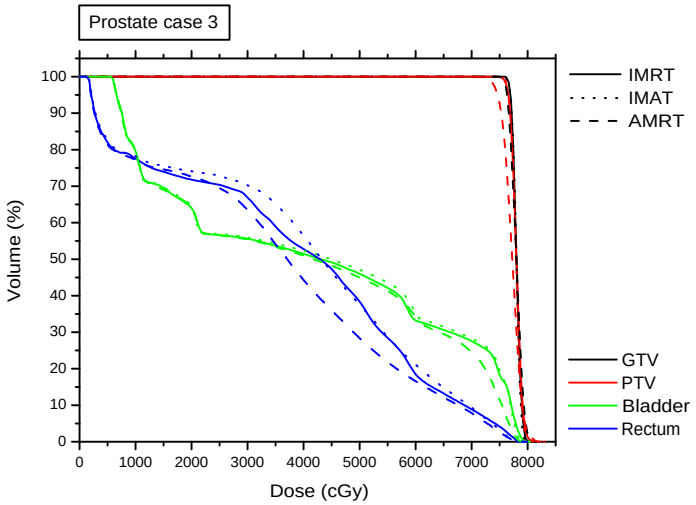
Case	Plan	Estimated Delivery Time (min)
Prostate 1	IMRT	4.63
	IMAT	7.00
	AMRT	2.50
Prostate 2	IMRT	3.36
	IMAT	7.00
	AMRT	2.50
Prostate 3	IMRT	3.96
	IMAT	7.00
	AMRT	2.50



(a)



(b)



(c)

Figure 6.8: DVH comparisons between the IMRT, IMAT, and AMRT plans for (a) prostate case 1, (b) prostate case 2, and (c) prostate case 3.

6.9 Summary

From 12 patient case studies, it was found that AMRT was capable of providing comparable dose distributions to IMRT while IMAT may obtain the most superior plan quality. Due to the full rotational arc arrangement, IMAT and AMRT may be more probable to produce the optimal dose distributions as IMRT may miss the optimal beam angles with only a few multiple static fields. On the other hand, both IMAT and AMRT cannot select the preferred beam angles manually as an attempt to minimise irradiation to normal tissues as in IMRT but partial arcs can be used. Nonetheless, from the plan comparisons, the full 360-degree arcs in AMRT and IMAT, in general, did not increase the dose to the surrounding critical organs although it was true that the lower dose volumes have increased. Furthermore, the number of MU in AMRT and IMAT has substantially increased compared to IMRT due to the increasing number of MLC apertures (see tables 6.4, 6.7, 6.10, and 6.13). The larger amount of apertures and MU may potentially increase the relative proportion of MLC transmission and scatter dose to the patient. In addition, the current version of AMRT algorithm does not control the minimum MLC opening area and the AMRT MLC sequences may obtain apertures with very small leaf openings. As a result, these small aperture openings may introduce residual dose to the plan and affect the target dose homogeneity as seen in figures 6.2 (c) and 6.7 (c).

IMAT obtained the best plan quality because the MLC segments were not as closely spaced as it was in AMRT, i.e. a more relaxed MLC constraint could be applied. This increased the flexibility for the aperture shape to vary between the neighbouring planning beams such that complex dose distribu-

tions can be achieved. With only one single arc in AMRT, a potential conflict may occur between the need to deliver the closest possible intensity distributions to the ideal sets and the connectivity of the MLC shapes. When such conflicts occur, the deliverability will take preference and the plan quality is subsequently compromised. This plan quality, however, was still potentially better or comparable to IMRT. The significance of AMRT is not to outperform IMAT or IMRT in terms of dose distributions but the potentially higher delivery efficiency. With the appropriate MLC movement constraint enforced during leaf sequencing, plans with more than 400 apertures can be delivered in under 5 minutes. For IMAT, despite of its superior plan quality, the delivery time is much longer as it is directly proportional to the number of arcs. Because the delivery time of AMRT is much shorter than IMAT, a second single arc can be added to compensate any residual dose coverage. However, a balance must be considered between quality and efficiency. For simple cases, such as prostate, AMRT may not be remarkably better than IMRT in terms of efficiency since majority of the treatment time would still be spent on the initial set-up and image guidance procedures that may take up to 15 minutes. The advantage of AMRT is more distinct in complex cases such as those in the HN region where AMRT is capable of delivering a comparable dose distribution to IMRT in under 5 minutes while IMRT may require a delivery time of 9 minutes.

In conclusion, from the plan comparisons, it was found that AMRT was able to achieve a comparable target coverage to IMAT and IMRT; but in a few cases, AMRT showed greater reduction in normal tissue dose at the cost of less uniform dose distribution. However, the delivery time of AMRT was the shortest compared to IMAT and IMRT, proving it is a clinically efficient

treatment technique. In addition to this preliminary study, AMRT also deserves further investigations on its dosimetric capability using a larger amount of patient samples of the individual disease sites such that more statistically robust conclusions can be drawn.

Chapter 7

A brief overview on single-arc IMAT applications in clinic

The privilege of single-arc IMAT is its superior delivery time which is a very attractive feature to the clinic in terms of patient flow. Currently, AMRT is available for research purposes only but several commercial single-arc products are available for clinical implementation such as Elekta volumetric-modulated arc therapy (VMAT) and Varian RapidArc^{TM}. Elekta provides several different single-arc IMAT planning methods using Ergo++ (Elekta/3DLine, Milan, Italy). One of these planning algorithms uses the 2-step optimisation method. The single delivery arc is first approximated by a series of static planning beam angles and a step-and-shoot IMRT plan is generated based on these static beams. Similar to the previous method described in Chapter 5, the stacked apertures over a given angle is collapsed into a single arc such that the IMRT plan is translated into a VMAT plan by distributing the superimposed segments into the angular interval between the planning beams. Another VMAT planning method uses a semi-forward planning approach where the MLC apertures are defined by the target anatomy, i.e. BEV conformations, and the weighting of these apertures are optimised freely with varying MU.

Different from the Elekta VMAT algorithms, Varian RapidArc employs the

1-step optimisation approach [Otto 2008]. RapidArc adapts the DAO method but with a novel progressive sampling scheme. Demonstrated by Earl *et al*, IMAT planning with DAO is restrained by the initial MLC aperture shapes [Earl *et al* 2003]. In DAO, for a series of static beam angles, the optimisation is initiated by a set of arbitrary MLC apertures that conform to the target. A MLC displacement constraint is applied to these MLC apertures to ensure deliverability. Based on equation 3.1, the maximum MLC displacement allowed is governed by the angular spacing of the static planning beams and the mechanical speed limits of the MLC and the gantry. In clinical practice, the maximum mechanical MLC displacement allowed between planning beams spaced at every 10° is approximately 5 cm (with the maximum speeds of MLC and gantry of 2.5 cms^{-1} and 4.8 $degs^{-1}$, respectively); but with the consideration of dosimetric effects arising from large aperture shape variation as discussed in Chapter 4, the MLC constraint applied to planning is reduced to approximately 3 cm. Because of this stringent constraint, the MLC movement is over-limited such that the aperture shapes are very similar between the adjacent planning beams. Precisely speaking, the optimisation freedom is over-restrained by the MLC constraint which may affect the resultant plan quality. Otto solved this problem by employing a progressive sampling scheme for DAO, where the number of beams used to initialise the optimisation is reduced to only 8 to 10 beam angles. With fewer beams, the MLC constraint becomes more relaxed and hence the optimisation freedom is less restricted, giving more opportunities for the MLC shapes to change in the initial stage of the optimisation. The progressive sampling scheme is composed of 5 different resolution levels. As the optimisation progresses and the resolution level transits from one to the next, the number of beam increases and the

MLC displacement constraint becomes more tightened. Using such progressive sampling scheme for DAO, a single-arc IMAT plan can be optimised with sufficient optimisation freedom while benefiting from the efficiency of a 1-step optimisation method.

7.1 In comparison with IMRT and tomotherapy

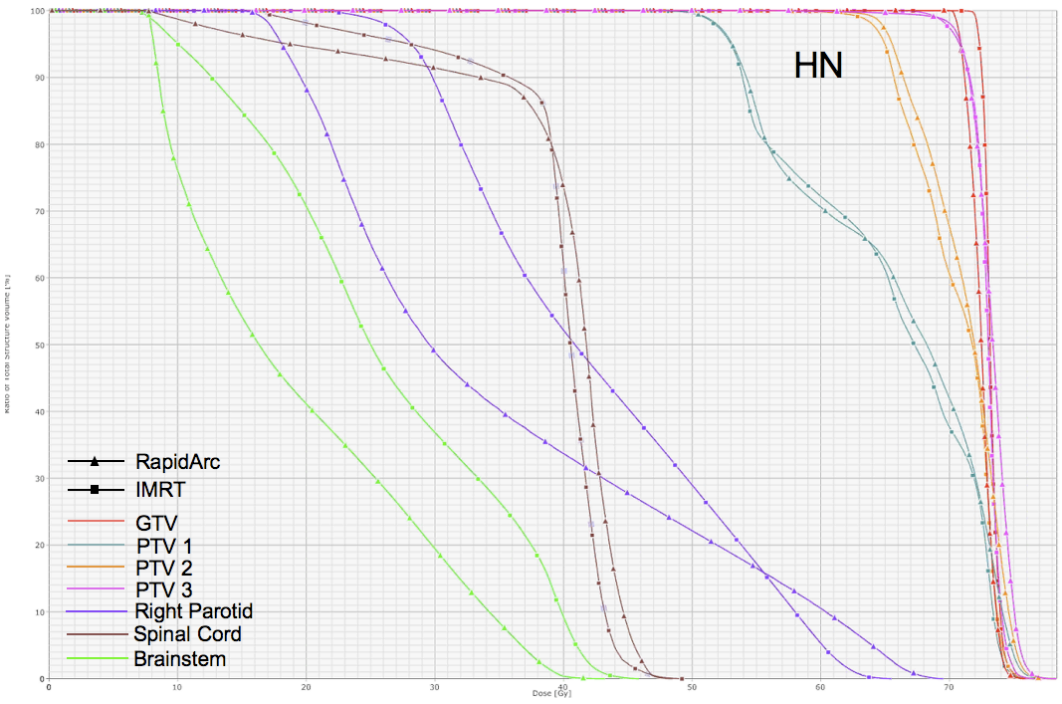
Compared to other clinically-established treatment techniques such as multi-field IMRT and helical tomotherapy, these new clinical single-arc implementations have shown comparable plan quality with improved delivery efficiency [Bortfeld & Webb 2009][Tang *et al* 2009a][Rong *et al* 2009]. Figures 7.1 and 7.2 display the DVH and isodose comparisons of a RapidArc plan and an IMRT plan[1] generated with Pinnacle^3 for a HN case and a brain case, respectively. For convenience, the dose of the IMRT plans in Pinnacle^3 were exported as digital imaging and communications in medicine (DICOM) files and were imported to Eclipse for plan comparison with RapidArc. The overall target coverage in the RapidArc plans were similar to that in the IMRT plans as expected while the dose to the normal tissues were further reduced in RapidArc. This observation could also be seen in several other cases in different disease sites for 2 HN, 3 brain, 2 lung, and prostate 2 cases. A summary of the comparison results is tabulated in table 7.1. Note that the RapidArc plans and the IMRT plans were generated separately by two different planners with the same planning goals. For a fair study, both sets of plans were locked until they were ready for comparison analyses.

Similar to the RapidArc and IMRT comparison study at the University

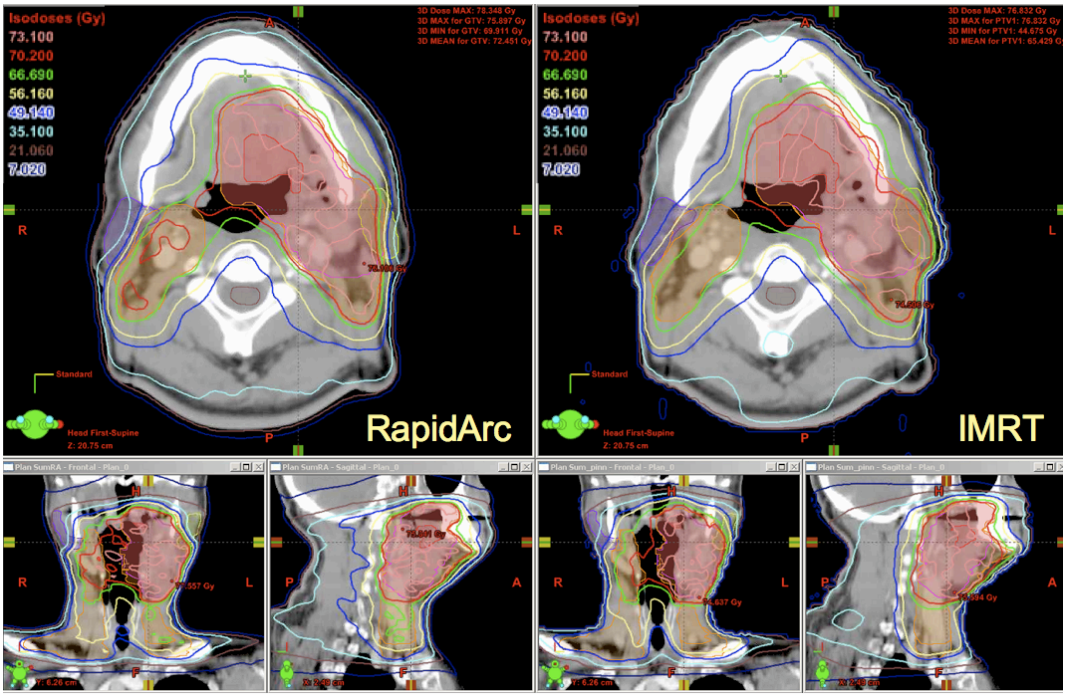
[1] All IMRT plans were generated by the dosimetry group of the Department of Radiation Oncology in the University of Maryland.

of Maryland, a plan quality comparison study for RapidArc and helical tomotherapy was performed with the collaboration of University of Maryland and University of Wisconsin. A total of 16 patients were selected for this retrospective study including 4 HN, 4 brain, 4 lung and 4 prostate patients. Each institution was responsible for providing 2 sets of patient data from each disease site. For each patient the RapidArc and tomotherapy[2] plans were generated with the same planning goals. The quality of the plans were compared as shown in table 7.2. In general, the dose distributions between RapidArc and tomotherapy plans were fairly similar but RapidArc tended to produce plans with slightly higher maximum dose as demonstrated in figure 7.3. The dose in the target was slightly less homogeneous in the RapidArc plans as compared to tomotherapy. For the OARs that were located parallel to the target received lower doses in the tomotherapy plans, while for those that were located superior or inferior to the target received higher doses. This was a limitation of the jaw size used in tomotherapy and a jaw width of 2.5 cm was used in all cases in this study. With a finer jaw width, the resolution would be improved but the delivery time may be lengthened. In addition, the actual BOT of the plans were recorded for all the plans and are tabulated in table 7.3. The overall delivery time is much lower in RapidArc while the widest jaw width was already used for the tomotherapy plans.

[2]All tomotherapy plans were generated by Dr. Yi Rong from the University of Wisconsin Cancer Center, Riverview.

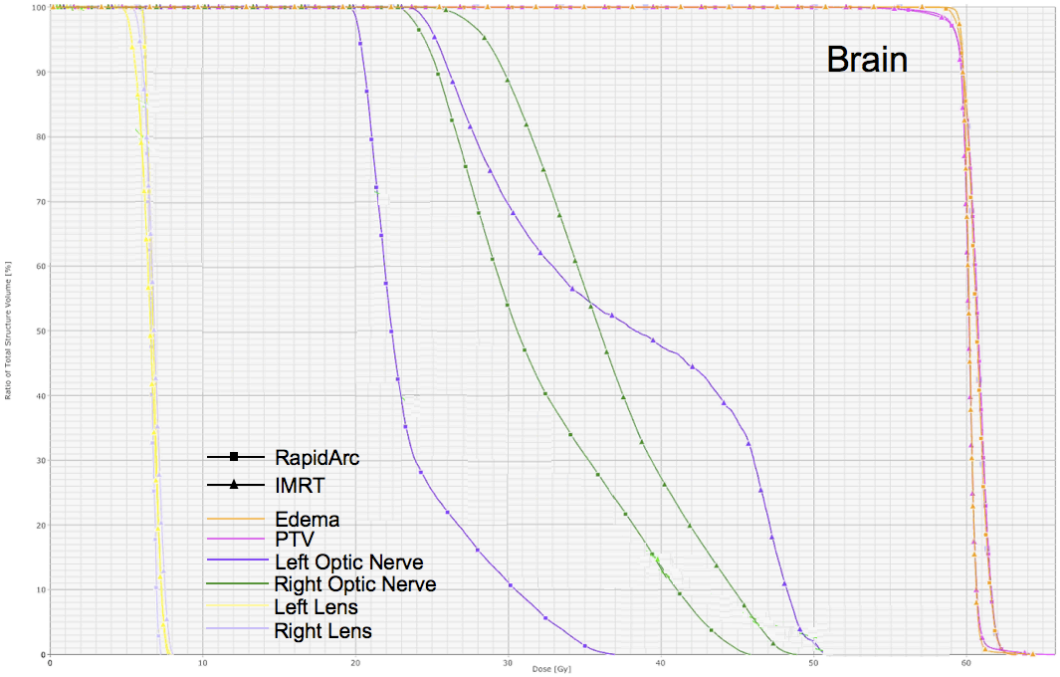


(a)

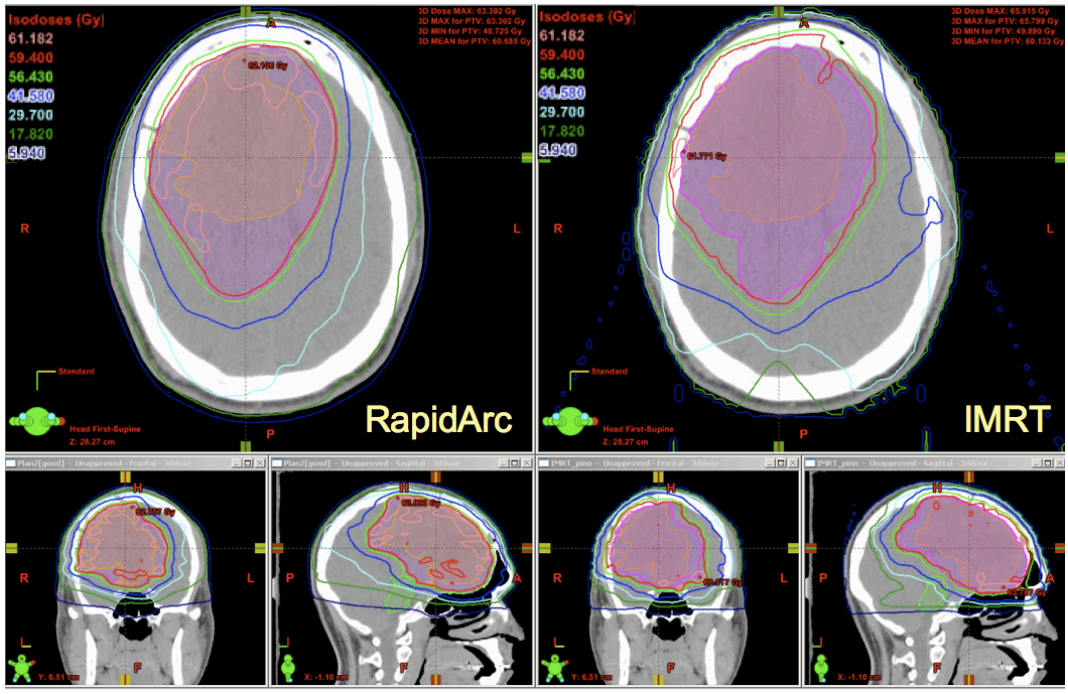


(b)

Figure 7.1: Comparison of RapidArc and IMRT for a HN case in (a) DVH and (b) isodose distributions.



(a)



(b)

Figure 7.2: Comparison of RapidArc and IMRT for a brain case in (a) DVH and (b) isodose distributions.

Table 7.1: Summary of the plan quality comparison between RapidArc and IMRT for 2 HN, 3 brain, 2 lung and 2 prostate patients. All plans were normalised at the mean dose of the PTV for comparison purposes. The dose uniformity of the PTV is described by the homogeneity index, where $HI = (D_5 - D_{95})/D_{mean}$.

Case	ROI	Dose Index	RapidArc	IMRT
1 (HN)	GTV	D_{95}	100%	98.0%
	PTV	D_{95}	98.4%	97.0%
		HI	0.078	0.090
	Left Parotid	Mean	19.0 Gy	25.2 Gy
	Right Parotid	Mean	10.9 Gy	9.27 Gy
2 (HN)	GTV	D_{95}	98.8%	100%
	PTV	D_{95}	98.6%	97.9%
		HI	0.052	0.050
	Right Parotid	Mean	34.7 Gy	41.1 Gy
	Brainstem	Max	39.8 Gy	33.9 Gy
	Spinal Cord	Max	45.7 Gy	43.0 Gy
3 (Brain)	GTV	D_{95}	98.7%	100%
	PTV	D_{95}	98.1%	98.5%
		HI	0.072	0.059
	Left Lens	Max	8.88 Gy	9.56 Gy
	Right Lens	Max	8.50 Gy	6.47 Gy

Continued on next page

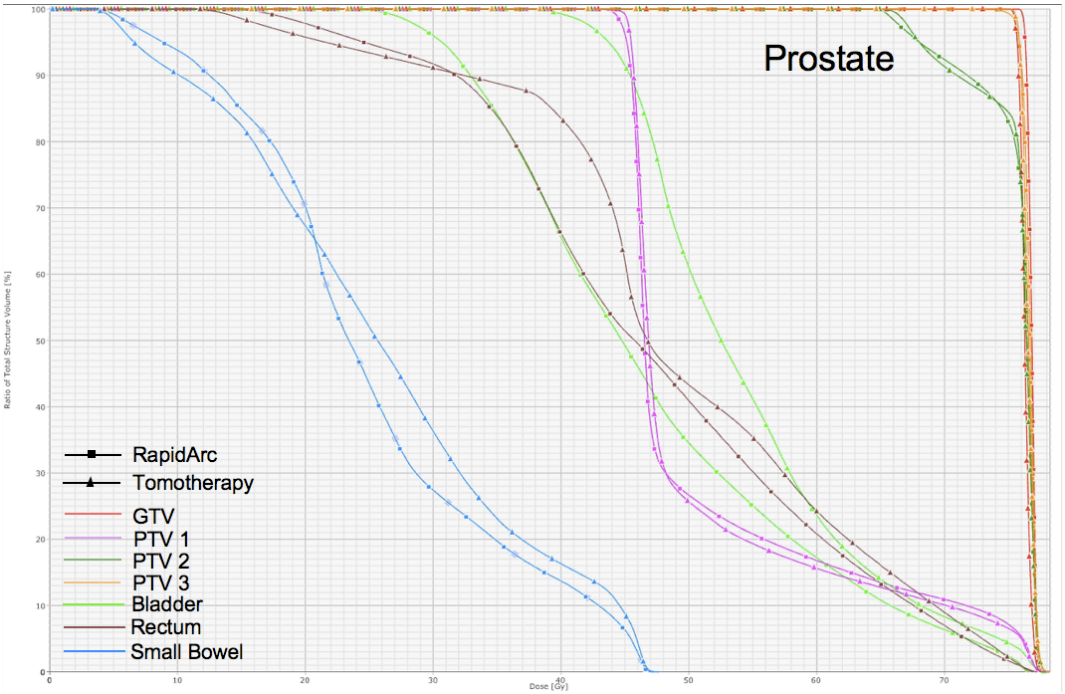
Table 7.1 – continued from previous page

Case	ROI	Dose Index	RapidArc	IMRT
4 (Brain)	GTV	D_{95}	100%	99.6%
	PTV	D_{95}	97.8%	97.4%
		HI	0.072	0.075
	Left Cochlea	Max	37.8 Gy	39.1 Gy
	Left Eye	Max	23.4 Gy	36.1 Gy
	Right Eye	Max	44.1 Gy	28.7 Gy
5 (Brain)	PTV	D_{95}	98.9%	99.9%
		HI	0.040	0.024
	Left Optic Nerve	Max	36.6 Gy	50.8 Gy
	Right Optic Nerve	Max	45.4 Gy	48.9 Gy
	Brainstem	Max	50.7 Gy	48.8 Gy
6 (Lung)	GTV	D_{95}	97.8%	100%
	PTV	D_{95}	95.3%	98.0%
		HI	0.108	0.047
	Left Lung	V_5	2.14%	14.1%
	Right Lung	V_{20}	13.7%	16.1%
		V_5	57.1%	63.6%
7 (Lung)	GTV	D_{95}	100%	97.3%
	PTV	D_{95}	96.1%	93.5%
		HI	0.080	0.141
	Left Lung	V_{20}	32.9%	41.1%
		V_5	69.2%	64.6%
	Right Lung	V_{20}	26.4%	18.1%

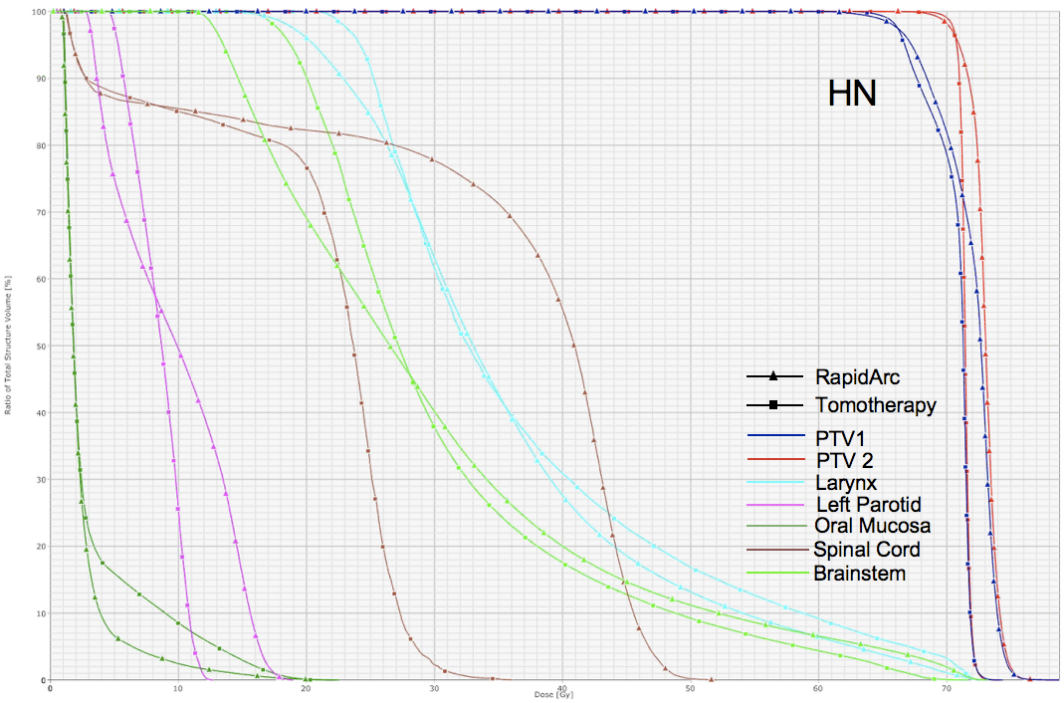
Continued on next page

Table 7.1 – continued from previous page

Case	ROI	Dose Index	RapidArc	IMRT
		V_5	63.6%	63.0%
	Spinal Cord	Max	44.3 Gy	47.2 Gy
8 (Prostate)	GTV	D_{95}	97.7%	100%
	PTV	D_{95}	97.6%	98.1%
		HI	0.037	0.027
	Bladder	D_{35}	13.2 Gy	34.2 Gy
		D_{15}	63.4 Gy	69.0 Gy
	Rectum	D_{35}	51.7 Gy	49.6 Gy
		D_{15}	65.7 Gy	69.0 Gy
9 (Prostate)	GTV	D_{95}	98.8%	100%
	PTV	D_{95}	98.4%	98.5%
		HI	0.034	0.027
	Bladder	D_{35}	3.83 Gy	7.28 Gy
		D_{15}	13.9 Gy	19.9 Gy
	Rectum	D_{35}	13.1 Gy	18.9 Gy
		D_{15}	31.8 Gy	33.2 Gy



(a)



(b)

Figure 7.3: DVH comparisons of RapidArc and tomotherapy of (a) a prostate case and (b) a HN case.

