AMIDE: A Free Software Tool for Multimodality Image Analysis

Andreas M. Loening Sanjiv S. Gambhir

Crump Institute for Molecular Imaging, UCLA School of Medicine
Department of Radiology and the Bio-X Program, Stanford University

INTRODUCTION

With the increasing use of multimodality imaging, a need has arisen for software tools that can seamlessly handle the data sets generated by these studies (e.g. PET/CT, SPECT/MRI). To fill this need, AMIDE [1] (Amide's a Medical Image Data Examiner) has been developed as an open source tool for viewing and analyzing both single and multimodality volumetric data sets, and is freely available from the AMIDE web site (http://amide.sf.net).

Central to the program's capabilities is the automatic, non-destructive data reslicing implemented within the program. Multiple data sets can be freely shifted, rotated, viewed, and analyzed, with the program automatically handling the requisite data interpolation from the original data sets. This ability also facilities the abstraction of the underlying digital data, allowing for an intuitive and consistent user interface that avoids fixed image planes and voxel based dimensions.

FEATURES

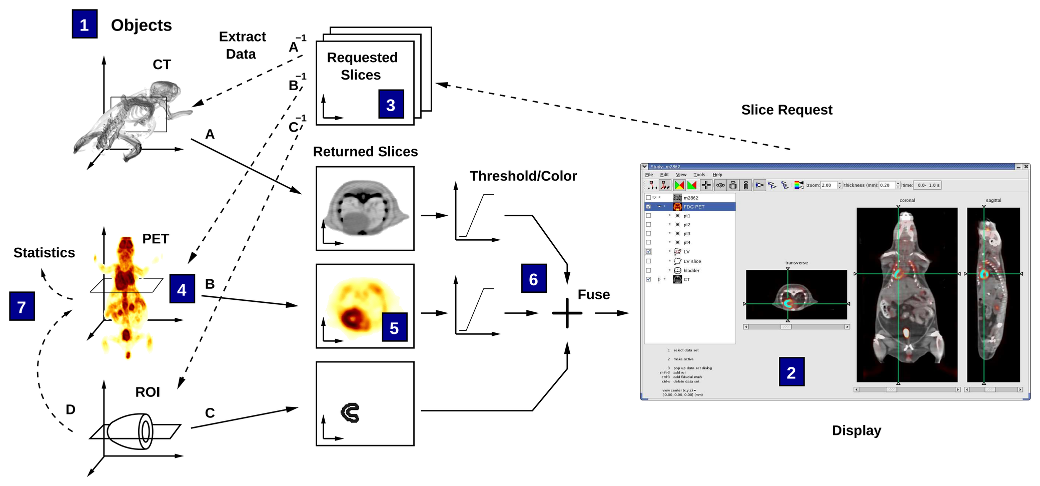
- User's view of data is divorced from the underlying digital format of the data, allowing slices of arbitrary orientation, thickness, zoom, and time period to be viewed.
- Multiple data sets can be simultaneously handled and displayed, with independent rotations, shifts, color maps, and thresholding for each data set.
- Slices are automatically interpolated directly from the original data, allowing instantaneous data set rotations and shifts. Data sets are never destructively resliced.
- Imports Concorde, ECAT 6/7, Acr/Nema 2.0, Analyze (SPM), DICOM 3.0, and InterFile 3.3, along with a variety of raw data types (32/64 bit float, signed or unsigned 8/16/32/64 bit integer) in little, big, or PDP endian formats. Formated file type importation uses the (X)MedCon library [2].
- GPL License (GNU General Public License [3]). License ensures that AMIDE is and will remain free to use, distribute, and modify.
- Volumetric regions of interest (ROI's) can be drawn and analyzed. Currently, ellipsoids, elliptic cylinders, boxes, and isocontours (2D or 3D) are supported.
- Volume renders single and multiple data sets. Sequences of renderings can be output as MPEG1 files.
- Series of slices from the data set can be viewed over time or space. Fly through movies can be generated and output as MPEG1 files.
- Internal file format is a portable and readable XML based approach.
- Real world values (mm, seconds) used for units. Data set values can optionally be presented in terms of Percent Injected Dose per gram tissue (%ID/g) or Standardized Uptake Value (SUV) metrics.
- Rigid body registration can be accomplished through the use of fiducial markers.
- Filtering and cropping of data sets. Gaussian and Median filters are currently implemented.

