# Imaginer: a graphical user interface for automated analysis of emission images

**Carolyn T. Wick, Christine J. Dykstra, Ronald Harrop, M. Stella Atkins**

**Medical Computing Lab,**
**School of Computing Science, Simon Fraser University,**
**Burnaby, BC, Canada V5A 1S6**

## ABSTRACT

Imaginer is a graphical user interface currently being developed for automated analysis of emission images. It will be the first application to implement a new feature extraction method of contiguous volume analysis on an unlimited number of image formats. Its development was prompted by the desire to simplify the steps involved in that analysis and to improve visualization and interpretation of results through a graphical user interface. This paper discusses difficulties that have arisen in generalizing the method of contiguous volume analysis to work with an unlimited number of image formats, as well as in abstracting the visualization techniques to effectively represent all types of data used during analysis. Issues in creating a flexible, intuitive, and extensible user interface for scientific investigation and clinical use are discussed, along with several usability issues that have arisen during development. Prototypes of Imaginer and its software components are described. Designed for ease of use, flexibility, extensibility and portability, Imaginer will enable users to assess the appropriateness of this method of feature extraction for various clinical and research purposes, and to use it in contexts for which it is found to be appropriate.

**Keywords:** automated analysis, contiguous volume analysis, feature extraction, graphical user interface, emission images, prototype, image visualization.

## 1. INTRODUCTION

Three-dimensional (3D) functional images can contain a great amount of information about physiological activity in the living body. Unlike 3D anatomical images, whose cross-sectional slices display internal structure, images produced by modalities such as positron emission tomography (PET) and single photon emission computed tomography (SPECT) are not always easy to understand since less is known about the physiology they represent. Visual inspection is often performed as an initial means of interpreting such images. However, because visual inspection is prone to bias and inefficiency, an automated form of image analysis is often desirable. Dykstra's method of contiguous volume analysis attacks the problem of analyzing physiological data, and has several advantages over other techniques for extracting features of interest from 3D images.[1,2] Because it can be completely automated, it is non-subjective and reproducible. By using only the functional data itself, errors which might have been introduced with registration to an anatomical image are avoided. The method analyzes images in 3D, so slice positioning is less important and registration becomes less of a problem. Finally, the method further reduces bias by characterizing the entire image and not just regions of interest, or other data that is presumed before analysis to yield significant information.

Up until now, the Dykstra method has been used on only one type of image. It was rigorously tested on that type of image and was shown to be useful, especially in separating disease images from normals. We now strongly believe that other people could benefit from applying the method to their own images to see if it works as well for them. Interest has been expressed in using this method of feature extraction for locating foci of epileptic seizures, as well as for identifying patterns of activity characteristic to patients with Alzheimer's disease. However, for other people to use the method, we needed to implement a program that could perform the method on an unlimited number of image types. Also, since visualization of results is critical to analysis, the visualization techniques and graphics would have to be generalized so they could display a wider range of data,

0-8194-1779-3/95/$6.00

including results that we cannot yet anticipate. Further, to help users assess the appropriateness of this new method for clinical and research purposes, the program would have to be easy to use and flexible enough to enable users to modify parameters and view assumptions that affect its interpretation of images.

Imaginer (Image intensity examiner) is a program currently being developed to meet this need. It has been designed for flexibility, extensibility, portability, and ease of use. As well, human factors and usability issues were considered throughout the program's design, largely to help avoid potential misuse of the application for purely automated diagnostic purposes. This paper discusses the difficulties that have arisen so far in generalizing the method and in creating a flexible user interface for medical applications. Section 2 explains the method, while Sections 3 and 4 address design issues that arose in generalizing the method to work on an unlimited number of image formats. Section 5 then outlines several human factors related to Imaginer's design, usability and development, and describes one of Imaginer's prototypes.

## 2. METHOD OF ANALYSIS

Imaginer is primarily intended to be a tool for analyzing emission images using a method of contiguous volume analysis based on work by Dykstra.[1] In order to create an application for analyzing an unlimited number of image types using her method, the method had to be refined and abstracted for implementation and extension. This section explains the fundamental concepts central to Dykstra's method of analysis, but presents them in a more generalized form than in her thesis, breaking up the separate stages of analysis to reflect Imaginer's high-level design. Figure 1 displays the principal types of data used and generated

