Journal of Digital Imaging
https://doi.org/10.1007/s10278-017-0047-6

OIPAV: an Integrated Software System for Ophthalmic Image Processing, Analysis, and Visualization

Lichun Zhang [1] · Dehui Xiang [1] · Chao Jin [1] · Fei Shi [1] · Kai Yu [1] · Xinjian Chen [1]

Abstract
Ophthalmic medical images, such as optical coherence tomography (OCT) images and color photo of fundus, provide valuable information for clinical diagnosis and treatment of ophthalmic diseases. In this paper, we introduce a software system specially oriented to ophthalmic images processing, analysis, and visualization (OIPAV) to assist users. OIPAV is a cross-platform system built on a set of powerful and widely used toolkit libraries. Based on the plugin mechanism, the system has an extensible framework. It provides rich functionalities including data I/O, image processing, interaction, ophthalmic diseases detection, data analysis, and visualization. By using OIPAV, users can easily access to the ophthalmic image data manufactured from different imaging devices, facilitate workflows of processing ophthalmic images, and improve quantitative evaluations. With a satisfying function scalability and expandability, the software is applicable for both ophthalmic researchers and clinicians.

Keywords Software system · Ophthalmic image · Image processing · Image analysis · Image visualization · Computer aided diagnosis

Introduction

Ophthalmic diseases especially retinal diseases have been the leading cause of irreversible blindness worldwide, and ophthalmic images play an indispensable role in preclinical research, clinical diagnosis and treatment. Optical coherence tomography (OCT), magnetic resonance (MR), and fundus photography are typical examples that have been widely used. There is a strong demand for a software system to assist researchers and clinicians to process ophthalmologic images.

There are a significant amount of software and toolkits for medical image processing and visualization. Osirix (http://www.osirix-viewer.com) [1], 3D Slicer (https://www.slicer.org) [2], and Paraview (https://www.paraview.org) [3] are typical example for application system based on Visualization Toolkit (VTK) (http://www.vtk.org) [4, 5] and Insight Toolkit (ITK) (https://itk.org) [6], whose purpose is to assist diagnosis and therapy. Other software consists on frameworks aims to help researchers build specific applications such as MITK (http://mitk.org/wiki/MITK) [7, 8] and MeVisLab (http://www.mevislab.de) [9]. They are all suitable for general medical images and provide rich common functionalities including image processing (e.g., segmentation, registration) and visualization. However, they are inapplicable to ophthalmic images. In fact, there is no commonly known software system specially oriented to ophthalmic images and suitable for both researchers and clinicians.

Processing, analyzing, and visualizing ophthalmologic images have its particularity and require targeted algorithms and special software design. First, unlike radiology images, which have unified standards and formats (DICOM) [10], there is no standard for transferring ophthalmic images and associated information between devices manufactured by various

✉ Xinjian Chen
xjchen@suda.edu.cn

Lichun Zhang
lczhang2011@gmail.com

Dehui Xiang
xiangdehui@suda.edu.cn

Chao Jin
jinchao_mipav@qq.com

Fei Shi
shifei@suda.edu.cn

Kai Yu
yukai.suda@gmail.com

[1] School of Electronics and Information Engineering, Soochow University, No.1 Shizi Street, Suzhou, Jiangsu Province 215006, China

Published online: 05 September 2018

venders. That requires the software system to parse multiple image formats from different ophthalmic equipment manufacturers. Second, many different kinds of datasets are involved in ophthalmic imaging and processing. Developers have to design an appropriate data management mechanism and the data conformation and rendering modes for different data of various types in ophthalmic images. More targeted display modes are also required to show ophthalmic image features more intuitively. Third, it is necessary to implement suitable image processing techniques for ophthalmic images because of the difference in the ophthalmic imaging principle and structures. For example, OCT image segmentation is an important step to provide quantitative assessment for the diagnosis and study of ophthalmic diseases. Therefore, image segmentation of anatomical and pathological structures of the eye (e.g., retinal layers) is one of essential functions in the ophthalmic software system. Forth, computer-aided detection is the future trend, but it is still challenging to design the algorithms of automatic ophthalmic diseases detection and integrate them into the software system to meet the practical and clinical requirements. Finally, there is a strong need for reliable and efficient methods to assist users to quantitatively analyze ophthalmic data. The software system should integrate rich functions of data analysis. All these factors make the software system specially oriented to ophthalmic images greatly different from other general medical image processing software. To implement such a system, it requires specialized expertise in multiple areas including software engineering, image processing, pattern recognition, radiology and clinical ophthalmology. Therefore, it is more difficult to design such a system. To address these problems, an integrated cross-platform software system is designed for ophthalmic clinicians and researchers. The system is concentrated on ophthalmic image processing, analysis and visualization.

The main objective of this system is to provide a platform to assist users in handling ophthalmic images. Compared to the other medical imaging software, our software system has many significant advantages.

(1) OIPAV provides various file formats support and local/remote database communications. Besides the DICOM support, it can read the ophthalmic data directly from devices manufactured by various venders (e.g., OCT images from Topcon). The data can be transferred into local SQL database and remote DICOM servers for data management. Furthermore, the system provides comprehensive solutions for standardizing and visualizing multiple types of ophthalmic images. The design of the data tree and the multi-tab viewer facilitates the navigation through images.

(2) In image processing, in addition to extensive commonly employed operations, OIPAV provides targeted functionalities to recognize and segment eye structures automatically. Combining with our laboratory research experience of many years on ophthalmic diseases detection and analysis [11–22], the system integrates our algorithms to provide the "one button" solutions for various ophthalmic diseases' detection.

(3) The analysis module in OIPAV includes automatic important parameters measuring, lesion areas assessing, recovery tracking, and diagnostic report generating. It will assist users to master the situation of eyes accurately, reduce diagnostic errors and improve quantitative evaluations as well as diagnosis efficiency.

(4) Finally, we adopt the modular design and implement the plugin mechanism to develop all functions to make the software system extensible. Developers can expand and integrate functions into the system easily without changing the framework code.

Methods

System Design and Architecture

OIPAV is oriented to ophthalmic images processing, analysis and visualization. It is entirely written in C/C++ and can run on Windows, Linux, and MacOS platforms with sharing a single code base. Since it is a sophisticated system that involves different aspects of retinal imaging, we follow the modularity in developments. Modularity is essential for software stability and components reusability. In OIPAV, modularity is implemented using layers of abstractions and componentized functional units. Figure 1 shows the general architecture of the software system. From top to down, the architecture is constituted of five layers: applications, development modules, high-level libraries, low-level libraries and hardware level. Benefiting from the layer of high-level libraries, we can avoid the low levels and the repetitive work of underlying development to focus on the specific functions and applications.

In the layer of high-level libraries, we use some popular toolkit libraries in the medical community including ITK, DCMTK (http://dcmtk.org) [23], VTK, vtkINRIA3D [24] to provide base functions for higher layers. This layer is divided into three main blocks. The first block deals with image processing purposes and is based on ITK and DCMTK. ITK is an open-source and cross-platform library. It aims to handle image processing tasks, i.e., segmentation and registration and has become the de facto standard in scientific research. DCMTK is a collection of libraries and applications implementing large parts of the DICOM standard. The second block is based on VTK and vtkINRIA3D and dedicated to data visualization and manipulation. VTK provides a pipeline framework allowing users to modify and interact with medical

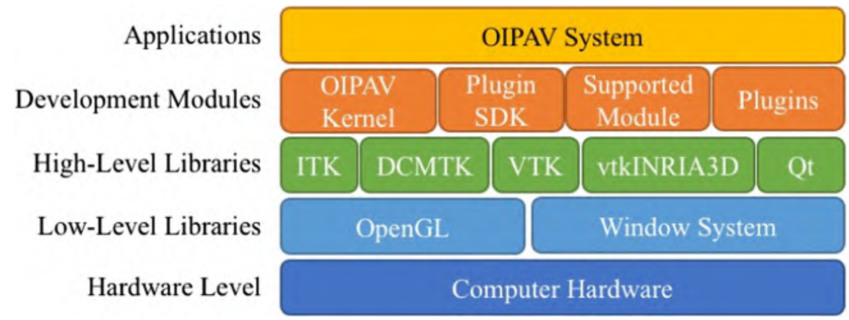


Fig. 1 The general architecture of the system

images. vtkINRIA3D, an extension of VTK, decreases the learning difficulty of making a versatile software for medical imaging. They together provide the foundation visualization and manipulation for higher layers. The third block corresponds to the graphic user interface. We choose Qt [25] to develop the software system as its simple signals/slots mechanism and full adaptation to VTK objects. Figure 2 shows the connections between these libraries in the layer of high-libraries of the system. Blocks are independent from one to the others. While integrating all these libraries is a challenge, there would be great advantages such as functions and features, higher quality of functionalities and a shorter product development cycle.

The layer of the development modules is the key of the whole system. OIPAV is a complicated system, and each part has different algorithms and parameters. Accordingly, we employ a plugin mechanism to reduce the coupling among different modules and provide an extensible and flexible framework. Four necessary modules are designed to cooperate together to implement the plugin mechanism as shown in Fig. 3. Plugin SDK module mainly defines the interface specifications for writing plugins and the base data class. It allows users to create handle functions and corresponding widgets including menu items, tool panels, and dialog. Supported module provides a set of tools for developing plugins, including internal standard data classes and commonly used tools in the platform. Through interfaces provided by plugin SDK module and tools in supported module, all practical functions are implemented in plugins in the form of dynamic link library with the uniform specifications. The functions integrated in the system can be divided into six groups: data I/O, image processing, interaction, ophthalmic diseases' detection, data analysis, and visualization. Figure 4 shows the functional architecture of the system. OIPAV kernel module is the core of the software system, and it take responsibility for identifying, loading various plugins dynamically, creating corresponding menu items and calling operations provided by various plugins when corresponding menu item is activated.

The highest layer is the applications provided by the software system concentrated on ophthalmic image processing, analysis and visualization. In research and clinical situations, different applications can be combined for a particular need. By using the module design method and plugin mechanism, we achieve the software system extensible, reuse and customization for different needs. Develops can easily combine existing function plugins or develop new functionalities from the requirements without changing the framework code.

In order to automatically discover dependent libraries, integrate these components and generate build environment for the various operating system and compilers/IDEs, we employ the CMake [26] utility as VTK and ITK for easy compilation. For source code control, Git [27] is employed to maintain the source code repository.

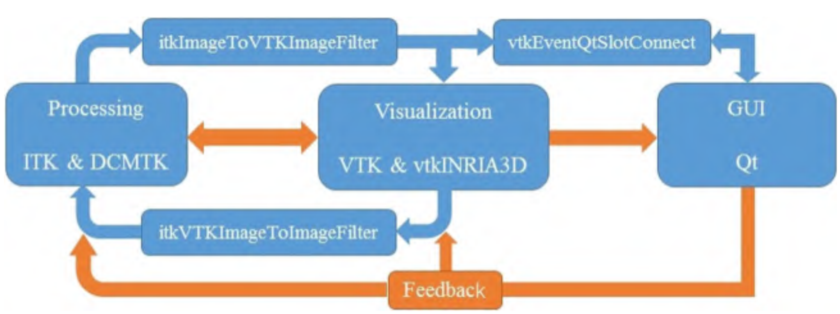


Fig. 2 The interactions between the three library blocks of the system. The orange arrows indicate the data flow, from processing block to the user interface and user interface send controls to the other two blocks. The blue arrows indicate the calling relations between blocs. Therefore, the user manipulates the processing and visualization through the interface. Because of the independence between the blocks, the interface can be rewritten with minimum effort

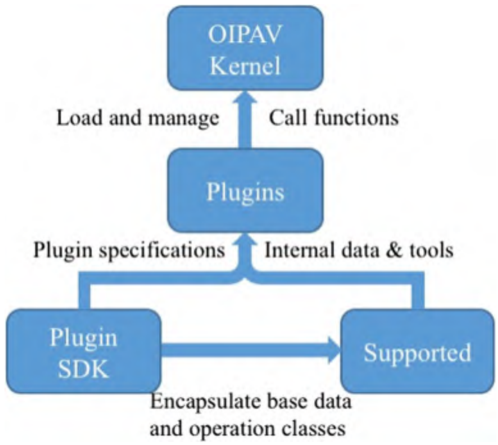


Fig. 3 The plugin framework in the system and connections between different modules

Ophthalmic Data I/O

Regarding to data formats, ophthalmic images are significantly different from other medical images. While most radiology images have the DICOM unified standard format, different ophthalmic medical instrument vendors (e.g., Topcon, Zeiss, and Heidelberg in OCT) define their own specific data formats. That makes it difficult to access image data from different ophthalmic imaging devices.

OIPAV provides various ophthalmic image formats supports. To the specific ophthalmic image data formats such as FDS from Topcon, we design algorithms to parse and convert them to other common formats. Now our system can read the OCT image data of Topcon and other manufacturers. Besides the DICOM file formats support, OIPAV also supports other common 2D/3D medical image file formats such as Raw, VTK, Analyze, TIFF, and JPEG. The input and output functions of different data are all implemented in the form of plugins to guarantee the further extension for other data types.

As a client software, OIPAV can be installed in the workstations connected to the corresponding ophthalmic instruments or independent computers. When installed in the workstations, the software will load the new arrival data automatically after users specify the data folders in the instruments in settings. Users can also choose data to load by the system manually in the independent computers. With the DCMTK library, OIPAV provides the networking DICOM image querying and retrieving capability, which makes it easily connect to the picture archiving and communication system (PACS) [28]. After processing, the various data in the system can be exported into a local database for easy future management. Meanwhile, the files conforming to DICOM standards can be constructed by combining image data and related information inputted and transferred to DICOM server. Figure 5 shows the data flow in the system.

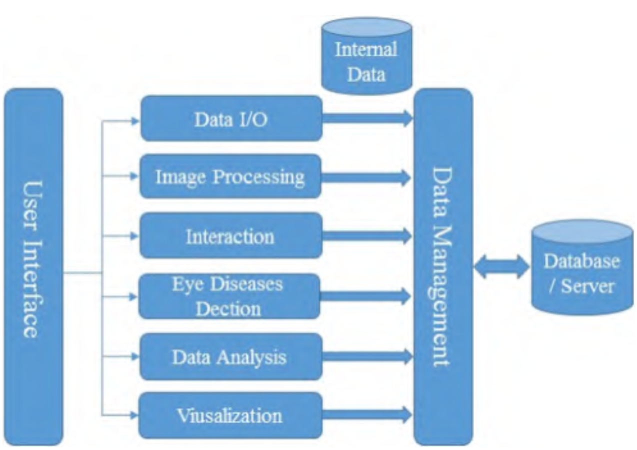


Fig. 4 The function module of OIPAV

Data Object and Visualization

Data representation is one of the cores of the system. In the system, ophthalmic data contain many different kinds of datasets, such as images (OCT images, fundus images), curves (2D retinal layers), surfaces (3D retinal layers, lamina cribiosa layers), and volumes (3D exudation model). In order to allow easier use and display of various datasets, OIPAV defines three standard data types based on VTK and PluginSDK. Therefore, data is divided into three types: ImageData, SurfaceData, and VolumeData. The ImageData represents a vtkImageData and has a geometric structure that is a topological and geometrical regular array of points. A large spectrum of 2D/3D ophthalmologic images is included in this type. The VolumeData carries a vtkPolyData object with a geometric structure consisting of vertices, lines, polygons, and/or triangle strips. The SurfaceData supports vtkUnstructuredGrid objects representing any combinations of any cell types. These OIPAV data types encapsulate the data storage structure used in ophthalmic image processing and details of reading/writing and make a unified abstract representation of various data. Furthermore, they provide unified access interfaces for algorithms that allow easier use and display of various datasets. Figure 6 shows the hierarchy of the standard data classes in the OIPAV system.

Data objects are hierarchically organized at run-time in a data tree. All information about the properties of the data required for rendering (e.g., orientation, position, etc.) is contained in the data tree. As shown in Fig. 7, the hierarchy shows the physical interrelationships and logical dependencies. Each data tree node can be assigned properties such as visibility. Different types of visualizations are realized by the corresponding dataset types to achieve the optimal display results. As display devices available in the medical application are two-dimensional, resliced, multiplanar reconstruction (MPR) [29] is used to display images (ImageData) in 2D mode. The lesion areas in ophthalmic images like exudation and artificial objects like painted areas or segmentations are

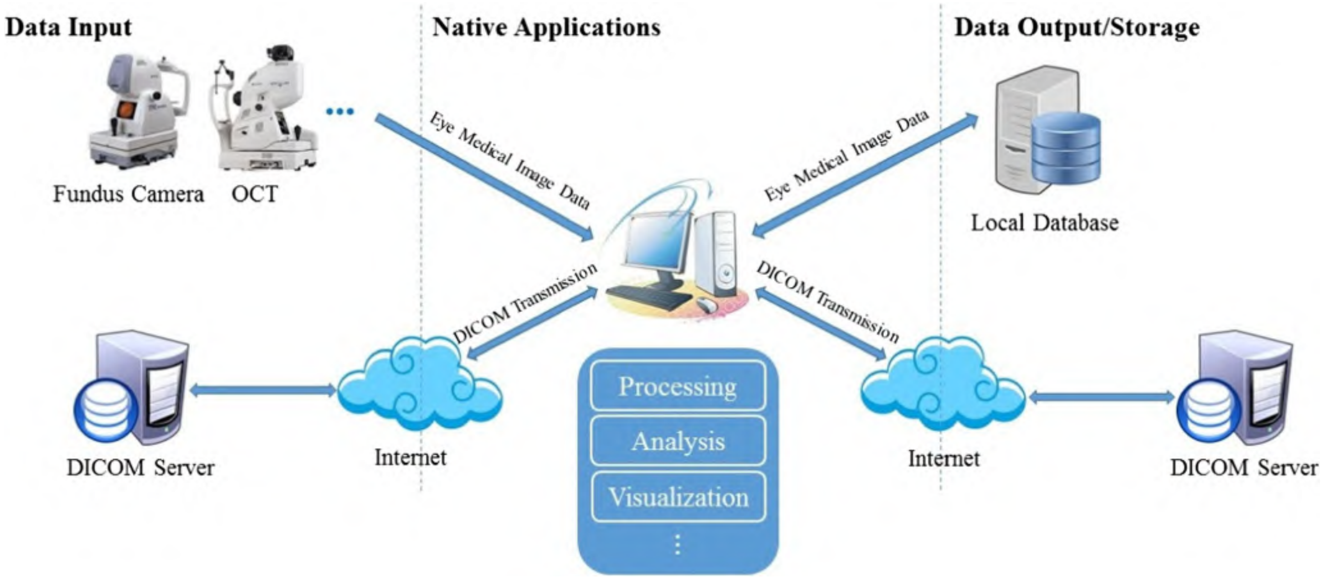


Fig. 5 The data flow in OIPAV system. The data from various instruments will be parsed to the standard format and the data in the DICOM Server can be transmitted into the system following the DICOM standard on the Internet. After processing, analysis, visualization, etc. on the client computer, the data are saved and exported to local database or DICOM server

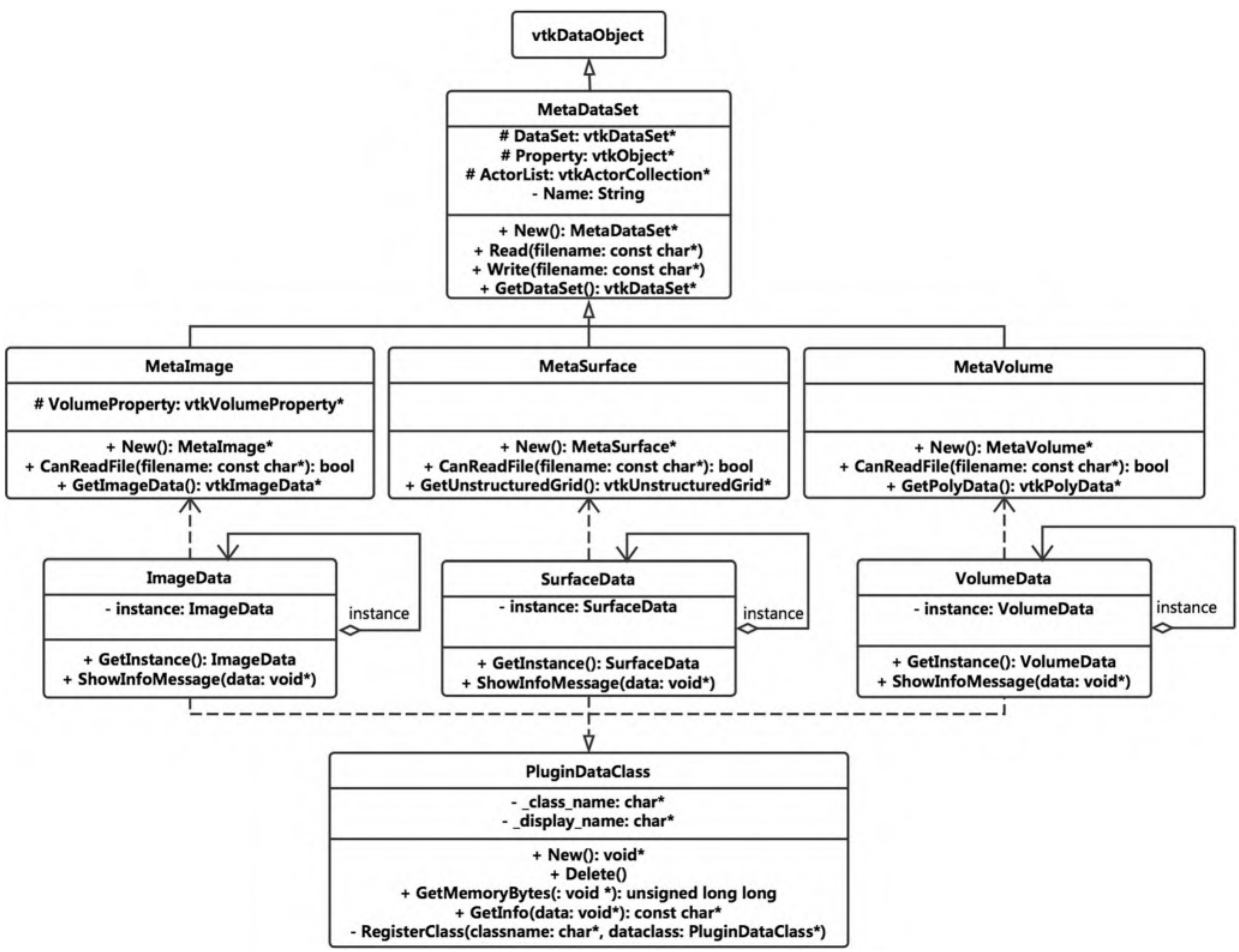


Fig. 6 The hierarchy of the standard data classes in the OIPAV system

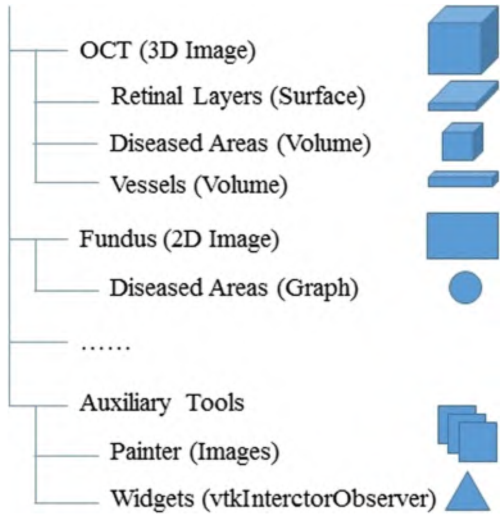


Fig. 7 Data tree to hierarchically organize data objects. All Information are associated to describe their spatiotemporal and rendering properties

often geometric structures and they are encapsulated into the VolumeData type. The 2D/3D interactive widgets are primarily derived from the vtkInteractorObserver class in VTK, which observe invoked events and support interactive manipulation of objects.

OIPAV provides sufficient hardware-accelerated volume rendering algorithms, including the classical ray-casting algorithm [30] and 3D texture mapping algorithms [31] to display these data in 3D mode. Data of some structures of the eye such as retinal layers are composed of point sets. The system reconstructs these layers into the SurfaceData from dispersed points and employs marching cube algorithms [32] to render surfaces in 3D mode.

In order to assist users to evaluate eye structures, OIPAV offers multiple display modes, including the optical intensity view, the fusion image of optical intensity and retinal layer thickness, and the 3D pseudo color image in the maximum intensity projection techniques. These views display ophthalmic image features more intuitively and help the clinician analyze the health of eyes. For example, the retinal RNFL thickness value is frequently used to evaluate glaucoma [33, 34] and retinal layers' intensities show a better performance in discriminating mild diabetic retinopathy and control eyes [35–37]. These views visualize the structural and optical information intuitively to meet the clinical needs.

Instead of building up single data object in each window, we implement the multi-tab viewer design to display and interact with a data tree or branches of a data tree in multiple viewer pages in the software window. Each viewer page contains a classic four-window layout, where the axial, sagittal, coronal and 3D windows are arranged in a grid pattern of resizable windows. This gives users a similar experience to browse web pages and facilitate the navigation through multi-groups images. Optimal display results can be achieved through tuning various property parameters in the tool panel including representation, color, lighting, and material.

Ophthalmic Image Processing and Interaction

In the ophthalmic image acquisition process, due to differences in acquisition conditions and characteristics of the imaging sensor, the ophthalmic images have uneven illumination, noise, and other problems. Raw ophthalmic image data is often unsuitable for processing directly and thus pre-processing is essential to facilitate the subsequent processing. Moreover, ophthalmic images have unique anatomical and pathological structures, which are different from other medical images. Many diseases' detection and evaluation are based on these structures. For example, in most cases, the retinal layer segmentation is essential for retinal diseases detection and evaluation [11]. Additionally, the lamina cribrosa (LC), a sieve-like structure, has been known to play an essential role in the physiopathology of glaucoma and has been investigated as a potential location to identify early glaucomatous damage [38, 39]. Therefore, some important pre-processing steps and especially the eye structure segmentation operations are required to prepare the data to build the model of the anatomical and pathological structures of the eyes for subsequent analysis.

Based on these considerations, our system provides extensive processing functionalities. The key and common image processing operations can be divided into six groups: geometric transformation, filtering, image algebra, image morphology, segmentation and registration. Especially in segmentation, our system provides a variety of functions of automatically recognizing and segmenting eye structures including the automated 3-D retinal 11-layer segmentation of macular OCT images [11], the identification of Bruch's membrane opening (BMO) reference plane, the segmentation of the anterior LC surface (the surface separating the prelaminar tissue and LC structure) [38, 39], as well as optic cup (OC) and optic disc (OD). We implement these automatic processing based on some algorithms including Graph-Cut [40], Support Vector Machine (SVM) [41], Adaboost [42], and Random Forest [43]. These automatic processing functions free users from time-consuming manual segmentation and reduce subjectivity.

All processing algorithms are implemented with the plugin mechanism and users can add new algorithms by developing their own plugins. The corresponding parameters are set in the GUI control widgets in the plugins. To keep an excellent user experience, the system applies the multithread technology for some time-consuming operations and presents a progress dialog to provide the user with visual feedback.

Interaction is an important part of medical image applications for interactive manipulations as well as manual

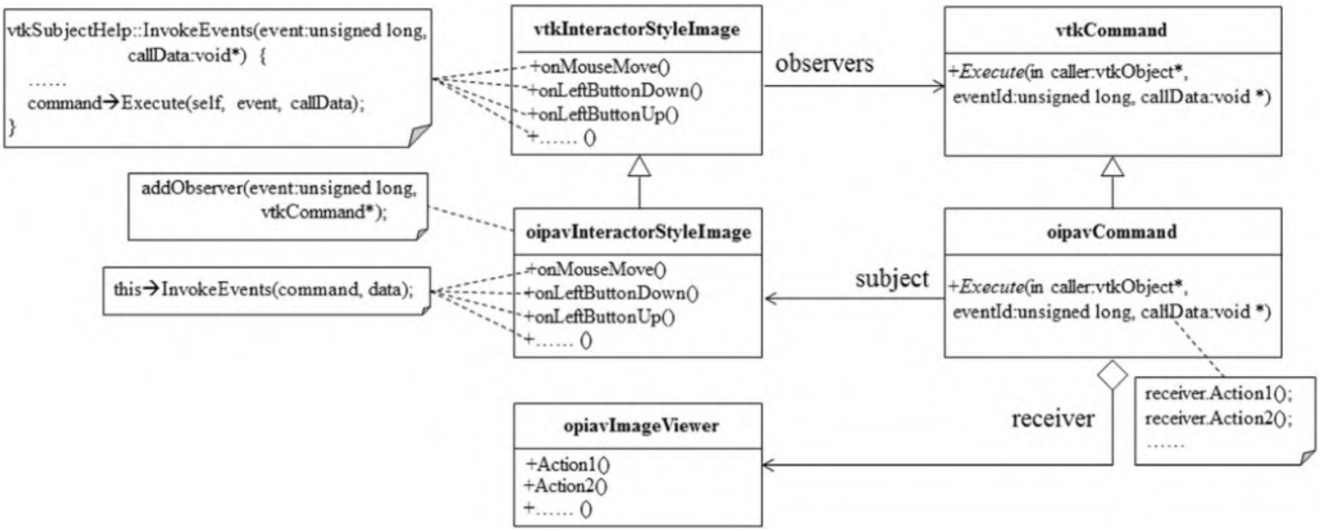
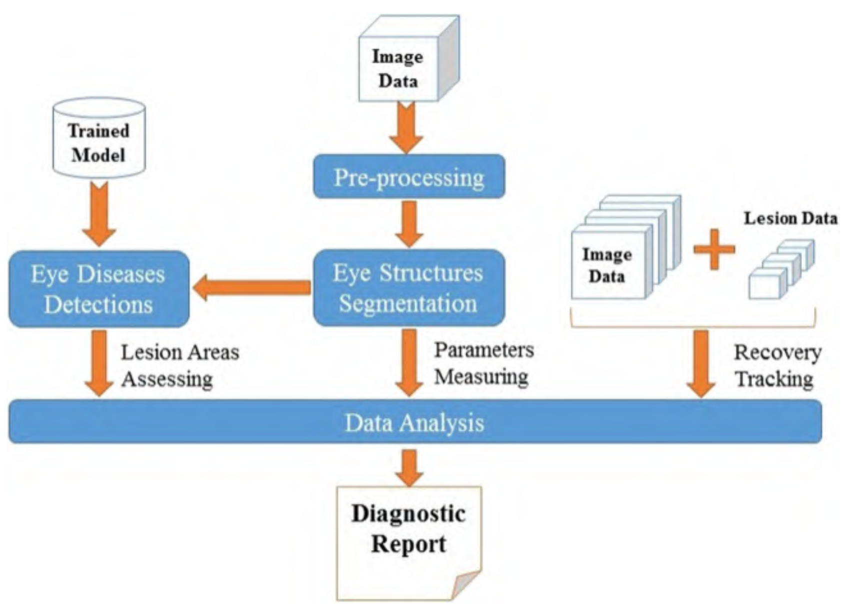


Fig. 8 The observer/command pattern used in the system

and semi-automatic processing like segmentation and registration. OIPAV follows the interaction concepts in VTK to structure and standardize the manipulation of data in 2D/3D views. Figure 8 shows the UML model of the observer/command pattern [44, 45] used in the system. The oipavInteractorStyleImage class and oipavCommand class are inherited from vtkInteractorStyleImage and vtkCommand in VTK respectively. We monitor various interactions like mouse events, keyboard events, focus events, and widget events from rendering windows in oipavInteractorStyleImage and load different interaction behaviors in oipavCommand. The oipavImageViewer class which is responsible for displaying images and other objects in 2D/3D scenes is the receiver in the command pattern and provides interactive manipulations tools to specific workflows that loaded in oipavCommand. Based on this design pattern, functions, OIPAV provides varieties of tools for conventional 2D/3D interactive manipulation to facilitate and improve the navigation and image manipulation, including image browsing, length/angle measurement, window width/level adjusting, zoom, pan, and manual painting/erasing. For ophthalmic images, the accurate segmentation and measurement of specific tissue and structure are crucial for diagnosis and analysis. Thus, when segmentations are inaccurate, manual adjustments are often required. OIPAV integrates some new interactive functions especially for ophthalmic images including manual extraction of a volume of interest (VOI), adjustment of retinal layers and relocation of BMO reference plane etc.

Fig. 9 The flowchart of the data analysis module in the system including automatic important parameters measuring, lesion areas assessing, recovery tracking, and diagnostic report generating. The orange arrows indicate the data flow



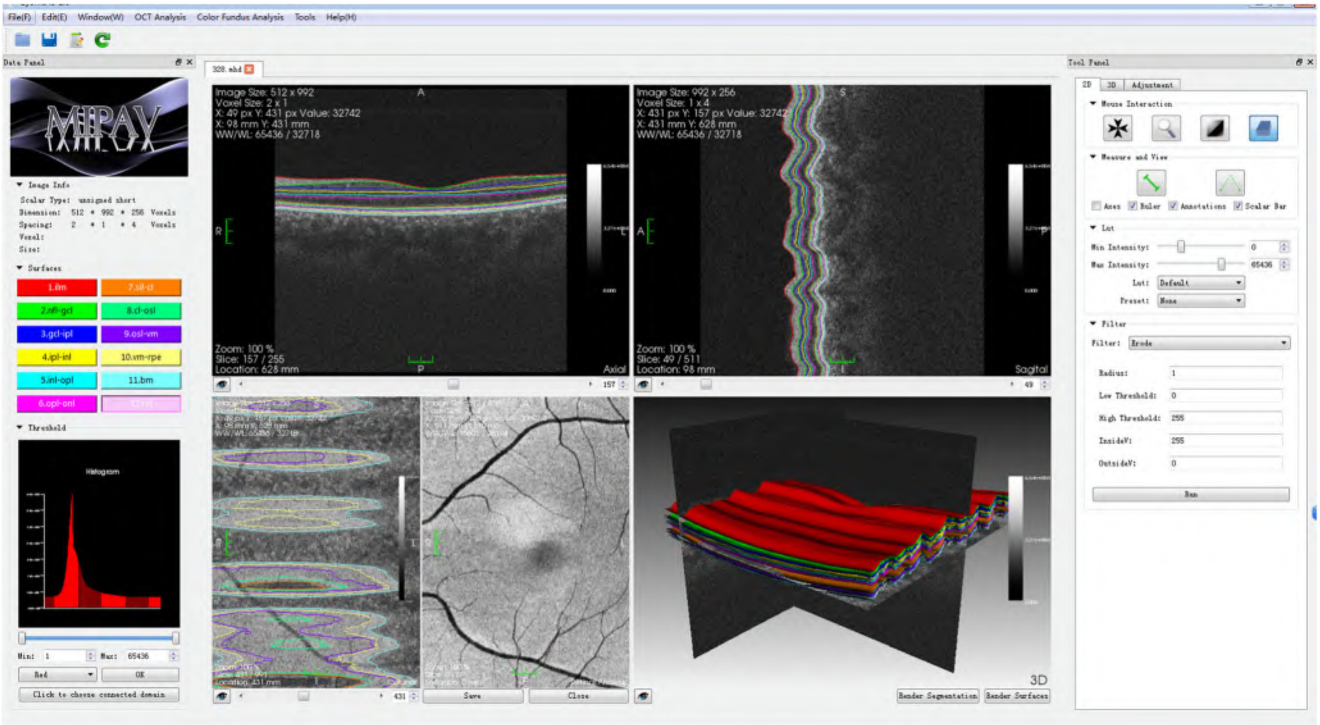
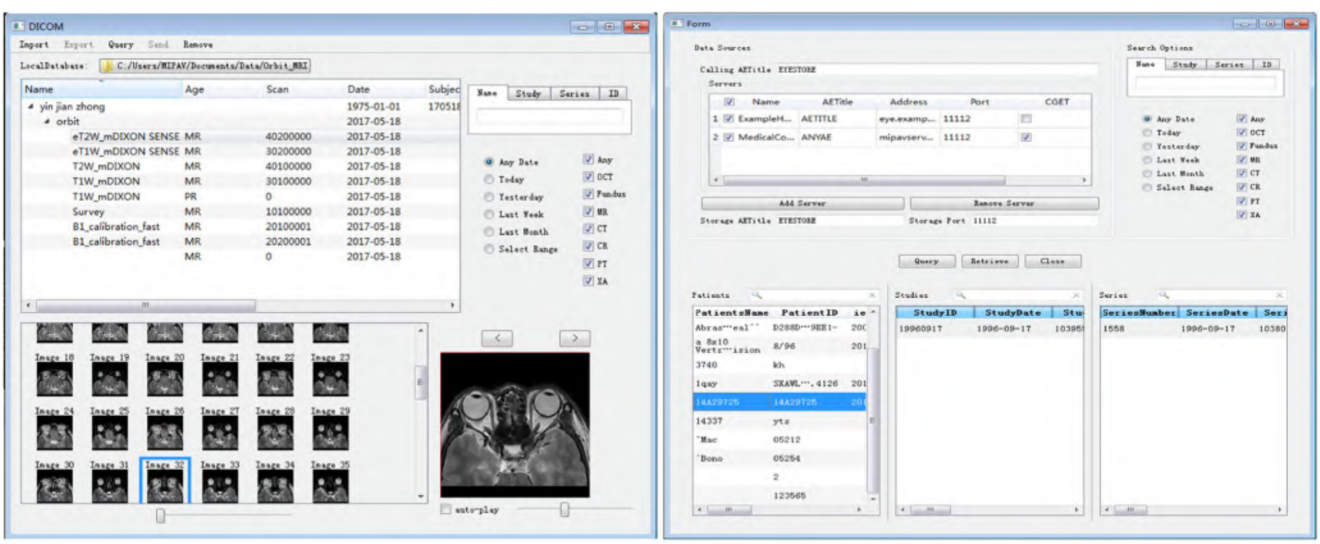


Fig. 10 The main GUIs of OIPAV. Every viewer page contains a classic four-window layout where the axial, sagittal, coronal and 3D windows are arranged in a grid pattern of resizable windows. On both sides there are tool panels that function as the interface of data management and manipulation. The picture above shows the views of retinal resliced and multiplanar reconstruction (MPR), retinal optical intensity and retinal layer surface rendering. The data is Topcon OCT image data

Ophthalmic Diseases' Detection and Analysis

Computer-aided detection (CADe) is one of the research that focuses in the medical applications and effective CADe systems speed up the medical diagnostic process, reduce diagnostic errors and improve quantitative evaluations [46]. Combining with our laboratory research experience of many years on ophthalmic diseases detection and analysis [11–22], we implement automatic detection algorithms for multiple ophthalmic diseases in the



(a) (b)

Fig. 11 DICOM database management and transmission. **a** Manage and browse data in local DICOM database. **b** Query and retrieve data from remote DICOM server

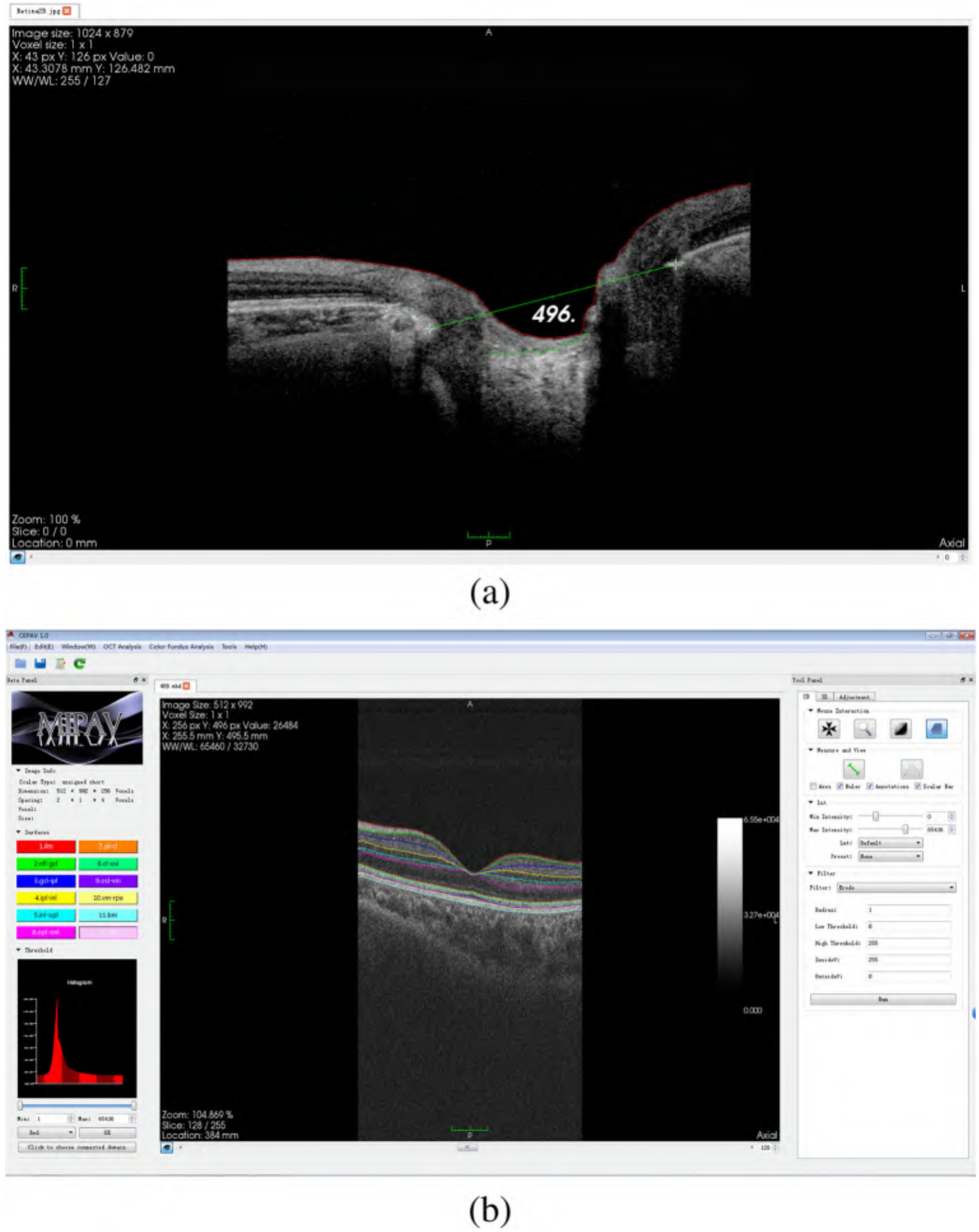
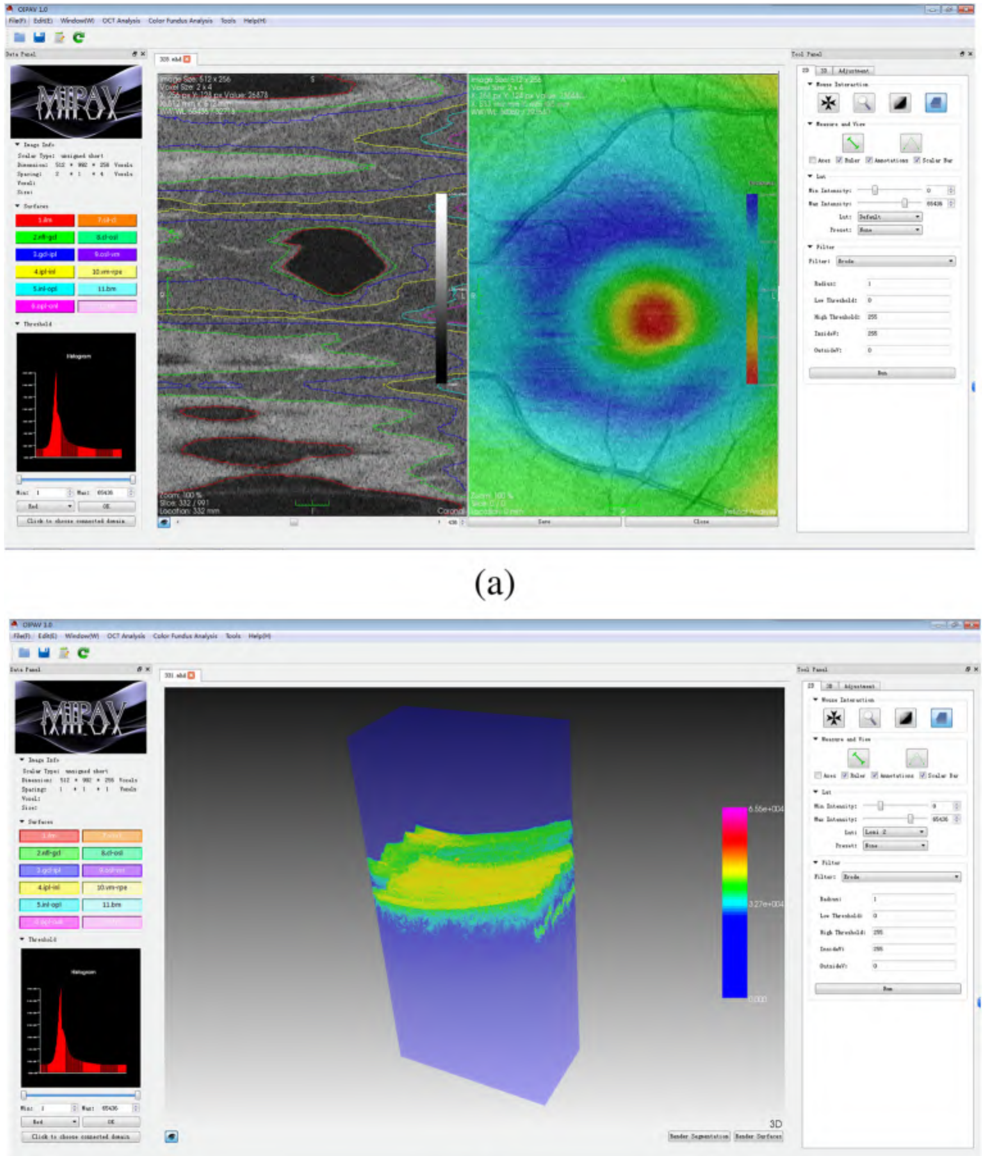


Fig. 12 Eye structures segmentation. **a** The 2D OCT image of ONH. The red curve represents the inner limiting membrane (ILM) surface and the green curve represents the anterior LC surface. The green line represents the BMO reference plane. **b** The macular OCT image retinal image with 11 retinal layers after our automated 3-D retinal 11-layer segmentation. Different color curves represent various retinal layer surfaces

system including branch retinal artery occlusion (BRAO), symptomatic exudate-associated derangements (SEAD), pigment epithelial detachment (PED), micro aneurysm, and exudation. All algorithms are pre-trained and without parameter optimization to provide the "one button" solutions. The 2D/3D lesion areas in ophthalmic images will be represented in internal standard data that is managed by the data tree and visualized for direct observation.

It is very important for the ophthalmologist to have reliable and efficient methods to quantitatively analyze ophthalmic data. The analysis module in OIPAV includes the following components: automatic important parameters measuring, lesion areas assessing, recovery tracking, and diagnostic report generating. We choose some key parameters to assess which are sensitive indicators of eye health. Based on the object system, OIPAV describes the geometric parameters and optical parameters of the individual objects and interrelationship among objects mainly in the optic nerve head (ONH) area. The geometric parameters include the anterior lamina cribrosa surface depth (ALCSD), the length of BMO reference plane, the rim area, the volume of optic cup, and cup/disc area ratio. Methods for various intensity-based measures are implemented in the system for analysis of optical characteristics. The quantitative assessments of lesion areas are fundamental to prevalence analysis and treatment. After eye disease detection processing, OIPAV will automatically



(a)

(b)

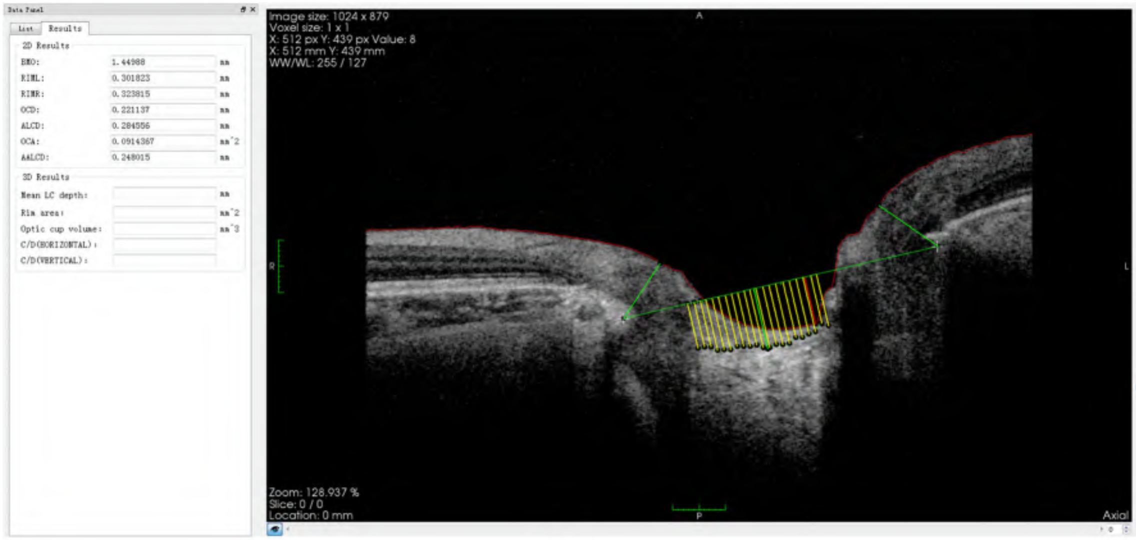
Fig. 13 Display modes for OCT images with 12 retinal layers. **a** The fusion image of optical intensity and retinal layer thickness. The different colors reflect and marks thickness between layers. **b** The 3D pseudo color retinal image in the maximum intensity projection technique

detect the data objects which represent the lesion areas and measure parameters including the number, area, volume etc. In clinical treatment, the eye recovery should be assessed quantitatively to tailor the therapy project such as the frequency of the medicine injections. Based on the multi-tab viewer architecture and using the data tree design, the system supports loading multiple sets of image of the same patient over time. Methods are implemented to track the lesion areas in the ophthalmic image at different times. The detailed statistics as well as the images of related 3D model will be shown in the diagram. That will greatly guide users by the amount of lesion areas over time and avoid substantial inconsistency in treatment resulting from estimating subjectively. All the analysis value is displayed real-time in the corresponding widgets.

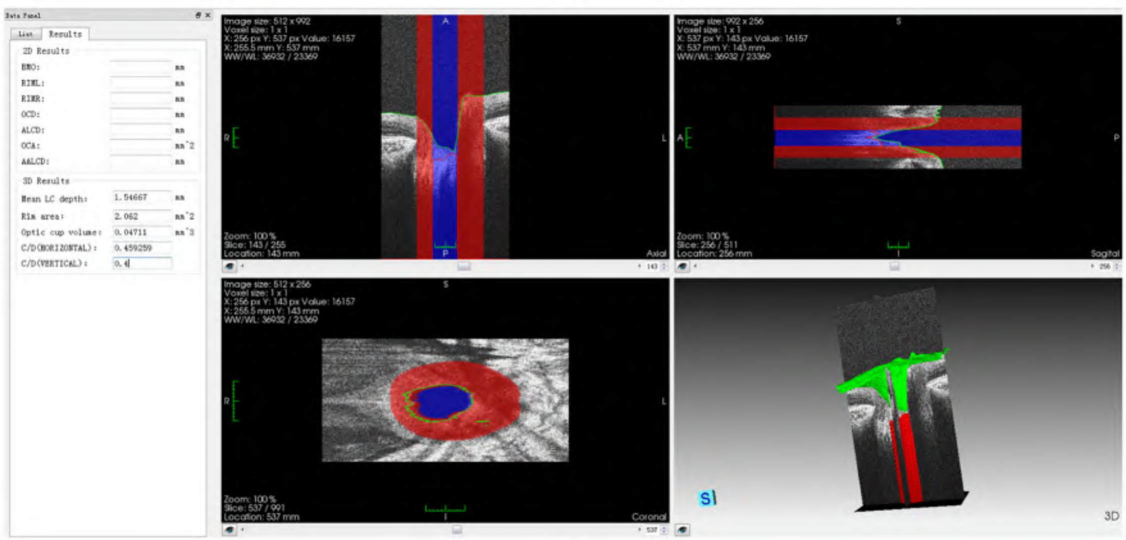
After the whole workflow of preprocessing, segmentation, disease detection, and data analysis, the auxiliary diagnostic reports which include all significant data and views will be generated as documents. Figure 9 shows the flowchart of the data analysis. The final reports are not designed to replace ophthalmologists to diagnose but help users master the overall situation of eyes quickly and accurately, free them from heavy work, and improve the diagnosis efficiency.

Experimental Results

The software system for ophthalmic image processing, analysis, and visualization was successfully developed. The main



(a)



(b)

Fig. 14 Examples of automatic measuring the important parameters of the ONH OCT images based on the localization of BMO reference plane and segmentation of anterior lamina cribrosa surface as well as retinal layers. The precise results are show in the left data panel. **a** Measurements of 2D OCT image. The red curve represents the ILM surface and green curve represents the anterior LC surface. **b** Measurements of 2D OCT image. The green and red curves represent the ILM surface and anterior LC surface. The blue area represents OC and red area represents OD

functions of six groups (data I/O, image processing, interaction, ophthalmic diseases detection, data analysis and visualization) were completed without problems. The ophthalmic image data tested mainly include 2D/3D OCT images, 2D colorful fundus images, and 2D/3D MRI images of various file formats. As a highly portable desktop system, OIPAV runs on three major operating systems: Windows, Linux, and Mac OS with a single code base. The software system is mainly tested on the following standard computer: two Intel Core i7-4790 3.60 GHz CPU, two 8 GB RAM, 1 TB Hard Disk, and a NVIDIA GTX 745 graphics. The following experimental examples presented demonstrate the effectiveness and practicability of the system.

Figures 10, 11, 12, 13, 14, 15, and 16 demonstrate some experiment examples using our system. Figure 10 shows the GUIs of OIPAV. Every viewer page contains a classic four-window layout. The axial, sagittal, coronal, and 3D windows are arranged in a grid pattern of resizable windows. On both sides, there are tool panels that function as the interface of data management and manipulation. The data in the figure is Topcon OCT data and the picture shows the views of retinal resliced and multiplanar reconstruction (MPR), retinal optical intensity and retinal layer surface rendering. Figure 11 shows the DICOM database management and transmission. Users can manage and browse data in local DICOM database or query and retrieve image data from remote DICOM server

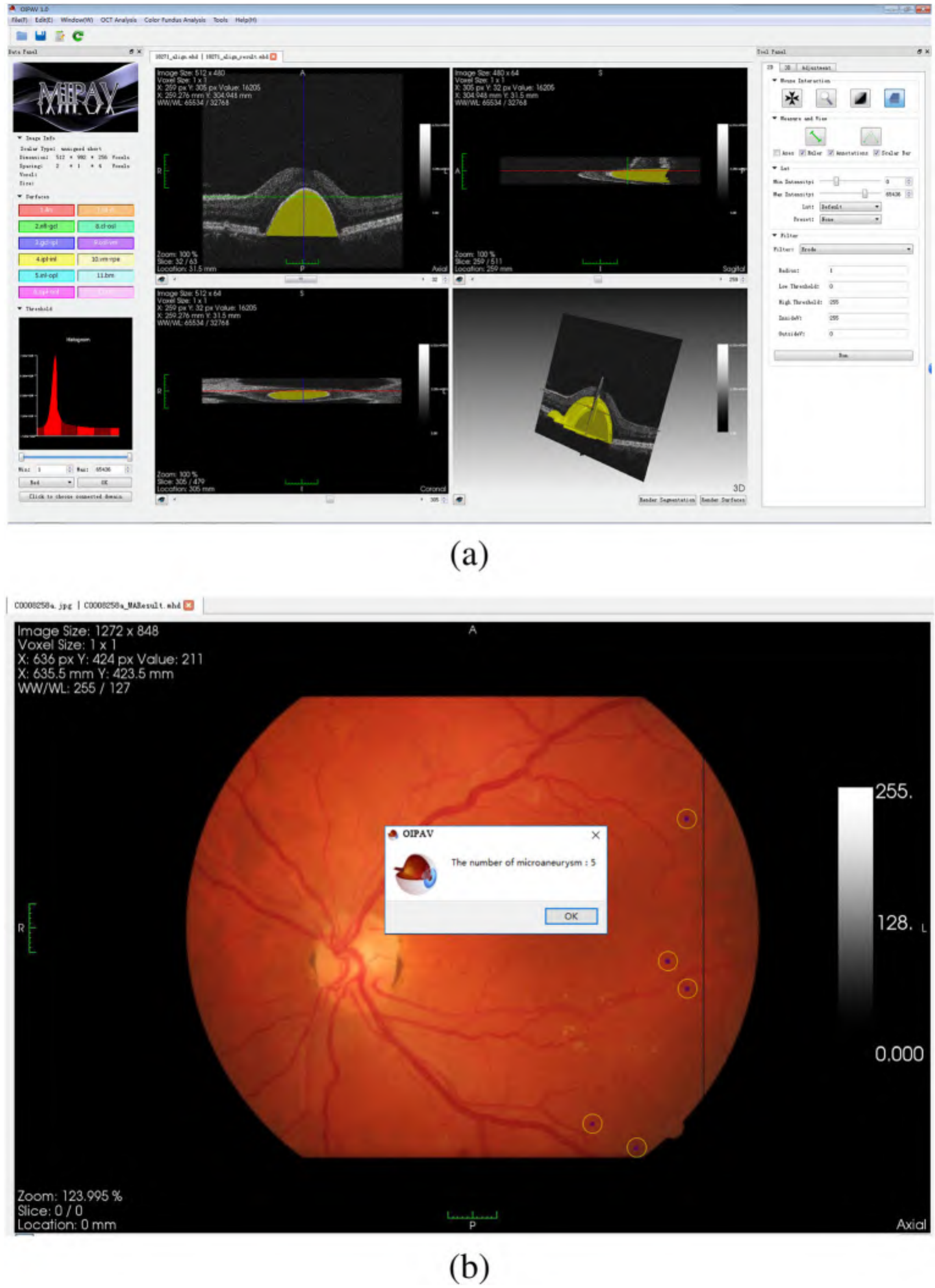


Fig. 15 The results of automatic eye diseases detection. **a** PED in the 3D OCT image. **b** Micro-aneurysms in the 2D colorful fundus image

by networks. After loading ophthalmic images in system, automatic structure segmentations are applied for subsequent analysis. Figure 12 shows the results of eye structure segmentation. With the data of this segmentation, the special view modes display the features of ophthalmic images more intuitively. Figure 13 shows the fusion image of optical intensity and retinal layer thickness, and the 3D pseudo color retinal image in the maximum intensity projection. Based on the results of structures segmentations, users can evaluate, detect and analyze ophthalmic diseases. Figure 14 shows automatic important parameters measuring of the 2D/3D ONH OCT images based on segmentations of eye structures. Figure 15 illustrates the results of automatic ophthalmic diseases detection on 2D colorful fundus images and 3D OCT images. Figure 16 illustrates the statistical chart of recovery tracking of choroid neovascularization (CNV) when loading serial images with diseases over time.

Currently, the system has been on trial in several relevant laboratories and ophthalmic hospitals. Most of users give positive evaluations of our work and think the software system has advantages in easy-using, functionality and practicability.

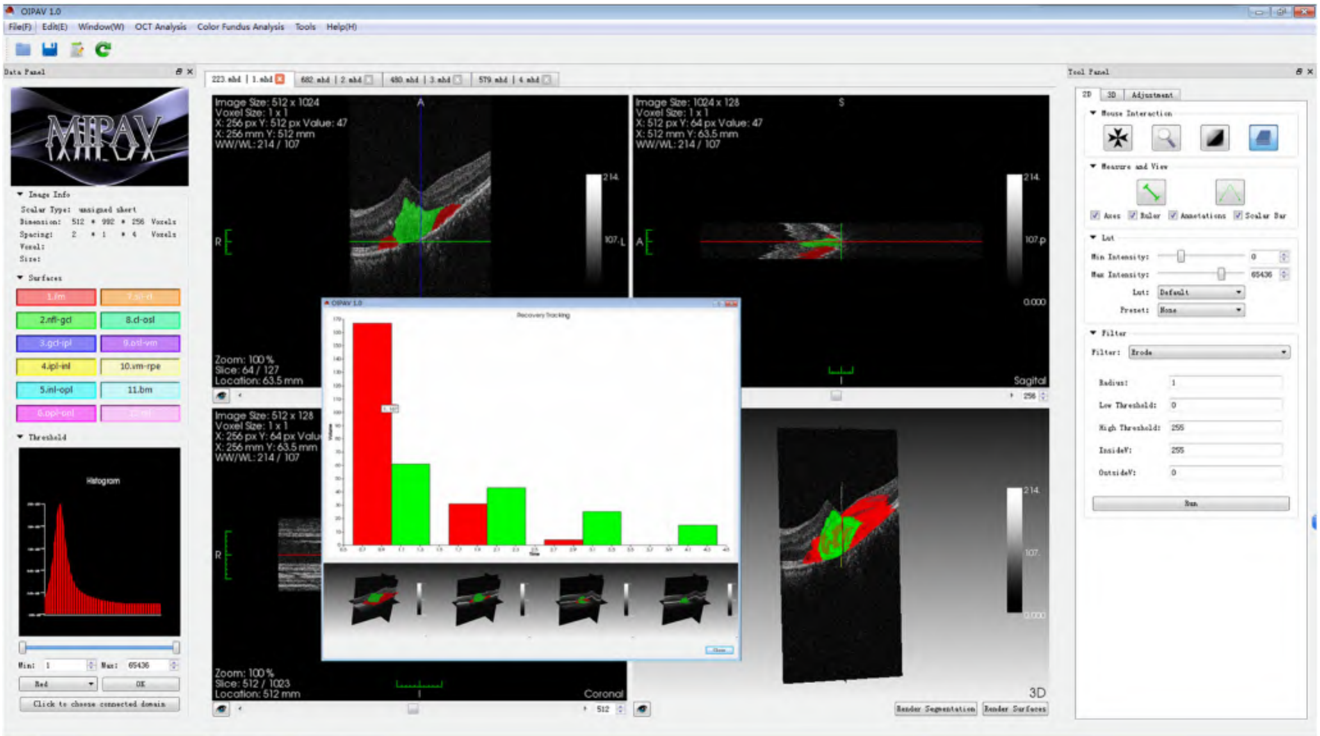


Fig. 16 An example of tracking recovery of CNV. The green area represents CNV area and the red represents sub-retinal fluid (SRF). The system contrasts different result images over time and showing change of volume in the chart to assist ophthalmologists plan clinical treatments

Discussion

As a cross-platform software system oriented for handling with ophthalmic images, OIPAV has been experimentally confirmed that it is applicable for various ophthalmic image processing, analysis and visualization. To cope with problems in the ophthalmic image filed including ophthalmic data I/O, structures segmentation, disease detection, and data analysis, OIPAV is built in the reasonable architecture and integrate rich and powerful targeted functionalities. Benefited from the module design method and plugin mechanism, OIPAV has features of flexible function configuration and easy extension. Furthermore, as a desktop application, the system has the advantages of stable, efficient, and good security. Meanwhile, client/server architecture leads to the obvious potential disadvantages as well such as inconvenient software upgrading and maintenance.

To the best of our knowledge, this is the first report on the introduction and design of the cross-platform software system specially oriented to ophthalmic images. Although the initial version of OIPAV achieves the intended functions, the system still needs to be improved. One weakness of this system is the lack of undo and redo functions. Currently, the initial version only supports restoring images to original states, which means it is often time consuming to correct a false instruction. We plan to implement and add the undo/redo functions based on the inverse command strategy [47] in the next version. Another issue is extending the range of the ophthalmic diseases detection algorithms. Now the number of ophthalmic diseases that supported detecting is still limited. In the next work, we will take more methods based on machine learning, such as deep learning to detect more ophthalmic diseases effectively.

Conclusion

In this paper, we have proposed a comprehensive cross-platform software system specially oriented for ophthalmic images. The main goal of OIPAV is to provide a platform to assist both researchers and clinicians in ophthalmic image processing, analysis and visualization. In the design of OIPAV, five main requirements are implemented:

- It provides various file formats support and local/remote database communications. Besides the DICOM support, it can deal with the ophthalmic data from different ophthalmic equipment manufacturers.
- It provides the comprehensive solutions for standardizing and visualizing multiple types of data in ophthalmic images. The design of the data tree and the multi-tab viewer facilitate the navigation through images. More targeted display modes show the ophthalmologic images features more intuitively.

- It provides functionalities to recognize and segment anatomical and pathological structures of the eyes automatically to prepare the data for subsequent processing and analysis.
- It provides the “one button” solutions for various eye disease detection with professional algorithms based on our laboratory previous work.
- It provides rich automatic data analysis functions to measure important parameters, assesse lesion areas, track recovery, and make auxiliary diagnostic reports. That helps users master situation of eyes accurately, reduce diagnostic errors and improve quantitative evaluations as well as diagnosis efficiency.

Moreover, the system has function integrity, operability, scalability, and other advantages. We believe that OIPAV possesses many key features that, taken together, make it a unique and valuable tool for the ophthalmologic imaging society.

We will consider providing a free trial version of the software by inviting other researchers in the near future at our website (http://mipav.net) for further improvements on reliability and practicability.

Funding Information This work has been supported in part by the National Basic Research Program of China (973 Program) under Grant 2014CB748600 and in part by the National Natural Science Foundation of China (NSFC) under Grants 81371629, 61401293, 61401294, 81401451, and 81401472.

References

1. Rosset A, Spadola L, Ratib O: OsiriX: an open-source software for navigating in multidimensional DICOM images. J Digit Imaging 17(3):205–216, 2004
2. Pieper S, Halle M, Kikinis R: 3D Slicer[C]. IEEE International Symposium on Biomedical Imaging: Nano To Macro. IEEE 1: 632–635, 2004
3. Fabian N, Moreland K, Thompson D, et al: The ParaView Coprocessing Library: A scalable, general purpose in situ visualization library. Large Data Analysis and Visualization. IEEE: 89–96, 2011
4. Schroeder WJ, Avila LS, Hoffman W: Visualizing with VTK: A tutorial[J]. IEEE Comput Graph Appl 20(5):20–27, 2000
5. Schroeder WJ, Martin KM, Lorensen WE: The design and implementation of an object-oriented toolkit for 3D graphics and visualization. Conference on Visualization. IEEE Computer Society Press, 1996, 93-ff
6. Ibanez L, Schroeder WJ, Ng L et al.: The ITK Software Guide: The Insight Segmentation and Registration Toolkit[J]. Comput Stat Data Anal 21:231–256, 2003
7. Nolden M, Zelzer S, Seitel A et al.: The Medical Imaging Interaction Toolkit: challenges and advances: 10 years of open-source development. Int J Comput Assist Radiol Surg 8(4):607, 2013
8. Wolf I, Vetter M, Wegner I, Nolden M et al.: The Medical Imaging Interaction Toolkit (MITK) – a toolkit facilitating the creation of interactive software by extending VTK and ITK. Proc SPIE 5367: 16–27, 2004
9. Muehler K, Tietjen C, Ritter F et al.: The Medical Exploration Toolkit: An Efficient Support for Visual Computing in Surgical Planning and Training[J]. IEEE Trans Vis Comput Graph 16(1): 133–146, 2010
10. Mildenberger P, Eichelberg M, Martin E: Introduction to the DICOM standard[J]. Eur Radiol 12(4):920, 2002
11. Shi F, Chen X, Zhao H, et al: Automated 3-D Retinal Layer Segmentation of Macular Optical Coherence Tomography Images. IEEE Trans Med Imaging Papers 34(2): 441–452
12. Chen X, Niemeijer M, Zhang L, et al: 3D Segmentation of Fluid-Associated Abnormalities in Retinal OCT: Probability Constrained Graph-Search-Graph-Cut. IEEE Trans Med Imaging Papers 31(8): 1521–1531
13. Sun Z, Chen H, Shi F, et al: An automated framework for 3D serous pigment epithelium detachment segmentation in SD-OCT images. Scientific Reports 6(298): 243–244
14. Wu B, Zhu W, Shi F, et al: Automatic detection of microaneurysms in retinal fundus images.[J]. Computerized Medical Imaging & Graphics the Official Journal of the Computerized Medical Imaging Society, 2016
15. Zhang L, Zhu W, Shi F, et al: Automated segmentation of intraretinal cystoid macular edema for retinal 3D OCT images with macular hole[C]// IEEE, International Symposium on Biomedical imaging. IEEE 1494–1497, 2015
16. Guo J, Zhu W, Shi F et al.: A Framework for Classification and Segmentation of Branch Retinal Artery Occlusion in SD-OCT.[J]. IEEE Trans Image Process 26(7):3518–3527, 2017
17. Zhu S, Fei S, Xiang D, et al: Choroid Neovascularization Growth Prediction with Treatment Based on Reaction-Diffusion Model in 3D OCT Images[J]. IEEE Journal of Biomedical & Health Informatics PP(99):1–1, 2017
18. Sun M, Zhang Z, Ma C, et al: Quantitative analysis of retinal layers on three-dimensional spectral-domain optical coherence tomography for pituitary adenoma[J]. Plos One in Press, 2017 https://doi.org/10.1371/journal.pone.0179532
19. Zhu W, Chen H, Zhao H et al.: Automatic Three-dimensional Detection of Photoreceptor Ellipsoid Zone Disruption Caused by Trauma in the OCT[J]. Sci Rep 6:25433, 2016
20. Kopriva I, Shi F, Chen X: Enhanced low-rank + sparsity decomposition for speckle reduction in optical coherence tomography[J]. J Biomed Opt 21(7):76008, 2016
21. Shi F, Tian B, Zhu W et al.: Automated choroid segmentation in three-dimensional 1-μm wide-view OCT images with gradient and regional costs[J]. J Biomed Opt 21(12):126017, 2016
22. Chen X, Zhu W, Wang L, et al: Automated Detection of IS/OS Defect Regions in 3D OCT Images. ARVO 2014
23. Eichelberg M, Riesmeier J: Ten years of medical imaging standardization and prototypical implementation: the DICOM standard and the OFFIS DICOM toolkit (DCMTK)[J]. Proc SPIE Int Soc Opt Eng 5371:57–68, 2004
24. Toussaint N, Sermesant M, Fillard P: vtkinria3d: A VTK Extension for Spatiotemporal Data Synchronization, Visualization and Management. In Proc. of Workshop on Open Source and Open Data for MICCAI 2007
25. Blanchette J, Summerfield M: C++ GUI Programming with Qt 4[J] 45(6): 747–749, 2006
26. Martin KW, Hoffman B: Mastering CMake : A Cross-Platform Build System[M]. Kitware 2010
27. Loeliger J: Version Control with Git: Powerful Tools and Techniques for Collaborative Software Development[D]. O'Reilly Media, Inc 2012
28. Huang HK: PACS and Imaging Informatics: Basic Principles and Applications[M] 2005
29. Eibel R, Türk T, Kulinna C et al.: [Value of multiplanar reformations (MPR)in multi-slice spiral CT of the lung][J]. Rofo 173(1): 57–64, 2001

30. Levoy M: Display of surfaces from volume data[J]. IEEE Comput Graph Appl 20(3):29–37, 1988
31. Cabral B: Accelerated volume rendering and tomographic reconstruction using texture mapping hardware[C]. Symp Onvisualization DBLP 91–98, 1995
32. Lorensen WE, Cline HE: Marching cubes: A high resolution 3D surface construction algorithm[C]//ACM siggraph computer graphics. ACM, 21(4):163–169, 1987
33. Gao E, Chen B, Yang J et al.: Comparison of Retinal Thickness Measurements between the Topcon Algorithm and a Graph-Based Algorithm in Normal and Glaucoma Eyes[J]. Plos One 10(6): e0128925, 2015
34. Hood DC, Anderson SC, Wall M, Kardon RH: Structure versus function in glaucoma: an application of a linear model. Invest Ophthalmol Vis Sci 48(8):3662–3668, 2007
35. Chen H, Chen X, Qiu Z et al.: Quantitative analysis of retinal layers' optical intensities on 3D optical coherence tomography for central retinal artery occlusion[J]. Sci Rep 5(10):9269, 2015
36. Chen B, Gao E, Chen H et al.: Profile and Determinants of Retinal Optical Intensity in Normal Eyes with Spectral Domain Optical Coherence Tomography[J]. Plos One 11(2):e0148183, 2016
37. Gao W et al.: Investigation of changes in thickness and reflectivity from layered retinal structures of healthy and diabetic eyes with optical coherence tomography. J Biomed Sci Eng 4:657–665, 2011
38. Belghith A, Bowd C, Medeiros FA, et al: Automated segmentation of anterior lamina cribrosa surface: How the lamina cribrosa responds to intraocular pressure change in glaucoma eyes[C]// IEEE, International Symposium on Biomedical Imaging. IEEE 222–225, 2015
39. Nadler Z, Wang B, Wollstein G et al.: Automated lamina cribrosa microstructural segmentation in optical coherence tomography scans of healthy and glaucomatous eyes.[J]. Biomed Opt Express 4(11):2596–2608, 2013
40. Kwatra V, Essa I, Turk G, et al: Graphcut textures: image and video synthesis using graph cuts[C]// ACM SIGGRAPH. ACM 277–286, 2003
41. Williams CKI: Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond[J]. IEEE Trans Neural Netw 16(3):781–781, 2005
42. Freund Y: Experiment With a New Boosting Algorithm[C]. Thirteenth Int Conf Mach Learn 148–156, 1996
43. Liaw A, Wiener M: Classification and Regression by randomForest[J]. R News 23(23), 2002
44. Gamma E, Helm R, Johnson R, et al: Design patterns: elements of reusable object-oriented software[J] 49(2):241–276, 1995
45. Kitware: The VTK user's guide[J]. 2010
46. Van GB, Schaefer-Prokop CM, Prokop M: Computer-aided diagnosis: how to move from the laboratory to the clinic[J]. Radiology 261(3):719, 2011
47. Conway RW, Schneider FB: User Recovery and Reversal in Interactive Systems[M]. Cornell University 1981