International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.6, No.5 (2013), pp.143-152
http://dx.doi.org/10.14257/ijsip.2013.6.5.13

Three Dimensional Visualization Toolbox for Medical Images Based on IDL

Min Tang[1] and Feng Chen[2]

[1]*School of Electronics and Information Engineering, Nantong University, Nantong 226007, China*
[2]*School of Electrical Engineering, Nantong University, Nantong, 226007, China*
[1]*tangmnt@yahoo.com.cn,* [2]*tangmnt@hotmail.com*

Abstract

Medical image visualization system, which is of great value in medical research and clinic diagnosis, has been a focal field in recent years. IDL (Interactive Data Language) has a vast library of built-in math, statistics, image analysis and information processing routines, therefore, it has become an ideal software for interactive analysis and visualization of two or three dimensional scientific datasets. The principles and development are proposed to design a novel three dimensional visualization toolbox for medical images based on IDL platform using object oriented programming and object graphics techniques, with experimental results exhibited. It is demonstrated that our developed toolbox can carry out interactive volume and slice operations in both two and three dimensions effectively and efficiently, meanwhile, it has advantages of extensive applicability, friendly interaction, convenient extension and favorable transplantation, which is adequate for medical images processing and analysis.

Keywords: *Visualization in scientific computing, Interactive data language, Medical images, Object oriented programming, Object graphics*

1. Introduction

As the gradual developments and wide applications of modern medical apparatuses, all kinds of medical images have been playing critical role in clinic diagnostic and medical research. Up to date, physicians are conditioned to discover and observe focus's information through multiple two dimensional (2D) slices of CT/MRI. This process is a qualitative analysis procedure of medical images, with noises brought about unavoidably. In recent years, scientific visualization of medical images has been a focal field, which is an important application of computer graphics, image processing, artificial intelligence and pattern recognition in bioengineering.

The main task of scientific visualization of medical images is to apply the techniques of visualization in scientific computing to medical images, namely to transform, process, analyze and display medical volume data obtained from experimental results, medical apparatuses and computational models, such as B ultrasonic, CT, MRI, PET, SPECT, histological sections and so on. It can make up the deficiencies of medical apparatuses, provide physicians for three dimensional (3D) medical images to analyze complex anatomies and their relationship by means of qualitative and quantitative methods, and increase the speed and accuracy of diagnoses. In a word, 3D reconstruction and visualization of medical images play a great role in clinic diagnostic, surgery planning and simulation, radiotherapy planning, education training and so on.

ISSN: 2005-4254 IJSIP

Several commercial software and source code toolkits for processing and analysis of medical images have emerged in recent 30 years, of which the representatives are Analyze [1], Amira [2], VTK [3], ITK [4], OpenGL [5], OSIRIS [6], Image-Pro Plus [7], 3DViewnix [8], Alice [9], Movx [10] and et al. However, similar research has been carried out in recent 10 years in our country for lack of advanced medical apparatuses. 3D Med [11] of institute of automation, Chinese academy of sciences and 4DView [12] of the key laboratory of biomedical engineering in ministry of education, Zhejiang university are two reported embryonic forms of visualization software for medical images. In this paper, the principles and development are proposed to design a novel 3D visualization toolbox for medical images based on IDL (Interactive Data Language) [13], with experimental results demonstrated its performance and characteristics.

IDL, a significant product of Research Systems Inc., is an ideal software for interactive analysis and visualization of 2D/3D scientific data. It is a powerful array-oriented interpreted language, which combines all of the tools needed for any type of project, from interactive analysis and display to large scale commercial programming projects. The outstanding advantages of IDL include: capability to analyze large scale volume data; advanced image processing abilities; 2D/3D interactive graphics techniques; object oriented programming; OpenGL based hardware accelerated graphics; a vast library of built-in math, statistics, image analysis, information processing routines; flexible input/output facilities; multi-platform graphical user interface (GUI) widgets and preferable portability. Researchers, developers, and engineers at myriad commercial corporations, governments organizations and academic institutions (*e.g.*, Mars Exploration Team, NASA, Los Alamos National Lab's Biophysics Group) use IDL to bring products to market and make discoveries less expensively and in less time [13].

As a result of the advantages of IDL mentioned above, it is decided to program in IDL, alternative to programming in C, C++, FORTRAN or MATLAB, to design and implement medical images visualization toolbox. The architecture of this article is as following: first the characteristics and advantages of IDL are introduced; secondly the several kernel programming techniques are explained, including object oriented programming, graphics system, graphical user interface and IDL virtual machine; then the experimental results are illustrated and finally the functions and advantages of our toolbox are concluded.