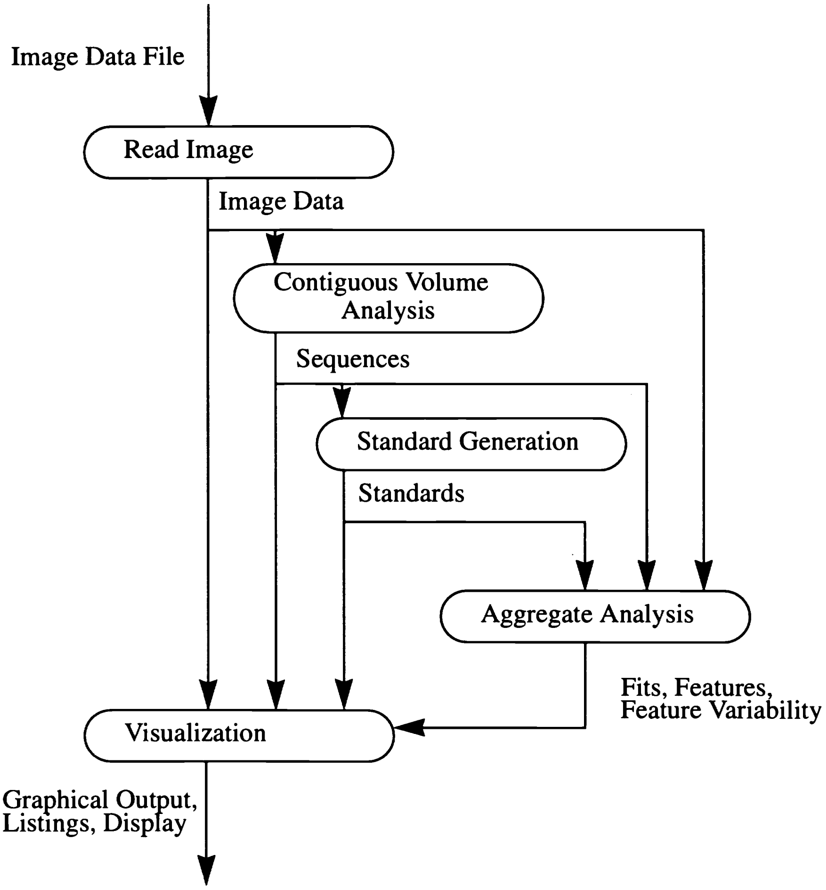


**FIGURE 1. Data and stages involved in analysis**

at each analysis stage and illustrates the many possible paths (represented by arrows) that analysts may choose to take. For example, image data could be read in and visualized immediately, or processed using the contiguous volume analysis method to obtain sequences characterizing the data as patterns of intensity. The method is independent of the implementation; each stage (represented as a bubble in the diagram), could be a software routine within a larger program, a stand-alone executable reading and writing input and output to files, or an entirely manual process. Imaginer will implement routines that automate most of the analysis stages and will integrate them under a single graphical user interface; the interface will not only make it easy for users to apply any aspect of the method to an image, but will also provide services for working with multiple images simultaneously.

## 2.1 Contiguous volume analysis

The method of contiguous volume analysis automatically identifies significant patterns of intensity levels within 3D image data sets that can be characterized as contiguous volumes of high activity. The method finds the positions, growth, and merge patterns of these volumes and stores the analysis results in a data structure called a sequences list. Definitions for voxels, volumes, and sequences, the three main terms associated with the method, are given in Table 1.

**TABLE 1. Definitions for terms associated with contiguous volume analysis**

Voxel: A voxel is a particular (x, y, z) coordinate in a 3D image formed from cross-sectional slices produced by medical imaging modalities such as PET and SPECT. Each voxel has a value associated with it. The z direction is taken to be the axial direction of the scanner.

Volume: A contiguous volume is a largest connected set of one or more voxels whose values are all at or above a given threshold, where all voxels are connected by at least one vertex to the rest of the volume. Each voxel of an image is contained in at most one volume for a given threshold.

Sequence: A sequence is a nested set of volumes which, as the threshold is dropped, do not merge with any other significant (non-noise) volumes. Therefore, a sequence may be either a tip-sequence representing a hot spot of activity which grows and remains distinct through any number of threshold changes (including none) and ends when a non-noise sequence merges with it, or, it may be a non-tip-sequence formed by the merging of two or more sequences that continues until it merges with another sequence of significant size. The resolution of the image and the size of a voxel are both taken into account when distinguishing between noise and significant sequences.

The contiguous volume analysis method works as follows. For a series of thresholds, from the scan maximum down to some minimum, all contiguous volumes above the threshold are found. Each of these volumes is characterized by size and centroid position, and is tracked through the threshold changes. As the threshold is decreased, the volumes grow and the regions begin to merge, until the entire image is contained in a single volume. The pattern of growing and merging volumes is stored as a list of sequences, which can be visualized in a number of different data displays, or used in forms of aggregate analysis.

## 2.2 Standard generation

Some sequences have shown spatial consistency across large data sets and form a pattern against which other images or groups of images can be compared. Standards store information about these sequences, and are generally used to define normal patterns of activity for certain data sets. A standard generated and used by Dykstra, for example, consisted of the centroids of the 14 longest (seemingly most significant) sequences that were found in the average of 10 normal PET $F^{18}$-fluorodeoxyglucose (FDG) images. These sequences were consistently found in a further 53 individual normal images, and when sequences of other brain images were compared against this standard, significant differences between normal and diseased images were found in how well they matched the standard. Defining appropriate standards for images is a complex process sensitive to several human factors; issues related to defining new standards for images are addressed in more detail in Section 4.3.

## 2.3 Aggregate analysis

Image data, sequences, and standards can be combined in several ways to produce additional data and forms of analysis. Three important forms of aggregate analysis which will be discussed in this section are: comparing sequences against standards, comparing multiple images and, defining feature variability across images.

### 2.3.1 Comparing sequences against standards

Researchers who have defined a standard pattern of activity may wish to compare images against that standard. For example, they may wish to look for a "normal" pattern in a suspected disease image, or test for a disease pattern in the same image and compare results. The sequences list provides a concise description of an image which can be automatically tested against a standard to determine how well an image matches a particular pattern. When the sequences of an image are compared against a standard, (which by our definition may contain a set of regions which themselves can also be called standards), features are found.

A feature of an image is the region of activity that has the best fit to a particular standard region. The criteria for finding the features of an image from its data, its sequences list, and a particular standard will vary according to the type of information recorded for the standard. Dykstra's standard simply stored centroids of typical contiguous regions of high activity. To find features, her algorithm first looked for all peaks of activity, where a peak was defined as being one or more tip-sequences sufficiently close together in the image to form a coherent hot spot. The peaks' centroids were then compared against a standard centroid. Goodness of fit was determined by how many voxels away a peak's centroid was from the standard centroid, and was dependent on the image's resolution, voxel size and slice separation. If no peak of activity existed with reasonable fit to a standard, then the amount of activity in the area where the peak was expected was measured and recorded to describe the feature.

