CERR: A computational environment for radiotherapy research

Joseph O. Deasy[a)]
Department of Radiation Oncology and Department of Biomedical Engineering,
Mallinckrodt Institute of Radiology, Alvin J. Siteman Cancer Center, Washington University Medical Center,
St. Louis, Missouri 63110

Angel I. Blanco and Vanessa H. Clark
Department of Radiation Oncology, Mallinckrodt Institute of Radiology, Alvin J. Siteman Cancer Center,
Washington University Medical Center, St. Louis, Missouri 63110

(Received 11 January 2002; accepted for publication 25 February 2003; published 24 April 2003)

A software environment is described, called the computational environment for radiotherapy research (CERR, pronounced ''sir''). CERR partially addresses four broad needs in treatment planning research: (a) it provides a convenient and powerful software environment to develop and prototype treatment planning concepts, (b) it serves as a software integration environment to combine treatment planning software written in multiple languages (MATLAB, FORTRAN, C/C++, JAVA, etc.), together with treatment plan information (computed tomography scans, outlined structures, dose distributions, digital films, etc.), (c) it provides the ability to extract treatment plans from disparate planning systems using the widely available AAPM/RTOG archiving mechanism, and (d) it provides a convenient and powerful tool for sharing and reproducing treatment planning research results. The functional components currently being distributed, including source code, include: (1) an import program which converts the widely available AAPM/RTOG treatment planning format into a MATLAB cell-array data object, facilitating manipulation; (2) viewers which display axial, coronal, and sagittal computed tomography images, structure contours, digital films, and isodose lines or dose colorwash, (3) a suite of contouring tools to edit and/or create anatomical structures, (4) dose–volume and dose–surface histogram calculation and display tools, and (5) various predefined commands. CERR allows the user to retrieve any AAPM/RTOG key word information about the treatment plan archive. The code is relatively self-describing, because it relies on MATLAB structure field name definitions based on the AAPM/RTOG standard. New structure field names can be added dynamically or permanently. New components of arbitrary data type can be stored and accessed without disturbing system operation. CERR has been applied to aid research in dose–volume-outcome modeling, Monte Carlo dose calculation, and treatment planning optimization. In summary, CERR provides a powerful, convenient, and common framework which allows researchers to use common patient data sets, and compare and share research results. © *2003 American Association of Physicists in Medicine.* [DOI: 10.1118/1.1568978]

Key words: AAPM format, RTOG format, treatment planning, MATLAB

I. INTRODUCTION

The design of intensity modulated radiation therapy (IMRT) treatment plans is complex and typically involves very specialized software. In addition, evaluation of the resulting treatment plans is complex and requires a careful analysis of the resulting dose distribution(s), through visual dose review, dose–volume histograms (DVHs) and possibly outcome-related plan evaluations [e.g., normal tissue complication probabilities (NTCP) or tumor control probability (TCP)]. Unfortunately, when results of research in this area have been presented, either orally or in print, a key element of the scientific method has typically been missing because *the results are typically not reproducible or even reviewable in detail by other researchers*.

This may be partially due to the lack of a widely available graphical data analysis and programming environment which could be used to read, review, and compare dose, contour, and image data from a wide range of clinical and academic planning systems. This hurdle reduces the probability that a new treatment planning algorithm or tool will be compared either with previously published algorithms or with other algorithms in the future. Optimized treatment planning, in particular, lacks inter-institutional method comparisons. As others have pointed out, computational research should be reproducible wherever possible.[1,2] We describe a computational environment for radiotherapy research (CERR, pronounced ''sir''), developed to facilitate reproducible research in radiation oncology treatment planning. CERR, currently at ''version 2,'' presently includes tools for reading AAPM/RTOG (America Association of Physicists in Medicine and Radiation Therapy Oncology Group, respectively), archive treatment plans, along with graphical plan review tools, further described in the following. We view this system, along with the underlying MATLAB environment, as a basic framework to make inter-institutional treatment planning research more feasible.