Table 7.2: Summary of the plan quality comparison between RapidArc and tomotherapy for 4 HN, 4 brain, 4 lung and 4 prostate patients. All plans were normalised at D_{95} of the PTV to the 100% of the prescription dose for comparison purposes. Apart from the homogeneity index where $HI = (D_2 - D_{98})/D_{pres}$, the dose conformality of the PTV is further described by D_{max}/D_{pres}, which indicates the fraction of excessive dose delivered to the target (D_{max} is the maximum dose in the target and D_{pres} is the prescription dose).

Case	ROI	Dose Index	RapidArc	Tomotherapy
HN A	PTV	D_{max}/D_{pres}	1.12	1.09
		HI	1.120	1.043
	Right Parotid	Mean	18.6 Gy	9.0 Gy
	Left Optic Nerve	Max	4.1 Gy	27.7 Gy
	Optic Chiasm	Max	13.0 Gy	17.8 Gy
HN B	PTV	D_{max}/D_{pres}	1.11	1.06
		HI	1.071	1.033
	Left Parotid	Mean	9.7 Gy	8.5 Gy
	Spinal Cord	Max	52 Gy	36.1 Gy
		1 cc	46.7 Gy	28.59 Gy
HN C	PTV	D_{max}/D_{pres}	1.09	1.04
		HI	1.092	1.045

Continued on next page

Table 7.2 – continued from previous page

Case	ROI	Dose Index	RapidArc	Tomotherapy
	Left Parotid	Mean	18.4 Gy	12.6 Gy
	Right Parotid	Mean	12.2 Gy	8.1 Gy
	Larynx	Mean	4.8 Gy	21.1 Gy
	Spinal Cord	Max	36.6 Gy	31.5 Gy
		1 cc	32.6 Gy	27.3 Gy
HN D	PTV	D_{max}/D_{pres}	1.08	1.06
		HI	1.069	1.061
	Eyes	Max	25.8 Gy	13.9 Gy
	Lenses	Max	11.0 Gy	5.4 Gy
	Left Optic Nerve	Max	4.5 Gy	33.8 Gy
	Right Optic Nerve	Max	18.4 Gy	38.0 Gy
	Right Parotid	Mean	39.7 Gy	14.9 Gy
	Spinal Cord	Max	44.0 Gy	40.8 Gy
		1 cc	41.3 Gy	36.1 Gy
Brain A	PTV	D_{max}/D_{pres}	1.13	1.07
		HI	1.108	1.073
	Left Eye	Max	37.0 Gy	20.7 Gy
	Right Eye	Max	19.0 Gy	18.4 Gy
	Left Lens	Max	9.1 Gy	6.9 Gy
	Right Lens	Max	9.0 Gy	5.8 Gy
	Optic Nerves	Max	39.8 Gy	33.8 Gy
Brain B	PTV	D_{max}/D_{pres}	1.06	1.03
		HI	1.048	1.030

Continued on next page

Table 7.2 – continued from previous page

Case	ROI	Dose Index	RapidArc	Tomotherapy
	Left Eye	Max	2.6 Gy	5.7 Gy
	Right Eye	Max	2.5 Gy	5.6 Gy
	Optic Nerves	Max	9.1 Gy	9.1 Gy
Brain C	PTV	D_{max}/D_{pres}	1.06	1.05
		HI	1.065	1.053
	Left Eye	Max	18.1 Gy	15.5 Gy
	Right Eye	Max	26.7 Gy	20.4 Gy
	Left Lens	Max	7.8 Gy	3.7 Gy
	Right Lens	Max	8.0 Gy	3.8 Gy
Brain D	PTV	D_{max}/D_{pres}	1.06	1.05
		HI	1.062	1.058
	Eyes	Max	11.0 Gy	6.3 Gy
	Lenses	Max	5.6 Gy	1.9 Gy
	Left Optic Nerve	Max	13.0 Gy	9.6 Gy
	Right Optic Nerve	Max	15.4 Gy	17.2 Gy
Lung A	PTV	D_{max}/D_{pres}	1.08	1.05
		HI	1.072	1.057
	Spinal Cord	Max	23.9 Gy	11.5 Gy
		1 cc	16.9 Gy	8.66 Gy
	Right Lung	V_{20}	23.7%	21.7%
	Esophagus	Mean	10.6 Gy	6.0 Gy
Lung B	PTV	D_{max}/D_{pres}	1.07	1.04
		HI	1.076	1.040

Continued on next page

Table 7.2 – continued from previous page

Case	ROI	Dose Index	RapidArc	Tomotherapy
	Spinal Cord	Max	31.0 Gy	29.8 Gy
		1 cc	24.4 Gy	28.0 Gy
	Heart	Mean	10.9 Gy	10.6 Gy
	Total Lungs	V_{20}	34.7%	51.2%
	Esophagus	Mean	15.4 Gy	17.9 Gy
Lung C	PTV	D_{max}/D_{pres}	1.10	1.05
		HI	1.078	1.053
	Spinal Cord	Max	19.1 Gy	14.1 Gy
		1 cc	17.1 Gy	13.2 Gy
	Total Lungs	V_{20}	8.9%	9.4%
	Esophagus	Mean	7.6 Gy	10.6 Gy
Lung D	PTV	D_{max}/D_{pres}	1.10	1.06
		HI	1.087	1.053
	Spinal Cord	Max	31.0 Gy	29.7 Gy
		1 cc	27.9 Gy	29.0 Gy
	Heart	Mean	15.9 Gy	14.2 Gy
	Total Lungs	V_{20}	24.7%	25.1%
	Esophagus	Mean	16.1 Gy	17.9 Gy
Prostate A	PTV	D_{max}/D_{pres}	1.05	1.05
		HI	1.046	1.049
	Bladder	D_{30}	8.6 Gy	19.8 Gy
	Bladder	D_{15}	22.3 Gy	33.5 Gy
	Rectum	D_{30}	31.6 Gy	25.1 Gy

Continued on next page

Table 7.2 – continued from previous page

Case	ROI	Dose Index	RapidArc	Tomotherapy
	Rectum	D_{15}	51.7 Gy	38.5 Gy
Prostate B	PTV	D_{max}/D_{pres}	1.10	1.04
		HI	1.071	1.040
	Bladder	D_{30}	68.5 Gy	69.7 Gy
	Bladder	D_{15}	75.0 Gy	73.5 Gy
	Rectum	D_{30}	67.0 Gy	59.0 Gy
	Rectum	D_{15}	71.4 Gy	68.0 Gy
Prostate C	PTV	D_{max}/D_{pres}	1.10	1.04
		HI	1.084	1.038
	Bladder	D_{30}	16.1 Gy	38.0 Gy
	Bladder	D_{15}	44.3 Gy	55.0 Gy
	Rectum	D_{30}	41.3 Gy	34.5 Gy
	Rectum	D_{15}	52.5 Gy	53.7 Gy
Prostate D	PTV	D_{max}/D_{pres}	1.10	1.05
		HI	1.083	1.052
	Bladder	D_{30}	28.2 Gy	33.0 Gy
	Bladder	D_{15}	58.8 Gy	60.3 Gy
	Rectum	D_{30}	47.7 Gy	34.5 Gy
	Rectum	D_{15}	58.5 Gy	59.0 Gy

Table 7.3: Comparison of the total BOT of the RapidArc and tomotherapy plans.

Case	RapidArc	Tomotherapy
HN A	1.25 min	3.22 min
HN B	2.47 min	11.6 min
HN C	1.25 min	4.74 min
HN D	2.47 min	9.16 min
Brain A	0.57 min	5.10 min
Brain B	1.23 min	4.25 min
Brain C	1.91 min	7.95 min
Brain D	1.26 min	4.56 min
Lung A	1.25 min	3.42 min
Lung B	1.28 min	3.27 min
Lung C	1.28 min	3.96 min
Lung D	1.27 min	4.26 min
Prostate A	1.28 min	2.72 min
Prostate B	3.74 min	10.3 min
Prostate C	1.38 min	5.96 min
Prostate D	1.69 min	5.32 min

7.2 Clinical QA procedures for single-arc IMAT

Because single-arc IMAT delivery requires simultaneous variations of MLC speed, gantry speed and dose rate, the mechanical accuracy of the linac may demand more intensive care and quality control [Ling *et al* 2008]. In addition to the typical QA procedures such as MLC position accuracy checks, gantry rotation and dose-rate variation accuracy checks may be included for machine-specific QA for single-arc IMAT. The accuracy of MLC positions during gantry rotation with different dose rates should be monitored.

Patient-specific QA procedures for single-arc IMAT have been explored by several groups using different detectors such as EPID and 2D ion chamber arrays [Van Esch *et al* 2007][Nicolini *et al* 2008][Korreman *et al* 2009]

[Létourneau *et al* 2009][Cao *et al* 2009][Yi *et al* 2009]. At the University of Maryland, a 2D diode array (MapCHECK^{TM}, Sun Nuclear Corporation, Melbourne, FL, USA) is used for RapidArc patient-specific QA [Tang & Yi 2009]. Apart from the traditional approach using film measurements, MapCHECK has been widely used in the clinic for IMRT patient-specific QA [Jursinic & Nelms 2003][Létourneau *et al* 2004]. For rotational dosimetry purposes, MapCHECK is inserted into a water-equivalent phantom for QA measurements (MapPHAN^{TM}, Sun Nuclear Corporation, Melbourne, FL, USA) (see figure 7.4). Essential detector calibrations are conducted using the MapCHECK software so that the derived correction factors can be applied for relative and absolute dose measurements. The MapCHECK software also provides online data analysis using profile measurements and gamma analysis for patient-specific QA.

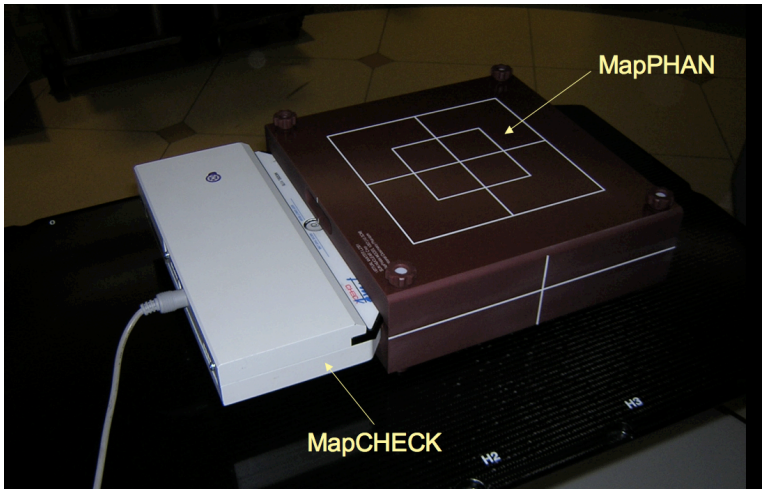


Figure 7.4: MapCHECK and MapPHAN setup for rotational therapy QA.

Chapter 8

Alternative planning and delivery approach of single-arc IMAT

8.1 Background and objectives

Since its emergence, single-arc IMAT has gained interests worldwide. The number of clinics realising such technique is continuously growing. In particular, the RapidArc^{TM} product developed by Varian has been widely implemented in the world. For RapidArc, a plan is optimised with variable MLC aperture shapes, aperture MU weightings, and gantry speeds. Dose-rate variation is a key component in the requirements of RapidArc delivery as each of the evenly-spaced MLC segment within the arc can have different MU weightings, while the gantry speed variation is kept minimal. In short, to effectuate RapidArc treatments, a clinic should either 1) purchase new machines which are enabled with continuous dose-rate variation during rotational delivery, or 2) upgrade the existing machine for dose-rate variation. Either option demands funding support which is often limited in most of the clinics. Although the latter option requires a lower budget, the upgrade is available to the newer linac models only, e.g. the Varian Triology EX or iX machines. To date, the manufacturer does not support RapidArc upgrade on the previous machine models. These older machines can perform rotational delivery but dose rate

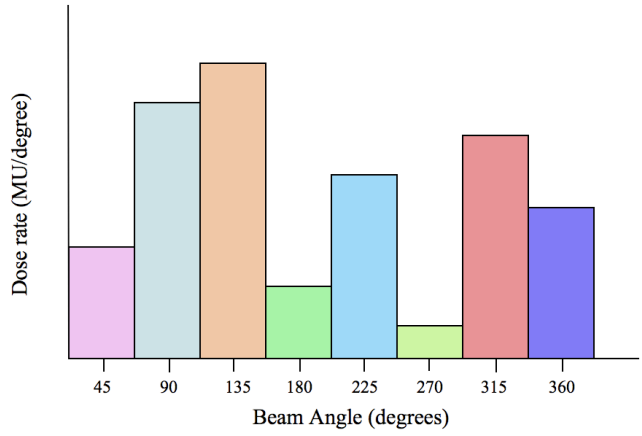
cannot be dynamically varied. As a result, clinics are restricted from using their existing resources for the application of the latest radiotherapy technique. Thus, the potential benefits of single-arc IMAT to both the clinic and patients will be delayed until the hardware requirement is fulfilled.

While some commercial products such as Philips SmartArc^{TM} (Philips Medical, Madison, WI, USA) offers single-arc IMAT treatments with constant dose-rate(CDR), Palma *et al* had concluded that single-arc plans optimised with variable dose-rate(VDR) provide superior plan quality to those optimised with CDR [Palma *et al* 2008]. Note that the treatment planning method used in the study by Palma *et al* and in the commercial products follow the traditional planning approach which utilises a series of evenly spaced static beams. The regular spacing of these static planning beam angles imposes a limitation on the optimisation freedom and delivery approach such that beams with variable MU weightings require VDR delivery or beams with constant MU weightings require CDR delivery. To dissociate from this restriction, an alternative planning and delivery approach for single-arc IMAT was developed [Tang *et al* 2009b]. CDR-deliverable plans were created by modulating the angular spacing of the beams or apertures of the VDR single-arc plans while the MU weighting of these apertures remain variable.

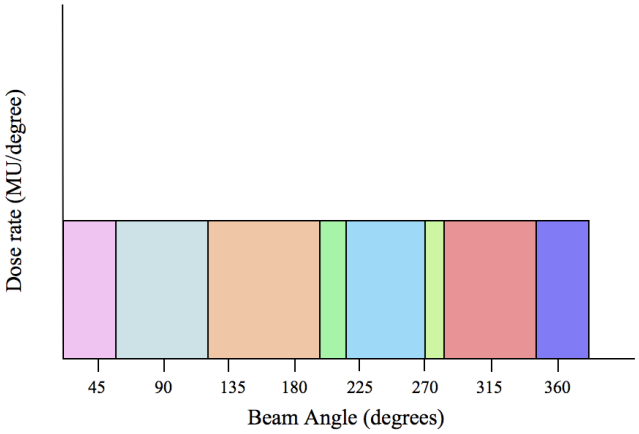
8.2 The concept of amplitude modulation *vs.* frequency modulation

In Chapter 5, it has been demonstrated that displacing the apertures from their planned angles to a slightly different position may result in only minimal dosimetric errors. Several other publications have also concluded the

same finding and have adapted it for single-arc IMAT planning where the stacked apertures are optimised at sparsely spaced static beam angles and re-distributed to the angular interval between the adjacent planning beams into a single-arc IMAT sequence[Crooks *et al* 2003][Cameron 2005][Wang *et al* 2008]. Based on this observation, it is conceivable that the plan quality between an evenly-spaced VDR-optimised plan (where apertures have variable MU weightings) and the corresponding CDR plan is equivalent, as the apertures in the CDR plan is spaced using a constant MU per degree. This is analogous to amplitude modulation (AM) and frequency modulation (FM) in radio-broadcasting as illustrated in figure 8.1, where the dose rate profile in MU per degree is plotted for all beam angles of a simple case. Figure 8.1 (a) represents an evenly-spaced VDR-optimised plan, where the beam angles are evenly distributed within the arc and the dose rate used for each beam is different as it is needed to deliver the beams with different MU weightings at a constant gantry rotation. To deliver the same plan with CDR, the apertures are spaced with a different angular interval that is proportional to their individual weightings using a constant MU per degree. As a result, the dose rate is constant throughout the delivery but the angular occupation of the apertures are varied as shown in figure 8.1 (b). Note that each bar dictates the MU weighting that is designated to the individual apertures. These bars have the same width but different heights in the VDR plan as in AM radio-broadcasting (figure 8.1 (a)), while the bars have the same height but different widths in the corresponding CDR plan as in FM radio-broadcasting (figure 8.1 (b)).



(a)



(b)

Figure 8.1: Dose rate profiles of (a) an evenly-spaced VDR plan (AM) and (b) the corresponding unevenly-spaced CDR plan (FM).

8.2.1 Proof of concept using RapidArc

To prove the hypothesis that the plan quality of a CDR plan with variable angular spacing is equivalent to VDR plan with regular beam spacing, VDR RapidArc plans of 5 patient cases including 3 HN, 1 brain, and 1 prostate were converted to CDR plans. The RapidArc plans were generated using the Varian Eclipse TPS where the MLC positions, MU weightings, and gantry speed were optimised. Since each of the beams had different MU weightings and required VDR delivery, these original RapidArc plans were referred to as VDR plans.

After the VDR plans were created, the DICOM plan file that contain all the plan information such as gantry angle of the beams, MLC aperture positions and MU weightings, was exported from Eclipse for all the cases. For the ease of process, these DICOM files were converted into ASCII files using a free open source (`www.dvtk.org`). The resultant ASCII files that contain the VDR plan information were then imported to an in-house program written in C for the conversion into CDR plans (see appendix G). The conversion program generated new sets of MLC sequence, which were imported back to Eclipse for final dose calculation and plan evaluation. The CDR plans are now readily deliverable with a conventional linac that is not VDR-enabled.

The CDR conversion principle was to redistribute the evenly-spaced beams in the original VDR plan according to their individual MU weightings in proportional to the total MU of the entire single-arc. The angular interval occupation of these re-spaced beams was governed by the spacing factor s, which was defined by the total MU and the total arc range of the treatment arc Θ with N beams,

$$s(\mathrm{MUdeg}^{-1}) = \frac{\sum_{i=1}^{N} MU_i}{\Theta(\mathrm{deg})}, \qquad \text{where } i = 1, 2, 3, ..., N. \tag{8.1}$$

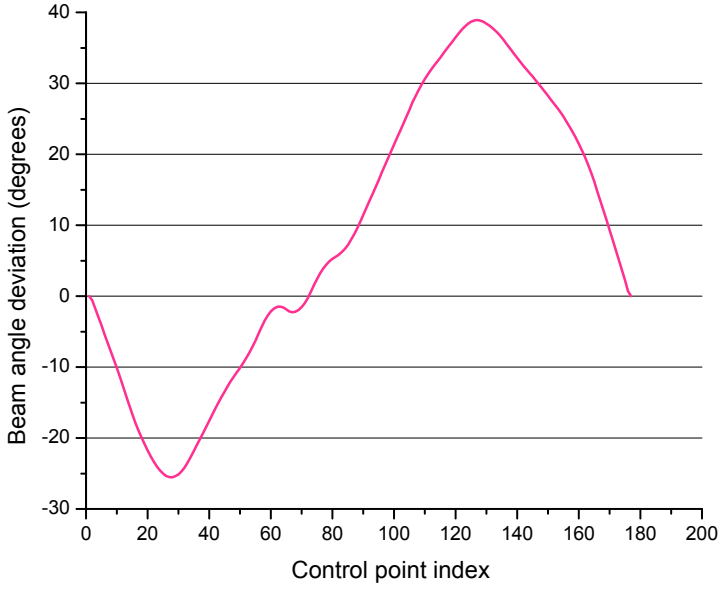
With a constant value of MU per gantry angle, the beams in the VDR plans were relocated with an irregular spacing of $\Delta\theta_i$ such that the beams with higher MU weightings occupied a larger angular interval and vice versa,

$$\Delta\theta_i(\mathrm{deg}) = \frac{\mathrm{MU}_i}{s(\mathrm{MUdeg}^{-1})}. \tag{8.2}$$

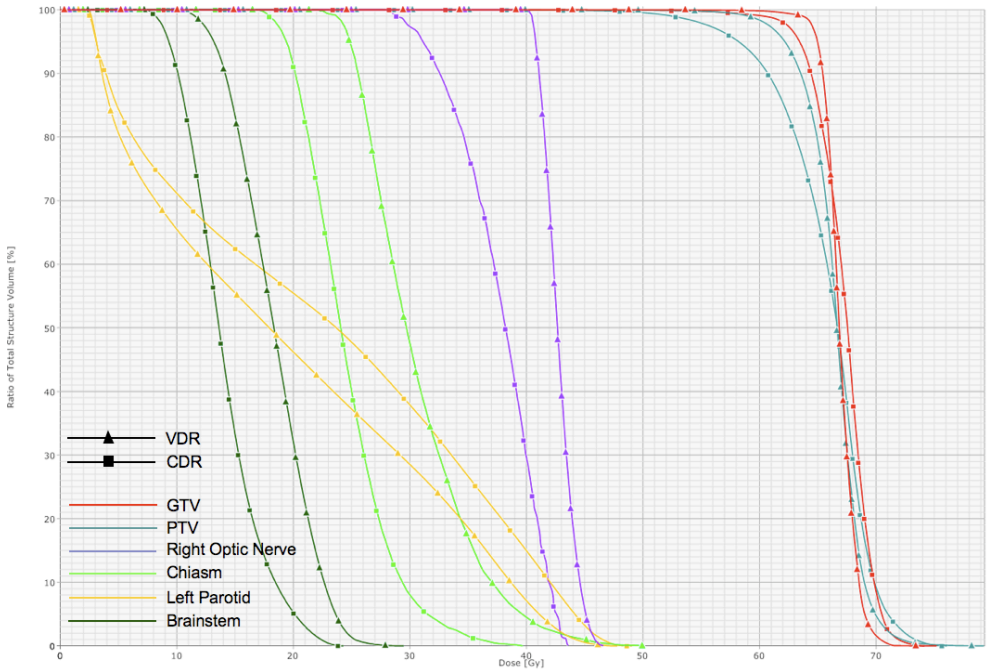
Although the dose distribution may not be susceptible to small angular changes ($\pm$ 5°), the simplistic methodology described by equations 8.1 and

8.2 may yield CDR plans with largely displaced beams that may affect the resultant plan quality. This was exemplified in HN case 1 where some of the beams were redistributed to a significantly different angular position up to almost 40° away from the original planned angle. This is shown in figure 8.2 (a) where the angular differences between the original beam angle in a VDR plan and the altered beam angle in the corresponding converted CDR plan was plotted for each control point (i.e. each beam). Most of the beams resulted in large angular changes and such large angular deviations led to an undesirable plan degradation in the converted CDR plan as illustrated in the DVH comparison in figure 8.2 (b).

The magnitude of angular deviations strongly depends on the MU distribution of the plan. In figure 8.3, the absolute MU of the beams in HN case 1 were plotted against a circular scale which represents the actual beam (gantry) angle, while the origin of the graph coincides with the isocentre of the plan. The global fluctuation of MU weightings within the treatment arc was fairly irregular and uneven. The entire MU distribution was skewed downwards as seen in figure 8.3 (a), which can be explained by the location of the target and isocentre as shown in figure 8.3 (b). Based on this uneven and heavily skewed MU distribution, the angular displacements resulted from the CDR conversion were maximised which subsequently maximised the dosimetric effects of the conversion procedure.

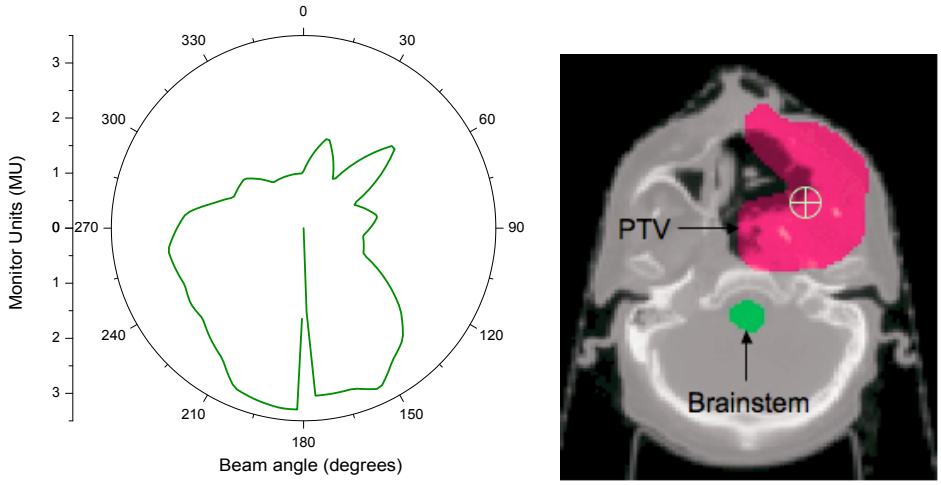


(a)



(b)

Figure 8.2: (a) Angular differences of the beams in the VDR and the converted CDR plans, and (b) DVH comparison between the VDR and CDR plans of HN case 1.



(a) (b)

Figure 8.3: (a) MU distribution of the RapidArc plan, and (b) a transverse CT slice at the isocentre plane of HN case 1.

Certainly, it was crucial to sustain the original plan quality as one of the primary rationale of this proof of principle study. As discussed above, the dosimetric perturbation depends on the angular deviation of the beams during the CDR conversion, which is directly affected by the MU distribution of the plan. Thus, it was important to minimise the MU fluctuation within the arc. However, the planning system could not restrict or control the MU weighting of the beams during optimisation. In order to maintain the plan quality and prevent large angular deviations, the resultant VDR-optimised treatment arc was partitioned into multiple sectors such that the local MU profile or distribution within each sector was comparatively less irregular. The dissection point of the sectors were determined by the cumulative angular deviation at a particular aperture within the arc. Following the observation seen in Chapter 5, a maximum angular alteration of $\pm 5^{\circ}$ was set as the dissection limit.

The overview of the conversion procedure is illustrated with a flow chart

in figure 8.4. The dissection or breaking point of the sectors is where the local MU fluctuation is minimal such that the beams would not largely deviate from their original angles during the re-spacing process. To search for the appropriate breaking points, the conversion process starts by assuming the breaking point is at the last aperture of the arc, N (and that N is the total number of apertures of the arc), i.e. the breaking point is at the last beam such that the entire arc is one "sector" as illustrated in figure 8.5 (a). The resultant angular deviations of the beams from the original angles are then evaluated. If any beam is shifted $> \pm 5^\circ$ away from the original planning angle, the plan will be partitioned into sectors as the breaking point moves to beam N - 1, narrowing down the arc range of the original sector (that contained N beams).

Note that there are 2 sectors in the plan now, with the first sector containing beams 1 to N - 1, and the second sector containing only one beam (beam N), as shown in figure 8.5 (b). The search of the breaking point continues from beam N - 1, neglecting beam N. These beams are re-spaced using equations 8.3 and 8.4,

$$s_j(\mathrm{MUdeg}^{-1}) = \frac{(\sum_{i=1}^{n} \mathrm{MU}_i)_j}{\Theta_j(\mathrm{deg})}, \qquad \text{where } i = 1, 2, 3, ..., n, \tag{8.3}$$

$$(\Delta\theta_i)_j(\mathrm{deg}) = \frac{(\mathrm{MU}_i)_j}{s_j(\mathrm{MUdeg}^{-1})}, \tag{8.4}$$

where n is the number of beams or apertures in sector j and n for this sector (j = 1) is N - 1. After the CDR conversion, the angular deviation of these beams are evaluated again. This step is repeated until all angular displacements within a sector are $< \pm 5^\circ$ and this sector will be endorsed and locked, i.e. a

particular s_j value is established for the CDR conversion for this sector or arc range of the treatment plan. The conversion process then restarts and searches for the second breaking point but neglects the beams that are involved in the previous endorsed sector. For example, in figure 8.5 (c), sector j is established with n_j beams. The search of the breaking point of sector $j + 1$ only takes into the account of beams $n_j + 1$ to N. This procedure repeats until all the beams in the plan are grouped into sectors. Because the beams in each sector are re-spaced with a different spacing factor s_j, the resultant CDR plan will be delivered with up to j different values of CDR. The CDR plan is still delivered with a single gantry sweep as in VDR delivery but the linac must stop at each sector junction to change to a different dose rate. However, since the last aperture of sector j is at the same beam angle with the same shape as the first aperture of sector $j + 1$, the machine does not need to change MLC positions or gantry angles in between sector transitions. Figure 8.6 illustrates the delivery arrangement of a CDR plan, which contains 3 broken sectors. The sectors are delivered one after another with a short beam-hold (radiation turned off) as the machine prepares for the change of dose rate. Although the fast delivery time benefit of "a single continuous gantry sweep" of the single-arc IMAT techniques may perish with such broken sector delivery pattern, the overall delivery time of CDR plans may still be faster than multiple static-field IMRT as there is no need for the machine to change the gantry position and MLC positions between sectors.

