ATLANTIS PRESS
Atlantis Studies in Uncertainty Modelling, volume 1
11th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2019)

A new approach to Color Edge Detection

Pablo A. Flores-Vidal[a] and Daniel Gómez[b] and Javier Castro[b] and Guillermo Villarino[b] and Javier Montero[a]
[a]Statistics and Operational Research, Faculty of Mathematics, Complutense university,
Plaza de las Ciencias 3, pflores@ucm.es
[b]Statistics and Data Science, Faculty of Statistical Studies, Complutense university,
Avda. Puerta de Hierro s/n, dagomez@estad.ucm.es

Abstract

Most edge detection algorithms deal only with grayscale images, and the way of adapting them to use with RGB images is an open problem. In this work, we explore different ways of aggregating the color information of a RGB image for edges extraction, and this is made by means of well-known edge detection algorithms. In this research, it is been used the set of images from Berkeley. In order to evaluate the algorithm's performance, F measure is computed. The way that color information -the different channels- is aggregated is proved to be relevant for the edge detection task. Moreover, post-aggregation of channels performed significatively better than the classic approach (pre-aggregation of channels).

Keywords: Color edge detection, Multichannel edge detection, RGB, Pre-aggregation, Post-aggregation, Crispy aggregation, Fuzzy aggregation.

1 Introduction

RGB images for image processing have not been employed so often compared with grayscale images. This is specially true in the case of the edge detection problem, where dealing with color images introduces some complications.

The RGB images are built in the RGB space color. This space color it is based in human perception as human vision has three different cone cells, one captures the red luminosity, meanwhile the other two do the same with green and blue luminosity respectively. Human vision employes rods, a second kind of cell, but this one only can process intensity not color [1]. Due to this, "Three numerical components are necessary and sufficient to describe a color..." as it is said by Bogumil [1]. In this sense, the RGB color space should be ideal for retaining all the color information, as it is using three components for it (Red, Green and Blue).

Marr [16] pointed out that color could be relevant due that it "carries information that often has important biological significance". This information could help to distinguish "wheter a fruit is ripe, wheter a leaf is green and supple, wheter an insect is likely to be poisonous, and many other things".

As we will see in the comparatives, the edge detection using RGB channels performs better than just using grayscale edges (see Section 4).

The remaining of this paper is organized as follows: The next section 2 is dedicated to some preliminaries in edge detection problems, including the case of color edge detection. The different approaches for aggregating the color channels are presented in section 3. Finally, in the section 4 and section 5 we present some comparative results and conclusions respectively.

2 Preliminaries

In this section are introduced some needed concepts of image processing and edge extraction problem. Let us denote by I a digital image, and by (i,j) the pixel coordinates of the *spatial domain*. For simplification purposes the coordinates are integers, where each point (i,j) represents a pixel with $i = \ldots, n$ and $j = \ldots, m$. Therefore, the size of an image, $n \times m$, is the number of its horizontal pixels multiplied by its number of verticals. As we are dealing with color images, then a $k = 1, \ldots, \tilde{k}$ index is needed for expressing the number of channels in the image. Thus, let us denote by I=$P^k_{i,j}$ the spectral information associated with each pixel (i,j) at channel k. As well, $I = P^k_{i,j}$ is equivalent to $\tilde{k}$ images of one single channel (grayscale images) $I = \{I^1, \ldots, I^{\tilde{k}}\}$. The values range of this information depends on the type of image considered.

- *binary map*: $I_{bin} = P_{i,j} \in \{.255\}$ (as well it is usually expressed as $\{.1\}$).
- *grayscale image*: $I_{gray} = P_{i,j} \in \{.1, \ldots, 255\}$.
- *RGB image*: $I_{RGB} = P_{i,j} \in \{.1, \ldots, 255\}^3$. (R=*Red*; G=*Green* and B=*Blue*).
- *soft image*: $I_{soft} = P_{i,j} \in [.1]$. As well it is referred as a normalized grayscale image.

