

# **AMIDE: a free software tool for multimodality medical image analysis**

**Andreas Markus Loening<sup>1,4</sup> and Sanjiv Sam Gambhir<sup>1,2,3,5</sup>**

<sup>1</sup>The Crump Institute for Molecular Imaging

<sup>2</sup>UCLA-Jonsson Comprehensive Cancer Center

<sup>3</sup>Department of Biomathematics

UCLA School of Medicine, Los Angeles, California

and

<sup>4</sup>Department of Bioengineering

University of California, Los Angeles, California

and

<sup>5</sup>Department of Radiology, Bio-X Program

Stanford University, Stanford, California

**Running title:** Multimodality Image Analysis

**Keywords:** Multimodality, Image Fusion, Registration, Image Analysis, Image Quantitation

**Please address all correspondence to:**

Sanjiv Sam Gambhir, M.D., Ph.D.

UCLA School of Medicine, B3-399 BRI

700 Westwood Plaza, Los Angeles, CA 90095-1770

Email: [sgambhir@mednet.ucla.edu](mailto:sgambhir@mednet.ucla.edu), [sgambhir@stanford.edu](mailto:sgambhir@stanford.edu)

Phone: 310-206-1798

FAX: 310-206-8975

**Nonstandard Abbreviations:**

[<sup>18</sup>F]FDG [<sup>18</sup>F]-fluoro 2-deoxy-glucose

ROI Region of Interest

%ID/g Percent Injected Dose per gram tissue

SUV Standardized Uptake Value

GPL GNU General Public License

## Abstract

AMIDE (Amide's a Medical Image Data Examiner) has been developed as a user friendly, open source software tool for displaying and analyzing multimodality volumetric medical images. Central to the package's abilities to simultaneously display multiple data sets (e.g. PET, CT, MRI) and regions of interest, is the on demand data reslicing implemented within the program. Data sets can be freely shifted, rotated, viewed, and analyzed with the program automatically handling interpolation as needed from the original data. Validation has been performed by comparing the output of AMIDE with that of several existing software packages. AMIDE runs on UNIX, Macintosh OS X, and Microsoft Windows platforms, and is freely available with source code under the terms of the GNU General Public License (GPL).

### Nonstandard Abbreviations with Definitions:

**[<sup>18</sup>F]FDG** [<sup>18</sup>F]-fluoro 2-deoxy-glucose - A fluorinated glucose analog used for studying glucose metabolism.

**ROI** Region of Interest - A subset of data in which the researcher is interested. Statistics are generally calculated for such a region.

**%ID/g** Percent Injected Dose per gram tissue - A semiquantitative measure. The %ID/g relates the activity concentration of an ROI normalized by the injected dose.

**SUV** Standardized Uptake Value - A semiquantitative measure. The SUV is the concentration of activity in an ROI relative to the concentration of injected activity if it were distributed uniformly throughout the body.

**GPL** GNU General Public License - A software license designed to maintain the user's freedom to run, study, redistribute, and improve a program.

# 1 Introduction

In the molecular and medical imaging community today, there is a paucity of software tools available for volumetric image analysis that are freely available, modifiable, and relatively feature complete. While a number of packages exist, few of these packages encompass all the features that a researcher may desire, and only a subset of these can be freely modified by the researcher for her or his needs (see Table 1). For a researcher wishing to do multimodality image analysis, the choices are further constrained as the majority of packages are restricted to strictly orthogonal or planar processing of data sets. This particular limitation will become more pronounced as the role of multimodality imaging increases in importance [1].

In light of this need, AMIDE (Amide's a Medical Image Data Examiner) has been developed to provide the research community with a relatively full-featured, freely available, and open source solution for single and multimodality volumetric medical image analysis. AMIDE, licensed under the GPL [2], is freely modifiable and redistributable, and is not dependent on any proprietary underlying packages.

In addition to being open source, AMIDE is unique in that it has been designed to avoid specific constraints of previous software packages. Data sets (e.g. PET, CT, MRI) and regions of interest (ROI's) are logically organized within a tree structure so that an unlimited number of these items can be displayed, modified, and analyzed simultaneously. Furthermore, data sets in AMIDE are not restricted to processing along orthogonal directions. Instead, information is continuously interpolated as needed from the original data to allow for non-destructive and non-orthogonal reslicing of anisotropic data sets. This ability facilitates manual alignment and fused viewing of multiple medical images within an AMIDE session, and allows for seamless handling of data sets with differing voxel sizes and dimensions.