In order to generalize the method of finding features so that other standards can be used requires making assumptions about the image attributes that will define a consistent pattern of activity for an unknown set of images. Section 4.3 outlines some of the issues involved in implementing a standard-independent feature finding algorithm for comparing the sequences of images against new standards.

### 2.3.2 Comparing multiple images

Although multiple images can be compared by visual inspection alone, the amount of information to comprehend is so great that even highly trained analysts risk the chance of overlooking or unjustifiably assuming significant differences or similarities between images. The definition of sequences as volumes of high activity provides an unbiased description of an image which can be used in comparisons with other images. Furthermore, sequences enable standards to be defined so that multiple images can be compared against one another with respect to that standard. Thus, from within Imaginer, in addition to visually comparing image slices, researchers can compare the number of sequences found in each image, analyze the locations of highest activity, and for each standard region of activity, examine each image's feature, comparing fits, relative activity levels, and feature locations.

### 2.3.3 Defining feature variability across images

After multiple images have been compared, one may want to define feature variability for a set of images to record the typical amount of variability in the location, growth rate, and merge pattern of a particular feature. One can then not only test if an image contains a standard pattern, but also examine how well an image matches the pattern relative to a large number of other images. All standard features may for example be found in a diseased image, but the activity level of a particular feature may be typically much lower relative to the rest of the image for a disease than for a normal. By defining typical feature variability for a set of normals or for a certain disease, features of any image can more easily be interpreted as being either reasonably similar or dissimilar to a standard, and any such dissimilarity may be found to be itself of a typical or atypical nature. Using feature variability, one can observe "normal" variance across images that match the standard pattern, and more easily detect images that fail to match well to the standard pattern. After defining typical feature variability for a class of images, researchers might discover a subset of images that tend to deviate from the standard pattern in a consistent way; they might then choose to attempt to define a new standard for that particular image type if the deviant images all correspond to a particular disease, age, dosage of pharmaceutical, or other known common attribute.

## 2.4 Visualization

Visualization is the final stage in analysis. Several ways of interpreting images and the results of contiguous volume analysis involve visual analysis that becomes practical with Imaginer. Graphical representations and detailed listings have been designed for each of the principal types of data used in analysis, including voxel values, volumes, sequences, standards, fits, features, and feature variability. Additionally, flexibility in enabling visualization routines to combine data was desired for simplifying interpretation and analysis. For example, overlaying on the image slices the sequences found during contiguous volume analysis can provide visual information about where volumes of activity are located and about their growth and merge patterns. Overlaying the centroid positions of standards on image slices enables images to be visually inspected at the location where standard activity is expected. Finally, overlaying the feature positions on image slices shows where peaks of activity were found to best match certain standards. Being able to visually analyze where features are found in addition to viewing where the standards are sought could help considerably in interpreting results, especially when results have been strongly influenced by sometimes unavoidable idiosyncrasies like severely tilted or rotated image data.

# 3. APPLYING THE METHOD TO NEW IMAGE FORMATS

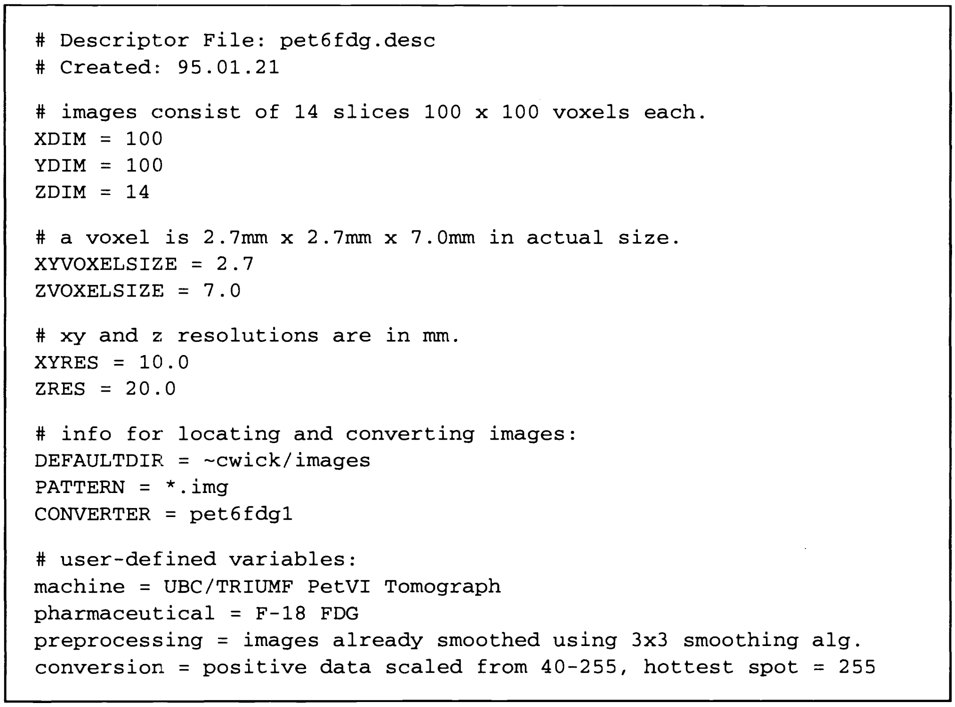
Dykstra's method of contiguous volume analysis was developed and tested on normal and diseased brains imaged using 14-slice $F^{18}$ FDG PET scans obtained on the UBC/TRIUMF PET VI tomograph[3]; generalizing it to work with other image formats, pharmaceuticals, machines and modalities will facilitate the use of the method to characterize additional types of images as patterns of spatially consistent hot spots that grow and merge.

