Hand segmentation

General information

This feature is one of the essential features of the system. Many other features rely on good segmentation of only the hands with the background.

The feature receives a regular three-channel image and output GMM (Gaussian Mixture Model) model and a list of labels that belong to the hands and the sleeves.

Implementation

This feature implementation is part of the calibration at the beginning of the AR application. First, we draw hand and sleeve contours on the output image, so the user puts his hand inside the contours. We give the user countdown on the screen to know how long he needs to keep his hand inside the contours. The countdown begins with the substruction background image (the first image in the calibration flow) and the background and the hand. It required a match of a least 25% of the hand inside the contour.

After capturing the image with a good match between the hand and the contour, we convert the image to LAB color space. We needed a good separation between the hand and the background for good hand segmentation. The main problem could be shadows that can double the hand and create an odd shape. There are two main color spaces HSV and LAB. Those spaces have a unique channel for the lightness that, by removing it, we can overcome such lightning problems. We choose LAB for his ability to distinguish better between colors. It does not have the issue of the Helmholtz–Kohlrausch effect (when the intense saturation of a spectral hue is perceived as part of the color’s luminance). In addition, HSV does not match their perceptual analogs to the original image.

The central part of the flow is the Gaussian Mixture Model. After the conversion to LAB, we removed the L channel to overcome the problems that we mentioned before and then trained the GMM model on that image. The GMM EM is a soft clustering algorithm when all the data points are assigned to all clusters with a certain weight (each point is likely to be in a specific cluster). The histogram of an image contains a mixture of gaussian distribution that describes each part of the image. Therefore, this algorithm is good for fuzzy image segmentation by separating the Gaussians to N components we choose. The algorithm is an iterative Estimation and Maximization until there is a convergence.

The output of the GMM model is a labeled image where each label is the gaussian that the pixel has the best probability to be. In this part, we count the appearance of each label inside the hand and sleeve contours. We choose the most labeled in each contour and assign that label to hand, sleeve or background. The output is two lists of labels related to the hand and the sleeve. We sort good labels to one and bad labels to zero for each segmentation. In the end, we get a masked image with only hands or only sleeves.

Assumptions and limitations

First and most important, we assumed a uniform light condition because the feature is based on colors. Also, the color of light should be natural and not colored to maintain good diversity and resolution of colors.

Second, in the beginning, we did only a two-component GMM model, and we assumed that the hand was the main object in the image. But then we came across a problem when the hand wasn’t the main object, and then the GMM model segment also objects in the background as hands as they had similar colors. Therefore, we had to raise the amount of Gaussian mixtures to four. To distinguish between the hand and the sleeve, the user needs to match them to a contour presented on the image.

At a later stage, when we started to connect all the components of the AR system, we understood that when following the table corners dynamically, we had a problem with hiding. We solved it (there is an explanation next in the paper) by segmenting the sleeve. Therefore, another limitation is that we need a sleeve to operate the system. The sleeve’s color needs to be far enough (in AB color space) from the hand color so that the GMM will distinguish between them well enough.

Real-time adaptation

To maintain good segmentation, we had to train the GMM model with the best resolution image to get the best diversity of Gaussian mixtures. The best resolution is a trade-off to computational run time. We had to change the H.264 stream protocol parameters to get a good enough image with correction code to get the best quality from the camera to the PC. Another parameter is the computation time of the GMM; more components mean more computation time. We trained with full resolution at the calibration process, but later in the process, we had to resize the image to a smaller size and give up upon resolution for a run time between two frames.

Results

The result of the hand segmentation was pretty good but sensitive to light conditions.

[full 4 channel segmentation]

[Jacard image with segmentation and canny filter on specific ROI to see the match]

Some GMM models segment both hand and sleeve contour with the same color, apparently because of darker shades of shadows on the hand and the sleeve. We had to choose if to take only the inner segmentation but lose the resolution of the segmentation or recalibrate the GMM model.

[problem with edges of the segmentation]

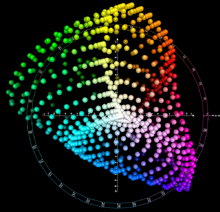
We had to choose the best number of components for the GMM while we had the trade-off resolution and computation time.

[4 images with 2comp 4comp 6comp 8comp] and explain about each of them

Two components in the most robust with noise on the hand but not possible because we needed at least three components.

A high number of components will cause high noise in the image and won’t distinguish between objects in the image.

We tried several sleeves for the calibration, and in the end, we used a blue sleeve as it gave us the best distinguish between the hand and the sleeve. We can see from the LAB color space top view that the shades of blue are most far from the body color.



# <https://en.wikipedia.org/wiki/CIELAB_color_space#Where_CIELAB_is_used>

# <https://en.wikipedia.org/wiki/HSL_and_HSV#Disadvantages>

# <https://en.wikipedia.org/wiki/Helmholtz%E2%80%93Kohlrausch_effect>

# <https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html>

# <https://en.wikipedia.org/wiki/Mixture_model#Gaussian_mixture_model>

# Interduce to image processing – Ben Gurion University.

# <https://en.wikipedia.org/wiki/EM_algorithm_and_GMM_model>