Edge detection technique has not a single acepted definition [13]. The most common is to consider it as a technique that identifies the significant luminosity changes in the image.

The concept of edge detection in color images is more advanced than in grayscale edge detection, and this makes it much more complex to deal with. Now, the concept of color similarity becomes crucial. Different approaches to deal with color edge detection have been proposed:

(I) *Individual channel*: The edge detection is computed for each single channel. This is the approach employed in this paper.

(II) *Vector-based approach*: An aggregation function is applied, such a median filter [20], a range operator [22], or other statistical aggregation methods [6].

Other authors ([18, 5]) have considered as a third important approach the one of working with gradients that result of combining from different colors, but we consider that this approach can be included in the vector-based approach and this simplifies the taxonomy of color edge detection. The interesting approach developed in [18] which is based on computing distributions of colors of two hypothesized regions could be considered as a different one, but we did not find many other authors in the literature following this aproach. As well, in this paper we have focused in the specific case of RGB images instead of dealing with multispectral or hyperspectral imagery as it is more common in remote sensing field [15].
From a mathematical point of view an edge detection algorithm is a function that converts a digital image into a binary image. We would like to emphasize that most of edge detection algorithms only deal with grayscale images, meanwhile there are a high number of segmentation algorithms dealing with color images [12, 11]. From this idea, it is clear that an edge detection algorithm transforms an image into a binary image or I_{bin}. In this binary image, the white pixels (the pixels where $P_{i,j}$=1) are those that have been identify by the edge detection algorithm as edge pixels. In the case of I_{bin} being the output of an edge detection algorithm we call it I_{solut}, as this image is the solution of the algorithm.

3 Aggregating channels in color edge detection

Although there exist some approaches (vector-based approaches) that detect edges in color images based on dissimilarity/distance measures between colors, in this work, we have focused on individual channel approach. We have chosen this approach instead of vector-based since it is easy to find situations in which any distance measure tends to compesate significant differences in specific channels. Contrarely, with the individual channel aproach this limitation related to these compesations can be easily avoid. All the aggregation methodologies are applied after the operator's gradient is computed -as well known as *blending-aggregation* phase [9]- for different directions. We are aware of the possibilities of starting our study considering the different directions -horizontal and vertical- of change in the spectral information prior to their gradient aggregation, but we decided to start this research after the blending-aggregation phase for simplification purposes.

(a) *Crispy pre-aggregation (Methodology A)*: Aggregating the intensity components in one single gray channel. This means that first we aggregate with simple addition the different color channels into one single gray channel and then an edge detection algorithm is applied over the resulted grayscale image. This can be seen as the classic method, as it is the most common procedure employed in the literature. When dealing with RGB images we have three channels ($\tilde{k}$=3) I^1_{RGB}, I^2_{RGB} and I^3_{RGB} corresponding to red, green and blue channels respectively.

$$I_{gray} = \frac{I^1_{RGB}+I^2_{RGB}+I^3_{RGB}}{3}$$

As we could be dealing with more channels, a more general formula can be written:

$$I_{gray} = \frac{I^1+\ldots+I^k}{\tilde{k}}$$

As well, it has been used the weighted addition with interesting results [6]. This approach allows giving different importance to each color channel.

Once we have a single grayscale channel, an algorithm's operator is applied following standard procedures: $I_{soft} = edge(I_{gray})$ as a general formula, and the specific cases employed in this paper being

$I_{soft} = Sobel(I_{gray})$ and $I_{soft} = Canny(I_{gray})$. After the operator the thinning -non maximum suppression- is applied. As the final step, over the thinned version of the soft image the binarized image I_{solut} is produced. It results from the soft image after a threshold value is applied (for a detailed explanation of the different steps in edge detection techniques see [9, 8, 7]): $I_{solut} = threshold(I_{soft})$ where $threshold()$ means that a threshold function is applied over the thinned version of the soft image. An scheme of the three methodologies is summarized in Figure 1.

