Pattern Recognition Project

Florian Cousin
Computer Science
Bishop's University
Sherbrooke, Québec
fcousin18@ubishops.ca

November 28, 2018

I Introduction

In this project, we want to highlight skin in images with hands in it using Bayesian theory. All the code is done in Matlab.

II Problem Description

We are given a database of 2,350 hand images. The hands have actually been generated by a computer so some joints may be weird. I would have preferred real hand but I did not find any satisfying database on the internet. However, the colour of the hands respects the reality and these generated images does not affect the method nor the way to think so it does not really matter.

The images are like these:

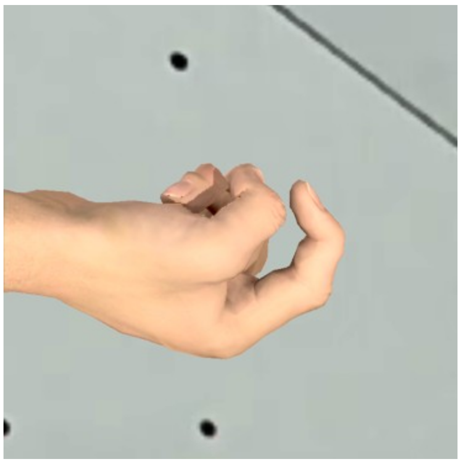

Figure 1: Image Example

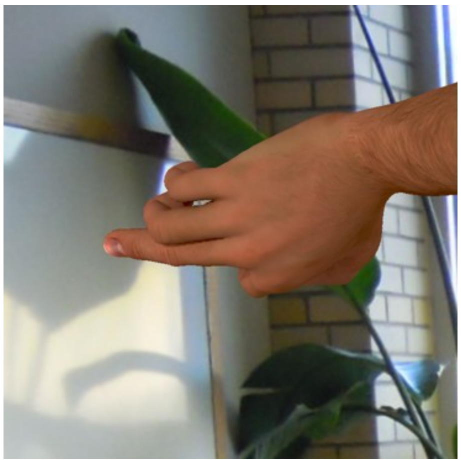

Figure 2: Image Example

The whole dataset is plit into the folders training and test in the dataset folder.

The goal is to split the hand and the background of each image using Bayesian theory and gaussian estimation in a supervised way.

III Solution Method

A Gathering Data

We want to split the image in a supervised way so the first thing we have to do is gathering pixels belonging to the hand skin and some others belonging to the background. The pixels belonging to the skin are stored in the skinPixels.mat file and the ones to the background are stored in backPixels.mat. Both files are in the training folder. They have been filled and can be still updated by the functions updateSkinData and updateBackgroundData. Here is the one for the skin.

```
function updateSkinData(images)

% Updates the collected data of the
    pixels of the hands skin.
%
% Input
% images: string array of the images to
     consider in the update. The images
```

```matlab
%   with no data yet are always considered first. The '.jpg' is not
%   included in the names.

if nargin < 1
    files = dir('dataset/training/*.jpg');
    images = string(zeros(1, numel(files)));

    for i = 1:numel(files)
        images(i) = convertCharsToStrings(regexprep(files(i).name, '.jpg', ''));
    end
end

if isfile('dataset/training/skinPixels.mat')
    % skinPixels is a struct. The fields of skinPixels are the names of the
    % treated images without '.jpg' and with a f at the beginning
    data = load('dataset/training/skinPixels.mat', 'skinPixels');
    skinPixels = data.skinPixels;
else
    skinPixels = struct();
end

alreadySeen = string(zeros(1, length(images)));
alreadySeenInd = 1;

% We first compute alle the images that has no data yet
for i = 1:length(images)
    fileName = char(images(i));

    if isfield(skinPixels, strcat('f', fileName))
        alreadySeen(alreadySeenInd) = convertCharsToStrings(fileName);
        alreadySeenInd = alreadySeenInd + 1;
        continue
    end

    skinPixels.(strcat('f', fileName)) = cropAndLin(strcat('dataset/training/', fileName, '.jpg'));

    save('dataset/training/skinPixels.mat', 'skinPixels');
end

% We then compute the image with data yet and we overwrite the data
for i = 1:length(alreadySeen)
    fileName = char(alreadySeen(i));

    % If fileName is equal to '0', it means there is no more images to deal
    % with
    if strcmp(fileName, '0')
        break
    end

    skinPixels.(strcat('f', fileName)) = cropAndLin(strcat('dataset/training/', fileName, '.jpg'));

    save('dataset/training/skinPixels.mat', 'skinPixels');
end


end
```

The updateBackground function is very similar. The cropAndLin function is suppose to make the user crop the image via a dialog box in which he can draw a rectangle like the following one.