CERR was built using a high-level graphical analysis and

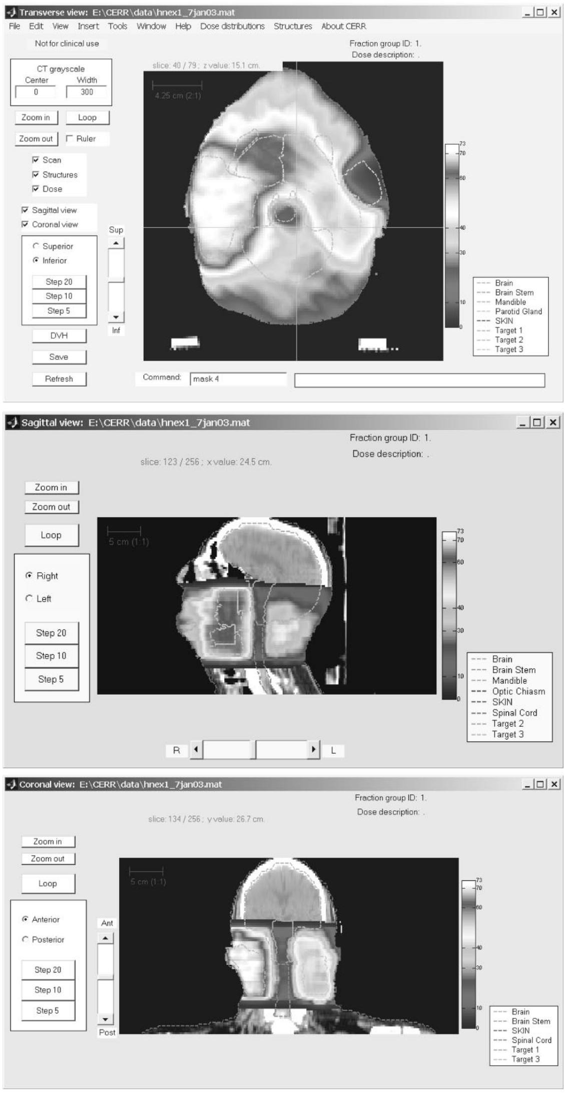


FIG. 1. A grayscale image of the slice viewers: the axial viewer (top), the sagittal viewer (middle) and the coronal viewer (bottom). A full-color image is available in EPAPS (Ref. 16).

programming package, MATLAB (The Mathworks, Inc.: www.mathworks.com). CERR can be viewed as an "open source" environment to develop and share developments in treatment planning tools, while leveraging the high-level graphical and programming tools, and widespread adoption, of the MATLAB simulation system.

Many clinical radiation therapy treatment planning systems can automatically export patient data files to the RTOG data exchange format (referred to hereafter as the AAPM/RTOG format). The precursor to this format was designed by Michael Goitein[3] based partly on an earlier format for exchanging digital images.[4] The history and current specification of the format is available by following links found at the Image Guided Therapy Center (the ITC, formerly known as the 3-D RTOG Quality Assurance Center).[5] Systems which currently implement AAPM/RTOG format output include six commercial systems and eight custom academic systems; several more commercial systems have announced intentions to provide AAPM/RTOG format archiving. See the aforementioned web page for an up-to-date list. The newer DICOM extensions for radiation therapy-specific objects make it an attractive method for importing treatment planning information into MATLAB. This is just what Spezi *et al.* have done, in the form of a "DICOM-RT Toolbox for MATLAB."[6] We are in the process of interfacing their toolbox with CERR, to expand the import options and improve the convenience of importing data.

The MATLAB system can be used to interactively manipulate the plan archive once it is converted into the CERR format using the supplied program and associated utilities, as further discussed in the following. The slice viewing programs allow for visual review of the CT scan data, contours, dose distributions, and digital films, in the axial, sagittal, and coronal planes (see Fig. 1). AAPM/RTOG format ASCII data fields can be easily queried at the MATLAB command line. Along with these functions, the plan viewing programs serve as extensive examples of how to conveniently access treatment planning data in the CERR format from within MATLAB. Care was taken to make the plan archive format within CERR as coherent and succinct as possible, to facilitate manipulation of treatment plans.