(b) *Crispy post-aggregation (Methodology B)*: Applying an edge detection operator over each channel separately, which produces $\tilde{K}$ different edges maps. Then, all the $\tilde{K}$ resulted binarized images will be aggregated into a single one. We can see in Algorithm 1 this methodology:

Algorithm 1 Crispy post-aggregation

```
1: procedure (Crispy post-aggregation of channels)
2:     for k = 1, ..., k̃ do
3:         I^k_soft = edge(I^k_RGB)
4:         I^k_solut = threshold(I^k_soft)
5:     I_solut = Θ(I^1_solut, ..., I^k̃_solut)
```

(c) *Fuzzy post-aggregation (Methodology C)*: In this case the aggregation function is applied not over the already binarized image but over the soft image corresponding to each color channel. This soft image is made of what we consider as *candidates to be edge* pixels. The binarized image is produced at the last step of the algorithm, following a soft approach that we have called "fuzzy" approach:

Algorithm 2 Fuzzy post-aggregation

```
1: procedure (Fuzzy aggregation of channels)
2:     for k = 1, ..., k̃ do
3:         I^k_soft = edge(I^k_RGB)
4:     I_soft = Θ(I^1_soft, ..., I^k̃_soft)
5:     I_solut = threshold(I_soft)
```

We have used different aggregation functions $\Theta()$:

(I) *The sum*: $I_{solut} = \sum_{k=1}^{\tilde{k}} I_{solut}^{k}$.

(II) *The mean*: $I_{solut} = \frac{\sum_{k=1}^{\tilde{k}} I_{solut}^{k}}{n}$.

(III) *The maximum*: $I_{solut} = \max(I_{solut}^{1}, \ldots, I_{solut}^{\tilde{k}})$.

4 Comparatives and results

The well-known algorithms of Sobel [19] and Canny [2] have been employed. In the case of Sobel's we have used the standard version of it. By contrast, for simplification purposes the Canny's version employed has a Gaussian filter size of 5x5 pixels and $\sigma = 2$, which is enough for a good quality in the edge extraction process. Both algorithms have been tested in the three different methodologies -A, B and C- explained in Section 3. C method has been applied with two different aggregation function, mean (C1) and maximum (C2), meanwhile B method employed only the maximum. In Table 1 we see the maximum F results (corresponding to the human with the highest F-measure values for each image) after applying the different versions of Sobel algorithm -for each methodology- for the first 50 images of the training dataset of Berkeley [17]. The automated threshold value is written next to each F value. The same is showed for Canny's in Table 2 where we used the low threshold value as a 40% of the high threshold value as we have often employed often this percentage after testing with the whole Berkeley set of images and verified that performs well enough.

4.1 Statistical analysis

This section is aimed to assess the improvements and differences achieved by the methods proposed in this paper. To do so, we use some hypothesis validation techniques in order to give statistical support to the analysis of the results.

Specifically, the Wilcoxon rank test [21] is employed as a non-parametric statistical procedure for making pairwise comparisons between two algorithms. For multiple comparisons, we use the Friedman aligned ranks test, which is recommended in the literature [4, 10] to detect statistical differences among a group of results by considering different levels of confidence for each method. Finally, the Holm post-hoc test [14] is performed to find which algorithms reject the equality hypothesis with respect to a selected control method.

The Wilcoxon test for the Sobel's case, which is showed in Table 3, can be interpreted in different ways. Firstly, it is possible to asses whether the post-aggregation approach outperforms the pre-aggregation approach by comparing methods B, C1 and C2 with respect to A. And effectively they do it (with a low p-value).

Going beyond, we contrasted the hypothesis of equality between the results reached by each fuzzy based method (C1 and C2) and each crispy one (A and B) and rejected it for each possible combination with a p-value of exactly 0.

