Image Segmentation Using Color, Coordinate, and Complement Features

Salah Ameer
Adjunct Professor in Ontario Colleges, Canada

Abstract

A novel Eigen formulation is proposed for image segmentation. A vector composed of three features: normalized intensity, x-coordinate, and y-coordinate, is used to represent each pixel. A fourth component (complement) can be attached to the vector to produce a "unit" vector. The auto-correlation matrix is computed for the image using this unit vector. The first component (corresponding to intensity) from all Eigen vectors, obtained from the auto-correlation matrix, are used as the multi-level thresholds. The number of thresholds can be further increased by attaching the complement of each feature rather than one complement for all features. The process can be generalized to any feature space as demonstrated by the incorporation of the RGB values. Results on a wide range of images are demonstrated to show the effectiveness of the proposed schemes. The significance of this research lies in the use of a global (in contrast to local) framework for image segmentation.

Keywords— Image segmentation, Eigen structure, Multi-level thresholding.

1. Literature review

Multi-level image thresholding is essential in many image segmentation tasks needed by many computer vision schemes. The ultimate goal is to delineate the image in such a way to obtain useful descriptions of the objects comprising the scene. To achieve this goal, many algorithms has been (and still being) developed. Details regarding categorization of these algorithms and the feature space used can be found in many traditional survey papers such as [1]. In fact, the field is so diverse that there are survey papers on a single subcategory e.g., [2–4].

2. Introduction

In this paper, the image segmentation problem is considered as a multilevel thresholding task. A simple but effective Eigen structure is proposed as a solution based on the results of a recent work [5].

The idea is simply to concatenate the features (a mixture of intensity, color, and coordinate values) in one vector for each pixel. An addition component (the complement) is then appended to obtain a unit vector.

The mathematical descriptions are illustrated in section 3 followed by some experimentation in section 4. Some conclusions and suggestions for future work are discussed in sections 5 and 6.

3. Method

Without loss of generality, the original image is normalized to the interval [0, 1] (or [–1, 1]). Each pixel has three features, *intensity* g_i, x coordinate, and y coordinate. Each feature is then concatenated (using all the pixels in the image) to produce a column vector of size Nx1, N is the number of pixels in the image. The vector representing each pixel is then extended to a 4D "*unit"* vector given by

$$G_i = \left[g_i \quad x_i \quad y_i \quad \sqrt{1 - \mathrm{w}_1 g_i^2 - \mathrm{w}_2 x_i^2 - \mathrm{w}_3 y_i^2}\right] \quad (1)$$

Where w_i is a weighting vector that sums to one. In this work, the uniform, power dependent, and variance weighting schemes have been tested for w_i with minor differences in their performances. The author is not confident that other schemes can provide higher performance results.

(A_G) of size 4x4 is then constructed from G as

$$A_G = G^T G \quad (2)$$

The next step is to solve the Eigen formula,

$$A_G V = \lambda V \quad (3)$$

The Eigen vectors of A_G represent the axes of inertia for the data set. The largest Eigen vector V_{max} (corresponding to the maximum Eigen value λ_{max}) points toward the direction of maximum inertia [5]. Similarly. V_{min} (corresponding to the minimum Eigen value λ_{min}) points toward the direction of minimum inertia No components of each resulting Eigen vector except for the first one, corresponding to the intensity, are suitable to be used as a discriminator.

This motivates the author to use the first component of all Eigen vectors (V_{max}) as thresholds. Care should be taken to remove out of range values (outside [0, 1] or [–1, 1]) to avoid unnecessary computations.

An alternative scheme would be to extend Eq. (1) to have six components given by

$$G_i = \left[g_i \quad \sqrt{1 - g_i^2} \quad x_i \quad \sqrt{1 - x_i^2} \quad y_i \quad \sqrt{1 - y_i^2}\right] \quad (4)$$

Effectively, there can be four schemes as follows

- PosXY4: Eq. (1) is used with intensity and coordinates normalized to [0, 1]. Power weighting is used, each component is weighted by the sum of the squares of all its elements.
- PosXY6: Similar to PosXY4 using Eq. (4).
- NegXY4: Similar to PosXY4 with intensity and coordinates normalized to [–1, 1].
- NegXY6: Similar to NegXY4 using Eq. (4).

In fact, the schemes can be easily generalized to add other features like color for example. The main obstacle is the weighting scheme choice for Eq. (1) and whether to pick Eq. (4) instead. Further investigation is reported in section IV.

Generalizing beyond color components is currently under investigation.