Another key design goal of AMIDE was to avoid encumbering it with an overly complex user

interface. With the recent extension of medical imaging modalities into the realm of small animal research (e.g. microCAT, microPET), there has been a steady increase in the number of basic science researchers using these technologies who have not trained in medical imaging science. One of the major hurdles encountered by these researchers has been negative experiences with existing software packages. With this in mind, development has aimed at providing a consistent and intuitive interface for the casual research user. As one step in this process, AMIDE abstracts away the underlying digital representation of the medical data set whenever possible. For instance, the user is not presented with a fixed image plane and voxel based dimensions. Instead, slices of data are automatically extracted from the volumetric data sets at any user specified angle and thickness. Additionally, dimensions are handled in terms of real world units, and image units and statistics can be presented in terms of Percent Injected Dose per gram tissue(%ID/g) or Standardized Uptake Value (SUV) metrics.

AMIDE provides a variety of additional features useful to the molecular imaging researcher, including fully three dimensional ROI drawing and analysis for static and dynamic images, two and three way linked viewing (dual cursor mode), rigid body registration using fiducial markers, filtering and cropping of data sets, movie generation, series viewing, and volume rendering.

## **2 Description**

### **Underlying Concepts**

The data hierarchy within AMIDE is built around a tree abstraction composed of a succession of objects such as data sets and ROI's (described below). Conceptually, any object type can be the child of any other object type, although not all pairings are necessarily logical. Generally, data set objects will be the children of the study object, and ROI objects will be the children of either the

study object or a specific data set object. The tree based hierarchy allows operations performed on an object, such as shifts and rotations, to be successively mapped down to all of that object's children.

The following object types have been implemented in AMIDE:

**Study** The root object in AMIDE, this object is used for grouping a set of related medical images and ROI's into a logical unit, and keeps track of parameters that affect the whole study.

**Data Set** Used for encapsulating volumetric medical images, this object contains the raw image data along with information needed for interpreting that data (voxel sizes, color table, thresholds, patient weight, injected dose, calibration factors, etc.).

**ROI** Region of interest objects specify a volume of space over which statistics are to be calculated. Currently implemented ROI's are ellipsoids, boxes, cylinders, and isocontours (2D or 3D).

**Fiducial Marker** Fiducial reference markers encode only a location in space and are used for rigid body registration of data sets.

Each object in AMIDE is assigned its own Euclidean space, and the location of this local coordinate frame is defined with respect to the global coordinate frame. When information from one object is needed by another object, AMIDE automatically handles the requisite affine (linear plus translation) transformations between the spaces, as shown in Figure 1. This approach allows the rotation or movement of a data set object to be accomplished by a simple alteration of the parameters specifying the object's local coordinate frame, rather than the destructive reslicing of the image data.

## Implementation

The C programming language was chosen for the coding of AMIDE for several reasons, the three most important being the general familiarity of most researchers with this language, the ready availability of high performance C compilers on current operating systems, and the desire to avoid burdening the program with requirements on underlying proprietary packages such as Matlab or IDL. The specific compiler used in this work was GCC (GNU Compiler Collection, [gcc.gnu.org](http://gcc.gnu.org)).

Version 2 of the GTK+/GNOME toolkit ([www.gnome.org](http://www.gnome.org)) was used for the user interface and object model. This toolkit was chosen for a combination of portability, a C language interface, and free licensing. Much of the core functionality of AMIDE has been written as an extension to this toolkit in order to provide a convenient interface for using AMIDE functionality in separate pieces of code.

Raw data in AMIDE is stored in the data format (8/16/32 bit integer, 32/64 bit float) and with the voxel size (isotropic or anisotropic) of the imported data file. The original data is never altered, rather, the program interpolates directly from the original data set as needed. This approach makes data set movements, scalings, and rotations computationally trivial as only the associated coordinate information is altered for these operations. The trade-off is that slice viewing is computationally more expensive compared to standard orthogonal data viewing. Zero order (nearest neighbor) and first order (trilinear) interpolation algorithms have been implemented for speed and image quality, respectively. Higher order interpolation methods [3] have not been employed since successive interpolations are never performed and because these interpolators become computationally prohibitive in three dimensions.

