Multidimensional Analysis and Visualization Software for dynamic SPECT

Niklas Röber[1], Torsten Möller[2], Anna Celler[3], Troy Farncombe[3], Thomas Strothotte[1]

[1] Otto-von-Guericke Universität Magdeburg, Germany, [2] Simon Fraser University, Vancouver, Canada, [3] Medical Imaging Research Group, University of British Columbia, Vancouver, Canada

Objectives

Our objective is to create tools to aid in the visual exploration of multi-dimensional medical data sets. The application presented herein is concerned with the visualization and analysis of time-varying dynamic SPECT data and allows the quantitative identification of 3D regions of interest for use in kinetic modeling. The visualization of the data should be intuitively and visually understandable. A parametric display illustrate the flow and the speed of the activity.

Background

The reconstruction method to compute dynamic SPECT data was developed by the Medical Imaging Research Group at the University of British Columbia, Vancouver, Canada, and the Laboratoire Mathematiques pour l'Industrie et la Physique, Université Paul Sabatier, Toulouse, France. The visualization of this data is the first collaborative project between the Medical Imaging Research Group and the Computational Visualistics Program at the Otto-von-Guericke Universität-Magdeburg, Germany. This application is implemented on a Macintosh in IDL 5.0.3 from Research Systems. It is planned to port the program to the PC platform, where additional visualization packages can be used.

Methods and Design

The display screen is divided into independent sections that permit for two different ways of analysis of the same data or the comparison of two different data sets. You can connect the two viewports to interact with both views simultaneously. Common techniques for interaction like rotation, translation, scaling or slicing are implemented and can also be animated. Oblique slices can be used to reorient the data for use in the Slicer Window. The application allows to import raw data of any size or data type and data from an ICON-Database. Specification of lower and upper thresholds allows to blend of high and low noise from the data, while smoothing removes noise from the data. By using clipping planes you can clip off unimportant parts of the data. The focus for the visualization in this application lies on volume rendering, where you can chose between two different rendering techniques, alpha blending and maximum intensity projection. You also can specify and adjust color and opacity tables to enhance the visualization. To speed up the rendering process, you can increase the step width and turn off the interpolation. By selecting iso-values, the user can interactively explore different layers of the data which are extracted and visualized as polygonal iso-surfaces. Quantitation is performed by 3D regions of interest which later can also be used for kinetic modeling. Finding and defining those 3D regions of interest can be done by either painting on each slice, interpolation between two shapes on two different slices, or by 3D region growing using thresholds. In order to get a better understanding of the location of the activity, you can overlay attenuation maps and in the future also fuse your data with images from a CT or MRI scanner. Animations can be created from the time-varying data and/or by animating parameters. The animations can be saved as image sequences or as MPEG movies. Parametric objects are used to visualize the flow and the speed of the activity. To visualize these data, we have to use an external package because of the limitations of IDL in this area. We work with the Visualization Toolkit (VTK) from Kitware which is used as an external C++ library which can be called directly from IDL.

Volume Graphics vs. Vector Graphics

Because of the nature of the data, which is inherently volumetric, the best way to visualize this data is by using volume graphics to convey information within these data. With volume graphics you are able to visualize a whole 3D data set at once. By putting one slice on top of each other you are creating a 3D matrix containing the 2D slices. Classification is performed by assigning color tables and opacity tables to the data set. While color tables enhance the difference in the density distribution of the activity, an opacity table adds transparency to the data, to see the internal structure of the data. You can use several different composite functions to determine the final color of the screen pixel. From your camera position you are sending several rays into your volume and compute the pixel colour by evaluation of the chosen composite function. Alpha blending is used to create semi-transparent volumes, while the maximum intensity projection displays the voxel with the highest gray tone. Another common technique in volume graphics is to visualize the different layers of the data. This can be done by extracting an iso-surface for a given gray tone and visualize these iso-surface either as opaque polygon or as translucent polygon together with the volume. All voxels which are lying on this iso-surface have the same gray tone. The iso-surface is computed by the Marching-Cubes algorithm.

