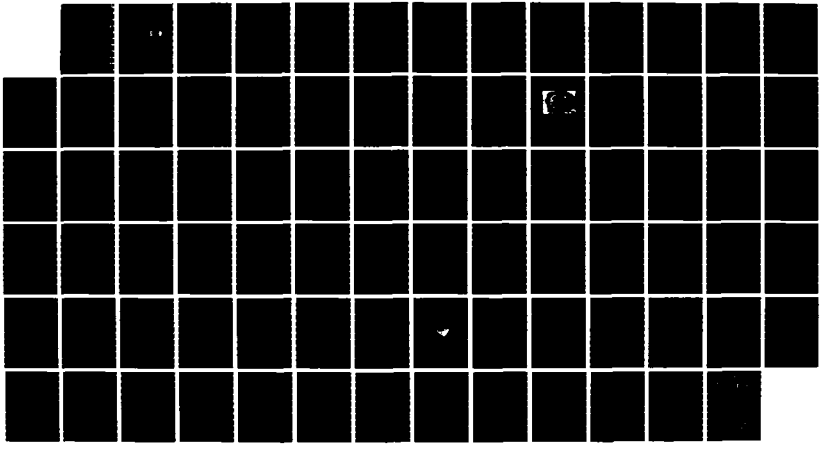
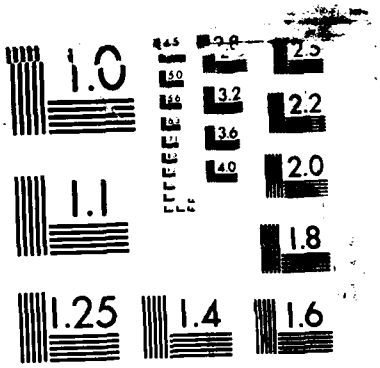
AD-A168 179 RAY-TRACING IN THREE DIMENSIONS FOR CALCULATION OF RADIATION DOSE CALCULATIONS(U) ARMY MILITARY PERSONNEL CENTER ALEXANDRIA VA PROGRAM ANALYSIS OFFICE D R KENNEDY 27 MAY 86 1/1

UNCLASSIFIED F/G 6/18 NL





MICROCOPY RESOLUTION TEST CHART

Ray-Tracing in Three Dimensions for Calculation of Radiation Dose Calculations

Cpt Dennis R. Kennedy
HQDA, MILPERCEN (DAPC-OPA-E)
200 Stovall Street
Alexandria, VA 22332

27 May 1986

A thesis submitted to University of Washington in Seattle, WA in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD A168 179	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RAY-TRACING PRISMS IN THREE DIMENSIONS FOR CALCULATION OF RADIATION DOSE DISTRIBUTIONS.		5. TYPE OF REPORT & PERIOD COVERED THESIS, 23 MAY 1986
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) DENNIS R. KENNEDY		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Student, HQDA, MILPERCEN (DAPC-OPA-E) 200 Stovall Street Alexandria, VA 22332		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS HQDA, MILPERCEN, ATTN: DAPC-OPA-E 200 Stovall St. Alexandria, VA 22332		12. REPORT DATE 23 May 1986
		13. NUMBER OF PAGES 68
14. MONITORING AGENCY NAME & ADDRESS (If different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Thesis, University of Washington, Seattle, Wa 98195

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

RAY-TRACING
PRISMS
RADIATION THERAPY TREATMENT PLANNING
STRIP TREES

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis addresses several methods of calculating the radiation dose distribution for use by technicians or clinicians in radiation therapy treatment planning. It specifically covers the calculation of the effective pathlength of the radiation beam for use in beam models representing

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

the dose distribution. A two-dimensional method by Bentley and Milan is compared to the method of Strip Trees developed by Duda and Hart and then a three-dimensional algorithm built to perform the calculations in three-dimensions. The use of PRISMS conforms easily to the obtained CT Scans and provides a means of only doing two-dimensional ray-tracing while performing three-dimensional dose calculations. This method is already being applied and used in actual calculations.

Ray-Tracing Prisms in Three-Dimensions for Calculation of Radiation Dose Distributions

by

Dennis Roy Kennedy

A thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science

University of Washington

1986

Approved by Ira J. Kalet

(Chairperson of Supervisory Committee)

Program Authorized
to Offer Degree Department of Computer Science

Date 13 May 86

Master's Thesis

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this thesis is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U. S. Copyright Law. Any other reproduction for any purpose or by any means shall not be allowed without my written permission.

Signature_____

Date_____

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to Ira Kalet for his direction and guidance in the preparation of this thesis and especially for the manner in which he did so. The opportunity to earn my thesis and provide a usable item in such a worthwhile area has made this a truly enjoyable experience.

Table of Contents

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RAY-TRACING PRISMS IN THREE DIMENSIONS FOR CALCULATION OF RADIATION DOSE DISTRIBUTIONS.		5. TYPE OF REPORT & PERIOD COVERED THESIS, 23 MAY 1986
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Dennis R. Kennedy		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Student, HQDA, MILPERCEN(DAPC-OPA-E) 200 Stovall St. Alexandria, VA 22332		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS HQDA, MILPERCEN ATTN: DAPC-OPA-E 200 Stovall St Alexandria, VA 22332		12. REPORT DATE 23 May 1986
		13. NUMBER OF PAGES 68
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Thesis, University of Washington, Seattle, Wa 98195

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

RAY-TRACING
PRISMS
RADIATION THERAPY TREATMENT PLANNING
STRIP TREES

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis addresses several methods of calculating the radiation dose distribution for use by technicians or clinicians in radiation therapy treatment planning. It specifically covers the calculation of the effective pathlength of the radiation beam for use in beam models representing the

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

dose distribution. A two-dimensional method by Bentley and Milan is compared to the method of Strip Trees developed by Duda and Hart and then a three-dimensional algorithm built to perform the calculations in three-dimensions. The use of PRISMS conforms easily to the obtained CT Scans and provides a means of only doing two-dimensional ray-tracing while performing three-dimensional dose calculations. This method is already being applied and used in actual calculations.

List of Tables

List of Figures

Chapter 1

Definitions

The following terms are defined for the reader's clarification and are *emphasized* at their first occurrence in the text.

CONTOUR: A contour is a collection of points representing a curve. The collection of points is determined to represent the patient's body, major organs or a change in tissue density as determined by a computerized tomography (CT) Scan. Therefore, in this application the curve will always be a closed curve and there are no contour-contour intersections.

INHOMOGENEITY: In this context inhomogeneity merely means that the subject to be irradiated is made of matter of varying densities.

PATHLENGTH: The length of the beam from source to calculation point or point of interest. The calculation point is typically moved through the patient in the area of the tumor and any sensitive organs that may be affected by the radiation to present a dose distribution across this area.

DENSITY: In the context of radiation therapy, density(ρ), can be defined as the ratio of the mass energy absorption coefficient of the material to that of water, e.g.:

$$\rho = \frac{\mu_{material}}{\mu_{water}}$$

EFFECTIVE PATHLENGTH: This is the *pathlength* of the beam through the body taking into account the inhomogeneities. It is determined by computing the length of the beam

through each contour of a different density and incorporating the density of the contours into the pathlength. The basic formula for determining the effective pathlength is:

$$EffectivePathlength = \int_{SourcePt}^{CalculationPt} densityofpath\ d(length)$$

The effective pathlength yields the radiation dose at the point of interest by using the derived radiation dose data for the homogeneous medium water at a depth equal to the effective pathlength. Further discussion of the effective pathlength is contained in Section 3.6.

PLANE UNIT NORMAL: The plane unit normal for a plane in 3-space is the vector that is perpendicular to the plane and is of length 1. It can be most easily found by extracting the constants A, B, and C from the equation for the plane, using $Ax + By + C = 0$ as the equation of the normal, and then dividing by the length $\sqrt{A^2 + B^2 + C^2}$.

DIRECTION COSINES: The direction cosines are the angles formed by a line with each of the axes.

SCALAR EQUATIONS: The symmetric scalar equations are a representation of the relationship of the change of each of the coordinates along a line. The three-dimensional scalar equation of a line through two points $P_0(x_0, y_0, z_0)$ and $P_1(x_1, y_1, z_1)$ is written:

$$\frac{x - x_0}{x_1 - x_0} = \frac{y - y_0}{y_1 - y_0} = \frac{z - z_0}{z_1 - z_0}$$

BEAM MODEL: A mathematical description of the absorbed dose of radiation produced by a single radiation beam in a patient equivalent medium.

PATIENT EQUIVALENT MEDIUM: A medium for which the absorbed dose is known from laboratory data. All material is related to water, chosen density 1.0, which most closely approximates the majority of the human body.

ISODOSE CHARTS: The distribution of absorbed dose produced by a single beam in a reference geometry. A digitized isodose chart is one that is represented by a two-dimensional matrix for computer data input and interpolation. See Figure 3.2 on page 12.

DOSIMETER: A radiation sensitive device. In most cases, dosimeter refers to the radiation sensitive device and its reading equipment, e.g., ionization chamber (device) and electrometer (reader), or film and densitometer.

CT SCAN: Short for Computerized Tomography Scan. A numerical representation of the attenuation coefficients of x-rays in the tissue being scanned, which is then projected as a gray scale image of these coefficients. This means that based on the attenuation of the x-rays, the gray scale image will represent the varying densities of the body. For an example see Figure 3.1 on page 10. Also known as a CAT (Computer Assisted Tomography) Scan.

TRANSVERSE AXIAL TOMOGRAPH: An earlier and rougher version of the CT Scan where the patient was in the sitting position. The image was not as clear and its results were not as effective as the CT Scan.

ALDERSON RANDO PHANTOM: Mock-up of the human body with major internal organs represented.

PENUMBRA: The region on the edges of a radiation beam where a transition occurs from points that are exposed to primary radiation from the whole beam source to points that are exposed to no primary radiation. It is denoted as P in Figure 1.1.

TISSUE-AIR RATIO(TAR): The ratio formed by irradiating water(known in radiation oncology as a water phantom) and air with a beam of a specific radius at a specific depth. It is represented by the following formula:

$$T(d, r_d) = \frac{R_X}{R_{X'}}$$

where: R_X is the radiation dose reading at depth d in water of a beam with radius r_d.
$R_{X'}$ is the same reading in air.

TISSUE-MAXIMUM RATIO(TMR): Ratio similar to the TAR above except that it is the ratio formed by the the irradiation of a tissue mass over that of a water phantom,e.g.:

$$TMR(d, r_d) = \frac{R_Y}{R_X}$$

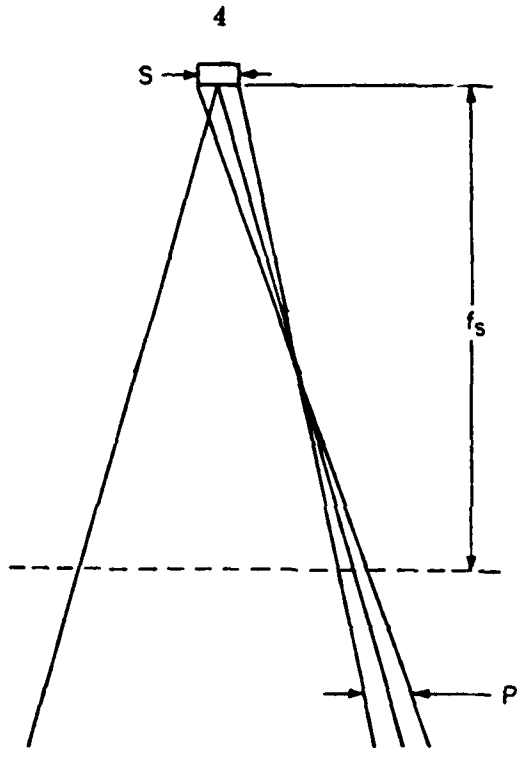


Figure 1.1: Radiation Beam showing the Source(S), the Penumbra(P), and the Source to Surface Distance(f_s).

where: R_Y is the radiation dose for the tissue mass at a depth d.

R_X is the reading in water as above.

DOSE DISTRIBUTION: The manner in which the total radiation dosage received from all radiation sources is distributed over the patient cross-section. Since there can be multiple radiation sources it is the sum of the individual distributions of these sources. It is normally displayed to the technician or clinician by a wireframe diagram of specific radiation levels, e.g., frames showing 30, 60, 90, and 150 rads.

SOURCE TO SURFACE DISTANCE (SSD): This is the distance from the radiation beam source to the surface of the tissue to be irradiated. It is represented by f_s in Figure 1.1.

Chapter 2

Introduction

With the continued advances in technology and computer hardware and software an increasing number of specific computer applications are being presented in the area of medical treatment. One of these specific areas is cancer research, treatment and planning. This investigation deals with the subtopic of radiation therapy treatment planning where the goal is to accurately predict the physical effects of a proposed dose of radiation on a cancer or other malignant disease, the adjacent non-malignant tissue, and any sensitive organs. This requires defining the patients' physical dimensions, the dimensions and location of the tumor, the arrangement of the radiation sources, and the desired region of calculation of the radiation dose. The computer program must be able to calculate and display the resulting predicted radiation *dose distribution*. In order to select the optimal treatment, the physician must be able to evaluate many simulated treatments and ascertain the most effective and safe treatment. This has led to the development of several methods of calculating the dose distribution for a beam source external to the patient. The method used in this investigation involves finding the length of the path of an external radiation source through a three-dimensional variable *density* mass representing the human body, determining the *effective pathlength* of the beam, and using one of the beam models to determine the dose at that pathlength. (For definitions see Chapter 1).