Software system selection. (Compared to other parts of this report, this section is naturally based more on opinion.) The current version of MATLAB (version 6.1 or later, hereafter referred to as MATLAB-6) was selected as the software tool for this project, after a review of alternatives, due to: (a) simple syntax, (b) convenient and flexible high-level language constructs, (c) an intuitive and convenient user interface including an integrated debugger, (d) extensive graphical capabilities, including 3-D visualization tools, (e) multidimensional array capabilities, (f) extensive scientific base of user-contributed code, (g) good floating-point computational speed, especially when the instructions can be written using vector operations, (h) cross-platform availability, especially between Windows, Unix/Linux, and Macintosh versions, (i) the ability to compile applications (with some workable constraints) into stand-alone graphical user interface (GUI) programs which can be distributed freely (compilation was immediately successful for CERR version 1, but early attempts to compile version 2 have met some obscure errors—we will attempt to work around the apparent compiler limitations to produce stand-alone executables), and (j) the wide use of MATLAB among researchers. Similar commercial high-level systems, such as IDL (from Research Systems, Inc., http://www.rsinc.com) and PV-WAVE (from Visual Numerics, Inc., http://www.vni.com) have some but not all of these advantages. JAVA has advantages in terms of number of platforms supported, ease of running parallel threads, cost, and extensiveness of the GUI toolkit, but is otherwise less functional than MATLAB [points (a)–(e)]. In particular it is not convenient to manipulate multidimensional data in JAVA, and the functional programming style of MATLAB seems to be better suited to numerically intensive work compared to object-oriented programming. Python is an attractive general purpose programming language which has high-level scripting-style syntax, cross-platform portability, is freely available, and has useful "open source" scientific modules.[7] However, Python lacks integrated visualization capabilities, does not have comparable debugging facilities, seems to evolve new functional capabilities at a slower rate than MATLAB, and has far fewer scientific users. CERR could potentially have been developed using any of these environments. PV-WAVE has been used to good advantage in radiotherapy research.[8,9] However, we believe MATLAB is an excellent choice, and arguably the current best choice, to within quibble uncertainty, to provide an open, cross-platform, web-distributable, environment for the comparison of reproducible radiotherapy research results. Of course, all of these points are debatable.

II. DESCRIPTION OF CERR

A. The CERR format

The CERR format is intended to be a compact, self-describing ''object'' containing all the treatment plan archive data. The CERR format's basic structure is storage of different types of information in different elements of a cell array. Most elements of the cell array are structures associated with a different type of AAPM/RTOG data object: dose distributions, CT scans (or other image scans), dose volume histograms, structure contours, digital films, or beam definitions. All objects of a given type are stored in a structure array within a single cell. (A MATLAB-6 cell array is an indexed list of arbitrarily heterogeneous objects. A MATLAB-6 structure array is also an indexed list, but each element has container fields which are named, thus: var.fieldName. The fields can store arbitrarily heterogeneous objects.)

The keywords defined in the AAPM/RTOG format are fundamental to the CERR format. Essentially, the CERR format follows the AAPM/RTOG format regarding definitions of information of interest. For each type of treatment planning object (image scans, dose distributions, etc.), a MATLAB structure is created with fields to match the fields defined for that object by the AAPM/RTOG format. For example, in a particular archive, planC{8}(3).structureName returns ''BLADDER,'' because, in that archive, the eighth element in the cell array holding the archive (named planC) contains structure outline information, and the name of the third structure is ''BLADDER.'' The corresponding RTOG key words are ''STRUCTURE NAME.''

Going from the AAPM/RTOG specification key words to CERR field names is straightforward. Our variable and function naming convention is to capitalize all words which make up the variable name except the first. An exception is that abbreviations (CT, Rx) are always capitalized. Spaces and nonletter characters (i.e., '',' '' ''-,'' ''(,'' and '')'') are removed. Thus, ''Head in/out'' becomes ''headInOut.'' All key word values/settings are stored as strings or numerical objects in the appropriate structure field.

Currently, CERR reads all AAPM/RTOG format data except that associated with brachytherapy seeds, which would be straightforward to include if needed.

As another example, for a certain plan archive, the MATLAB command

planC{4}(1).contour(46).segments(2).points

returns a matrix of size 40×3 which contains the coordinates of the points for the second segment on the 46th scan slice for the first structure. Structure information is all contained in the fourth cell of the planC cell array.

However, some data stored in files other than the aapm0000 file (which contains catalogs of the contents of the other files) required additional non-RTOG structure field names in CERR. For example: ''doseArray'' is the field in the dose structure which contains 3-D arrays of double precision dose values. If there were four dose distributions then the structure which contains dose data would have four indexed entries. Table I describes the CERR formats for the treatment plan objects. New structure fields, with descriptive names, will probably be added in the future as CERR capabilities increase. The arbitrary addition of new key words does not collide with the use of current keywords, and therefore does not break the code.

The structure to hold all the plan archive 3-D image data has several field names, including: ''scanArray,'' ''scanInfo,'' and ''scanType.'' scanInfo contains a structure which stores all the CT slice key word values for all slices as described by the information in the aapm0000 file; scanArray holds the associated scan array multidimensional array; scanType is a string giving the type of scan, e.g., ''CT.'' As a result, the first element of this data object will contain all data associated with one scan, the second element will contain all data associated with the second scan (which may or may not be the same type of scan), etc.