While with volume graphics we visualize just the density of the activity, with vector graphics we are able to visualize the flow of the activity. To extract vector information out of the data set, a 3D Horn-Schunck algorithm to compute optical flow is used. The constraints of this algorithm are that the activity for the current voxel does not change very much between two volumes and that neighbouring vectors point to nearly the same direction. These vectors can be interpreted as flow of the activity, but we can also visualize the speed and the acceleration of the activity. For the graphical visualization of these vectors we use oriented 3D glyphs. A simple example are cones, where we use the tip of the cone to point at the direction of the flow and model the size and the color to the magnitude of the flow. To give a better orientation of the location of those vectors, an iso-surface of the volume is also visualized as a wire frame mesh. Other possible vector visualization techniques are hedgehogs or line integrated convolution (LIC).

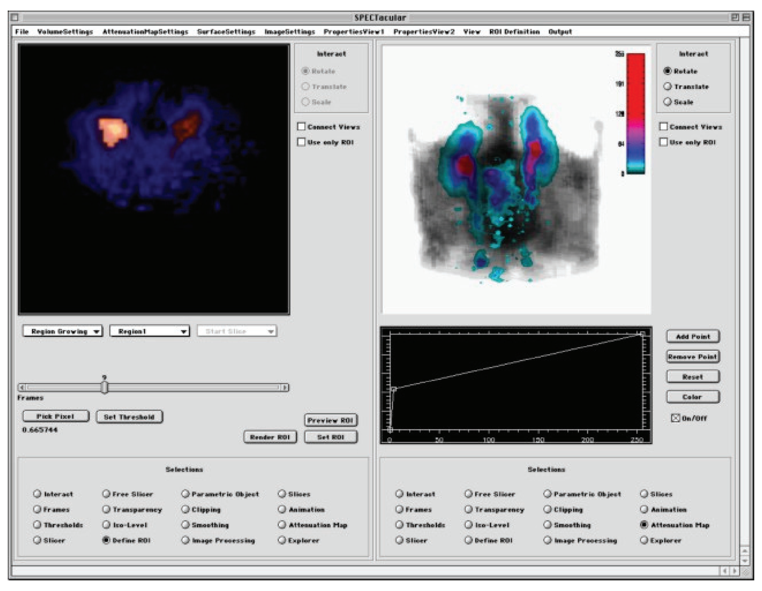
Conclusions and Future Work

With the dynamic SPECT method and this program for the visualization of these data you can study a physiological process as it happens in nature - over time. So it is easier to discover unusual changes or abnormalities. Using the vector visualization algorithms you are also able to detect flow abnormalities. Our program shows a novel approach to the analysis and visualization of dynamic SPECT data. It provides interactive tools for the qualitative (visual) and quantitative (analysis) of these data sets. With the future improvements it will be a powerful tool to explore, discover, and finally understand your data.

To improve the understanding of the anatomical location of the activity, it is planned to fuse CT and/or MRI data sets with dSPECT data. Kinetic modeling will be used to concentrate the quantitative information which can result in a better understanding of the underlying physiological process. In order to use external packages, like the VTK or VGL, to extend the graphical capabilities of IDL it is necessary to port the program from the MAC platform to the PC platform. To speed up the volume rendering process we plan to use either the VTK or the VGL as an external renderer for IDL. On a view:update event, IDL will call an external library to render the image. After the image is finished it will be displayed in the IDL viewport. This all can be done without user notification. The Visualization Toolkit will be used to evaluate new vector visualization techniques, like streamlines, streamtubes or glyphs, to improve the essence of the data.

