Muscle Movement Quantification

André Saraiva (60623) Pedro Conde (64346)

***Resumo* – Neste trabalho desenvolveu-se um programa de processamento de imagem ecográfica, utilizando C++ em conjunto com as librarias de funções do OpenCV, com o intuito de quantificar o movimento de músculos em ecografias.**

***Abstract* – This project consists of a muscle movement quantification program developed using C++ and the functions libraries provided by OpenCV.**

I. INTRODUCTION

With the increased use of medical images and videos usage, it's becoming essential to find tools to aid doctors and specialists on their job.

The aim of this project is to create three different approaches of analysis of echographies and to track and quantify muscle movement.

This report presents three different analysis methods for muscle movement quantization. A description of each analysis method will be followed by a comparison between the three.

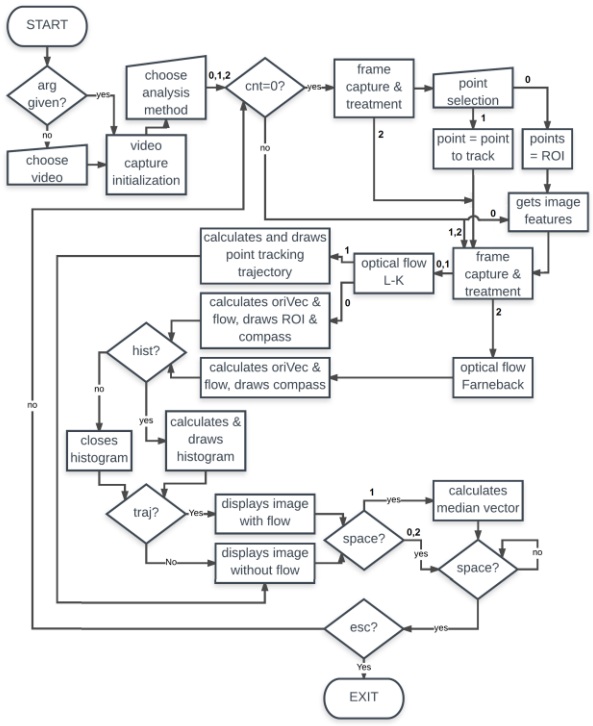

Fig. 1 - Simplified code flow chart. Lozenges – conditions; Squares – executions; Crooked squares – user options.

II. IMPLEMENTATION

The code flow works based on user input. As we can see from *Fig.1*, the program starts by asking the video and the method of analysis.

A. If the first method, Region of Interest (Lukas-Kanade) is selected, then the following will happen:

1. The first frame is captured and treated (resized, for display purposes, and converted to grayscale);
2. The user selects the points of the region of interest;
3. The good image features are obtained (determines strong corners on an image);
4. The optical flow is calculated using the Lukas-Kanade method;
5. The vectors' orientation and flow is calculated and the ROI and compass are drawn.
6. The user gets to choose either if he wants the histogram and the trajectory of the vectors displayed or not, pressing *H* and *S*, respectively;
7. The user can also press *SPACE* at any time to pause the video and press *ESC* to exit the program;
8. Repeat from the third step until there's no more remaining frames;
9. Program exits.

B. If the second method, Point Tracking (Lukas-Kanade), is selected, then the following will happen:

1. The first frame is captured and treated (resized and converted to grayscale);
2. The user selects the point to track;
3. The selected point is treated as a feature for calculating the optical flow of it;
4. The optical flow is calculated using the Lukas-Kanade method;
5. The tracked point trajectory is calculated and drawn;

6. The user can also press *SPACE* at any time to pause the video and press *ESC* to exit the program;
7. The next frame is captured and treated (resized and converted to grayscale). Repeat from forth step until there's no more remaining frames;
8. Program pauses for viewing purposes and exits by pressing *ESC*.

C. If the third method, Optical Flow (using Farneback), is selected, then the following will happen:

1. The user will be prompted with the choice of a dense or sparse output flow;
2. The first frame is captured and treated (resized and converted to grayscale);
3. The optical flow is calculated using the Farneback method, thus obtaining a dense optical flow;
4. The orientation vector and flow of interest are calculated and the compass drawn;
5. The user gets to choose either if he wants the histogram and the trajectory of the vectors displayed or not, pressing *H* and *S*, respectively;
6. The user can also press *SPACE* at any time to pause the video and press *ESC* to exit the program;
7. The next frame is captured and treated (resized and converted to grayscale). Repeat from the third step until there's no more remaining frames;
8. Program exits.