AMIDE saves studies in an XML (eXtensible Markup Language) based directory format. Each object's parameters and data format information are saved in a text file, with the raw image data saved as a separate binary file. This approach allows data files to be easily viewed and manipulated

externally to the program, guards against endian incompatibilities (incompatibilities due to the inconsistent ordering of stored data between different processor architectures), and makes backward and forward file compatibility easy to maintain between different versions of the program.

Data importing is done primarily through the (X)MedCon image conversion library [4], a software project that provides image reading and conversion between many of the more commonly used medical image formats. Currently supported file formats include DICOM 3.0, ECAT 6.4/7.2, Acr/Nema 2.0, Analyze (SPM), InterFile 3.3, and Concorde. Additionally, raw data importation in big, little, and PDP endian formats is handled natively for both integer and floating point data.

Volume rendering in AMIDE is performed using the VolPack [5] volume rendering library, which accelerates rendering using a shear-warp factorization algorithm. This software library based approach is portable and provides for true volume rendering capability, as opposed to the surface rendering approaches provided by many libraries and hardware accelerators. Series of rendered images, along with series of slices (“fly throughs”), can be encoded into MPEG-1 video files using the fame MPEG encoding library (fame.sf.net).

Rigid body registration is implemented inside of AMIDE through the use of fiducial reference markers and the Procrustes rigid body alignment algorithm without scaling [6]. Briefly, the transform needed to minimize the least squares error between a set of fixed and a set of movable fiducial marks is calculated. This transform is then applied to the coordinate space of the data set to be aligned.

Filtering is implemented using a “wizard” interface. Currently, Gaussian and median filters have been implemented, although any finite impulse response filter would be a trivial extension. Finite impulse response filters are implemented using an overlap+add method with a  $64^3$  point fast Fourier transform. Median filters are of variable kernel size, and can be run as separable 1D or a single 3D filter. In the interest of algorithmic simplicity, spatial coherence is ignored and the median filter is implemented using a partial sort median finding algorithm [7].

## Validation

The validation animal data set, consisting of PET and CT scans, was acquired as follows: A nude mouse (Charles River Laboratories), anesthetized with pentobarbital, was injected with 200  $\mu\text{Ci}$  [ $^{18}\text{F}$ ]-fluoro 2-deoxy-glucose ([ $^{18}\text{F}$ ]FDG). One hour was allowed for tracer uptake and clearance. The mouse was then placed on a plastic bed and 4 fiducial reference markers were affixed. Fiducial markers consisted of 200  $\mu\text{l}$  PCR tubes containing 1  $\mu\text{Ci}$  of [ $^{18}\text{F}$ ]FDG and 10  $\mu\text{l}$  Omnipaque (iohexal) nonionic iodinated contrast solution. The mouse was scanned on a microPET scanner built at UCLA [8] using 7 bed positions at 4 minutes/bed. Immediately after, a two bed position CT scan with 196 views/bed was acquired using an ImTek microCAT scanner [9] with the X-ray tube at 50 KVp, 300 mA, and 1.0 mm Aluminum filtration. All animal care and euthanasia was performed with the approval of the University of California Animal Research Committee. The validation cylinder consisted of a 37 mm diameter polysulfone cylinder filled with 262  $\mu\text{Ci}$  of [ $^{18}\text{F}$ ]FDG in 70 ml water, and was scanned in a single bed position for 4 hours.

MicroPET scans were reconstructed using the MAP reconstruction algorithm [10] with a beta value of 0.5, and multiple beds were combined into a single image. MicroCAT scans were reconstructed using the company supplied 3D-filter back projection reconstruction software. ImTek's file format was converted to ECAT 6.4 format using the `imtk_conv` program supplied with (X)MedCon, and the two beds were combined into a single image. The resolutions of the data sets were 1.5 mm and 0.4 mm for the PET and CT, respectively.