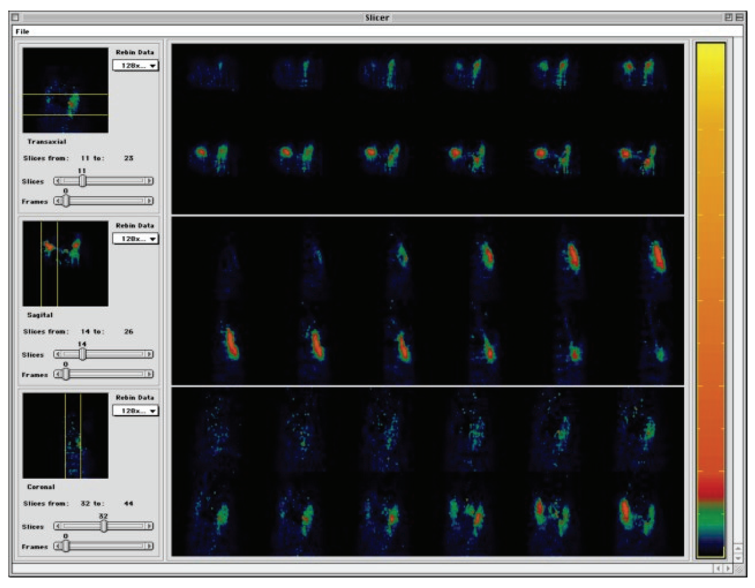
SPECTacular

This is the main window of the application. You see the two worksheets, both contain the same data set, a kidney study. On the left side you can see the dialog to define 3D regions of interest using region growing, while on the right side you can see the dialog to set up some properties for the attenuation map. At the bottom of both worksheets you see the functions of the program. Each of them guides you through different dialogs or result in sub-applications. In the dialog right beside the views you can specify the current interaction method. In the menu on top of the application you can set some properties to influence the visualization.



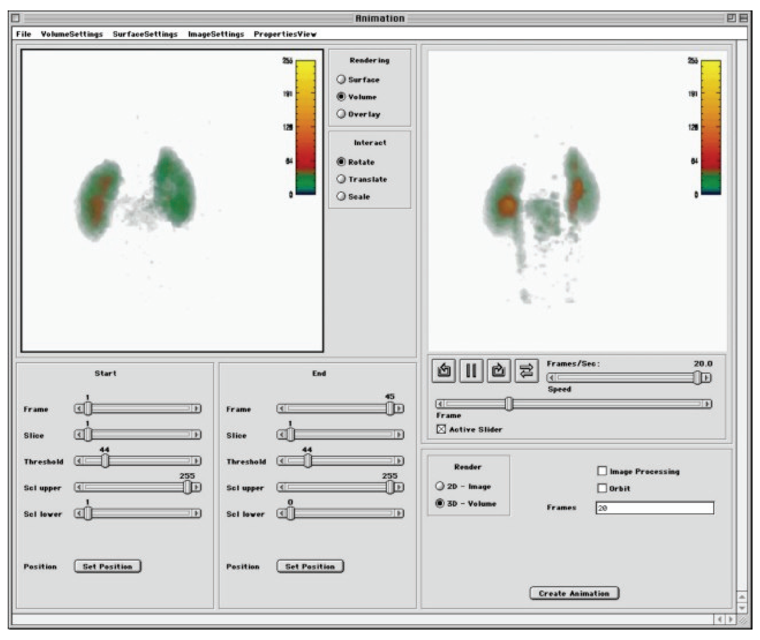
Slicer

In this sub-application you can visualize orthogonal or oblique slices. The program can be started from the main application and you can decide to use either the original orthogonal slices of your data or the oblique slices, created in the main application. The slices are grouped as transaxial, sagital and coronal slices. In the small windows on the left side you can see which slices are currently displayed. You can switch to other slices and frames by using the slider bars.. Basic image processing techniques like smoothing or rebinning are implemented.