III. REGION OF INTEREST (LUKAS-KANADE)

The first analysis consists on the application of the Lukas-Kanade optical flow algorithm in a region of interest. In the Fig.2 we can see the capture of the first frame and the selection of the region of interest.

The results are displayed on the Fig.3 and Fig.4. On the corner of Fig.3 we have a direction compass to represent each directions' colour. In the Fig.3 there's the calculated optical flow inside the region of interest previously selected and the corresponding orientation histogram on the right. Each colour corresponds to a certain direction. On Fig.3 we have the average direction vector of all the direction vectors inside the ROI. This vector is multiplied by a factor scale just so it can be better seen, with its true size value displayed in the top-left corner of the image.

This implementation is faster compared to the Fanerback method, because it's calculating good features to track followed by the optical flow, compared with calculating the optical flow for every pixel in the image and then selecting the pixels of interest to analyse (arranged in a grid). The good features to track are also being calculated every frame to guarantee that the points of interest don't get lost, even if momentarily (for example, for a single frame).

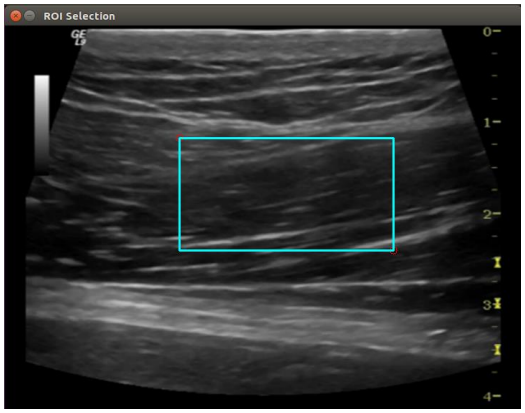

Fig. 2 - First frame captured. Region of Interest selection.

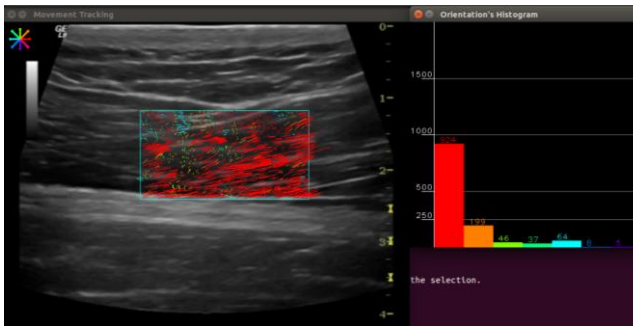

Fig. 3 - Movement Tracking. Optical flow on the left and histogram with movement quantization on the right.

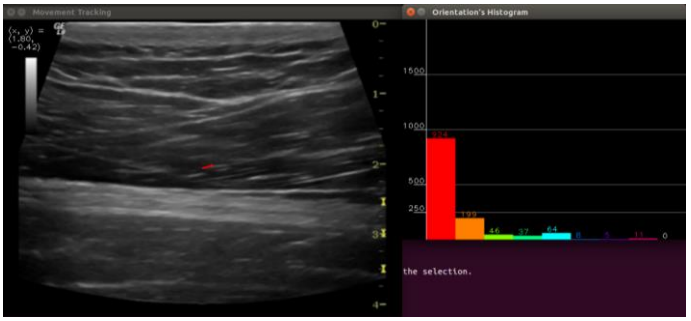

Fig. 4 - Movement Tracking. Average of all direction vectors displayed in red on the left. Histogram with movement quantization on the right.

IV. POINT TRACKING (LUKAS-KANADE)

The second analysis consists of a point tracking system also based on the Lukas-Kanade method.

The results are shown in Fig.5 where the user previously selected a point to track and by the end of the program it's possible to see the resulting trajectory of the same point during the video's timeframe.

This implementation is faster compared to the other two and works very well in real-time analysis, because it's calculating optical flow using Lukas-Kanade method without the need of the good features extraction and instead focusing on one pixel only.