The performance was assessed through the traditional root mean square error (RMSE) given by

$$RMSE = \sqrt{\frac{1}{\|x\|}\sum_{m,n}\left(x_{mn-}y_{mn}\right)^2} \quad (5)$$

where, x and y stand for original and segmented images, and $\|x\|$ is the cardinality of the set. Adjustment should be placed when the range of images are different. In addition, it is unfair to compare performance between images having different number of segments.

Another evaluation scheme is the SSIM given by [6]:

$$SSIM = \frac{2\mu_x\mu_y+C_1}{\mu_x^2+\mu_y^2+C_1}\frac{2\sigma_{xy}+C_2}{\sigma_x^2+\sigma_y^2+C_2} \quad (6)$$

where, μ is the mean, σ^2 is the variance, C_1 = 0.0001, C_2 = 0.0009, and σ_{xy} is the covariance between x and y.

4. Experimental results

A set of test images are shown in Figure 1 together with their ground truth. Figure 2–5 illustrate the results obtained using schemes PosXY4, PosXY6, NegXY4, and NegXY6 respectively. Table 1–2 list the values of RMSE and SSIM respectively for all the four proposed schemes implemented on images in Figure 1.

Care should be taken when using RMSE and SSIM as they have deficiency regarding scale. All the segmented images will have inferior values using these measures if the images were scaled back to [0, 1] or [–1, 1] instead of using the mean of the segmented region.

Keeping in mind that more results are needed, comparison of Figure 2 and Figure 3 reveals a favour for schemes PosXY4. On the other hand, comparison of Figure 4 and Figure 5 reveals a favour for schemes NegXY6.

To further illustrate the effectiveness of the proposed schemes, some gray scale images from Berkley Segmentation Database (BSD) [7] are segmented in Figure 6. The values of RMSE and SSIM are listed in Table 3–4 respectively.

It is easily noticed from these images that using the normalization [–1, 1] produces higher number of regions compared to [0, 1]. The latter has better performance for thresholding tasks while the former is advantageous for segmentation tasks. Extensive results may be needed in this regard.

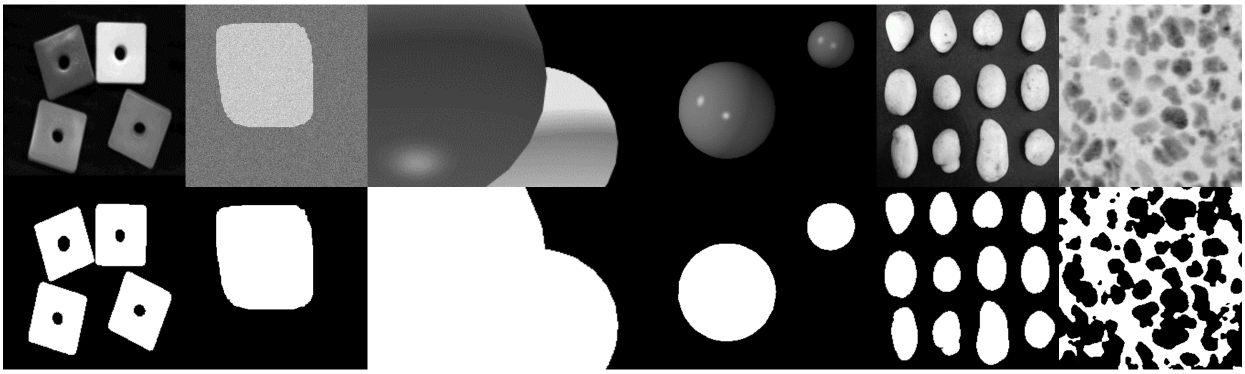

Figure 1. Test images used and their ground truth

Figure 2. Resultant images for scheme PosXY4 with number of thresholds (left to right): 3, 3, 2, 2, 2, and 2

Figure 3. Resultant images for scheme PosXY6 with number of thresholds (left to right): 4, 3, 3, 2, 4, and 3.

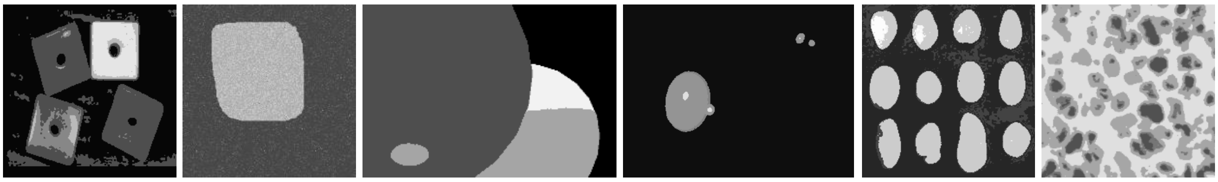

Figure 4. Resultant images for scheme NegXY4 with number of thresholds (left to right): 4, 3, 3, 3, 4 and 3.

