Progress Report for 2nd Quarter, Aug-OCT 2017

OCT Image Analysis System for Grading and Diagnosis of Retinal Diseases and its Integration in i-Hospital

Milestone 1: OCT image preprocessing and layer segmentation

A fully automated segmentation framework is proposed here that is used for the segmentation of retinal layers from OCT images. The proposed algorithm is based on three phases. In the first phase, the candidate input scan is denoised using non-linear edge preserving noise suppressing bilateral filter. The purpose of denoising is to remove acquisition artifacts and highlight retinal morphology. The second phase of proposed algorithm is related to extract retinal layers from candidate scan by computing coherent tensors at each orthogonal orientation. Afterwards, the highly coherent tensor is selected by measuring the degree of coherency within the tensor grid. The third phase of the proposed algorithm is related to estimating the missing data and tracing the retinal pathology.

Preprocessing:

An input OCT scan $I(s,t)$ is initially loaded into the proposed system where it is normalized to the common spatial resolution of 480x1280. Then it is de-noised using 2D adaptive low pass wiener filter. The reason for de-noising the candidate image is to increase the sparsity of intra-retinal pathology within $I(s,t)$. Wiener filter adaptively suppresses noisy outliers by measuring an average intensity of the surrounding pixels within the filtering kernel as expressed in Eq. (1-3):

$$\beth = \frac{1}{w_s w_t}\sum_{s_a \in w_s}\sum_{t_b \in w_t} I(s_a, t_b) \tag{1}$$

$$\aleph^2 = \frac{1}{\omega_s \omega_t}\sum_{s_a \in w_s}\sum_{t_b \in w_s} I^2(s_a, t_b) - \beth^2 \tag{2}$$

$$I_D(s_a, t_b) = \beth + \frac{\aleph^2 - \varphi^2}{\aleph^2}(I(s_a, t_b) - \beth) \tag{3}$$

Where $I_D(s_a, t_b)$ represent the sparsely strained pixel, w_s represents the row of a smoothing window, w_t represents the column of the smoothing window, $\beth$ represents the localized mean within the kernel, $\aleph^2$ represents the localized variance within the kernel and φ^2 is the mean of all $\aleph^2$ kernels [23].

Retinal Layers Segmentation:

In order to segment retinal pathology from the candidate scan, a 2nd order structure tensor grid is computed in our proposed system that takes a candidate de-noised scan $I_D(s,t)$ and generate its partial derivatives at the orientation of 0 and $\pi/2$ radians. Since the gradients are computed along two predominant orientations so these gradients are fused together to generate four possible tensors as expressed mathematically in Eq. (4 to 7)

$$\Im(s,t) = \begin{bmatrix} \Gamma_{SS}(s,t) & \Gamma_{ST}(s,t) \\ \Gamma_{TS}(s,t) & \Gamma_{TT}(s,t) \end{bmatrix} \tag{4}$$

$$\Gamma_{SS}(s,t) = \sum_{s_i \in \zeta_x}\sum_{t_j \in \zeta_j} \xi(s_i, t_j)\, \Delta_{SS}(s - s_i, t - t_j) \tag{5}$$

$$\Gamma_{ST}(s,t) = \Gamma_{TS}(s,t) = \sum_{s_i \in \zeta_x}\sum_{t_j \in \zeta_j} \xi(s_i, t_j)\, \Delta_{ST}(s - s_i, t - t_j) \tag{6}$$

$$\Gamma_{TT}(s,t) = \sum_{s_i \in \zeta_x}\sum_{t_j \in \zeta_j} \xi(s_i, t_j)\, \Delta_{TT}(s - s_i, t - t_j) \tag{7}$$

Where $\Im(s,t)$ represents a 2nd order structure tensor matrix containing all possible tensors among two predominant orientations, $\Gamma_{SS}(s,t), \Gamma_{ST}(s,t), \Gamma_{TS}(s,t)$ and $\Gamma_{TT}(s,t)$ represents the convolution sum of gradient products at the respective orientation [24]. The gradient products (Δ_{SS}, Δ_{ST}, Δ_{TS} and Δ_{TT}) are mathematically expressed in Eq. (8-10):

$$\Delta_{SS} = (I_{D_{SS}}(s,t))^2 \quad (8)$$

$$\Delta_{ST} = I_{D_S}(s,t).I_{D_T}(s,t) \quad (9)$$

$$\Delta_{TT} = (I_{D_{TT}}(s,t))^2 \quad (10)$$

In order to smoothen each tensor within the tensor grid, a localized Gaussian window $\xi(s,t)$ is computed that is convolved with the gradient products. Out of these tensors, a highly coherent tensor is obtained which has the maximum coherency ($\mathfrak{H}$). $\mathfrak{H}$ is computed using Eq. (11):

$$\mathfrak{H} = \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}\right)^2 \quad (11)$$

Where λ_1 and λ_2 represents the eigenvalues of partial derivatives computed along 0 and $\pi/2$ radians. The computed tensors are shown in Figure 1.

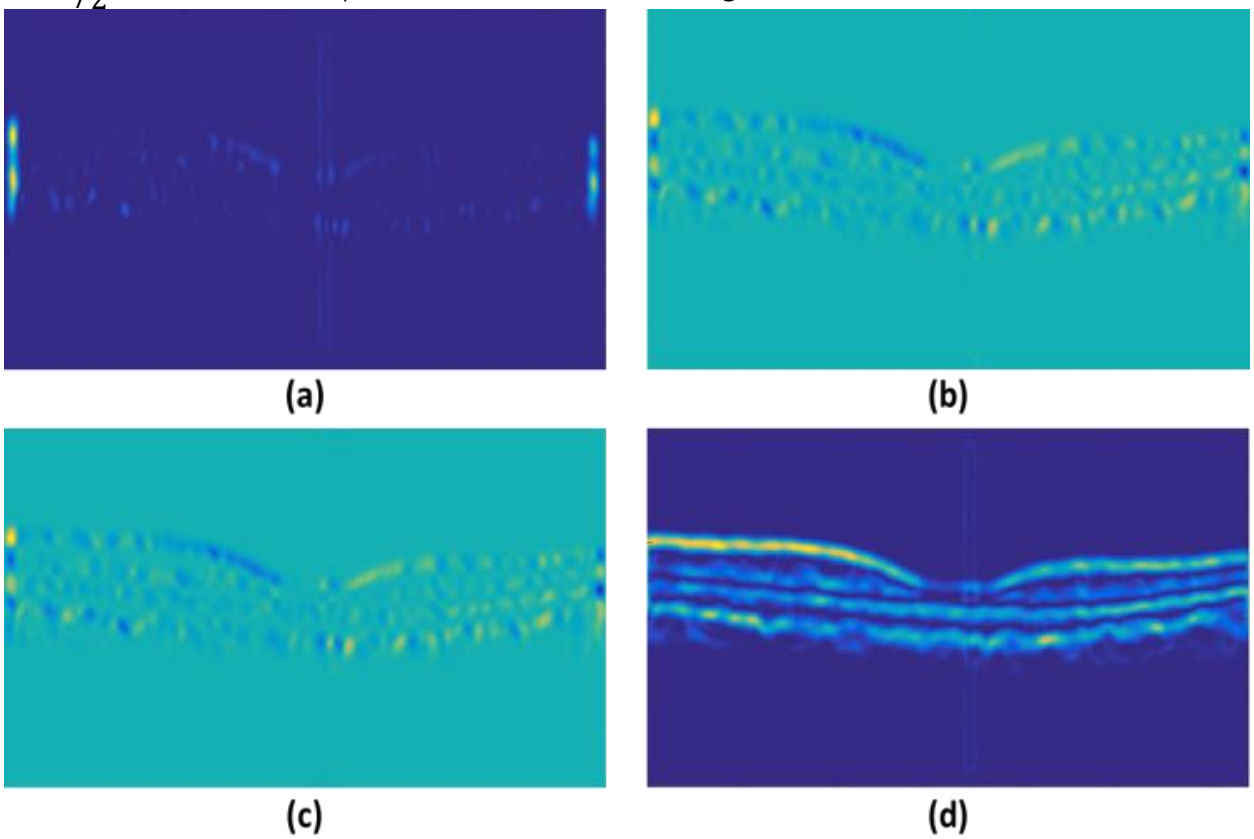


Figure 1: 2nd order structure tensor grid (a) tensor computed through the dot product of horizontal gradient, (b) tensor computed through the dot product of horizontal and vertical gradients, (c) tensor computed through the dot product of vertical and horizontal gradients, (d) tensor computed through the dot product of vertical gradient.

After extracting the highly coherent tensor $I_C(s,t)$, the binary map $I_B,(s,t)$ of $I_C(s,t)$ is computed using Otsu algorithm [25]. The digitalized retinal map is also shown in Figure 2. Afterwards retinal layers are extracted from digitalized map $I_B,(x,y)$ by computing retinal edges using canny edge detection [26]. The retinal layers edges are often thick and contains more

foreground pixels for each A-scan which causes incorrect layer tracing. For this purpose, the proposed framework skeletonized the retinal edges through medial axis transformation. The skeletonization is also shown in Figure 3.

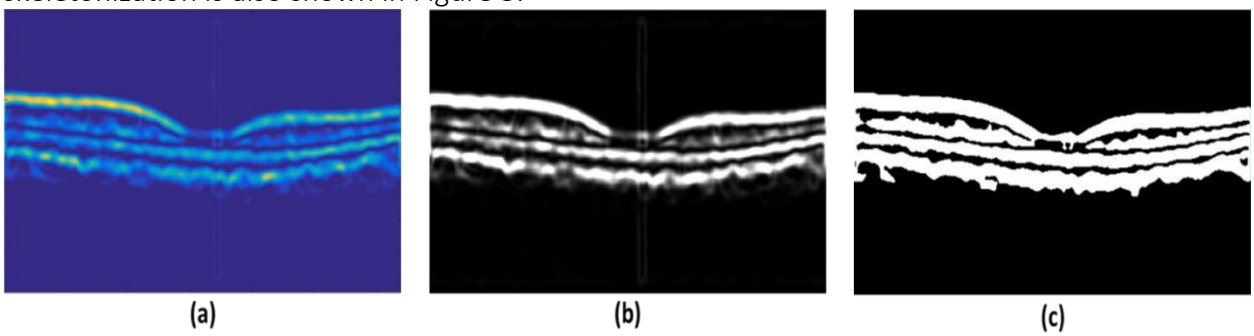


Figure 2: Segmented retinal and choroidal layers: (a) highly coherent 2D structure tensor $I_C(s,t)$, (b) binary map $I_B,(s,t)$ of highly coherent tensor, (c) canny edge detection of retinal and choroid layer

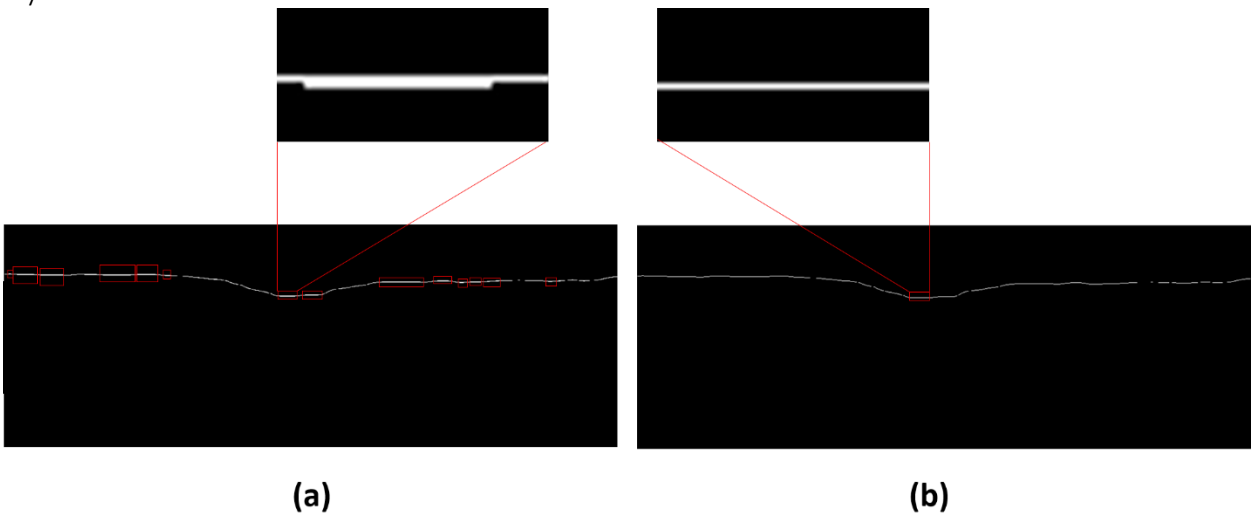


Figure 3: Retinal Layers Skeletonization: (a) ILM edges, (b) Skeletonized ILM

Retinal Layers Estimation and Tracing:

The third phase of proposed segmentation framework is related to accurately extracting the retinal morphology. For this, the proposed framework traces each layer iteratively. The tracing algorithm works in a way that it first decomposes the candidate image into an undirected graph where each pixel corresponds to a node and adjacent nodes are connected to each other via 4-neighbor connectivity. The algorithm then automatically initializes the seed points as the image top and bottom rows and generate binary maps for each retinal layer. At each iteration, the seed points traverses to nearest node by measuring the intensity differences. If the intensity difference between two or more nodes is same then top seed points give priority to downward nodes. Similarly, the bottom seed points give priority to upward nodes. When the top or bottom seed point observe a transition between foreground and background pixel then they include that pixel into the respective layer map and change the foreground pixel to background. The whole algorithm converges when the initialized seed points becomes equal. The working of the proposed tracing algorithm is also illustrated in Figure 4 and it is used to automatically extract the retinal

layers morphology. Afterwards, the extracted layer points are joined together by traversing the respective layer map from left to right and connecting adjacent 8-connectivity based neighboring pixels. Missing layer information in each map as well as the incorrect pickings are automatically corrected by fitting Savitzky-Golay polynomials.

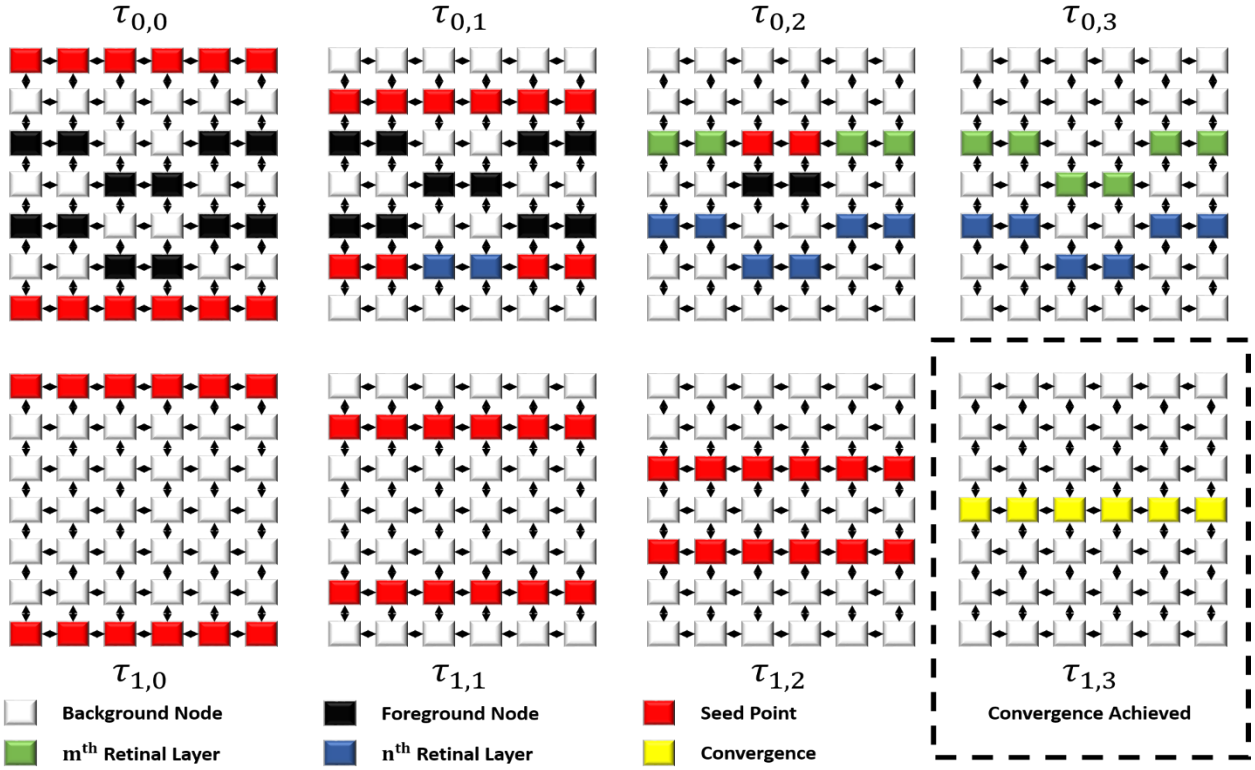


Figure 4: Proposed Retinal Tracing Algorithm

Savitzky-Golay is a smoothing filter that smooth out each retinal layer by convolving its successive subsets with the lower order polynomial coefficients through least squares method. Each retinal layer is mapped as a function $f(z_p)$ that consists of n data points where each data point is located at coordinate z_p as expressed below:

$$z_p = [z_i, z_j]^T, \quad p = 1, \dots, n \tag{12}$$

$$z = \frac{z_i - \overline{z_i}}{\Delta} \tag{13}$$

$$\Delta = \frac{z_{n-1} - z_o}{n} \tag{14}$$

$$\hat{f}(z_p) = \sum_{j} \sum_{i=-\frac{m-1}{2}}^{\frac{m-1}{2}} C_i \, f(z_p)_{j+i}, \quad \frac{m-1}{2} \le j \le n - \frac{m-1}{2} \tag{15}$$

where $\hat{f}(z_p)$ is the estimated layer, $\overline{z_i}$ is the mean value of independent variable z_i, Δ is the step size, C represents the convolutional coefficients and m denotes the order of fitted polynomial. The convolutional coefficients C are computed as:

$$C = (J^T J)^{-1} J^T \tag{16}$$

$$J = \begin{bmatrix} 1 & z_j & z_j^2 & z_j^3 & z_j^4 & \cdots & z_j^{m-1} \\ 1 & z_{j+1} & z_{j+1}^2 & z_{j+1}^3 & z_{j+1}^4 & \cdots & z_{j+1}^{m-1} \\ 1 & z_{j+2} & z_{j+2}^2 & z_{j+2}^3 & z_{j+2}^4 & \cdots & z_{j+2}^{m-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & z_{j+(m-2)} & z_{j+(m-2)}^2 & z_{j+(m-2)}^3 & z_{j+(m-2)}^4 & \ddots & \vdots \\ 1 & z_{j+(m-1)} & z_{j+(m-1)}^2 & z_{j+(m-1)}^3 & z_{j+(m-1)}^4 & \cdots & z_{j+(m-2)}^{m-1} \end{bmatrix}_{j=\frac{1-m}{2}} \quad (17)$$

and

$$J^T J = \begin{bmatrix} m & \sum z_j & \sum z_j^2 & \sum z_j^3 & \sum z_j^4 & \cdots & \sum z_j^m \\ \sum z_{j+1} & \sum z_{j+1}^2 & \sum z_{j+1}^3 & \sum z_{j+1}^4 & \sum z_{j+1}^5 & \cdots & \sum z_j^{m+1} \\ \sum z_{j+2}^2 & \sum z_{j+2}^3 & \sum z_{j+2}^4 & \sum z_{j+2}^5 & \sum z_{j+2}^6 & \cdots & \sum z_j^{m+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum z_{j+(m-1)}^{j+m} & \sum z_{j+(m-1)}^{j+m} & \sum z_{j+(m-1)}^{j+m} & \sum z_{j+(m-1)}^{j+m} & \sum z_{j+(m-1)}^{j+m} & \cdots & \sum z_j^{m+m} \end{bmatrix}_{j=\frac{1-m}{2}} \quad (18)$$

After extracting the retinal layers, they are used to measure retinal thickness profiles to discriminate between different types of macular pathology. The proposed framework can extract up to four retinal layers from retinal scans having different macular syndromes. One of the randomly selected scan along with its extracted layers is shown in Figure 5.

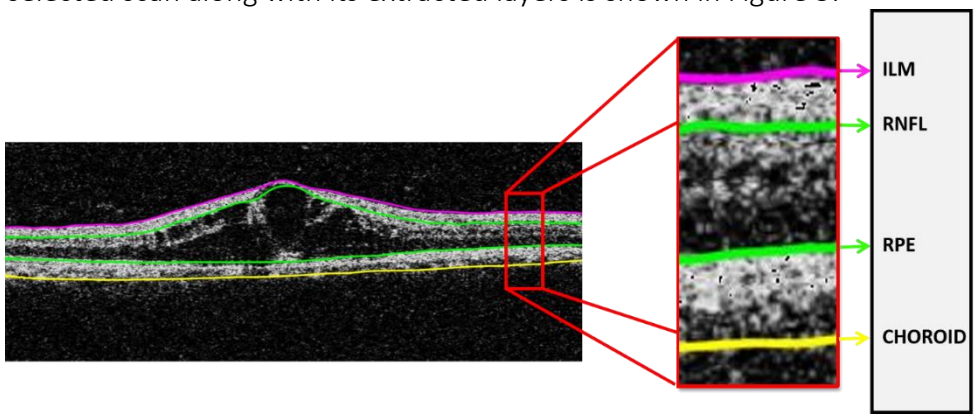


Figure 5: Segmented retinal layers: ILM, retinal nerve fiber layer (RNFL) and choroid.

3D Retinal Surfaces:

After extracting ILM, RNFL, RPE and Choroidal layer from B-scans within OCT volumes, they are used to reconstruct 3D retinal surfaces. Each retinal surface is generated by first smoothing it through low pass Gaussian filter and then taking a mean of 8 consecutive profiles for accurate representation of retinal morphology. The thickness surface is also generated by taking the absolute difference between Choroidal and ILM layers.

Results:

The proposed framework was tested on a custom prepared dataset that has been acquired by Armed Forces Institute of Ophthalmology (AFIO) and it has been validated by multiple expert

ophthalmologists. The detailed description about the dataset is presented in first quarter report. Apart from this, the retinal layers extracted by the proposed framework was compared with the manual annotations and some of the randomly selected samples are shown in Table 1.

Table 1: Mean Retinal Layer Differences

Layers	ILM	RNFL	RPE	Choroid
AMD	0.0045	0.0110	0.0954	0.0990
	0.0594	0.0498	0.0707	0.0838
	0.0559	0.0397	0.0904	0.1039
	0.0541	0.0461	0.0226	0.0310
	0.0460	0.0352	0.0597	0.0733
Mean	**0.0440**	**0.0364**	**0.0678**	**0.0782**
STD	**0.0202**	**0.0136**	**0.0260**	**0.0259**
ME	0.0102	0.0111	0.0756	0.0725
	0.0122	0.0151	0.0614	0.0615
	0.0247	0.0203	0.0662	0.0632
	0.0518	0.04	0.0861	0.0736
	0.0102	0.0111	0.0756	0.0725
Mean	**0.0247**	**0.0216**	**0.0723**	**0.0677**
STD	**0.0166**	**0.0111**	**0.0095**	**0.0054**
Healthy	0.0500	0.0526	0.1552	0.1102
	0.0479	0.0219	0.1816	0.1609
	0.0546	0.0614	0.1695	0.1416
	0.0443	0.0237	0.1729	0.1522
	0.0555	0.0392	0.126	0.1024
Mean	**0.0505**	**0.0398**	**0.161**	**0.1335**
STD	**0.0042**	**0.0156**	**0.0195**	**0.0231**

Table 2 shows the retinal thickness profiles that have been computed by taking the absolute difference between the respective layers. We can observe from Table 2 that for ME cases the retinal thickness between ILM and RPE as well as the retinal thickness between ILM and Choroid is quite high as compared to others. Also for AMD cases, the thickness profile of RPEDC section is quite discriminating as compared to ME and healthy subjects. Figure 12 shows 5 randomly selected OCT scans of healthy, AMD and ME pathology onto which extracted retinal layers are overlaid. It can be seen from Figure 12 that the proposed framework is quite robust in extracting retinal layers from healthy pathology even in poor resolution scans. Figure 13 shows the reconstructed 3D surface of ILM, RNFL, RPE and Choroid for the randomly selected healthy subject. It can be observed from Figure 13 that the proposed framework is quite robust in evaluating retinal surface especially the representation of foveal section. Furthermore, the proposed framework extracts the objective visualization of retinal thickness by taking the absolute difference between ILM and choroid as shown in Figure 14.

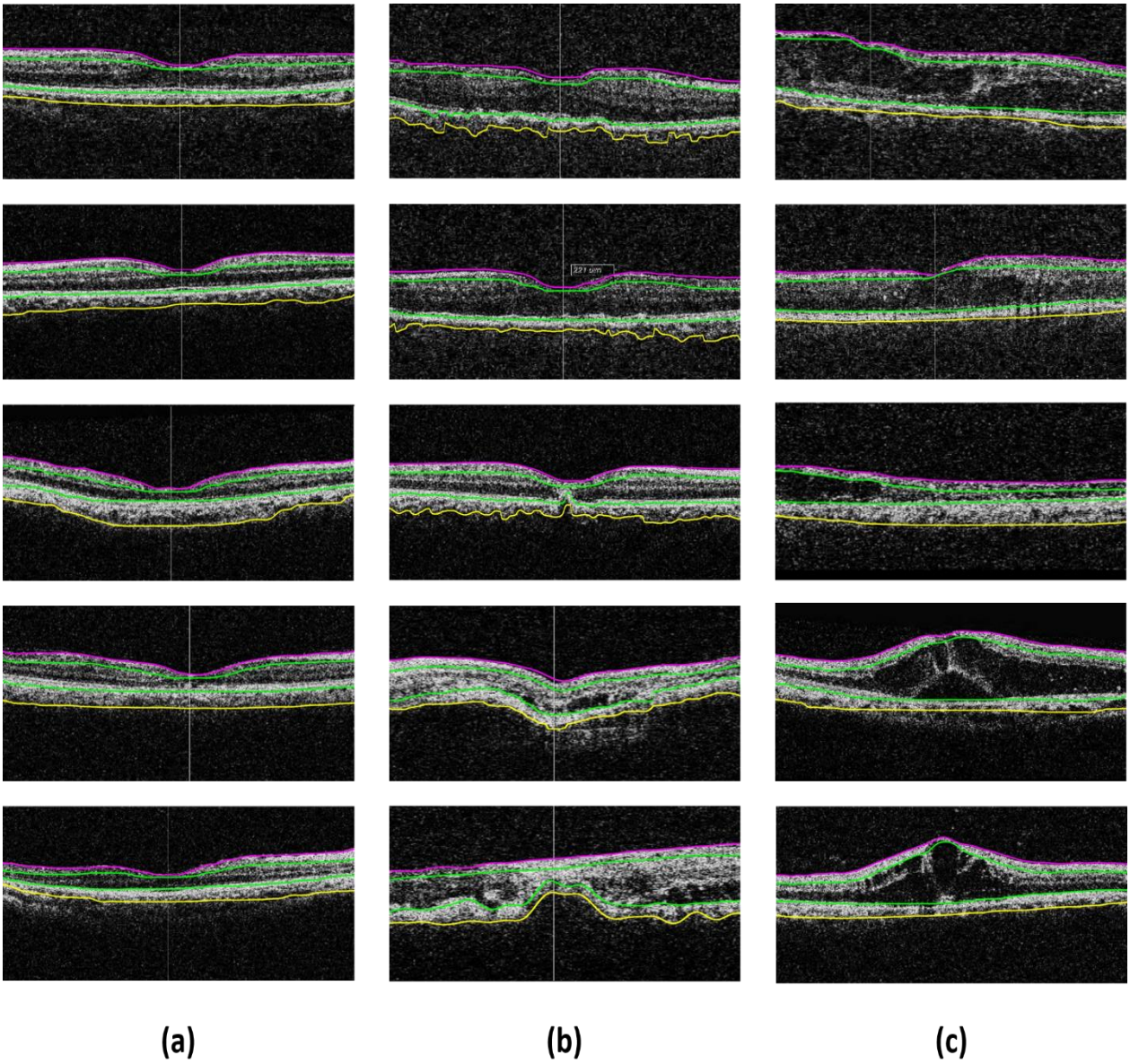


Figure 12: Segmented Layers from (a) Healthy, (b) AMD and (c) ME affected subjects

Table 2: Retinal Thickness Profiles

Profiles	ILM-Choroid	ILM-RNFL	RPE-Choroid	ILM-Choroid
AMD	148.2	25.7	27.5	120.7
	156.3	21.7	27.1	129.2
	173	22.4	26.9	146.1
	178.2	25.5	33.4	144.8
	163	24.6	31.1	131.8
Mean	**163.7**	**24.0**	**29.2**	**134.5**
STD	**10.90**	**1.60**	**2.60**	**9.700**
ME	204.4	24.4	28.4	176
	176	31.5	28.3	147.7

	140.3	36.2	48.1	92.2
	206.8	28.2	25.3	181.5
	198.4	27.8	30.1	168.3
Mean	**185.2**	**29.6**	**32.0**	**153.1**
STD	**24.90**	**4.00**	**8.20**	**32.60**
Healthy	149.1	25.2	32.8	116.3
	143.8	28.1	38.3	105.5
	153.6	21.2	42	111.6
	150.2	30.9	39.9	110.4
	138.5	28.3	36.2	102.3
Mean	**147.0**	**26.7**	**37.8**	**109.2**
STD	**5.300**	**3.30**	**3.10**	**4.900**

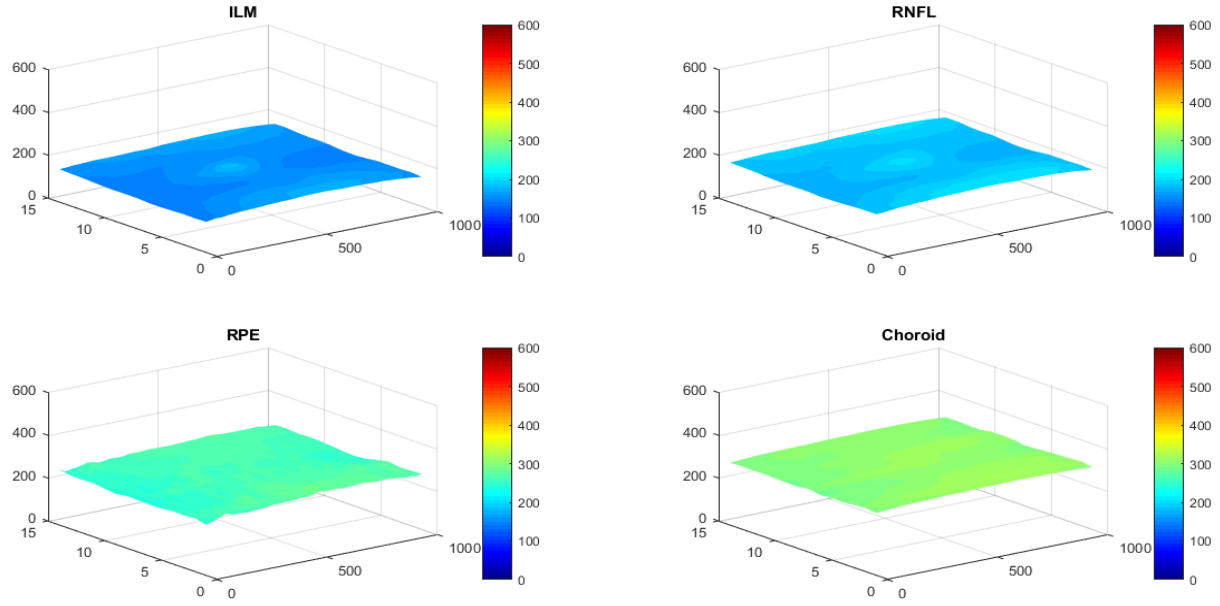


Figure 13: Reconstructed Retinal Surfaces

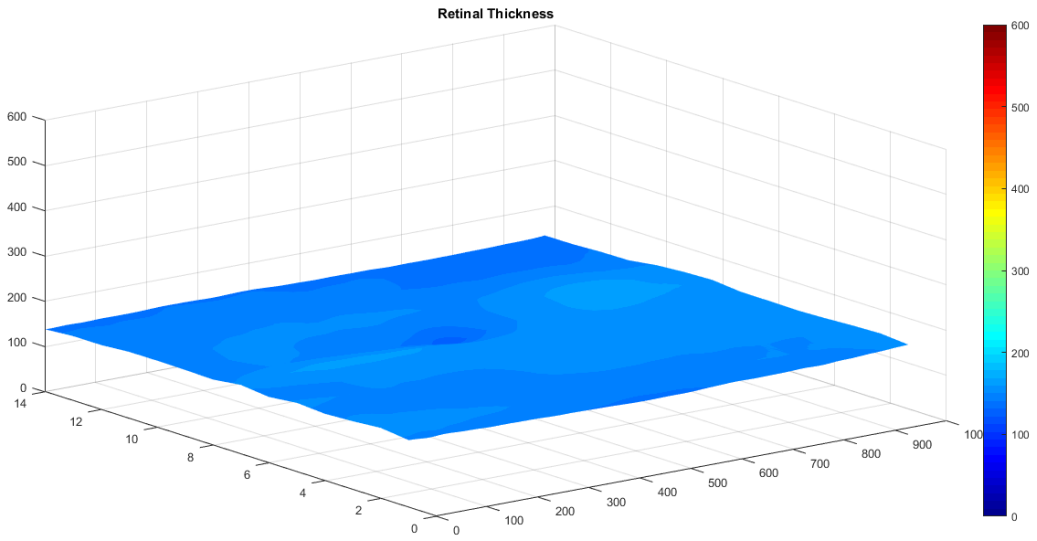


Figure 14: Retinal Thickness Surfaces

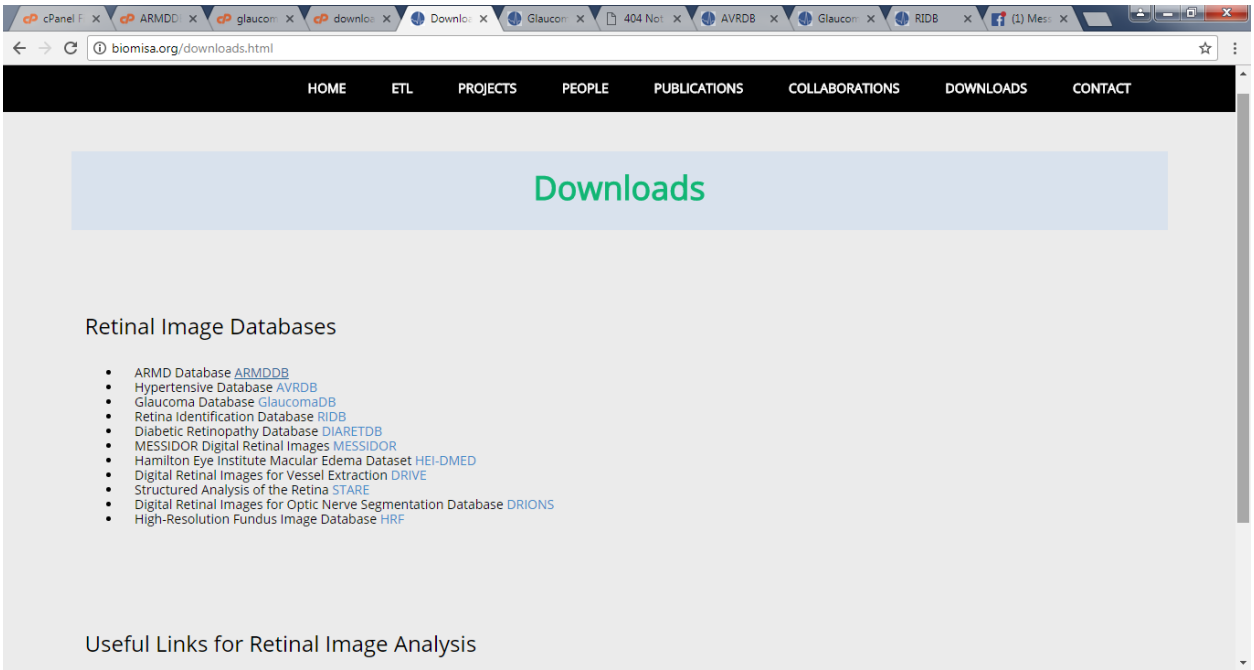
Deliverable: MATLAB API for Preprocessing and Layer Segmentation

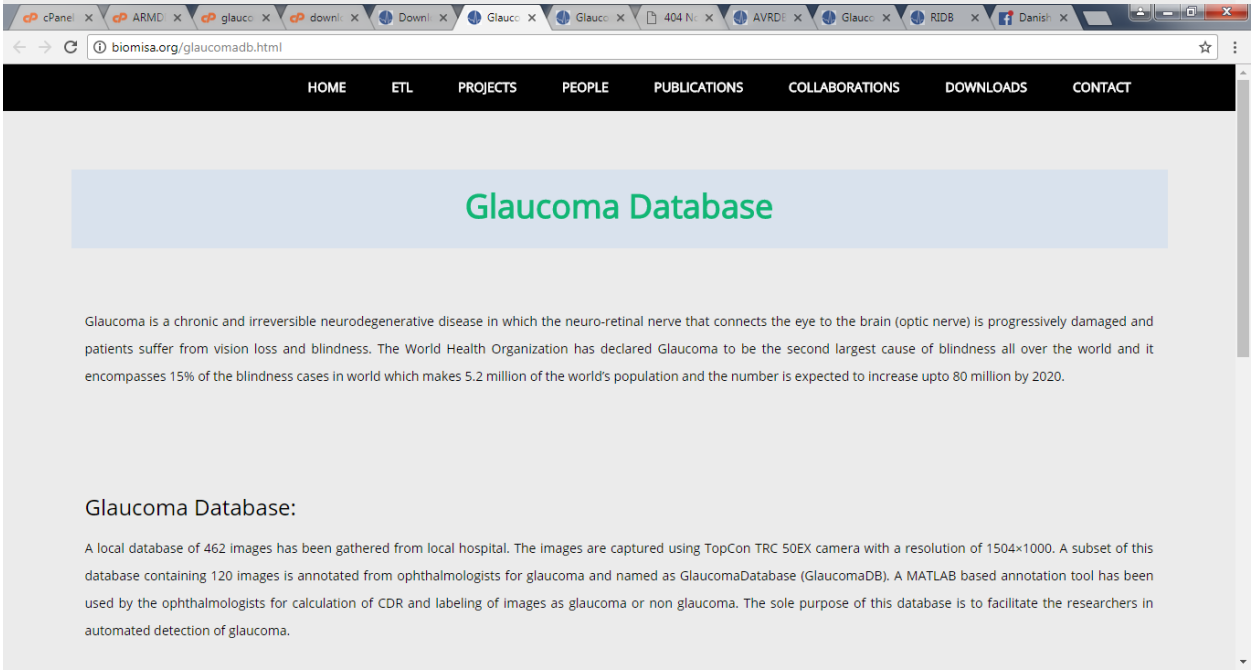
Results: The matlab module for preprocessing and layer segmentation is complete.

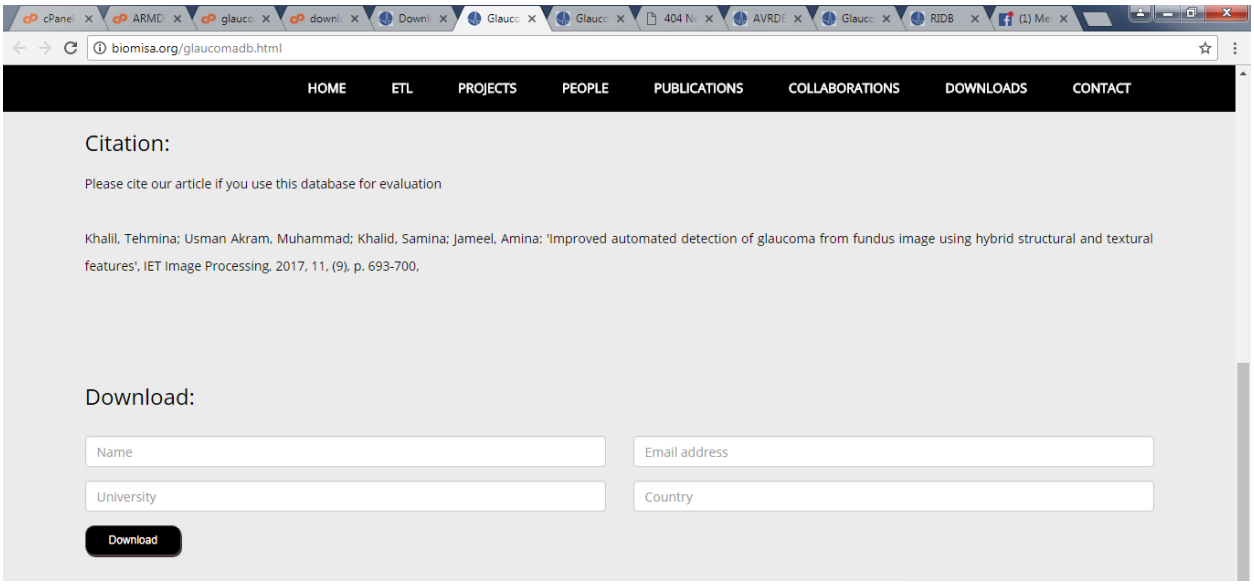
Milestone 2: Webpage designing for Benchmark dataset including registration and download pages and also designing license sheet

Different webpages have been designed to download datasets collected and annotated in this project. Links are given below
http://biomisa.org/glaucomadb.html
http://biomisa.org/avrdb.html
http://biomisa.org/ARMDDB.html

Screen shots for web pages are as following


biomisa.org/downloads.html
HOME
ETL
PROJECTS
PEOPLE
PUBLICATIONS
COLLABORATIONS
DOWNLOADS
CONTACT
Downloads
Retinal Image Databases
ARMD Database ARMDDB
Hypertensive Database AVRDB
Glaucoma Database GlaucomaDB
Retina Identification Database RIDB
Diabetic Retinopathy Database DIARETDB
MESSIDOR Digital Retinal Images MESSIDOR
Hamilton Eye Institute Macular Edema Dataset HEI-DMED
Digital Retinal Images for Vessel Extraction DRIVE
Structured Analysis of the Retina STARE
Digital Retinal Images for Optic Nerve Segmentation Database DRIONS
High-Resolution Fundus Image Database HRF
Useful Links for Retinal Image Analysis


biomisa.org/glaucomadb.html
HOME
ETL
PROJECTS
PEOPLE
PUBLICATIONS
COLLABORATIONS
DOWNLOADS
CONTACT
Glaucoma Database
Glaucoma is a chronic and irreversible neurodegenerative disease in which the neuro-retinal nerve that connects the eye to the brain (optic nerve) is progressively damaged and patients suffer from vision loss and blindness. The World Health Organization has declared Glaucoma to be the second largest cause of blindness all over the world and it encompasses 15% of the blindness cases in world which makes 5.2 million of the world's population and the number is expected to increase upto 80 million by 2020.
Glaucoma Database:
A local database of 462 images has been gathered from local hospital. The images are captured using TopCon TRC 50EX camera with a resolution of 1504×1000. A subset of this database containing 120 images is annotated from ophthalmologists for glaucoma and named as GlaucomaDatabase (GlaucomaDB). A MATLAB based annotation tool has been used by the ophthalmologists for calculation of CDR and labeling of images as glaucoma or non glaucoma. The sole purpose of this database is to facilitate the researchers in automated detection of glaucoma.



A sample license sheet for Glaucoma Dataset is as following

LICENSE AGREEMENT
FOR NON-COMMERCIAL RESEARCH USE

Effective as of date:
_____
Affiliation of LICENSEE:
_____
_____
_____
Having offices at Address of
LICENSEE:_____
_____
_____
(Hereinafter "LICENSEE") in consideration of the mutual covenants contained herein, the parties, intending to be legally bound hereby, hereto agree as follows:

1. LICENSOR developed certain valuable intellectual property known as GLAUCOMADB and desires to grant a license to LICENSEE. LICENSOR do not warrant to LICENSEE for the database.
2. LICENSEE has no rights with respect to the database or any portion of it and thus shall not use the database except as expressed in this Agreement.
3. Subject to the TERMS and CONDITIONS of this Agreement, LICENSOR hereby grants to LICENSEE for non-commercial research use only, for an initial period of 2 years starting at the effective date above mentioned renewable upon the discretion of LICENSOR. A royalty-free, non-exclusive, non-transferable, license subject to the following conditions:

    3.1 The Database is only for the non-commercial research use of LICENSEE and, in a need to know basis, of those direct research colleagues who belongs to the same research institution as LICENSEE and have adhered to the TERMS of this license.

3.2 The Database will not be copied nor distributed in any form other than for backup of LICENSEE.

3.3 The Database will only be used for research purposes and will not be used nor included in commercial applications in any form (e.g. original files, encrypted files, files containing extracted features, etc.)

3.4 The LICENSEE should cite the article "Khalil, Tehmina; Usman Akram, Muhammad; Khalid, Samina; Jameel, Amina: 'Improved automated detection of glaucoma from fundus image using hybrid structural and textural features', IET Image Processing, 2017, 11, (9), p. 693-700,"

_____

SIGNATURE & STAMP of LICENSEE

Deliverable: Online Benchmark dataset

Results:. Web page is functioning and researcher across the globe can download these datasets for research purposes.

Milestone 3: Design of a general purpose database to save different types of data based on SRS

Designing of a General Purpose Health Informatics Database:

Storage of Medical Data Information must follow the common terminology being used in medical science, take an example of a medical encounter that take place in a hospital between a patient and a care provider at a certain location. During an encounter a care provider makes certain observations about a patient, like vital signs. An observation can be viewed as an enquiry or a question that a care provider is interested in finding answer to, which directly corresponds to the patient's medical state. Patient's temperature, upon finding, turns out to be 98 degree Fahrenheit, thus being an answer to the enquiry/question posed by the care provider.

To record encounters and observations made in them, a certain standard is followed that is represented on the Patient Charting; Paper Forms, having mandatory and optional fields which must be filled by the care provider. These fields may include the date and time, location, patient's name, address, visit and appointment information along with the observations.

Medical Standards Organizations, impose these standards to be practiced. Electronic medical record systems must incorporate those standards when recording patient health information.

Coming up with a health informatics platform that not only fulfills the purpose of being general purpose but also ensuring the standards, is a task that requires a considerable amount of research and development done closely with the various medical institutions.

One such effort has been successfully put up by the OpenMRS, an open source community which provides a medical informatics platform that has evolved into supporting health care delivery and research in nearly every continent.

OpenMRS and its Information Model

OpenMRS is an electronic medical record system (EMR) platform, designed for use in the developing world and first established in 2004.

The system is designed to be usable in very resource-poor environments and can be modified with the addition of new data items, forms and reports without the need to write complicated application code. It is intended as a platform that organizations can adopt and modify, avoiding the need to develop a system from scratch.

The Information Model:

Data

The actual information you want to record in OpenMRS is called **Data**. Examples of Data in OpenMRS are Patients, Encounters, and Observations. To support this data, and describe its meaning, you need additional **Metadata**.

When a user deletes a piece of data in OpenMRS, the information actually remains in the database. It is marked as **voided**, so that it will not show up in the interface, but it is not immediately deleted from the database. If a user deletes a piece of data by accident, an administrator canunvoid it to return it to the system. To permanently delete data from the database, an administrator must **purge** that data. Typically, this should never be done in a production system.

Metadata

The fundamental expectation of OpenMRS's design is that you will customize it for your clinical program's use case. The system has no built-in idea of the patient's weight or seeing the patient in an outpatient visit. Instead, you can configure these things yourself, to match your project's workflow. Generally speaking, the things that you need to configure in order to describe the real patient information you will be capturing are referred to as **metadata**. An example of a piece of metadata is a Location that represents a hospital.

An administrator may also **retire** metadata in OpenMRS. This does not mean that the metadata is deleted, but rather that it is not intended to be used going forward. Old information that refers to the retired metadata remains valid. An administrator may **unretire** metadata if it becomes relevant to active use again. If no actual data refers to a piece of metadata, an administrator may **purge** the metadata to permanently remove it from the database.

For example, the hospital you refer patients to closes. Therefore, you can no longer refer patients there. This Location can now be retired in OpenMRS. This would not invalidate the fact that many patients were referred there in the past.

Concepts and concept dictionary

The most important part of the system's metadata is the **Concept Dictionary**, which is a list of all the medical and program-related terms that you will use as questions and answers in Observations. This dictionary does not need to be complete when you begin using OpenMRS. You should expect new terms to be added and old terms to be retired as your use of the system evolves. It is better to start with a pre-populated Concept Dictionary, rather than starting from scratch yourself. See the chapter "Sharing Concepts and Metadata" for more details.

Every question you ask about a patient needs to be defined by a **Concept**. (For example, to record a patient's weight you need a concept like **Weight in kilograms**.)

If you want to ask a question that has a fixed set of coded answers, those answers are also Concepts. (For example, the question concept **Blood Type** may have 4 different answer concepts: **A, B, AB,** and **O**)

Persons

Every individual who is referred to in a patient record in OpenMRS is stored in the system as a **Person**. These include Patients, any patient relative or caretaker, Providers, and Users.

All Persons have these characteristics.

Names

A person can have one or more names, one of which must be marked as the **preferred** name. The preferred name will be displayed in search results and patient screens.

Addresses

A person may have zero or more contact addresses. You may configure the format of these addresses for your particular locale.

Person Attributes

To support your local needs, you can define additional pieces of information about the people in your system, on top of those that are natively supported by OpenMRS. You can define the datatype of a Person Attribute, as well as any constraints on the possible values, using metadata. This metadata is called a Person Attribute Type.

Person Attributes are suitable for storing other information. But historical values of person attributes are not retained. For example, you should use a person attribute to record a patient's contact telephone number. This information may change, but if it does so, the system need only store the most recent value, and need not retain previous values. It is not appropriate to use a person attribute to store something like the patient's height, which is recorded at a given point in time, but can be expected to change and should be tracked as it does so.

Patients

Anyone who receives care in OpenMRS must be a **Patient** (for example, anyone who has an Encounter or who is enrolled in a Program). Every Patient must have at least one Identifier, which is explained below.

A Patient is also a Person, meaning they must have at least one name and they may have addresses.

Patient Identifier

The Patient Identifier is a medical record number assigned by your facility, used to identify and re-identify the patient on subsequent visits.

A **Patient Identifier Type** defines the format of a particular kind of patient identifier. For example, you might define thatAmani ID is an identifier type that is required for every patient; the format is 2 letters followed by 6 digits and uses a particular check digit algorithm.

A **Check Digit** is an extra digit that is added to the end of an identifier, and depends on the rest of the identifier. It allows OpenMRS to determine whether an identifier has been mistyped. For example using a Luhn check digit, "1234-1" is valid, but "1234-5" is incorrect. It is a strongly recommended best practice to use check digits in all patient identifiers that you assign. For more information about check digits, see

A **Relationship** is a bidirectional link between two Persons in OpenMRS.

The metadata that describes a particular kind of relationship is a **Relationship Type**. It defines the names of each direction of the relationship. Typical Relationship Types are Parent/Child and Doctor/Patient.

At the Amani Clinic, it is necessary to use relationships to link a mother's patient record to the patient record of her children. One might also use relationships to record the link between a patient and their primary care provider.

Visits

AVisit in OpenMRS represents exactly what it sounds like: a time period when a patient is actively interacting with the healthcare system, typically at a location. The metadata differentiating different types of visits is a **Visit Type**. Visit Types are displayed in the user interface, and can be searched against.

A visit contains encounters, which store more granular data about treatments or services.

At the Amani Clinic, a patient might typically check-in at registration, be seen by a doctor, and receives medication dispensed in the pharmacy. This would be recorded as one visit of type of **Outpatient**, and contain three encounters (**Registration**, **Consultation**, and **Dispensing**).

Encounters

A moment in time where a patient is seen by providers at a location, and data are captured. Generally speaking, every time you enter a form in OpenMRS this creates an **Encounter**. Encounters typically belong to a visit, but they may also stand alone.

The metadata that describes a kind of encounter is an **Encounter Type**. These are displayed in the user interface, and you may also search against them.

During a typical Amani Clinic Outpatient Visit, a patient checks in at registration, is seen by a doctor, and receives meds dispensed in the pharmacy. This would be recorded as one visit containing three encounters, whose types are **Registration**, **Consultation**, and **Dispensing**.

Providers

A **Provider** is a person who provides care or services to patients. A provider may be a clinician like a doctor or nurse, a social worker, or a lab tech. Generally speaking, any healthcare worker that a patient can have an encounter with is a provider.

Providers may have full records in OpenMRS as persons, or they may just be a simple name and ID number.

Locations

A **Location** is a physical place where a patient may be seen.
Locations may have a hierarchy, for example **Children's Ward** might be a location within the location **Amani Clinic**.
You might also store physical areas (for example **Eastern Province**, or **California**) as Locations. You should not use locations to represent logical ideas like **All District Hospitals**.

Observations

An **Observation** is one single piece of information that is recorded about a person at a moment in time.
Every observation has a Concept as its question, and depending on the datatype of the concept, it has a value that is a number, date, text, Concept, etc.
Most of the information you store in OpenMRS is in the form of Observations, and most Observations happen in an Encounter. When you enter a form in OpenMRS, typically one Encounter is created with anywhere between tens or hundreds of Observations.
Note that an individual Observation is valid only at one moment in time, and it does not carry forward. You may query the system for the last observation for **pregnancy status** but this does not tell you whether or not the patient is pregnant at any point after the moment of that observation.
Examples of observations include **Serum Creatinine of 0.9mg/dL** or **Review of cardiopulmonary system is normal**.

Observation groups

Sometimes a single Observation is not sufficient to capture an entire piece of patient information, and you need to use multiple Observations that are grouped together.
For example, recording that a patient had a rash as an allergic reaction to penicillin would need to be stored as two observations plus a third one that groups the previous two together:
1 Concept = "Allergen", coded value = "Penicillin", group = (3)
2 Concept = "Reaction", coded value = "Rash", group = (3)
3 Concept = "Allergic Reaction Construct", group members = (1), (2)

Orders

An **Order** is an action that a provider requests be taken regarding a patient.
For example a provider could order a Complete Blood Count laboratory panel for a patient.
An Order only records an intention, not whether or not the action is carried out. The results of an Order are typically recorded later as Observations.
Prescribing a medication is a **Drug Order**. A drug order can be placed for a generic drug, represented by a Concept (for example, **500mg of Ciprofloxacin, twice a day**). If you are using OpenMRS to manage a formulary of specific medications (i.e., **Drugs** in OpenMRS), you may also record **Drug Orders** against those. For example, a drug order might be **One 500mg tablet of Ciprofloxacin, twice a day**.

Allergy lists

OpenMRS lets you manually maintain an **Allergy List** for a patient, including the allergen, reaction, severity, etc.

This list is managed separately from Observations: observing an allergic reaction to a drug does not automatically add an Allergy to the list.
Unlike an Observation (which happens at one moment in time), an Allergy is longitudinal data, with start and end dates.

Problem lists

OpenMRS lets you manually maintain a **Problem List** for a patient. This list is managed separately from Observations: observing that the patient has "Diagnosis Present = Diabetes" does not automatically add a problem to the list. Unlike an observation (which happens at one moment in time), a problem is longitudinal data, with start and end dates.

Program enrollments, workflows, and states

A **Program** represents an administrative program or study that a patient may be enrolled in (for example, **Child Nutrition Study** or **DOTS Tuberculosis Treatment Program**).
A **Program Enrollment** represents the fact that a patient is enrolled in one of these programs over a time period at a Location. This is longitudinal data with a start date and end date.
A Program can also define administrative **Workflows**, and possible **States** the patient may have within those workflows. An **Initial State** is one that a patient is allowed to start in when they are first enrolled in a program. A **Terminal State** is one that closes the program enrollment if the patient is placed in it.
For example a research study on infant nutrition might have a workflow called **Study Enrollment Status** with the states:

- Patient Identified (initial)
- Mother Consented to Study
- Study Complete (terminal)
- Lost to Followup (terminal)

These states are meant to represent administrative statuses, not clinical ones. For example putting a patient in a **Loss to Followup** state represents an official declaration and will not happen automatically even if no encounters are entered for the patient for several months.

Forms

A **Form** represents an electronic form that may be used for entering or viewing data. The basic OpenMRS system does not define a specific technology for entering forms. You will need to use one of the community-developed form entry modules. See the chapter "Data Entry" for more details.
The Form Entry (Infopath) and XForms modules rely on a **Form Schema**, where you define which Concepts are used on the Form. The HTML Form Entry module does not require you to manage the schema.

Users, roles, and privileges

A **User** in OpenMRS is an account that a person may use to log into the system.
The real-life person is represented by a Person record in OpenMRS, and a person may have more than one user account. If you want a patient to be able to view her own record in OpenMRS, then you need to create a user account and link it to the patient.

A **Role** represents a group of privileges in the system. Roles may inherit privileges from other roles, and users may have one or more roles.

A **Privilege** is an authorization to perform a particular action in the system. The list of

available privileges are defined by the core system and by add-on modules (for example, **Delete Patients** and **Manage Encounter Types**), but you need to configure which roles have which privileges while you are configuring your system.

The information model in use at Amani Clinic

A patient named Asaba arrives at Amani Clinic, where the registration clerk James creates her electronic record and stores her contact phone number as 312-555-7890. On paper the Nurse, Kissa, records Asaba's weight as 61.5kg and orders a pregnancy test. James enters these onto an electronic screen.

From the perspective of the OpenMRS model, we have the following metadata:

- The nurse, Kissa (a Provider)
- The registration clerk, James (a User)
- Contact Phone Number (a Person Attribute Type)
- Weight, in kilograms (a Concept, with class **Finding** and datatype **Numeric**)
- Urine Pregnancy Test (a Concept, with class **Test**)
- Amani Clinic (a Location)
- Outpatient Visit (an Encounter Type)
- Outpatient Triage Form (a Form)

When Asaba is first seen at the registration desk, James creates the following data:

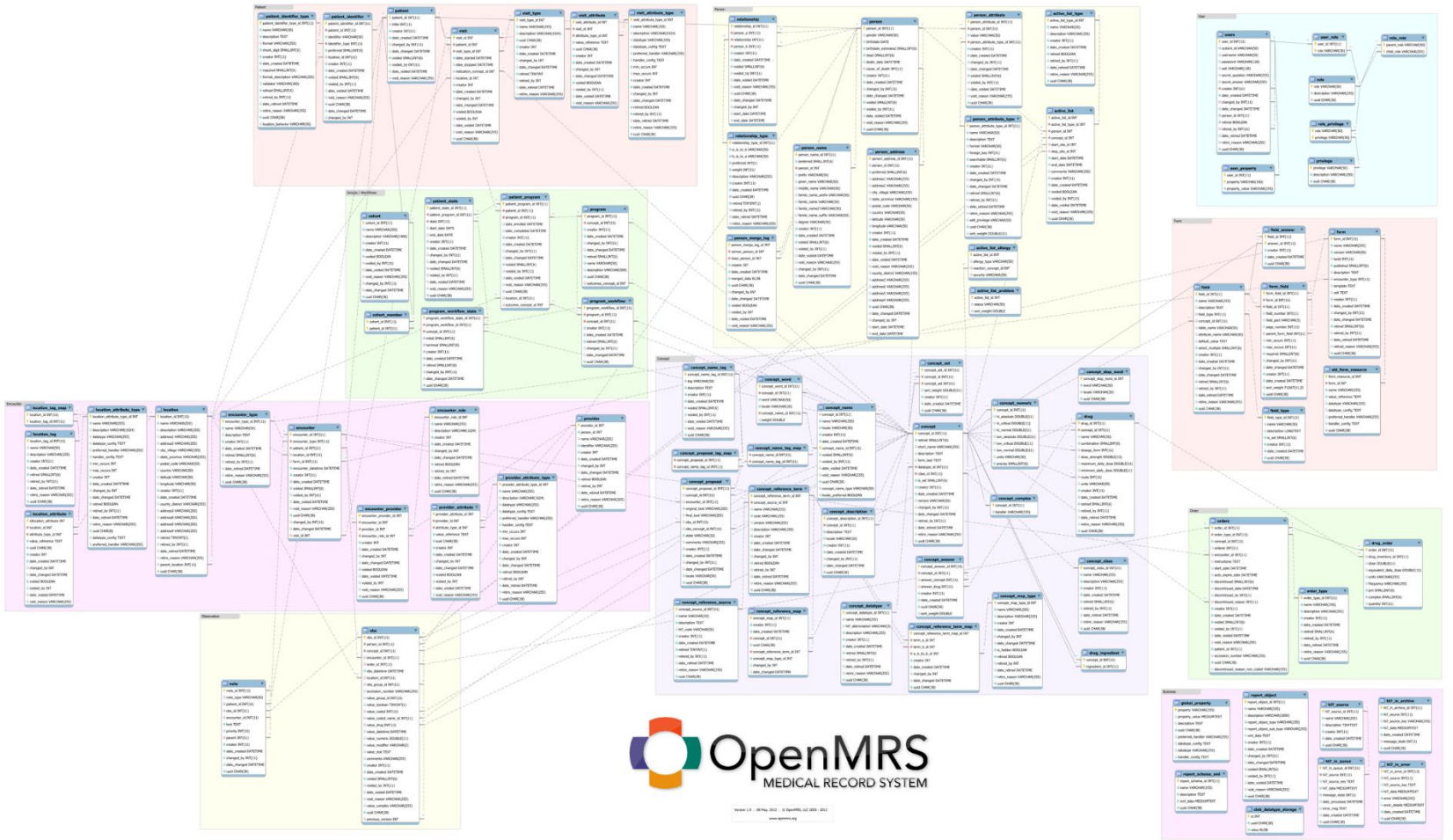
- A Patient (Asaba)
- A Person Attribute (type = Contact Phone Number, value = 312-456-7890).

After Asaba sees the nurse, who gives a paper form to James, he creates more data:

- An Encounter with:
  - patient = Asaba
  - type = Outpatient Visit
  - form = Outpatient Triage Form
  - location = Amani Clinic
  - provider = Nurse Kissa
  - creator = Registration Clerk James
- An Observation (in that encounter), of **Weight in kilograms** = 61.5.

An Order (in that encounter), for **Urine Pregnancy Test**

OpenMRS Database Design Schema:


OpenMRS
MEDICAL RECORD SYSTEM

Deliverable: SQL based Database for medical record system
Results: The milestone has been achieved.