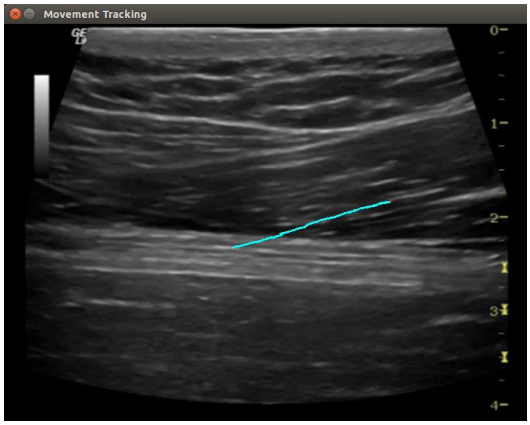

Fig. 5 - Movement Tracking. Resulted line drawn of point tracking.

V. OPTICAL FLOW (FARNEBACK)

The third analysis consists of an optical flow motion display based on the Farneback algorithm.

The user has the option to either select the sparse grid (Fig.6) or the dense grid (Fig.7) of the optical flow, which translates into less or more optical flow vectors displayed, respectively. The sparse uses 15 pixels of distance between each vector while the dense uses a denser grid to analyse (every other pixel is analysed).

Also, since this method creates a grid over the whole video, it was needed to remove points that were moving very little or not moving at all, such as points beyond the echography scan or along the echography border, resulting in no white points over the grid (points that didn't move). Thus, the zero elements in the last white histogram bar.

VI. VECTOR'S ORIENTATIONS AND COLOUR CODING

An equation (Eq.1) was used to calculate the vectors' angle, which in turn made possible a colour based orientation categorization to be made. One point corresponding to the point of arrival of the vector and the other to a point over a given horizontal line different from the starting point of the vector. In this calculation, the *acos* isn't able to find values in the third and fourth quadrants, which meant that it was needed to calculate in advance the vectors' general direction (positive, corresponding to the first and second quadrants, or negative, corresponding to the third and fourth quadrants).

$$angle = sign(\overrightarrow{pt1} \times \overrightarrow{pt2}) acos \frac{\overrightarrow{pt1} \cdot \overrightarrow{pt2}}{|\overrightarrow{pt1}| . |\overrightarrow{pt2}|}$$

Eq. 1 – Equation for vector's angle.

The vectors were grouped in nine groups, considering their angle. Eight for the cardinal directions and ordinal directions (north, north-east, east, south-east, south, south-west, west, north, west) and a ninth group for other possible directions (i.e. none, if the vector had a null size). Each of these orientations were given a colour as to distinguish between them, with the colour choice being based on the "value" dimension of the HSV colorspace (for example, a vector with an orientation of "east" is given the red colour, as it's angle is close to zero degrees), and the lack of direction as white.

VII. HISTOGRAM

The histogram groups the vectors into frequency of orientations. This gives it a column per orientation subdivision. The histogram was not based on the existing OpenCV histogram functions, but entirely purpose built. It can be seen in Fig. 6, for example.

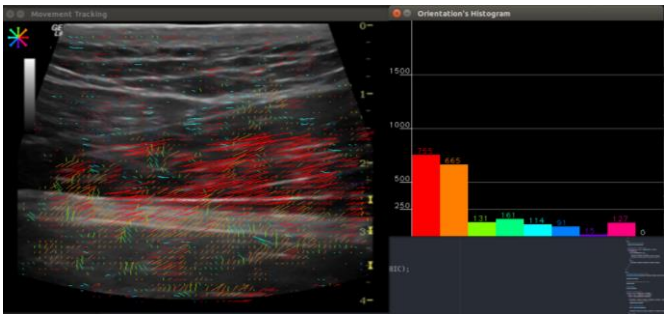

Fig. 6 - Movement Tracking. Sparse optical flow grid (Farneback) on the left. Movement quantization histogram on the right.

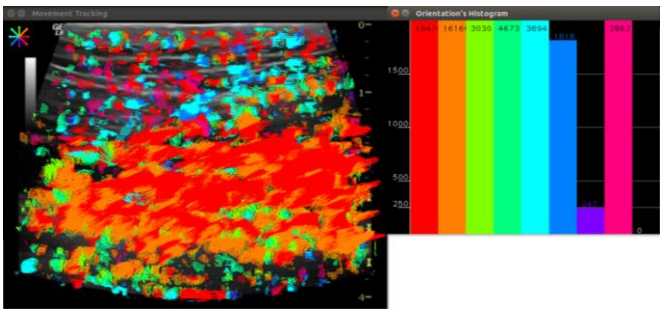

Fig. 7 - Movement Tracking. Dense optical flow grid (Farneback) on the left. Movement quantization histogram on the right.