The 5-degree criterion seems to work well in the CDR conversion process as none of the converted CDR plans result in significant plan quality degradation. Nonetheless, "5-degree" is just an arbitrary value used in a previous study. Theoretically, the smaller the angular alteration, the smaller the resul-

tant dosimetric effects. As an effort to maintain the VDR plan quality and minimise the dosimetric differences between the VDR and CDR plans, a refining step is added to the conversion process. The VDR plan is converted again using a more stringent breaking criterion as the allowed angular deviation is decreased by 1°. By doing so, the breaking points will be re-adjusted and new sectors will be formed. If the number of sectors in this new converted plan is the same as the original CDR plan, then the new converted plan will be accepted and the original plan will be discarded as the new CDR plan contains beams that are less deviated and the dose distribution is expected to be less different from the VDR plan. This refinement step is repeated until the number of sectors in the new CDR plan increases compared to the previous plan. When the number of sectors increases, the delivery time may be lengthened, therefore the new CDR plan will be voided and the previous CDR plan will be accepted as the final plan.

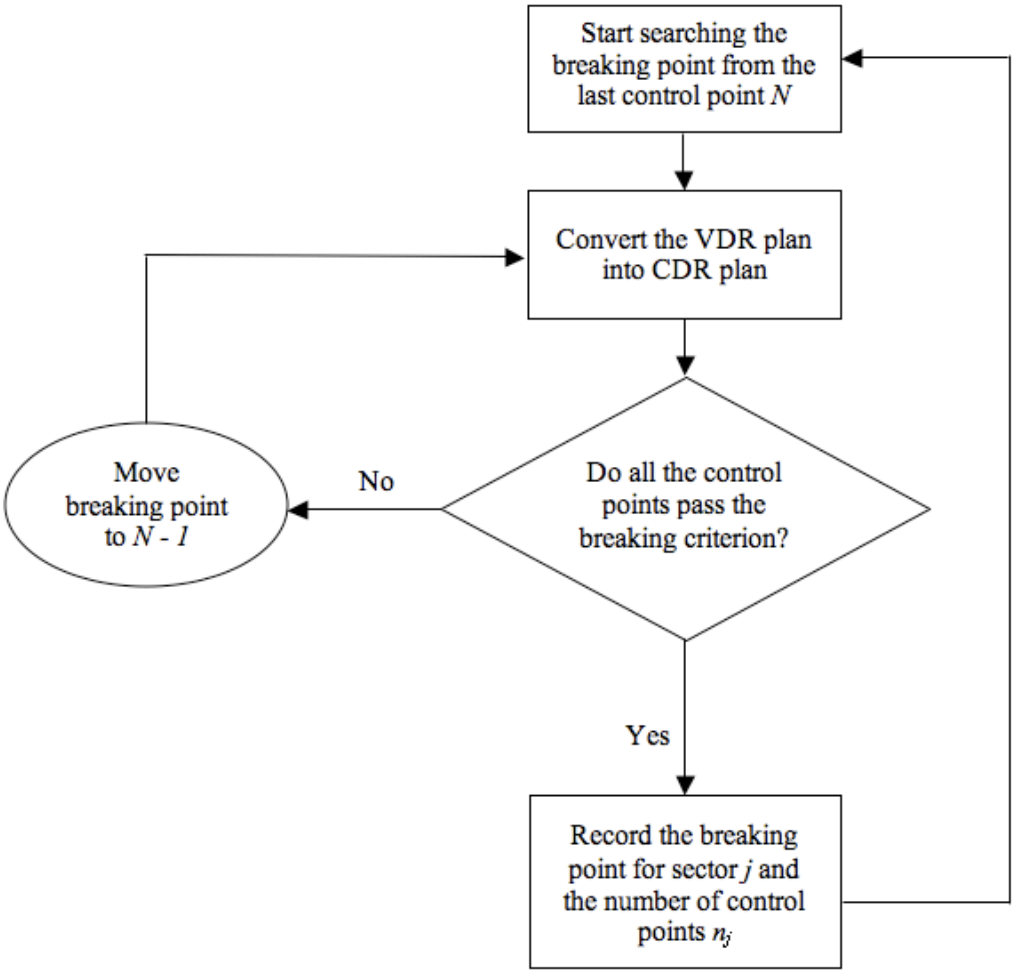


Figure 8.4: Logistic flow of the CDR conversion process.

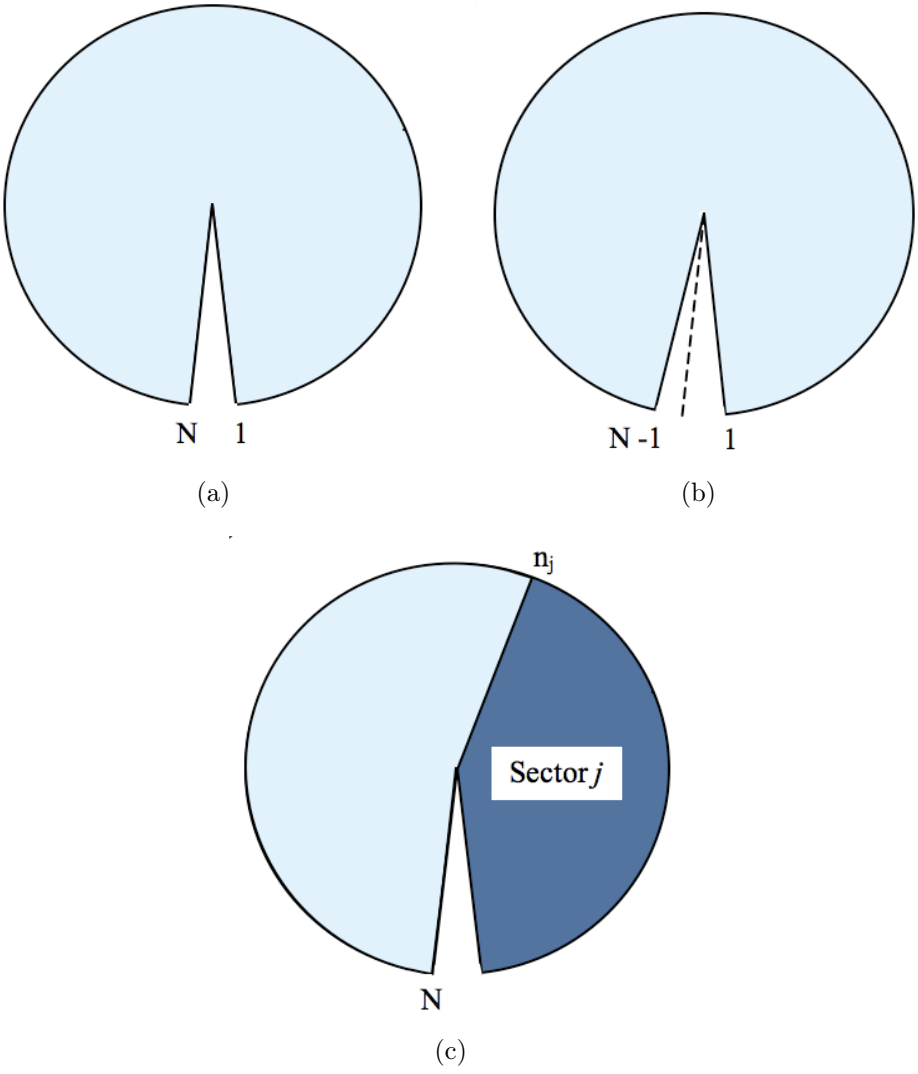


Figure 8.5: Search of the breaking points of the sectors starts backwards of the arc (a) assuming the entire arc is one sector with N beams, i.e. the breaking point is at beam N. (b) If a beam deviates $> \pm 5°$ after the CDR conversion, the breaking point is moved to beam N - 1, narrowing down the sector range. (c) When a sector j is found, the conversion restarts from beam N again but neglecting n_j beams that are involved in sector j.

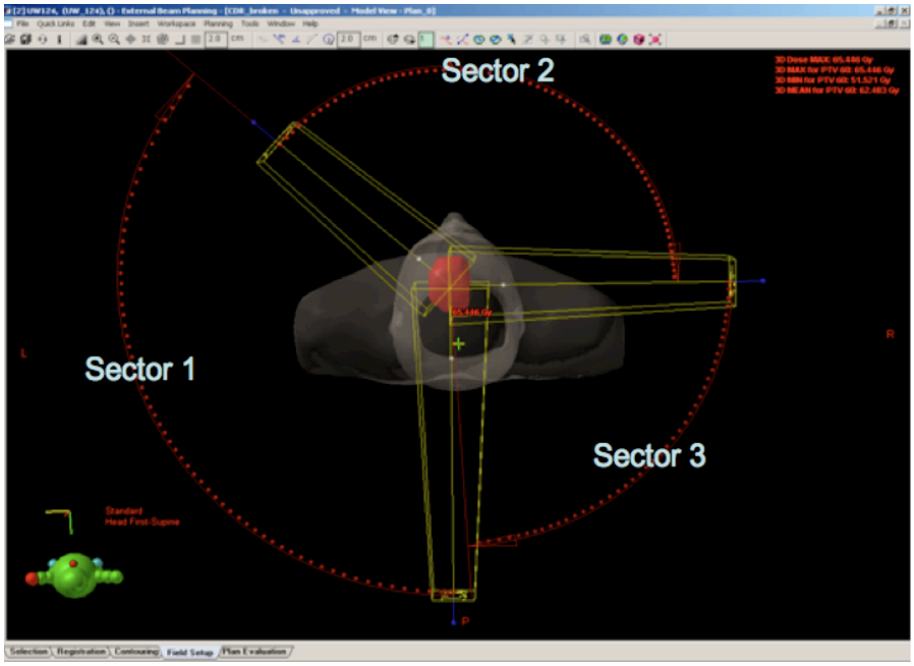


Figure 8.6: Delivery pattern of a CDR plan with 3 broken sectors.

8.3 Plan quality comparison

Compared to the original VDR plans, the corresponding CDR plans generally showed minimal discrepancies in the patient case studies. Some differences were observed in the normal tissues but the maximum allowed toxicity levels were not violated. Figures 8.7 - 8.11 display the DVH comparisons of the plans and table 8.1 summarises the results of the plan comparison of the patient cases (the DVH of all plans were normalised at the mean dose of the PTV). The dose differences between the VDR and CDR plans were quantified by Δd,

$$\Delta d = \frac{D_{CDR} - D_{VDR}}{D_{VDR}} \times 100(\%), \tag{8.5}$$

where D_{VDR} and D_{CDR} represent the dose calculated for the VDR and CDR plans, respectively.

The differences in D_{95} of the PTV were less than 0.6% for all cases, except for HN case 3 where a difference of -1.62% was seen. This larger discrepancy may be due to the additional number of arcs involved. While all other cases only contain one single-arc, HN case 3 required 2 single-arcs in order to encompass the large target volume. This is a mechanical issue with the Varian linac where the jaws are limited to a maximum opening of 14 cm in the x-direction. Since the MLC carriages are also fixed, at least 2 arcs are required to cover a target volume that is larger than 14 cm in x or y direction. Apart from the slightly different target doses observed in this complex HN case, the dosimetric effects on the normal tissues were minimal. For HN cases 1 and 2, some differences were observed in the normal tissue toxicity, in particular, the mean dose of the left parotid was increased by 7.7%. However, none of these differences exceeded the maximum allowed dose limits. On the other hand, most of the dosimetric discrepancies observed in the normal tissues were negative differences, i.e. decrease in dose. This may only be a random effect and cannot be concluded as a benefit or an advantage of proceeding the VDR plans with a CDR conversion since it was not intended or controlled whether the displacements of the beams would result in a less normal tissue toxicity during the conversion.

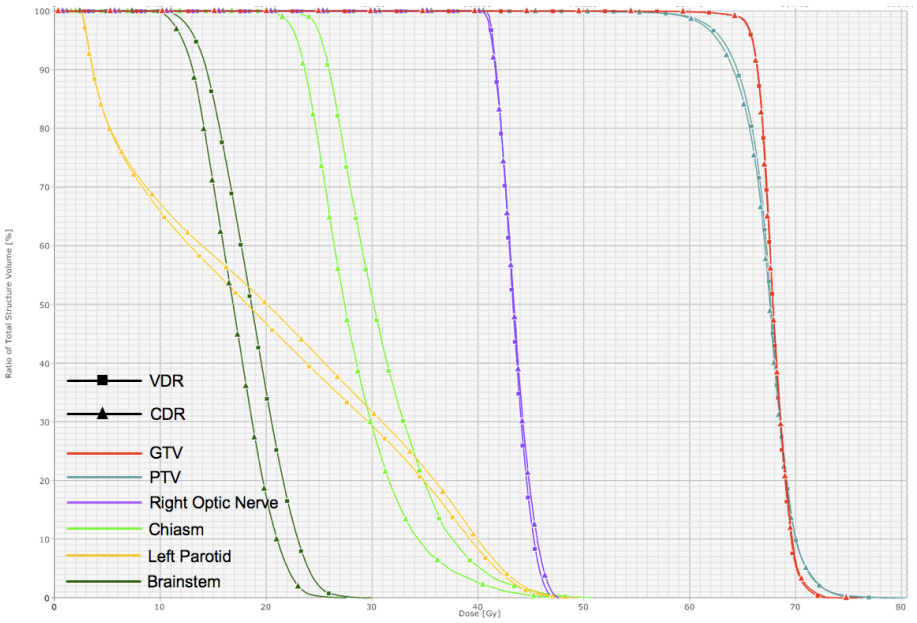


Figure 8.7: DVH comparison between the VDR and CDR plans for HN case 1.

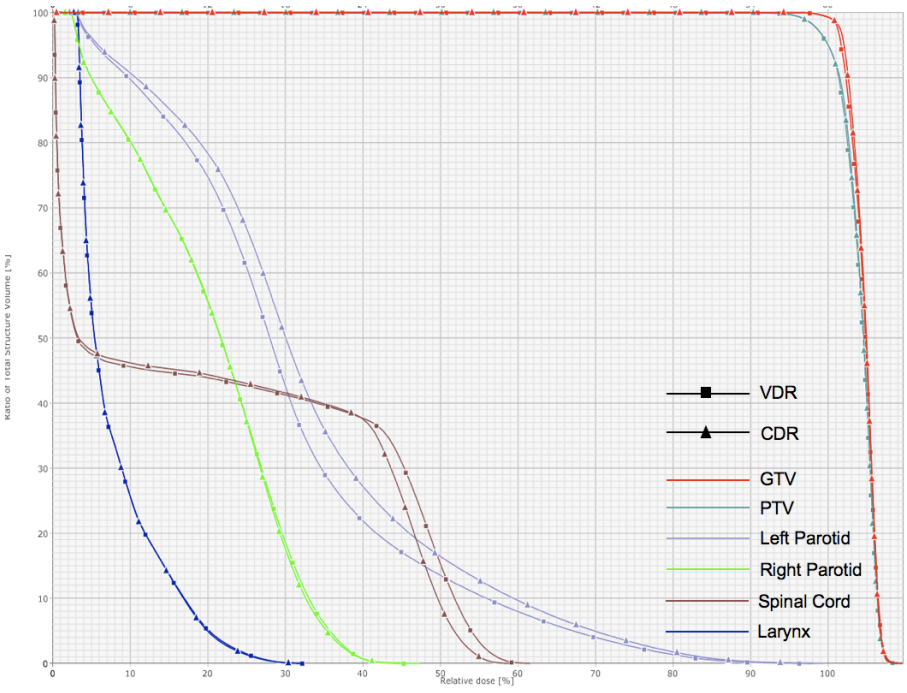


Figure 8.8: DVH comparison between the VDR and CDR plans for HN case 2.

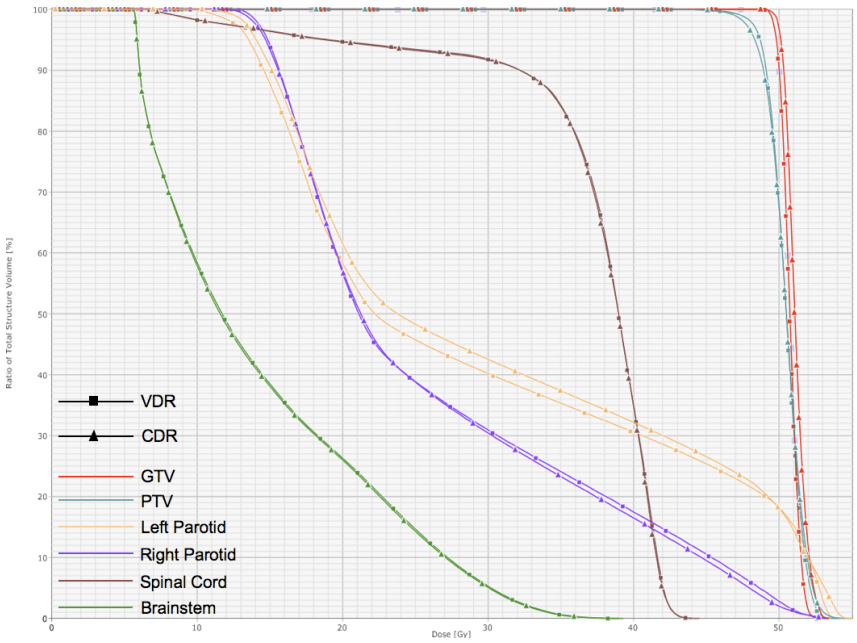


Figure 8.9: DVH comparison between the VDR and CDR plans for HN case 3.

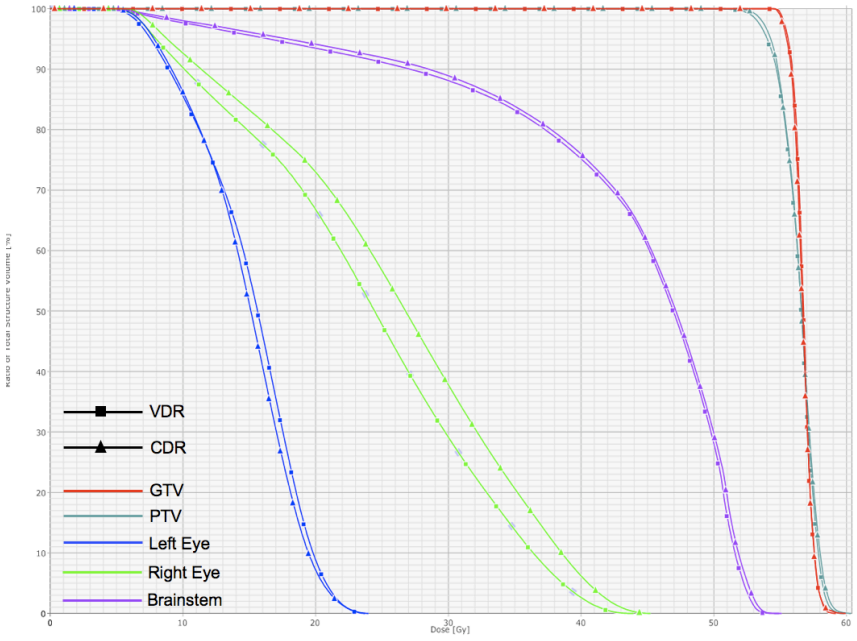


Figure 8.10: DVH comparison between the VDR and CDR plans for the brain case.

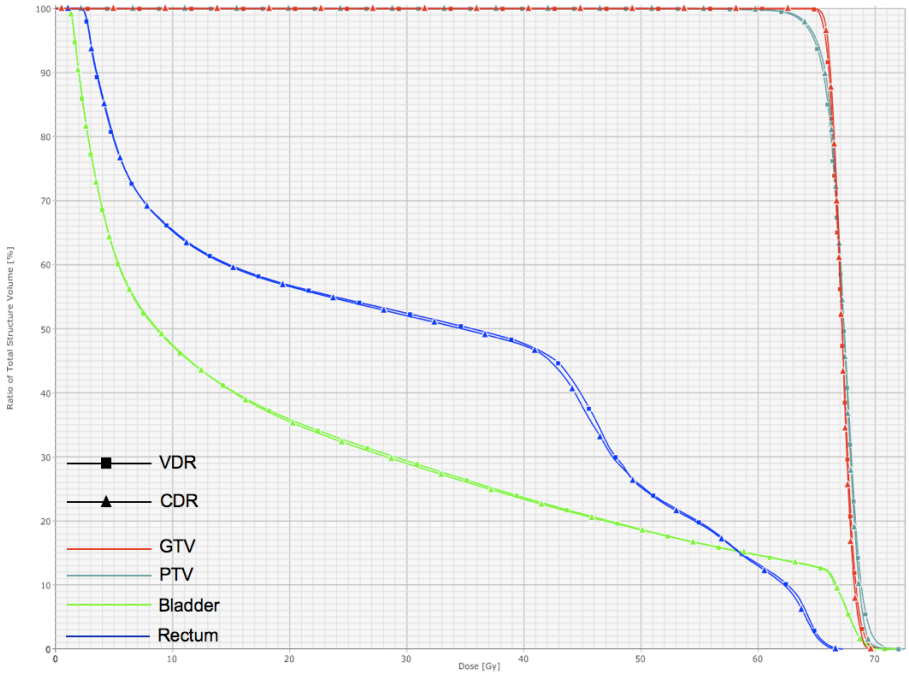


Figure 8.11: DVH comparison between the VDR and CDR plans for the prostate case.

Table 8.1: Differences between the VDR and CDR plans for a few dose indices of the patient cases, Δd denotes the percentage dose difference between the VDR and CDR plans.

Case	# sectors	$\Delta\theta$	ROI	Dose Index	Δd (%)
HN 1	6	5°	GTV	Mean	0.03
			PTV	D_{95}	-0.61
				D_5	0.09
			Left Parotid	Mean	4.23
			Brainstem	Max	-7.70
			Optic Chiasm	Max	-6.20
			Right Optic Nerve	Max	0.70
HN 2	3	4°	GTV	Mean	0.10
			PTV	D_{95}	0.00
				D_5	0.00
			Left Parotid	Mean	7.73
			Right Parotid	Mean	0.43
			Spinal Cord	Max	-4.35
			Larynx	Max	-0.33
HN 3	5 (arc 1)	3° (arc 1)	GTV	Mean	1.35
	3 (arc 2)	5° (arc 2)	PTV	D_{95}	-1.62
				D_5	0.53
			Left Parotid	Mean	-0.40
			Right Parotid	Mean	1.96
			Spinal Cord	Max	-2.23
			Larynx	Max	-0.66
Brain	4	5°	GTV	Mean	-0.04
			PTV	D_{95}	-0.67
				D_5	0.43
			Left Eye	Max	-1.35
			Right Eye	Max	-3.08
			Brainstem	Max	-0.60
Prostate	4	4°	GTV	Mean	0.13
			PTV	D_{95}	-0.43
				D_5	0.58
			Bladder	V_{65}	0.00
			Rectum	V_{60}	0.00

8.4 Dosimetric verification

All plans were delivered for dosimetric verification purposes and all plans were delivered successfully without any machine interlocks. The VDR plans were delivered with the Varian 21 iX Trilogy linac with RapidArc capability and the CDR plans were delivered with the Varian 21 EX Silhouette machine which was not enabled with continuous dose-rate variation during rotational delivery. The MapCHECK was fitted into a 5 cm thick MapPHAN to measure the composite dose of the plans. The measurements of the 2D planar dose at the coronal plane from the MapCHECK/MapPHAN setup were compared to that computed by Eclipse (version 8.6). For all plans, dose distributions were calculated using a grid size of 2.5 x 2.5 x 3 mm^3. The calculated dose planes were exported from Eclipse and imported to the MapCHECK software for analyses as demonstrated in figure 8.12. A gamma comparison was performed for all cases with an analysis criteria of 3%/3 mm. The gamma passing rates of the plans are tabulated in table 8.2. The passing rate quantifies the differences between the dose distribution optimised in the treatment plan and the dose distribution that was actually delivered based on the analysis criteria. The gamma analysis presented in table 8.2 is based on the analysis criteria of 3%/3 mm, which denotes if any data point mismatches 3% in relative dose or 3 mm in distance between the optimised plan (computed dose) and the actual delivery (MapCHECK measurement), the data point will be counted as a failure. A passing score $> 95\%$ was achieved by all cases, indicating that both the VDR and the converted CDR plans could be accurately delivered with clinically-approved standards.

Table 8.2: Passing rates of the gamma analysis with the criteria of 3%/3 mm of each of the plans of all cases.

Case	Gamma passing rate	
	VDR	CDR
HN 1	100%	98.7%
HN 2	100%	100%
HN 3	97.3%	96.3%
Brain	100%	100%
Prostate	98.1%	96.1%

8.5 Delivery time comparison

To address the concern in delivery time extension in the CDR plans, the delivery time of all VDR and CDR deliveries were recorded and compared. Note that the delivery time was the time taken for the treatment plan to be delivered and the setup times were not included, i.e. the time measurement started at the instance the beam was turned on and ended when the entire treatment plan was delivered. For HN case 3 where the RapidArc plan consisted of 2 single-arcs, the delivery time was measured separately for each individual treatment arc. The delivery time of all CDR plans also included the time taken for the machine to transit between the successive sectors. These additional time can be regarded as extra mode-up time (xMOT). The xMOT was the time spent on machine internal calibrations only, since the linac did not have to change the gantry angle and MLC shapes between the sector transitions. Table 8.3 tabulates the measured delivery time of all plans. For all cases, the CDR plans prolonged the delivery times with the xMOT compared to the VDR plans. On average, the time needed to deliver the CDR plans was approximately 1 minute longer than that required for the VDR plans. The extra time spent on CDR delivery was proportional to the number of sectors in the plans as shown in table 8.4. From all CDR plans, an average xMOT

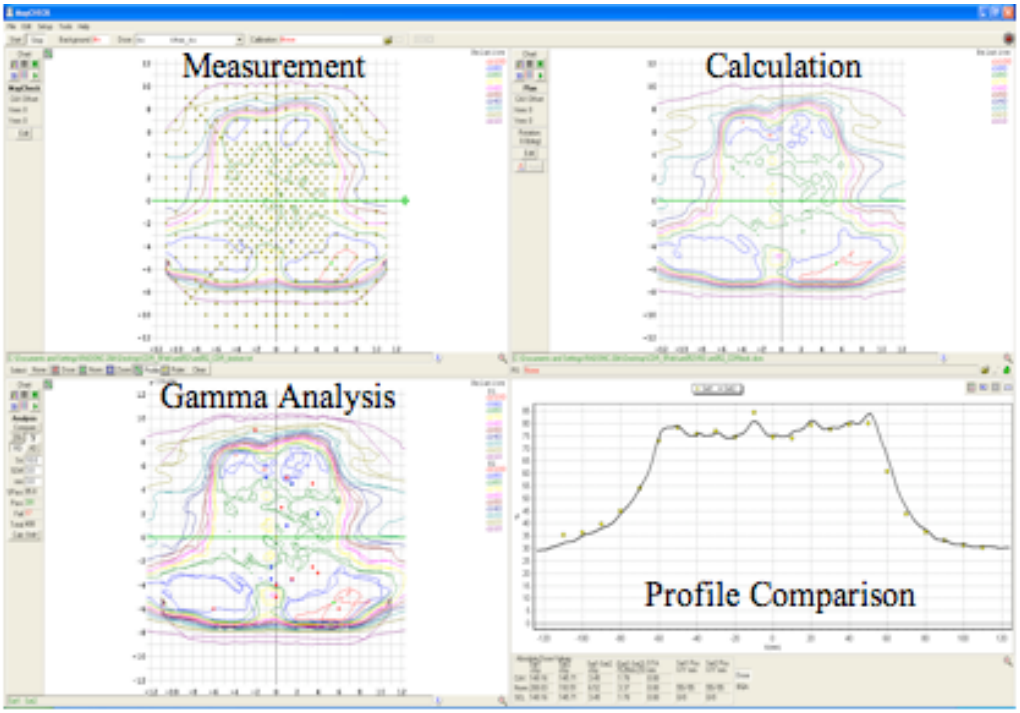


Figure 8.12: Comparison of MapCHECK measurement and Eclipse calculation of the CDR plan of HN case 3 using the MapCHECK software.

of 0.31 minute per sector transition was found. Neglecting the xMOT, the BOT of VDR delivery was approximately 4% longer than that in the CDR delivery. This may be caused by the slower gantry rotation in the VDR plans for the delivery of high MU values and/or allowing the MLC to have sufficient time to displace between the adjacent apertures. As the apertures were redistributed in the CDR conversion based on the MU weightings, the aperture with a higher MU value may require longer time for delivery over a larger angular interval while the gantry remained enslaved by the MLC displacement. Nonetheless, none of the 5 cases required the gantry to rotate at a significantly slower speed in the CDR delivery and resulted in a slightly faster BOT than that in the VDR plans.

