Image Segmentation System Design Using Colour Eigen Structures on FPGA

O.Naga Ramya

August 2020

Summer Internship Project Report
National Institute of Technology, Andhra Pradesh

Supervised by
Dr. V. Sandeep *Dept. EEE NIT AP, India*
K. Prasanna Kumar *Dept. ECE IISC, India*

Índice

Department of EEE ,NIT AP,Andhra Pradesh, India

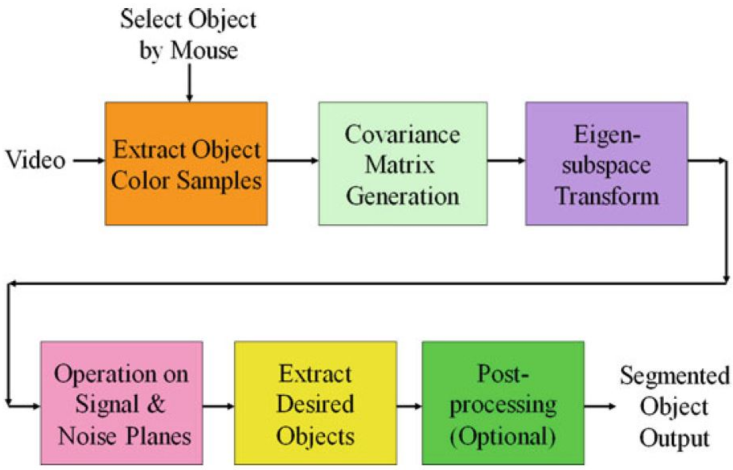
1. Introduction

image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as image objects). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image.

Image segmentation is a mid-level processing technique used to analyze the image and can be defined as a processing technique used to classify or cluster an image into several disjoint parts by grouping the pixels to form a region of homogeneity based on the pixel characteristics like gray level, color, texture, intensity and other features. The segmentation process can be divided into various category based on the parameter selected for segmentation like pixel intensity, homogeneity, discontinuity, cluster data, topology etc. Each approach has its own advantages and disadvantages.

Our objective is to build a **Image Segmentation System Design Using Color Eigen Structures on FPGA**.

2. Our Approach



Flow Chart Representing Steps Involved in Image Segmentation Using Color Eigen Structures

We adopted the method which dependent on the characteristics of eigen-structure.The eigen-subspaces are obtained from eigen-decomposition of the covariance matrix, which is computed from the selected color samples. Hence, the color space can be transformed into the signal subspace and its orthogonal noise subspaces. After statistical analysis of eigenstructure of target color samples, the color eigen-structure segmentation algorithm are then designed to extract the desired objects, which are close to the color samples.

2.1. Color Eigen-Subspaces

Assumed the desired objectsexhibit an average colorsensation, The kth sample in the RGB color vector is given by

$$s_k = [r_k, g_k, b_k]^T$$

where $r_k, g_k, and b_k$ are red, green, and blue levels of the kth sample in each color plane and the superscript T denotes the transpose of the argument vector.It is noted that the derivations can be applied to any other color space.However,we develop the algorithm for the RGB color space only.In orderto divide the three-dimension color spaces into noise and signal subspaces, we further assume that the desired objects contain no more than two large-displaced colors in the average sense. In other words, the number of the desired colors is limited to $p = 1 or 2$.As to the texture or the shadow effect of the desired objects, the variation of colors in the desired objects are modeled as independent noises and expressed by

$$g_k = s_k + n_k = [r_k, g_k, b_k]^T + [n_{r,k}, n_{g,k}, n_{b,k}]^T$$

where $n_{r,k}, n_{g,k}, andn_{b,k}$ are the kth sampled color noises, which are assumed to be statistically independent to the desired color vector s_k and uncorrelated with each other. The covariance matrix Rs of the sampled color vectors is defined as

$$R_g = E[g_k g_k^T]$$

Since the number of average color vectors, p is limited under two, i.e., p = 1 or 2, the noise free color covariance matrix can be expressed by p principal components as

$$R_s = \sum_{i=1}^{p} \lambda_i v_i v_i^T$$

where λ_i represents the ith eigenvalue of R_g and v_i denotes its corresponding eigenvector. The span of $s_i, i = 1, ..., p$ is equal to the span of $v_i, i = 1, ..., p$, which is called the signal subspace.Due to the independent assumption of sample noises, the covariance matrix of the sample noises can be modeled as

$$R_n = \sigma_n^2 I$$

The covariance matrix of sampled color vectors composed of both signal and noise components can be expressed by

$$R_g = R_s + R_n = \sum_{i=1}^{p} \lambda_i v_i v_i^T + \sigma_n^2 I = \sum_{i=1}^{p} (\lambda_i + \sigma_n^2) v_i v_i^T + \sum_{i=p+1}^{3} \sigma_n^2 v_i v_i^T$$