2. Characteristics and advantages of IDL

IDL has become an ideal software for 4th scientific computation and visualization language, and it is regarded as the integration of VC, VB, JAVA, FORTRAN, MATLAB and OPENGL. In the concrete, IDL performs the same function as VC as far as menu selection, message transmission and class definition. IDL is easy to design GUI, which is similar with VB. IDL performs well as JAVA in favorable transplantation among different operation systems. IDL has an analogy to FORTRAN as far as function calling, data transmission and syntax rules. Moreover, IDL outperforms OPENGL in the respect of 2D/3D image representation.

IDL is a powerful array-oriented language and a complete computing environment for the interactive analysis and visualization of scientific data. It has become an ideal software for data analysis, graphics visualization and cross-platform application for it's rich library of built-in math, statistics, image analysis, information processing routines. The outstanding advantages of IDL include [13-14]:

1) IDL supports virtually every data format, type and size.

2) It is easy to use IDL’s built-in library of math, statistics, image processing and signal processing routines.

3) IDL provides tools to easily create powerful visualization from simple 2D plots to OpenGL accelerated 3D graphics.

4) Taking advantage of IDL’s complete, native toolkit and drag-and-drop GUI builder, it is convenient to build the ideal interface.

5) IDL’s interpreted language reduces design-compile-link-test cycle by issuing commands on the fly and seeing results immediately.

3. Several Kernel Programming Techniques

3.1. Object Oriented Programming

Compared to traditional programming techniques, object oriented programming begins to model the objects containing both routines and data. Object oriented programming can provide a layer of abstraction that allows the programmer to build robust applications from groups of reusable elements.

From version 5.0, IDL provides a set of tools for developing object oriented applications. IDL's object graphics engine is object oriented, and a class library of graphics objects allows to create applications that provide equivalent graphics functionality regardless of computer platform, output devices, etc. Programmers can use IDL's traditional procedures and functions as well as the new object features to create new object modules. Applications built from object modules are, in general, easier to maintain and extend than the traditional counterparts.

3.2. Object Graphics Systems

Direct graphics and object graphics are the two distinct graphics systems supported by IDL. Direct graphics can be understood as a device-oriented mode, *i.e.*, the graphics can be output through a special appointed device (*e.g.*, screen, printer) chosen by programmer. The appreciable merit of direct graphics lies that it can be used in either command lines or programs simply and conveniently. However, once a graphics is drawn to the graphics device, it cannot be altered or reused, meaning that the IDL commands must be reissued to create the same graphics on the different device. This is the chief disadvantage of direct graphics. The direct graphics devices supported by IDL include screen, postscript, printer, Z-buffer and so on [14-16].

On the contrary, object graphics uses an object oriented programmer’s interface to create graphics objects, which are drawn explicitly to the destination chosen by programmer. The outstanding aspects of object graphics are [14-16]:

1) Object graphics can be accelerated by hardware in three dimensions, which enhances flexibility and interaction.

2) While it is still possible to create and use graphics objects directly from the IDL command lines, the syntax and naming conventions of object graphics make it more convenient to build a program offline than to create graphics objects on the fly.

3) Graphics objects encapsulate functionality, *i.e.*, individual objects include method routines that provide functionality specific to the individual object.

4) Object graphics is device-independent and any graphics object can be displayed on any physical device.

5) Object graphics is object oriented and can be recreated and reused.

6) It is of great use for the programmer to cognize the memory issue and memory leakage in that object graphics persist in memory.

To explore the key techniques of medical images visualization, including surface reconstruction, volume reconstruction, rotation, zoom, translation and section, it is clear to adopt IDL object graphics to develop the three dimensional visualization toolbox for medical images. An object graphics display can be thought of as a group of graphics objects organized into a hierarchy tree, shown in Figure 1 [14].

