

# Hand recognition on a video with active contour

Paul Schreiner, Antun Aleksa

28/11/2016

## 1. Two ideas for color based hand detection

### Backprojection detection

If the hand is in the center of the image, we can select a small square at the center of the image which will be on this hand. Its histogram gives a description of the pixel colors of the hand. We then use the backprojection algorithm to detect the whole hand : we use the histogram computed with the small rectangle to record how well the pixels of the whole image fit the distribution in this histogram using this algorithm. This method is quite effective even in poor lighting conditions, but the hand on the first video image needs to be centered.

### YUV color space

The skin can be easily detected using the YUV color space : Y represents intensity and U,V (Cb,Cr) represent chromatic distances to pure blue or pure red. Intensity is not very significant to detect skin, but we can find narrow intervals for U and V which represent skin color. However, this method might not work in poor lighting conditions.

## 2. Active Contour detection

Active contour model is used to detect an outline of a certain object in an image. This model is appropriate for our problem because it is used when the approximate shape of the boundary is known from before (in our case the shape of a hand) and it can be used to track dynamic objects.

The snake curves to fit the object by minimizing it's predefined energy. The energy of the snake is defined as a sum of internal and external forces which affect it's shape. The main parameters that influence the snake's shaping behaviour are  $\alpha$ ,  $\beta$ ,  $\gamma$ . Parameter  $\alpha$  defines the contour length,  $\beta$  is the shape smoothness parameter and  $\gamma$  parameter defines the contour's distance from the image edges.

## 3. Algorithm

### 1. Hand detection

Backprojection algorithm or YUV space color detection.

### 2. Noise filtering

We must find a method to remove the unwanted detected elements

### 3. Initialization of the snake

The snake must be initialized close to the hand.

### 4. Active contour detection

The Active contour algorithm's parameters must be optimized.

## 4. Tools

We chose python because we are familiar with the language and we have a lot of libraries at our disposal. We decided to use OpenCV to detect colors and run the backprojection or YUV color detection on the image, and Scikit-Image for its Active Contour function.

**Language :** Python

**Libraries :** OpenCV (Color detection), Scikit-Image (Active Contour)

## 5. First Results



Initial Image

### Backprojection detection

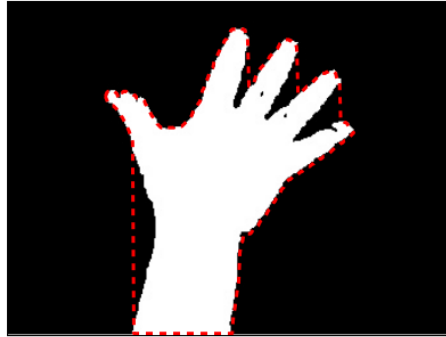
We run the backprojection algorithm returnig a color mask several times until hand is well recognized.



Color mask

### Snake Initialization

**Algorithm :** We start top left of the image. We go through the column until we find a white pixel, and mark him as the next point for the initial snake. We then go through the next column and repeat this until all columns have been visited. The same thing is done starting from bottom right and moving left.



Initial Snake

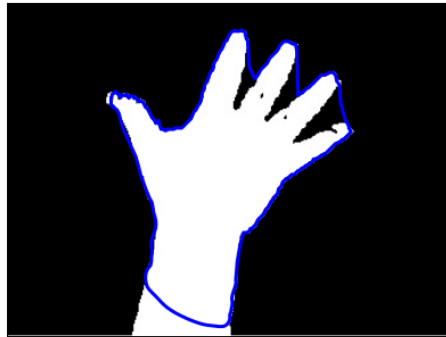
### Active Contour

**Parameters :**

$\alpha = 0.1$  Small value of elastic energy in order to keep the snake long enough to adapt to the hand.

$\beta = 10$  High value of stiffness energy to favour the curving of the snake to adapt to the hand.

$\gamma = 0.01$



Final Snake



## 6. Problems

**Snake size :** The snake can only shrink. If the initial snake isn't long enough, it cannot adapt to the hand properly, especially between fingers.

**Noise :** The algorithm is efficient if no noise is detected in the image.