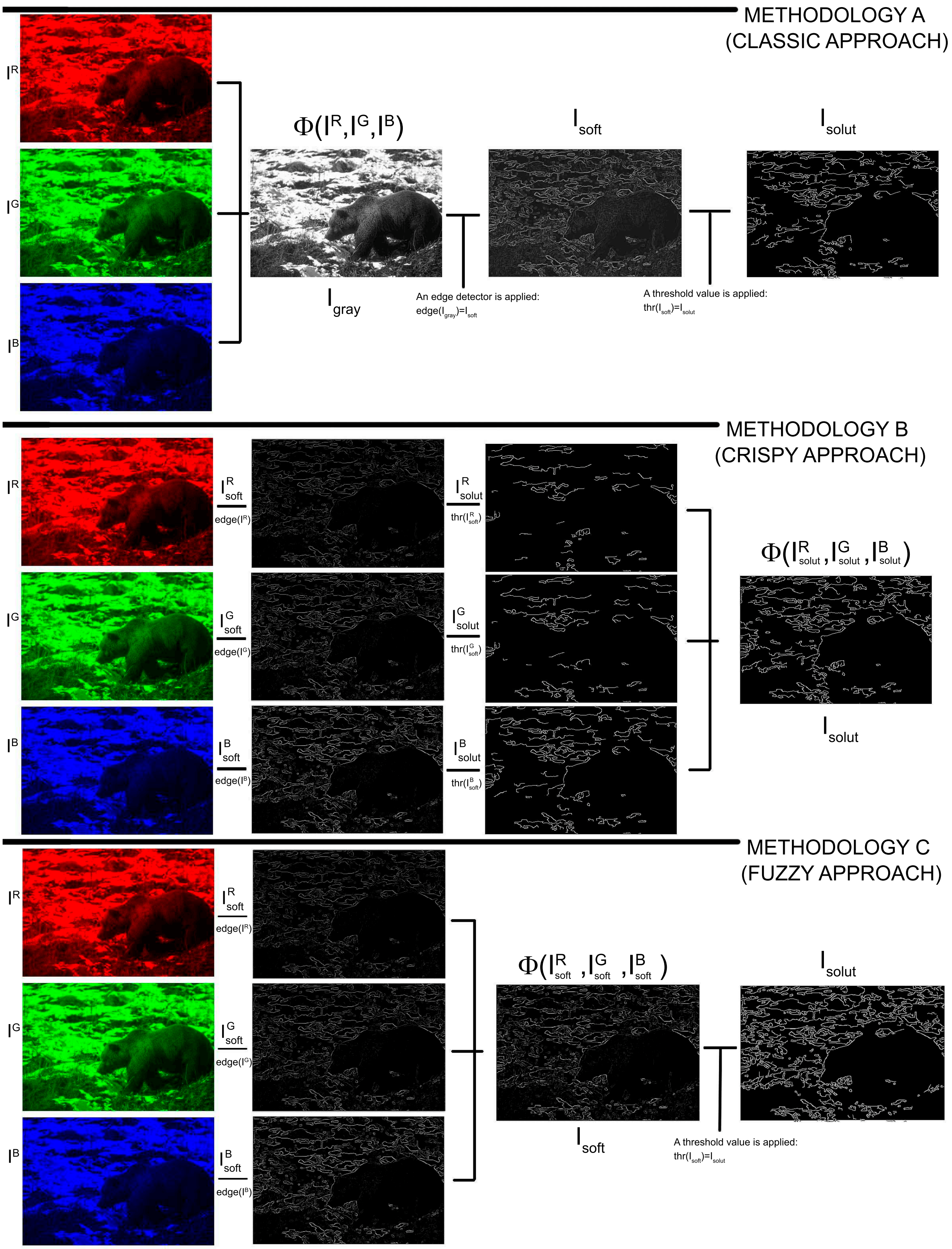

Figure 1: Scheme of methodologies A, B and C.