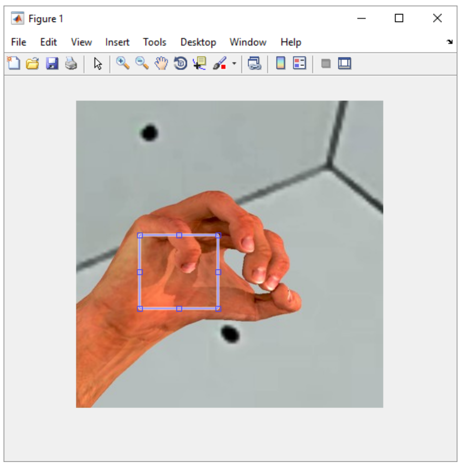

Figure 3: Crop Dialog Box

After drawing the rectangle, the pixels in it are put in a row (3 by m*n array instead of m by n by 3 array), and they are finally saved in the skinPixels.mat file.

The data was not gathered on every training images because it would have been too long. The processing has been done with 200,000 skin pixels and 500,000 background pixels, which is enough.

B *Thresholding*

For every pixels of the image that we want to split, we compute both the aposteriori skin and aposteriori background probability, and we put true or false in the associated pixel of a binary image so that the binary image has in white the hand and in black the background. To compute these aposteriori probabilities, we use the Bayesian theorem saying $p(pixel|s) =$

$\frac{p(s|pixel)p(s)}{\sum_{state} p(state|pixel)p(state)}$ with s a state. Now to compute the likelihood probability, we will use the following method.

1 Gaussian

We will first assume that both states (skin and background) can be estimated as a gaussian. This gaussian is estimated with the function getGaussianEstimate that returns the mean and the covariance matrix according to data. The apriori probabilities are set to 0.5 for both states. Now the threshold over all the test folder is done by the threshold function and gives rather good results.

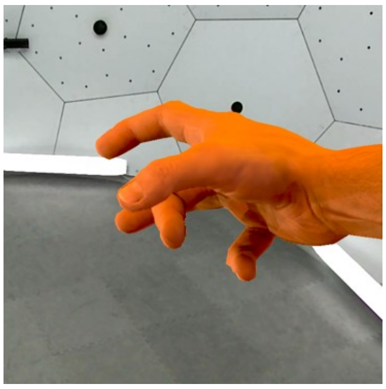
Figure 4: Original image

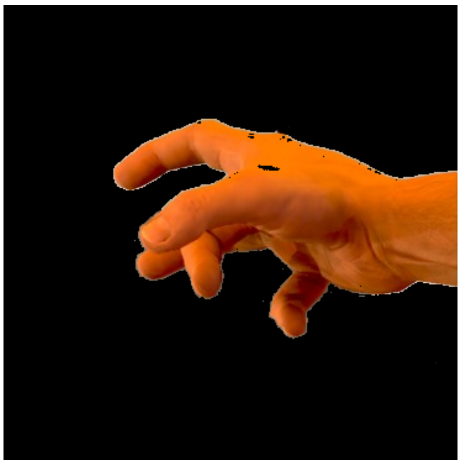
Figure 5: Segmented Image with the Hand Only

However, two problems can be noticed. The first one is that if the background is close to a hand, it will be taken as the hand as in the following example.

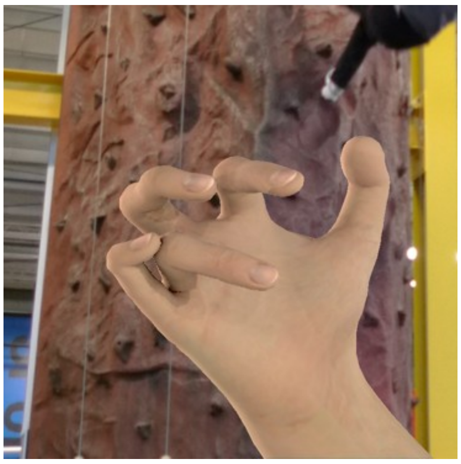
Figure 6: Original image

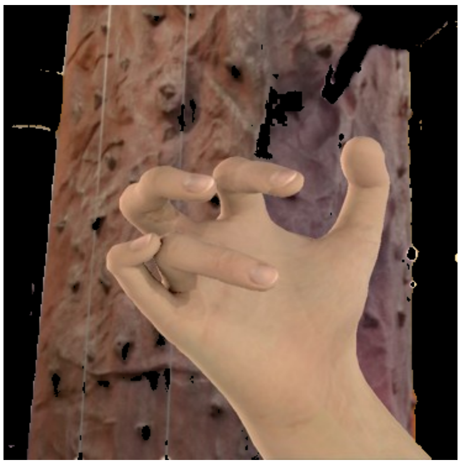
Figure 7: Segmented Image with some Background

To solve this problem I tried to change the apriori probabilities, to grant more importance to the background. I thus put the skin apriori to 10^{-10} and the background apriori to $1 - 10^{-10}$. At these probabilities, the thresholding of the previous image began to improve slightly but the other hands began to be worse because the shaded part of hands was not considered as hand anymore.