The effective pathlength computation involves tracing the radiation beam through the *contours* of interest. This procedure is called ray-tracing and is similar to the ray-tracing

used in computer graphics in such areas as surface rendering or shading. In graphics these rays are traced from the viewer, through the corners of the picture elements to or through the desired object. The ray is then intersected with the object to determine the angle of intersection for use in shading or light reflection calculations. In this application the radiation beam becomes the 'focal point' of the viewer and the beam is traced through the desired volume to a specified array of points. The rays are traced through the representation of the patient and the effective pathlength of each ray calculated to present a dose distribution to the clinician.

This thesis encompasses the following major areas:

1. A discussion of the history and background of representing patient dimensions, *beam models* used in computer calculations, and methods of determining the effective pathlength.

2. The design and implementation of the two dimensional ray-tracing or intersection methods compared here. The first is an adaptation of the Bentley-Milan Method [1] as accomplished by Dr Ira J. Kalet of the University Hospital Radiation Oncology Department. The second is a design and implementation of the Duda and Hart Strip Trees [2] for representation of the patient contours and then intersection of the beam with the contours.

3. The development of a three-dimensional algorithm with the incorporation of the two above methods and a speed comparison.

---

[1] J. Milan and R. E. Bentley, The Storage and Manipulation of Radiation Dose Data in a Small Digital Computer, *British Journal of Radiolgy*, Volume 47, Number 554, February 1974, pp. 115-121.

[2] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, 1973.

Chapter 3

The Use of the Computer in Radiation Therapy

In this chapter the history of the use of computers in radiation therapy planning and treatment will be covered. Although a short history of slightly more than a quarter century, it is one that has already been through a growth period and then experienced a period of stagnation in the application of the methods and procedures developed. In this chapter the following areas will be addressed: the clinical requirements for radiation treatment software, methods for modelling the external and internal regions of the patient, the basic beam models developed for an idealized patient, requirements and methods for altering the idealized model, and finally three methods of computing the effective pathlength.

3.1 The Beginning

K. C. Tsien [1] is usually credited with the first application of automatic computing machinery to dosage calculations in 1954-55. He reduced *isodose charts* for a single radiation beam to tabular form and stored the data on punch cards. This could then be manipulated to produce dose distributions for multiple beams whose possible locations were specified by the clinicians. Beginning with this attempt, further developments and procedures were

[1] K. C. Tsien, Application of Automatic Computing Machinery to Radiation Dose Calculations, *British Journal of Radiology*, Volume 28, Number 332, August 1955, pp. 432-439.

investigated to ease the workload and increase the efficiency of the calculations for the technicians and clinicians in radiation therapy. The actual methods developed by the investigators were determined to a large extent by the locally available computer facilities and to a greater extent by their involvement in actual clinical problems.

3.2 Clinical Requirements

Technically, beam therapy involves the manipulation of a finite set of parameters which control the behavior of the beam in time and space relative to a target in a patient. Radiotherapy is still mostly empirical in nature and not all of the criteria for optimization of radiation therapy in any given case is known. Therefore, the purpose of these computer methods is to provide the radiotherapist with sufficiently accurate information on the physical aspects of the proposed treatment of a patient. This information must be produced within a reasonable amount of time, require minimal preparatory action, and be displayed in easily and directly interpretable form. This allows management and/or manipulation of the input parameters to optimize the proposed treatment.

There are two essential areas in the radiation therapy planning process. These are the patient model and the beam model. The patient model is an adequate description of the relevant portion of the patient in terms of shape and composition. The beam model is a mathematical description of the absorbed dose produced by a single beam in a *patient equivalent medium*.

3.3 The Patient Model

The patient model requires that the external contour and internal structure of the patient be adequately represented in some manner. For many years representing the patient's external contour in two dimensions was considered acceptably accurate. Some of the more common techniques included the interlocking beads technique, the dipstick

principle (not automotive), the parallel prong device, and the pantograph.[2] The internal structure was approximated for many years by outlines taken from a representation of the human body in a cross-sectional anatomy atlas. This was very crude but the only way, short of opening up the patient. Eventually some treatment centers were able to use a *transverse axial tomograph* that gave better information, but the use of the *CT Scan*, with its first commercial appearance in 1972, facilitated the representation of the internal structures. This also led to increased emphasis on the handling of *inhomogeneities* in the beam model, which will be covered in Section 3.5.

The importance of the accuracy of the internal representation was demonstrated by Sontag et. al.[3] One of the sections of an *Alderson Rando Phantom* of the chest region was radiographed and then *dosimeters* placed inside. The phantom was reassembled and irradiated by a cobalt beam with the dosimeters measuring the actual dose received. Using the old method of the cross-sectional anatomy atlas and the newer method where the radiograph represents the CT Scan, the dosages were calculated and compared to the actual dose. In the anatomy atlas method, when no correction was made for the different densities of the lungs and bones, the difference between calculated and actual dosage was on the average about 10%, with a maximum difference of 30%. When a density correction method was applied, improved accuracy was obtained for some points, but for points in the area of the lung wall where the largest change of density took place maximum differences increased. When the correct anatomical information was used (radiograph), the difference between calculated and measured dosages was 3.4% on the average and 7% on the maximum. It is the use of the CT Scan that has made the proper representation of both the internal structures and surface outline more viable and allowed much greater

[2]Michael J. Day and R. M. Harrison, Cross-Sectional Information and Treatment Simulation, *Radiation Therapy Planning*, Bleehen, Glatstein, and Haybittle, ed., 1983, pp. 89-95.

[3]M. R. Sontag, J. J. Battista, M. J. Bronskill, and J. R. Cunningham, Implications of Computed Tomography for Inhomogeneity Corrections in Photon Beam Dose Calculations, *Radiology*, Volume 124, Number 1, July 1977, pp. 143-149.

accuracy in the patient model and therefore in the outcome of the treatment planning process. For an example of a CT Scan see Figure 3.1.

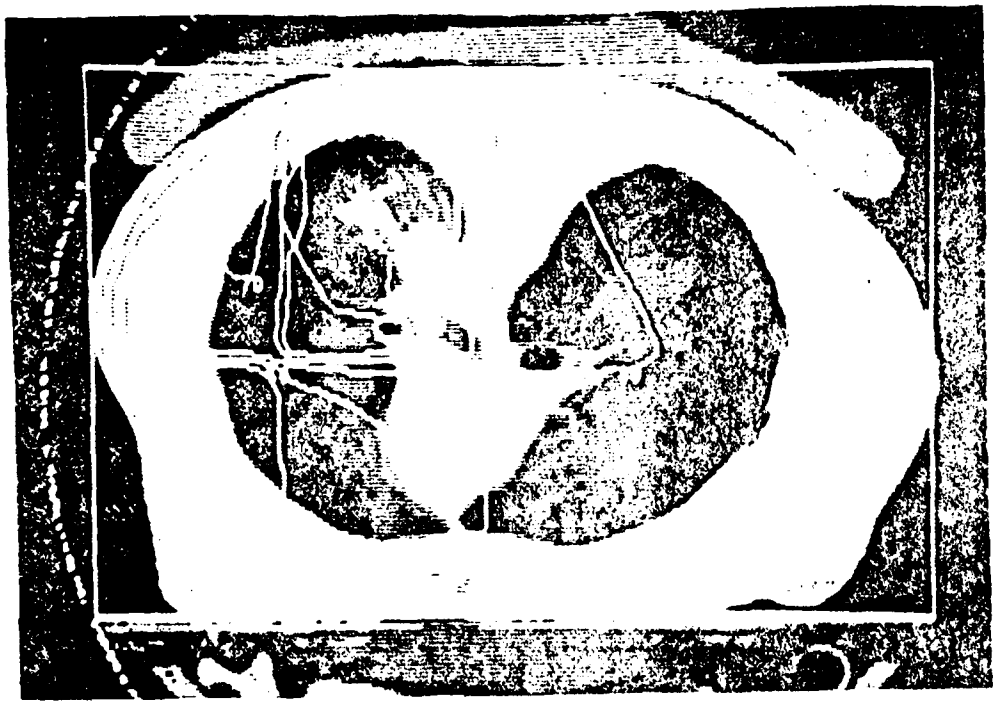


Figure 3.1: Example of a CT Scan

3.4 Beam Models

The beam models used fall into a limited number of categories. In fact, the basic formula for the accumulated dose at a point (x, y, z) is the same for all models and can be stated as follows:

$$D(x, y, z) = N \sum_{k} P_k(x', y', z') W_k C_k(x', y', z')$$

where:

- (x, y, z) is the point in the patient coordinate system.
- (x', y', z') is the same point in the beam coordinate system.

- N is a suitable normalization factor.
- P_k is the amount of dose at (x', y', z') in an idealized water-equivalent geometry. This factor is known as the percentage depth dose.
- W_k is the weight factor for beam k.
- C_k is the total correction applied for any difference between the situation for which P_k refers and the actual situation.

In the next four subsections the structures of several types of beam models, i.e., bases for the percentage depth dose (P_k) will be discussed, and then corrections necessary for surface curvature and inhomogeneity will be covered.

3.4.1 Isodose Charts

The first beam model method to be discussed is the isodose chart. The digitized isodose chart, upon which K. C. Tsien first applied computing machinery to dose distribution calculations, is used to compute directly the dose distributions from the isodose matrix determined by superimposing a grid matrix of points over an experimentally determined dose distribution. The value at any arbitrary point is obtained by applying some method of interpolation between the points. The accuracy of this, of course, depends on the spacing of the grids and the gradient within that area. For an example of an isodose chart see Figure 3.2 on page 12.

An example of an economically stored digitized isodose chart that became a commercial treatment planning system was the Programmed Console(PC) developed by W. E. Powers and J. R. Cox of St. Louis.[4] In this application a nonorthogonal coordinate system was used in which lines diverging from the radiation source and lines at a series of depths defined the grid. The PC was a forerunner of a number of commercial systems since it

[4] J. R. Cunningham and J. Milan, Chapter 6 in *Computers in Biomedical Research*, Stacey and Waxman, ed., New York, 1969, page 159.

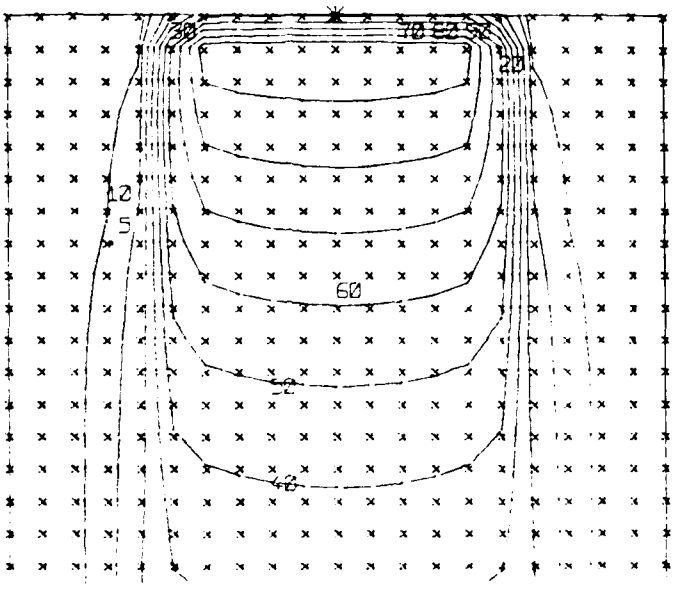


Figure 3.2: Isodose Chart Using Standard Grid Matrix

pioneered the concept of a small dedicated computer for radiation therapy. It was also the first system designed specifically with two basic principles in mind: special adaptation to graphical input and output and minimum cost. The PC handled graphical input by a special device called a rho-theta device with a linear and circular potentiometer. The programs were entered on magnetic strip cards. Manipulation of the viewing window and beam sources on an oscilloscope display allowed some ability of optimal treatment planning and then the chosen distribution could be printed on an incremental plotter.

The main problem with digitized isodose charts is that the information is *at best* of the same quality as that obtainable by the manual conventional method, mainly because it is a statement of dose as a function of position. A serious disadvantage is that it does not offer the possibility of extension into the third dimension. Due to these problems and those with the acquisition and storage of the charts this method did not become widely used.

3.4.2 Special Functions

The second beam model method to be covered is the use of special functions to represent the beam distribution. Several mathematical functions were developed during the decade of 1964-74 which approximate the absorbed dose at any arbitrary point. Four of these are presented here as a representation of the development of usuable functions of this period, two developed for two-dimensional use and two for three-dimensional. All four were specifically developed for routine clinical use with digital computers with the primary objective of reducing data over the digitized isodose chart.

The formula for the dose common to all of these mathematical functions is the product of two functions:

$$D(x, y, z) = P(y) \times F(x, z) \tag{3.1}$$

where:

- P(y) is the percentage depth dose on the central ray at depth y.
- F(x,z) is a function expressing the ratio of the dose at point (x,z) with depth y to the dose on the axis at depth y.

This section will present four methods of expressing F(x,z) (F(x) in the two-dimensional case).

The first two-dimensional function to be covered was developed by Richter and Schirrmeister.[5,6] They adapted an existing empirical expression for the central ray dose of a Cobalt-60 beam and included the off-axis behavior by means of complicated curve fitting.

[5]J. Richter and D. Schirrmeister, Ein Verfahren zur Berechnung der Dosisverteilungen mit digitalen Rechenautomaten (A Procedure for Calulating Dose Distribution with Computers), *Strahlentherapie*, Volume 123, Number 1, January 1964, pp. 45-58.