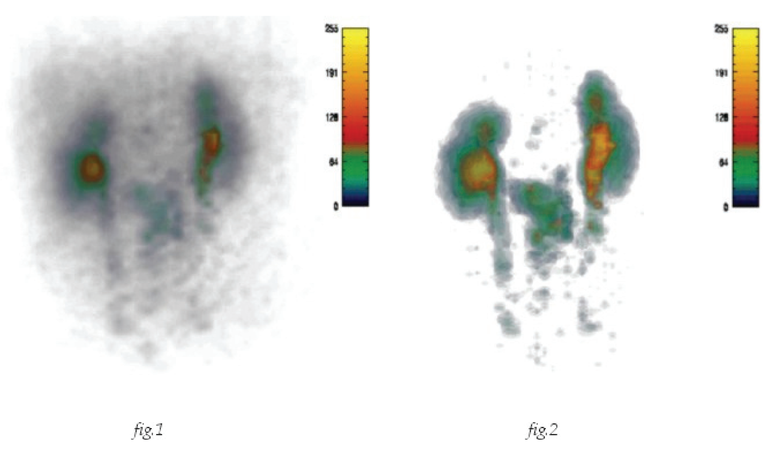
Animation

This sub-application assists you to create animations, which can be either saved as image sequences or as MPEG movies. The view on the left side contains the data from the main application with all the properties set. Now you have to specify some parameters for the animation and select the type of animation, either 2D slices or 3D volumes. Press create animation and enjoy. A future implementation will include full keyframe animation to set up the parameters more precisely.



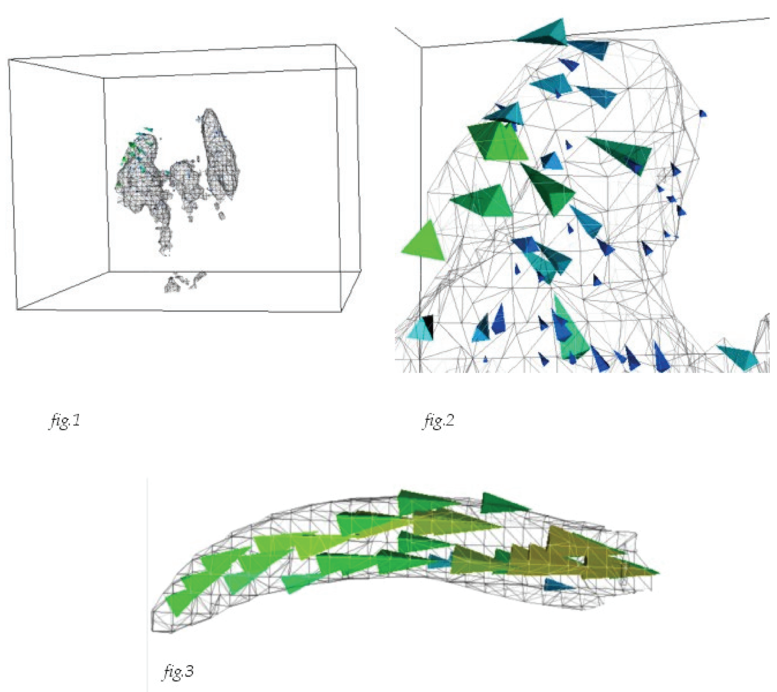
Thresholds

Thresholds are used to extract and enhance portions of the entire data set. We use a lower and a upper thresholds to spread the data between these two thresholds to the whole range. The resulting image is contrast enhanced and low and high noise are removed. The first image shows you a volume without thresholding, and the second one is scaled by a lower threshold of 8 and a higher threshold of 150.



