Gait Energy Image Extraction from silhouettes using Python

Mohamed GAID
https://github.com/Mohamed-Gaid

Introduction

Created by J. Han and B. Bhanu and published in 2006, Gait Energy Image uses a "spatio-temporal gait representation, to characterize human walking properties for individual recognition by gait." GEI is defined as follows:

$$G_{x,y} = \frac{1}{N}\sum_{t=1}^{N} B_t(x, y) \tag{1}$$

Fig. 1 An example of a gait energy image

This guide will explain my code for extracting a modified version of GEI from silhouettes. Mainly, the ones from CASIA-B dataset. The only modification is in the alignment of the silhouettes, and is explained in section 3.

1 Border

The border of the smallest rectangle that contains a silhouette can be defined using the coordinates of its four corners. These coordinates can be extracted using the minimum of the minimum values of each row and column, and the maximum of the maximum values of each row and column.

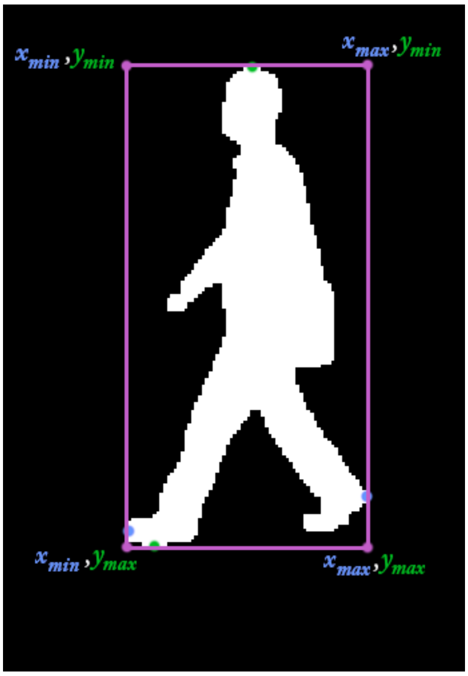

Fig. 2 The smallest rectangle that contains a silhouette

$$x_{min} = min\{min_k\{x_i\}_{i=1}^{l_n}\}_{k=1}^{N_r} \tag{2}$$

Where x_i is the x coordinate of each pixel, l_n is the length of the n-th row, k is the number of the row, and N_r is the total number of rows.

In the same way:

$$x_{max} = max\{max_k\{x_i\}_{i=1}^{l_n}\}_{k=1}^{N_r} \tag{3}$$

$$y_{min} = min\{min_k\{y_i\}_{i=1}^{l_n}\}_{k=1}^{N_c} \tag{4}$$

$$y_{max} = max\{max_k\{y_i\}_{i=1}^{l_n}\}_{k=1}^{N_c} \tag{5}$$

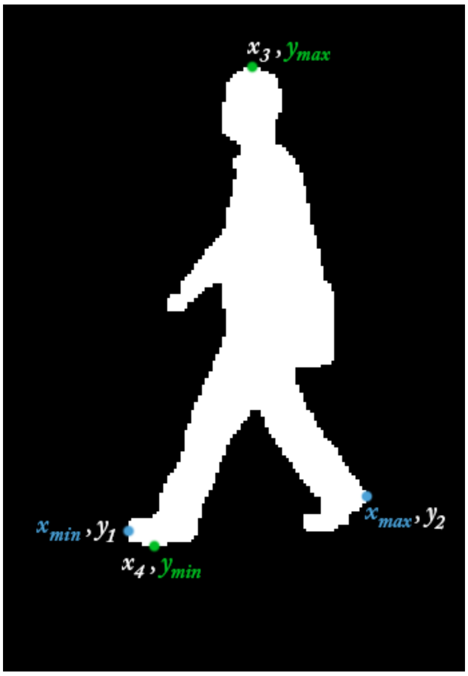

Fig. 3 The minimum and maximum x,y coordinates of a silhouette in blue and green respectively

2 Center of Mass

In a binary image, the center of mass is the point where the sum of relative coordinates of all white pixels is 0.

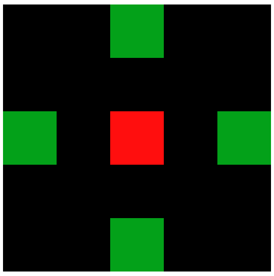

Fig. 4 The center of mass of four green pixels shown in red

To get the coordinates of the center of mass of a silhouette, we calculate the weighted mean of the center of each row to get the x coordinate, and the weighted mean of the center of each column to get the y coordinate.

$$C_x = \frac{1}{S_r} \sum_{n=1}^{H} l_n X_n \tag{6}$$

Where C_x is the x coordinate of the center of mass of the silhouette, l_n is the weight of the n-row which is its length, X_n is the x coordinate of the center of the n-th row, and S_r is the sum of weights of all rows. Since the weights of rows are their lengths, S_r can be expressed as the sum of lengths of all rows.