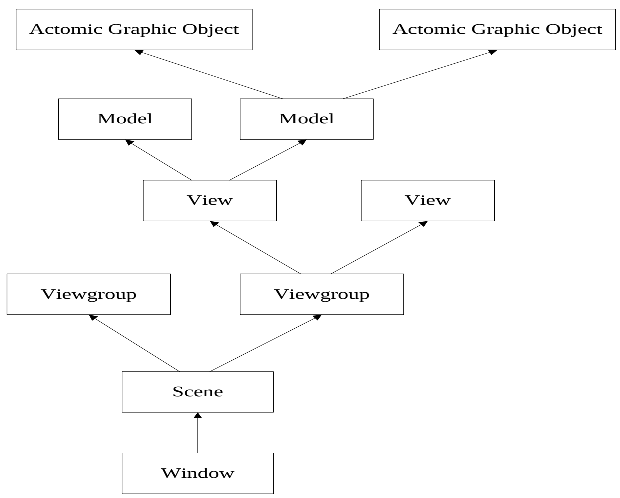

Figure 1. Object Graphics Hierarchy Tree

The operations of IDL object graphics system are referred to IDL Online Manuals [14] in more details, here only described briefly [15-16].

Object Creation: To create an object from the IDL object class, use the function.

$$oOBJ = Obj\_new('IDLgrYyyyy')$$

Object Properties: Some IDL objects have properties associated with them—things like color, line style, size and so on. Properties can be set when creating the object or changed after creation.

$$oOBJ = Obj\_new('IDLgrYyyyy', property = value)$$

Object Methods: It is allowed to define method procedures and functions using all of the programming tools available in IDL. Method routines are identified as belonging to an object class via a routine naming convention.

$$\begin{aligned}
&oOBJ -> GetProperty &&;retrieving\ property\ setting\\
&oOBJ -> SetProperty &&;setting\ properties\ of\ existing\ objects\\
&oOBJ -> Add &&;adding\ property\\
&oOBJ -> Remove &&;deleting\ property
\end{aligned}$$

Object Destruction: Objects are persistent and they exist in memory until destroyed.

$$Obj\_Destroy, oOBJ$$

3.3. Graphical User Interface

GUI is a human-computer interface including windows, icons and menus and it can be manipulated by a mouse or often to a limited extent by a keyboard as well. The major advantage of GUI is that it makes computer operation more intuitive, and thus easier to learn and use. Except for this intuitiveness of operation, GUI generally provides users with immediate, visual feedback about the effect of each action, referred to "look-and-feel."

When creating an application, many object oriented tools exist to facilitate writing a GUI. Each GUI element is defined as a class widget from which the programmer can create object instances for different applications. Code or modify prepackaged methods can be used directly to respond to user stimuli.

Two methods can be chosen to design GUI in IDL, based on procedure or based on GUIBuilder, respectively. Here, the first method is chosen, with the schematic diagram of GUI composition shown in Figure 2.

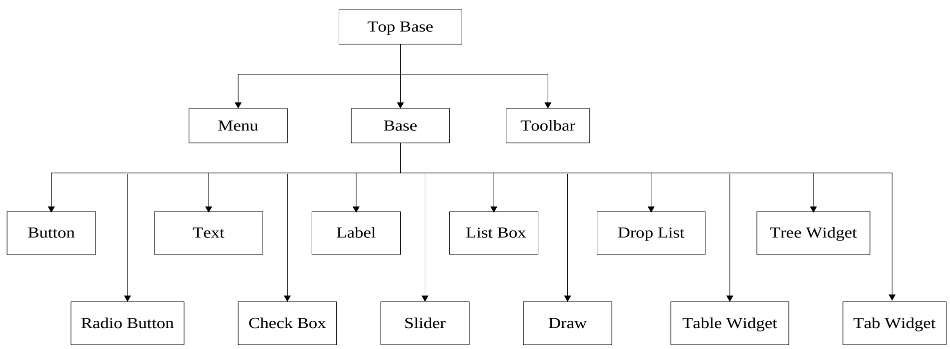

Figure 2. Schematic Diagram of GUI Composition

3.4. IDL Virtual Machine

An IDL program compiled in IDL 6.0 can be saved as a .sav file that will run in the IDL Virtual Machine or to be used as an execute function, which is convenient for software release and distribution. The IDL virtual machine is designed to provide users with a simple, no-cost method for distributing IDL applications in all supported platforms.

Our developed 3D visualization toolbox for medical images is generated as an execute function, the steps of which are as follows: 1) Create .sav files from one or more compiled .pro files with the Save procedure. 2) Create .sav files from a project by selecting Project→Export with the Save File (.sav) option specified.

4. Experimental Results

The 3D visualization toolbox for medical images is developed based on IDL 6.4 with common configured computer, and it is also generated as an execute function for software release and distribution.