It is noted that the random noises in the average sense do not change the direction of original signal subspace but add the noise power (variation) σ_n^2 to the true eigenvalues of R_s. The remaining subspaces in the RGB color coordinate system, which are called the noise subspaces, become the span of $v_i, i = (p+1), ..., 3$. It is obvious that the eigen vectors of a symmetrical matrix are orthogonal to each other. Accordingly, the signal subspace and noise subspace will be orthogonal to each other.

2.2. Color Eigen-Subspaces Segmentation

For given sampled color vectors g_k of the target objects, we can obtain the sampled covariance matrix $\hat{R}_g$ as in (2.1). Through the eigen-analysis procedures, we can obtain the eigenvectors $\hat{v}_1, \hat{v}_2$,and $\hat{v}_3$ corresponding to the eigenvalues, $\hat{\lambda}_1, \hat{\lambda}_2$, and $\hat{\lambda}_3$ which are arranged in the descending order as

$$\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \hat{\lambda}_3$$

Since the covariance matrix $\hat{R}_g$ can be expressed by

$$\hat{R}_g = \sum_{i=1}^{p} (\hat{\lambda}_i + \hat{\sigma}_n^2) \hat{v}_i \hat{v}_i^T + \sum_{i=p+1}^{3} \hat{\sigma}_n^2 \hat{v}_i \hat{v}_i^T$$

The least eigenvalue of the sampled covariance matrix can be used as the estimator of the noise power as

$$\hat{\sigma}_n^2 \approx \hat{\lambda}_3$$

and the corresponding eigenvector is estimated by $v_3 \approx \hat{v}_3$. The estimated principal eigenvalues are given by

$$\lambda_1 \approx \hat{\lambda}_1 - \hat{\lambda}_3$$

and

$$\lambda_2 \approx \hat{\lambda}_2 - \hat{\lambda}_3$$

While the estimated eigen vectors are $v_1 \approx \hat{v}_1$ and $v_2 \approx \hat{v}_2$.

2.3. Algorithm Based on Color Eigen-Structures

In order to classify the input color vector into the signal and the noise subspace, we can project the color vector on the eigenvectors of the sample covariance matrix as

$$y_{i,k} = v_i^T.g_k, \quad for\ i = 1, 2, 3$$

Now, we should statistically analyze the projection length of $y_{i,k}$ by taking expectation of the power as

$$E[y_{i,k}y_{i,k}^T] = E[v_i^T g_k g_k^T v_i] = v_i^T \hat{R}_g v_i = \hat{\lambda}_i$$

Thus, the average length of the eigenvector projection should become

$$| y_{i,k} |=| v_i^T.g_k |= \sqrt{\hat{\lambda}_i}, \quad for\ i = 1, 2, 3.$$

For the principal component approach, we can simply detect the color pixel by choosing

$$| y_{1,k} |=| v_1^T.g_k |\geq \sqrt{\hat{\lambda}_1}$$

This is the so-called signal subspace projection. Any pixel color vector, which has large enough projection onto the direction of the principal color vector, will be treated as the object pixel for color segmentation.

we classify the color space by using both signal and noise subspaces by determining the threshold values in the transformed color spaces. First, we should detect the signal space component more precisely. In order to include 97.5 % confidence interval of the principal projection, we propose the signal-subspace detection criterion as

$$\sqrt{\hat{\lambda}_1 + k_{1s}\sigma_{\lambda_1}} \geq | y_{1,k} | \geq \sqrt{\hat{\lambda}_1 - k_{1s}\sigma_{\lambda_1}}$$

where k_{1s} is a constant that equals to 3. The deviation $\sigma_{\lambda_1}$of the first principal eigenvalue is given by

$$\sigma_{\lambda_1}^2 = \frac{\lambda_1^2}{N}$$

We relax the lower bound by three deviations to include the possible shadow colors and add the upper bound with three deviations to exclude the unwanted brighter colors. Thus, we can eliminate the incorrect luminance pixels as possible. The pixels, which meet the signal subspace criterion stated above, could be very possible mixed color pixels.