Because not all image types are potentially appropriate for contiguous volume analysis, the range of formats that Imaginer is designed to work with has been restricted to three-dimensional physiological data. Nevertheless, the types of images potentially suitable for contiguous volume analysis can still vary in their slice widths and heights, voxel dimensions, machine resolutions, slice separation, number of slices, and so on--ways that can influence analysis results. The only common attribute of the images is that they can each be uniquely represented as a series of values, ordered by the (x, y, z) coordinate of the voxel to which they correspond. Because it would thus be impossible to predict all possible image formats that one might want to use with Imaginer, it was decided to have Imaginer understand only one basic image format and to have Imaginer convert a user's own images to this file format by accessing conversion utilities provided by a user for each unique image type. Descriptor files store information necessary for converting, reading, and processing the user's images. They are described below along with Imaginer's generic image file format and our conversion method.

## 3.1 Descriptor files

An image descriptor file is necessary for converting, reading and processing image data. The file specifies the conversion utility that will convert images of a particular type to Imaginer's generic image format, and gives the x, y, and z dimensions for an image, plus information about the machine's resolution and other parameters that are used during analysis. Since exactly one descriptor file needs to be created for a single image type, and because it is expected that at this stage users will generally only work with one image type at a time, the descriptor file for the type of image to be analyzed during a session can be loaded once and used for all images to be subsequently processed during that session of using Imaginer.

The format of a descriptor file is intuitive and extensible. Keywords identify the parameters that are essential to reading and processing converted image data, and specify information to help Imaginer speed up administrative tasks like converting image formats and locating images of this type. Additionally, users can define their own variables for describing the image type to help reduce the risk of a user inadvertently comparing images of incompatible types. All descriptor file information will be made apparent during analysis to help avoid misinterpretation of images. Figure 2 provides an example of a descriptor file.



```
# Descriptor File: pet6fdg.desc
# Created: 95.01.21

# images consist of 14 slices 100 x 100 voxels each.
XDIM = 100
YDIM = 100
ZDIM = 14

# a voxel is 2.7mm x 2.7mm x 7.0mm in actual size.
XYVOXELSIZE = 2.7
ZVOXELSIZE = 7.0

# xy and z resolutions are in mm.
XYRES = 10.0
ZRES = 20.0

# info for locating and converting images:
DEFAULTDIR = ~cwick/images
PATTERN = *.img
CONVERTER = pet6fdgl

# user-defined variables:
machine = UBC/TRIUMF PetVI Tomograph
pharmaceutical = F-18 FDG
preprocessing = images already smoothed using 3x3 smoothing alg.
conversion = positive data scaled from 40-255, hottest spot = 255
```

**FIGURE 2. Sample descriptor file**

## 3.2 Image files

Once an image has been converted to Imaginer's format, and if Imaginer knows the image's x, y, and z dimensions specified in the descriptor file for the image type, Imaginer can read the generic image file. A generic image file consists of a binary file of characters. Each character represents one voxel of the image by a threshold value from 0 to 255. The characters are ordered by the (x, y, z) coordinate of the voxel to which they correspond. The coordinates determine the position in the image and are ordered in the file such that x changes most rapidly (the file's first value corresponds to voxel (0, 0, 0), the second to voxel (1, 0, 0), etc.). The z value determines the slice number, where 0 is the superior-most slice. The y value determines the anterior/posterior position of a voxel; the anterior-most (top) row of voxels in each slice is numbered 0. The x value corresponds to the left/right position, where 0 is the left-most column.

## 3.3 Format conversion

The conversion program is a utility that a user can provide for each image format they wish to use with Imaginer. This program will read in original image data and convert it to Imaginer's generic image format described by an accompanying descriptor file. Imaginer can thus work with an unlimited number of image formats, so long as a conversion program and descriptor file is provided for each. When users wish to analyze a new image, Imaginer's interface allows them to specify the name and directory of the image as well as the image's type. Using information in the appropriate descriptor file, Imaginer will then automatically invoke a conversion program, passing it two arguments: the name of the original image file, and the name of a temporary file in which to place a copy of the image data in Imaginer's format. The temporary file is read into memory and the data is processed. The temporary file is then deleted so that duplicate image files do not waste precious disk space, effectively keeping only one copy of an image on disk at a time.

Users need to create or obtain conversion programs for each of their image types in advance of using Imaginer. In order to create a generic-format image file, the conversion program may uncompress image data, strip off any header information, reorder the data in the file, and scale or convert voxel values from their original representation to an 8-bit binary value between 0 and 255. For example, if the thresholds at each voxel are originally represented as floating point values between -1.0 and 1.0, positive radiation values between 0 and the scan maximum may be scaled during conversion from 40 to 255, with negative scan values falling below 40. In this example, 255 would always correspond to the hottest spot in an image and a particular voxel value after conversion would indicate an intensity relative to the rest of its image rather than an absolute intensity for direct comparison across multiple images. The details of conversion are up to the user, and are best described in the image type's descriptor file so that the information can be made apparent during analysis by Imaginer.