The compact CERR representation of the treatment plan archive reduces code complexity and length. The self-

TABLE I. AAPM/RTOG data objects and their corresponding CERR representations.

AAPM/RTOG data object	CERR format
CT scan	Stored in a single unsigned 2 byte integers (unit16) 3-D matrix. A ''uniformized'' CT data set with uniform axial spacing for coronal and sagittal viewing is also created.
Structure outlines	Matrices of double precision numbers stored in a hierarchical structure object with field names ''contour,''''segments,'' and ''points.'' For faster access, axial co-registered binary masks are also stored in a sparse ''raster segment'' format (see the text). For sagittal and coronal visualization, structures are also co-registered to the uniformized CT scan in a sparse indexed format.
Beam geometry	The beam definition file is read into a cell array, where each cell is an ASCII line of the corresponding beam definition file. (More useful representations will be the subject of future work.)
Dose volume histogram	Matrix of double precision numbers.
Digital films	Matrix of unsigned 1 or 2 byte integer values.
Dose distributions	3-D array of double precision values.

describing nature of the format makes the code more humanly readable and reduces the need for interspersed code comments.

Once a plan archive has been converted into CERR format, it can be analyzed either graphically or via programmed manipulations (for example, to extract esophagitis dose-volume histogram data). The entire CERR plan archive is stored using the efficient MATLAB ‘‘.mat’’ binary format: all plan information is stored in a single file. Loading or saving a CERR plan archive typically takes only a few seconds. This method of loading or saving a treatment plan archive is far faster than reading AAPM/RTOG format files and of course bypasses all the initial parsing performed by CERR. Once CERR plans are read into MATLAB, they can easily be reviewed, manipulated, rearranged, or re-formatted.

In order to reduce stored-memory usage and network transfer times, we integrated the bzip2 compression utility into CERR (see, for example, http://sources.redhat.com/bzip2) ‘‘bzip2’’ runs on a variety of platforms, is freely available, and in our tests outperformed other compression formats such as gzip or zip. The user chooses the compressed or uncompressed format when saving a CERR archive.

The use and further development of CERR will probably spur further structure field additions to the CERR format. As mentioned, adding cell elements to represent completely new objects, or adding structure field names, can be done trivially without breaking the existing code. Hence, forward compatibility between CERR formatted archives and CERR software will be maintained.

B. The AAPM/RTOG archive conversion tool: CERRImport

The user can specify archives to be converted either through function call options in MATLAB (if, for example, many archives are to be processed) or through a file-selection GUI. The time required to convert an archive varies. In particular, creation of a ‘‘uniformized’’ CT scan and co-registered structure masks, to allow sagittal and coronal viewing, can by itself take up to 10 min or so even on a 2 GHz PC. The rest of the import process can take roughly 5 min or so on the same PC. As an example, one archive (a lung case) having 106 CT slices of size 512×512, and 13 structures, required 15 min for conversion on a 2 GHz Pentium IV. Most of this time was spent in producing the ‘‘uniformized’’ CT scan set and co-registered structure masks. The resulting archive was 115 Mbytes before compression, and 51 Mbytes after bzip2 compression.

The RTOG format specifies anatomical contours as transverse polygons. Accurate conversion of polygonal contours to the binary masks actually used for treatment planning is challenging for two reasons. First, user-defined contours often accidentally self-intersect. Second, there is no strict definition of when a voxel shall be included as being inside the contour. We solved the first of these problems by creating an ‘‘excision repair’’ routine, which snips off the polygonal loop created by the self-intersection. The exposed end point vertices are then connected by a new edge. The user is notified when this routine is used. The issue of which voxels are to be defined as being inside the contour was resolved by creating an algorithm which knows whether an axial CT voxel center is inside or outside the contour, and only includes voxel centers which are inside the contour (see the routine scanpoly.m under the ScanConversion subdirectory for more details). This inside/outside distinction is typically not tracked in standard polygon scan algorithms. The algorithm can miss a voxel if two enclosing sides (i.e., left/right or top/bottom) of the polygon both go through the voxel in question, but that is a rare special case. Figure 2 shows an example of a lung slice with a challenging lung contour having convoluted edges, but correct scan conversion.