Table 8.3: Time recorded for the VDR and CDR deliveries of all cases. Delivery time is the total time taken for the entire plan delivery including the xMOT and BOT. Note that there is no xMOT for the VDR deliveries.

Case	Plan	Delivery time (min)	xMOT (min)	BOT (min)
HN 1	VDR	1.25	-	1.25
	CDR	2.75	1.32	1.23
HN 2	VDR	1.25	-	1.25
	CDR	1.84	0.64	1.20
HN 3	VDR	1.24	-	1.24
(arc 1)	CDR	2.43	1.23	1.20
	VDR	1.24	-	1.24
(arc 2)	CDR	1.80	0.61	1.19
Brain	VDR	1.25	-	1.25
	CDR	2.38	1.17	1.21
Prostate	VDR	1.68	-	1.68
	CDR	2.35	0.84	1.51

Table 8.4: Absolute xMOT between each sector transition and the total xMOT (ΣxMOT) of each CDR plan.

Case	# arcs	# sectors	Sector Transition	xMOT (min)	ΣxMOT (min)
HN 1	1	6	$1 \rightarrow 2$	0.30	1.62
			$2 \rightarrow 3$	0.28	
			$3 \rightarrow 4$	0.34	
			$4 \rightarrow 5$	0.39	
			$5 \rightarrow 6$	0.31	
HN 2	1	3	$1 \rightarrow 2$	0.34	0.64
			$2 \rightarrow 3$	0.30	
HN 3	1	5	$1 \rightarrow 2$	0.29	1.23
			$2 \rightarrow 3$	0.31	
			$3 \rightarrow 4$	0.34	
			$4 \rightarrow 5$	0.29	
	2	3	$1 \rightarrow 2$	0.10	0.61
			$2 \rightarrow 3$	0.51	
Brain	1	4	$1 \rightarrow 2$	0.48	1.17
			$2 \rightarrow 3$	0.40	
			$3 \rightarrow 4$	0.29	
Prostate	1	4	$1 \rightarrow 2$	0.29	0.84
			$2 \rightarrow 3$	0.37	
			$3 \rightarrow 4$	0.18	

8.6 Limitations of retrospective CDR conversion

Although encouraging results have been shown from the CDR plans, the retrospective conversion method used was for proof of principle purposes only. The conversion scheme described above is limited by the property of the original VDR-optimised plan and is especially dependent on the MU distribution of the plan which governs the angular deviation of the beams in the converted CDR plan. Since the MU distribution pattern cannot be controlled during the optimisation, the plans may need to be broken into multiple sectors in order to be delivered with CDR. Without breaking the arc, large angular deviations can be caused by an irregular MU distribution and can subsequently affect the resultant plan quality. On the other hand, if the MU distribution is relatively smooth and centrally distributed, the beams will be less deviated from their original planned positions during the redistribution. Thus, the plan can be partitioned into fewer sectors in order to sustain the original plan quality and the delivery time will be minimised. This is exemplified by HN case 2, where the MU distribution is uniform and there was only 3 sectors in the CDR plan (see figure 8.13).

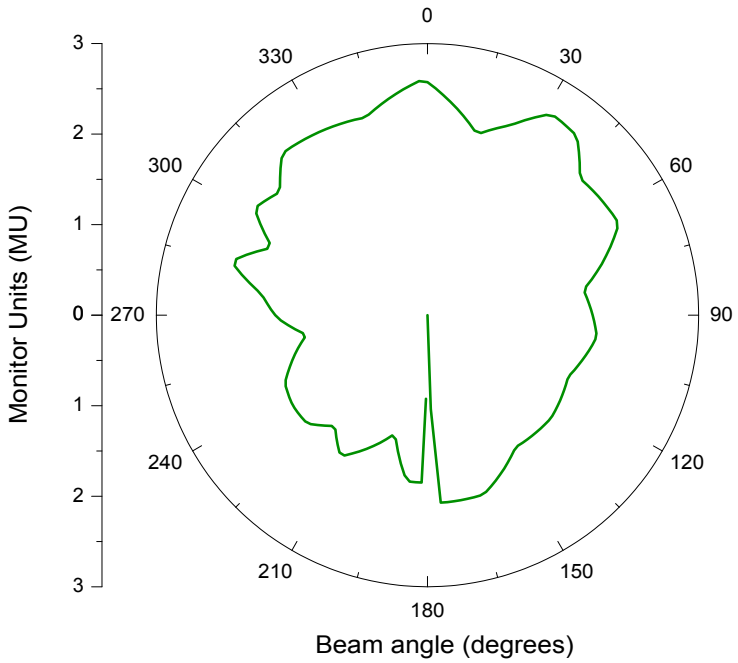


Figure 8.13: MU distribution of HN case 2.

Because the limitation of angular displacement of the beams was not considered during the original VDR plan, by applying such retrospective conversion procedure may unnecessarily deviate the beams and affect the final plan quality for some cases. Although the refinement step in the conversion process helped to further minimise the angular deviations of the redistributed beams, a prospective CDR planning approach may be able to minimise the number of sectors, or in some cases, multiple sectors are not even necessary. For example, during the progressive sampling in RapidArc optimisation, the additional new beam angles can be unevenly spaced instead of distributing with a regular spacing. The new beam angles can be predetermined based on the relative importance of all beam angles using some angular cost functions [Yu *et al* 2006]. By evaluating the relative angular importance *a priori*, the new additional beams can be allocated to the appropriate positions. Such prospective CDR planning strategy may minimise or even eliminate the use of

multiple broken sectors, maintaining similar delivery efficiency to the original VDR delivery.

8.7 Summary

Commercial single-arc IMAT products have been released for clinical uses. Products such as Varian RapidArc requires continuous dose-rate variation during the arc delivery. New machines are equipped with this capability while the existing linacs do not yet support this deliverability. The mode of delivery is limited by the planning approach where a series of evenly-spaced beams are used to approximate the continuous delivery arc for optimisation. For example, VDR is required for delivery if the weightings of these evenly-spaced beams are allowed to vary while selecting CDR delivery mode will require restriction on constant MU weightings of these beams. While it is widely misconceived that VDR optimisation is superior in plan quality to those optimised with CDR, this feasibility study has provided evidence that shows otherwise. An alternative planning approach was investigated for CDR delivery. Based on the hypothesis that FM is equivalent to AM as in radio-broadcasting, VDR-optimised plans were converted for CDR delivery. The plan quality was very similar between the VDR plans and the corresponding CDR plans with slightly lengthened delivery times. With a prospective CDR planning strategy, the delivery efficiency may be further improved for the CDR plans.

In addition, one of the benefits of CDR delivery is that clinics which do not have sufficient funding for the purchase of new machines can implement the latest radiotherapy techniques such as single-arc IMAT with the existing resources. As a result, the user base of this latest technology may expand and

increase the experience and input into its research, which ultimately will translate into clinical practices and enhance future patient treatment outcomes.

Chapter 9

Conclusions and future work

9.1 Conclusions

This project aims to further expand the potentials of IMAT in the clinic by exploring various treatment planning and delivery issues. The accuracy and efficiency of IMAT treatment planning have been improved by using the MC method for dose calculations. With the advantage of random sampling in MC where CPU time becomes independent of the number of beams, IMAT plans can now be calculated more accurately and efficiently in less than half an hour which is considered clinically acceptable. On the other hand, it was found that the necessity of using a large number of beams for dose calculation depends on the aperture shape and aperture weighting variations between the planning beams. When the aperture shape and weighting vary largely, dose distribution should be calculated using a larger number of beams to minimise dosimetric errors and degradation in plan quality.

Following on to MC-improved planning efficiency, the treatment efficiency of IMAT has also been augmented by designing a new mode of delivery. Compared to the typical delivery time of a conventional multi-arc IMAT plan that may take 5 - 15 minutes, single-arc IMAT can be delivered in under 4 minutes indicating that this new implementation is competitive to other current standard radiotherapy techniques such as IMRT. In comparing plan quali-

ties between single-arc IMAT, multi-arc IMAT and IMRT, it was found that multi-arc IMAT outperforms both single-arc IMAT and IMRT while single-arc IMAT can match IMRT with similar dose distributions. This study showed that single-arc IMAT is an efficient radiotherapy treatment technique that may be beneficial to patient care and clinic workflow.

Since single-arc IMAT plan is often optimised with variable aperture shapes and aperture weightings using a series of evenly-spaced beams, the final treatment plan is always required to be delivered using expensive newer or upgraded machines with dose-rate variation capability. In order to widen the clinical availability, an alternative planning and delivery approach of single-arc IMAT has been proposed using only constant dose rates so that the plans become deliverable using existing linacs, allowing the clinic to early endeavour into rotational IMRT exploiting its delivery efficiency and treatment efficacy.

9.2 Future work

- MC-based dose calculations at appropriate stages of iterations during IMAT optimisation can be performed to reduce the differences between treatment planning and actual delivery, which may be translated to producing better plans.

- Perform systematic analysis of AMRT using a larger number of patient case samples to account for inter-patient variance in the individual disease sites.

- Using a general statistical strategy, systematic VDR to CDR conversions can be conducted to arrive at a more conclusive equivalency.

- Instead of retrospectively converting VDR plans, CDR single-arc IMAT plans can in principle be obtained prospectively where the planning beam angles in the optimisation can be unevenly-spaced. For example, during RapidArc optimisation, the additional new beam angles can be determined based on the relative angular importance which can be evaluated *a priori*. By doing so, the number of sectors broken in the CDR plan may be minimised hence further improving the efficiency of CDR delivery.

Chapter 10

Publications and conference presentations

Below is a list of peer-reviewed publications and presentations arising from the work done in this project.

10.1 Peer-reviewed publications

1. **Grace Tang**, Matthew A Earl, Shuang Luan, Chao Wang, Daliang Cao, Cedric X Yu and Shahid A Naqvi. *Stochastic versus deterministic kernel-based superposition approaches for dose calculation of intensity-modulated arcs.* Phys Med Biol, vol. 53, no. 17, pages 4733 - 46, 2008.

2. C Clifton Ling, Pengpeng Zhang, Yves Archambault, Jiri Bocanek, **Grace Tang** and Thomas LoSasso. *Commissioning and quality assurance of RapidArc radiotherapy delivery system.* Int J Radiat Oncol Biol Phys, vol. 72, no. 2, pages 575 - 81, 2008.

3. Chao Wang, Shuang Luan, **Grace Tang**, Danny Z Chen, Matt A Earl and Cedric X Yu. *Arc-modulated radiation therapy (AMRT): a single-arc form of intensity-modulated arc therapy.* Phys Med Biol, vol. 53, no. 22, pages 6291- 303, 2008.

4. **Grace Tang**, Matthew A Earl and Cedric X Yu. *Variable dose rate single-arc IMAT delivered with constant dose rate and variable angular spacing.* Phys Med Biol, vol. 54, no. 21, pages 6439-56, 2009.

5. **Grace Tang**, Matthew A Earl, Shuang Luan, Chao Wang, Majid M Mohiuddin and Cedric X Yu. *Comparing Radiation Treatments Using Intensity-Modulated beams, Multiple Arcs and Single Arcs.* Int J Radiat Oncol Biol Phys, vol. 76, no. 5, pages 1554-62, 2010.

6. Yi Rong, **Grace Tang**, James Welsh, Majid M Mohiuddin, Bhudatt R Paliwal and Cedric X Yu. *Helical tomotherapy vs. single-arc intensity-modulated arc therapy (IMAT): A collaborative dosimetric comparison between two institutions.* Int J Radiat Oncol Biol Phys, Under Review, 2009.

7. **Grace Tang**, Kristen Asbury, Amy Aurand, Madjid M Mohiuddin, Young Kwok, Bin Zhang, Lisa Richards, Kim Marter and Cedric X Yu. *Plan Quality Comparison of Intensity-Modulated Radiation Therapy and RapidArc for multiple cancer cites* In Preparation, 2009.

10.2 Conference presentations

10.2.1 Oral presentations

1. **Grace Tang**, Matthew A Earl, Shahid A Naqvi and Cedric X Yu. *Monte Carlo/kernel-based dose calculation for Intensity-Modulated Arc Therapy (IMAT).* The 49th annual meeting of the American Association of Physicists in Medicine, 2007.

2. **Grace Tang**, Matthew A Earl, Shuang Luan, Chao Wang, Shahid A Naqvi and Cedric X Yu. *Is dose-rate variation crucial to single-arc radiation therapy delivery?* The 49th annual meeting of the American Association of Physicists in Medicine, 2008.

10.2.2 Poster presentations

1. **Grace Tang**, Matthew A Earl, Shuang Luan, Chao Wang, Shahid A Naqvi and Cedric X Yu. *Converting multiple-arc Intensity-Modulated Arc Therapy into a single arc for efficient delivery.* General poster at the 49th annual meeting of the American Society for Therapeutic Radiology and Oncology, 2007.

2. **Grace Tang**, Matthew A Earl, Shuang Luan, Chao Wang, Shahid A Naqvi and Cedric X Yu. *Comparison of Intensity- Modulated Radiation Therapy, Intensity-Modulated Arc Therapy and Arc-Modulated Radiation Therapy.* Moderated poster at the 50th annual meeting of the American Association of Physicists in Medicine, 2008.

3. **Grace Tang** and Byong Yong Yi. *Patient-Specific QA of Intensity-Modulated Arc Therapy with 2D Diode Array: Initial Experience.* General poster at the 51st annual meeting of the American Association of Physicists in Medicine, 2009.

4. **Grace Tang**, Matthew A Earl and Cedric X Yu. *Alternative Delivery Approach for RapidArc Using Constant Dose Rate.* Moderated poster at the 51st annual meeting of the American Association of Physicists in Medicine, 2009.

Appendix A

C codes for interpolated-static beam calculation

The codes below were compiled to perform a linear interpolation between the adjacent planning beams in an arc plan for interpolated-static beam calculation.

```
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include <math.h>
#include <ctype.h>

#define INMLC_SIZE 40
#define OUTMLC_SIZE 10000
#define ARC_SIZE 15

struct iBEAM                        //input beams
{
  float MU;
  float hMU;
  float GANT;                        //gantry angle of beam
  char MLC_FILE[60];
};

struct xBEAM                      //expanded beams
{
  int ID;
  float MU;                           /linearly interpolated MU
  float hMU;                         //half the MU
  float GANT;
};

struct inMLC
{
  int ID;                               //control point index
  float MU;                           //relative MU
  float Y2;                            //Y2 back-up jaw position
  float Y1;                            //Y1 back-up jaw position
  float X1;                            //X1 back-up jaw position
  float X2;                           //X2 back-up jaw position
  int lp[60];                          //leafpair ID, count from 1 to 60
  float leafpair[60][2];       //leafpair positions
};

struct outMLC
{
  int ID;                               //control point index
  float MU;                           //relative MU
  float Y2;                            //Y2 back-up jaw position
  float Y1;                            //Y1 back-up jaw position
  float X1;                            //X1 back-up jaw position
  float X2;                           //X2 back-up jaw position
  int lp[60];                          //leafpair ID, count from 1 to 60
  float leafpair[60][2];       //leafpair positions
};

int main(int argc, char *argv[])
{
```

```
FILE *read_file;
FILE *read_mlc;
FILE *edit_file;
FILE *edit_mlc;
FILE *fp[OUTMLC_SIZE];
char in_mlc[100];
char out_file[100];
char out_mlc[100];
char mlc_path[100];

struct iBEAM * ibeam ;                //beams from static beam planning
struct xBEAM * cbeam;               //beams in continuous delivery
struct xBEAM * xbeam;               //rediscretized beams for MCKS
struct inMLC ** INmlc;                //from static beam planning
struct outMLC ** cOUTmlc;       //from continuous delivery
struct outMLC ** OUTmlc;         //from rediscretized beams

int a, m, ibeamID, arcID;
float dsep;                             //planned angular separation of beams within arc (
    degrees)
int finebeam;                         //number of interpolated beams between planning
    interval
float dfine;                            //fine-tuning angular separation of beams (degrees)
float coll;                             //collimator angle
float ISOx, ISOy, ISOz;   //3D coordinates of isocentre
float xbeamGANT;             //gantry angle of the expanded beams in degrees
float xbeamGANTrad;       //gantry angle of the expanded beams in radians
int temp_sl ;

ibeam = (struct iBEAM *) malloc ( INMLC_SIZE * sizeof ( struct iBEAM) );
cbeam = (struct xBEAM *) malloc ( OUTMLC_SIZE * sizeof (struct xBEAM) );
xbeam = (struct xBEAM *) malloc ( OUTMLC_SIZE * sizeof (struct xBEAM) );

INmlc = (struct inMLC **) malloc ( INMLC_SIZE * sizeof (struct inMLC *) );
temp_sl = 0 ;
while ( temp_sl < INMLC_SIZE ){
  INmlc[temp_sl] = (struct inMLC *) malloc ( ARC_SIZE * sizeof ( struct inMLC) ) ;
  temp_sl ++ ;
}

//the output MLCs based on the planning beams, without the kick-off and ending beams
cOUTmlc = (struct outMLC **) malloc ( OUTMLC_SIZE * sizeof (struct outMLC *) );
temp_sl = 0 ;
while ( temp_sl < OUTMLC_SIZE ){
  cOUTmlc[temp_sl] = (struct outMLC *) malloc ( ARC_SIZE * sizeof ( struct outMLC) ) ;
  temp_sl ++ ;
}

OUTmlc = (struct outMLC **) malloc ( OUTMLC_SIZE * sizeof (struct outMLC *) );
temp_sl = 0 ;
while ( temp_sl < OUTMLC_SIZE ){
  OUTmlc[temp_sl] = (struct outMLC *) malloc ( ARC_SIZE * sizeof ( struct outMLC) ) ;
  temp_sl ++ ;
}

if(argc==1)
  {
    printf("Sorry, .beams file is not found. Exiting program.\n");
    exit(0);
  }

//prompt user for information on interpolation
int conbeam;           //number of interpolation beams for the continuous sequnece

printf("Please enter the number of fine-tuning beams between the planning angular
    interval desired (odd number).\n");
scanf("%d", &conbeam);

printf("Please specify a folder for locating the interpolated MLC files.\n");
scanf("%s", mlc_path);

read_file  = fopen(argv[1], "r");

int pbcount = 0;                //total number of input beams
int aindex = 0;                  //number of arcs for array definition
int arccount = 0;              //number of arcs
int segcount = 0;             //total number of apertures in plan

while(fscanf(read_file, "%d", &ibeamID) !=EOF)
  {
    aindex = -1;

    ibeamID = ibeamID + 1;

    fscanf(read_file, "%f %f %f %f %f %s", &ibeam[ibeamID].MU, &ibeam[ibeamID].GANT,
        &coll, &ISOx, &ISOy, &ISOz, in_mlc);

    read_mlc = fopen(in_mlc, "r");
```

```
    while(fscanf(read_mlc, "%d", &arcID) != EOF)
    {
        aindex ++;

        fscanf(read_mlc, "%f %f %f %f %f", &INmlc[ibeamID][aindex].MU, &INmlc[ibeamID][
            aindex].Y2, &INmlc[ibeamID][aindex].Y1, &INmlc[ibeamID][aindex].X1, &INmlc[
            ibeamID][aindex].X2);

        for(m = 0; m < 60; m++)
          {
            fscanf(read_mlc, "%d %f %f", &INmlc[ibeamID][aindex].lp, &INmlc[ibeamID][
                aindex].leafpair[m][0], &INmlc[ibeamID][aindex].leafpair[m][1]);
          }
        segcount++;
      }
    fclose(read_mlc);
    pbcount++;
  }

fclose(read_file);

arccount = segcount/pbcount;
dsep = fabs(ibeam[2].GANT - ibeam[1].GANT);
float dcon;                          //continuous sequence angular separation

//counting total MU

float ibMU;                           //total MU of the input beams
ibMU = 0;
int u;

for(u = 0; u < pbcount; u++)
  {
    ibMU += ibeam[u].MU;
  };

//interpolate mlc

int x, t, w, n, D, M, LP, h;

float p = (float)1/(conbeam+1);
float arcSTART, arcEND, arcRANGE;
arcSTART = ibeam[1].GANT + (dsep/2);                          //starting gantry angle (
    radians)
arcEND = ibeam[pbcount].GANT - (dsep/2);
arcRANGE = arcSTART   + (360-arcEND);                        //assuming the arc goes pass 0,
    anticlockwise

dcon = (float)(dsep/2)/(((conbeam - 1)/2)+1);

int tcbcount;                                    //total number of beams count after
    interpolation for the continuous beams
int kepbcount;
kepbcount = pbcount + 2;                    //pbcount + 2 because of the kick-off and
    ending beams
int h_conbeam;                               //conbeams per 5 degrees
h_conbeam = (conbeam - 1)/2;
tcbcount = kepbcount + ((pbcount-1)*conbeam) + (2*h_conbeam);

//adding kick-off and ending beams to the original sequence

INmlc[0] = INmlc[1];
INmlc[pbcount+1] = INmlc[pbcount];

float pp = (float)1/(h_conbeam+1);        //edited 10April07, fraction of interpolation

int KD, ED;                                 //initisalising counters
KD = -1;
D = h_conbeam;
ED = h_conbeam + ((pbcount - 1)*conbeam) + kepbcount - 3 ;

//interpolating shapes for continuous beams
for(w = 0; w < (kepbcount-1); w++)
   {
       if( w == 0)                                                  //the kick-off interval only
           occupies 5 degrees
       {
         for(n = 0; n <= h_conbeam; n++)                    //n is the number of fraction
             the beams are being expanded
           {
             KD++;

             for(a = 0; a < arccount; a++)
               {
                 cOUTmlc[KD][a].MU = INmlc[w][a].MU + (n*pp*(INmlc[w+1][a].MU - INmlc[w
                     ][a].MU));
```

```
                cOUTmlc[KD][a].Y2 = INmlc[w][a].Y2 + (n*pp*(INmlc[w+1][a].Y2 - INmlc[w
                    ][a].Y2));
                cOUTmlc[KD][a].Y1 = INmlc[w][a].Y1 + (n*pp*(INmlc[w+1][a].Y1 - INmlc[w
                    ][a].Y1));
                cOUTmlc[KD][a].X1 = INmlc[w][a].X1 + (n*pp*(INmlc[w+1][a].X1 - INmlc[w
                    ][a].X1));
                cOUTmlc[KD][a].X2 = INmlc[w][a].X2 + (n*pp*(INmlc[w+1][a].X2 - INmlc[w
                    ][a].X2));

                for(LP = 0; LP < 60; LP++)
                  {
                    cOUTmlc[KD][a].leafpair[LP][0] = INmlc[w][a].leafpair[LP][0] + (n*
                        pp*(INmlc[w+1][a].leafpair[LP][0]-INmlc[w][a].leafpair[LP][0])
                        );
                    cOUTmlc[KD][a].leafpair[LP][1] = INmlc[w][a].leafpair[LP][1] + (n*
                        pp*(INmlc[w+1][a].leafpair[LP][1]-INmlc[w][a].leafpair[LP][1])
                        );
                  }
              }
          }
        continue;
      }

    if( w == (kepbcount - 2))
      {
        for(n = 0; n <= h_conbeam; n++)                    //n is the number of fraction
            the beams are being expanded
          {
            ED++;

            for(a = 0; a < arccount; a++)
              {
                cOUTmlc[ED][a].MU = INmlc[w][a].MU + (n*pp*(INmlc[w+1][a].MU - INmlc[w
                    ][a].MU));
                cOUTmlc[ED][a].Y2 = INmlc[w][a].Y2 + (n*pp*(INmlc[w+1][a].Y2 - INmlc[w
                    ][a].Y2));
                cOUTmlc[ED][a].Y1 = INmlc[w][a].Y1 + (n*pp*(INmlc[w+1][a].Y1 - INmlc[w
                    ][a].Y1));
                cOUTmlc[ED][a].X1 = INmlc[w][a].X1 + (n*pp*(INmlc[w+1][a].X1 - INmlc[w
                    ][a].X1));
                cOUTmlc[ED][a].X2 = INmlc[w][a].X2 + (n*pp*(INmlc[w+1][a].X2 - INmlc[w
                    ][a].X2));

                for(LP = 0; LP < 60; LP++)
                  {
                    cOUTmlc[ED][a].leafpair[LP][0] = INmlc[w][a].leafpair[LP][0] + (n*
                        pp*(INmlc[w+1][a].leafpair[LP][0]-INmlc[w][a].leafpair[LP][0])
                        );
                    cOUTmlc[ED][a].leafpair[LP][1] = INmlc[w][a].leafpair[LP][1] + (n*
                        pp*(INmlc[w+1][a].leafpair[LP][1]-INmlc[w][a].leafpair[LP][1])
                        );
                  }
              }
          }
        continue;
      }

    else
      for(n = 0; n <= conbeam; n++)                        //n is the number of fraction the
          beams are being expanded
        {
          D++;

          for(a = 0; a < arccount; a++)
            {
              cOUTmlc[D][a].MU = INmlc[w][a].MU + (n*p*(INmlc[w+1][a].MU - INmlc[w][a
                  ].MU));
              cOUTmlc[D][a].Y2 = INmlc[w][a].Y2 + (n*p*(INmlc[w+1][a].Y2 - INmlc[w][a
                  ].Y2));
              cOUTmlc[D][a].Y1 = INmlc[w][a].Y1 + (n*p*(INmlc[w+1][a].Y1 - INmlc[w][a
                  ].Y1));
              cOUTmlc[D][a].X1 = INmlc[w][a].X1 + (n*p*(INmlc[w+1][a].X1 - INmlc[w][a
                  ].X1));
              cOUTmlc[D][a].X2 = INmlc[w][a].X2 + (n*p*(INmlc[w+1][a].X2 - INmlc[w][a
                  ].X2));

              for(LP = 0; LP < 60; LP++)
                {
                  cOUTmlc[D][a].leafpair[LP][0] = INmlc[w][a].leafpair[LP][0] + (n*p*(
                      INmlc[w+1][a].leafpair[LP][0]-INmlc[w][a].leafpair[LP][0]));
                  cOUTmlc[D][a].leafpair[LP][1] = INmlc[w][a].leafpair[LP][1] + (n*p*(
                      INmlc[w+1][a].leafpair[LP][1]-INmlc[w][a].leafpair[LP][1]));
                }
            }
        }
  }

//for the last tcbcount beam
```

```
for(a = 0; a < arccount; a++)
  {
    cOUTmlc[tcbcount-1][a].MU = INmlc[pbcount+1][a].MU;
    cOUTmlc[tcbcount-1][a].Y2 = INmlc[pbcount+1][a].Y2;
    cOUTmlc[tcbcount-1][a].Y1 = INmlc[pbcount+1][a].Y1;
    cOUTmlc[tcbcount-1][a].X1 = INmlc[pbcount+1][a].X1;
    cOUTmlc[tcbcount-1][a].X2 = INmlc[pbcount+1][a].X2;

    for(LP = 0; LP < 60; LP++)
      {
        cOUTmlc[tcbcount-1][a].leafpair[LP][0] = INmlc[pbcount+1][a].leafpair[LP][0];
        cOUTmlc[tcbcount-1][a].leafpair[LP][1] = INmlc[pbcount+1][a].leafpair[LP][1];
      }
  }

//printing the individual mlc file to the corresponding expanded beams
int J, ix;
int tbcount;                                        //total number of interpolated beams for
    MCKS
tbcount = (tcbcount-1)/2;
dfine = (float)(arcRANGE/tbcount);

//extract the interpolated beams for MCKS from the contintuous sequence

for(x = 0; x < tbcount; x++)
  {
    ix = 2*x + 1;

    for(a = 0; a < arccount; a++)
      {
        OUTmlc[x][a] = cOUTmlc[ix][a];
      }
  }

//writing the output of MLC files
int xx;

for(xx = 0; xx < tbcount; xx++)
  {
    sprintf(out_mlc, "%s/intp_%d.mlc", mlc_path, xx);
    fp[xx] = fopen(out_mlc, "w");

    for(a = 0; a < arccount; a++)
      {
        fprintf(fp[xx], "%d\n", a);
        fprintf(fp[xx], "%0.5f\n", OUTmlc[xx][a].MU);
        fprintf(fp[xx], "%0.1f %3.1f\n %0.1f %3.1f\n", OUTmlc[xx][a].Y2, OUTmlc[xx][a].
            Y1, OUTmlc[xx][a].X1, OUTmlc[xx][a].X2);

        for(J = 1; J <= 60; J++)          //J is the number of leafpair, counting from 1
          {
            int JJ = J -1;
            fprintf(fp[xx], "%d %3.3f %3.3f\n", J, OUTmlc[xx][a].leafpair[JJ][0],
                OUTmlc[xx][a].leafpair[JJ][1]);
          };
      };
    fclose(fp[xx]);
  };

//discretize the beams within the individual arcs with equally averaged weighting (MU)

sprintf(out_file, "newIntp_%s", argv[1]);
edit_file = fopen(out_file, "w");          //open file again and write

int k, cbeamID;
cbeamID = 0;

//managing the continuous beams

//setting the first continuous beam MU to zero

cbeam[0].MU = 0;

for(w = 1; w <= pbcount; w++)
  {
    for(k = 1; k <= (conbeam+1); k++)
      {
        cbeamID ++;

        cbeam[cbeamID].MU = ibeam[w].MU/(conbeam+1);
      }
  }

//rediscretize the continuous beams into the input beams for MCKS

int  temp;
temp = 1;
```

```
  for(k = 0; k < tbcount; k++)
    {
      xbeam[k].MU = cbeam[temp].MU + cbeam[temp+1].MU;
      temp = temp + 2;
    }

  //defining the last beam angle

  //loop for printing interpolated planning beams
  int i, ii;
  int xbeamID;
  xbeamID = -1;                                    //start counting after kick-off and its
       interpolated beams

  for(i = 0; i < tbcount; i++)
    {
      if(i == 0)
        {
          if(ibeam[1].GANT - ibeam[2].GANT < 0)
            {
              //for clockwise plan

              xbeam[0].GANT = ibeam[1].GANT - (dsep/2) + (dfine/2);
            }
          else
            //for anticlockwise plan

            xbeam[0].GANT = ibeam[1].GANT + (dsep/2) - (dcon);
        }
      else
        xbeam[i].GANT = xbeam[i-1].GANT - dfine;

      if(xbeam[i].GANT < 0)
        {
          xbeam[i].GANT = 360 + xbeam[i].GANT;
        }

      fprintf(edit_file,"%d\n %0.4f %3.2f %3.2f %5.2f %3.2f %3.2f\n %s/intp_%d.mlc\n", i,
           xbeam[i].MU, xbeam[i].GANT, coll, ISOx, ISOy, ISOz, mlc_path, i);
    }

  fclose(edit_file);

  printf("Your plan was orginially with angular spacing of %0.2f degree, it has now been
       successfully interpolated to every %0.2f. The interpolated MLC files are in the %s
       directory.\n", dsep, dfine, mlc_path);

  free (ibeam) ;
  free (cbeam);
  free (xbeam) ;

  temp_sl = 0 ;
  while ( temp_sl < INMLC_SIZE ){
    free (INmlc[temp_sl]);
    temp_sl ++ ;
  }
  free ( INmlc );

  temp_sl = 0 ;
  while ( temp_sl < INMLC_SIZE ){
    free (cOUTmlc[temp_sl] );
    temp_sl ++ ;
  }
  free ( cOUTmlc ) ;

  temp_sl = 0 ;
  while ( temp_sl < INMLC_SIZE ){
    free ( OUTmlc[temp_sl] );
    temp_sl ++ ;
  }
  free ( OUTmlc ) ;

  return 0;
}
```