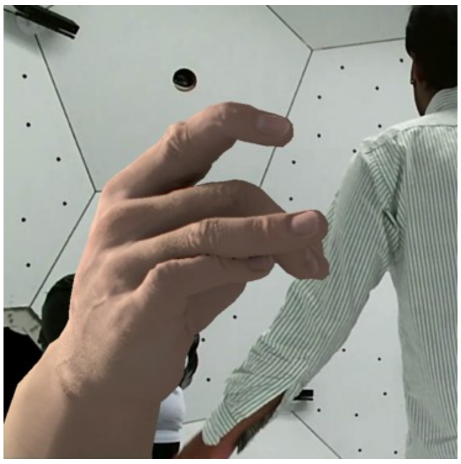
Figure 8: Original image

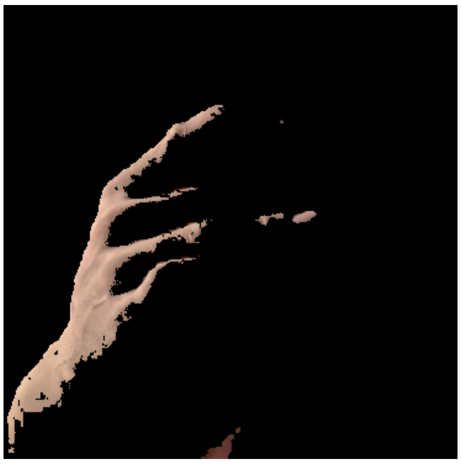

Figure 9: Segmented Image without Shaded Parts

We conclude that this idea was not a very good idea.

Now the other problem is the following.

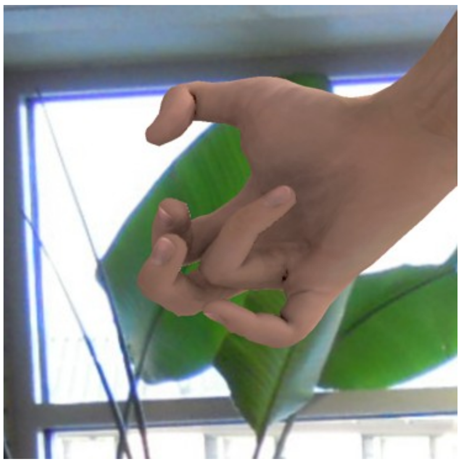

Figure 10: Original image

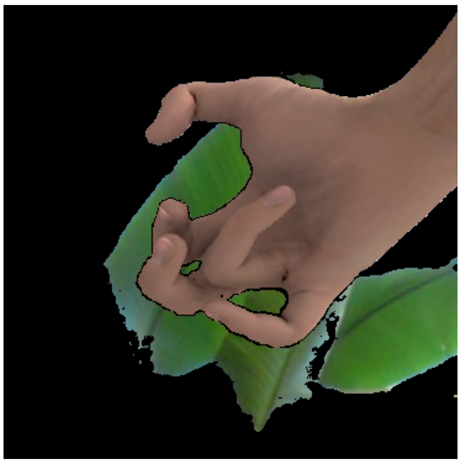

Figure 11: Segmented Image with some Background

Here the leaves are considered as hand even if the colour is not close to the one of the hand. This gives rise to an important question: can the background really be estimated as a gaussian ? Indeed, the background has several different colours, so it may not be a good idea to estimate the background likelihood as a gaussian. This is why I tried to estimate it with the kernel density estimation.

2 Kernel Density Estimation

To implement this, I used the mvksdensity matlab function which is a kernel smoothing function estimate for multivariate data. I used a bandwidth of 0.05. The kernel type is by default a gaussian. Unfortunately, this technique makes things worse for most of the images.