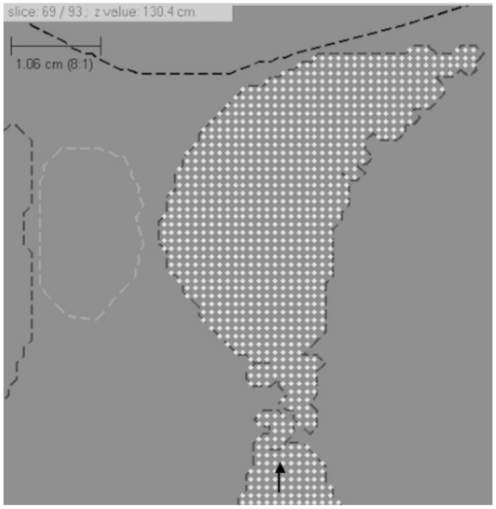
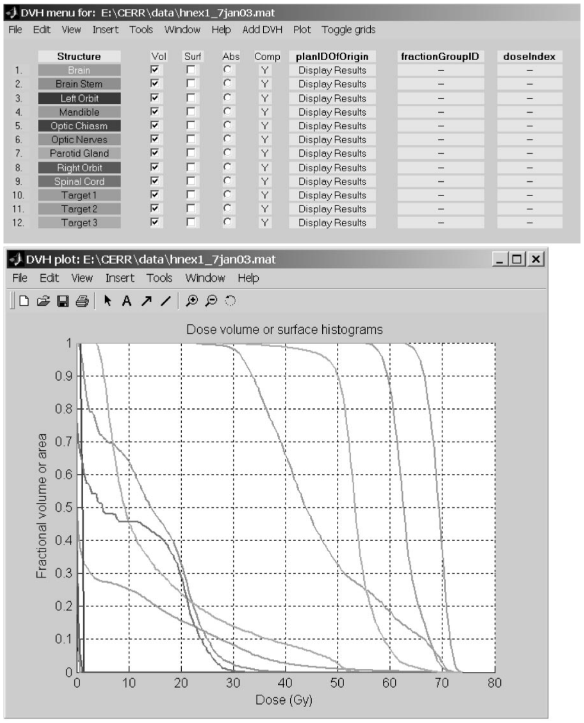


FIG. 2. Conversion of anatomical contour polygons into binary voxel masks. An axial lung contour and the resulting binary mask are shown. Note the convoluted self-intersecting contour at the lower left of the image. The associated binary mask, however, was correctly determined.

After producing the scan-converted binary mask for a given structure and axial CT slice, structures are stored in CERR in ‘‘raster segment’’ format: the positions of the first and last voxels of row-wise filled-in mask segments are stored instead of all the voxel positions in the segments. This is a relatively memory-efficient sparse format, is unambiguous in terms of registration with the image data, and can be rapidly converted into a full mask.

In order to implement sagittal and coronal viewing, along with future uses that will require 3-D data sets, the user has the option to create a ‘‘uniformized’’ CT scan and co-registered anatomical structure masks. The uniformized CT scan is simply a CT scan data set covering the same axial range as the first CT data set, but with a constant inter-slice spacing. Another data structure holds a 32 bit mask, each 32 bit number being registered to a uniformized CT voxel. Each bit specifies the presence or absence of a structure in that voxel: if the first bit has a value of 1, then the co-registered CT voxel holds the first structure, etc. It is these structures which are viewed in the sagittal and coronal viewers. However, DVHs are computed based on the original scan conversion co-registered to the original axial CT slices (see Fig. 3).


DVH menu for: E:\CERR\data\hnex1_7jan03.mat
File Edit View Insert Tools Window Help Add DVH Plot Toggle grids
DVH plot: E:\CERR\data\hnex1_7jan03.mat
File Edit View Insert Tools Window Help
Dose volume or surface histograms
Fractional volume or area
Dose (Gy)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
10
20
30
40
50
60
70
80

	Structure	Vol	Surf	Abs	Comp	planIDOfOrigin	fractionGroupID	doseIndex
1.	Brain	☑	□	○	Y	Display Results	–	–
2.	Brain Stem	☑	□	○	Y	Display Results	–	–
3.	Left Orbit	☑	□	○	Y	Display Results	–	–
4.	Mandible	☑	□	○	Y	Display Results	–	–
5.	Optic Chiasm	☑	□	○	Y	Display Results	–	–
6.	Optic Nerves	☑	□	○	Y	Display Results	–	–
7.	Parotid Gland	☑	□	○	Y	Display Results	–	–
8.	Right Orbit	☑	□	○	Y	Display Results	–	–
9.	Spinal Cord	☑	□	○	Y	Display Results	–	–
10.	Target 1	☑	□	○	Y	Display Results	–	–
11.	Target 2	☑	□	○	Y	Display Results	–	–
12.	Target 3	☑	□	○	Y	Display Results	–	–

FIG. 3. A grayscale image of the CERR dose-volume histogram and dose-surface histogram menu (top) and plotting window (bottom). Plots can be made on a single relative scale in addition to individual plots showing absolute volumes or surface areas. A full-color image is available in EPAPS (Ref. 16).