[6]J. Richter and D. Schirrmeister, Die Berücksichtigung von Gewebeinhomogenitäten bei der Ermittlung von Dosisverteilungen mit digitalen Rechenautomaten (Handling Tissue Inhomogeneities in the Calculation of Dose Distribution on Computers), *Strahlentherapie*, Volume 127, Number 4, August 1965, pp. 550-559.

They had to develop different equations for different values of a parameter ξ. It represented the relationship between the distance of a point off the central axis of the beam to the fractional difference in the *Source to Surface Distance*(SSD) and the SSD plus depth of the tissue.

Another two-dimensional function was developed by Siler and Laughlin.[7] They were the first to introduce a semi-empirical function for the calculation of the absorbed dose at an arbitary point, though still in a single plane. The term $F(x)$ in equation 3.1 (no z term since only 2D) was called the 'off-center ratio' in reference to the distance off the central axis of the beam. This idea is closely related to the 'decrement line' concept to be covered in Section 3.4.3.

In the domain of three-dimensional functions Sterling et. al.[8] employed mathematical curve-fitting techniques to obtain expressions for P(y) and F(x,z). They developed a cumulative, normal probability distribution for F(x,z) which represented the dose at a point x-distance and z-distance away from the central axis of the beam, at depth y. This was mainly chosen because the sigmoidal shape resulted in a close fit to experimental dose profile data.

The second of the three-dimensional functions was developed by Van de Geijn.[9] In his work the formula was expressed in terms of the appropriate *tissue-air ratio*, the back-scatter factor, the inverse square of the fraction relating the distance to the peak dose to the distance to the point of interest, and the product of the x and z distance off-axis ratios. It was an extremely useful model and was incorporated into a commercial treatment planning system called the Philips TPS(Treatment Planning System). This model was developed

[7]W. Siler and J. S. Laughlin, A Computer Method for Radiation Treatment Planning, *Communications of the ACM*, Volume 5, Number 7, July 1962, pp. 407-408.

[8]T. D. Sterling, H. Perry, and L. Katz, Automation for Radiation Treatment Planning, IV, Derivation of a Mathematical Expression for the Per Cent Depth Dose Surface of a Cobalt-60 Beam and Visualization of Multiple Field Dose Distributions, *British Journal of Radiology*, Volume 37, Number 439, July 1964, pp. 544-560.

[9]J. Van de Geijn, Computational Methods in Beam Therapy Planning, *Computer Programs in Biomedicine*, Volume 2, Number 3, 1972, pp. 153-168.

further by a group at Memorial Hospital into a widely used commercial method that is covered in Section 3.5.5.

These four mathematical beam models succeeded in making the representation of radiation distribution easier for computer calculation. The next step was to adapt these idealized methods for body curvature and inhomogeneities, and then investigate methods of representing the contours and finding the depth to the calculation point or point of interest.

3.4.3 Decrement Lines

Another way of representing a radiation beam was proposed by Orchard.[10] This method is very similar to the isodose chart method in which a grid system is used to cover the beam representation. However in this case, lines are drawn from the source as a loci of points where the doses are fixed fractions of the dose at that same depth on the central axis of the beam. (See Figure 3.3)

This method is a statement of dose as a function of position similar to the isodose chart and is therefore not particularly useful for computer input.

3.4.4 Separation of Absorbed Dose

The last of the four methods for representing beam models to be covered was developed by Cunningham [11] in whose work the total dose of radiation is separated into the primary and scattered components. Both components are calculated by semi-empirical means. By separating the two components, Cunningham developed a method that, while accurate, was too slow for everyday use and diffucult to adapt for the non-idealized situation. It is more often used for the calculation of basic data for special situations or other algorithms.

[10]P. G. Orchard, Decrement Lines: A New Presentation of Data in Cobalt-60 Beam Dosimetry, *British Journal of Radiology*, Volume 37, Number 442, October 1964, pp. 756-763.

[11]J. R. Cunningham, Computer Methods for Calculation of Dose Distribution, *Radiation Therapy Planning*, Bleehen, Glatstein, and Haybittle, ed.,1983, pp. 217-263.

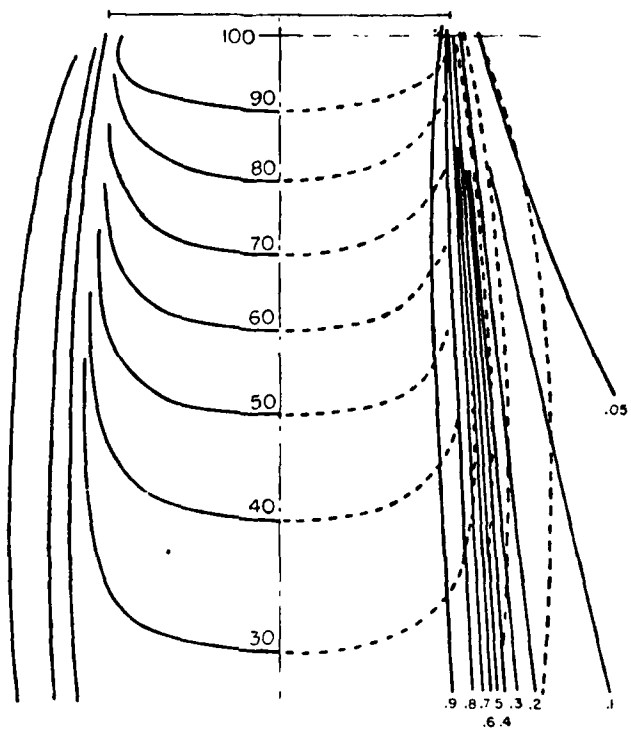


Figure 3.3: Decrement Lines on a Beam Representation

3.5 Corrections to Idealized Models

All basic beam models refer to the idealized situation: a regular-shaped beam at right angles to a flat surface of a homogeneous medium of water density(1.0). If the area to be irradiated is at an angle to the direction of the beam, then account must be taken of this angle. In all the beam models the actual pathlength is used so obliquity is not addressed in these methods. Only in the isodose chart based methods and methods accounting for a scattered dose component are corrections necessary for body curvature. Since the treatment may also require irradiation through a portion of the body that has varying densities, all beam models must be adapted to account for inhomogeneities. The following sections will cover some of the more common methods of dealing with body curvature and inhomogeneities.

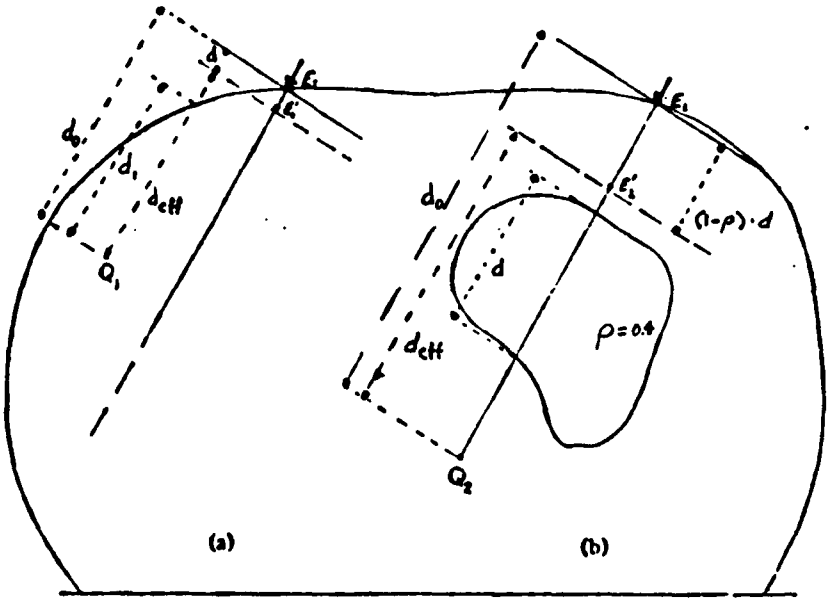


Figure 3.4: Isodose Shifts for Obliquity and Inhomogeneity

3.5.1 Isodose Chart Corrections

In methods using the isodose chart as a basis(isodose matrix models, decrement lines, etc.), an 'isodose shift' is used as a correction for obliquity and inhomogeneity. The body curvature method involves 'shifting down' an isodose chart. Using Figure 3.4(a) as an example, the depth of Q_1, d_1, is measured parallel to the central axis of the beam. The length d is found from $d = a(d_0 - d_1)$ where the value recommended for 'a' varies from $\frac{1}{2}$ to $\frac{1}{3}$. To read the corrected dose value for Q_1 from an isodose chart, the chart is shifted distance d along the line indicating the beam axis and the chart now reads correctly for points along the line through Q_1 parallel to the central axis of the beam.

The correction for inhomogeneities is similar in that the chart is shifted a specified distance. In this case the shift distance is proportional to the pathlength in or through the inhomogeneity. Using Figure 3.4(b) the length of the ray to Q_2 is d_0 and the length through the inhomogeneity is d. The shift distance is found by relating the density of the inhomogeneity to the equivalent medium density(1.0). This distance, the equivalent depth, becomes $d_{eq} = d_0 + (\rho - 1)d$, and the isodose shift distance becomes a downward

shift of $(1-\rho)d$ for $\rho < 1$ and an upward shift of $(\rho-1)d$ for $\rho > 1$.

3.5.2 Batho Power Law

Sontag and Cunningham[12] built upon a method developed by H. F. Batho for handling inhomogeneities. The basis for this method is to develop a relationship between the tissue-air ratios for the different density media. This results in a correction factor(CF) that, multiplied by the idealized dose for the point of interest in a homogeneous medium, gives the adjusted dose for this point.

Referring to Figure 3.5, Batho developed an equation for the correction factor as follows:

$$CF = \left[\frac{T(d_1, A)}{T(d_2, A)}\right]^{1-\rho_b} \tag{3.2}$$

where A is the field dimensions of the beam and the tissue-air ratios are determined by:

1. $T(d_1, A)$ is the tissue-air ratio of the point of interest at depth d_1 in tissue with density ρ_a where in Batho's formula $\rho_a = 1.0$(equivalent density).

2. $T(d_2, A)$ is the tissue-air ratio of the point of interest at depth d_2 in tissue with density ρ_b.

Sontag and Cunningham generalized the equation for the case when the point of interest lies within a medium of density other than one, and accounted for the differences in the atomic numbers of the different density masses by adding the ratio for the mass energy absorption coefficients, μ. Their resulting formula is called the 'Generalized Batho Equation' and is:

$$CF = \frac{T(d_1, A)^{\rho_a - \rho_b}}{T(d_2, A)^{1-\rho_b}} \frac{\left(\frac{\mu_{en}}{\rho}\right)_{\rho_a}}{\left(\frac{\mu_{en}}{\rho}\right)_{\rho_b}} \tag{3.3}$$

[12]M. R. Sontag and J. R. Cunningham, Corrections to Absorbed Dose Calculations for Tissue Inhomogeneities, *Medical Physics*, Volume 4, Number 5, September/October 1977, pp. 431-436.

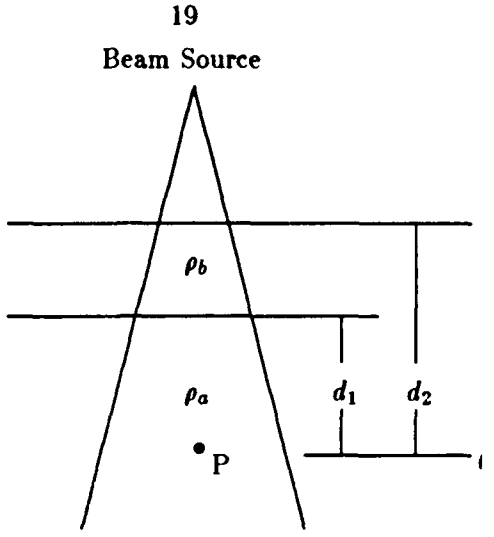


Figure 3.5: Inhomogeneity Example for the Batho-Power Law

where the point of interest lies in the mass at depth d_2 with a density of ρ_a and the overlying inhomogeneity is of thickness $(d_2 - d_1)$ which is the same as it was for the equation 3.2. The only difference in the tissue-air relationship from equation 3.2 and equation 3.3 is that ρ_a is not assumed to be the equivalent density of 1.0.

Using this equation, the pathlength is found and this CF used to account for the inhomogeneity. For multiple inhomogeneities a CF is determined for each homogeneity individually and a composite correction factor calculated by the product of all the individual correction factors. The actual pathlength of the ray is used and therefore no additional correction factors are required for obliquity.

While the experimental results in this paper show excellent accuracy, Wong and Henkelman[13] did more extensive comparisons and calculations and showed that large beam fields and the multiplicative rule of correction factors for multiple inhomogeneities

[13]J. W. Wong and R. M. Henkelman, Reconsideration of the Power-Law(Batho) Equation for Inhomogeneity Corrections, *Medical Physics*, Volume 9, Number 4, July/August 1982, pp. 521-530.

diminish the accuracy of the Batho correction. They proposed several changes to decrease the errors in these cases. However, the ray-tracing effort is still required to determine the pathlength through the heterogeneous medium, then the idealized dose must be determined, the correction factors for each inhomogeneity calculated, and finally the combined CF multiplied by the idealized dose for every point to be charted. For everyday use, the speed of the computations appears to be the determining factor for not using this method for treatment planning. However, it would be extremely useful for the calculation of basic data for special situations or other algorithms.