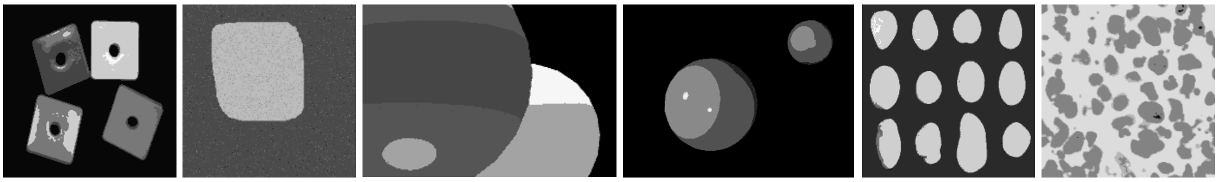

Figure 5. Resultant images for scheme NegXY6 with number of thresholds (left to right): 6, 6, 6, 6, 6, and 6.

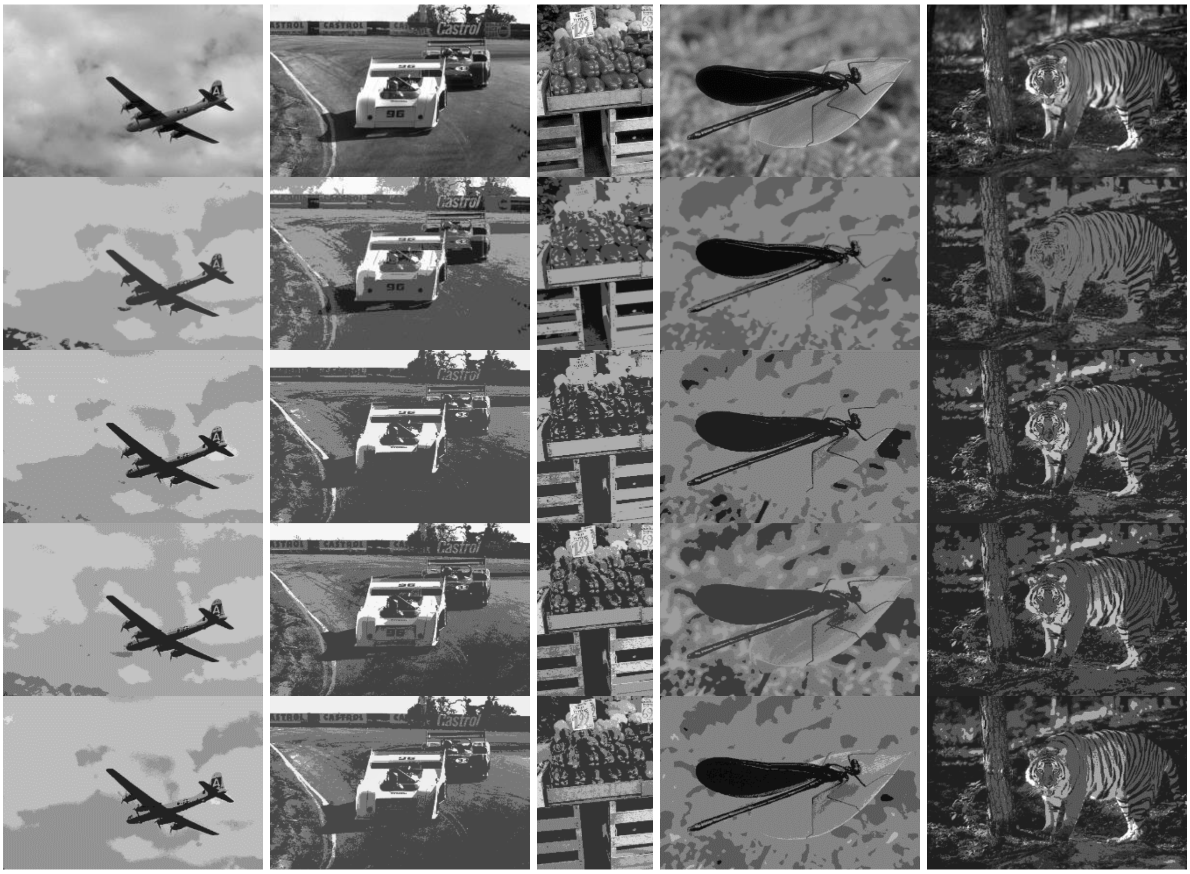

Figure 6. (Top to bottom) Original images from BSD followed by segmentations using schemes PosXY4, PosXY6, NegXY4, and NegXY6 with their respective number of thresholds, (images are left to right) 3096 (3, 4, 3, 6), 21077 (3, 3, 3, 6), 25098 (2, 3, 3, 6), 35070 (3, 3, 3, 6), and 108005 (2, 4, 3, 6).