The advantages of our method of dealing with different image formats are threefold. First, Imaginer's use is in this way not restricted to a limited number of image formats. Second, users do not have to modify their original data files; rather, copies of images in Imaginer's format are automatically created for the user. Third, no duplicate copies of image data need to be stored on disk between sessions of using Imaginer. Additionally, being able to read an unlimited number of image formats helps make Imaginer more portable, flexible, and easy to use.

# 4. IMPLICATIONS OF A GENERALIZED ANALYSIS METHOD

Generalizing the analysis method to work with an unlimited number of image formats requires not only generalized routines for reading in image data, but also generalized routines for processing and displaying the data. Since we cannot make many assumptions in advance about the types of images that one might choose to analyze with Imaginer, and because we do not want to bias the program by making unjustifiable assumptions about what results the method will produce when applied to new image types, the image-processing and visualization routines become very complex. The generalized method for reading in images was described in Section 3. Four additional design issues that arise in generalizing the application to achieve a high level of flexibility are described below.

## 4.1 Constants and modifiable parameters

The algorithm developed by Dykstra for finding contiguous volumes in an image had originally been based on several justified assumptions about the image type being used. For example: the highest point of activity in each of her images always had a corresponding threshold value of 255 (after conversion); the total number of sequences found in an image never exceeded 600; volumes of fewer than 10 voxels in size could comfortably be disregarded as noise; and, the entire brain would always merge into a single sequence before the threshold had dropped as low as 90. These assumptions were reasonable for the type of image on which the method had first been developed. However, several of these assumptions are inappropriate for other image types.

In designing Imaginer, two decisions were made to circumvent this problem and help prevent incorrect analysis results: explicit constants would be avoided, and all parameters would be made apparent during analysis so that they could be overridden by the user should their calculated values turn out to be unreasonable for some particular image type.

One consequence of these decisions is that memory space for storing an image and its sequences, volumes, standards, features and other variable data is dynamically allocated. As well, the algorithm for finding contiguous volumes has been modified to work on images of any dimension, with any maximum and minimum threshold value. Expressions replace constants for calculating the size of noise, the z-scale factor (which is required to cope with the combined effects of slice separation and z-resolution on matches across slices), the measure of goodness of fit, and the linear tolerance used to determine when two sequences are part of the same coherent region of activity. These decisions all illustrate how the analysis routines were generalized to accommodate the flexible image format; however, the now unrestricted range of image types and data values has also affected the design of Imaginer's graphical user interface and visualization routines in other ways that were not as easily resolved.

## 4.2 Interface flexibility

Designing a flexible interface for entering and displaying data whose attributes are not always known in advance presents a challenge to developers of applications like Imaginer. Many issues on how to best maximize flexibility have focussed on the look-and-feel and design of the user interface. For example, without knowing in advance how many features might typically be sought in an image, each feature's attributes were better appended to a scrolling list box or text window rather than displayed in a separate text field or represented graphically within a small non-resizable, non-scrolling area of the screen. Such issues are addressed in more detail in Section 6.1, where examples of Imaginer's prototypes are also provided.

One set of issues that were not easily resolved concerned the best way to visually represent image data and analysis results. Dykstra's thesis describes some graphics that were found to be useful for visually representing the sequences and features found in her images.[1] Many aspects of those techniques relied on assumptions about the image type with which she worked. Now, with other image formats possible, we had to design visual representations for a wider variety of analysis data.

Consider the Feature Analysis (FA) graph displayed in Figure 3. This graph provides a visual representation of the image's sequences list, along with information about how well the sequences compared to 14 standard centroids.