The introduction interface is appeared as Figure 3 when double clicking the toolbox logo. After entering the correct password, the main user interface is displayed, with the menu arranged as Figure 4. Figure 5 and Figure 6 are surface reconstruction and volume reconstruction of head CT images and MRI images, respectively. Volume and slices operations are shown in Figure 7-Figure 9, including rotation, zoom, translation and section, with the meaning of principle buttons annotated adjacently.

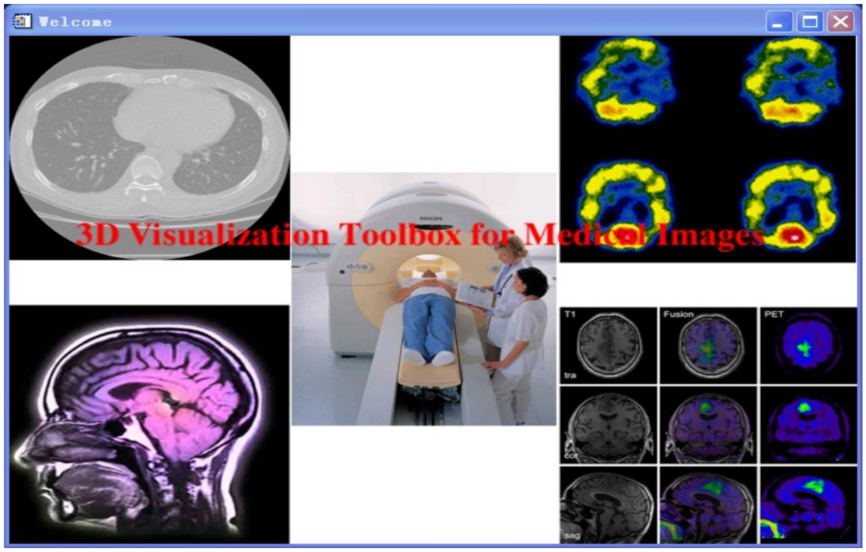

Figure 3. Introduction Interface of our Toolbox

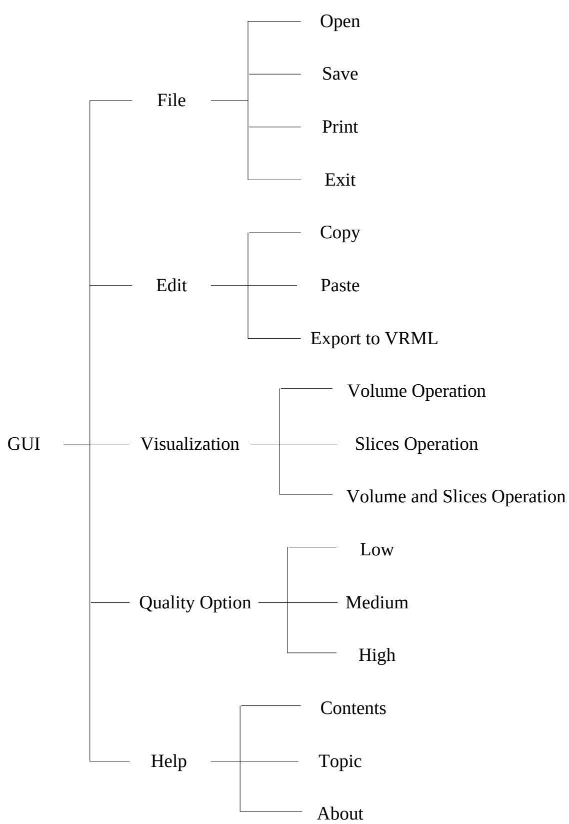

Figure 4. Menu Arranged of our Toolbox

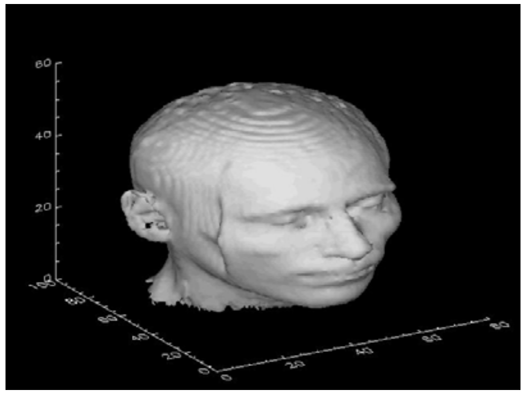

Figure 5. Surface Reconstruction of Head CT Images

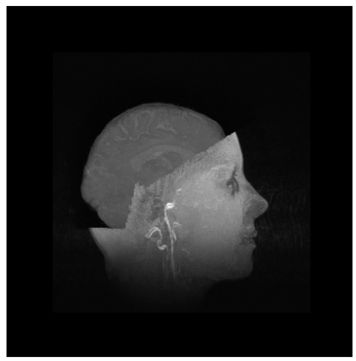

Figure 6. Volume Reconstruction of Head MRI Images

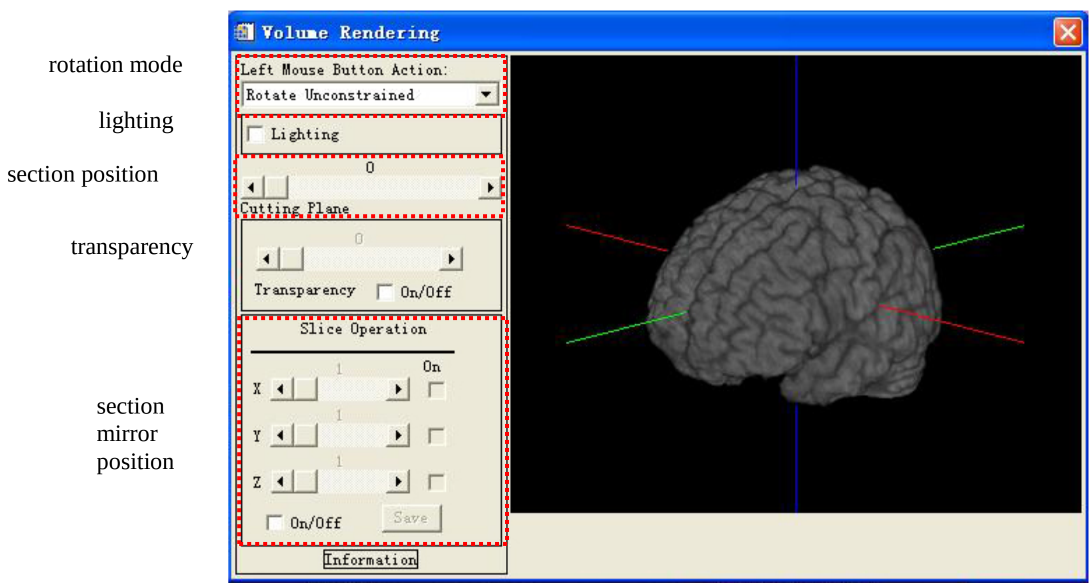

Figure 7. Volume and Slice Operations of our Toolbox

slice direction

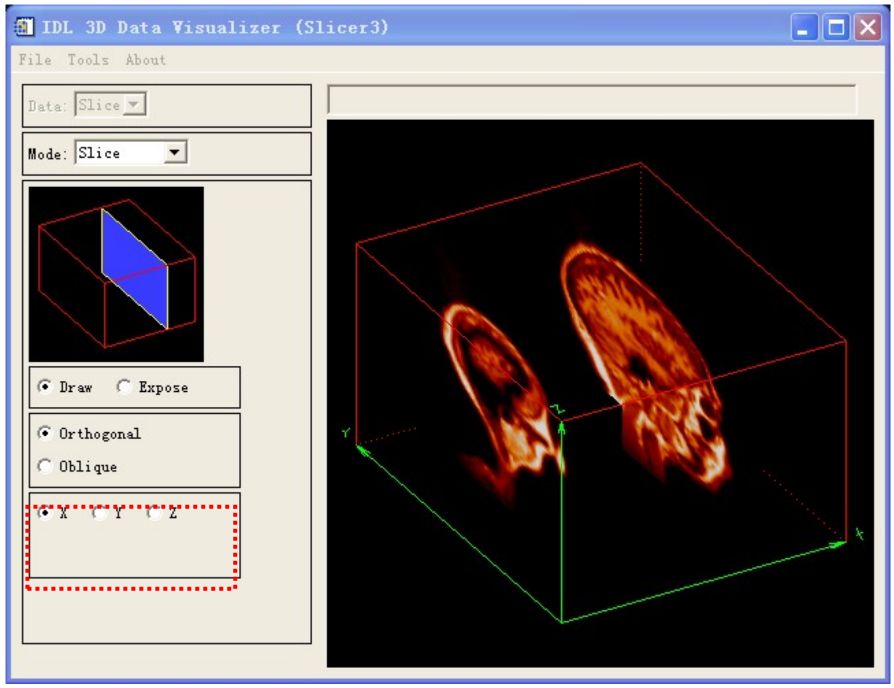

Figure 8. Slice Operation of our Toolbox

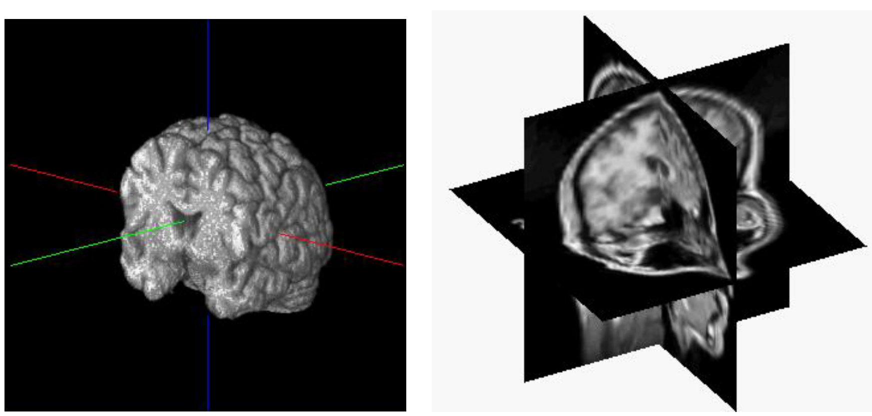

Figure 9. Section Slice from Coronal, Transversal and Sagittal Directions

5. Discussion and Conclusions