3.5.3 Equivalent TAR

The equivalent tissue-air ratio(TAR) method was developed by Sontag and Cunningham.[14] The essence of this method is that a quantity may be determined for phantoms containing non-water equivalent materials by scaling the depth and field size in an appropriate manner. This is accomplished by scaling the tissue-air ratio by the density of the surrounding tissue. For a homogeneous medium this expression is:

$$T(d,r)_\rho = T(d \cdot \rho, r \cdot \rho)$$

For an inhomogeneous medium, a weighted average is found for the distance to the point, d', radius of the scattered components, r', and density of the paths, ρ', as follows:

1. The weighted average distance, d', is determined by the sum of the individual distances of each segment through each inhomogeneity times the density of the inhomogeneity, divided by the number of segments:

$$d' = \frac{d \cdot \sum_{j=1}^{n} \rho_j}{n}$$

[14]M. R. Sontag and J. R. Cunningham, The Equivalent Tissue-Air Ration Method for Making Absorbed Dose Calculations in a Heterogeneous Medium, *Radiology*, Volume 129, Number 2, December 1978, pp. 787-794.

2. The weighted average radius, r', is found by the product of the radius and the weighted average density: $r' = r \cdot \rho'$.

3. The weighted average density is calculated by the sum of the surrounding densities scaled by weighting factors, e.g.:

$$\rho' = \frac{\sum_i \sum_j \sum_k \rho_{ijk} \cdot W_{ijk}}{\sum_i \sum_j \sum_k W_{ijk}}$$

where the weighting factors are complicated expressions of which most can be pre-calculated and stored as a table.

Using these weighted averages the equivalent TAR becomes:

$$T(d', r') = T(d', 0) + S(d', r')$$

where $T(d', 0)$ is the tissue-air ratio for the primary radiation and $S(d', r')$ the scattered radiation tissue-air ratio.

Sontag and Cunningham have also adopted a method to simplify the integration of the computation over the total volume called a 'coalescing procedure'. However, even with this assumption, though the accuracy tested excellent, the method appears to be extremely slow due to the multiple weighted average calculations in the tissue-air ratio computation. The testing discussed in the article covered the accuracy, but no timing. It is interesting to note that the determination of the average weighted distance, d', includes the computation of the effective pathlength as will be discussed in Section 3.6.

3.5.4 Delta Volume Method

After testing several methods as mentioned in the Batho Power-Law section, Wong and Henkelman addressed the calculation of dose from the viewpoint of the CT Scan itself.[15]

[15]J. W. Wong and R. M. Henkelman, A New Approach to CT Pixel-Based Photon Dose Calculations in Heterogeneous Media, *Medical Physics*, Volume 10, Number 2, March/April 1983, pp. 199-208.

This method uses the picture elements generated by the CT Scan as a basis for calculations. They deduced that most methods of clinical dose calculations correct for the primary dose but, since scattered doses are approximated, the merits of any method depend on how well this is accomplished. Wong and Henkelman then developed an augmented first scatter dose by the typical ray-tracing method and approximated the residual multiple-scatter dose. Wong et. al. further developed this method into what is now called the 'Delta Volume Method'.[16] The final expression for the total dose was:

$$Primary(med) + \sum \rho_i \Delta S^*_{1,i}(w) f_{0,i} f_{1,i} + \frac{SAR_m(\overline{\rho} d, \overline{\rho} r)}{SAR_m(d,r)} \left[S_m(w) + \sum \frac{\overline{\rho} - \rho_i}{\overline{\rho}} \Delta H_i \right]$$

where:

- (med) means in the heterogeneous medium.
- (w) means in water(equivalent density medium).
- The term $\sum \Delta S^*_{1,i}(w)$ is the first-scatter dose of the ith element $(1,i)$ adjusted for the change in primary attenuation, $f_{0,i}$, the change in first-scatter attenuation, $f_{1,i}$, and density, ρ_i, and as much of the total scatter dose that behaves like the first-scatter dose, hence the term, 'augmented first scatter dose'.
- The last term is sum of the ith void (ΔH_i) times the relative density difference of the element from its average environment. ($\overline{\rho}$ being the mean density of the heterogeneous medium) This plus the remaining scatter dose is multiplied by the ratio of the scatter-air ratios using the density scaling method.

The specifics of these terms are not as important as the fact that the results of experimental verification show this method to account for several specific situations that other

[16] J. W. Wong, E. D. Slessinger, F. U. Rosenberger, K. Krippner, J. A. Purdy, The Delta Volume Method for 3-Dimensional Photon Dose Calculations, *Proceedings of the Eighth International Conference on the Use of Computers in Radiation Therapy*, IEEE Computer Society Press, July 1984, pp. 26-30 and 78-82.

methods do not. However, this method does require the same ray-tracing time expenditure as all methods calculating the scattered dose of radiation. Wong et. al. implemented this method and determined that about two days are required just for the attenuation calculations. So, while this method appears to be the closest yet to performing the dose calculation with reasonable accuracy for all situations, two days is still too long for radiation treatment planning. The methods used for treatment planning must be faster and therefore cannot afford the time expenditure involved in this type of ray-tracing.

3.5.5 Memorial Hospital Method

A group at the Memorial Hospital in New York continued development of the 'Off-Center Ratio'(OCR) Method as developed by Van de Geijn and discussed briefly in Section 3.4.2.[17] This beam model does not separately calculate primary and scattered radiation but instead utilizes tissue-air ratio(TAR), *tissue-maximum ratio*(TMR), and OCR tables as a data base. The formula they used in their software was:

$$Dose = 100\left(\frac{F}{F-y}\right)^2 \cdot T(d,W_y) \cdot OCR(d,\frac{x}{w_y}) \cdot \frac{T(\tau,w_y)}{T(s,w_y)} \cdot WF$$

where:

- $\left(\frac{F}{F-y}\right)^2$ is an inverse square correction for the difference in the distances: (F), the distance from the beam source to the calibration point(also called the point of reference), and $(F-y)$, the distance along the central axis from the source to the plane containing the point of interest. (y is the length of the segment from the point of calibration to this plane.)

- $T(d,W_y)$ is the TAR or TMR for the beam field size W_y at depth d in the tissue of the point of interest.

[17] *External Beam Program: User's Guide*, Clinical Unit of Memorial Sloan-Kettering Cancer Center, Memorial Hospital for Cancer and Allied Diseases, New York, New York, November 1971.

- $OCR(d, \frac{x}{W_y})$ is the OCR of the point of interest at depth d where the off-center distance, x, is represented as a fraction of the beam field size, W_y.

- $\frac{T(r,W_y)}{T(s,W_y)}$ is the relationship between the TARs or TMRs of field width W_y at depths s and r. s is the slant height of: the depth of the point along the beam central axis, translated to the line from the beam source to the point of interest. r is the effective pathlength and is discussed further in Section 3.6. These last two terms are the manner in which this method corrects for obliquity and inhomogeneity.

- WF is a correction factor for any beam shaping wedges. This topic will not be addressed.

This method retreives all the TAR, TMR, and OCR data from stored tables and computes the percent of the dose delivered to the point of interest based on the dose at the reference point. Because this method does not compute a scattered dose, there is only one ray-tracing computation per point. This makes the calculations much faster since ray-tracing is the slowing factor in radiation dose calculations. Therefore, the Memorial Hospital method of radiation dose calculation is the method of choice for the University of Washington Radiation Oncology Center. There is a certain loss of accuracy over some methods, but at a tremendous gain in speed. As long as the accuracy remains sufficient to allow the planning of radiation therapy, this is obviously the best solution. Since the main slowing factor remains the ray-tracing of the beam and the computation of the effective pathlength, there have been numerous research efforts into the most effective method of performing this calculation. These will be addressed in the following sections.

3.6 Effective Pathlength

In all the different beam models there is one common calculation that is necessary in all of them: finding the distance from the radiation source to the point of calculation or point of interest. This concept is the 'effective pathlength' and is central to all algorithms

which must calculate the radiation dose in a heterogeneous medium. The calculation involves a method of ray tracing and incorporation of the density along that path. Three methods of accomplishing this will be discussed in the following sections and one of these, the Bentley-Milan Method, in the following chapter with the implementation.

The basic formula for determining the effective pathlength is:

$$effpathlen = \int_S^P \rho(r)dl \tag{3.4}$$

where:

- S is the source point.
- P is the calculation point.
- $\rho(r)$ is the heterogeneous medium density.

3.6.1 Bentley-Milan Method

The Bentley-Milan Method[18] considered the solution to the effective pathlength to be a sum-over-segments problem. This was accomplished by finding the intersection of the beam with all the contours over an array of endpoints and sorting the points of each ray-trace by distance from the beam source. As each intersection point is found, the index of the contour that it intersects is assigned to the point. Based on this index, it can be determined whether or not the segment is entering a new contour, exiting an old one, or exiting the current one. The length of each segment is computed and multiplied by the density of that region. In this manner the segments are computed from the source to the calculation point. The formula for the effective pathlength (equation 3.4) expressed as a sum-over-segments is:

[18]R. E. Bentley and J. Milan, The Storage and Manipulation of Radiation Dose Data in a Small Digital Computer, *British Journal of Radiology*, Volume 47, Number 554, February 1974, pp. 115-121.

$$effpathlen = \sum_{j=1}^{j=N_s} l_s(j)\rho_s(j) \tag{3.5}$$

where:

- N_s is the number of segments.
- $l_s(j)$ and $\rho_s(j)$ are the length and density of segment j, respectively.

Since this is the current method implemented in the program in use at the University of Washington Radiation Oncology Department it was used throughout this thesis.

3.6.2 Siddon Method

In the method by Siddon[19] the solution proposed was to determine the effective pathlength not as a sum-over-segments as in the Bentley-Milan case but as a sum-over-regions. For the sum-over-segments solution the points had to be ordered from source to calculation point, and the number of segments, lengths of the segments, and the density of the segment had to be determined. As discussed in the Bentley-Milan Method this required solving the topological problem of which region contained a particular segment for the determination of the density for that segment by use of an index. What Siddon proposed was to avoid this topological problem and solve the pathlength as a sum of the inhomogeneous regions as follows:

1. Compute the intersection of the beam with each contour individually and determine the length of each of these segments.
2. Next find the 'effective density' of the segment as a difference between the contour of the segment and the contour immediately enclosing the first contour. (Using figure 3.6 on page 28 it can be seen that the effective density of the third contour would be: $\rho(3)' = \rho(3) - \rho(2)$.)

[19]Robert L. Siddon, Calulation of the Radiological Depth, *Medical Physics*, Volume 12, Number 1, January/February 1985, pp. 84-87.

3. Sum the values of the segments times their respective densities. So the sum-over-segments solution (equation 3.3) yields for figure 3.6:

$$effpathlen = l_s(1)\rho_s(1) + l_s(2)\rho_s(2) + l_s(3)\rho_s(3) + l_s(4)\rho_s(2) + l_s(5)\rho_s(1)$$

And the sum-over-regions solution:

$$effpathlen = l_s(1+2+3+4+5)\rho_s(1)' + l_s(2+3+4)\rho_s(2)' + l_s(3)\rho_s(3)'$$

where:

- $l_s(numbers)$ is the sum of all those segments.
- $\rho_s(1)' = \rho_s(1)$
- $\rho_s(2)' = \rho_s(2) - \rho_s(1)$
- $\rho_s(3)' = \rho_s(3) - \rho_s(2)$

Since this method did not require solution of the topological problem as discussed Siddon determined through his bench-marking that it was eight times as fast as the sum-over-segments solution. There is a different topological problem in his method, however, of finding which contour is enclosed by which. Siddon proposed that this be handled by the technician in an interactive manner. Since this thesis takes the approach that all calculations should be handled by the computer based on one set of input data, Siddon's interaction is not acceptable as a method of effective pathlength calulation.

3.6.3 Mohan and Antich Method

Mohan and Antich[20] proposed a sort of 'inner contour subtraction' idea. It is similar to the sum-over-regions solution in that the pathlength for each contour is computed individually; however in this case a fractional pathlength is introduced as:

[20]R. Mohan and P. Antich, A Method of Correction for Curvature and Inhomogeneities in Computer Aided Calculation of External Beam Radiation Dose Distributions, *Computer Programs in Biomedicine*, Volume 9, Number 3, May 1979, pp. 247-257.

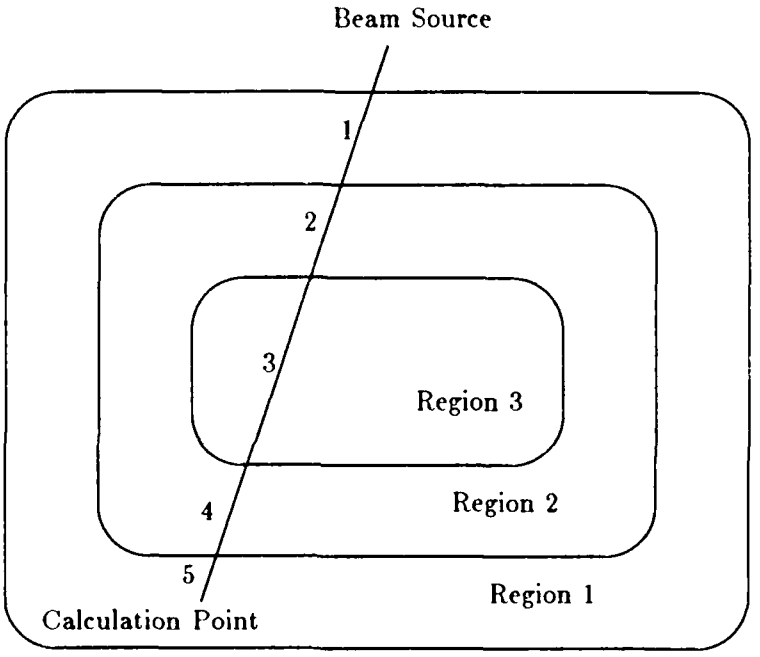


Figure 3.6: Effective Pathlength Diagram

$$effpathlen = \tau L$$

where τ is the fractional pathlength.