Appendix B

C codes for converting *.3ddose files to *.img files

The codes below were compiled to perform a conversion of the dose files output from MCKS (*.3ddose) to a dose file format in Pinnacle[3](*.img).

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

void swap32(void* p);           //defining th functions of swap32 and swap16
void swap16(void* p);

int main(int argc, char *argv[1])
{
  int dimx, dimy, dimz, i, j, k;
  float dump;

  float *dose;

  dose = new float [6000000];

  FILE *mcksdose;
  FILE *binaryimgfile;
  char img[200];

  if(argc==1)
    {
      printf("Sorry, .3ddose file is not found. Exiting program.\n");
      exit(0);
    }

  mcksdose = fopen(argv[1], "r");

  if(mcksdose == NULL)
    {
      printf("Failed to read file. Exiting program.\n");
    };

  //reading dose form 3ddose file

 fscanf(mcksdose, "%d %d %d", &dimx, &dimy, &dimz);

 for(i = 0; i < dimx; i++)
   {
      fscanf(mcksdose, "%f", &dump);
    }

  for(j = 0; j < dimy; j++)
    {
      fscanf(mcksdose, "%f", &dump);
    }

  for(k = 0; k < dimz; k++)
    {
      fscanf(mcksdose, "%f", &dump);
    }


  for(k = 0; k < dimz; k++)
    {
      for(j = 0; j < dimy; j++)
        {
          for(i = 0; i < dimx; i++)
            {
              long index = (dimx*dimy*k) + (dimx*j) + i;
```

```
                fscanf(mcksdose, "%f", &dose[index]);
              }
          }
      }

    fclose(mcksdose);

    float temp_float;

    sprintf(img, "%s.binary.img", argv[1]);
    binaryimgfile = fopen(img, "wb");         //open file again and write

    for(k = 0; k < dimz; k++)
      {
        for(j = 0; j < dimy; j++)
          {
            for(i = 0; i < dimx; i++)
              {
                long index = (dimx*dimy*k) + (dimx*j) + i;

                temp_float = dose[index];
                swap32(&temp_float);
                dose[index] = temp_float;
              }
          }
      }

    long num_items = dimx*dimy*dimz;

    for(i = 0; i<num_items; i++)
      {
        fwrite(&dose[i], sizeof(float), 1, binaryimgfile);
      }

    fclose(binaryimgfile);

    return 0;

}

void swap32(void* p)
{
    swap16(p);
    swap16((short*)p+1);
    ((unsigned short*)p)[0]^=((unsigned short*)p)[1];
    ((unsigned short*)p)[1]^=((unsigned short*)p)[0];
    ((unsigned short*)p)[0]^=((unsigned short*)p)[1];
}

void swap16(void* p)
{
    ((unsigned char*)p)[0]^=((unsigned char*)p)[1];
    ((unsigned char*)p)[1]^=((unsigned char*)p)[0];
    ((unsigned char*)p)[0]^=((unsigned char*)p)[1];
}
```

Appendix C

C codes for converting multi-arc IMAT into single-arc IMAT (A)

The codes below were compiled to convert multi-IMAT plans into singe-arc IMAT plans without geometric correction to the shifted beams.

```
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include <math.h>
#include <ctype.h>

#define IN_SIZE 40
#define ARC_SIZE 12
#define OUT_SIZE 500

struct iBEAM                //input beams
{
  float MU;                 //monitor units of the individual beam
  float GANT;               //gantry angle of beam
  char MLC_FILE[60];        //mlc file
};

struct sBEAM                //output beams
{
  float MU;                 //monitor units of the individual beam
  float GANT;               //gantry angle of beam
};

struct inMLC
{
  int ID;                  //control point index
  float MU;                //relative MU
  float Y2;                //Y2 back-up jaw position
  float Y1;                //Y1 back-up jaw position
  float X1;                //X1 back-up jaw position
  float X2;                //X2 back-up jaw position
  int lp[60];                  //leafpair ID, count from 1 to 60
  float leafpair[60][2];       //leafpair positions
  float eAreaT[10];            //elementary area determined by the top 10 leafpairs
  float eAreaM[40];            //elementary area determined by the middle 40 leafpairs
  float eAreaB[10];            //elementary area determined by the bottom 10 leafpairs
  float SeAreaT;               //sum of eAreaT
  float SeAreaM;               //sum of eAreaM
  float SeAreaB;               //sum of eAreaB
  float area;                  //total area of aperture; sum of SeAreaT, SeAreaM and
      SeAreaB
  float eMomentXT[10];          //elementary first moment of opening area determmined by
      the top 10 lp about the y-axis
  float eMomentXM[40];          //elementary first moment of opening area determmined by
      the middle 40 lp about the y-axis
  float eMomentXB[10];          //elementary first moment of opening area determmined by
      the bottom 10 lp about the y-axis
  float eMomentYT[10];          //elementary first moment of opening area determmined by
      the top 10 lp about the x-axis
  float eMomentYM[40];          //elementary first moment of opening area determmined by
      the middle 40 lp about the x-axis
  float eMomentYB[10];          //elementary first moment of opening area determmined by
      the bottom 10 lp about the x-axis
  float SeMomentXT;             //sum of eMomentXT
  float SeMomentXM;             //sum of eMomentXM
  float SeMomentXB;             //sum of eMomentXB
  float SeMomentYT;             //sum of eMomentYT
  float SeMomentYM;             //sum of eMomentYM
  float SeMomentYB;             //sum of eMomentYB
  float momentX;               //the total first moment of aperture relative to the y-axis
  float momentY;               //the total first moment of aperture relative to the x-axis
```

```
  float eCentroidXT[10];          //the x-coordinate of elementary centroid of the top 10
      lp
  float eCentroidXM[40];          //the x-coordinate of elementary centroid of the middle
      40 lp
  float eCentroidXB[10];          //the x-coordinate of elementary centroid of the bottom
      10 lp
  float eCentroidYT[10];          //the y-coordinate of elementary centroid of the top 10
      lp
  float eCentroidYM[40];          //the y-coordinate of elementary centroid of the middle
      40 lp
  float eCentroidYB[10];          //the y-coordinate of elementary centroid of the bottom
      10 lp
  float centroidX;            //the x-coordinate of the centroid of aperture
  float centroidY;            //the y-coordinate of the centroid of aperture
};

struct outMLC
{
  float MU;               //relative MU
  float Y2;               //Y2 back-up jaw position
  float Y1;               //Y1 back-up jaw position
  float X1;               //X1 back-up jaw position
  float X2;               //X2 back-up jaw position
  float leafpair[60][2];          //leafpair positions
  float centroidX;                //the x-coordinate of the centroid of aperture
};

struct LEFTMOST           //groups for left to right sorting
{
  float MU;               //relative MU
  float Y2;               //Y2 back-up jaw position
  float Y1;               //Y1 back-up jaw position
  float X1;               //X1 back-up jaw position
  float X2;               //X2 back-up jaw position
  float leafpair[60][2];          //leafpair positions
  float centroidX;                //the x-coordinate of the centroid of aperture
};
struct RIGHTMOST          //groups for right to left sorting
{
  float MU;               //relative MU
  float Y2;               //Y2 back-up jaw position
  float Y1;               //Y1 back-up jaw position
  float X1;               //X1 back-up jaw position
  float X2;               //X2 back-up jaw position
  float leafpair[60][2];          //leafpair positions
  float centroidX;                //the x-coordinate of the centroid of aperture
};

//setting parameters for program execution

int main(int argc, char *argv[])
{
  FILE *read_file;
  FILE *read_mlc;
  FILE *edit_file;
  FILE *edit_mlc;
  FILE *fp[OUT_SIZE];         //file pointer array for OUTmlc files
  char in_mlc[50];
  char out_file[50];
  char out_mlc[50];
  char mlc_path[50];                    //specifying path of the folder for MLC files
  struct iBEAM ibeam[IN_SIZE];
  struct sBEAM sbeam[OUT_SIZE];
  struct inMLC INmlc[IN_SIZE][ARC_SIZE];
  struct outMLC OUTmlc[OUT_SIZE];
  struct LEFTMOST leftmost[IN_SIZE][ARC_SIZE];
  struct RIGHTMOST rightmost[IN_SIZE][ARC_SIZE];
  int a, m, ibeamID, arcID;
  float coll;               //collimator angle
  float ISOx, ISOy, ISOz;       //3D coordinates of isocentre

  if(argc==1)
    {
      printf("Sorry, .beams file is not found. Exiting program.\n");

      exit(0);
    }

  read_file  = fopen(argv[1], "r");

  int pbcount = 0;                      //pbcount is the total number of input beams
  int segcount = 0;                     //counting total number of input segments
  int arccount = 0;                     //counting number of arcs

  while(fscanf(read_file, "%d", &ibeamID) !=EOF)
    {
      fscanf(read_file, "%f %f %f %f %f %s", &ibeam[ibeamID].MU, &ibeam[ibeamID].GANT,
          &coll, &ISOx, &ISOy, &ISOz, &in_mlc);
```

```
    read_mlc = fopen(in_mlc, "r");

    while(fscanf(read_mlc, "%d", &arcID) != EOF)
      {
        fscanf(read_mlc, "%f %f %f %f %f", &INmlc[ibeamID][arcID].MU, &INmlc[ibeamID][
            arcID].Y2, &INmlc[ibeamID][arcID].Y1, &INmlc[ibeamID][arcID].X1, &INmlc[
            ibeamID][arcID].X2);

        for(m = 0; m < 60; m++)
          {
            fscanf(read_mlc, "%d %f %f", &INmlc[ibeamID][arcID].lp, &INmlc[ibeamID][
                arcID].leafpair[m][0], &INmlc[ibeamID][arcID].leafpair[m][1]);
          }

        segcount++;
      }
    fclose(read_mlc);
    pbcount++;
  }
fclose(read_file);

arccount = segcount/pbcount;

float dsep, dspread;                          //planning and spreading angular separation
    in degrees

dsep = fabs(ibeam[1].GANT - ibeam[0].GANT);
dspread = dsep/arccount;

//re-normalised the MU of the control points

int r, rr;

for(r = 0; r < pbcount; r++)
  {
    for(rr = 0; rr < arccount; rr++)
      {
        INmlc[r][rr].MU = INmlc[r][rr].MU/ibeam[r].MU;
      }
  }

//for Varian Millenium MLC, the top (leafpair 1 to 10) and bottom (leafpair 51 - 60)
    have heights of 1 cm and the middle part of the collimator system (leafpar 11 - 50)
     has MLC leaf heights of 0.5 cm.

//for leafpairs 1 to 10 = eAreaT
int x, xx, t, w, n, D, M, LP, h, cx;

for(w = 0; w < pbcount; w++)
  {
    for(n = 0; n < arccount; n++)
      {
        for(x = 0; x < 10; x++)
          {
            cx = 40 - x;

            INmlc[w][n].eCentroidXT[x] = (INmlc[w][n].leafpair[x][1] + INmlc[w][n].
                leafpair[x][0])/2;
            INmlc[w][n].eCentroidYT[x] = cx - 0.5;
            INmlc[w][n].eAreaT[x] = (INmlc[w][n].leafpair[x][1] - INmlc[w][n].leafpair[
                x][0])*1;
            INmlc[w][n].SeAreaT += INmlc[w][n].eAreaT[x];
            INmlc[w][n].eMomentXT[x] = INmlc[w][n].eCentroidXT[x]*INmlc[w][n].eAreaT[x
                ];
            INmlc[w][n].eMomentYT[x] = INmlc[w][n].eCentroidYT[x]*INmlc[w][n].eAreaT[x
                ];
            INmlc[w][n].SeMomentXT += INmlc[w][n].eMomentXT[x];
            INmlc[w][n].SeMomentYT += INmlc[w][n].eMomentYT[x];
          }
      }
  }

//for leafpairs 11 - 50 = eAreaM

for(w = 0; w < pbcount; w++)
  {
    for(n = 0; n < arccount; n++)
      {
        for(xx = 0; xx < 40; xx++)
          {
            x = xx + 10;

            INmlc[w][n].eCentroidXM[x] = (INmlc[w][n].leafpair[x][1] + INmlc[w][n].
                leafpair[x][0])/2;
            INmlc[w][n].eCentroidYM[x] = 29.75 - (0.5*xx);
            INmlc[w][n].eAreaM[x] = (INmlc[w][n].leafpair[x][1] - INmlc[w][n].leafpair[
                x][0])*1;
```

```
            INmlc[w][n].SeAreaM += INmlc[w][n].eAreaM[x];
            INmlc[w][n].eMomentXM[x] = INmlc[w][n].eCentroidXM[x] * INmlc[w][n].eAreaM[
                x];
            INmlc[w][n].eMomentYM[x] = INmlc[w][n].eCentroidYM[x] * INmlc[w][n].eAreaM[
                x];
            INmlc[w][n].SeMomentXM += INmlc[w][n].eMomentXM[x];
            INmlc[w][n].SeMomentYM += INmlc[w][n].eMomentYM[x];

          }
      }
  }

//for leafpairs 51 - 60 = eAreaB

for(w = 0; w < pbcount; w++)
  {
    for(n = 0; n < arccount; n++)
      {
        for(xx = 0; xx < 10; xx++)
          {
            x = xx + 50;
            cx = 10 - xx;

            INmlc[w][n].eCentroidXB[x] = (INmlc[w][n].leafpair[x][1] + INmlc[w][n].
                leafpair[x][0])/2;
            INmlc[w][n].eCentroidYB[x] = cx - 0.5;
            INmlc[w][n].eAreaB[x] = (INmlc[w][n].leafpair[x][1] - INmlc[w][n].leafpair[
                x][0])*1;
            INmlc[w][n].SeAreaB += INmlc[w][n].eAreaB[x];
            INmlc[w][n].eMomentXB[x] = INmlc[w][n].eCentroidXB[x]*INmlc[w][n].eAreaB[x
                ];
            INmlc[w][n].eMomentYB[x] = INmlc[w][n].eCentroidYB[x]*INmlc[w][n].eAreaB[x
                ];
            INmlc[w][n].SeMomentXB += INmlc[w][n].eMomentXB[x];
            INmlc[w][n].SeMomentYB += INmlc[w][n].eMomentYB[x];
          }
      }
  }

for(w = 0; w < pbcount; w++)
  {
    for(n = 0; n < arccount; n++)
      {
        INmlc[w][n].area = INmlc[w][n].SeAreaT + INmlc[w][n].SeAreaM + INmlc[w][n].
            SeAreaB;
        INmlc[w][n].momentX = INmlc[w][n].SeMomentXT + INmlc[w][n].SeMomentXM + INmlc[w
            ][n].SeMomentXB;
        INmlc[w][n].momentY = INmlc[w][n].SeMomentYT + INmlc[w][n].SeMomentYM + INmlc[w
            ][n].SeMomentYB;
        INmlc[w][n].centroidX = INmlc[w][n].momentX/INmlc[w][n].area;
        INmlc[w][n].centroidY = INmlc[w][n].momentY/INmlc[w][n].area;
      }
  }

//group spreading beams into leftmost struct and rightmost struct

int ww = -1;

for(w = 0; w < pbcount; w+=2)
  {
    ww++;
    for(n = 0; n < arccount; n++)
      {
        leftmost[ww][n].MU = INmlc[w][n].MU;
        leftmost[ww][n].Y2 = INmlc[w][n].Y2;
        leftmost[ww][n].Y1 = INmlc[w][n].Y1;
        leftmost[ww][n].X1 = INmlc[w][n].X1;
        leftmost[ww][n].X2 = INmlc[w][n].X2;
        leftmost[ww][n].centroidX = INmlc[w][n].centroidX;

        for(h = 0; h < 60; h++)
          {
            leftmost[ww][n].leafpair[h][0] = INmlc[w][n].leafpair[h][0];
            leftmost[ww][n].leafpair[h][1] = INmlc[w][n].leafpair[h][1];
          }

        rightmost[ww][n].MU = INmlc[w+1][n].MU;
        rightmost[ww][n].Y2 = INmlc[w+1][n].Y2;
        rightmost[ww][n].Y1 = INmlc[w+1][n].Y1;
        rightmost[ww][n].X2 = INmlc[w+1][n].X2;
        rightmost[ww][n].X1 = INmlc[w+1][n].X1;
        rightmost[ww][n].centroidX = INmlc[w+1][n].centroidX;

        for(h = 0; h < 60; h++)
          {
            rightmost[ww][n].leafpair[h][0] = INmlc[w+1][n].leafpair[h][0];
            rightmost[ww][n].leafpair[h][1] = INmlc[w+1][n].leafpair[h][1];
          }
```

```
      }
  }

//comparing only the x-coordinate of the centroid of apertures and sorting them in
     designed order and assign the appropriate spreading angle

//sorting leftmost to rightmost

int sort = pbcount/2;
struct LEFTMOST lhold;
int k, kk;

for(w = 0; w <=sort; w++)
  {
    for(k = arccount-1; k > 0; k--)
      {
        for(n = 0; n < k; n++)
          {
            if (leftmost[w][n].centroidX > leftmost[w][n+1].centroidX)
              {
                lhold  = leftmost[w][n];
                leftmost[w][n] = leftmost[w][n+1];
                leftmost[w][n+1] = lhold;
              }
          }
      }
  }

//sorting rightmost to leftmost

struct RIGHTMOST rhold;

for(w = 0; w <=sort; w++)
  {
    for(k = arccount-1; k > 0; k--)                    //bubble sort loop
      {
        for(n = 0; n < k; n++)
          {
            if (rightmost[w][n].centroidX < rightmost[w][n+1].centroidX)
              {
                rhold = rightmost[w][n];
                rightmost[w][n] = rightmost[w][n+1];
                rightmost[w][n+1] = rhold;
              }
          }
      }
  }

//transferring leftmost and right structs to OUTmlc

int wt, tt, lt, rt;
lt = -1;
wt = -1;

 for(w = 0; w < pbcount/2; w++)
  {
    for(n = 0; n < arccount; n++)
      {
        wt ++;
        lt = wt + (w*arccount);
        rt = lt + arccount;

        OUTmlc[lt].MU = leftmost[w][n].MU;
        OUTmlc[lt].Y2 = leftmost[w][n].Y2;
        OUTmlc[lt].Y1 = leftmost[w][n].Y1;
        OUTmlc[lt].X1 = leftmost[w][n].X1;
        OUTmlc[lt].X2 = leftmost[w][n].X2;
        OUTmlc[lt].centroidX = leftmost[w][n].centroidX;

        for(h = 0; h < 60; h++)
          {
            OUTmlc[lt].leafpair[h][0] = leftmost[w][n].leafpair[h][0];
            OUTmlc[lt].leafpair[h][1] = leftmost[w][n].leafpair[h][1];
          }

        OUTmlc[rt].MU = rightmost[w][n].MU;
        OUTmlc[rt].Y2 = rightmost[w][n].Y2;
        OUTmlc[rt].Y1 = rightmost[w][n].Y1;
        OUTmlc[rt].X1 = rightmost[w][n].X1;
        OUTmlc[rt].X2 = rightmost[w][n].X2;
        OUTmlc[rt].centroidX = rightmost[w][n].centroidX;

        for(h = 0; h < 60; h++)
          {
            OUTmlc[rt].leafpair[h][0] = rightmost[w][n].leafpair[h][0];
            OUTmlc[rt].leafpair[h][1] = rightmost[w][n].leafpair[h][1];
          }
      }
```

```
  }

 //assigning MU to the individual spreading beams

 int nw;

 for(w = 0; w < pbcount; w++)
   {
     for(n = 0; n < arccount; n++)
       {
         nw = (w*arccount) + n;
         sbeam[nw].MU = ibeam[w].MU*OUTmlc[nw].MU;
       }
   }

//printing the individual mlc file to the corresonding spreading beams
int J;

for(x = 0; x < segcount; x++)
  {
    sprintf(out_mlc, "spread_MLC/spread_%d.mlc", x);
    fp[x] = fopen(out_mlc, "w");

    fprintf(fp[x], "%d\n", x);
    fprintf(fp[x], "1.00\n");              //all MLC has relative MU of 1
    fprintf(fp[x], "%0.1f %3.1f\n %0.1f %3.1f\n", OUTmlc[x].Y2, OUTmlc[x].Y1, OUTmlc[x
        ].X1, OUTmlc[x].X2);

    for(J = 1; J <= 60; J++)          //J is the number of leafpair, counting from 1
      {
        int JJ = J -1;
        fprintf(fp[x], "%d %3.3f %3.3f\n", J, OUTmlc[x].leafpair[JJ][0], OUTmlc[x].
            leafpair[JJ][1]);
      };
    fclose(fp[x]);
  }

//printing .beams file
sprintf(out_file, "spread_%s", argv[1]);
edit_file = fopen(out_file, "w");          //open file again and write

int i;
int sbeamID = -1;                           //always count beams from zero
float sbeamGANT;
sbeamGANT = ibeam[0].GANT + (((arccount-1)/2)*dspread) + dspread;        //counting anti
    -clockwise

if (arccount == 2 || arccount == 4 || arccount == 6 || arccount == 8 || arccount == 10)
       //for even number arcs
  {
    if (ibeam[0].GANT - ibeam[1].GANT < 0)          //for clockwise arcs
      {
        sbeamGANT = ibeam[0].GANT - ((((arccount/2)-1)*dspread) + (dspread/2)) +
            dspread;
      }

    else
      sbeamGANT = ibeam[0].GANT + ((((arccount/2)-1)*dspread) + (dspread/2)) + dspread;
                //for anticlockwise arcs
  }

else                    //for odd number arcs
  {

    if (ibeam[0].GANT - ibeam[1].GANT < 0)          //for clockwise arcs
      {
        sbeamGANT = ibeam[0].GANT - (fabs((arccount-1)/2)*dspread) + dspread;
            //for clockwise arcs
      }

    else
      sbeamGANT = ibeam[0].GANT + (fabs((arccount-1)/2)*dspread) + dspread;
          //for anticlockwise arcs
  }

//loop for counting spreading beams (sbeamID)
for(i = 0; i < segcount; i ++)
  {
    sbeamID++ ;

    if (ibeam[0].GANT - ibeam[1].GANT < 0)          //clockwise
      {
        sbeamGANT+= dspread;                //counting clockwise
      }
    else
      sbeamGANT-= dspread;                  //counting anticlockwise

    if (sbeamGANT < 0)
```

```
        {
          sbeamGANT = 360 + sbeamGANT;
        }
      fprintf(edit_file,"%d\n %0.4f %3.2f %3.2f %5.2f %3.2f %3.2f\n spread_MLC/spread_%d.
          mlc\n\n", sbeamID, sbeam[i].MU, sbeamGANT, coll, ISOx, ISOy, ISOz, sbeamID);
    };

  fclose(edit_file);

  printf("Your stacking plan has now been successfully spread!\n");

  return 0;

}
```

Appendix D

C codes for converting multi-arc IMAT into single-arc IMAT (B)

The codes below were compiled to convert multi-IMAT plans into singe-arc IMAT plans with geometric correction to the shifted beams.