Table 1. RMSE for the results of schemes PosXY4, PosXY6, NegXY4, and NegXY6 on images in Figure 1

Image / Method	1	2	3	4	5	6
PosXY4	0.055	0.063	0.091	0.060	0.084	0.071
PosXY6	0.045	0.064	0.120	0.035	0.070	0.082
NegXY4	0.090	0.054	0.044	0.127	0.074	0.060
NegXY6	0.051	0.059	0.047	0.028	0.074	0.085

Table 2. SSIM for the results of schemes PosXY4, PosXY6, NegXY4, and NegXY6 on images in Figure 1

Image / Method	1	2	3	4	5	6
PosXY4	0.978	0.949	0.921	0.938	0.962	0.921
PosXY6	0.985	0.948	0.855	0.979	0.974	0.893
NegXY4	0.940	0.963	0.982	0.653	0.971	0.943
NegXY6	0.981	0.956	0.980	0.986	0.970	0.883

Table 3. RMSE / SSIM for the results of schemes PosXY4, PosXY6, NegXY4, and NegXY6 on images from BSD Figure 6

Method / Image	**PosXY4**	**PosXY6**	**NegXY4**	**NegXY6**
3096	0.0574 / 0.9184	0.048 / 0.942	0.047 / 0.943	0.052 / 0.932
21077	0.1052 / 0.8859	0.087 / 0.923	0.069 / 0.952	0.076 / 0.942
25098	0.1051 / 0.8963	0.113 / 0.876	0.083 / 0.935	0.086 / 0.930
35070	0.0611 / 0.9157	0.060 / 0.918	0.078 / 0.850	0.052 / 0.939
108005	0.0967 / 0.8069	0.064 / 0.922	0.070 / 0.903	0.057 / 0.937

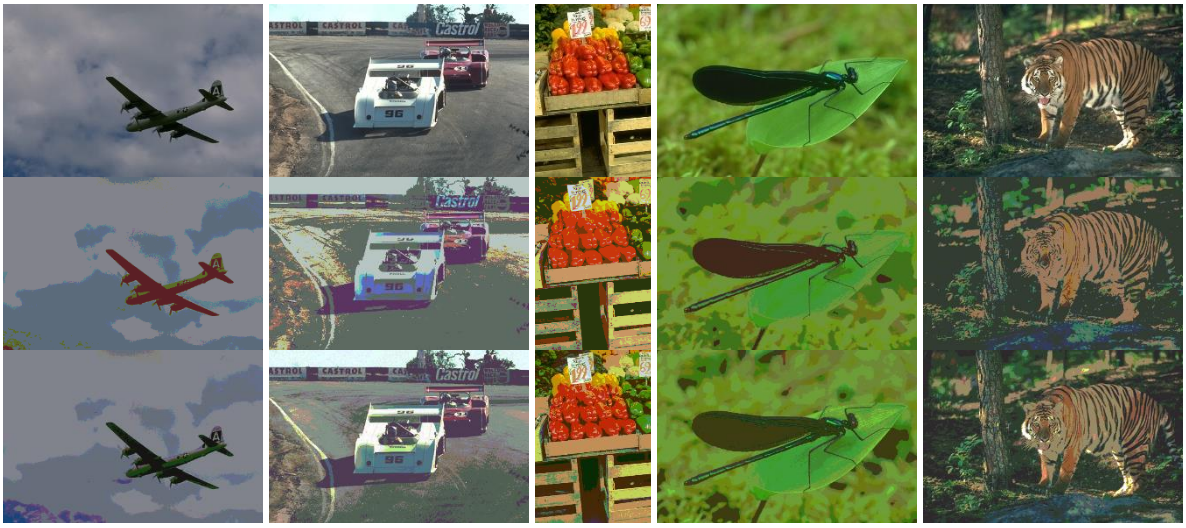

Figure 7. (Top to bottom) Original colored images from BSD followed by segmentations using two schemes based on Eq. (1) and Eq. (4) with their respective number of regions, (images are left to right) 3096 (27, 343), 21077 (12, 343), 25098 (27, 343), 35070 (27, 343), and 108005 (12, 343)

Extension to color images is shown in Figure 7 using Eq. (1) with [0,1] normalization (weighting similar to that of PXY4) and Eq. (4) with images normalized to [–1,1]. Please note that using Eq. (1) can result in upto 4 thresholds per color component or a maximum of 64 segments per image. On the other hand using Eq. (4) can result in up to 6 thresholds per color component or a maximum of 216 segments per image.