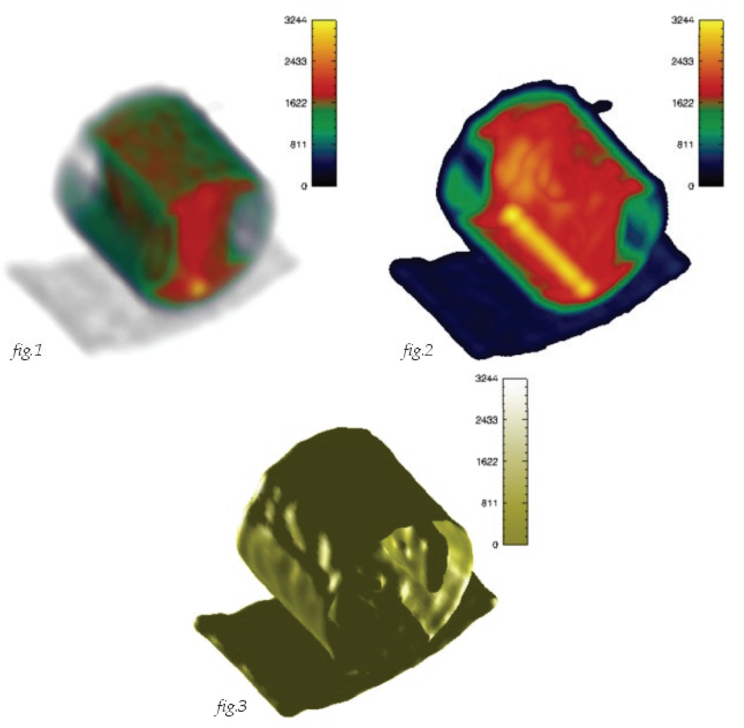
Parametric Object

Parametric objects are used to visualize quantitative results. We are using these parametric images to visualize the flow of the activity. Therefore you have to extract vector information from the data set and visualize these vectors in a visually understandable way. The data which was used for these images comes from a kidney study. In the images you can see an iso-surface, visualized as wire frame mesh, and inside this mesh vectors, visualized as cones. The tip of each cone points into the direction of the flow, and the size and the color reflects the magnitude of the flow. The first image gives you an overview, while the second and the third showing some details.



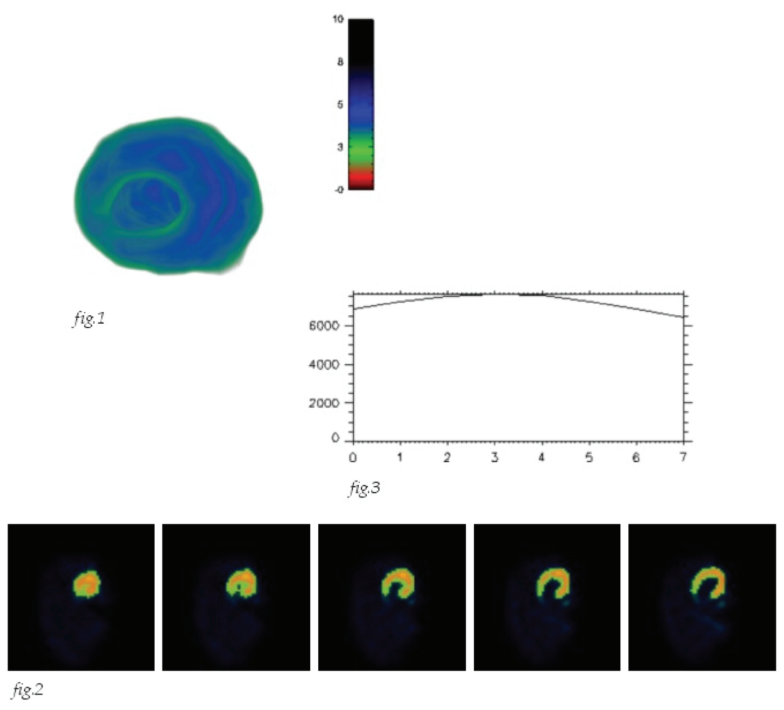
Volume Rendering

The focus of the visualization in the application presented here lies on volume graphics. Here are some examples of different rendering methods using volume rendering. The data comes from a dynamic phantom developed by the Medical Imaging Research Group. In these examples only the attenuation map of this phantom is visualized. The first image shows the data with applied color- and opacity table rendered using alpha blending. The second image is the same, but this time using the maximum intensity projection, and the last shows an extracted iso-surface from gray level 25, rendered as a polygonal surface.