Mohan and Antich proposed that the fractional pathlength be computed for the pathlength in the following manner:

1. Find the fractional value of each segment with the total pathlength of the beam.
2. Find the effective density in the same manner as proposed by Siddon.
3. Finally, find fractional pathlength, τ, by summing these segment fractional values times the effective densities. The formula is:

$$\tau = \rho_1 |\sum_i (-1)^i t_r^i| + (\rho_2 - \rho_1) |\sum_j (-1)^j t_r^j| + (\rho_3 - \rho_1) |\sum_k (-1)^k t_r^k| + \cdots$$

where:

- t_r^i, t_r^j, t_r^k represent the fractional distances from the source to each intersection.
- ρ_i is the density of the ith contour.

This method, however, requires more calculations than the Bentley-Milan solution and is slower to find the effective pathlength.

3.7 Future Directions

The current direction of the research in computer usage in radiation therapy is towards the development of more efficient and accurate algorithms for the computation of radiation distribution. There are several examples of this, including the Delta Volume Method discussed previously. While this has led to the development of a more accurate method of determining the radiation dose distributions, the time was extremely lengthy. This research is absolutely necessary and positive for the development of better ways to provide

these distributions to the clinician, but the amount of effort applied to make this type of research into usuable application software is much less than necessary. Cunningham pointed out in his article[21] that

> Following the development of small computers a number of commercial companies have taken over the job of supplying computerized treatment planning equipment to the radiotherapy community. The chief intent of this development is to supply fast, graphically oriented displays of patient cross-sections and dose distributions. Innovation, where it has taken place, has been in packaging rather than in the content and few calculation methods have been introduced since the formative period.

Most of the research has been in the area of more accurate methods of determining the radiation dose distribution, not in developing a manner of presenting a current method fast enough for everyday use. If the efforts of all the research is to benefit the patients then more must be done in the area of providing usable software with the most reasonable (fastest) method of calculation.

---

[21] J. R. Cunningham, Computer Methods for Calculation of Dose Distribution, *Radiation Therapy Planning*, Bleehan, Glatstein, and Haybittle, ed., 1983, page 230.

Chapter 4

Two Dimensional Methods

This chapter will discuss the two methods of doing the two-dimensional effective pathlength correction for inhomogeneities in external beam radiation dose distribution calculations that were implemented and compared. These two methods differ in their ways of representing contours and performing ray tracing for the radiation beam, then both use the Bentley-Milan Method of calculating the effective pathlength as implemented by Dr. Ira J. Kalet of the Radiation Oncology Department of the University of Washington Hospital. They are: the implementation of the Bentley-Milan Method that has been completed by Dr. Kalet, and the implementation of strip trees.

4.1 The Bentley-Milan Method

In this implementation the contours are translated from their current coordinate system to a coordinate system formed with the radiation beam along the x-axis in a method developed by Bentley and Milan [1] (see Figure 4.1). This requires a translation and rotation of the axes and all coordinates are translated by the standard rotation formulas:

$$xnew = (xold - newxorigin) \times \cos\theta + (yold - newyorigin) \times \sin\theta$$

$$ynew = -(xold - newxorigin) \times \sin\theta + (yold - newyorigin) \times \cos\theta$$

[1] R. E. Bentley and J. Milan, The Storage and Manipulation of Radiation Dose Data in a Small Digital Computer, *British Journal of Radiology*, Volume 47, Number 554, February 1974, pp. 115-121.

Once all coordinates have been translated, they are projected onto the y-axis. Intersections are then determined to occur when the projected y-value of the current point crosses the projected y-value of the calculation point and the x-value of the current point is less than the x-value of the calculation point.

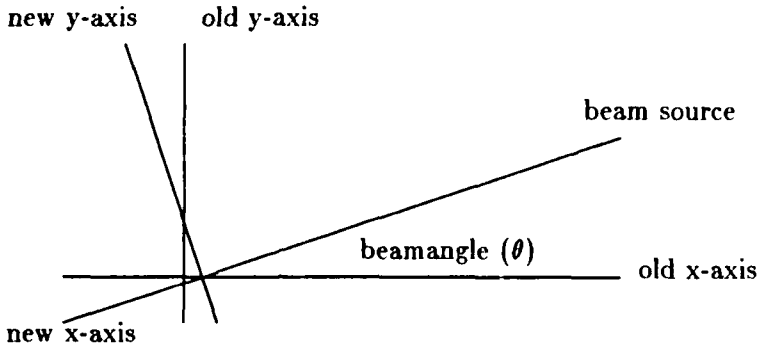


Figure 4.1: Rotation of axes in Bentley-Milan Method

Once the intersection points have been determined, the records of these intersection points are sorted into decreasing value from the beam source using the following Wirth algorithm where in this case the testedfield is the x-coordinate:[2]

Algorithm 4.1 : SORT ROUTINE

```
for i := 2 to the number of intersections do
    temp := intersectionpoint[i]
    intersectionpoint[0] := temp
```

[2]N. Wirth, *Algorithms + Data Structures = Programs*, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1976.

```
j := i - 1
while temp.testedfield > interpt[j].testedfield do
    intersectionpoint[j+1] := intersectionpoint[j]
    j := j - 1;
intersectionpoint[j+1] := temp
```

The final step is to determine the effective pathlength. This algorithm takes the intersection points, determines the distance between each point, the density of the matter between them, and then computes the effective pathlength which is merely the sum of the distances times the density. It is an implementation of the sum-over-segments solution developed by Dr. Ira Kalet of the University of Washington Radiation Oncology Department. The arrays **density** and **intlist** are last-in, first-out stacks which keep track of the nesting of the contours.

Algorithm 4.2 : SUM-OVER-SEGMENTS

```
assign current density and index to level 0
for 1 to number of intersections do
   assign current intersection point to temp values
   next := number + 1
   if next < number of intersections then
      assign next intersection point to temp values
      length := current point to next point distance
      density := density[level]
      if the points are of the same index{contour} then
         {going to next level down,i.e.,leaving contour}
         decrement level
      else {going up}
         {save values for when passing through this contour}
         increment level
```

```
            density[level] := density of current point
            intlist[level] := index of current point
        else {segment goes to calculation point}
            length := current to calculation point distance
            density := density[level]
        effectivepathlength := effectivepathlength + density * length
end
```

4.2 Strip trees

The second method investigated was the method of representing a curve by strip trees, as defined by Duda and Hart [3] and refined in an article by Dana Ballard, to attempt to obtain the logarithmic order time provided by a tree data structure. [4] The strip tree is then used to find all the intersections of the beam with the contour through the intersection of the beam with the strip tree extents.

4.2.1 Definition

A strip S is defined by Dana Ballard [5] to be the six-tuple $(\mathbf{x}_b, \mathbf{x}_e, w_l, w_r)$ where $\mathbf{x}_b = (x_b, y_b)$ denotes the beginning of the strip as determined by a directed segment and $\mathbf{x}_e = (x_e, y_e)$ the endpoint. w_l and w_r denote the maximum distances to the side of the strip as determined by the points of the curve between the endpoints (See Figure 4.2). In addition to these six parameters the slope of the segment from $\mathbf{x}_b$ to $\mathbf{x}_e$, mv, was added because of its prevalence in the intersection routines discussed later.

A strip tree is then defined as nodes that consist of (Strip, LeftSon, RightSon) where LeftSon and RightSon are strip trees or null.

---

[3] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley-Interscience, New York, 1973.

[4] Dana Ballard, Strip Trees: A Hierarchical Representation for Curves, *Communications of the ACM*, Volume 24, Number 5, May 1981, pp. 310-321.

[5] Dana Ballard, page 312.

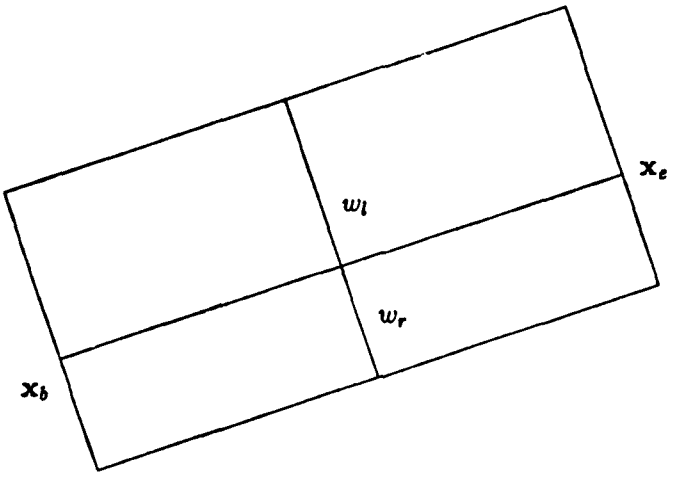


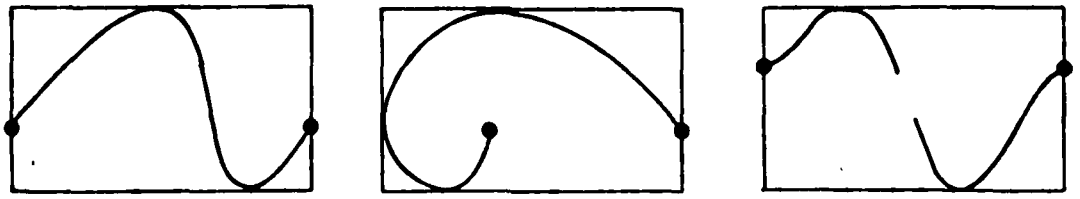
Figure 4.2: Strip Tree Extent

An important concept in strip trees is to determine whether or not a strip is regular. A strip is defined as regular if it's underlying curve:

1. is connected,

2. has its endpoints touching the ends of the strip.

Some examples of regular and non-regular curves are given in Figure 4.3.



Regular Non-Regular

Figure 4.3: Examples of Regular and Non-Regular Curves

4.2.2 Implementation

The contour is passed into the routines as a list of points. The first step is to find the furthest two points apart to insure that the strips remain regular and make these two points the endpoints of the first two strips. One strip is directed from point A to point B and the second the other direction, with all points in between assigned to the two strips according to their position in the ordered list describing the curve.

With the strip record defined as follows:

```
strip =
    record
        xb, yb, xe, ye, wl, wr, mv : real;
        LSon, RSon : pointers to strips;
    end;
```

the strips are recursively determined by the following method:

1. Using the endpoints determine the slope of the directed segment.

2. Determine the perpendicular distance of every point between the endpoints from the directed segment. This is accomplished by a polar transformation of the axes to make the strip segment the x-axis. The angle between the strip segment and the line from $\mathbf{x}_b$ to each point yields the direction to the left or right of the segment and the new y-coordinate of each point the distance w_l or w_r.

3. Choose the larger value of w_l and w_r as the next point to divide the strip into the two sons, with the line from $\mathbf{x}_b$ to this point as leftson and the line from this point to $\mathbf{x}_e$ the rightson.

4. Repeat recursively until there are no more points between the segment endpoints or both w_l and w_r are zero.

4.2.3 Intersection Routine

The intersection of the radiation beam with the strip extent was accomplished by an intersection of lines solution. Each point at the corner of the strip extent is determined as needed and this line segment and the beam ray intersected. This is accomplished in the following manner:

1. Determine the endpoints of the corners of the strip extent. Two different methods of this were tested.

   (a) The first method was by using trigonometric rules. The slope of a line is $\frac{\Delta y}{\Delta x}$ and the tangent of θ is equal to the slope when defined as shown in Figure 4.4. The slope of the line from the endpoint to the cornerpoint is $-\frac{1}{slope of segment}$ since it is perpendicular to the segment by definition.

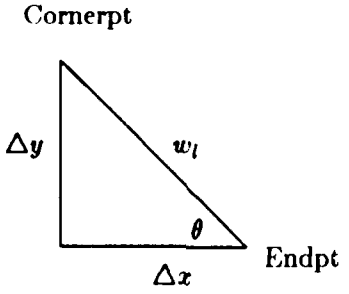


Figure 4.4: Trigonometric Method

So the following equation can be used to find the cornerpoint:

$$atan(m) = \theta$$

$$x_{corner} = x_{endpoint} + width \times \cos\theta$$

$$y_{corner} = y_{endpoint} + width \times \sin\theta$$

where width is w_l or w_r depending on the side of the strip, corner means the corner of the extent to be determined, endpoint is the end of the segment being used to determine the corner, and m is the slope of the line from the endpoint to the cornerpoint. Special cases had to be developed for horizontal and vertical lines because the slope values are 0 and undefined respectively.

(b) The second method was done by simultaneous solution of the distance between two points:

$$(width)^2 = (x_{corner} - x_{endpoint})^2 + (y_{corner} - y_{endpoint})^2$$

and the two-point slope equation

$$y_{corner} - y_{endpoint} = m(x_{corner} - x_{endpoint})$$

to yield

$$x_{corner} = x_{endpoint} + \frac{width}{\sqrt{1 + (slope)^2}}$$

Once the x value has been determined the y value is found by solution of the two point slope form of the equation. Special cases again had to be developed for horizontal and vertical lines.

(c) When both methods were implemented and timing of the methods was accomplished the results showed a 10% decrease using the second method. The trigonometric method was too slow because of the infinite series solution of the trigonometric functions. The results are shown in Table 4.1. The times were determined by using the 'time' variable in Unix on a VAX 11-780, where the program was executed on the same contour of points with the only difference being the method of endpoint determination.