```
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include <math.h>
#include <ctype.h>

#define IN_SIZE 40
#define ARC_SIZE 12
#define OUT_SIZE 500
#define Pi 3.14159265

struct iBEAM                //input beams
{
  float MU;                 //monitor units of the individual beam
  float GANT;               //gantry angle of beam
  char MLC_FILE[60];        //mlc file
};

struct sBEAM                //output beams
{
  float MU;                 //monitor units of the individual beam
  float GANT;               //gantry angle of beam
};

struct inMLC
{
  int ID;                  //control point index
  float MU;                //relative MU
  float Y2;                //Y2 back-up jaw position
  float Y1;                //Y1 back-up jaw position
  float X1;                //X1 back-up jaw position
  float X2;                //X2 back-up jaw position
  int lp[60];                 //leafpair ID, count from 1 to 60
  float leafpair[60][2];      //leafpair positions
  float eAreaT[10];           //elementary area determined by the top 10 leafpairs
  float eAreaM[40];           //elementary area determined by the middle 40 leafpairs
  float eAreaB[10];           //elementary area determined by the bottom 10 leafpairs
  float SeAreaT;              //sum of eAreaT
  float SeAreaM;              //sum of eAreaM
  float SeAreaB;              //sum of eAreaB
  float area;                 //total area of aperture; sum of SeAreaT, SeAreaM and
      SeAreaB
  float eMomentXT[10];          //elementary first moment of opening area determmined by
      the top 10 lp about the y-axis
  float eMomentXM[40];          //elementary first moment of opening area determmined by
      the middle 40 lp about the y-axis
  float eMomentXB[10];          //elementary first moment of opening area determmined by
      the bottom 10 lp about the y-axis
  float eMomentYT[10];          //elementary first moment of opening area determmined by
      the top 10 lp about the x-axis
  float eMomentYM[40];          //elementary first moment of opening area determmined by
      the middle 40 lp about the x-axis
  float eMomentYB[10];          //elementary first moment of opening area determmined by
      the bottom 10 lp about the x-axis
  float SeMomentXT;             //sum of eMomentXT
  float SeMomentXM;             //sum of eMomentXM
  float SeMomentXB;             //sum of eMomentXB
  float SeMomentYT;             //sum of eMomentYT
  float SeMomentYM;             //sum of eMomentYM
  float SeMomentYB;             //sum of eMomentYB
  float momentX;               //the total first moment of aperture relative to the y-axis
```

```
  float momentY;              //the total first moment of aperture relative to the x-axis
  float eCentroidXT[10];          //the x-coordinate of elementary centroid of the top 10
      lp
  float eCentroidXM[40];          //the x-coordinate of elementary centroid of the middle
      40 lp
  float eCentroidXB[10];          //the x-coordinate of elementary centroid of the bottom
      10 lp
  float eCentroidYT[10];          //the y-coordinate of elementary centroid of the top 10
      lp
  float eCentroidYM[40];          //the y-coordinate of elementary centroid of the middle
      40 lp
  float eCentroidYB[10];          //the y-coordinate of elementary centroid of the bottom
      10 lp
  float centroidX;            //the x-coordinate of the centroid of aperture
  float centroidY;            //the y-coordinate of the centroid of aperture
};

struct outMLC
{
  float MU;               //relative MU
  float Y2;               //Y2 back-up jaw position
  float Y1;               //Y1 back-up jaw position
  float X1;               //X1 back-up jaw position
  float X2;               //X2 back-up jaw position
  float leafpair[60][2];          //leafpair positions
  float centroidX;                //the x-coordinate of the centroid of aperture
};

struct LEFTMOST           //groups for left to right sorting
{
  float MU;               //relative MU
  float Y2;               //Y2 back-up jaw position
  float Y1;               //Y1 back-up jaw position
  float X1;               //X1 back-up jaw position
  float X2;               //X2 back-up jaw position
  float leafpair[60][2];          //leafpair positions
  float centroidX;                //the x-coordinate of the centroid of aperture
};
struct RIGHTMOST          //groups for right to left sorting
{
  float MU;               //relative MU
  float Y2;               //Y2 back-up jaw position
  float Y1;               //Y1 back-up jaw position
  float X1;               //X1 back-up jaw position
  float X2;               //X2 back-up jaw position
  float leafpair[60][2];          //leafpair positions
  float centroidX;                //the x-coordinate of the centroid of aperture
};

//setting parameters for program execution

int main(int argc, char *argv[])
{
  FILE *read_file;
  FILE *read_mlc;
  FILE *edit_file;
  FILE *edit_mlc;
  FILE *fp[OUT_SIZE];         //file pointer array for OUTmlc files
  char in_mlc[50];
  char out_file[50];
  char out_mlc[50];
  char mlc_path[50];                    //specifying path of the folder for MLC files
  struct iBEAM ibeam[IN_SIZE];
  struct sBEAM sbeam[OUT_SIZE];
  struct inMLC INmlc[IN_SIZE][ARC_SIZE];
  struct outMLC OUTmlc[OUT_SIZE];
  struct LEFTMOST leftmost[IN_SIZE][ARC_SIZE];
  struct RIGHTMOST rightmost[IN_SIZE][ARC_SIZE];
  int a, m, ibeamID, arcID;
  float coll;               //collimator angle
  float ISOx, ISOy, ISOz;       //3D coordinates of isocentre
  float SAD;

  if(argc==1)
    {
      printf("Sorry, .beams file is not found. Exiting program.\n");

      exit(0);
    }

  printf("What is the SAD of the plan? (in cm)\n");
  scanf("%f", &SAD);


  read_file  = fopen(argv[1], "r");

  int pbcount = 0;                      //pbcount is the total number of input beams
  int segcount = 0;                     //counting total number of input segments
```

```
int arccount = 0;                    //counting number of arcs

while(fscanf(read_file, "%d", &ibeamID) !=EOF)
  {
    fscanf(read_file, "%f %f %f %f %f %f %s", &ibeam[ibeamID].MU, &ibeam[ibeamID].GANT,
          &coll, &ISOx, &ISOy, &ISOz, &in_mlc);

    read_mlc = fopen(in_mlc, "r");

    while(fscanf(read_mlc, "%d", &arcID) != EOF)
      {
        fscanf(read_mlc, "%f %f %f %f %f", &INmlc[ibeamID][arcID].MU, &INmlc[ibeamID][
            arcID].Y2, &INmlc[ibeamID][arcID].Y1, &INmlc[ibeamID][arcID].X1, &INmlc[
            ibeamID][arcID].X2);

        for(m = 0; m < 60; m++)
          {
            fscanf(read_mlc, "%d %f %f", &INmlc[ibeamID][arcID].lp, &INmlc[ibeamID][
                arcID].leafpair[m][0], &INmlc[ibeamID][arcID].leafpair[m][1]);
          }

        segcount++;
      }
    fclose(read_mlc);
    pbcount++;
  }
fclose(read_file);

arccount = segcount/pbcount;

float dsep, dspread;                 //planning and spreading angular separation
    in degrees

dsep = fabs(ibeam[1].GANT - ibeam[0].GANT);
dspread = dsep/arccount;

//re-normalised the MU of the control points

int r, rr;

for(r = 0; r < pbcount; r++)
  {
    for(rr = 0; rr < arccount; rr++)
      {
        INmlc[r][rr].MU = INmlc[r][rr].MU/ibeam[r].MU;
        printf("INmlc[%d][%d].MU %f\n", r, rr, INmlc[r][rr].MU);
      }
  }

//for Varian Millenium MLC, the top (leafpair 1 to 10) and bottom (leafpair 51 - 60)
    have heights of 1 cm and the middle part of the collimator system (leafpar 11 - 50)
     has MLC leaf heights of 0.5 cm.

//for leafpairs 1 to 10 = eAreaT
int x, xx, t, w, n, D, M, LP, h, cx;

for(w = 0; w < pbcount; w++)
  {
    for(n = 0; n < arccount; n++)
      {
        for(x = 0; x < 10; x++)
          {
            cx = 40 - x;

            INmlc[w][n].eCentroidXT[x] = (INmlc[w][n].leafpair[x][1] + INmlc[w][n].
                leafpair[x][0])/2;
            INmlc[w][n].eCentroidYT[x] = cx - 0.5;
            INmlc[w][n].eAreaT[x] = (INmlc[w][n].leafpair[x][1] - INmlc[w][n].leafpair[
                x][0])*1;
            INmlc[w][n].SeAreaT += INmlc[w][n].eAreaT[x];
            INmlc[w][n].eMomentXT[x] = INmlc[w][n].eCentroidXT[x]*INmlc[w][n].eAreaT[x
                ];
            INmlc[w][n].eMomentYT[x] = INmlc[w][n].eCentroidYT[x]*INmlc[w][n].eAreaT[x
                ];
            INmlc[w][n].SeMomentXT += INmlc[w][n].eMomentXT[x];
            INmlc[w][n].SeMomentYT += INmlc[w][n].eMomentYT[x];
          }
      }
  }

//for leafpairs 11 - 50 = eAreaM

for(w = 0; w < pbcount; w++)
  {
    for(n = 0; n < arccount; n++)
      {
        for(xx = 0; xx < 40; xx++)
          {
```

```
            x = xx + 10;

           INmlc[w][n].eCentroidXM[x] = (INmlc[w][n].leafpair[x][1] + INmlc[w][n].
                leafpair[x][0])/2;
            INmlc[w][n].eCentroidYM[x] = 29.75 - (0.5*xx);
            INmlc[w][n].eAreaM[x] = (INmlc[w][n].leafpair[x][1] - INmlc[w][n].leafpair[
                x][0])*1;
            INmlc[w][n].SeAreaM += INmlc[w][n].eAreaM[x];
            INmlc[w][n].eMomentXM[x] = INmlc[w][n].eCentroidXM[x] * INmlc[w][n].eAreaM[
                x];
            INmlc[w][n].eMomentYM[x] = INmlc[w][n].eCentroidYM[x] * INmlc[w][n].eAreaM[
                x];
            INmlc[w][n].SeMomentXM += INmlc[w][n].eMomentXM[x];
            INmlc[w][n].SeMomentYM += INmlc[w][n].eMomentYM[x];

          }
      }
  }

//for leafpairs 51 - 60 = eAreaB

for(w = 0; w < pbcount; w++)
  {
    for(n = 0; n < arccount; n++)
      {
        for(xx = 0; xx < 10; xx++)
          {
            x = xx + 50;
            cx = 10 - xx;

            INmlc[w][n].eCentroidXB[x] = (INmlc[w][n].leafpair[x][1] + INmlc[w][n].
                leafpair[x][0])/2;
            INmlc[w][n].eCentroidYB[x] = cx - 0.5;
            INmlc[w][n].eAreaB[x] = (INmlc[w][n].leafpair[x][1] - INmlc[w][n].leafpair[
                x][0])*1;
            INmlc[w][n].SeAreaB += INmlc[w][n].eAreaB[x];
            INmlc[w][n].eMomentXB[x] = INmlc[w][n].eCentroidXB[x]*INmlc[w][n].eAreaB[x
                ];
             INmlc[w][n].eMomentYB[x] = INmlc[w][n].eCentroidYB[x]*INmlc[w][n].eAreaB[x
                 ];
             INmlc[w][n].SeMomentXB += INmlc[w][n].eMomentXB[x];
             INmlc[w][n].SeMomentYB += INmlc[w][n].eMomentYB[x];
          }
      }
  }

for(w = 0; w < pbcount; w++)
  {
    for(n = 0; n < arccount; n++)
      {
        INmlc[w][n].area = INmlc[w][n].SeAreaT + INmlc[w][n].SeAreaM + INmlc[w][n].
            SeAreaB;
        INmlc[w][n].momentX = INmlc[w][n].SeMomentXT + INmlc[w][n].SeMomentXM + INmlc[w
            ][n].SeMomentXB;
        INmlc[w][n].momentY = INmlc[w][n].SeMomentYT + INmlc[w][n].SeMomentYM + INmlc[w
            ][n].SeMomentYB;
        INmlc[w][n].centroidX = INmlc[w][n].momentX/INmlc[w][n].area;
        INmlc[w][n].centroidY = INmlc[w][n].momentY/INmlc[w][n].area;
      }
  }

//group spreading beams into leftmost struct and rightmost struct

int ww = -1;

for(w = 0; w < pbcount; w+=2)
  {
    ww++;

    for(n = 0; n < arccount; n++)
      {
        leftmost[ww][n].MU = INmlc[w][n].MU;
        leftmost[ww][n].Y2 = INmlc[w][n].Y2;
        leftmost[ww][n].Y1 = INmlc[w][n].Y1;
        leftmost[ww][n].X1 = INmlc[w][n].X1;
        leftmost[ww][n].X2 = INmlc[w][n].X2;
        leftmost[ww][n].centroidX = INmlc[w][n].centroidX;

        for(h = 0; h < 60; h++)
          {
            leftmost[ww][n].leafpair[h][0] = INmlc[w][n].leafpair[h][0];
            leftmost[ww][n].leafpair[h][1] = INmlc[w][n].leafpair[h][1];
          }

        rightmost[ww][n].MU = INmlc[w+1][n].MU;
        rightmost[ww][n].Y2 = INmlc[w+1][n].Y2;
        rightmost[ww][n].Y1 = INmlc[w+1][n].Y1;
        rightmost[ww][n].X2 = INmlc[w+1][n].X2;
```

```
          rightmost[ww][n].X1 = INmlc[w+1][n].X1;
          rightmost[ww][n].centroidX = INmlc[w+1][n].centroidX;

          for(h = 0; h < 60; h++)
            {
              rightmost[ww][n].leafpair[h][0] = INmlc[w+1][n].leafpair[h][0];
              rightmost[ww][n].leafpair[h][1] = INmlc[w+1][n].leafpair[h][1];
            }
        }
  }

//comparing only the x-coordinate of the centroid of apertures and sorting them in
     designed order and assign the appropriate spreading angle

//sorting leftmost to rightmost

int sort = pbcount/2;
struct LEFTMOST lhold;
int k, kk;

for(w = 0; w <=sort; w++)
  {
    for(k = arccount-1; k > 0; k--)
      {
        for(n = 0; n < k; n++)
          {
            if (leftmost[w][n].centroidX > leftmost[w][n+1].centroidX)
              {
                lhold  = leftmost[w][n];
                leftmost[w][n] = leftmost[w][n+1];
                leftmost[w][n+1] = lhold;
              }
          }
      }
  }

//sorting rightmost to leftmost

struct RIGHTMOST rhold;

for(w = 0; w <=sort; w++)
  {
    for(k = arccount-1; k > 0; k--)                    //bubble sort loop
      {
        for(n = 0; n < k; n++)
          {
            if (rightmost[w][n].centroidX < rightmost[w][n+1].centroidX)
              {
                rhold = rightmost[w][n];
                rightmost[w][n] = rightmost[w][n+1];
                rightmost[w][n+1] = rhold;
              }
          }
      }
  }

//transferring leftmost and right structs to OUTmlc

int wt, tt, lt, rt;
lt = -1;
wt = -1;

 for(w = 0; w < pbcount/2; w++)
  {
    for(n = 0; n < arccount; n++)
      {
        wt ++;
        lt = wt + (w*arccount);
        rt = lt + arccount;

        OUTmlc[lt].MU = leftmost[w][n].MU;
        OUTmlc[lt].Y2 = leftmost[w][n].Y2;
        OUTmlc[lt].Y1 = leftmost[w][n].Y1;
        OUTmlc[lt].X1 = leftmost[w][n].X1;
        OUTmlc[lt].X2 = leftmost[w][n].X2;

        for(h = 0; h < 60; h++)
          {
            OUTmlc[lt].leafpair[h][0] = leftmost[w][n].leafpair[h][0];
            OUTmlc[lt].leafpair[h][1] = leftmost[w][n].leafpair[h][1];
          }

       OUTmlc[rt].MU = rightmost[w][n].MU;
        OUTmlc[rt].Y2 = rightmost[w][n].Y2;
        OUTmlc[rt].Y1 = rightmost[w][n].Y1;
        OUTmlc[rt].X1 = rightmost[w][n].X1;
        OUTmlc[rt].X2 = rightmost[w][n].X2;
```

```
        for(h = 0; h < 60; h++)
          {
            OUTmlc[rt].leafpair[h][0] = rightmost[w][n].leafpair[h][0];
            OUTmlc[rt].leafpair[h][1] = rightmost[w][n].leafpair[h][1];
          }

      }
  }

 //field size correction

 int L = fabs(arccount/2);           //number of spread beams on the left side of the
      plan beam angle
 int R = fabs(arccount/2);           //number of spread beams on the right side of the
      plan beam angle
 int wL, wR;
 float rspread;                      //spreading angle in radians
 rspread = dspread*(Pi/180);

 for(w = 0; w < segcount; w += arccount)
   {
     //correct for the spread beams on the left side of the plan beam angle

     for(n = 0; n < L; n++)
       {
         wL = w + n;

         for(h = 0; h < 60; h++)
           {
             //correct for shortened field size on the left MLC
             OUTmlc[wL].leafpair[h][0] = SAD*((OUTmlc[wL].leafpair[h][0]*cos(rspread))
                  /(SAD - (OUTmlc[wL].leafpair[h][0]*sin(rspread))));

             //correct for elongated field size on the right MLC
             OUTmlc[wL].leafpair[h][1] = SAD*((OUTmlc[wL].leafpair[h][1]*cos(rspread))
                  /(SAD + (OUTmlc[wL].leafpair[h][1]*sin(rspread))));
           }
       }

     //correct for the spread beams on the right side of the plan beam angle

     for(n = 0; n < R; n++)
       {
         wR = w + (arccount - 1) - n;

         for (h = 0; h < 60; h++)
           {
             //correct for elongated field size on the left MLC

             OUTmlc[wR].leafpair[h][0] = SAD*((OUTmlc[wR].leafpair[h][0]*cos(rspread))
                  /(SAD + (OUTmlc[wR].leafpair[h][0]*sin(rspread))));

             //correct for elongated field size on the right MLC
             OUTmlc[wR].leafpair[h][1] = SAD*((OUTmlc[wR].leafpair[h][1]*cos(rspread))
                  /(SAD - (OUTmlc[wR].leafpair[h][1]*sin(rspread))));
           }
       }
   }

 //assigning MU to the individual spreading beams

 int nw;

 for(w = 0; w < pbcount; w++)
   {
     for(n = 0; n < arccount; n++)
       {
         nw = (w*arccount) + n;
         sbeam[nw].MU = ibeam[w].MU*OUTmlc[nw].MU;
       }
   }

//printing the individual mlc file to the corresonding spreading beams
int J;

for(x = 0; x < segcount; x++)
  {
    sprintf(out_mlc, "fcspread_MLC/fcspread_%d.mlc", x);
    fp[x] = fopen(out_mlc, "w");

    fprintf(fp[x], "%d\n", x);
    fprintf(fp[x], "1.00\n");              //all MLC has relative MU of 1
    fprintf(fp[x], "%0.1f %3.1f\n %0.1f %3.1f\n", OUTmlc[x].Y2, OUTmlc[x].Y1, OUTmlc[x
        ].X1, OUTmlc[x].X2);

    for(J = 1; J <= 60; J++)           //J is the number of leafpair, counting from 1
      {
        int JJ = J -1;
```

```
            fprintf(fp[x], "%d %3.3f %3.3f\n", J, OUTmlc[x].leafpair[JJ][0], OUTmlc[x].
                leafpair[JJ][1]);
          };
        fclose(fp[x]);
      }

    //printing .beams file
    sprintf(out_file, "fcspread_%s", argv[1]);
    edit_file = fopen(out_file, "w");          //open file again and write

    int i;
    int sbeamID = -1;                          //always count beams from zero
    float sbeamGANT;
    sbeamGANT = ibeam[0].GANT + (((arccount-1)/2)*dspread) + dspread;        //counting anti
        -clockwise

    if (arccount == 2 || arccount == 4 || arccount == 6 || arccount == 8 || arccount == 10)
            //for even number arcs
      {
        if (ibeam[0].GANT - ibeam[1].GANT < 0)          //for clockwise arcs
          {
            sbeamGANT = ibeam[0].GANT - ((((arccount/2)-1)*dspread) + (dspread/2)) +
                dspread;
          }
        else
          sbeamGANT = ibeam[0].GANT + ((((arccount/2)-1)*dspread) + (dspread/2)) + dspread;
                    //for anticlockwise arcs
      }

    else                    //for odd number arcs
      {
        if (ibeam[0].GANT - ibeam[1].GANT < 0)          //for clockwise arcs
          {
            sbeamGANT = ibeam[0].GANT - (fabs((arccount-1)/2)*dspread) + dspread;
                //for clockwise arcs
          }
        else
          sbeamGANT = ibeam[0].GANT + (fabs((arccount-1)/2)*dspread) + dspread;
              //for anticlockwise arcs
      }

    //loop for counting spreading beams (sbeamID)
    for(i = 0; i < segcount; i ++)
      {
        sbeamID++ ;

        if (ibeam[0].GANT - ibeam[1].GANT < 0)          //clockwise
          {
            sbeamGANT+= dspread;                 //counting clockwise
          }
        else
          sbeamGANT-= dspread;                  //counting anticlockwise

        if (sbeamGANT < 0)
          {
            sbeamGANT = 360 + sbeamGANT;
          }

        fprintf(edit_file,"%d\n %0.4f %3.2f %3.2f %5.2f %3.2f %3.2f\n fcspread_MLC/
            fcspread_%d.mlc\n\n", sbeamID, sbeam[i].MU, sbeamGANT, coll, ISOx, ISOy, ISOz,
            sbeamID);
      };

    fclose(edit_file);

    printf("Your stacking plan has now been successfully spread!\n");

    return 0;
}
```

Appendix E

C codes for converting MLC sequence into a Pinnacle^3 script

The codes below were compiled to convert the MLC sequence output from the k-link and AMRT algorithms to a Pinnacle^3 shell script.

```
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include <math.h>
#include <ctype.h>

#define INMLC_SIZE 40
#define ARC_SIZE 20

struct iBEAM                //input beams
{
  float MU;                 //monitor units of the individual beam
  float hMU;
  float GANT;               //gantry angle of beam
  char MLC_FILE[60];        //mlc file
};

struct inMLC
{
  int ID;                    //control point index
  float MU;                 //relative MU
  float Y2;                  //Y2 back-up jaw position
  float Y1;                  //Y1 back-up jaw position
  float X1;                  //X1 back-up jaw position
  float X2;                  //X2 back-up jaw position
  int lp[60];                /leafpair ID, count from 1 to 60
  float leafpair[60][2];         //leafpair positions
};

//setting parameters for program execution

int main(int argc, char *argv[])
{
  FILE *read_file;
  FILE *read_header;
  FILE *read_mlc;
  FILE *edit_script;
  char in_mlc[100];
  char out_file[100];
  char mlc_path[100];                    //specifying path of the folder for MLC files
  char header_path[200];

  struct iBEAM ibeam[INMLC_SIZE];
  struct inMLC INmlc[INMLC_SIZE][ARC_SIZE];

  int i, a, k, m, ibeamID, arcID;
  float coll;                //collimator angle
  float ISOx, ISOy, ISOz;      //3D coordinates of isocentre

  if(argc==1)
    {
      printf("Sorry, .beams file is not found. Exiting program.\n");

      exit(0);
    }

  //prompt user for header information

  printf("Please enter the name of the header file...\n");
  scanf("%s", &header_path);
```

```
//read header file

read_header = fopen(header_path, "r");

char dump;
float couch_angle, couch, dump2;
char machine[100];

float jaw_y2, jaw_y1, jaw_x1, jaw_x2;
float xdim, ydim;
float xstart, ystart;
float xpixdim, ypixdim;
float voxel_x, voxel_y, voxel_z;
float grid_dimx, grid_dimy, grid_dimz;
float origin_x, origin_y, origin_z;

fscanf(read_header, "%s %s", &dump, &machine);
fscanf(read_header, "%s %f", &dump, &couch);
fscanf(read_header, "%s %f", &dump, &dump2);
fscanf(read_header, "%s %f", &dump, &coll);
fscanf(read_header, "%s %f", &dump, &jaw_x1);
fscanf(read_header, "%s %f", &dump, &jaw_x2);
fscanf(read_header, "%s %f", &dump, &jaw_y2);
fscanf(read_header, "%s %f", &dump, &jaw_y1);

fscanf(read_header, "%s %f", &dump, &xdim);
fscanf(read_header, "%s %f", &dump, &ydim);
fscanf(read_header, "%s %f", &dump, &xstart);
fscanf(read_header, "%s %f", &dump, &ystart);
fscanf(read_header, "%s %f", &dump, &xpixdim);
fscanf(read_header, "%s %f", &dump, &ypixdim);
fscanf(read_header, "%s %f", &dump, &dump2);
fscanf(read_header, "%s %f", &dump, &voxel_x);
fscanf(read_header, "%s %f", &dump, &voxel_y);
fscanf(read_header, "%s %f", &dump, &voxel_z);

fscanf(read_header, "%s %f", &dump, &grid_dimx);
fscanf(read_header, "%s %f", &dump, &grid_dimy);
fscanf(read_header, "%s %f", &dump, &grid_dimz);
fscanf(read_header, "%s %f", &dump, &origin_x);
fscanf(read_header, "%s %f", &dump, &origin_y);
fscanf(read_header, "%s %f", &dump, &origin_z);

fclose(read_header);

//read in information from .beams file

read_file  = fopen(argv[1], "r");

int pbcount = 0;                        //pbcount is the total number of input beams
int aindex = 0;                         //counting number of arc for array definition
int arccount = 0;                       //counting number of arcs
int segcount = 0;                       //counting total number of apertures in the entire
    plan

while(fscanf(read_file, "%d", &ibeamID) !=EOF)
  {
    aindex = 0;

    fscanf(read_file, "%f %f %f %f %f %f %f %s", &ibeam[ibeamID].MU, &ibeam[ibeamID].
        GANT, &coll, &couch_angle, &ISOx, &ISOy, &ISOz, &in_mlc);

    read_mlc = fopen(in_mlc, "r");

    while(fscanf(read_mlc, "%d", &arcID) != EOF)
      {
        aindex ++;

        fscanf(read_mlc, "%f %f %f %f %f", &INmlc[ibeamID][aindex-1].MU, &INmlc[ibeamID
            ][aindex-1].Y2, &INmlc[ibeamID][aindex-1].Y1, &INmlc[ibeamID][aindex-1].X1,
             &INmlc[ibeamID][aindex-1].X2);

        for(m = 0; m < 60; m++)
          {
            fscanf(read_mlc, "%d %f %f", &INmlc[ibeamID][aindex-1].lp, &INmlc[ibeamID][
                aindex-1].leafpair[m][0], &INmlc[ibeamID][aindex-1].leafpair[m][1]);
          }
        segcount++;
      }
    fclose(read_mlc);
    pbcount++;
  }

fclose(read_file);

arccount = segcount/pbcount;