After loading the data sets into AMIDE, the four fiducial reference markers were located for each of the two scans, and the data sets were aligned using a rigid body alignment. Results of this alignment are shown in Figures 3 and 4. The calculated fiducial reference error for this alignment was 0.2 mm/reference point.

For validation of the ROI statistics, similar ROI's were drawn in AMIDE, CTI's Clinical Ap-



plications Programming Package (CAPP), Mediman [11], MRIcro, and CRIISP, an IDL based image package developed previously in our laboratory. The results are shown in Table 2. The values generated by AMIDE were not significantly different from any of the other packages when compared using a two tailed paired t-test at a significance level of  $p < 0.2$ .

## **Availability/Requirements**

All required source code and installation instructions can be found through the AMIDE web site ([amide.sf.net](http://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. The Macintosh version relies on the freely available Fink ([fink.sf.net](http://fink.sf.net)) add-on distribution.

There are no strict hardware requirements for running AMIDE, but the computer will need to have at least as much memory available as the cumulative size of the loaded data sets. Additionally, since the continuous reslicing approach is computationally expensive, a modern processor ( $\geq 500$  MHz) is recommended when working with larger ( $512^3$ ) data sets. Representative computational times needed for various functions are shown in Table 3.

## **Walk-through**

Figure 2 points out the most important elements of the AMIDE user interface. A brief walk-through follows to explain how to use the program.

1. On program startup, the user is presented with a blank study, and can then import the requisite medical images into the program. Most medical image formats are correctly detected by the program (e.g. DICOM, Analyze, Concorde, ECAT 7, InterFile). If the format cannot be determined, the user is allowed to explicitly specify the import file type. After import-

ing data, the entire study can be saved in AMIDE's native file format for direct loading in subsequent sessions.

2. The data sets will now appear as objects in the study's tree. Left clicking on an object will select it for appearance in the three orthogonal views, and a check will appear in the corresponding checkbox. Right clicking on any object in the tree will bring up a dialog for changing parameters relevant to that object, such as voxel sizes, scale factors, and thresholds. Note that one of the data sets in Figure 2 is highlighted, which indicates that this data set is the "active" object. When operations are performed that can logically apply to only one data set, the active object is the one chosen. For instance, pressing the thresholding tool button will bring up the thresholding dialog for the active object. The middle mouse button can be used for switching the active object.
3. Context sensitive help is displayed in the lower left corner of the application to explain what the different mouse buttons and key strokes will do at any given point. Complete documentation is also available from the help menu.
4. Moving through the data set is accomplished by directly clicking on any of the orthogonal views. For instance, clicking on the transverse view will update the coronal and sagittal views to correspond to the chosen point. In Figure 2, all three orthogonal views are shown, although the viewer can select fewer views by toggling the view selector buttons on the toolbar. The active data set can be shifted or rotated with respect to the other objects in the study by using the shift key together with the left or middle mouse buttons, respectively.
5. The zoom and the thickness of the viewed slices can be altered by using the corresponding entries on the toolbar. For dynamic studies, clicking the "frame selector" button will pop-up a dialog for frame selection purposes. Also on the toolbar is a set of toggle buttons for switching between single and multiple (linked) cursor mode viewing. An example of three

cursor mode is shown in Figure 3.

6. ROI's are added to the study either from the menu, or by directly clicking in the tree. After an ROI has been added to the tree, the next mouse click on any of the views will initiate the process of drawing an ROI, with the left button initiating edge-to-edge drawing and the middle button initiating center-out drawing. Subsequent modifications of the ROI can be done by clicking on the ROI in any of the views. Shifting, rotating, and scaling are accomplished by the left, middle, and right mouse buttons, respectively. Statistics for ROI's are generated by selecting "ROI Statistics" underneath the tools menu.
7. Underneath the view menu are options for generating series of slices and volume renderings of the currently selected data sets. Both of these options will pop up separate windows for the corresponding purpose. Series of slices can be displayed over space or time. From the volume rendering dialog, animated movies can be generated and saved as MPEG-1 files. An example of a stereoscopically rendered fusion data set generated inside AMIDE is shown in Figure 4.
8. Entries under the tools menu start wizards for various functions such as filtering, rigid body alignment, cropping of data sets, and fly through movie generation. Examples of both fly through and rendered movies can be found at the AMIDE web site ([amide.sf.net/output.html](http://amide.sf.net/output.html)).