3D visualization toolbox for medical images based on IDL environment is discussed and developed in this article, with the performance and advantages demonstrated. The main functions of our toolbox include surface reconstruction, volume reconstruction, rotation, zoom, translation and section. It is convenient for physicians to observe complex anatomies efficiently and effectively, and it has important significance on science and worthiness in practical applications. Moreover, the advantages of our toolbox include:

1) Extensive applicability: it is applied to reconstruct several medical image formats, such as dicom, bmp, jpeg, tiff and so on.

2) Fast operation: the time used for visualization is greatly reduced as IDL supports for OpenGL-based hardware accelerated graphics.

3) Favorable interaction: the operator can control the scale factor, rotation angles, program precision and program termination.

4) Preferable portability: IDL programs run across all supported platforms (Windows, Unix and Macintosh) with little or no modification.

Acknowledgements

The authors acknowledge the support of National Natural Science Foundation of China (No. 61005054), Natural Science Foundation of Jiangsu Educational Committee (No. 09KJD510004), Nantong Municipal Natural Science and Technological Application Foundation (No. K2009032), and Scientific Research Start-Up Foundation for PhD of Nantong University (No. 08B15).

The authors also thank for the editors and reviewers for their valuable and helpful suggestions. These should be brief and placed at the end of the text before the references.

References

[1] Analyze, http://www.mayo.edu/bir/, **(2012)**.
[2] Amira, http://www.amiravis.com/, **(2012)**.
[3] VTK, http://public.kitware.com/VTK/index.php, **(2012)**.
[4] ITK, http://www.itk.org/, **(2012)**.
[5] OpenGL, http://www.sgi.com/products/software/opengl/, **(2012)**.
[6] OSIRIS, http://www.sim.hcuge.ch/osiris/01_Osiris_Presentation_EN.htm, **(2012)**.
[7] Image-Pro Plus, http://www.mediacy.com, **(2012)**.
[8] 3DViewnix, http://mipgsun.mipg.upenn.edu/~Vnews/, **(2012)**.
[9] Alice, http://www.alicesoftware.com/index.htm, **(2012)**.
[10] Mvox, http://www.anamedic.dk/, **(2012)**.
[11] 3D Med. http://www.3dmed.net, **(2012)**.
[12] J. Q. Liu, J. Y. Feng, C. Cai, and H. L. Duan, J. Biomed. Eng., vol. 3, no. 19, **(2002)**.
[13] IDL, http://www.rsinc.com, **(2012)**.
[14] Research System Inc, IDL Online Manuals, **(2003)**.
[15] D. W. Yan, "Basics and Improvements of IDL Visualization Tools", China Machine Press, Beijing, **(2003)**.
[16] P. Y. Han, Comput. Eng., vol. 6, no. 33, **(2007)**.

Author

Min Tang, female, was born in Nantong, Jiangsu Province, P. R. China in 1977. She received her bachelor's degree from Nantong University in 1999 and her PhD degree from Nanjing University of Aeronautics & Astronautics in 2007. Now she is a associate professor in School of Electronics and Information in Nantong University. Her major research fields include image processing, analysis and visualization.