IMPLEMENTATION

Object Handling

In order that multiple objects (Data Sets, ROIs, Fiducial Markers) can be independently handled within AMIDE, each object is assigned its own Euclidean space. When information is needed from an object, AMIDE automatically handles the requisite affine (linear plus translation) transformation between spaces.

For instance, data sets (1) in AMIDE are stored in memory with their original voxel size and format (8/16/32 bit integer, 32/64 bit float). When the program (2) needs new slices, slice



request objects are formed (3) and translated into the local coordinate frames of the appropriate data sets (4). The slice request objects are then filled with data (5) using either nearest neighbor (better speed) or trilinear (better quality) interpolation, followed by thresholding, coloring, and fusing (6) for display to the user (2). This approach allows data set movements, rotations, and scalings to be accomplished by simple alterations of the object's coordinate space parameters, rather than destructive reslicing of the actual data.

The data hierarchy in AMIDE is built around a tree abstraction composed of a succession of objects, allowing movement operations to be successively mapped down to an object's children. For example, when a data set is moved, ROI's and fiducial markers that are children of that specific data set will follow along with the movement, while unassociated ROI's and fiducial markers will not.

Language/Toolkit

The C programming language was chosen for a variety of reasons, including the familiarity of researchers with this language, the ready availability of high performance compilers, and the desire to avoid burdening the program with requirements on underlying proprietary packages (e.g. IDL, Matlab). The specific compiler used in this work was GCC (GNU Compiler Collection, gcc.gnu.org).

Version 2 of the GTK+/GNOME toolkit (www.gnome.org) was used for the user interface, and much of AMIDE's core functionality is actually implemented as an extension to this toolkit. The toolkit was chosen for a combination of free licensing (GPL), a C language interface, and portability (both UNIX and Microsoft Windows are supported).

Internal File Format

AMIDE studies are saved in an XML (eXtensible Markup Language) based directory format, in which each object's parameters and format information are saved in a text file, with any associated raw image data saved in a separate binary file. This approach allows header information to be easily accessed and manipulated externally to the program, guards against endian incompatibilities, and makes backward and forward file compatibility easy to maintain between versions. A flat file version is also supported.

Algorithms

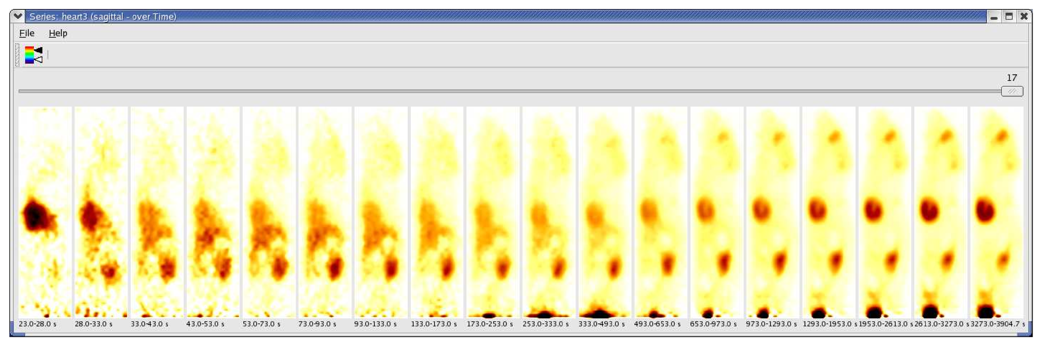
Volume rendering in AMIDE is accomplished using the VolPack volume rendering library [4], which accelerates rendering using a shear-warp factorization algorithm. This software library based approach is flexible, portable, and provides for true volume rendering (as opposed to the surface rendering of many other approaches). Transformation of rendered sequences to MPEG-1 files is done using the fame MPEG encoding library (fame.sf.net).

Rigid body registration is implemented through the use of fiducial markers and the Procrustes rigid body alignment algorithm without scaling [5].

Finite impulse response (FIR) filtering (e.g. Gaussian) is implemented using an overlap+add method with a 64^3 point FFT. Median filtering is accomplished using a partial sort median finding algorithm [6], with spatial coherence ignored for algorithmic simplicity.