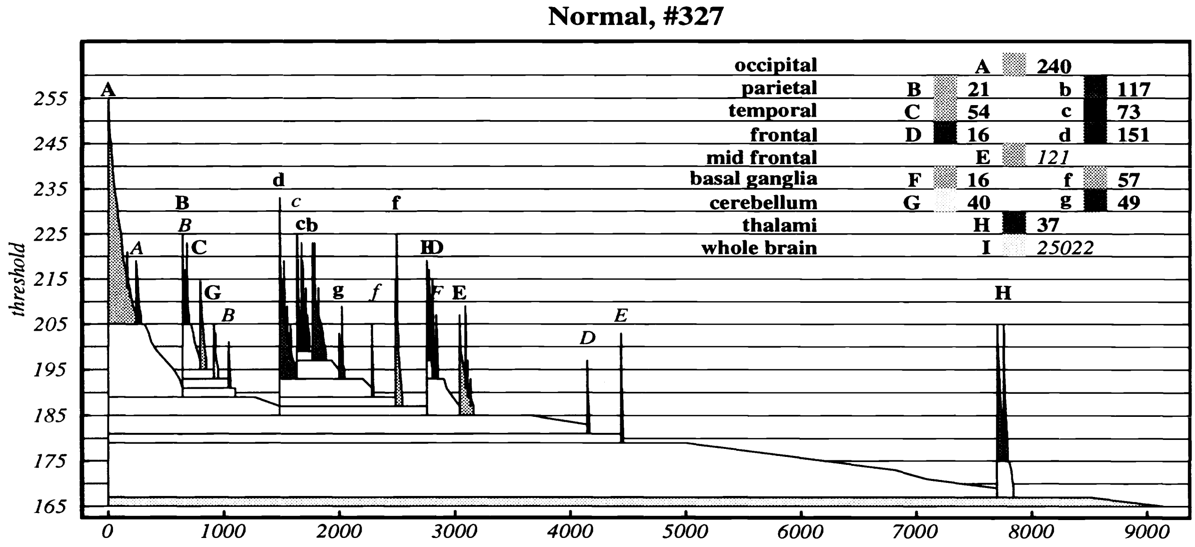


**FIGURE 3. Sample feature analysis graph**

The FA graph's horizontal axis provides a scale for measuring the size (in voxels) of the volumes making up each sequence, while the vertical axis indicates the threshold values at which they were found. Hot spots in the data are shown in the graph as uppermost points of polygons. As the threshold drops, the polygons increase in width and join with other polygons to form new ones. The polygons represent the tip- and non-tip-sequences found during contiguous volume analysis of the image. The width of a polygon at a particular threshold value represents the size of the sequence's volume whose voxels are all at or above the threshold. The horizontal positioning of polygons in the graph indicates the pattern in which the sequences merge. Each sequence's actual location in the image is not represented by the polygons themselves, although in Figure 3 the polygon labels and shading indicate which of 14 standard centroids a particular peak was closest to. Features of an image are labeled in bold, either above the sequences that formed the peak best matching the standard, or above an asterisk marking the sequence that grew to contain the standard region if no peak actually existed in the image near the standard centroid. Finally, the legend gives additional information about the features including fits, final sizes, and corresponding standard names.

Many difficulties surround the creation of readable, detailed FA graphs for new image types and standards. FA graphs for images containing a high number of sequences appear cluttered. Printing the summary of feature information in the upper-right corner of the graph itself may often be inappropriate when it interferes with polygons or labels or is simply too long to fit. When an image's sequences are compared against many standards, characters for labeling the matches may overlap or be ambiguous (for example, l, I, i, 1, c, C, o, O, 0). In seeking alternative visual presentations, colour was explored as a replacement for labels to help make the graph more readable; however, the number of distinct colours between which viewers could distinguish on a single graph was limited, and the colours in hardcopy reproductions often looked quite different from colours on the computer screen. Finally, a considerable amount of useful information about sequences or features has been omitted from this FA graph because of limitations in how much can be effectively displayed; centroids of sequences and exact fits between sequences and each standard are two examples of missing information that users may be interested in associating with the polygons of FA graphs.

In the end, the version of FA graph that we believe offers the most in flexibility and extensibility is an interactive graph which takes advantage of modern user interface techniques at the cost of showing less information on the displayed or printed graph itself. Our interactive FA graph has been designed to appear like Figure 3 but without any shading or labels except for the unambiguous identification of the features found in the image. Zoom and pan options would help the user view only the details in which they were interested. A mouse or other interaction device would enable users to select any polygon and view all of its attributes (its maximum and minimum threshold, final size, centroid position, standard name that it best matched, and so on). Some of those attributes could even be displayed directly on the slice image; when a user selects a sequence by clicking on its polygon in the FA graph, Imaginer could for example highlight the voxels in the slice image which the sequence contains. Such an interactive means of viewing FA graphs will be an integral part of Imaginer's final graphical user interface. Similarly, interactive visualization techniques will help resolve difficulties in generalizing other types of graphs and data displays. Mock-ups of the interface are described in Section 5.1.

## 4.3 Standard-independent feature finding algorithms

What details will people want to consider when defining new standards? Centroids may not be enough to represent the standard pattern of activity for a particular image type or disease; the relative activity level of a hot spot as compared with the rest of the image, the order in which regions merge, the final size of a hot spot before it begins to merge with other regions--these are all details that may be significant when characterizing a particular activity pattern. Because we do not yet know which attributes will best define new standards, and because standard generation is a sensitive process (a poorly defined standard could dangerously lead to misinterpreting images as containing or failing to contain abnormal activity patterns), standard generation will initially remain a highly involved, manual process, rather than a largely automated one.

The decision not to fully automate the process of defining new standards will enable experienced clinicians to give essential input into determining what makes a good standard, and will help make the standard generation process more flexible. However, this flexibility in defining standards translates into additional difficulties in other stages of analysis. The algorithm that compares sequences against a standard needs to know what characteristics to search for to find features, and the technique of registering a standard to an image may need to be modified for different types of images and standards.

Until more standards have been defined and more clinicians have had a chance to give input into how images are best compared with standards, the feature-finding algorithm will have to be modified manually for each new standard. Once more people have become involved in generating standards, the matching routine might be implemented to use an algorithm that can be guided by textual specifications in a "standard definition file" so that recompilation of the program will no longer be necessary even when extremely unusual combinations of attributes are used to define a new standard. Users might define new standards from a fixed set of attributes that can be represented in a standard definition file, and might specify how images are to be compared against a standard by choosing from a predetermined set of algorithms and by ranking the attributes to specify how goodness of fit is to be determined. The feature finding algorithm would itself be independent of a particular standard, varying its method of finding features according to the information in the file. Such a standard-independent feature finding algorithm is among the future extensions proposed for Imaginer.

## 4.4 Registration

The Dykstra method deliberately did not use any scaling, interpolation or warping of original image data in the process of extracting features of interest from images. The nature of Dykstra's images (brain scans) enabled her to define the active 3D component of an image as being the last large significant sequence found before only high numbers of noise sequences merged to include the entire image in a single volume. The translational registration vector which mapped the centroid of this active part of the brain to a standard origin was found; images were then compared against the standard by registering the standard to the image using the inverse transform to find the appropriate placement of standard centroids on images. With new image types and standards, registration may or may not be needed. If a standard considered only the number of separate hot spots without bothering about their position in the image, then no registration would be necessary. Otherwise, the registration method could be extended to implement alternative methods of mapping the standards to an image. The important aspect of registration is that, to preserve the unbiased nature of Dykstra's method, translational or rotational registration vectors should be determined purely in terms of the sequences in the image and not through an attempt to find anatomical features from an image which is of functional and not anatomical type. The inclusion of registration information in the proposed standard definition file or the enabling of users to dynamically link in their own registration routines are two possible enhancements to Imaginer for attacking the generalized registration problem.