Coordinate and color features can be combined to perform segmentation, as shown in Figure 8–10, are simple explorations of the effect on limiting the choice of Eigen vectors according to their respective Eigen value. In fact, using the blue channel is sufficient as illustrated in Figure 11. However, this is image dependent and seems to be influenced by the weighting scheme used as indicated in Eq. (1).

5. Discussion

Simple algorithms are suggested in this paper to perform multi-level image segmentation. The algorithms are fully automatic and no adjustment to any sort of parameters is needed. The proposed schemes are very effective as demonstrated by the values of RMSE and SSIM.

Unfortunately, increasing the number of features may not be the best option as it means more segments can be generated.

Figure 8. (Top to bottom) Segmentations using two schemes based on Eq. (1) and Eq. (4) using color and coordinate features. See Figure 7 for original images

Figure 9. Segmentation of image 3096 using Eq. (4) (left to right). The number of the largest Eigen vectors used is: 9, 6, and 4 respectively

Figure 10. Segmentation of image 3096 using Eq. (1) (left to right). The number of the largest Eigen vectors used is: 5, 4, 3, and 2 respectively

Figure 11: Segmentation of image 3096 using Eq. (1) with the blue channel only (left to right). The number of the largest Eigen vectors used are: 6, 5, 3, and 2 respectively

6. Recomendation and Future Work

More elaboration is needed on the best aggregation used in selecting the thresholds. Selecting all of them may not produce the best performance. For binary thresholding, selecting V_{max} can be used. A combination of all Eigen vectors weighted by their respective Eigen values can also be used. A similar procedure can be adopted for multi-level thresholding.

Work is currently in progress to extend the algorithm to an arbitrary feature space (beyond color and intensity) and finding a suitable scheme to determine the optimum selection and/or weighting. Other color spaces may perform better. A similar argument can be applied for the coordinate system used.

The component(s) added to obtain a unit vector, Eq. (1) or Eq. (4), can be generalized to any fuzzy complement. However, more work is needed to find the best formula and whether significant improvements can be attained, see [5] for some suggestions in this regard.

The preference of using [0, 1] or [–1, 1] may also need further insight as can be implied from Figure 2–5. Figure 6 and beyond for BSD show some trend in preferring [0, 1] for thresholding, while [–1, 1] for segmentation.

A more important aspect is whether the scheme(s) can lend itself to a predefined number of thresholds.

7. References

[1] M. Sezgin, and B. Sankur, “Survey Over Image Thresholding Techniques And Quantitative Performance Evaluation”. *J. Electronic Imag.*, 2004, 13: pp 146–165. DOI: 10.1117/1.1631315.

[2] D. Ilea and P. Whelan, “Image segmentation based on the integration of colour – texture descriptors — A review”. *Pattern Recognition* 2011, 44, pp 2479–2501.

[3] B. Peng, L. Zhang, and D. Zhang, “A survey of graph theoretical approaches to image segmentation”. *Pattern Recognition* 2013, 46 pp. 1020–1038.

[4] R. Unnikrishnan and M. Hebert, “Measures of Similarity”. *Seventh IEEE Workshops on Application of Computer Vision* 2005, pp. 394–400.

[5] S. Ameer, “Image Thresholding using the Complement Feature”. *American Journal of Engineering and Applied Sciences*, 2020, pp 311 – 317. DOI: 10.3844/ajeassp.2020.311.317.

[6] Z. Wang, A. Bovik, H. Sheikh and E. Simoncelli, “Image quality assessment: From error visibility to structural similarity”. *IEEE Trans. Image Process.*, 2004, 13 pp 1–14. DOI: 10.1109/TIP.2003.819861.

[7] D. Martin, C. Fowlkes, D. Tal and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics”. *Proceedings of the 8th IEEE International Conference on Computer Vision*, Jul. 7–14, 2001, Vancouver, BC, Canada, pp: 416–423. DOI: 10.1109/ICCV.2001.937655.