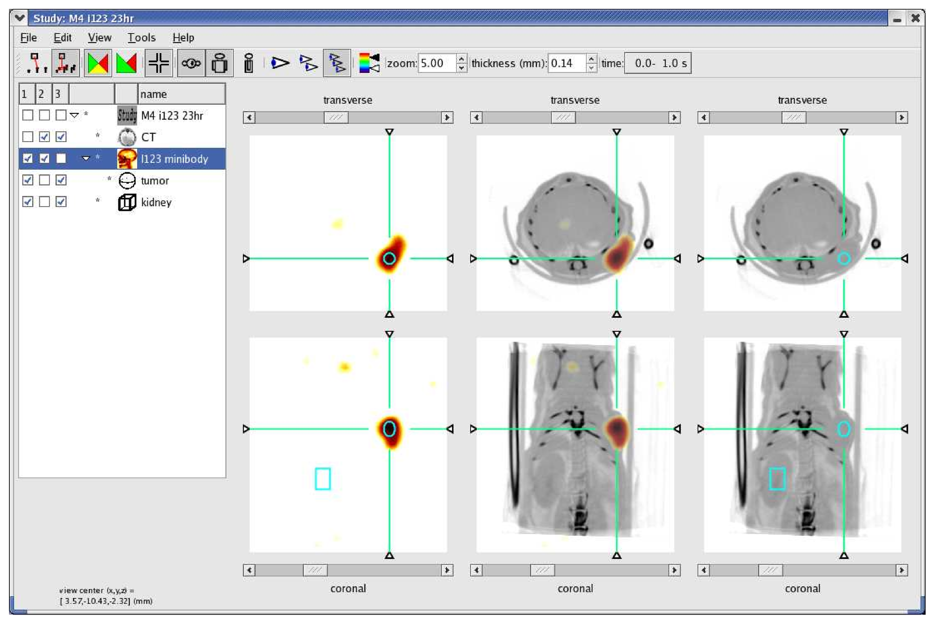
EXAMPLES

Dynamic Series



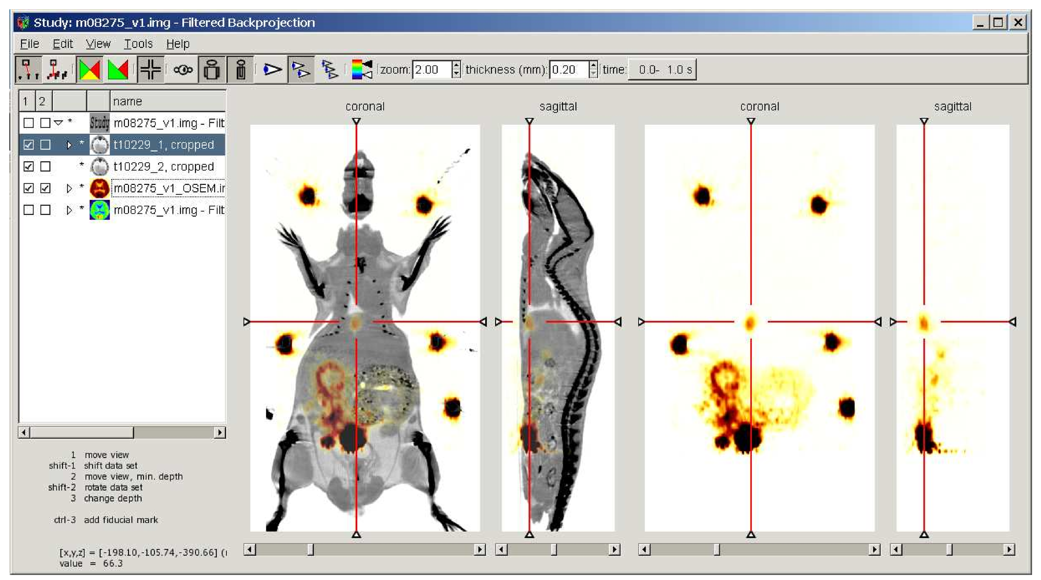
An [18]F]FDG dynamic scan of a mouse from a Concorde P4 scanner. A sagittal slice is shown over time using the slice viewer.

microSPECT/microCT



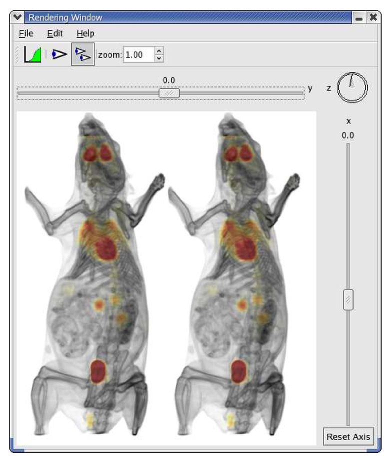
A microSPECT/microCT data set acquired with a Gamma Medica A-SPECT system. The mouse has been injected with a [^{123}I] labeled minibody, and was scanned 23 hours post injection. Image courtesy of Dr. Anna M. Wu.

microPET/microCT



A fused [^{18}F]FHBG PET/CT image shown using the Microsoft Windows version of AMIDE. The PET date was acquired on a Concorde Focus while the CT data was acquired on an ImTek microCAT. Fiducial markers used for alignment are visible externally to the mouse. Image courtesy of Dr. Christophe Deroose.