## 5. USABILITY FACTORS IN GUI DEVELOPMENT FOR MEDICAL APPLICATIONS

Time to learn, speed of performance, rate of errors by users, retention over time, subjective satisfaction--these are all usability issues that have been identified as being worthy of consideration[4] and which influenced the design of Imaginer. Some were addressed early in Imaginer's development through prototypes, described in Section 5.1. Others will be best addressed later when the program is fully functional to evaluate if indeed Imaginer's design minimizes the number of errors made and the time required to learn and perform tasks. The following subsections outline the prototyping process we used to develop Imaginer's specifications, along with two additional usability issues particularly relevant to medical applications: issues in automating analysis, and trade-offs in designing a system for at least two distinct types of use (medical research and clinical practice).

### 5.1 Iterative prototyping

To ensure that Imaginer would provide medical experts with the appropriate tools for interpreting images, prototypes of Imaginer were created to stimulate feedback early in Imaginer's development. Several separate prototypes were created. Each one further refined the features and functionality of the application, and proved invaluable in clarifying the program requirements. Some windows from one of our prototypes are shown in Figure 2. They were designed using Sun's DevGuide. These proto-

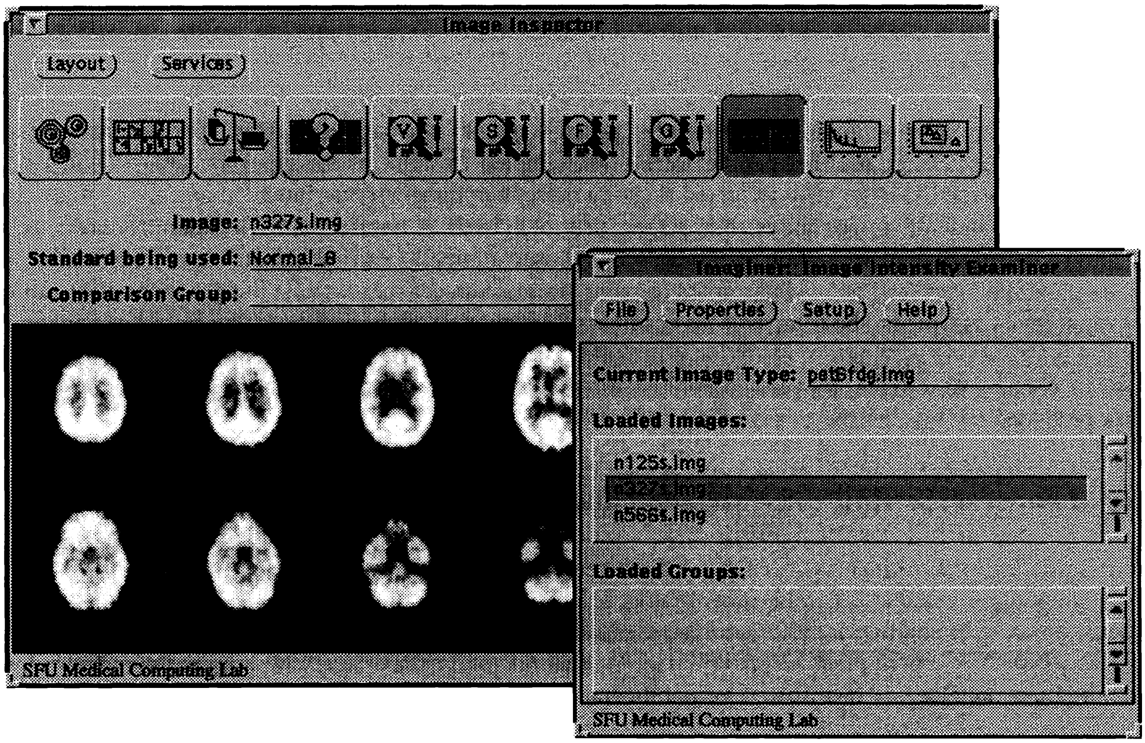


Image Inspector
Layout
Services
Imaginer: Image Intensity Examiner
File
Properties
Setup
Help
SFU Medical Computing Lab
SFU Medical Computing Lab

Image: n327s.img

Standard being used: Normal_8

Comparison Group:

Current Image Type: pet6fdg.img

Loaded Images:

n125s.img
n327s.img
n566s.img

Loaded Groups:

FIGURE 4. Sample image inspector and main window from prototype

types are mock-ups of the program and do not actually do any analysis or visualization. They were created primarily to explore functional and aesthetic alternatives by the developers and potential users. When all the low-level software components are complete, they will be integrated under a graphical user interface whose look-and-feel and features will be as close to the final prototype as possible.

Our iterative prototyping process involved creating a prototype and then collecting feedback, refining the specifications, modifying the design, and revising the prototype to address developer and user priorities. Throughout this process, we strove for a look-and-feel and interface design that compromised as little as possible between extensibility, flexibility, portability and ease of use. Some things which we learned may be useful to developers of other medical applications. We found menus and scrolling list boxes or text fields were more extensible and flexible than icons and specific buttons, but that icons made the program far more user-friendly and intuitive. Users tended to want to customize nearly all attributes of the display, including the way slice images were tiled. Some potential misuses of the program, discussed in Section 5.2, were avoided. For example, because Imaginer will be able to read multiple image formats, an ensuing concern arose about how to respond when a user chose to compare features of two images of incompatible formats. To avoid such a misuse of the system, users will initially be limited to working with only one image type at a time by specifying a current image type in the main window of the program; only images of this type can be loaded and analyzed within Imaginer at one time. Future extensions to Imaginer may explore ways of enabling users to work with images of different formats simultaneously.