In order to further exclude the mixed color pixels, we should use the noise space criterion to remove the pixel color pixels from the signal subspace pixels, which satisfy the signal space criterion stated above. The noise subspace criterion can be discussed in two cases:$p = 1andp = 2$. For $p = 1$, the noise subspace now becomes the span of v_1, v_2. We should perform the noise subspace criterion as

$$| y_{i,k} |=| v_i^T.g_k |> \sqrt{\hat{\lambda}_i - k_{in}\sigma_{\lambda_i}}, \quad for\ i = 2, 3.$$

to remove the unwanted pixels, where k_{in} is a constant to specify the confidence interval of the noise. We know that the pixels, whose projections to the noise subspace should be as small as$\hat{\lambda}_i$ for $i = 2, 3$, are matched with the desired color modal. For any other pixels with mixed colors, their color vectors project onto the noise subspace will be larger than $\hat{\lambda}_i$ for $i = 2, 3$. Similarly, we can keep the desired pixels once we find their projections to the noise subspace are beyond the limits of $k_{in}.\sqrt{\hat{\lambda}_i \pm k_{2n}\sigma_{\lambda_i}}$ for $i = 2, 3$. For For p = 1, we perform the detection of

$$k_{1n}.\sqrt{\hat{\lambda}_i + k_{2n}\sigma_{\lambda_2}} \geq| y_{2,k} |=| v_2^T.g_k |\geq k_{1n}.\sqrt{\hat{\lambda}_i - k_{2n}\sigma_{\lambda_2}}$$

and

$$k_{1n}.\sqrt{\hat{\lambda}_i + k_{2n}\sigma_{\lambda_3}} \geq| y_{3,k} |=| v_3^T.g_k |\geq k_{1n}.\sqrt{\hat{\lambda}_i - k_{2n}\sigma_{\lambda_3}}$$

to remove the unwanted pixels. We set the constants$k_{1n} = 3$ and $k_{2n} = 3$ in our experiment to achieve the best results. In above equations, $\sigma_{\lambda_2}$and σ_{λ_3} denote the deviation of the second and third eigenvalues respectively given by