2. After determining two corner points determine the constants in the equation $Ax + By + C = 0$ for this line. This is done by solving this equation based on the axes

Number of Contours	Trig Method Time	Math Method Time
1	56.5	45.7
2	81.3	76.8
3	94.0	85.9
4	109.7	100.0
5	130.5	112

All times are in seconds

Table 4.1: Comparison of Endpoint Determination Methods

intersections to yield the following:

$$A = -(y_{pt2} - y_{pt1})$$

$$B = x_{pt2} - x_{pt1}$$

$$C = -(A \times x_{pt1} + B \times y_{pt1})$$

Special cases involving vertical and horizontal lines solve easily since the equation for a line reduces to the x or y term equal to a constant.

3. Determine the constants for the beam line in the same manner.

4. Once the constants have been determined the intersection of two lines is found by the use of the following equations. These were derived from the matrix solution of the equations of the two lines at the intersection point.

$$xint = \frac{B_1 \times C_2 - B_2 \times C_1}{A_1 \times B_2 - A_2 \times B_1}$$

$$yint = \frac{C_1 \times A_2 - C_2 \times A_1}{A_1 \times B_2 - A_2 \times B_1}$$

5. The intersection coordinates must now be tested to determine if the intersection point is between the endpoints of the strip extent segment and the beam segment. This is accomplished by the following algorithm:

$$min[min(x_1, x_2), min(x_1', x_2')] \leq xint \leq max[max(x_1, x_2), max(x_1', x_2')]$$

$$min[min(y_1, y_2), min(y_1', y_2')] \leq yint \leq max[max(y_1, y_2), max(y_1', y_2')]$$

where the points are: Beam Source = (x_1, y_1)

Beam Endpoint = (x_2, y_2)

Strip Endpoint1 = (x_1', y_1')

Strip Endpoint2 = (x_2', y_2')

6. If this item is determined to be a good intersection point by the above algorithm and the strip is a leaf node then this point is inserted into the intersection point list. If the strip is not a leaf node the LSon and then RSon are tested. If there is no intersection the remaining sides of the strip extent are tested in order. If the beam does not intersect the strip extent it does not intersect the curve.

7. Once all the intersections points have been determined they are sorted by algorithm 4.1 on page 32. The effective pathlength is then computed using algorithm 4.2 on page 33 as implemented by Dr. Kalet.

4.3 Comparison of Methods

In determining the most effective method for use by clinicians or technicians the most important aspect is speed. In routine use by these personnel immediate response and interaction in treatment planning is the most effective and efficient use of their time and the planning sequence.

When running these tests the calculation was conducted over a 40×40 (1600) array of points. The beam source was maintained in one location and the beam endpoint (or calculation point) was varied over the array with the effective pathlength calculated for every point.

Table 4.2 summarizes the results conducted on a Vax 11-780 machine with the Unix operating system. The 'Total Pts' listed in the table represents the total number of points defining the contours.

Number of Pts/Contour	Bentley-Milan Program Time(sec)	Strip Tree Program Time(sec)
60	8.3	71.9
90	9.8	82.3
120	10.4	96.5
150	12.0	98.2
180	13.5	103.2
210	15.1	104.1
240	16.3	104.3
270	18.0	104.7
300	19.6	104.9

Table 4.2: Comparison of 2D Methods

The table figures reveal that for this application the Bentley-Milan Method is faster than the Strip Tree Method. As the number of points defining the contour is increased the Bentley-Milan Method shows a decidely linear increase in time. The Strip Tree Method shows a curvelinear increase as the number of points increases that approaches 105 seconds. Extrapolation of the graphing of these two functions shows that if the strip tree can be assumed to approach 105 seconds then the two methods will intersect at a point where there are 2100 points/contour, and if the last four points of the strip tree graph were taken to be linear the intersection would be at 2850 points/contour. The actual intersection will fall between these two values. However, an analysis of actual data from typical radiation treatment cases revealed that the number of points in a contour fell between 70 and 150 points. An analysis of the Strip Tree Method showed a major problem to be that the strip extent covers an area of (strip segment length $\times$ $(w_l + w_r)$) and that intersections can be found with the strip extent that do not intersect the curve and are not discovered until the strip tree is evaluated further down. This means that several branches of the strip tree could be taken that later are found to be nil. The second and most major problem that caused the strip trees version to be slower is that there are numerous expensive calculations to be done by the intersection routine. Two examples of these expensive calculations are the intersection of lines solution with, possibly, all four sides of the strip tree extent and

the algorithm to determine if the point is within the line segment. So the faster of the two methods for this application is definitely the Bentley-Milan Version.

Chapter 5

Three-Dimensional Implementation

In this chapter the extension of the calculation of the effective pathlength into three-dimensions will be covered.

5.1 Coordinate Systems

The layout of the axes for this discussion will be ordered with the x-axis to the patient's left when the patient is prone, the y-axis directed into the patient and the z-axis directed towards the patient's head. This places the $x-y$ axes in the plane of the CT Scan. This is slightly different than the axes definitions used by Siddon[1] but defined for the same reason: to put two of the axes in the plane of the CT Scan and one perpendicular towards the patients' head.

5.2 Algorithm

As discussed by Siddon and Kijewski[2], insuring that the axes are defined as above can reduce the problem to a two-dimensional one very easily. The algorithm used here

[1]Robert L. Siddon and P. K. Kijewski, 3-D Ray Depth Calculation for Radiotherapy Applications, In *Proceedings of the Eighth International Conference on the Use of Computers in Radiation Therapy*, July 1984, pp. 201-204.

[2]Siddon and Kijewski, pp. 201-204.

is similar to the one they describe but the main source for this algorithm came from Kajiya.[3] In this article the object is defined in two-dimensions and extended into three-dimensions along a specified height, h. This is called a prism by Kajiya (See Figure 5.1). The three-dimensional ray is then intersected with the top and bottom of the prism, the capplane and baseplane, respectively. The ray is projected into two-dimensions on the baseplane and intersected with the contours (Kajiya's object). These intersections are then to be projected back into three dimensions and sorted, with the capplane and baseplane intersections, by distance from the beam source. The intersections are discerned to be within the prism by a complicated algorithm to be fully covered later.

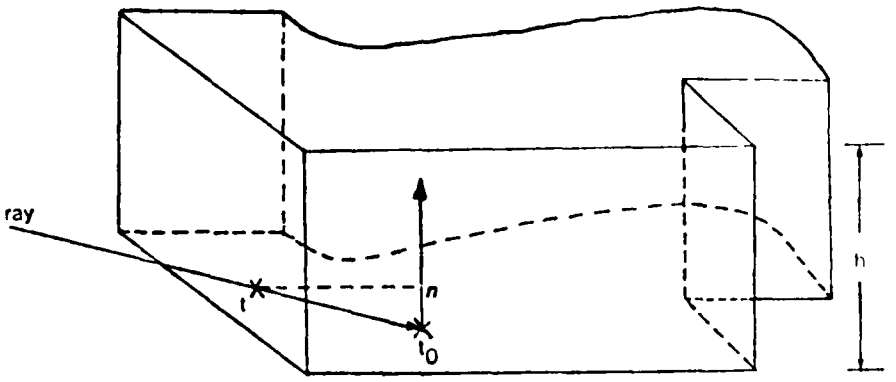


Figure 5.1: A prism, with height h, as defined by Kajiya, with a ray shown striking the side of the prism at point t and the baseplane, with *normal* n, at point t_0.

[3]James T. Kajiya, New Techniques for Ray Tracing Procedurally Defined Objects, *ACM Transactions on Graphics*, Volume 2, Number 3, July 1983, pp. 161-181.

5.2.1 Determination of the Plane Equation

In this case the contours are received in the same manner as the two-dimensional case, from CT Scans. The height is determined by half the distance to the CT Scans on either side. Since CT Scans may be taken very close together (as close 1 mm) this approximation of the contour in 3d for such a small distance renders a very reasonable approximation of the body. This will allow the computation of the effective pathlength to be done across each of the CT Scans and summed for the total of the beam length.

Once the height is given the equations of the baseplane and capplane are determined. This is accomplished by taking three points of the baseplane and calculating the vectors from one point to the other two. These vectors are placed in normalized parametric form, the *direction cosines* found and the constants in the planar equation $Ax+By+Cz+D=0$ are determined from the direction cosines. This method can be shortened because of the use of common values throughout the method and is summarized below.

Given points: $(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3)$ determine $\overrightarrow{V_1}$ and $\overrightarrow{V_2}$ as follows:

$$\overrightarrow{V_1} = (x_2 - x_1)\,\overrightarrow{i} + (y_2 - y_1)\,\overrightarrow{j} + (z_2 - z_1)\,\overrightarrow{k}$$

$$\overrightarrow{V_2} = (x_3 - x_1)\,\overrightarrow{i} + (y_3 - y_1)\,\overrightarrow{j} + (z_3 - z_1)\,\overrightarrow{k}$$

These vectors are then normalized by their length, d, and the direction cosines for $\overrightarrow{V_1}$ determined by:

$$\cos\alpha = (x_2 - x_1) \div d_{\overrightarrow{V_1}}$$

$$\cos\beta = (y_2 - y_1) \div d_{\overrightarrow{V_1}}$$

$$\cos\gamma = (z_2 - z_1) \div d_{\overrightarrow{V_1}}$$

The direction cosines for $\overrightarrow{V_2}$ (α^*, β^*, and γ^*) are determined similarly.

The constants are found by the cross product of these cosines:

$$A = \cos\beta \times \cos\gamma^* - \cos\beta^* \times \cos\gamma$$

$$B = \cos\gamma \times \cos\alpha^* - \cos\gamma^* \times \cos\alpha$$

$$C = \cos\alpha \times \cos\beta^* - \cos\alpha^* \times \cos\beta$$

$$D = -(Ax_1 + By_1 + Cz_1)$$

Once this is accomplished the equation for the capplane is determined in the same manner except the coordinates are transformed to be the specified height above the baseplane. In this special case where the baseplane lies in the x-y plane the capplane coordinates are exactly height h above in the z direction. In the general case the coordinates would be found by going h in the direction of the plane unit normal.

5.2.2 The Intersections

Once the baseplane and capplane equations are found the intersection of the beam with these planes is needed. This is computed by the intersection of a line and a plane. First the direction cosines as defined above are determined for the beam line and the following algorithm evaluated.

If

$$A_{plane} \times \cos\alpha_{line} + B_{plane} \times \cos\beta_{line} + C_{plane} \times \cos\gamma_{line} = 0$$

then the line is parallel

If the above equation is true and

$$A_{plane} \times x_{line} + B_{plane} \times y_{line} + C_{plane} \times z_{line} = 0$$

then the line lies in the plane

Else

$$t = \frac{A_{plane} \times x_{line} + B_{plane} \times y_{line} + C_{plane} \times z_{line} + D_{plane}}{A_{plane} \times \cos\alpha_{line} + B_{plane} \times \cos\beta_{line} + C_{plane} \times \cos\gamma_{line}}$$

And the intersection points for the two planes are:

$$x_{int} = x_1 - t \times \cos\alpha$$

$$y_{int} = y_1 - t \times \cos\beta$$

$$z_{int} = z_1 - t \times \cos\gamma$$

5.2.3 Two Dimensional Intersections

The intersection of the contours by the beam in the baseplane is then accomplished by translating the beam points into the baseplane and performing the two dimensional intersection by one of the methods discussed in Chapter 4. Note that the points used are the beam source and beam calculation points, not the baseplane and capplane intersection points.

5.2.4 Sort Routine

Once all the intersections have been found they are then translated back into three dimensions along the beam line. This is done by solving the *scalar equations* for z to yield:

$$z_{int} = \frac{x_{int} - x_{beamsource}}{x_{calcpt} - x_{beamsource}} \times (z_{calcpt} - z_{beamsource}) + z_{beamsource}$$

If the difference in the x values is zero then the solution of the scalar equations with the y values is used in the above equation. Once all the z values have been found the distance from the beam source is calculated and the points are sorted by increasing distance from the beam source. This is done by algorithm 4.1, where testedfield is the three-dimensional distance.

5.2.5 Determination of the Prism Intersections

The evaluation of the intersection points to determine if they are entering the prism is the next step. From the article by Kajiya[4] the simplistic algorithm is:

1. If the intersection point is a capplane strike point then

   (a) If inside the contour then intersection point is good.

[4]James T. Kajiya, New Techniques for Ray Tracing Procedurally Defined Objects, *ACM Transactions on Graphics*, Volume 2, Number 3, July 1983, page 171.

(b) Otherwise it is not in the prism and continue to next point.

2. If the intersection point is a baseplane strike point then

   (a) If inside the contour then intersection point is good.

   (b) Otherwise it is not in the prism and continue to next point.

3. If a contour intersection then

   (a) If the intersection point z coordinate is between the capplane and baseplane then the intersection point is good.

   (b) Otherwise, it is not in the prism and continue to the next point.

The implementation was very similar except for some minor variations, the most obvious of which is that there are multiple contours to be intersected in the radiation therapy application. The algorithm developed will be summarized here. Note that if the contour has not been crossed then there can be *no* valid intersection points. The algorithm is:

1. While not contour intersection get next intersection point {Find the first contour intersection}

2. If the z-coordinate of the contour intersection is between the capplane and baseplane then inside prism {entered through side}.