//writing script for Pinnacle
```

```
sprintf(out_file, "%s.Script", argv[1]);
edit_script = fopen(out_file, "w");          //open file again and write

fprintf(edit_script, "TrialList.Current.PrescriptionList.CreateChild = \"\";\n");
fprintf(edit_script, "TrialList.Current.PrescriptionList.Current.Method = \"Set Monitor
    Units\";\n");
fprintf(edit_script, "//\n");
fprintf(edit_script, "// Setup the dose grid.\n");
fprintf(edit_script, "//\n");
fprintf(edit_script, "TrialList.Current.DoseGrid.VoxelSize.X = %0.4f;\n", voxel_x);
fprintf(edit_script, "TrialList.Current.DoseGrid.VoxelSize.Y = %0.4f;\n", voxel_y);
fprintf(edit_script, "TrialList.Current.DoseGrid.VoxelSize.Z = %0.4f;\n", voxel_z);
fprintf(edit_script, "TrialList.Current.DoseGrid.Dimension.X = %0.4f;\n", grid_dimx);
fprintf(edit_script, "TrialList.Current.DoseGrid.Dimension.Y = %0.4f;\n", grid_dimy);
fprintf(edit_script, "TrialList.Current.DoseGrid.Dimension.Z = %0.4f;\n", grid_dimz);
fprintf(edit_script, "TrialList.Current.DoseGrid.Origin.X = %0.6f;\n", origin_x);
fprintf(edit_script, "TrialList.Current.DoseGrid.Origin.Y = %0.6f;\n", origin_y);
fprintf(edit_script, "TrialList.Current.DoseGrid.Origin.Z = %.6f;\n");
fprintf(edit_script, "//\n");
fprintf(edit_script, "// Create the beam and set weights, etc.\n");
fprintf(edit_script, "//\n");

for(i = 0; i < pbcount; i++)
  {
    fprintf(edit_script, "CreateNewBeam = \"\"\n");
    fprintf(edit_script, ";TrialList.Current.BeamList.Current.Name = \"Beam %d\";\n", i
        +1);
    fprintf(edit_script, "TrialList.Current.BeamList.Current.Name = \"Beam %d\";\n", i
        +1);
    fprintf(edit_script, "TrialList.Current.BeamList.Current.Machine = \"WHITE\";\n");
    fprintf(edit_script, "TrialList.Current.BeamList.Current.SetBeamType = \"Step &
        Shoot MLC\";\n");

    fprintf(edit_script, "TrialList.Current.SetBeamMonitorUnits.Current.Address =
        \"%6.1f\";\n", ibeam[i].MU);

    fprintf(edit_script, "//\n");
    fprintf(edit_script, "// Set beam geometry and jaws to be independent\n");
    fprintf(edit_script, "//\n");

    fprintf(edit_script, "TrialList.Current.BeamList.Current.CPManager.
        IsLeftRightIndependent = 1;\n");
    fprintf(edit_script, "TrialList.Current.BeamList.Current.CPManager.
        IsTopBottomIndependent = 1;\n");
    fprintf(edit_script, "TrialList.Current.BeamList.Current.CPManager.Collimator = %d
        ;\n", (int)coll);
    fprintf(edit_script, "TrialList.Current.BeamList.Current.CPManager.Gantry = %d;\n",
         (int)ibeam[i].GANT);
    fprintf(edit_script, "TrialList.Current.BeamList.Current.CPManager.Couch = 0;\n");
    fprintf(edit_script, "TrialList.Current.BeamList.Current.ClipMLCDisplay = 1;\n");
    fprintf(edit_script, "TrialList.Current.BeamList.Current.SolidMLCDisplay = 1;\n");

    fprintf(edit_script, "//\n");
    fprintf(edit_script, "// This will move the isocenter\n");
    fprintf(edit_script, "//\n");

    fprintf(edit_script, "TrialList.Current.BeamList.Current.Isocenter = \"ISOCENTER
        \";\n");

    fprintf(edit_script, "//\n");
    fprintf(edit_script, "// Add n-1 children to the control point list\n");
    fprintf(edit_script, "//\n");

    fprintf(edit_script, "TrialList.Current.BeamList.Current.CPManager.ControlPointList
        .AddChildren = %d;\n", arccount-1);

    for(a = 0; a < arccount; a++)
      {
        fprintf(edit_script, "//\n");
        fprintf(edit_script, "// Now build the control point list\n");
        fprintf(edit_script, "//\n");
        fprintf(edit_script, "// BEAM #%d CONTROL POINT #%d\n", i, a);

        fprintf(edit_script, "TrialList.Current.BeamList.Current.CPManager.
            ControlPointList.#\"#%d\".WeightAsPercent = %3.2f;\n", a, (INmlc[i][a].MU/
            ibeam[i].MU)*100 );
        fprintf(edit_script, "TrialList.Current.BeamList.Current.CPManager.
            ControlPointList.#\"#%d\".WeightLocked = 1;\n", a);
        fprintf(edit_script, "TrialList.Current.BeamList.Current.CPManager.
            ControlPointList.#\"#%d\".LeftJawPosition = %3.1f;\n", a, INmlc[i][a].X1
            *-1);
        fprintf(edit_script, "TrialList.Current.BeamList.Current.CPManager.
            ControlPointList.#\"#%d\".RightJawPosition = %3.1f;\n", a, INmlc[i][a].X2);
        fprintf(edit_script, "TrialList.Current.BeamList.Current.CPManager.
            ControlPointList.#\"#%d\".TopJawPosition = %3.1f;\n", a, INmlc[i][a].Y2*-1)
            ;
```

```
            fprintf(edit_script, "TrialList.Current.BeamList.Current.CPManager.
                 ControlPointList.#\"#%d\".BottomJawPosition = %3.1f;\n", a, INmlc[i][a].Y1)
                 ;
            fprintf(edit_script, "TrialList.Current.BeamList.Current.CPManager.
                 ControlPointList.#\"#%d\".Gantry = %d;\n", a, (int)ibeam[i].GANT);
            fprintf(edit_script, "TrialList.Current.BeamList.Current.CPManager.
                 ControlPointList.#\"#%d\".Collimator = %d;\n", a, (int)coll);
            fprintf(edit_script, "TrialList.Current.BeamList.Current.CPManager.
                 ControlPointList.#\"#%d\".Couch = 0;\n", a);
            fprintf(edit_script, "TrialList.Current.BeamList.Current.CPManager.
                 ControlPointList.#\"#%d\".\n", a);

            fprintf(edit_script, "     MLCLeafPositions = {\n"
                        "         RawData = {\n"
                        "             NumberOfDimensions = 2;\n"
                        "             NumberOfPoints = 60;\n");
            fprintf(edit_script, "             Points[] = {\n");

            for (k = 0; k < 60; k++)
              {
                 fprintf(edit_script,"                 %7.2f, %7.2f,\n", (-1*INmlc[i][a].
                      leafpair[k][0]), INmlc[i][a].leafpair[k][1]);
              }
            fprintf(edit_script, "     };\n");
            fprintf(edit_script, "    };\n");
            fprintf(edit_script, "   };\n");
          }
      }

    fclose(edit_script);

    return 0;
}
```

Appendix F

C codes for converting AMRT sequence to multi-arc MLC sequence

The codes below were compiled to convert the AMRT MLC sequence to a multi-arc MLC sequence to enable import to Pinnacle[3] for segment weight optimization.

```
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include <math.h>
#include <ctype.h>

#define INMLC_SIZE 1000
#define OUTMLC_SIZE 40
#define ARC_SIZE 20
#define total_xbeam 36

struct iBEAM                //input beams
{
  float MU;                 //monitor units of the individual beam
  float hMU;
  float GANT;               //gantry angle of beam
  char MLC_FILE[60];        //mlc file
};

struct xBEAM                //expanded beams
{
  int ID;                   //beam ID
  float MU;                 //linearly interpolated MU
  float hMU;                //half the MU
  float GANT;
};

struct inMLC
{
  int ID;                   //control point index
  float MU;                 //relative MU
  float Y2;                 //Y2 back-up jaw position
  float Y1;                 //Y1 back-up jaw position
  float X1;                 //X1 back-up jaw position
  float X2;                 //X2 back-up jaw position
  int lp[60];                  //leafpair ID, count from 1 to 60
  float leafpair[60][2];          //leafpair positions
};

struct outMLC
{
  float MU;                 //relative MU
  float Y2;                 //Y2 back-up jaw position
  float Y1;                 //Y1 back-up jaw position
  float X1;                 //X1 back-up jaw position
  float X2;                 //X2 back-up jaw position
  int lp[60];                  //leafpair ID, count from 1 to 60
  float leafpair[60][2];          //leafpair positions
};

//setting parameters for program execution

int main(int argc, char *argv[])
{
  FILE *read_file;
```

```
FILE *read_mlc;
FILE *edit_file;
FILE *edit_mlc;
FILE *fp[INMLC_SIZE];
char in_mlc[100];
char out_file[100];
char out_mlc[100];
char mlc_path[100];

struct iBEAM * ibeam ;
struct xBEAM * xbeam;
struct inMLC * INmlc;
struct outMLC ** OUTmlc;

int a, m, ibeamID, arcID;
float dsep;
int finebeam;
float dfine;
float coll;
float ISOx, ISOy, ISOz;
float xbeamGANT;
float xbeamGANTrad;

int temp_sl ;

ibeam = (struct iBEAM *) malloc ( INMLC_SIZE * sizeof ( struct iBEAM) );
xbeam = (struct xBEAM *) malloc ( OUTMLC_SIZE * sizeof (struct xBEAM) );
INmlc = (struct inMLC *) malloc ( INMLC_SIZE * sizeof (struct inMLC ) );

OUTmlc = (struct outMLC **) malloc ( OUTMLC_SIZE * sizeof (struct outMLC *) );
temp_sl = 0 ;
while ( temp_sl < OUTMLC_SIZE ){
  OUTmlc[temp_sl] = (struct outMLC *) malloc ( ARC_SIZE * sizeof ( struct outMLC) ) ;
  temp_sl ++ ;
}

if(argc==1)
  {
    printf("Sorry, .beams file is not found. Exiting program.\n");

    exit(0);
  }

//prompt user for info

printf("What is the directory for the output MLC files?\n");
scanf("%s", mlc_path);

read_file  = fopen(argv[1], "r");

int total_ibeam = 0;                    //total number of input beams
int aindex = 0;                        //counting number of arc for array definition
int arccount = 0;                      //counting number of arcs

while(fscanf(read_file, "%d", &ibeamID) !=EOF)
  {
    aindex = 0;

    fscanf(read_file, "%f %f %f %f %f %f %s", &ibeam[ibeamID].MU, &ibeam[ibeamID].GANT,
        &coll, &ISOx, &ISOy, &ISOz, &in_mlc);

    read_mlc = fopen(in_mlc, "r");

    while(fscanf(read_mlc, "%d", &arcID) != EOF)
      {
        fscanf(read_mlc, "%f %f %f %f %f", &INmlc[ibeamID].MU, &INmlc[ibeamID].Y2, &
            INmlc[ibeamID].Y1, &INmlc[ibeamID].X1, &INmlc[ibeamID].X2);

        for(m = 0; m < 60; m++)
          {
            fscanf(read_mlc, "%d %f %f", &INmlc[ibeamID].lp, &INmlc[ibeamID].leafpair[m
                ][0], &INmlc[ibeamID].leafpair[m][1]);
          }
      }
    fclose(read_mlc);
    total_ibeam++;
  }

fclose(read_file);

arccount = total_ibeam/total_xbeam;

//stacking the single-arc beams
int i, w;
int x = -1;

for(i = 0; i < total_xbeam; i ++)
  {
```

```
      for(a = 0; a < arccount; a++)
        {
          x++;

          xbeam[i].MU += ibeam[x].MU;

          OUTmlc[i][a].MU = ibeam[x].MU;
          OUTmlc[i][a].X1 = INmlc[x].X1;
          OUTmlc[i][a].X2 = INmlc[x].X2;
          OUTmlc[i][a].Y1 = INmlc[x].Y1;
          OUTmlc[i][a].Y2 = INmlc[x].Y2;

          for(w = 0; w < 60; w++)
            {
              OUTmlc[i][a].leafpair[w][0] = INmlc[x].leafpair[w][0];
              OUTmlc[i][a].leafpair[w][1] = INmlc[x].leafpair[w][1];
            };
        };
    };

  //writing the output of MLC files
  int J;

  for(x = 0; x < total_xbeam; x++)
    {
      sprintf(out_mlc, "%s/beam_%d.mlc", mlc_path, x);
      fp[x] = fopen(out_mlc, "w");

      for(a = 0; a < arccount; a++)
        {
          fprintf(fp[x], "%d\n", a);
          fprintf(fp[x], "%0.5f\n", OUTmlc[x][a].MU);
          fprintf(fp[x], "%0.1f %3.1f\n %0.1f %3.1f\n", OUTmlc[x][a].Y2, OUTmlc[x][a].Y1,
                OUTmlc[x][a].X1, OUTmlc[x][a].X2);

          for(J = 1; J <= 60; J++)          //J is the number of leafpair, counting from 1
            {
              int JJ = J -1;
              fprintf(fp[x], "%d %3.3f %3.3f\n", J, OUTmlc[x][a].leafpair[JJ][0], OUTmlc[
                    x][a].leafpair[JJ][1]);
            };
        };
      fclose(fp[x]);
    };

  //writing output file

  sprintf(out_file, "amrt2pinnacle_%s", argv[1]);
  edit_file = fopen(out_file, "w");          //open file again and write

  for(i = 0; i < total_xbeam; i++)
    {
      xbeam[i].GANT = 175 - (i*10);          //5 degrees is the middle of the 10-degree
          sector

      if (xbeam[i].GANT < 0)
        {
          xbeam[i].GANT = xbeam[i].GANT + 360;
        }
      fprintf(edit_file,"%d\n %0.4f %3.2f %3.2f %5.2f %3.2f %3.2f\n %s/beam_%d.mlc\n", i,
            xbeam[i].MU, xbeam[i].GANT, coll, ISOx, ISOy, ISOz, mlc_path, i);
    }

  fclose(edit_file);

  free (ibeam) ;
  free (xbeam) ;

  temp_sl = 0 ;
  while ( temp_sl < OUTMLC_SIZE ){
    free ( OUTmlc[temp_sl] );
    temp_sl ++ ;
  }
  free ( OUTmlc ) ;

  return 0;
}
```

Appendix G

C codes for converting VDR plans into CDR plans

The codes below were compiled to convert the VDR plans into CDR plans by partitioning the single-arc into multiple sectors.

```
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include <math.h>
#include <ctype.h>

#define ARC_SIZE 30
#define CP_SIZE 500

struct IBEAM
{
  int ID;                         //control point index
  float GANT;                     //gantry
  float out_GANT;          //unevenspread gantry angle
  float final_GANT;        //final output gantry angle for unbiased delivery
  float final_MU;                 //final output MU for unbiased delivery
  float cMU;
  float MU;                       //absolute MU
  float tmp;
  char mlc[200];
  char out_mlc[200];   //final output in .beams file
};

struct ARC
{
  float MU;
  int CP;                         //number of control points
  float range;                    //arc range
  float space;                 //spacing factor
  float dtheta_end;        //angular interval between the last two beams of a broken arc
  float dtheta_beg;        //angular interval between the first two beams of a broken arc
  int tmp;
};

struct MLC
{
  int ID;                         //control point index
  float MU;                       //relative MU
  float Y2;                       //Y2 back-up jaw position
  float Y1;                       //Y1 back-up jaw position
  float X1;                       //X1 back-up jaw position
  float X2;                       //X2 back-up jaw position
  int lp[60];                     //leafpair ID, count from 1 to 60
  float leafpair[60][2];          //leafpair positions
};

struct final_MLC
{
  float MU;                       //relative MU
  float Y2;                       //Y2 back-up jaw position
  float Y1;                       //Y1 back-up jaw position
  float X1;                       //X1 back-up jaw position
  float X2;                       //X2 back-up jaw position
  int lp[60];                     //leafpair ID, count from 1 to 60
  float leafpair[60][2];          //leafpair positions
};

//setting parameters for program execution

int main(int argc, char *argv[])
{
```

```
FILE *read_file;
FILE *read_mlc;
FILE *out_beam;
FILE *log_file;
FILE *fp[ARC_SIZE];
char beams[200];
char log[200];
char out_mlc[200];

int i, a, p, beamID, mlcID, garb;
float coll, couch;
float ISOx, ISOy, ISOz;      //3D coordinates of isocentre
int arcID, trash, num_CP, num_arcs;
float totalMU;
int ii;
int pbcount = 0;
beamID = -1;
int n_arcs[10];

struct IBEAM ibeam[CP_SIZE];
struct ARC arc[ARC_SIZE];
struct MLC mlc[CP_SIZE];
struct final_MLC final_mlc[CP_SIZE];

read_file  = fopen(argv[1], "r");

while(fscanf(read_file, "%d", &beamID) !=EOF)
  {

    fscanf(read_file, "%f %f %f %f %f %f %s", &ibeam[beamID].MU, &ibeam[beamID].GANT, &
         coll, &ISOx, &ISOy, &ISOz, &ibeam[beamID].mlc);

    read_mlc = fopen(ibeam[beamID].mlc, "r");

    fscanf(read_mlc, "%d %f %f %f %f %f", &mlc[beamID].ID, &mlc[beamID].MU, &mlc[beamID
         ].Y2, &mlc[beamID].Y1, &mlc[beamID].X1, &mlc[beamID].X2);

    for(p = 0; p < 60; p++)
      {
        fscanf(read_mlc, "%d %f %f", &mlc[beamID].lp, &mlc[beamID].leafpair[p][0], &mlc
             [beamID].leafpair[p][1]);
      }
    fclose(read_mlc);

    pbcount++;
  }
fclose(read_file);

float dsep, totalrange;

totalrange = ibeam[0].GANT + (360 - ibeam[pbcount-1].GANT);

float sub;
sub = pbcount - 2;

dsep = totalrange/sub;              //range = (pbcount-2-1)*dsep + 2*(dsep/2) for
     RapidArc plans, with the first and last beam occupying half of dsep.

int stop = 1;

//Take out the first beam interval

arc[0].range = arc[0].range - (ibeam[0].GANT - ibeam[1].GANT);

//Optimizing the break up points

int kk = 0;
int z = -1;
int k, x;
int residual;
float tmp;

while(kk == 0)
  {
    x = 0;
    k = -1;
    residual = pbcount;

    z++;

    while (residual > 0)
      {
        x = 0;
        stop = 1;
        k++;

        while (stop > 0)
          {
```

```
start:

  arc[k].range = 0;
  arc[k].MU = 0;

  if(k == 0 && x == 0)
    {
      arc[k].range = ((pbcount - 2 - 1)*dsep) + (dsep/2);
    }

  if(k == 0 && x == 1)
    {
      arc[k].range = ((pbcount - 2 - 1)*dsep) + (dsep/2);
    }
  else
    if(k == 0)
      {
        arc[k].range = ((pbcount - x - 1)*dsep) ;
      }

  if(k > 0 && x == 0)
    {
      arc[k].range = ((residual - 1 - 1)*dsep) + (dsep/2);
    }

  if(k > 0 && x == 1)
    {
      arc[k].range = ((residual - 1 - 1)*dsep) + (dsep/2);
    }
  else
    if(k > 0)
      {
        arc[k].range = (residual - x)*dsep;
      }

  //Work out the MU of each broken arc

  beamID = pbcount - residual - 1;

  for(i = 0; i < (residual - x); i++)      // counting from the last beam
    {
      beamID++;

      arc[k].MU += ibeam[beamID].MU;

    }

  //Work out the spacing factor of each broken arc

  arc[k].space = arc[k].MU/arc[k].range;

  //checking the deviation of each angle

  beamID = pbcount - residual - 1;

  for( i = 0; i < (residual - x); i++)
    {
      beamID ++;

      if(beamID == 1 || beamID == 0 || i == 0)
        {
          ibeam[beamID].out_GANT = ibeam[beamID].GANT;
        }
      else
        {
          ibeam[beamID].out_GANT = ibeam[beamID-1].out_GANT - (ibeam[beamID
                -1].MU/arc[k].space);
         }

      if(ibeam[beamID].out_GANT < 0)
        {
          ibeam[beamID].out_GANT = 360 + ibeam[beamID].out_GANT;
        }

      ibeam[beamID].tmp = fabs(ibeam[beamID].out_GANT - ibeam[beamID].GANT);

      if(ibeam[beamID].tmp > 300)
        {
          ibeam[beamID].tmp = 360 - ibeam[beamID].tmp;
        }
    }

  beamID = pbcount - residual - 1;

  for(i = 0; i < (residual - x); i++)    //checking for the angle deviation
       of each beam
    {
      beamID++;
```

```
                if(ibeam[beamID].tmp > ( 5- z))
                  {
                    x++;
                    goto start;
                  }
                else
                  arc[k].CP = residual - x;
              }
            stop = 0;
          }
        residual = residual - arc[k].CP;
      }

    n_arcs[z] = k + 1;

    if(z == 0)
      {
        kk = 0;
      }
    else
      {
        if(n_arcs[z] > n_arcs[z-1])  //if the number of arcs is larger than stop the
            loop for reducing the angular deviation restriction
          {
            num_arcs = n_arcs[z-1];

            printf("The single-arc plan is broken into %d sectors with angular
                deviation < %d degrees.\n", n_arcs[z-1], 5-(z-1));

            kk = 1;
          }
      }
  }

int j, jj;
j = 0;

//get the spacing factor of each broken arc

for(ii = 0; ii < num_arcs; ii++)
  {
    j += arc[ii].CP;
    jj = j - arc[ii].CP;

    if(ii == 0) //for the first arc
      {
        if(ibeam[1].GANT - ibeam[j].GANT < 0)    //check if any beams go pass 0 degrees
          {
            arc[ii].range = (360 - ibeam[j].GANT) + ibeam[1].GANT;
          }
        else

          arc[ii].range = ibeam[1].GANT - ibeam[j].GANT;
      }
    else
      if(ii == (num_arcs -1))    //for the last arc
        {
          if(ibeam[jj].GANT - ibeam[j-1].GANT < 0)  //check if any beams go pass 0
              degrees
            {
              arc[ii].range = (360 - ibeam[j-1].GANT) + ibeam[jj].GANT + (dsep/2);
            }

          arc[ii].range = ibeam[jj].GANT - ibeam[j-1].GANT + (dsep/2);
        }
      else
        {
          if(ibeam[jj].GANT - ibeam[j].GANT < 0)     //check if any beams go pass 0
              degrees
            {
              arc[ii].range = (360 - ibeam[j].GANT) + ibeam[jj].GANT;
            }
          else
            arc[ii].range = ibeam[jj].GANT - ibeam[j].GANT;
        }
  }

//get the total MU for each broken arc

j = -1;

for(ii = 0; ii < num_arcs; ii++)
  {
    arc[ii].MU = 0;

    for(i = 0; i < arc[ii].CP; i++)
      {
```

```
        j++;

        arc[ii].MU += ibeam[j].MU;
      }
  }

//get the spacing factor for each broken arc

for(i = 0; i < num_arcs; i++)
  {
    arc[i].space = arc[i].MU/arc[i].range;
  }

//determinie the new angles of the beams for the broken arcs

j = -1;

for(ii = 0; ii < num_arcs; ii++)
  {
    for(i = 0; i < arc[ii].CP; i++)
      {
        j++;

        if(i == 0 && j != 0)
          {
            ibeam[j].out_GANT = ibeam[j-1].out_GANT - (ibeam[j-1].MU/arc[ii-1].space);
          }
        else
          if(j == 0 || j ==1)
            {
              ibeam[j].out_GANT = ibeam[j].GANT;
            }
          else

            ibeam[j].out_GANT = ibeam[j-1].out_GANT - (ibeam[j-1].MU/arc[ii].space);

        if(ibeam[j].out_GANT < 0)
          {
            ibeam[j].out_GANT = 360 + ibeam[j].out_GANT;
          }

      }
  }

//Re-adjusting the starting and stopping angle of the beams at the broken arc junction.
    Edited on 12 Jan 2009.

j = -1;

for(ii = 0; ii < num_arcs; ii++)
  {
    for(i = 0; i < arc[ii].CP; i++)
      {
        j++;

        if(i == 0 && ii != 0)
          {
            arc[ii].dtheta_beg = (ibeam[j-1].out_GANT - ibeam[j].out_GANT);

            if(arc[ii].dtheta_beg < 0)
              {
                arc[ii].dtheta_beg = 360 + arc[ii].dtheta_beg;
              }
          }
        else
          if(i == (arc[ii].CP - 1) && ii != (num_arcs -1))
            {
              arc[ii].dtheta_end = (ibeam[j-1].out_GANT - ibeam[j].out_GANT);

              if(arc[ii].dtheta_end < 0)
                {
                  arc[ii].dtheta_end = 360 + arc[ii].dtheta_end;
                }
            }
      }
  }

k = -1;
j = -1;
int m;
int total_outbeams;          //total number of beams in the delivery sequence, including
    the kick-off beams

for(ii = 0; ii < num_arcs; ii ++)
  {
    for(i = 0; i < arc[ii].CP; i++)
      {
        j++;
```

```
k++;

//shift the last beam backwards to 1/4 of the beam interval between the last
    two beams in a broken arc
if(i == (arc[ii].CP - 1)  && ii != (num_arcs - 1))
  {
    ibeam[k].final_GANT = ibeam[j].out_GANT + (arc[ii].dtheta_end/4);

    if(ibeam[k].final_GANT > 360)
      {
        ibeam[k].final_GANT = ibeam[k].final_GANT - 360;
      }

    ibeam[k].final_MU = ibeam[j].MU/2;

    strcpy(ibeam[k].out_mlc, ibeam[j].mlc);

    for(m = 0; m < 60; m++)
      {
        final_mlc[k].leafpair[m][0] = mlc[j].leafpair[m][0];
        final_mlc[k].leafpair[m][1] = mlc[j].leafpair[m][1];
      }

  //shift the first beam to the same as the last beam angle of the previous
      broken arc
  //if(i == 0 && ii!= 0)

    //j = j-1;
    k = k+1;

    ibeam[k].final_GANT = ibeam[k-1].final_GANT;
    ibeam[k].final_MU = 0;
    strcpy(ibeam[k].out_mlc, ibeam[j].mlc);

    for(m = 0; m < 60; m++)
      {
        final_mlc[k].leafpair[m][0] = mlc[j].leafpair[m][0];
        final_mlc[k].leafpair[m][1] = mlc[j].leafpair[m][1];
      }

    //add a second beam to the broken arc, which is half the MU of the original
        beam

    k = k+1;

    ibeam[k].final_GANT = ibeam[j].out_GANT - (arc[ii+1].dtheta_beg/4);

    if(ibeam[k].final_GANT < 0)
      {
        ibeam[k].final_GANT = 360 + ibeam[k].final_GANT;
      }

    ibeam[k].final_MU = ibeam[j].MU/2;
    strcpy(ibeam[k].out_mlc, ibeam[j].mlc);

    for(m = 0; m < 60; m++)
      {
        final_mlc[k].leafpair[m][0] = mlc[j].leafpair[m][0];
        final_mlc[k].leafpair[m][1] = mlc[j].leafpair[m][1];
      }
  }

else
  if(ii == (num_arcs -1) && i == (arc[ii].CP - 1))
    {
      ibeam[k].final_GANT = ibeam[j].out_GANT;
      ibeam[k].final_MU = ibeam[j].MU;

      strcpy(ibeam[k].out_mlc, ibeam[j].mlc);

      for(m = 0; m < 60; m++)
        {
          final_mlc[k].leafpair[m][0] = mlc[j].leafpair[m][0];
          final_mlc[k].leafpair[m][1] = mlc[j].leafpair[m][1];
        }
      total_outbeams = k+1;
    }

  else
    {
      ibeam[k].final_GANT = ibeam[j].out_GANT;
      ibeam[k].final_MU = ibeam[j].MU;
      strcpy(ibeam[k].out_mlc, ibeam[j].mlc);

      for(m = 0; m < 60; m++)
        {
          final_mlc[k].leafpair[m][0] = mlc[j].leafpair[m][0];
          final_mlc[k].leafpair[m][1] = mlc[j].leafpair[m][1];
```

```
                }
            }

        }
    }

//Printing out the .beams file

k = -1;
j = -1;

sprintf(beams, "brokenarc_refine_%s", argv[1]);
out_beam = fopen(beams, "w");

for(i = 0; i < total_outbeams; i++)
    {
      fprintf(out_beam, "%d\n", i);
      fprintf(out_beam, "%0.3f %5.3f %5.2f\t", ibeam[i].final_MU, ibeam[i].final_GANT,
           coll);
      fprintf(out_beam, "%0.3f %5.2f %5.2f\n", ISOx, ISOy, ISOz);
      fprintf(out_beam, "%s\n\n", ibeam[i].out_mlc);
    }
fclose(out_beam);

//Printing out the log file

x = 0;
m = 0;
j = 0;
jj = 0;

sprintf(log, "%s_brokenarc_refine.log", argv[1]);
log_file = fopen(log, "w");

fprintf(log_file, "Number of broken arcs: %d\n", num_arcs);

for(k = 0; k < num_arcs; k++)
    {

      j += arc[k].CP + 2;
      jj = j - arc[k].CP;

      if(k == 0)
        {
          fprintf(log_file, "Arc %d contains %d control points. The total MU for this arc
               is %f\n Arc %d starts at %f and ends at %f\n\n", k, arc[k].CP, arc[k].MU,
               k, ibeam[0].final_GANT, ibeam[j-1-2].final_GANT);
        }
      else
        if(k == 1)
          {
            fprintf(log_file, "Arc %d contains %d control points. The total MU for this
                 arc is %f\n Arc %d starts at %f and ends at %f\n\n", k, arc[k].CP, arc[k
                 ].MU, k, ibeam[jj-1-2].final_GANT, ibeam[j-1-2].final_GANT);
          }

        else
          fprintf(log_file, "Arc %d contains %d control points. The total MU for this arc
               is %f\n Arc %d starts at %f and ends at %f\n\n", k, arc[k].CP+2, arc[k].MU
               , k, ibeam[jj].final_GANT, ibeam[j].final_GANT);
    }

fclose(log_file);