$$\sigma^2_{\lambda_2} = \frac{\lambda_2^2}{N}$$

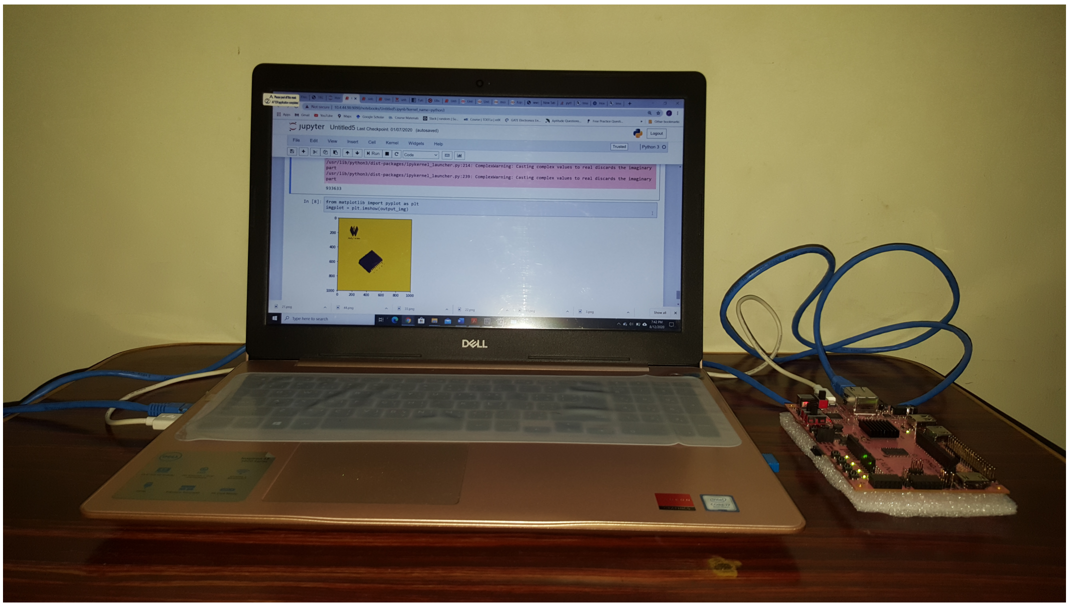
and

$$\sigma^2_{\lambda_3} = \frac{\lambda_3^2}{N}$$

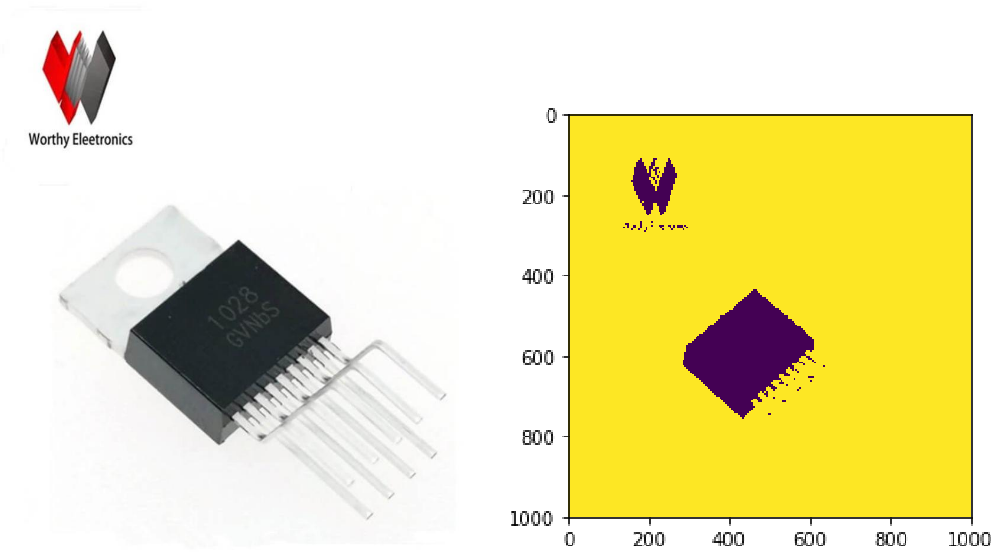
To clearly exhibit the desired color in the segmented images, we only show the desired color and undesired color objects by dark pixels with gray level=0 and bright pixels with gray level= 255, respectively.We assigned gray value black(black) to the pixels which satisfy the signal space criteria and not present in noise space and white to the pixels which satisfied the above two conditions.

3. Results

For testing our algorithm based on Color Eigen Structures we taken two images one palm image and a silicon chip image. We used the harware(FPGA) **PYNQ-Z2** which is a Python productivity for ZYNQ, it has ZYNQ-7020 SOC on the board.And we taken a PYNQ img of version 2.5 to boot the board. We uploaded these test images on to board and accessed them through code. We implemented the above algorithm using Python programming language and we run the code on PYNQ board. These are the output results for our input images.

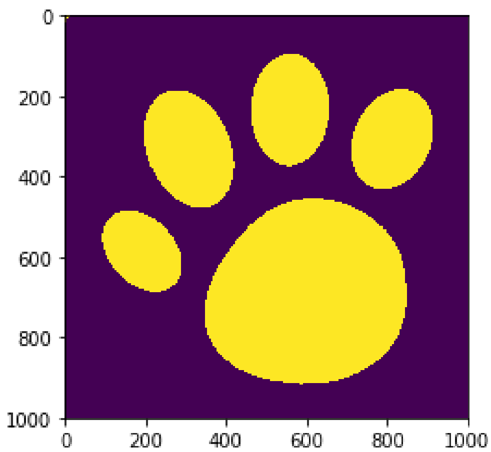


Hardware Set Up Using PYNQ-Z2 Board



Input Image for Segmentation and Output Image Object is the Chip(purple)


Input Image for Segmentation



Output Image Object is the Plam (yellow)

4. References

https://en.wikipedia.org/wiki/Image$_{s}egmentation$
$Chapter - 2 - Image - Segmentation - with - Eigen - Subspace - Yang - Hao$
$https : //en.wikipedia.org/wiki/Eigenvalues_{a}nd_{e}igenvectors$
$https : //en.wikipedia.org/wiki/Covariance_{m}atrix$

Contact Details

O.Naga Ramya
ogireddynagaramya@gmail.com
M.no:9515426799
Dept.Electronics and Communication Engineering,
RGUKT Nuzvid, Krishna District,
Andhra Pradesh, Pin:521202,
India