3. Otherwise get next intersection point. {Go to next point and test all points now that a contour has been entered}

4. While not inside prism do {Since the first contour intersection was not valid check all intersections until inside prism }

   (a) If intersection point is baseplane or capplane then inside prism {entered through the bottom or top}

   (b) Otherwise:

i. If contour intersection z-coordinate is between the capplane and baseplane then inside prism (entered through side)

ii. Otherwise, get next intersection point. {If contour intersection and not within the prism z values then try next point}

5. While (contour intersections) and z-coordinate is between baseplane and capplane add point to valid intersection point list, get next point. {As long as there are contour intersections they are still within the prism.}

6. {Now have reached the last prism intersection by exiting through the side or reaching a baseplane or capplane intersection} If not (contour intersection) and (still inside any contour) then add baseplane or capplane to the valid intersection list. {If still inside any contour then exiting through the bottom or top of the prism.}

If the contour is entered and it is not a valid intersection then the level is incremented and the index and density of the contour are saved. This is necessary because in the Effective Pathlength Procedure if the prism entry is through the capplane or baseplane these values are needed for the effective pathlength computation.

5.2.6 Effective Pathlength

Algorithm 4.2 used in this version is exactly the same except that the level of entry may be changed by the fact that a baseplane or capplane intersection could enter the prism already inside a contour. This was handled with the algorithm just covered in Section 5.2.5. If a contour was entered and the intersection point was outside the z-value necessary to make it a prism intersection than the level was incremented and the density and index maintained in an array. It was also necessary to keep an array of boolean values for determining whether a contour had been entered already and was being exited or reentered. In this case the level and density array were again altered.

5.3 Utilization

This algorithm is then applied to each of the CT Scans in order (See Figure 5-2) and the effective pathlength computed for a three-dimensional array of points in the region of interest to determine a wire-frame diagram of specific radiation levels. The remainder of the computations to determine the radiation dose are handled by the software for whichever beam model is being used.

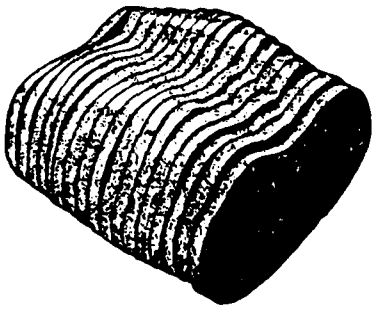


Figure 5.2: Multiple CT Scans Represented as Slabs

5.4 Comparison of Methods

Testing was accomplished on the three-dimensional method by incorporating the two two-dimensional methods. The data is represented in Table 5.1 and shows that the Bentley-Milan Method is clearly the faster of the two methods. The reasons are the same as for the two-dimensional versions. The three-dimensional algorithm is the same in both cases and the two-dimensional versions remain unchanged. In the testing it was also revealed that as the number of contours increased the Bentley-Milan Method also

Number of Contours	Total Contour Pts	Bentley-Milan Time	Strip Tree Time
1	60	5.8	40.4
2	98	7.5	69.2
3	110	8.2	77.5
4	130	8.9	97.3
5	152	10.3	112.6
6	163	10.8	116.8
7	183	11.6	122.7

All times are in seconds

Table 5.1: Comparison of 3D Methods

had a 50% savings in space. The table was compiled by computing the intersections over a three-dimensional area of 80 × 80 × 6 (x,y,z) incremented so that the number of points checked was 9 × 9 × 3 (243 points). Note that this version was run on data that serves as an example of the actual contours experienced in typical cases of radiation treatment planning.

5.5 Conclusions

In this application the logarithmic time savings afforded by strip trees did not improve on the linear method because of the small number of points in the contour. As determined in Chapter 4, the contours would have to be extremely large to make the strip tree method useful.

A couple of alternatives to the prism algorithm, although somewhat similar, are: the method developed by Wong et. al. already covered as the Delta Volume Method, and the method covered by Houlard and Dutreix. As described in Chapter 3, the Delta Volume Method is a method of ray-tracing through CT picture element densities and was much too time consuming. The second method involves the determination of optimal-triangular-tiling reconstruction from planar contours obtained from CT Scans, as defined by Keppel and Fuchs in separate papers. Then the ray-tracing method through these tiles was

proposed by Houlard and Dutreix[5]. However, the prism method is more easily applied to the already parallel CT Scans, is much more elegant, and is an extremely orderly implementation.

This prism implementation could possibly be used in other applications where ray-tracing is necessary. The graphics applications of surface rendering and shading in three-dimensions may be worthwhile where the object could be easily defined by prisms. As Kajiya mentioned in his article, many objects can be defined as collections of prisms; among them, block letters, machine parts, and simple urban architecture models. The prism concept would be useful in cases where surfaces have essential symmetries and the problem can be reduced from three-dimensions to two-dimensions for the ray-tracing to be time cost-effective.

[5]J. P. Houlard and A Dutreix, 3D Display of Radiotherapy Treatment Plans, *Proceedings of the Eighth International Conference on the Use of Computers in Radiation Therapy*, IEEE Computer Society Press, July 1984, pg. 219

Bibliography

[1] *External Beam Program: User's Guide.* Clinical Unit of Memorial Sloan-Kettering Cancer Center, Memorial Hospital for Cancer and Allied Diseases, New York, New York, November 1971.

[2] Dana H. Ballard. Strip Trees: A Hierarchical Representation for Curves. *Communications of the ACM*, 24(5):310–321, May 1981.

[3] R. E. Bentley and J. Milan. The Storage and Manipulation of Radiation Dose Data in a Small Digital Computer. *British Journal of Radiology*, 47(554):115–121, February 1974.

[4] John R. Cunningham. Computer Methods for Calculation of Dose Distribution. In Norman N. Bleehan, Eli Glatstein, and John L. Haybittle, editors, *Radiation Therapy Planning*, pages 217–263, Marcel Dekker, Inc, 1983.

[5] John R. Cunningham and J. Milan. Chapter 6. In R. W. Stacey and B. D. Waxman, editors, *Computers in Biomedical Research*, Academic Press, 1969.

[6] Michael J. Day and R. M. Harrison. Cross-Sectional Information and Treatment Simulation. In Norman N. Bleehan, Eli Glatstein, and John L. Haybittle, editors, *Radiation Therapy Planning*, pages 89–95, Marcel Dekker, Inc, 1983.

[7] J. Van de Geijn. Computational Methods in Beam Therapy Planning. *Computer Programs in Biomedicine*, 2(3):153–168, 1972.

[8] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley-Interscience, New York, 1973.

[9] H. Fuchs, Z. M. Kedem, and S. P. Uselton. Optimal Surface Reconstruction from Planar Contours. *Communications of the ACM*, 20(10):693–702, October 1977.

[10] Wolfgang K. Goloi. *Interactive Computer Graphics.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1978.

[11] J. P. Houlard and A. Dutreix. 3d Display of Radiotherapy Treatment Plans. In *Proceedings of the Eighth International Conference on the Use of Computers in Radiation Therapy*, page 219, IEEE Computer Society Press, July 1984.

[12] James T. Kajiya. New Procedures for Ray Tracing Procedurally Defined Objects. *ACM Transactions on Graphics*, 2(3):161–181, July 1983.

[13] E. Keppel. *IBM Journal of Research and Development*, 19:2, 1975.

[14] Radhe Mohan and Pietro P. Antich. A Method of Correction for Curvature and Inhomogeneities in Computer Aided Calculation of External Beam Radiation Dose Distributions. *Computer Programs in Biomedicine*, 9(3):247–257, May 1979.

[15] P. G. Orchard. Decrement Lines: A New Presentation of Data in Cobalt-60 Beam Dosimetry. *British Journal of Radiology*, 37(442):756–763, October 1964.

[16] J. Richter and D. Schirrmeister. Die Berucksichtigung von Gewebeinhomogenitaten bei der Ermittlung von Dosisverteilungen mit digitalen Rechenautomaten (Translation-Handling Tissue Inhomogeneities in the Calculation of Dose Distribution on Computers). *Strahlentherapie*, 127(4):550–559, August 1965.

[17] J. Richter and D. Schirrmeister. Ein Verfahren zur Berechnung der Dosisverteilungen mit digitalen Rechenautomaten (Translation- A Procedure for Calculating Dose Distribution with Computers). *Strahlentherapie*, 123(1):45–58, January 1964.

[18] R. L. Siddon and P. K. Kijewski. 3-D Ray Depth Calculation for Radiotherapy Applications. In *Proceedings of the Eighth International Conference on the Use of Computers in Radiation Therapy*, pages 201–204, IEEE Computer Society Press, July 1984.

[19] Robert L. Siddon. Calculation of the Radiological Depth. *Medical Physics*, 12(1):84–87, January/February 1985.

[20] W. Siler and J. S. Laughlin. A Computer Method for Radiation Treatment Planning. *Communications of the ACM*, 5(7), July 1962.

[21] M. R. Sontag, J. J. Battista, M. J. Bronskill, and J. R. Cunningham. Implications of Computed Tomography for Inhomogeneity Corrections in Photon Beam Dose Calculations. *Radiology*, 124(1):143–149, July 1977.

[22] M. R. Sontag and J. R. Cunningham. Corrections to Absorbed Dose Calculations for Tissue Inhomogeneities. *Medical Physics*, 4(5):431–436, September/October 1977.

[23] M. R. Sontag and J. R. Cunningham. The Equivalent Tissue-Air Ratio Method for Making Absorbed Dose Calculations in a Heterogeneous Medium. *Radiology*, 129(2):787–794, December 1978.

[24] T. D. Sterling, H. Perry, and L. Katz. Automation for Radiation Treatment Planning, IV, Derivation of a Mathematical Expression for the Per Cent Depth Dose Surface of a Cobalt-60 Beam and Visualization of Multiple Field Dose Distributions. *British Journal of Radiology*, 37(439):544–560, July 1964.

[25] K. C. Tsien. Application of Automatic Computing Machinery to Radiation Dose Calculations. *British Journal of Radiology*, 28(332):432–439, August 1955.

[26] Niklaus Wirth. *Algorithms + Data Structures = Programs*. Prentice-Hall, Inc, Englewood Cliffs, N.J., 1976.

[27] J. W. Wong and R. M. Henkelman. A New Approach to CT Pixel-Based Photon Dose Calculations in Heterogeneous Media. *Medical Physics*, 10(2):199–208, March/April 1983.

[28] J. W. Wong and R. M. Henkelman. Reconsideration of the Power-Law(Batho) Equation for Inhomogeneity Correction. *Medical Physics*, 9(4):521–530, July/August 1982.

[29] J. W. Wong, E. D. Slessinger, F. U. Rosenberger, K. Krippner, and J. A. Purdy. The Delta-Volume Method for 3-Dimensional Photon Dose Calculations. In *Proceedings of the Eighth International Conference on the Use of Computers in Radiation Therapy*, pages 26–30 and 78–82, IEEE Computer Society Press, July 1984.

Appendix A

2D Bentley-Milan Version

The following pseudocode was written to supply specifics in some areas and plain language in others to allow the code to be condensed and included in this thesis while providing the basics of the program.

```
program contour(output, contourdata, beamdata);

begin
    read all contour and beam input data;
    procedure setupcontours(contourdata,beamdata,reference  point)
    {Translates and rotates coordinates as described in Section 4.1 and also
     computes the projected y-value}
    begin {setup procedure}
    for all contours do
        begin
            {These values determined as described in Section 4.1 and
             placed into the input contours}
            xtemp := xlist[j] - xref;  ytemp := ylist[j] - yref;
            xtrans[j] := xtemp * cos(beamangle) + ytemp * sin(beamangle);
            ytrans[j] := -xtemp * sin(beamangle) + ytemp * cos(beamangle);
            yproj[j] := ytrans[j]*(sad/(sad-xtrans[j]));
        end;
    end; {procedure}
    procedure pathlength(beamdata, input contours, reference point)
    {This procedure compares the projected y-values and if there is a crossing
     over the calculation point projected y-value, checks the x-value, then
     places the point into the intersection point array. The next step is to
     sort the points using Algorithm 4.1 from Section 4.1. The final step is
     finding the effective pathlength using Algorithm 4.2 from Section 4.1. }
    begin
    number of intersections := 0;
```

```
for all contours do
begin
   {set starting parity based on above or below the calculation point}
   if yproj[pointi] > yproj[calculation point] then
       parity := up else parity := down;
   {now go around the contour}
   for all points do
   begin
      crossing := false;
      case parity of
      up: if yproj[next point] <= yproj[calc point] then {crossed calc pt}
          begin crossing := true; parity := down end;
      down: if yproj[next point] > yproj[calc point] then {crossed calc pt}
          begin crossing := true; parity := up end
      end; {case of parity}
      if crossing then {check x-value to see if it is a real crossing}
      begin
         if xvalue[nextpoint] <> xvalue[currentpoint] then
         begin
            {Determine portion of segment between points}
            fraction := (ytrans[i+1] - ytrans[i]) /
                        (xtrans[i+1] - xtrans[i]);
            xtemp := (yp - ytrans[i] + fraction * xtrans[i]) /
                        (yp/sad + fraction);
         end
         else xtemp := xvalue[currentpoint];
         if xtemp > (x of calc point) then {crossing is real}
         begin {add new crossing record to crossing list}
             increment number of intersections;
             {Add x,y,density and contour number to intersection array}
         end;
      end {if crossing is real}
   end {for all points}
end {for all contours}
{Algorithm 4.1 as described in Section 4.1}
for i := 2 to number of intersections do
begin
    temp := intersectionpoint[i];
    intersectionpoint[0] := temp;
    j := i - 1;
    while temp.x > intersectionpoint[j].x do
    begin
        intersectionpoint[j+1] := intersectionpoint[j];
        j := j - 1;
    end;
    intersectionpoint[j+1] := temp;
end;
```

```
{Now compute the pathlength using Algorithm 4.2 from Section 4.1}
    level := 1;
    if number of intersections > 0 then
    begin
       dlist[level] := assocdensity;
       nlist[level] := index;
    end;
    dlist[0] := 0.0;
    nlist[0] := 0;
    effpathlength := 0.0;
    for internumber := 1 to number of intersections do
    begin {go through crossing list}
       assign current intersectionpoint to temp1
       next := internumber + 1;
       if next <= number of intersections then
       begin {there is a next crossing}
          assign next intersectionpoint to temp2
          length := distance between temp1 and temp2;
          density := density[level];
          { now must determine if we are entering a new region,
             or leaving the current region for a "lower level" }
          if indexoftemp1 = indexoftemp2 then
             {going to next level down, i.e., leaving contour}
             decrement level
          else {going up a level}
             {must save density and intlist values on stack}
             density[level] := density of temp1;
             intlist[level] := index of temp1;
       end {next <= numcrosses, i.e. there was a next crossing}
       else {this segment goes from the crossing to the computation point}
       begin
          length := distance from current point to calculation point;
          density := dlist[level];
       end;
       {now we can add in the effective length of this segment}
       effpathlength := effpathlength + density of path*length;
    end; {end of list-done all crossings}
    end; {procedure}
end. {program}
```