//Printing out the MLC file for Eclipse import
//Printing the first broken sector MLC file independently

x = -1;

sprintf(out_mlc, "Refine_Varian_arc0.mlc");
fp[0] = fopen(out_mlc, "w");

for(i = 0; i < arc[0].CP; i++)
    {
      x++;

      if(i == 0)
        {
          fprintf(fp[0], "File Rev = H\n");
          fprintf(fp[0], "Treatment = Dynamic Arc\n");
          fprintf(fp[0], "Last Name = CDR_broken\n");
          fprintf(fp[0], "First Name = Unbiased\n");
          fprintf(fp[0], "Patient ID = %s\n", argv[1]);
          fprintf(fp[0], "Number of Fields = %d\n", arc[0].CP);
          fprintf(fp[0], "Model = Varian 120M\n");
          fprintf(fp[0], "Tolerance = 0.50\n\n");
        }
```

```
    fprintf(fp[0], "Field = Field 0.%d\n", i);            //e.g. Field 1.0 means arc 1,
        control point 0.
    fprintf(fp[0], "Index = %0.4f\n", ibeam[x].final_GANT);        //Index = gantry
        angle
    fprintf(fp[0], "Carriage Group = 1\n");
    fprintf(fp[0], "Operator = \n");
    fprintf(fp[0], "Collimator = %0.1f\n", coll);

    for(j = 1; j <= 60; j ++)
      {
        int jj;

        jj = 60 - j;

        if(j < 10)
          {
            fprintf(fp[0], "Leaf  %dA = %0.3f\n", j, final_mlc[x].leafpair[jj][1]);
          }
        else
          fprintf(fp[0], "Leaf %dA = %0.3f\n", j, final_mlc[x].leafpair[jj][1]);
      }

    for(j = 1; j <= 60; j ++)
      {
        int jj;

        jj = 60 - j;

        if(j < 10)
          {
            fprintf(fp[0], "Leaf  %dB = %0.3f\n", j, final_mlc[x].leafpair[jj][0]*-1)
                ;
          }
        else
          fprintf(fp[0], "Leaf %dB = %0.3f\n", j, final_mlc[x].leafpair[jj][0]*-1);
      }

    fprintf(fp[0], "Note = 0\n");
    fprintf(fp[0], "Shape = 0\n");
    fprintf(fp[0], "Magnification = 1.00\n\n");

    if( i == (arc[0].CP - 1))
      {
        fprintf(fp[0], "CRC = D375\n");
      }
  }
fclose(fp[0]);


//Now, print out the rest of the MLC files

for(k = 1; k < num_arcs; k++)
  {
    sprintf(out_mlc, "Refine_Varian_arc%d.mlc", k);
    fp[k] = fopen(out_mlc, "w");

    for(i = 0; i < (arc[k].CP+2); i++)
      {
        x++;
        //  printf("x %d    arc %d   CP %d\n", x, k, i);
        if(i == 0)
          {
            fprintf(fp[k], "File Rev = H\n");
            fprintf(fp[k], "Treatment = Dynamic Arc\n");
            fprintf(fp[k], "Last Name = BrokenArc\n");
            fprintf(fp[k], "First Name = CDR\n");
            fprintf(fp[k], "Patient ID = %s\n", argv[1]);
            fprintf(fp[k], "Number of Fields = %d\n", arc[k].CP+2);
            fprintf(fp[k], "Model = Varian 120M\n");
            fprintf(fp[k], "Tolerance = 0.50\n\n");
          }

        fprintf(fp[k], "Field = Field %d.%d\n", k, i);            //e.g. Field 1.0 means
            arc 1, control point 0.
        fprintf(fp[k], "Index = %0.4f\n", ibeam[x].final_GANT);        //Index =
            gantry angle
        fprintf(fp[k], "Carriage Group = 1\n");
        fprintf(fp[k], "Operator = \n");
        fprintf(fp[k], "Collimator = %0.1f\n", coll);

        for(j = 1; j <= 60; j ++)
          {
            int jj;

            jj = 60 - j;

            if(j < 10)
              {
```

```
                fprintf(fp[k], "Leaf␣␣%dA␣=␣␣␣%0.3f\n", j, final_mlc[x].leafpair[jj
                    ][1]);
              }
            else
              fprintf(fp[k], "Leaf␣%dA␣=␣␣␣%0.3f\n", j, final_mlc[x].leafpair[jj][1]);
          }

        for(j = 1; j <= 60; j ++)
          {
            int jj;

            jj = 60 - j;

            if(j < 10)
              {
                fprintf(fp[k], "Leaf␣␣%dB␣=␣␣␣%0.3f\n", j, final_mlc[x].leafpair[jj
                    ][0]*-1);
              }
            else
              fprintf(fp[k], "Leaf␣%dB␣=␣␣␣%0.3f\n", j, final_mlc[x].leafpair[jj
                  ][0]*-1);
          }

        fprintf(fp[k], "Note␣=␣0\n");
        fprintf(fp[k], "Shape␣=␣0\n");
        fprintf(fp[k], "Magnification␣=␣1.00\n\n");

        if( i == (arc[k].CP+2 - 1))
          {
            fprintf(fp[k], "CRC␣=␣D375\n");
          }
      }
    fclose(fp[k]);
  }
 return 0;
}
```

Bibliography

[Ahnesjö *et al* 1987] A Ahnesjö, P Andreo and A Brahme. *Calculation and application of point spread functions for treatment planning with high energy photon beams.* Acta Oncol, vol. 26, no. 1, pages 49–56, 1987.

[Ahnesjö *et al* 1992] A Ahnesjö, M Saxner and A Trepp. *A pencil beam model for photon dose calculation.* Med Phys, vol. 19, no. 2, pages 263–73, 1992.

[Ahnesjö 1989] A Ahnesjö. *Collapsed cone convolution of radiant energy for photon dose calculation in heterogeneous media.* Med Phys, vol. 16, no. 4, pages 577–92, 1989.

[Baró *et al* 2003] J Baró, J Sempau, F Salvat and J M Fernández-Varea. *PENELOPE: An algorithm for Monte Carlo simulation of the penetration and energy loss of electrons and positrons in matter.* Nuclear Instruments and Methods in Physics Research Section B, vol. 100, pages 31–46, 2003.

[Bauman *et al* 2004] G Bauman, E Gete, J Z Chen and E Wong. *Simplified intensity-modulated arc therapy for dose escalated prostate cancer radiotherapy.* Med Dosim, vol. 29, no. 1, pages 18–25, 2004.

[Bielajew *et al* 1994] A F Bielajew, H Hirayama, W R Nelson and D W O Rogers. *History, overview and recent improvements of EGS4.* Report NRC-PIRS-0436, 1994.

[Bortfeld & Webb 2009] Thomas Bortfeld and Steve Webb. *Single-Arc IMRT?* Phys Med Biol, vol. 54, no. 1, pages N9–20, 2009.

[Bortfeld *et al* 1990] T Bortfeld, J Bürkelbach, R Boesecke and W Schlegel. *Methods of image reconstruction from projections applied to conformation radiotherapy.* Phys Med Biol, vol. 35, no. 10, pages 1423–34, 1990.

[Bortfeld *et al* 1994] T R Bortfeld, D L Kahler, T J Waldron and A L Boyer. *X-ray field compensation with multileaf collimators.* Int J Radiat Oncol Biol Phys, vol. 28, no. 3, pages 723–30, 1994.

[Boyer & Mok 1985] A Boyer and E Mok. *A photon dose distribution model employing convolution calculations.* Med Phys, vol. 12, no. 2, pages 169–77, 1985.

[Brahme *et al* 1982] A Brahme, J E Roos and I Lax. *Solution of an integral equation encountered in rotation therapy.* Phys Med Biol, vol. 27, no. 10, pages 1221–9, 1982.

[Brahme 1988] A Brahme. *Optimization of stationary and moving beam radiation therapy techniques.* Radiother Oncol, vol. 12, no. 2, pages 129–40, 1988.

[Briesmeister 2000] J F Briesmeister. *MCNP-A general Monte Carlo n-particle transport code version 4c.* Report LA- 13709-M, 2000.

[Cameron 2005] C Cameron. *Sweeping-window arc therapy: an implementation of rotational IMRT with automatic beam-weight calculation.* Phys Med Biol, vol. 50, no. 18, pages 4317–36, 2005.

[Cao *et al* 2006] Daliang Cao, Matthew A Earl, Shuang Luan and David M Shepard. *Continuous intensity map optimization (CIMO): a novel approach to leaf sequencing in step and shoot IMRT.* Med Phys, vol. 33, no. 4, pages 859–67, 2006.

[Cao *et al* 2007] Daliang Cao, Timothy W Holmes, Muhammad K N Afghan and David M Shepard. *Comparison of plan quality provided by intensity-modulated arc therapy and helical tomotherapy.* Int J Radiat Oncol Biol Phys, vol. 69, no. 1, pages 240–50, 2007.

[Cao *et al* 2009] Daliang Cao, F Chen, Min Rao, Muhammad K N Afghan, J Ye and David M Shepard. *Study of VMAT Plan QA Using Film, Diode Based, and Ion Chamber Based QA Systems.* Med Phys, vol. 36, no. 6, page 2422, 2009.

[Carol *et al* 1996] M Carol, W H Grant 3rd, A R Bleier, A A Kania, H S Targovnik, E B Butler and S W Woo. *The field-matching problem as it applies to the peacock three dimensional conformal system for intensity modulation.* Int J Radiat Oncol Biol Phys, vol. 34, no. 1, pages 183–7, 1996.

[Carol 1995] M Carol. *A System for Planning and Rotational Delivery of Intensity Modulated Fields.* Int J Imaging Syst Tech, vol. 6, no. 1, pages 56–61, 1995.

[Center 2003] Memorial Sloan-Kettering Cancer Center. A practical guide to intensity-modulated radiation therapy. Medical Physics Publishing, 2003.

[Cheng *et al* 1989] C W Cheng, P K Kijewski and M Langer. *Field shaping by multileaf collimators, a comparison with conventional blocks.* Med Phys, vol. 16, page 671, 1989.

[Cho *et al* 1998] P S Cho, S Lee, R J Marks 2nd, S Oh, S G Sutlief and M H Phillips. *Optimization of intensity modulated beams with volume constraints using two methods: cost function minimization and projections onto convex sets.* Med Phys, vol. 25, no. 4, pages 435–43, 1998.

[Convery & Rosenbloom 1992] D J Convery and M E Rosenbloom. *The generation of intensity-modulated fields for conformal radiotherapy by dynamic collimation.* Phys Med Biol, vol. 37, no. 6, pages 1359–74, 1992.

[Crooks *et al* 2003] S M Crooks, Xiaodong Wu, C Takita, M Watzich and Lei Xing. *Aperture modulated arc therapy.* Phys Med Biol, vol. 48, no. 10, pages 1333–44, 2003.

[Earl *et al* 2003] M A Earl, D M Shepard, S Naqvi, X A Li and C X Yu. *Inverse planning for intensity-modulated arc therapy using direct aperture optimization.* Phys Med Biol, vol. 48, no. 8, pages 1075–89, 2003.

[Fippel *et al* 1999] M Fippel, W Laub, B Huber and F Nüsslin. *Experimental investigation of a fast Monte Carlo photon beam dose calculation algorithm.* Phys Med Biol, vol. 44, no. 12, pages 3039–54, 1999.

[Galvin *et al* 1992] J M Galvin, A R Smith, R D Moeller, R L Goodman, W Powlis, J Rubenstein, L Solin, Michael B, M Needham, C Huntzenger and M Kligerman. *Evaluation of multileaf collimator design for a photon beam.* Int J Radiat Oncol Biol Phys, vol. 23, no. 4, pages 789–801, 1992.

[Galvin *et al* 1993a] J M Galvin, X G Chen and R M Smith. *Combining multileaf fields to modulate fluence distributions.* Int J Radiat Oncol Biol Phys, vol. 27, no. 3, pages 697–705, 1993.

[Galvin *et al* 1993b] J M Galvin, A R Smith and B Lally. *Characterization Of A Multileaf Collimator System.* Int J Radiat Oncol Biol Phys, vol. 25, no. 2, pages 181–92, 1993.

[Gladwish *et al* 2007] Adam Gladwish, Mike Oliver, Jeff Craig, Jeff Chen, Glenn Bauman, Barbara Fisher and Eugene Wong. *Segmentation and leaf sequencing for intensity modulated arc therapy.* Med Phys, vol. 34, no. 5, pages 1779–88, 2007.

[Hoban *et al* 1994] P W Hoban, D C Murray and W H Round. *Photon beam convolution using polyenergetic energy deposition kernels.* Phys Med Biol, vol. 39, no. 4, pages 669–85, 1994.

[ICRU 1993] ICRU. *Prescribing, Recording, and Reporting Photon Beam Therapy.* ICRU Report, vol. 50, 1993.

[ICRU 1999] ICRU. *Prescribing, Recording, and Reporting Photon Beam Therapy (Supplement to ICRU Report 50).* ICRU Report, vol. 62, 1999.

[Jursinic & Nelms 2003] Paul A Jursinic and Ben E Nelms. *A 2-D diode array and analysis software for verification of intensity modulated radiation therapy delivery.* Med Phys, vol. 30, no. 5, pages 870–9, 2003.

[Kawrakow & Fippel 2000] I Kawrakow and M Fippel. *Investigation of variance reduction techniques for Monte Carlo photon dose calculation using XVMC.* Phys Med Biol, vol. 45, no. 8, pages 2163–83, 2000.

[Kawrakow *et al* 1996] I Kawrakow, M Fippel and K Friedrich. *3D electron dose calculation using a Voxel based Monte Carlo algorithm (VMC).* Med Phys, vol. 23, no. 4, pages 445–57, 1996.

[Kawrakow *et al* 2004] Kawrakow, E Mainegra-hing and D W O Rogers. *EGSnrcMP: the multi-platform environment for EGSnrc.* Report NRC-PIRS-877, 2004.

[Keall *et al* 2000] P J Keall, J V Siebers, R Jeraj and R Mohan. *The effect of dose calculation uncertainty on the evaluation of radiotherapy plans.* Med Phys, vol. 27, no. 3, pages 478–84, 2000.

[Korreman *et al* 2009] Stine Korreman, Joakim Medin and Flemming Kjaer-Kristoffersen. *Dosimetric verification of RapidArc treatment delivery.* Acta Oncol, vol. 48, no. 2, pages 185–91, 2009.

[Langer & Leong 1987] M Langer and J Leong. *Optimization of beam weights under dose-volume restrictions.* Int J Radiat Oncol Biol Phys, vol. 13, no. 8, pages 1255–60, 1987.

[Létourneau *et al* 2004] Daniel Létourneau, Misbah Gulam, Di Yan, Mark Oldham and John W Wong. *Evaluation of a 2D diode array for IMRT quality assurance.* Radiother Oncol, vol. 70, no. 2, pages 199–206, 2004.

[Létourneau *et al* 2009] Daniel Létourneau, Julia Publicover, Jakub Kozelka, Douglas J Moseley and David A Jaffray. *Novel dosimetric phantom for quality assurance of volumetric modulated arc therapy.* Med Phys, vol. 36, no. 5, pages 1813–21, 2009.

[Ling *et al* 2008] C Clifton Ling, Pengpeng Zhang, Yves Archambault, Jiri Bocanek, Grace Tang and Thomas Losasso. *Commissioning and quality assurance of RapidArc radiotherapy delivery system.* Int J Radiat Oncol Biol Phys, vol. 72, no. 2, pages 575–81, 2008.

[Low & Mutic 1997] D A Low and S Mutic. *Abutment region dosimetry for sequential arc IMRT delivery.* Phys Med Biol, vol. 42, no. 7, pages 1465–70, 1997.

[Luan *et al* 2008] Shuang Luan, Chao Wang, Daliang Cao, Danny Z Chen, David M Shepard and Cedric X Yu. *Leaf-sequencing for intensity-modulated arc therapy using graph algorithms.* Med Phys, vol. 35, no. 1, pages 61–9, 2008.

[Ma *et al* 2001] L Ma, C X Yu, M Earl, T Holmes, M Sarfaraz, X A Li, D Shepard, P Amin, S DiBiase, M Suntharalingam and C Mansfield. *Optimized intensity-modulated arc therapy for prostate cancer treatment.* Int J Cancer, vol. 96, no. 6, pages 379–84, 2001.

[Ma *et al* 2002] C M Ma, J S Li, T Pawlicki, S B Jiang, J Deng, M C Lee, T Koumrian, M Luxton and S Brain. *A Monte Carlo dose calculation tool for radiotherapy treatment planning.* Phys Med Biol, vol. 47, no. 10, pages 1671–89, 2002.

[Mackie *et al* 1985] T R Mackie, J W Scrimger and J J Battista. *A convolution method of calculating dose for 15-MV x rays.* Med Phys, vol. 12, no. 2, pages 188–96, 1985.

[Mackie *et al* 1987] T Rock Mackie, Anders Ahnesjö, Peter Dickof and A Snider. *Development of a convolution/superposition method for photon beams.* In Proceedings of the 9th ICCR International Conference on Computers in Radiation Therapy, pages 107–10, 1987.

[Mackie *et al* 1988] T R Mackie, A F Bielajew, D W Rogers and J J Battista. *Generation of photon energy deposition kernels using the EGS Monte Carlo code.* Phys Med Biol, vol. 33, no. 1, pages 1–20, 1988.

[Mackie *et al* 1993] T R Mackie, T Holmes, S Swerdloff, P Reckwerdt, J O Deasy, J Yang, B Paliwal and T Kinsella. *Tomotherapy: a new concept for the delivery of dynamic conformal radiotherapy.* Med Phys, vol. 20, no. 6, pages 1709–19, 1993.

[Mohan & Chui 1987] R Mohan and C S Chui. *Use of fast Fourier transforms in calculating dose distributions for irregularly shaped fields for three-dimensional treatment planning.* Med Phys, vol. 14, no. 1, pages 70–7, 1987.

[Mohan *et al* 1986] R Mohan, C Chui and L Lidofsky. *Differential pencil beam dose computation model for photons.* Med Phys, vol. 13, no. 1, pages 64–73, 1986.

[Murray *et al* 1989] D C Murray, P W Hoban, P E Metcalfe and W H Round. *3-D superposition for radiotherapy treatment planning using fast Fourier transforms.* Australas Phys Eng Sci Med, vol. 12, no. 3, pages 128–37, 1989.

[Naqvi *et al* 2003] Shahid A Naqvi, Matthew A Earl and David M Shepard. *Convolution/superposition using the Monte Carlo method.* Phys Med Biol, vol. 48, no. 14, pages 2101–21, 2003.

[Nicolini *et al* 2008] Giorgia Nicolini, Eugenio Vanetti, Alessandro Clivio, Antonella Fogliata, Stine Korreman, Jiri Bocanek and Luca Cozzi. *The GLAaS algorithm for portal dosimetry and quality assurance of RapidArc, an intensity modulated rotational therapy.* Radiat Oncol, vol. 3, page 24, 2008.

[Olteanu *et al* 2006] L Olteanu, Nick Reynaert, W De Gersem, B De Smedt, C De Wagter, W De Neve and H Thierens. *Monte Carlo simulation of Intensity-modulated arc therapy plans.* In First European Workshop on Monte Carlo Treatment Planning, 2006.

[Otto 2008] Karl Otto. *Volumetric modulated arc therapy: IMRT in a single gantry arc.* Med Phys, vol. 35, no. 1, pages 310–7, 2008.

[Palma *et al* 2008] David Palma, Emily Vollans, Kerry James, Sandy Nakano, Vitali Moiseenko, Richard Shaffer, Michael McKenzie, James Morris and Karl Otto. *Volumetric modulated arc therapy for delivery of prostate radiotherapy: comparison with intensity-modulated radiotherapy and three-dimensional conformal radiotherapy.* Int J Radiat Oncol Biol Phys, vol. 72, no. 4, pages 996–1001, 2008.

[Papanikolaou *et al* 2004] Nikos Papanikolaou, Jerry J Battista, Arthur L Boyer, Constantin Kappas, Klein Klein, T Rock Mackie, Michael Sharpe and Jake Van Dyk. *Tissue inhomogeneity corrections for megavoltage photon beams.* AAPM Report No. 85, 2004.

[Roger & Bielajew 1984] Roger and A F Bielajew. *The use of EGS for Monte Carlo calculations in medical physics.* Report PXNR-2692 NRC, 1984.

[Rogers *et al* 1995] D W Rogers, B A Faddegon, G X Ding, C M Ma, J We and T R Mackie. *BEAM: a Monte Carlo code to simulate radiotherapy treatment units.* Med Phys, vol. 22, no. 5, pages 503–24, 1995.

[Rogers 1991] D W Rogers. *The role of Monte Carlo simulation of electron transport in radiation dosimetry.* Int J Rad Appl Instrum A, vol. 42, no. 10, pages 965–74, 1991.

[Rong *et al* 2009] Yi Rong, Grace Tang, James Welsh, Majid M Mohiuddin, Bhudatt R Paliwal and Cedric X Yu. *Helical tomotherapy vs. single-arc intensity-modulated arc therapy (IMAT): A collaborative dosimetric comparison between two institutions.* Int J Radiat Oncol Biol Phys, Under Review, 2009.

[Sauer *et al* 1999] O A Sauer, D M Shepard and T R Mackie. *Application of constrained optimization to radiotherapy planning.* Med Phys, vol. 26, no. 11, pages 2359–66, 1999.

[Shepard *et al* 2002] D M Shepard, M A Earl, X A Li, S Naqvi and C Yu. *Direct aperture optimization: a turnkey solution for step-and-shoot IMRT.* Med Phys, vol. 29, no. 6, pages 1007–18, 2002.

[Shepard *et al* 2007] D M Shepard, D Cao, M K N Afghan and M A Earl. *An arc-sequencing algorithm for intensity modulated arc therapy.* Med Phys, vol. 34, no. 2, pages 464–70, 2007.

[Spirou & Chui 1994] S V Spirou and C S Chui. *Generation of arbitrary intensity profiles by dynamic jaws or multileaf collimators.* Med Phys, vol. 21, no. 7, pages 1031–41, 1994.

[Stein *et al* 1994] J Stein, T Bortfeld, B Dörschel and W Schlegel. *Dynamic X-ray compensation for conformal radiotherapy by means of multi-leaf collimation.* Radiother Oncol, vol. 32, no. 2, pages 163–73, 1994.

[Sterzing *et al* 2010] Florian Sterzing, Matthias Uhl, Henrik Hauswald, Kai Schubert, Gabriele Sroka-Perez, Yu Chen, Weiguo Lu, Rock Mackie, Jürgen Debus, Klaus Herfarth and Gustavo Oliveira. *Dynamic Jaws and Dynamic Couch in Helical Tomotherapy.* Int J Radiat Oncol Biol Phys, vol. 76, no. 4, pages 1266–73, 2010.

[Takahashi 1965] S Takahashi. *Conformation radiotherapy. Rotation techniques as applied to radiography and radiotherapy of cancer.* Acta Radiol Diagn (Stockh), pages Suppl 242:1+, 1965.

[Tang & Yi 2009] Grace Tang and Byong Yong Yi. *Patient-Specific QA of Intensity-Modulated Arc Therapy with 2D Diode Array: Initial Experience.* Med Phys, vol. 36, no. 6, pages 2574–75, 2009.

[Tang *et al* 2007] Grace Tang, Matthew A Earl, Shuang Luan, Shahid A Naqvi and Cedric X Yu. *Converting multiple-arc intensity-modulated arc therapy into a single arc for efficient delivery.* Int J Radiat Oncol Biol Phys, vol. 69, no. 3, page S673, 2007.

[Tang *et al* 2008] Grace Tang, Matthew A Earl, Shuang Luan, Chao Wang, Daliang Cao, Cedric X Yu and Shahid A Naqvi. *Stochastic versus deterministic kernel-based superposition approaches for dose calculation of intensity-modulated arcs.* Phys Med Biol, vol. 53, no. 17, pages 4733–46, 2008.

[Tang *et al* 2009a] Grace Tang, Kristen Asbury, Amy Aurand, Majid M Mohiuddin, Young Kwok, Bin Zhang, Lisa Richards, Kim Marter and Cedric X Yu. *Plan Quality Comparison of Intensity-Modulated Radiation Therapy and RapidArc for multiple cancer sites.* In Preperation, 2009.

[Tang *et al* 2009b] Grace Tang, Matthew A Earl and Cedric X Yu. *Variable dose rate single-arc IMAT delivered with a constant dose rate and variable angular spacing.* Phys Med Biol, vol. 54, no. 21, pages 6439–56, 2009.

[Tang *et al* 2010] Grace Tang, Matthew A Earl, Shuang Luan, Chao Wang, Majid M Mohiuddin and Cedric X Yu. *Comparing Radiation Treatments Using Intensity-Modulated beams, Multiple Arcs and Single Arcs.* Int J Radiat Oncol Biol Phys, vol. 76, no. 5, pages 1554–62, 2010.

[Van Dyk 1999] Jake Van Dyk. The modern technology of radiation oncology. Medical Physics Publishing, 1999.

[Van Esch *et al* 2007] Ann Van Esch, Christian Clermont, Magali Devillers, Mauro Iori and Dominique P Huyskens. *On-line quality assurance of rotational radiotherapy treatment delivery by means of a 2D ion chamber array and the Octavius phantom.* Med Phys, vol. 34, no. 10, pages 3825–37, 2007.

[Wang *et al* 2008] Chao Wang, Shuang Luan, Grace Tang, Danny Z Chen, Matt A Earl and Cedric X Yu. *Arc-modulated radiation therapy (AMRT): a single-arc form of intensity-modulated arc therapy.* Phys Med Biol, vol. 53, no. 22, pages 6291–303, 2008.

[Webb 1992] S Webb. *Optimization by simulated annealing of three-dimensional, conformal treatment planning for radiation fields defined by a multileaf collimator: II. Inclusion of two-dimensional modulation of the x-ray intensity.* Phys Med Biol, vol. 37, no. 8, pages 1689–704, 1992.

[Wong *et al* 2002] Eugene Wong, Jeff Z Chen and Jonathan Greenland. *Intensity-modulated arc therapy simplified.* Int J Radiat Oncol Biol Phys, vol. 53, no. 1, pages 222–35, 2002.

[Woo & Cunningham 1990] M K Woo and J R Cunningham. *The validity of the density scaling method in primary electron transport for photon and electron beams.* Med Phys, vol. 17, no. 2, pages 187–94, 1990.

[Xia & Verhey 1998] P Xia and L J Verhey. *Multileaf collimator leaf sequencing algorithm for intensity modulated beams with multiple static segments.* Med Phys, vol. 25, no. 8, pages 1424–34, 1998.

[Yi *et al* 2009] Byong Yong Yi, S Han-Oh, Fritz Lerma, B Berman and Cedric X Yu. *Fluoroscopic Verification of Intensity-Modulated Rotational Arc Therapy Delivery.* Med Phys, vol. 36, no. 6, page 2708, 2009.

[Yu *et al* 2002] Cedric X Yu, X Allen Li, Lijun Ma, Dongjun Chen, Shahid Naqvi, David Shepard, Mehrdad Sarfaraz, Timothy W Holmes, Mohan Suntharalingam and Carl M Mansfield. *Clinical implementation of intensity-modulated arc therapy.* Int J Radiat Oncol Biol Phys, vol. 53, no. 2, pages 453–63, 2002.

[Yu *et al* 2006] Cedric Yu, David Shepard, Matt Earl, Daliang Cao, Shuang Luan, Chao Wang and Danny Z Chen. *New developments in intensity modulated radiation therapy.* Technol Cancer Res Treat, vol. 5, no. 5, pages 451–64, 2006.

[Yu 1995] C X Yu. *Intensity-modulated arc therapy with dynamic multileaf collimation: an alternative to tomotherapy.* Phys Med Biol, vol. 40, no. 9, pages 1435–49, 1995.