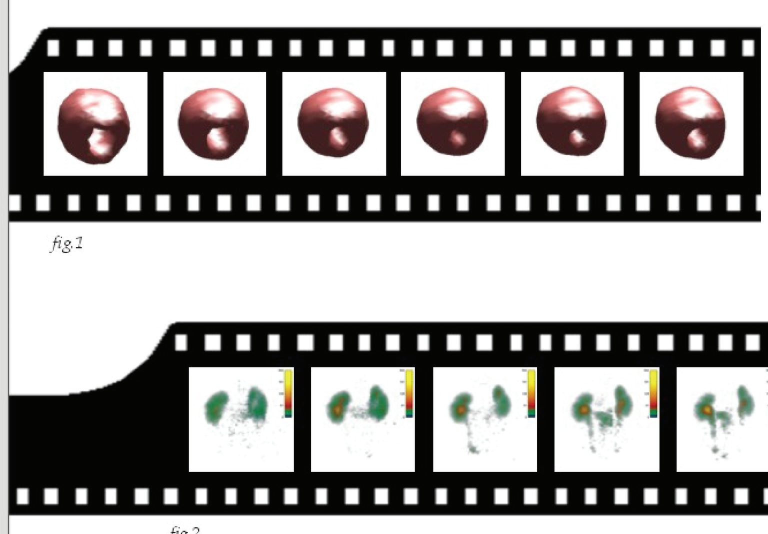
Quantitation by 3D ROI's

In order to get quantitative results you have to know how much activity is contained in a voxel at a given time. With this application you can define up to 255 different 3D regions of interest. This can be done either by painting on slices, interpolation between two shapes or 3D region growing. These roi's can later also be used for kinetic modeling. The activity inside the roi's are summed together and plotted over the time as time-activity curves. On the right side you can see a 3D roi, rendered as volume and below a few of the corresponding slices. Figure 3 shows you the time-activity curve for this region of interest.



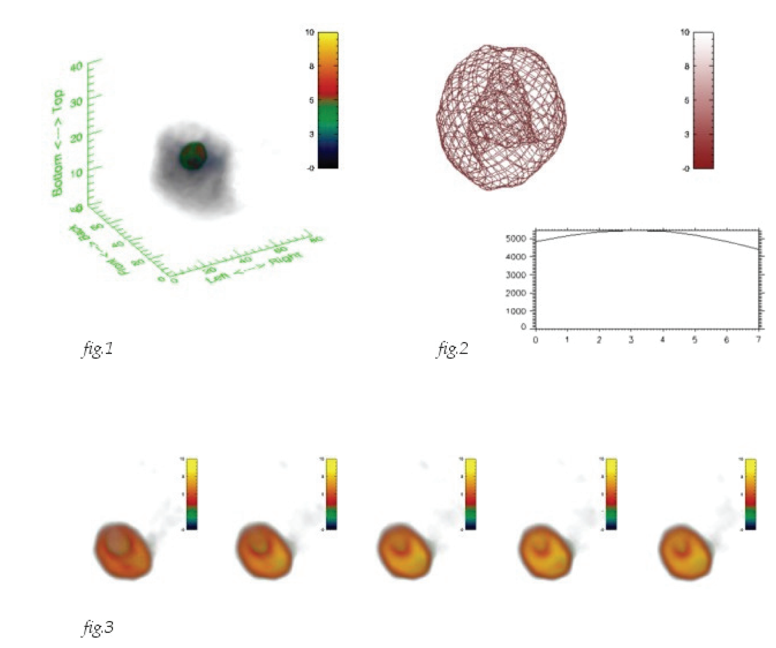
Time

The big advantage of dynamic SPECT is that you have information about the activity distribution over time. So you can see and analyse physiological processes as they happens in nature - over time. Each time frame is represented by a volume, which can be visualized like a static study. In the first example on the right you can see a few frames from a gated heart study, rendered as iso-surfaces. The second example illustrates a kidney study, which is visualized as semi-transparent volumes using alpha blending.



Patient Study - Heart

This heart study is a gated heart study, which was acquired using a single headed camera. The reconstruction was done using the OSEM algorithm. Figure 1 shows the whole data set at frame 0, rendered as semi-transparent volume using alpha blending. Figure 2 displays a 3D region of interest of the whole heart, rendered as wire frame mesh, and below the corresponding time activity curve. Figure 3 shows different time frames of the data, each of them rendered as volume.



Patient Study - Kidneys

This example is a kidney study which was acquired on a Siemens ecam camera with two heads, 90 degrees apart, and 90 degrees rotation. The total acquisition consists of 32 projections per head (20 sec per stop) and the reconstruction was done with dynamic expectation maximization. Figure 1 displays frame 16, rendered as volume with a gradient opacity table, while figure 2 displays the same frame, but this time with the top clipped off and with gray tones from 45 to 65 blended off. In figure 3 and figure 4 you can see two different 3D regions of interest and the corresponding time activity curves.