Appendix B

2D Version of Strip Tree Method

The following pseudocode was written to supply specifics in some areas and plain language in others to allow the code to be condensed and included in this thesis while providing the basics of the program.

```
program 2d intersection(output, contourdata, beamdata);

function widpt(strip, list of points):furthest point from strip;

begin
   {convert strip segment to polar coordinates}
   ppt1.x := endpt.x - beginpt.x;
   if ppt1.x = 0 then {vertical line}
      {cannot figure slope so set to zero as convention}
      angle1 := pi/2;
   else
      ppt1.y := endpt.y - beginpt.y;
      slope := ppt1.y / ppt1.x;
      delta1 := ppt1.y / length of strip segment;
      angle1 := arcsin(delta1);
   {Now loop through all the points between the segment begin and endpoints
    and determine the left(wl) and right(wr) distances. Then choose the
    largest and make it the next point at which to divide the strip.     }
   while not at segment endpt do
```

```
begin
   temppoint := next point{starting from segment beginpt};
   ppt2.x := temppoint.x - beginpt.x;
   ppt2.y := temppoint.y - beginpt.y;
   {set angle of segment using 360 degree coordinates}
   if ppt1.x > 0 then
      if ppty1.y >= 0 then
         angle2 := angle1;
      else
         angle2 := 2pi - abs(angle1);
   else
      if ppt1.y >= 0 then
         angle2 := pi - abs(angle1);
      else
         angle2 := pi + abs(angle1);
   {Since the segment is the x-axis, the new y-coordinate is wr or wl}
   ynew := -ppt2.x * sin(angle2) + ppt2.x * cos(angle2);
   {now determine if to the left or right of segment}
   if ynew >= 0 then
      if ynew > wltemp then
         wltemp := ynew;
         wlpointer := temppoint
   else
      if ynew < wrtemp then
         wrtemp := ynew;
         wrpointer := temppoint;
   advance temppoint to next point;
end; {while}
{Assign wl and wr to strip, then choose the largest and assign it to widpt}
procedure subtree
{Builds the strip tree recursively}
begin
   widestpoint := widpt(strip, list of points); {call for widest point}
   if widestpoint = nil then {no points in between that are not on segment}
      LSon and RSon := nil;
   else
   begin
      new LSon strip;
      beginpt := beginpt of father strip;
      endpt := widestpoint;
      subtree{recursively build tree}
      new RSon strip;
      beginpt := widestpoint;
      endpt := endpt of father strip;
      subtree{recursively build tree}
   end;
end;{subtree}
```

```
{Now the procedures and functions for the intersection routines}
function intpoints(linequations; beam segment; side endpoints):point;
{Takes the linear equations of the strip tree side and beam and the
 side endpoints and determines if there is an intersection between
 the segment endpoints of these two lines.}
begin
   if (segment y values are equal) and (endpts yvalues equal) then
      intpoints := nil; {parallel horizontal lines}
   else
      if (segment x values are equal) and (endpts xvalues equal) then
         intpoints := nil; {parallel vertical lines}
      else
      begin
         if stripextent.b = 0 then
            slope = 0;
         else
            slope = - stripextent.a / stripextent.b;
         if (slope = beam segment slope) and (stripextent.b <> 0) then
            intpoint = nil; {parallel lines}
         else
         begin
            {xint and yint determined by equations in Section 4.2.3}
            {Now check the intersection points as explained in Section 4.2.3}
            if (the point is a good intersection point) then
               intpoints := this intersection point;
         end;
      end;
end; {intpoints}
procedure stripint(strip; beam; intersectionpoints);
{This procedure determines the linear equation of the beam segment, then
 determines the endpoints of the side of the strip to be tested and its
 linear equation, calls function intpoints for the intersection point,
 then inserts the point if at a leaf node or recursively calls stripint}
begin
   {first determine the linear equation for the beam segment}
   if beam slope = 0 then
      if (xvalues equal) then {vertical}
       begin
         linequation.a := 0;
         linequation.b := -1;
         linequation.c := x;
       end
   else {all others}
   begin
      linequation.a := -(endpt.y - beginpt.y);
      linequation.b := endpt.x - beginpt.x;
      linequation.c := -((lineq.a * beginpt.x) + (lineq.b * beginpt.y));
   end;
```

```
n := 0
repeat
   {determine the endpoints and linear equation for the strip sides}
   increment n;
   if stripslope = 0 then
   begin
      if (x-values equal) then {vertical}
      begin
         if (endpt.y > beginpt.y) then
         begin {orientation of strip with endpt up}
            case n of
               1: begin
                  x[1] := beginpt.x - wl;
                  y[1] := beginpt.y;
                  x[2] := endpt.x - wl;
                  y[2] := endpt.y;
                  end;
               2: begin
                  x[3] := endpt.x - wr;
                  y[3] := endpt.y;
                  end;
               3: begin
                  x[4] := beginpt.x - wr;
                  y[4] := beginpt.y;
                  end;
               4: begin
                  x[5] := x[1];
                  y[5] := y[1];
                  end;
            end; {case}
         end {endpt greater}
      else
      begin
         {similar to above only the beginpt is greater}
      end
      if (n equals 1 or 3) then
         {same linequations as above for beam slope =  zero}
      else
         {same linequations as 'all others' from above}
      end; {vertical strips}
      else {horizontal strips}
      begin
         {determine points and equations similar to the vertical section}
      end;
   end; {zero slope special cases}
```

```
            else {for all others, i.e., all segments with non-zero slope}
            begin
               if endpt.y > beginpt.y then
               begin
                  case n of
                     1: begin
                        x[1] := beginpt.x - t;
                        y[1] := beginpt.y - mperp * t;
                        x[2] := endpt.x - t;
                        y[2] := endpt.y - mperp * t;
                        end;
                     2: begin
                        x[3] := endpt.x - u;
                        y[3] := endpt.y - mperp * u;
                        end;
                     {and so on}
                  end; {case}
               end {endpt greater}
               else
               begin
                  {do the same method for beginpt greater}
               {Now figure the line equation constants the same as was
                done above for all others}
               end; {else}
            end {all other strips}
         if (all linequation constants = 0){wl and wr = 0} then
            intersection points := nil; n := 4;
         else
            intersection points := intpoints(function call);
      until (n = 4) or (intersection points <> nil);
      if (both sons are nil){leaf node} and (intersection points <> nil) then
         place point in the intersection point list;
      if (LSon <> nil) and (intersection point <> nil) then
         recursively call this procedure to continue down the tree;
      if (RSon <> nil) and (intersection point <> nil) then
         recursively call this procedure to continue down the tree;
      end; {procedure}

   begin {main}
      read data
      for all contours do
         buildtree;
      for all contours do
         stripint;

   end. {program}
```

Appendix C

Code for the 3D Version

The following pseudocode was written to supply specifics in some areas and plain language in others to allow the code to be condensed and included in this thesis while providing the basics of the program. In this case the code for the 3D version is presented as it was written to incorporate the 2D versions.

```
program 3d(output, contourdata, beamdata);

function planequation(3 points): record;
{determines the equation of the planes for the baseplane and capplane}
begin
   {compute the direction cosines for the 3 points}
   d1 := 1 / distance from point1 to point2;
   d2 := 1 / distance from point1 to point3;
   a1 := (pt2.x - pt1.x) * d1;
   a2 := (pt2.y - pt1.y) * d1;
   a3 := (pt2.z - pt1.z) * d1;
   b1, b2, b3 computed similarly for points 1 and 3 and using d2;
   {plane equation constants are}
      a := (a2 * b3) - (b2 * a3);
      b := (a3 * b1) - (b3 * a1);
      c := (a1 * b2) - (b1 * a2);
      d := -((a * pt1.x) + (b * pt1.y) + (c * pt1.z));
   planequation := record with these values;
end; {planequation}
```

```
procedure initprism(pointlist);
{Creates both the baseplane and capplane based on three baseplane points}
begin
   p1,p2,p3 := first three points;
   baseplanequation := planequation(p1,p2,p3);
   {add height to the z value}
   capplanequation := planequation(p1,p2,p3);
end; {initprism}
function planeintersection(plane, segment): intersectionpoint;
{Finds the 3D intersection coordinates between a plane and a line}
begin
   d := 1/distance between segment points;
   a1,a2,a3 determined as in planequation;
   temp1 := (plane.a * a1) + (plane.b * a2) + (plane.c * a3);
   temp2 := (plane.a * pt1.x) + (plane.b * pt1.y) + (plane.c * pt1.z);
   if temp1 = 0 then {parallel to plane}
      planeintersection := nil;
   else
      if (temp1 and temp2 = 0) then {line lies in the plane}
         planeintersection := nil;
      else
      begin
         t := (temp2 + plane.d) / temp1;
         intersectionpoint.x := pt1.x - (t * a1);
         intersectionpoint.y := pt1.y - (t * a2);
         intersectionpoint.z := pt1.z - (t * a3);
      end;
end; {planeintersection}

begin {main}
   read data
   initprism(first contour pointlist); {determine plane equations}
   baseintersection := planeintersection(baseplane, beam);
   capintersection := planeintersection(capplane, beam);
   for all contours do
      {2d strip building routines}
   for all contours do
   begin
      {Do 2d intersections in baseplane}
      while (intersection points) do
      begin
         assign x and y and compute z by scalar equations from Section 5.2.4;
         compute distance from beam source;
         assign index as contour number and density of contour;
         assign type of intersection as contour;
      end
   end
```

```
if (no intersections) then
   done;
else
begin
   {add baseplane and capplane to intersection point list}
   assign type of intersection;
   assign x,y,z and distance as above;
   repeat
      {do 2d intersection of contours with baseplane and capplane as the
       beam endpoint. Then compute the closest distance to this point to
       determine the last contour that was entered before reaching the
       capplane or baseplane intersection. Save the index and density}
   until (all contours tested);
   index := index determined from above contour;
   density := density for same;
end
for i := 2 to number of intersections do {Sort Routine, Algorithm 4.1}
   {Same as covered in text and 2d Bentley-Milan Routine}
for all intersection points do
   {This short routine is to insure that if there happens to be two
    intersection points with the same values because of a capplane or
    baseplane intersection at a contour intersection then the contour
    one is discarded and the prism intersection kept}
{Now search for the valid intersection points}
While (not a contour intersection) and (still points) do
   get next point
incontour[this index] := true; {keep track of entering contour}
if (contour intersection) and (z-value within prism) then
begin
   inside := true;
   insert into valid intersection point array;
   firstin := contour {for effective pathlength later}
end
else
begin
   {entered contour but not within prism yet so need to save these
    values for the effective pathlength procedure later}
   increment templevel;
   assign point density into tempdensity[templevel]
   assign point index into tempindex[templevel]
end
{if this intersection was not good must now check for all prism intersections}
```

```
while (not inside prism) and (still points) do
   if (not contour intersection) then
   begin
      inside := true;
      insert into array;
      firstin := type of intersection;
      if (this was a point of common intersection from above) then
         increment templevel;
         save density and index;
   end
   else {contour intersection}
   begin
      if (z-values within prism) then
      begin
         inside := true;
         insert into array;
         firstin := contour;
      end
      if incontour[index] then {already in contour and leaving}
         set to false since leaving this contour;
         decrement templevel;
      else {entering contour}
         set to true;
         increment templevel and save density and index;
   end; {contour}
{Now entered the prism for the first time and must find all other intersection points}
while (typeofpoints = contour) and (z-values within prism) and (points) do
begin
   insert into array;
   if incontour[index] then {already in so leaving}
      set to false
   else set to true; {else entering contour}
   get next point;
end;
while (not inside) and (for all contours) do
   check incontour to see if still inside any contour;
if (not contour intersection) and (inside) then
begin
   insert into array;
   inside := false;
end;
```

```
    {Now set up effective pathlength calculation}
    if firstin = contour then
       level := 1;
    else {entering prism after already having entered contours so use temps}
       assign level, density array and index array the tempvalues;
    if intersections then
       density and index array entries := first point density and index;
    {Now do effective pathlength}
    {procedure is the same for the one given in the 2d Bentley-Milan version
     except for one added line. If the last intersection point is not a contour
     intersection then the calculation point is outside the prism before the
     end of the contours is reached so the density of path is set to zero
     before the effective pathlength calculation is done for this case.}

end. {program}
```

END

DTIC

7-86