Volume Rendering



A stereoscopic volume rendering of a fused [^{18}F]FDG PET and CT data set. The PET scan is from a microPET P4, the CT is from an ImTek microCAT. Alignment was accomplished using fiducial markers (not shown).

VALIDATION

To validate the program's ROI statistics, similar ROI's were drawn in AMIDE, CTI's Clinical Applications Programming Package (CAPP), Mediman [7], MRIcro, and CRIIISP, an IDL based image package developed previously in our laboratory. The results between AMIDE and the other packages were not significantly different when compared using a two tailed paired t-test at a significance level of $p<0.2$. Values are expressed as 1000$*$Mean Image Units$\pm$SD.

	AMIDE	CAPP	Mediman	MRIcro	CRIIISP
cylinder	0.46 ± 0.093	0.46 ± 0.097	0.46 ± 0.10	0.47 ± 0.084	0.47 ± 0.096
heart	5.5 ± 0.73	5.9 ± 0.55	5.6 ± 0.83	5.2 ± 0.58	5.2 ± 1.0
brain	1.9 ± 0.19	2.0 ± 0.10	1.9 ± 0.18	1.8 ± 0.18	1.9 ± 0.13
bladder	48 ± 14	50 ± 11	45 ± 14	45 ± 11	47 ± 14

AVAILABILITY

All required source code and installation instructions can be found through the AMIDE web site (amide.sf.net), along with binaries for Linux/i386, Macintosh OS X [Apple Computer, Inc., Cupertino, CA], and Microsoft Windows [Microsoft Corporation, Redmond, WA] platforms.

CONCLUSION

AMIDE has been developed as a user friendly tool for performing multimodality medical image analysis. The program's main advantages compared to pre-existing software tools stem from it being designed explicitly for the purposes of multimodality imaging. For instance, the program implements a non-destructive, automatic reslicing approach for data viewing that makes shift, rotation, and scaling operations on data sets trivial to perform, an essential facility when working with previously unregistered data. An additional feature for working with multimodality data is the tree based interface, which allows the user to easily manipulate multiple data sets and ROI's.

Another advantage of the software is the free licensing (GPL). It is hoped that other groups will seize upon the availability and extensibility of the package's source code, and choose AMIDE as a platform upon which their ideas and algorithms can be readily disseminated to the research community as a whole.

The core functionality of the program is now complete, and the software has demonstrated its capabilities in a variety of research studies in our and other laboratories. Future work on the software will focus on extending the number of interactive "wizard" interfaces for medical imaging algorithms, such as factor analysis and cardiac polar maps.

References

[1] Loening AM, et al. *Mol Imaging*, 2, 2003

[2] Nolf E, et al. *Eur J Nucl Med*, 30:S, 2003. URL `http://xmedcon.sf.net`

[3] Free Software Foundation, Inc. . GNU general public license, Version 2, 1991. URL `http://www.gnu.org/copyleft/gpl.html`

[4] Lacroute P, et al. In *Computer Graphics Proceedings, Annual Conference Series*, pp 451–458, Orlando, Florida, Jul 1994. ACM SIGGRAPH. URL `http://graphics.stanford.edu/software/volpack`

[5] Hill DLG, et al. 46:R1–R45, 2001

[6] Knuth DE.: *Sorting and Searching*, volume 3 of *The Art of Computer Programming*. Addison-Wesley, Reading, MA, 1973

[7] Coppens A, et al. *IEEE Trans Nucl Sci*, 40:950–955, 1993. URL `http://www.topo.ucl.ac.be/iv_mediman.html`

Acknowledgments

The authors would like to thank all the beta testers of the software, most notably Dr. Anna M. Wu, Dr. Christophe Deroose, and Helen Su, along with Zachary F. Walls for help in preparation of this poster. This work was supported in part by a Department of Defense NDSEG Fellowship (AML), NIH MSTP training grant GM08042 (AML), the Aesculapians Fund of the UCLA School of Medicine (AML), Department of Energy Contract DE-FC03-87ER60615 (SSG), and NIH grants P50 CA 86306 (SSG), R0-1 CA82214 (SSG), and SAIRP R24 CA92865 (SSG).