Documentation and Instruction Manual

Before using this tool: Please make sure you have the 64-bit version of VLC player installed on your machine, as the program is dependent on libraries within the 64-bit version. Additionally, this tool will only run on Windows and Linux operating systems; it is not compatible with Apple OS X devices.

How the program runs:

1. The program requires the inclusion of the VLCJ ".jar" files during compilation and run.
    a. Provided bash file does this for linux systems and provided bat file does this for windows systems
2. Main class of the project calls VFI_Map initialization method to create a mapping from a preloaded color key to velocity values.
3. Controller's start view method is called
    a. Instantiates the controller
    b. Call constructor of the view
        i. View instantiates 3 panels
            1. 1 panel for the buttons that control a video
            2. 1 panel for the buttons that control other features of the GUI
            3. 1 panel to host the still images or video
        ii. View's action listener is added to each button to allow the view's action performed method to make the appropriate calls when a button is selected
    c. Call constructor of model
        i. Instantiates an arraylist to store drawn vectors when the user clicks within a still image
4. GUI is now setup, action listener in view class registers button selection, action listener in controller class registers clicks on still images (from VFI video), and "repaint" method call in view's action performed method will call MyPanel's paintcomponent method to redraw image and vectors (if user made any clicks).

Buttons:

Upload File: Selecting this button allows users to upload an .avi or .png file for analysis

Set Velocity: Selecting this button will allow you set a new maximum velocity based on the value provided by the uploaded file's color map. The default velocity is set to be 236.5 cm/s. Failure to update velocity will affect measurements of vector components as well as magnitude. Only numerical values should be entered (the program will notify you if you entered anything incorrectly).

Get Average Speed in Area: Selecting this button will cause a smaller window to pop onto the screen, containing a smaller version of the uploaded image. A user can select a portion of the image to get the average speed in the highlighted area. A pop up window will show when the average is found.

Clear Vectors: Selecting this button will clear any created vectors from the screen.

Create Vector File (Drawn Vectors): Selecting this button will create a spreadsheet containing the X-Y coordinates, vertical and horizontal vector velocity components, and magnitude of all created vectors on an image. A pop up window will show when the file is done being created.

Create Vector File (All Color Pixels): Selecting this button will create a spreadsheet containing the X-Y coordinates, vertical and horizontal vector velocity components, and magnitude of all colored pixels in an image. A pop up window will show when the file is done being created.

Play: Selecting this button will play an uploaded .avi file within the frame. If a video is paused, click play to resume playing the video.

Pause: Selecting this button will pause the play of a file. To resume play, click the play button.

Next: Selecting this button will advance the video one frame.

Grab Frames: Selecting this button will split an uploaded .avi into individual frames (the user will be able to select which directory on the computer to save the frames), which users can then upload for vector analysis.

Future Work Recommendations:

The largest recommendation for future improvement would be to verify the accuracy of the readings achieved from the program in its current state. As it stands, our program produces results accurate relative to the color map included in the Doppler ultrasound scans, but these data sets need verification by other tools in order to have any real-world usability. Additionally, one could improve and expand upon the current controls of the program, most specifically with the addition of functionality to step backward or rewind when playing the .avi file. Implementing this would likely look like a function that set the time of the video to a set time before the then-current frame, likely one second or half a second (useful methods for jumping forward a frame is found in the implementation of the “grab frames” button found in the MyPanel.java file -- just subtract from current time, instead of adding).

Given enough time, our program could be improved by finding a way to incorporate a different media player into the program. While VLCJ is sufficient at this point, a different media player would potentially allow for creating and analyzing vectors on full .avi files as opposed to still images. This could also assist in packaging the program for access in MATLAB.