C. The axial slice viewer

CERR is normally invoked by starting the axial slice viewer program, shown in the top of Fig. 1. The slice-viewer has the following capabilities: (1) isodose or colorwash dose distribution display, (2) display of anatomical contours, (3) CT display, (4) adjustable CT center and width display settings, (5) a ruler, (6) resizable window, (7) zoom in/out, (8) navigation tools, (9) a menu listing the defined structures, (10) slice-specific legends, (11) many command line operations, and (12) movie loop display mode.

One of the key advantages of MATLAB is that GUI programming and testing is somewhat simplified. The slice viewer program code can be used as a guide and a template to utilizing CERR format plan archives within MATLAB (Fig. 3).

The CERR command line, shown as an editable text box in the axial slice review tool, is meant to be a powerful mechanism for adding sophisticated user actions. Valid CERR commands can always be determined by typing ‘‘help’’ at the command line. Various commands are available. For example, one can query AAPM/RTOG keyword tags using commands like: ‘‘list structure 1,’’ ‘‘list dvh 2,’’ etc. The command ‘‘list structure 3 points 45,’’ will list all the contour points of structure 3 on slice 45. The user can view digital films as grayscale images by typing ‘‘show film entry 2,’’ etc. (The word ‘‘entry’’ is used here, as digital films are likely numbered differently from their archive ordering.) In addition, any user-definable option can be changed interactively using the CERR command line, for example by issuing the command ‘‘set optS.CTLevel=100.’’ The options file can be reloaded with the command ‘‘reload options.’’ As a more exotic example, CERR can make ‘‘avi’’ movies of paging with the axial slice viewer for faster playback, archiving, or display outside of the CERR environment (e.g., PowerPoint).

D. Contouring tools

There are many applications in which new anatomical structures need to be defined or redrawn. For example, the user may wish to have a consistent definition for all the parotid salivary gland contours, or the treatment planning system may not allow for overlapping structures. For this reason, we implemented a suite of contouring tools. Anatomical contour manipulation tools include: copying to adjoining slices, nudging, scaling, deleting, modifying point positions, adding points, snipping out ‘‘bad’’ sections, and deleting. A detailed description of the individual contouring tools, and other CERR features, can be found in the cerr_user_instructions.html file, available from a link on the CERR Version 2 webpage (http://deasylab.info).

E. Utility programs

A number of general-purpose utilities were written to facilitate code development and are included with the CERR software package. For example, routines were written to convert locations between the AAPM coordinate system and matrix indices (and *vice versa*). Various ASCII parsing routines were also written, including: ‘‘words·m’’ (returns the number of blank delimited words in a string), ‘‘word·m’’ (returns the *n*th word in a string), ‘‘deblank2·m’’ (eliminates leading and trailing blanks in a string), ‘‘insert·m’’ (insert one string into another), and file2cell·m (load an ASCII file into a cell array, one line per cell). The utility function ‘‘opts4Exe·m’’ reads any file with commands to set fields for a structure, and passes the modified structure to the calling function. This was critical for loading user-defined settings in a way consistent with the eventual production of stand-alone executables which cannot use the dynamic evaluation command (the ‘‘eval’’ command).

F. Preliminary validation

The dosimetric accuracy of CERR was tested primarily by comparing dose-volume histogram lines computed in the original treatment planning system versus CERR computations. Typically, DVH lines nearly overlapped each other, but for some DVHs for the NOMOS Peacock plans, significant, though small, differences were seen, probably due to differences in the structure masks used by the NOMOS Peacock and those we determine from the NOMOS polygon structure definitions. Further comparisons of DVHs with other NOMOS plans and plans from other systems are ongoing and will be reported later. The contouring tools have been tested on dozens of head and neck plans for parotid recontouring.[10]

These simple tests are in no way expected to match the rigorous quality assurance required for clinical treatment planning systems. CERR is currently a research system only and is not licensed or appropriate for clinical use.