### **3 Discussion**

With the increasing prevalence of multimodality imaging in the research community, a need has arisen for new, more sophisticated software tools that can handle and analyze these increasingly complex data sets. AMIDE provides such a tool, and AMIDE's capability for manipulating multiple, non-orthogonal data sets will become increasingly critical as multimodality image analysis

becomes more common.

While a handful of proprietary tools exist that provide relatively comparable feature sets, to our knowledge, AMIDE is the only freely available and open source software package in its class. Since the source code is available, researchers are not only free to use the program but can also study and expand upon the program as fits their needs and interests. Furthermore, as the code is unencumbered by restrictive licensing users are free to redistribute both the code and any modifications made to it, although modified versions must be appropriately marked as such.

For the novice user, additional advantages of AMIDE are the simplified interface and unit handling. Continuous beta testing and feedback over the last two years from three basic science researchers with minimal imaging experience has been incorporated into the development of the user interface in order to make program interaction as intuitive as possible. Units in the program are, whenever practicable, specified in terms of real world values, and the underlying digital representation of the data is, to a great extent, divorced from the user. For instance, in the slices viewed from the data set, the thickness is not restricted to integer multiples of the voxel size. As another example, given the correct conversion constants the program can present data and statistics to the user directly in terms of %ID/g or SUV's.

The continuous reslicing approach adapted by AMIDE has proven itself to be flexible from a development aspect and crucial for the arbitrary image fusion abilities of the package. It makes movement, scaling, and rotation of data sets essentially free from a computational standpoint while avoiding destructive interpolation of the original data set. The trade-off is that the computational expense of slice generation is greatly increased compared to an orthogonal slice based approach. In practice, it has been found that modern processors ( $\geq 500$  MHz) are powerful enough that the added computational expense of this approach does not impact the user experience for standard sized data sets ( $\leq 512^3$ )

AMIDE now encompasses the core set of features needed for bringing multi-modality medical

image analysis to the molecular imaging research community. Further work is shifting towards extending upon these core facilities, particularly in providing interactive “wizard” interfaces for making advanced medical imaging algorithms (e.g. factor analysis, cardiac polar maps) more accessible to the casual research user. It is hoped that not only will the package be a valuable addition to the molecular and medical imaging software toolkit, but that other research 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 molecular and medical imaging research community as a whole.

## 4 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. Dr. Gobalakrishnan Sundaresan aided with the acquisition of the PET/CT data sets used in the examples.

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).

## References

- [1] Massoud TF and Gambhir SS (2003), Molecular imaging in living subjects: seeing fundamental biological processes in a new light. *Genes Dev* **17**, 545–580.
- [2] Free Software Foundation, Inc (1991), GNU general public license, Version 2. URL <http://www.gnu.org/copyleft/gpl.html>.

- [3] Thévenaz P, Blu T, and Unser M (2000), Interpolation revisited. *IEEE Trans Med Imaging* **19**, 739–758.
- [4] Nolf E, Voet T, Jacobs F, Dierckx R, and Lemahieu I (2003), (X)MedCon - an open-source medical image conversion toolkit. *Eur J Nucl Med* **30**, S246, URL <http://xmedcon.sf.net>.
- [5] Lacroute P and Levoy M (1994), Shear-warp factorization of the viewing transformation. In *Computer Graphics Proceedings, Annual Conference Series*, ACM SIGGRAPH, Orlando, Florida, pp. 451–458, URL <http://graphics.stanford.edu/software/volpack>.
- [6] Hill DLG, Batchelor PG, Holden M, and Hawkes DJ (2001), Medical image registration. *Phys Med Biol* **46**, R1–R45.
- [7] Knuth DE (1973), *Sorting and Searching*, vol. 3 of *The Art of Computer Programming*. Addison-Wesley, Reading, MA.
- [8] Chatziioannou AF, Cherry SR, Shao Y, Silverman RW, Meadors K, Farquhar TH, Pedarsani M, and Phelps ME (1999), Performance evaluation of microPET: a high-resolution lutetium oxyorthosilicate PET scanner for animal imaging. *J Nucl Med* **40**, 1164–1175.
- [9] Paulus MJ, Sari-Sarraf H, Gleason SS, Bobrek M, Hicks JS, Johnson DK, Behel JK, Thompson LH, and Allen WC (1999), A new X-ray computed tomography system for laboratory mouse imaging. *IEEE Trans Nucl Sci* **46**, 558–564.
- [10] Qi J, Leahy R, Cherry S, Chatziioannou A, and Farquhar T (1998), High-resolution 3D bayesian image reconstruction using the microPET small-animal scanner. *Phys Med Biol* **43**, 1001–1013.