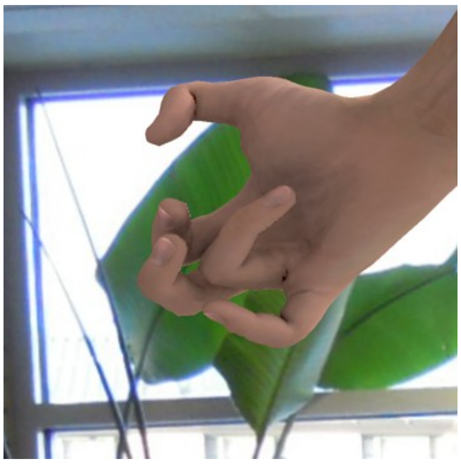

Figure 12: Original image

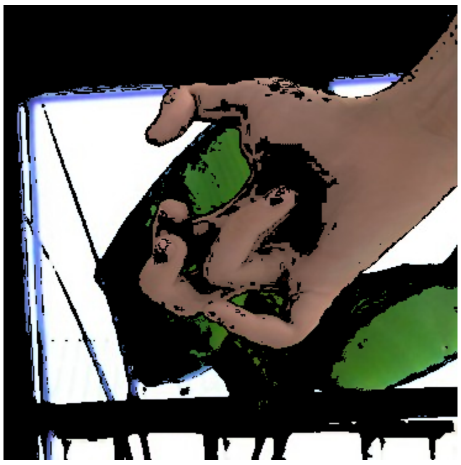

Figure 13: Segmented Image

Here is another bad segmentation.

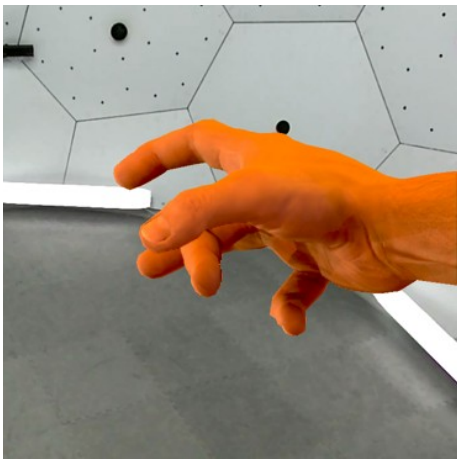

Figure 14: Original image

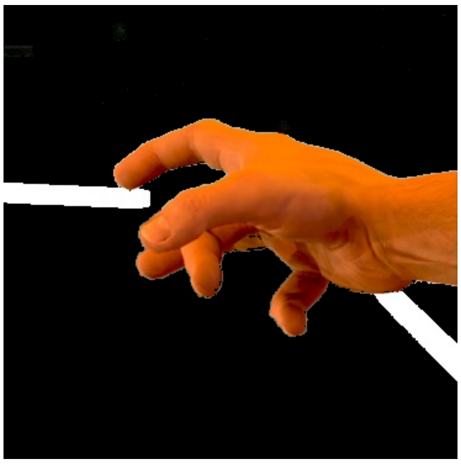

Figure 15: Segmented Image

I must admit that I have trouble to know why things are worse. It may be because I did not take enough background points, or maybe because the selected points did not reflect well all the possible background. Indeed, I could not take all the background points because the mvksdensity function took too long so I was forced to take only 100 points among the 500,000 of the background. However, if we look at the background points (Figure 16), we notice that there is seemingly no green points, i.e. no leaves points, which means that even with all the background points, things may not have been better.

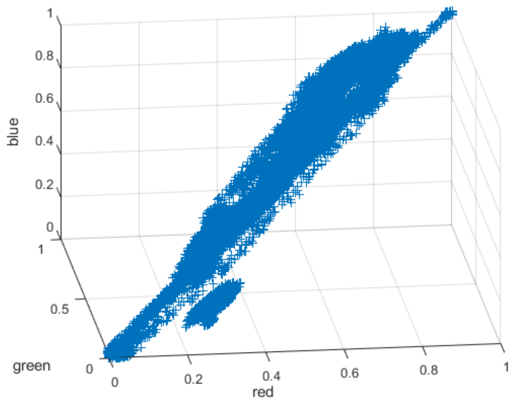

Figure 16: Pixels Distribution of the background

In the latter figure, we can notice 2 things. First it is not that crazy to consider the background pixels as one gaussian because it is not very far from it. Second, the rgb space is not fully used. It may have been better to go in a different colour space like YCbCr in which colour is kind of better described. I did not change the colour space because I wanted to fit in the pattern recognition class and I did not want to use colour tools but it would have been a good idea I think.

IV Conclusion

Eventually, when we estimate both skin and background states as gaussian, the segmentation works rather good. Yet, it is not strong enough when the background looks like the skin colour. For the kernel density estimation, results were rather disappointing.