G. Usage instructions

If the user of CERR needs access to only a portion of the RTOG format information, components that are not to be loaded (e.g., CT scans) can be specified through parameter settings described in the CERR user instructions. An effort was made to hard-code an absolute minimum of parameter values and instead pass as many such parameter values as possible through user settings. Various other user-definable options, also described in the user instructions file, are also read in from an ASCII file at the startup of either the CERRImport or CERRSliceViewer program. Expanding CERR user-defined options via this mechanism is straightforward and requires minimal coding changes.

H. Distribution

This release of CERR is available as a download from our web page. Improvements or bug fixes to the software are to be expected, and will be posted on that web site. More detail on features and use instructions can be found in the cerr_user_instructions.html file, available from a link on the CERR Version 2 webpage (http://deasylab.info).

An AAPM/RTOG example archive for testing import capability is available from the ITC webpage. We have also posted three CERR format compressed archives: a head and neck IMRT case, a lung case, and a prostate case (derived from the ITC data).

MATLAB source code and graphics programs are built to be inherently portable between operating systems. CERR was tested on Microsoft Windows 2000 and XP, although we expect it will also run on other systems which support MATLAB with only minor modifications (a Linux version is in beta testing). The only code in the system which is explicitly operating system dependent is the call to the bzip2 executable, which could easily be modified for other operating systems.

III. DISCUSSION

A. CERR as a radiation therapy treatment planning software integration tool

MATLAB can be linked to JAVA, FORTRAN, C, C++, or Visual Basic programs. This flexibility should prove useful for integrating, for example, Java GUI elements.

B. Copyright and collaboration philosophy

The base CERR package (i.e., the Washington University code) is free to use or modify for ‘‘research use,’’ which means any noncommercial and nonclinical use. We have adopted the philosophy that any contributor of computer code to CERR should retain the copyright to that code, and is therefore responsible for the usage terms of their code. Institutions which support research often expect to be able to protect the commercial value of their intellectual property. Any use of CERR code by commercial users should be negotiated with the copyright holders. However, we hope that users will use our code as a guide and template to creating other CERR-based tools.

C. Potential future uses

Several applications of CERR are ongoing in our laboratory, including integration with a Monte Carlo dose calculation code,[11] analysis of dose-volume-outcome data,[12] demonstration of new plan review tools,[13] TCP and NTCP calculations, and the investigation of intensity modulated radiation therapy optimization algorithms.

Incorporating DICOM-RT read and write capability into CERR is of particular interest. We are currently pursuing this in collaboration with the authors of the DICOM-RT Toolbox for MATLAB.[6] That capability, along with AAPM/RTOG import capability, would allow for plans to be imported from nearly any commercial or academic treatment planning system. Moreover, if the same CT scan and anatomical structures are pushed to multiple treatment planning systems, and the results are read into CERR, comparisons of different approaches to treatment planning could be made. The treatment planning systems need not reside at the same site.

Currently, using CERR requires purchasing MATLAB. (We believe we will also be able to make a stand-alone executable available, although there have been obscure errors from the Mathwork’s compiler in the initial compile attempts.) Nevertheless, we hope that users and collaborators will find CERR useful enough to spur interest in further developments, thereby leading to a yet more useful research environment. Previous efforts to develop treatment planning tools in radiation oncology, while successful at demonstrating important tool integration approaches,[14,15] have not provided a convenient framework through which software contributions and research results are widely shared. We believe CERR and MATLAB, probably combined with other ‘‘open source’’ tools such as the DICOM-RT Toolbox, have the potential to provide such an environment, primarily due to the strengths of MATLAB and common import/export formats.

IV. SUMMARY

The CERR programs combined with the MATLAB environment provide the following benefits.

(1) The ability to import treatment plans from a wide variety of commercial and academic treatment planning systems using the AAPM/RTOG archive format.
(2) A treatment plan data architecture which is self-describing, compact, easily manipulable, and extendable.
(3) The ability to manipulate treatment plans within a powerful data analysis and programming environment, for example for dose-volume-outcomes analyses.
(4) Visual plan review tools (axial, sagittal, and coronal viewers).

(5) The source code is supplied, allowing users to modify it and use it as a template for their own projects.
(6) The ability to link external programs in other languages.
(7) Cross-platform (Windows, Unix, Apple) capability with minimal modifications.
(8) Perhaps most importantly, *CERR can be used for sharing and reproducing radiation therapy treatment planning research results*.