In **Fig. 5**, we have an example of a binary image, with the center of each row in blue. Using equation (6), the x coordinate of the center of mass is:

$$C_x = \frac{1}{3+5+7+5+3}(3 \cdot 5 + 5 \cdot 7 + 7 \cdot 5 + 5 \cdot 3 + 3 \cdot 2) \tag{7}$$

which results in $C_x = 4.6$, and by rounding it, we get $C_x = 5$.

Fig. 5 The centers of each row of a random picture, in blue

X_n can be calculated by dividing the sum of the x coordinates of the white pixels of the n-th row on the length of the same row.

$$X_n = \frac{\sum_{i=1}^{l_n} x_i}{l_n} \tag{8}$$

By substituing (8) in (6), we get:

$$C_x = \frac{1}{S_r} \sum_{n=1}^{H} \sum_{i=1}^{l_n} x_i \tag{9}$$

Following the same steps, the y coordinate of the center of mass is:

$$C_y = \frac{1}{S_c} \sum_{n=1}^{W} \sum_{i=1}^{l_n} y_i \tag{10}$$

Using equation (9) on the same example of **Fig. 5**, we get:

$$C_x = \frac{1}{3+5+7+5+3}(4+5+6+5+6+7+8+9+2+3+4+5+6+7+8+1+2+3+4+5+1+2+3) \tag{11}$$

We get the same result as in equation (7), $C_x = 4.6$, and by rounding it, we get $C_x = 5$.

Fig. 6 The center of mass of a random silhouette

3 Alignment

In the original work by J. Han and B. Bhanu, the silhouettes were vertically resized to fit the height of one of them, and were aligned horizontally by centering the upper half silhouette part with respect to its horizontal centroid.

In my work however, I chose to keep the original size of each silhouette in order to avoid any loss or distortion of information from resizing.

I also chose to align the silhouettes both vertically and horizontally using their center of mass and not the upper parts centroids. The reason for this is that the upper part centroid is highly susceptible to clothing changes (coat, hat, bag...)

The code will align the center of mass of each silhouette to the center of a black image that has the same dimensions as the video frame.

We can combine images in OpenCV by providing the coordinates of the top left corner of the smaller image (the silhouette) in the bigger image's frame (the black image), and the height and width of the smaller image.

$$X = \frac{W}{2} - C_x \tag{12}$$

$$Y = \frac{H}{2} - C_y \tag{13}$$

Where X,Y are the x,y coordinates of the the top left corner of the silhouette, W,H are the width and height of the black image, and C_x,C_y are the coordinates of the center of mass, in equations (9) and (10).

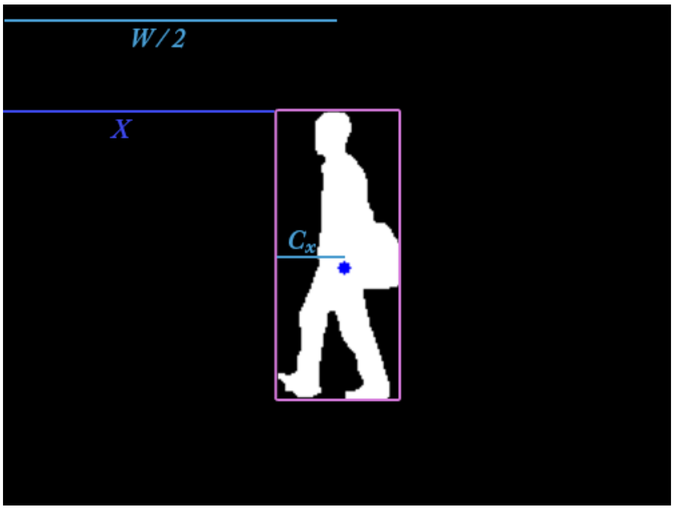

Fig. 7 Top left corner x coordinate of the silhouette image

4 Combination

As shown in equation (1), GEI calculates the mean value at each pixel of all the silhouettes binary images. The result is a gradient image where it is brighter in areas that are part of the silhouettes more often, and darker in areas that are part of the silhouettes less often.

We can calculate this mean, by adding each silhouette consecutively and calculating the current GEI. We repeat the following equation for each silhouette t:

$$G_{x,y} = \alpha B_t(x,y) + \beta B_{t-1}(x,y) \tag{14}$$

Where B_t is the intensity of a pixel in the current silhouette, B_{t-1} is the intensity of a pixel in the GEI up to the previous silhouette, and:

$$\alpha = \frac{1}{n+1} \tag{15}$$

Where n is the number of the silhouette. It starts from 1 if the first image is the first silhouette, and from 0 if the first image is the black image.

and:

$$\beta = 1 - \alpha \quad (16)$$

Fig. 8 Combining a silhouette with the GEI of the previous frames using equation (14)