## 5.2 Automating analysis

The harder something is to do manually, the stronger the pressure becomes to make it automatic. However, not all analysis should necessarily be automated. Automation is desirable to improve efficiency and eliminate bias, but in areas of medical research and imaging, automation is often better used in conjunction with manual inspection and intervention. To illustrate this point, consider the often difficult task of decision-making. The reason a decision-making process can be hard is often because the criteria are not well specified, in which case it absolutely should not be automated. For these reasons, some aspects of Imaginer will never be fully automatic. Standard generation is an example of one of those components. The process Dykstra used of selecting significant sequences in an averaged image of normals to define a standard pattern of activity can be duplicated by other researchers, but should not be fully automated because so many confounding factors in standard generation cannot easily be anticipated or automatically detected.

A second issue that arises when automating analysis is the possible danger that can result from accidental misuse of an automated system. In particular, we found that people were often tempted to think of Imaginer as a diagnostic tool, possibly because its prototypical features can be misinterpreted as doing more than Imaginer was intended or claimed to do. Imaginer is an adjunct in the interpretation of images and is not a diagnostic tool in itself. A poorly defined standard for normals, for example, could result in a detrimental interpretation of subject scans if used clinically. Even with satisfactory standards, images that conform to the standard pattern may be misinterpreted as being normal subject scans, while images that fail to conform to the standard pattern might be assumed to be from a diseased subject. Matching a standard or deviating from one neither proves nor disproves normality or the presence of disease. What it provides instead is evidence which might suggest or support a conclusion. As Imaginer is being developed, tested, and applied, attention must be paid to the fact that the printouts and graphical representations produced by Imaginer never give diagnostic conclusions.

## 5.3 Duality of purpose: scientific investigation and clinical use

Much of the design effort of Imaginer has been divided between addressing two separate uses of the system. First, Imaginer must be powerful enough to enable scientific investigators to evaluate the usefulness of our method of analysis for various image types and purposes. Secondly, we have assumed that researchers will find the method useful and that Imaginer should be appropriate for clinical use. In clinical use, a high degree of automation is desirable with minimal mandatory intervention so long as the program cannot easily be misused for purely automated diagnostic purposes.

Many trade-offs arose in meeting the two separate types of needs. With researchers in mind, Imaginer was designed to enable users to see and adjust any value used during analysis, as well as be able to determine and set appropriate defaults. The maximum size a volume could reach to still be disregarded as noise is an example of a parameter that might be considered calculable based on the machine specifications, but that a researcher might want to be able to modify. Calculated values may be precise and automatic but they may not be the most appropriate ones to use in every context; in this example, 'noise' may not always be a technical term with an explicit determination, but rather, may be a word used with semi-intuitive connotations. Clinicians on the other hand will likely appreciate a different set of features of Imaginer. Default values can be saved and used for almost all analysis parameters, and once the program has been set up to work with a clinician's image type and standards, no parameters need be adjusted to achieve useful results. These then are some of the ways that duality of purpose has affected the selection of features and functionality specifications. The design of the graphical user interface (described earlier in Section 5.1) was also affected through trade-offs between using intuitive, user-friendly icons, buttons or direct manipulation techniques and more powerful yet complex commands or macros. Finally, since neither researchers nor clinicians should have to be expert computer users as well, Imaginer should require as little time as possible for them to learn to use.

## 6. CONCLUSION

The interactive system that we are developing will aid in the analysis of physiological emission images. The system is designed to be flexible and powerful enough to give considerable freedom to the user who wishes to investigate the analysis method, while providing a convenient, unbiased, usable system for routine analysis. We have considered human factors and usability issues throughout the specification, analysis and design stages of Imaginer and believe that because of this, the system's use will minimize potential misinterpretation of patient data.

## 7. ACKNOWLEDGEMENTS

This work was supported by Simon Fraser University's Centre for Systems Science and the Natural Sciences and Engineering Research Council of Canada, Strategic Grant #4020. Image data used in developing Imaginer was provided by the UBC/TRIUMF PET program.

## 8. REFERENCES

1. Dykstra, C., "3D Contiguous Volume Analysis for Functional Imaging," *Ph.D. thesis, School of Computing Science, Simon Fraser University, Canada*, April 1994.

2. Dykstra, C., Harrop, R., and Atkins, M.S., "Defining a 3D Pattern for Normal PET Images," *Journal of Clinical Nuclear Medicine (Presented at the Annual Western Regional Meeting of the Society for Nuclear Medicine, Vancouver, Oct. 1993.),* Vol. 18(10), p. 931 (abstract only).

3. Evans, B., Harrop, R., Heywood, D., Mackintosh, J., Moore, R.W., Pate, B.D., Rogers, J.G., Ruth, T.J., Sayre, C., Sprenger, H., Van Oers, N., and Yao Xiao Guang, "Engineering developments on the UBC-TRIUMF modified PET VI positron emission tomograph," *IEEE transactions on Nuclear Science*, NS30:707-710, 1983.

4. Shneiderman, B., *Designing the User Interface*, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1992.