ACKNOWLEDGMENTS

We thank Walter Bosch, of the Image Guided Therapy Center at Washington University, for kindly making several treatment plan archives available to us, for comparison with CERR, through their on-line review tool. We also thank Dan Low and Mark Wiesmeyer for helpful conversations about CERR features. It is a pleasure to acknowledge discussions, circa 1994, with Timothy Holmes on the desirability of a MATLAB "radiotherapy toolbox." This work was partially supported by Grant No. R29 CA85181 and a grant from Computerized Medical Systems, Inc.

[a]Electronic mail: deasy@radonc.wustl.edu

[1] J. B. Buckheit and D. L. Donoho, "WaveLab and Reproducible Research," in *Wavelets and Statistics*, edited by A. Antoniadis and G. Oppenheim (Springer, New York, 1995), Vol. 103, pp. 55–81.
[2] J. Claerbout and M. Karrenbach, http://sepwww.stanford.edu/sep/jon/blurb.html, 1994.
[3] M. Goitein, "Specifications for Tape Format for Exchange of Planning Information, version 2.2," in *Evaluation of Treatment Planning for Particle Beam Radiotherapy*, edited by M. Goitein *et al.* (National Cancer Institute, Bethesda, MD, 1985), Sec. 5.11.
[4] B. S. Baxter, L. E. Hitchner, and G. Q. Maguire, Jr., "AAPM Report #10: A Standard Format for Digital Image Exchange," American Association of Physicists in Medicine, 1982.
[5] W. Harms, "Specifications for Tape/Network Format for Exchange of Treatment Planning Information, version 3.22," Image Guided Therapy Center at Washington University, http://rtog3dqa.wustl.edu, 1997.
[6] E. Spezi, D. G. Lewis, and C. W. Smith, "A DICOM-RT-based toolbox for the evaluation and verification of radiotherapy plans," Phys. Med. Biol. **47**, 4223–32 (2002).
[7] P. F. Dubois and T. Y. Yang, "Extending Python with Fortran," Comp. Sci. Eng. **1**, 66–73 (1999).
[8] A. Lomax, S. Scheib, G. Munkel, and H. Blattman, "The comparison of spot-scanning proton radiotherapy with conventional photon therapies," in *XIth International Conference on Computers in Radiation Therapy*, edited by A. R. Hounsell, J. M. Wilkinson, and P. C. Williams (Medical Physics Publishing, Madison, WI, 1994).
[9] C. J. Moore, A. R. Hounsell, P. Sharrock, A. Shaw, P. C. Williams, and J. M. Wilkinson, "A computerized system for conformal therapy using a Philips SL-25 linear accelerator and multi-leaf collimator: first and second generation developments," in Ref. 8, pp. 104–105.
[10] A. Blanco, C. Chao, J. Deasy, and D. Low, "Recovery kinetics of salivary function in patients with head and neck cancers receiving radiation therapy," Int. J. Rad. Oncol. Biol. Phys. **54**, 166 (2002).
[11] J. O. Deasy, E. K. Lee, I. Kawrawkow, and C. Zakarian, "A prototype Monte Carlo-based IMRT treatment planning research system," Med. Phys. **29**, 1254–1255 (2002).
[12] J. Bradley, J. Deasy, S. Bentzen, W. Bosch, and J. Purdy, "Irradiated esophageal surface area predicts for esophagitis in patients treated for non-small cell carcinoma of the lung," Int. J. Radiat. Oncol., Biol., Phys. **54**, 105–106 (2002).
[13] J. F. Dempsey, J. O. Deasy, A. Lomax, M. Wiesmeyer, W. Bosch, and D. Low, "Treatment plan review tools incorporating spatial dose-volume information," Int. J. Radiat. Oncol., Biol., Phys. **51**, 125 (2001).
[14] R. E. Drzymala, M. D. Holman, D. Yan, W. B. Harms, N. L. Jain, M. G. Kahn, B. Emami, and J. A. Purdy, "Integrated software tools for the evaluation of radiotherapy treatment plans," Int. J. Radiat. Oncol., Biol., Phys. **30**, 909–919 (1994).
[15] J. Jacky *et al.*, "Portable software tools for 3D radiation therapy planning," Int. J. Radiat. Oncol., Biol., Phys. **30**, 921–928 (1994).
[16] See EPAPS Document No. E-MPHYA6-30-027305 for color versions of Figs. 1 and 3. A direct link to this document may be found in the online article's HTML reference section. The document may also be reached via the EPAPS homepage (http://www.aip.org/pubservs/epaps.html) or from ftp.aip.org in the directory/epaps/. See the EPAPS homepage for more information.