- [11] Coppens A, Sibomana M, Bol A, and Michel C (1993), Mediman: an object oriented programming approach for medical image analysis. *IEEE Trans Nucl Sci* **40**, 950–955, URL [http://www.topo.ucl.ac.be/iv\\_mediman.html](http://www.topo.ucl.ac.be/iv_mediman.html).

## Figure Legends

- 1 Diagram of coordinate transforms done by AMIDE. Each object in AMIDE (1) is defined with respect to
- 2 Salient user interface elements of AMIDE. A standard AMIDE session is shown, with the most importa
- 3 Main window of AMIDE shown in three cursor mode with two aligned data sets loaded and displayed o
- 4 Example of fused data sets rendered stereoscopically by AMIDE using the VolPack volume rendering li



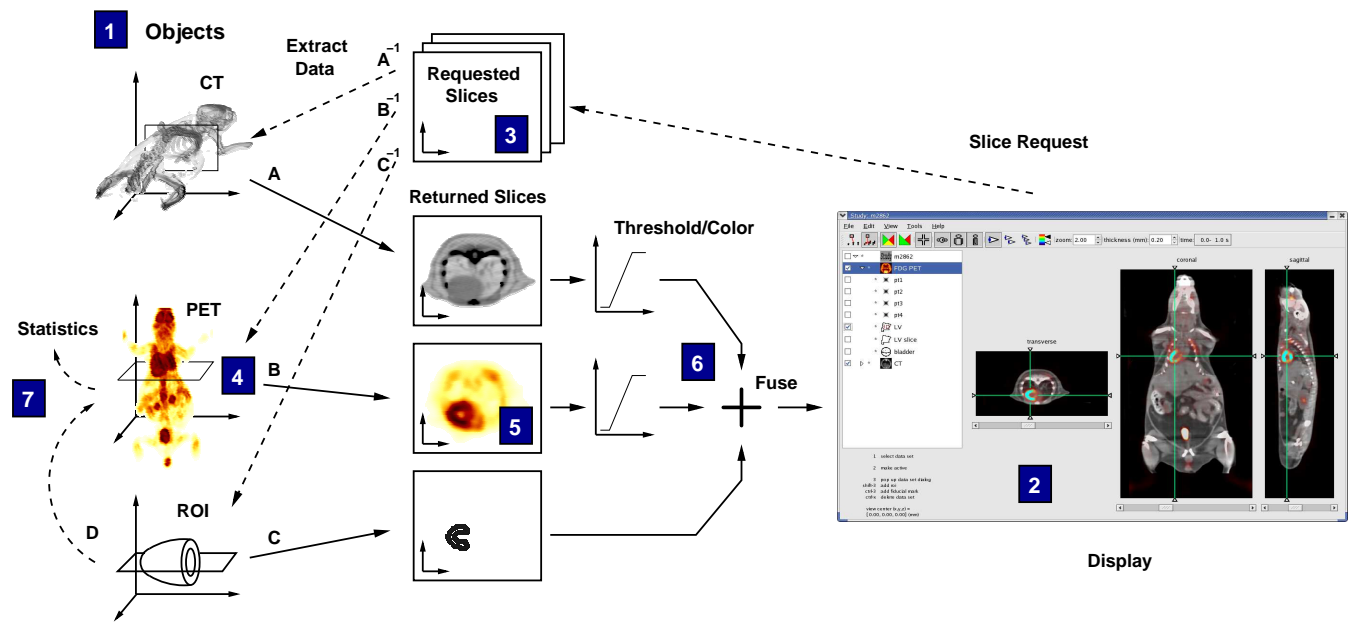


Figure 1:

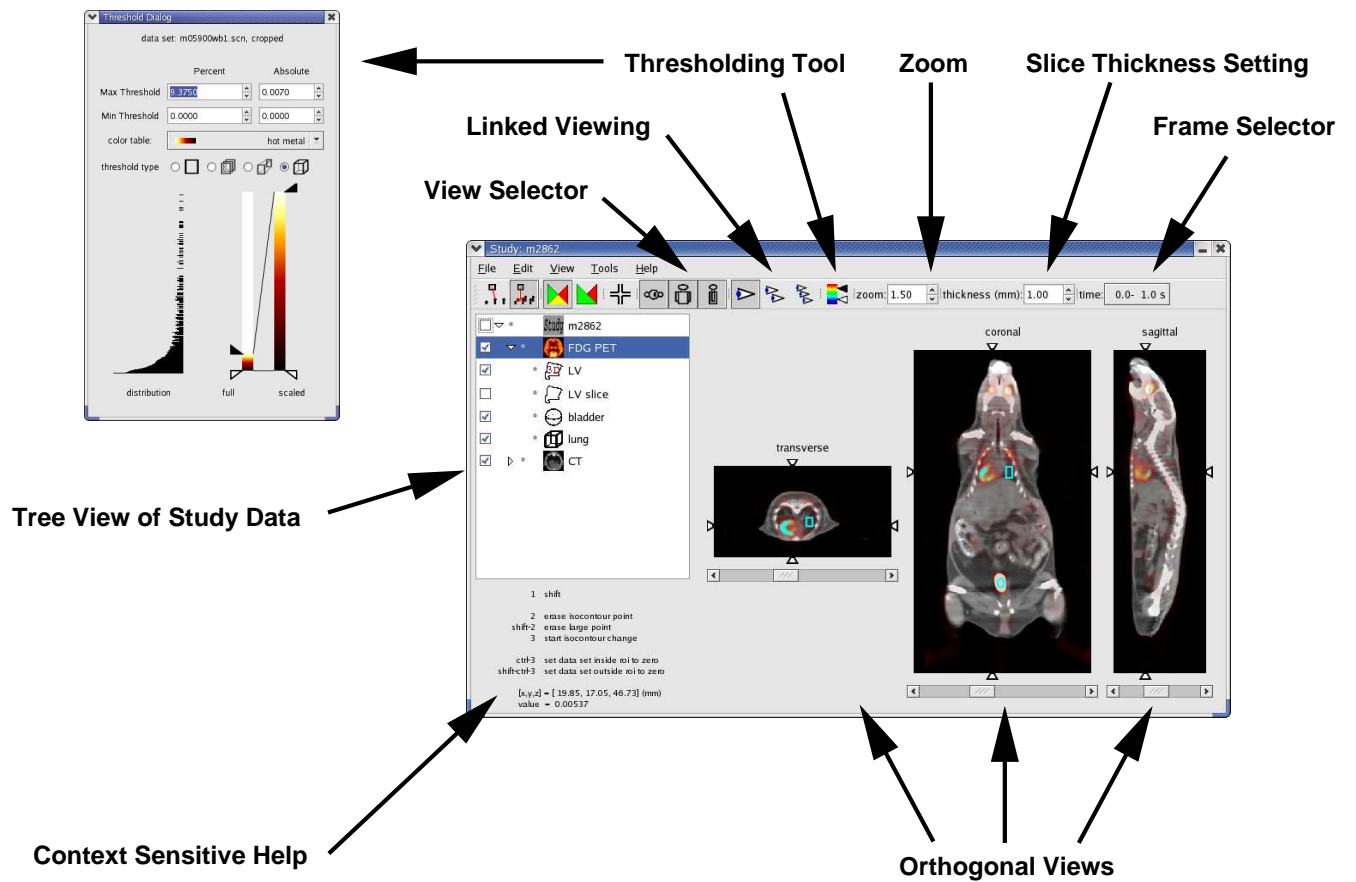


Figure 2:

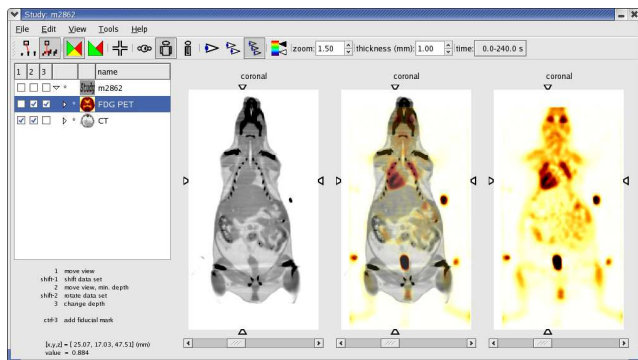


Figure 3:

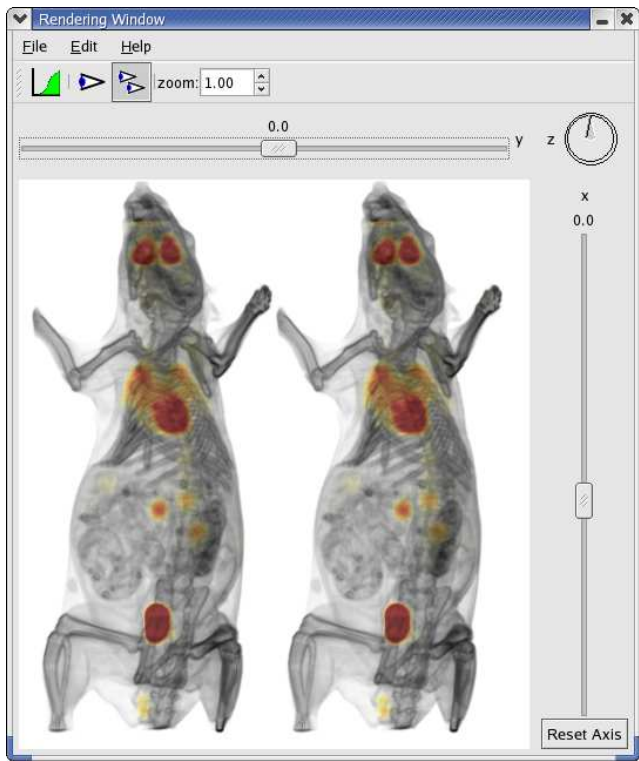


Figure 4:

## Table Legends

- 1 A list comparing AMIDE with several available molecular imaging software packages. The “non-ortho
- 2 ROI statistics generated for similarly placed and sized ROI’s using five different image analysis program
- 3 Times needed for performing various functions in AMIDE listed for a 128x128x159 PET image on a 75

Package	Free	Source Code	Interface	Fusion	Platform Compatibility	URL
AMIDE	Yes	Open	Non-orthogonal	Yes	Windows, Mac OS X, Unix	amide.sf.net
CAPP	No	No	Orthogonal	No	Solaris	www.cti-pet.com/www/products.nsf/pages/ecat.htm
Hermes	No	No	Non-orthogonal	Yes	Solaris	www.nuclear-diagnostics.com/proc/processing.shtml
Mediman	Yes	No	Slice Based	No	Unix	www.topo.ucl.ac.be/iv_mediman.html
MIM	No	No	Non-orthogonal	Yes	Windows, Mac OS	www.zalen.com
MRicro	Yes	No	Orthogonal	Overlap only	Windows, Linux	www.psychology.nottingham.ac.uk/staff/cr1/mricro.html
NucMed Image	Yes	No	Slice Based	No	Mac OS	nucmed.sluh.edu/NucMed_Image/NucMed_Image.html
OSIRIS	Yes	300€	Slice Based	Limited	Windows, Mac, Unix	www.expasy.org/www/UIN/html1/projects/osiris/osiris.html
3D-Doctor	No	No	Orthogonal	Yes	Windows	www.ablesw.com/3d-doctor/3ddoctor.html
Syngo	No	No	Non-orthogonal	Yes	Windows	www.syngo.com

Table 1:

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

Table 2:

Function	Time (s)
Extract transverse slice of data	0.01
Extract coronal/sagittal slice of data	0.04
Calculate statistics for 7500 voxel ROI	0.3/frame
Initial setup for volume rendering	4.8
Volume rendering a data set	0.06

Table 3: