# Project Medical Visualization : Carotid Visualization

Andrés Martínez Mora

9th January 2020

# Contents

**1 Motivation** **3**
1.1 Applications . . . 3
1.2 Dataset Features & Preprocessing . . . 4
1.3 Impact Analysis . . . 4

**2 Description** **6**
2.1 Data introduction into VTK . . . 6
2.2 Visualization modes . . . 6
2.3 User Interface . . . 8

# 1 Motivation

The study of the carotid arteries anatomy and their blood flow plays an important role in the diagnosis of carotid plaques. An early detection of carotid plaques can prevent the future appearance of cerebral stroke, what can be fatal.

This Project proposes a visualization tool to enhance the analysis of the anatomy and the blood flow in the carotids from 4D-flow and T1 MR images, with the final goal of simplifying and facilitating the diagnosis of carotid plaques.

## 1.1 Applications

The main applications of the proposed tool would be:

- For clinical diagnosis of carotid plaques or any other alteration of the carotid anatomy or flow. The patient would come to the clinics, take some images of his or her carotids with and these images would be stored in the PACS.

  In here, the end user would be the radiologist or vascular doctor in charge of stating if there is any problem with the acquired images. Ideally, the end user would select from the PACS what images he or she wants to extract relevant visualizations from. With the 3D and 4D visualizations, the diagnosis would be simpler and more effective than with state-of-the-art tools that mainly rely on reconstructions given in 2D slices.

  In order to help for diagnosis, the tool would provide extra information on 3D anatomy, 3D anatomy variation along time (4D) and flow for any point in space and time.

- For surgery planning in catheterization cases in the carotids. In this case, the patient that requires a catheterization in the carotids would be imaged, getting 3D and 4D images. As in the diagnosis case, these images would be stored in the PACS and when the end user wants to access them and extract 3D and 4D visualizations, the tool would enter the PACS to retrieve these images.

  The end user for this case would be a vascular surgeon or any other member of his or her team. The 3D anatomy given by the tool can help the team on how to plan to drive the catheter through the carotids, while the flow information can be useful to determine if the introduction of the catheter in the carotids could distort the patient's regular blood flow.

  This surgery planning application is highly related to the diagnosis application, as very usually some carotid plaque is first diagnosed and then the patient undergoes a carotid angioplasty surgery to place a stent in order to re-establish an adequate blood flow in the area and prevent future strokes. Consequently, the tool may not only be used for one application, but for several of them.

- For pedagogical purposes. It is becoming more frequent in medical schools the use of simulations to train medical students instead of laboratory work with corpses, what is very expensive and even controversial. From raw images coming from anonymized patients, the end users can obtain interesting information on the 3D anatomy and flow of the carotid trunk and on the 4D variation of this 3D anatomy along time. If the extracted views are quite informative, they could even be used as part of Anatomy or Physiology books, although it is true that the tool is quite specific, as it is only focused on the carotid trunk.

  In this case the end user would be a professor in a medical school (usually a Medical Doctor) and students.

## 1.2 Dataset Features & Preprocessing

The dataset for the tool consists on 4D flow and T1 MR images. The T1 images are static (3D, only one time frame), being acquired with and without contrast and in principle they do not require any preprocessing.

The 4D flow images have a more complex structure and thus require to be preprocessed. The raw files with the 4D information are three .mat files. One file contains the velocity information throughout space and time in the Antero-Posterior (AP) direction, other in the Right-Left (RL) direction and the last one in the Foot-Head (FH) direction. The information in these files could be understood as the x, y and z components of the velocity vectors for all the patient points and for the acquired frames in time. Two main images can be extracted from here:

- A 4D speed image whose voxel values throughout time code the velocity magnitude. This can be obtained as:

$$Speed = \sqrt{AP^2 + RL^2 + FH^2} \tag{1}$$

- A static (3D) Phase-Contrast Angiography (PC-MRA) image where each voxel value codes the flow direction in a given time frame. This can be obtained as the mean value of the coded velocities in the given directions.

While developing the tool, I observed that the contrast of the obtained speed and PC-MR images was a little bit poor, so optionally, an extra step for 4D flow images could be a contrast augmentation by means of some voxel transform as a logarithm or an exponential or a histogram equalization.

## 1.3 Impact Analysis

The Impact Analysis will be developed as a series of Pros & Cons with respect to the presented applications.

Among the pros or advantages of the incorporation of this tool into the medical workflow, we can find that:

- A more precise and easier diagnosis and detection of abnormalities in the carotid trunk. This would reduce the number of false negatives, this means, the number of patients that have some abnormality but due to low quality visualizations with state-of-the-art tools, they are not diagnosed with them. These patients have later on higher chances of suffering from side effects from irregular carotid blood flow, as can be strokes, for example.

  The cost, the time and the workload required by the Healthcare system to treat a stroke or any other ignored abnormality can be quite high and consequently it can be highly reduced with a better quality tool as the one presented here. This tool would aim at the early diagnosis, substituting the expensive late treatments by cheaper early treatments as could be an angioplasty, with a better quality of life for the patient.

- A higher and more realistic amount of information in surgery planning, as would be a carotid catheterization, increases the chances for a successful surgery and reduces the probabilities for the patient to suffer from side effects as strokes. This would provide a better quality of life for the patient and would reduce the number of patients that need secondary treatments due to failed surgeries, what can be seen as a waste of money, time and workload for the Healthcare system.

- Highly realistic simulations of the anatomy and the flow in the carotids can help medical students to get a better and faster knowledge without having to go to the lab to work with corpses. Working with corpses involves many legal issues with much bureaucracy and is highly expensive, so a tool like the one presented here could help to cut down on these expenses.

The disadvantages or cons have often to do with extra money, time and workload required by the System to allocate a tool like this. In more detail, some potential drawbacks could be:

- The initial money to be invested in the tool, especially as the tool is more complex, with more functionalities and more realistic. Some money has to be placed on the hardware workstation (special visualization screens, GPUs for fast computing, large memory amount...). Nowadays, a usual visualization workstation can have a price of dozens of thousands of dollars, just for the hardware part.

  Other part of the money would be dedicated to pay special software needed to develop the application and to pay the developers behind the software of the tool itself (back-end and front-end) and behind the software that would connect the tool to systems as the PACS.

  Money has also to be placed into passing different safety and quality tests, so that the tool is certified to be used for medical care. At last, more money has to be placed in eventual checks and repairs of the tool after it has been installed, so it must be ensured that these events will be as less probable as possible.

- Other important resource is time. Every time that some new tool is introduced into the medical workflow, the end user (mostly doctors and nurses here) have to place time to learn how to use it. This time is shorter as the tool is more usable, so apart from having a safe and effective tool, it is important that the tool is usable, too, as it will reduce the learning time for the end user and will avoid complications and misunderstandings.
- Additionally, the time spent by the medical staff understanding the tool is time that is not dedicated to the patients, so the care is worsened. This also becomes an extra task for the staff, who are usually overloaded. Sometimes, even developers from the company have to come to the clinics to explain how the tool works, so this is also a loss of money and time for them.
- The time during which the tool is being checked or repaired is lost time and money, so the tool must be as reliable as possible so that this time is as short as possible.

In the end, for the tool to be effective in the Healthcare system the time spent in learning and in checks and repairs has to be minimized with a usable design, while the potential for the tool to cut down on expenses and improve the patient's quality of life has to be maximized.

# 2 Description

## 2.1 Data introduction into VTK

As explained in the Section 1.2, the original dataset consisted of two T1 images with and without contrast and of three images for 4D flow with velocity information in AP, RL and FH directions, respectively. From the 4D flow images, a dynamic speed image and a static PC-MRA image were extracted in the preprocessing step. The images were required to be inputted into the main Python library responsible for the visualization pipeline: the Visualization ToolKit (VTK). The T1 images were .mhd files, so a .mhd reader for VTK was used as Source. In the meanwhile, the AP, FH and RL images were .mat files, so they were translated into arrays with Scipy and then used to obtain the speed and PC-MRA images as arrays (see Section 1.2). At last, the speed and PC-MRA arrays were saved as .nii.gz files with NiBabel and introduced in VTK with a NIFTI file reader.

## 2.2 Visualization modes

The first visualization mode to be extracted was Multi-Planar Reconstruction (MPR). This consists on a classical slice extraction into axial, coronal and sagittal orientations. The dimensions of the images in the real world were extracted and used to build a plane for reslicing with vtkImageReslice. The orientation of this plane varied for axial, coronal and sagittal cases. The resulting slices were left in grayscale, as they were just 2D visualizations. This mode does not really add a lot of value to the tool, but I thought that it could be interesting to add it, especially for later glyph visualization. In the final interaction panel (designed with PyQt5), the user can input the orientation (axial, sagittal, coronal), the slice number to be plotted with a slider and for 4D datasets he or she can also introduce the time frame to visualize. The slice number has to be converted into world coordinates for VTK, taking into account the spacing, the origin and the voxel size.

The second visualization mode was Maximum Intensity Projection (MIP). It consists in the projection of data from some inputted orientation (axial, coronal or sagittal), taking the highest values found in the third dimension. This mode also computes the image dimensions and obtains a 2D projection with the vtkImageReslice class. However, the maximum value of the projected voxels is taken with SetSlabTypeToMax.

The third visualization mode was Surface Rendering. This mode allows to obtain the surface of a segmented organ, using some method to compute polygons to make up the organs in the zone where the limits of the organ lie. In this case, the typical Marching Cubes algorithm was used with the class vtkMarchingCubes. A filter for image casting and a polydata mapper were used, too. As there were only segmentations available for the carotids in 4D flow, only these were surface-rendered for the speed and PC-MRA images. The user can additionally select a Look-Up Table (LUT) for the organ, being able to choose among grayscale, rainbow, red-blue or blue-red.

The next visualization mode implemented was Volume Rendering. This mode allows to obtain a 3D view of the whole dataset. A color transfer function is required for this, assigning to each gray value in the dataset histogram an intensity value from some LUT. The user can choose among grayscale, rainbow, red-blue or blue-red LUTs. An opacity transfer function is also required in order to regulate how transparent or opaque every voxel is, so that further voxels can be observed as well. The opacity transfer function follows a window technique: only those voxel values in interesting ranges are selected, between a minimum and a maximum. In here, the end user has several interaction options: regulating the minima and maxima of the windows with sliders, regulating the maximum opacity value (from 0 to 1) with another slider and choosing between a linear opacity function, an exponential opacity function or a logarithmic opacity function, so to achieve volumes with different contrasts. Both color and opacity transfer functions are assigned as volume properties that are later on linked to the rendering actor.

The last two techniques can be combined: the carotids can be surface-rendered while the rest of the dataset can be volume-rendered with the same opacity and color transfer functions as explained in the paragraph above. Each rendering method would have an actor. Both actors can be combined into the same renderer and thus provide a combined window. The same interaction options as in the separated modes of surface and volume rendering are also present in this combined visualization. However, as only the 4D flow has segmentations for the carotids, only the speed and PC-MRA images can display this multiple visualization while the T1 images cannot display it.

The next visualization mode has to do more with flow visualization than with anatomy visualization. It consists on glyph visualization. As only the speed image is 4D, glyphs can be only visualized with this image. Glyphs consist on graphical representations of a vectorial field (in this case velocity) in a given space (the patient anatomy). For every voxel in the speed image, there is an AP, a RL and a FH component of the velocity vector (like vx, vy and vz). Consequently, one can create a grid of coordinate points in VTK along a given orientation (preferably the axial one, as it is the one where information can be more easily observed).

In this grid, each coordinate point is assigned the velocity direction coming from the AP, RL and FH matrices and also a color. The color is assigned in an RGB system, being each color component one of the velocity components. All these features are saved as a polydata. The final glyphs are produced with the VTK class vtkGlyph3D, receiving the grid points with the colors and orientations and a geometry that codes the grid information. The end user can choose among arrows, spheres, cubes or cones. Together with the glyphs, the axial slice corresponding to the space codifying the glyphs is displayed as well, with a code similar to the MPR mode.

At last, a video on 4D volume rendering for the speed data was extracted by looping through all speed frames and saving them in a video file. This was not included in the final User Interface as I did not find a way to place a video in a PyQt5 interface.

## 2.3 User Interface

Apart from coding the different visualization modes, the Project also included the design of a User Interface (UI) with PyQt5, with the VTK widget available in that library. This interface is organized in a grid that contains a frame in the upper left part where the visualization is displayed and a series of buttons and sliders to regulate different functions of the visualization in the rest of the menu. The buttons are called "QRadioButtons", they are organized in exclusive multiple choice groups and have a title. The sliders are called "QSliders", allowing the user to move them between a minimum and a maximum value, having also a title. The different functions that can be controlled with these buttons and sliders are:

- Selection of the data to display (speed, PC-MRA, T1 without contrast, T1 with contrast)
- Selection of the visualization mode among the ones presented above
- Selection of orientation for MIP and MPR (axial, coronal, sagittal)
- Selection of slice for the chosen orientation in MPR visualization
- Selection of time frame to display for speed in MPR and glyphs

- Selection of LUT for surface rendering (grayscale, rainbow, blue-red, red-blue)
- Selection of LUT for volume rendering (grayscale, rainbow, blue-red, red-blue)
- Selection of opacity function for volume rendering (linear, exponential, logarithmic)
- Selection of maximum opacity value for volume rendering with a slider
- Selection of minimum and maximum values for the window display in volume rendering, with a slider
- Selection of slice for the chosen orientation in glyph visualization
- Selection of animation. The visualization modes that display 3D information (surface rendering, volume rendering and both at the same time) have the option to be shown rotating in parallax, to add an extra depth cue for the end user. Nevertheless, the display of animations can make the tool much slower, as it requires a high computational cost.