Image	Sobel Gray (Method A)		Sobel RGB Crispy Maximum Aggregation (Method B)		Sobel RGB Fuzzy Mean Aggregation (Method C1)		Sobel RGB Fuzzy Maximum-Aggregation (Method C2)	
	Fmax	Thr%	Fmax	Thr%	Fmax	Thr%	Fmax	Thr%
100075	.371	27	.407	31	.421	27	.454	29
100080	.304	24	.312	26	.383	22	.382	23
100098	.507	50	.508	54	.693	45	.698	50
103041	.372	25	.375	24	.555	21	.558	24
104022	.535	25	.535	29	.591	22	.595	24
105019	.649	28	.646	31	.656	29	.651	31
105053	.431	21	.430	23	.458	20	.450	23
106020	.194	15	.188	16	.253	22	.243	20
106025	.515	27	.541	27	.647	27	.674	27
108041	.222	29	.214	30	.286	22	.277	22
108073	.303	54	.290	45	.382	24	.363	42
109034	.171	36	.196	63	.217	27	.223	66
112082	.357	43	.344	51	.385	42	.373	47
113009	.632	19	.623	20	.733	18	.722	21
113016	.480	44	.482	55	.512	44	.515	57
113044	.451	50	.483	55	.467	48	.498	57
117054	.372	43	.373	47	.468	41	.469	41
118020	.311	33	.323	37	.433	24	.442	31
118035	.649	53	.709	56	.774	28	.799	58
12003	.350	34	.375	42	.544	28	.554	32
12074	.402	36	.382	38	.574	38	.544	45
122048	.352	41	.341	43	.479	29	.472	29
124084	.518	21	.580	39	.631	18	.665	30
126039	.509	21	.524	23	.595	19	.607	22
130034	.208	44	.207	47	.227	40	.234	42
134008	.206	22	.202	22	.243	23	.239	24
134052	.374	51	.366	55	.449	57	.442	56
135037	.387	38	.379	41	.748	13	.728	19
135069	.910	40	.908	53	.945	32	.943	41
138032	.188	41	.189	48	.317	28	.312	31
138078	.569	45	.603	45	.694	28	.692	29
140055	.227	20	.280	22	.278	21	.331	21
140075	.431	39	.459	46	.535	22	.547	30
144067	.233	41	.238	43	.263	41	.265	42
145014	.357	43	.374	50	.502	31	.515	36
145053	.351	31	.353	31	.412	17	.415	27
147021	.341	42	.343	43	.396	32	.401	37
147062	.385	62	.386	64	.465	59	.466	63
15004	.297	38	.309	40	.376	40	.387	40
15088	.542	55	.545	56	.569	55	.569	56
151087	.608	40	.621	45	.639	42	.648	45
153077	.437	25	.444	26	.503	23	.517	23
153093	.443	20	.457	24	.593	19	.602	24
155060	.493	37	.505	36	.570	38	.594	36
156079	.465	32	.463	42	.505	36	.505	44
157036	.449	46	.447	50	.703	48	.702	55
159029	.205	43	.213	46	.305	42	.315	44
159045	.449	35	.457	35	.514	30	.528	34
159091	.478	42	.483	46	.534	42	.540	25
16052	.208	21	.203	25	.445	20	.431	48
Min	.171	15	.188	16	.217	13	.223	19
Mean	.404	36	.412	40	.497	31	.502	36
Max	.910	62	.908	64	.945	59	.943	66

Table 1: Aggregating color channels using Sobel algorithm

	Canny Gray Channel (Method A)		Canny RGB Crispy Max Aggregation (Method B)		Canny RGB Fuzzy Mean-Aggregation (Method C1)		Canny RGB Fuzzy Maximum-Aggregation (Method C2)	
Image	Fmax	Thr%	Fmax	Thr%	Fmax	Thr%	Fmax	Thr%
100075	.438	42	.469	47	.436	38	.473	52
100080	.389	24	.386	27	.389	24	.391	27
100098	.718	75	.713	80	.713	69	.691	83
103041	.555	30	.545	34	.538	29	.533	34
104022	.630	31	.621	35	.614	30	.610	44
105019	.728	37	.726	40	.715	29	.711	40
105053	.491	29	.498	38	.504	25	.485	41
106020	.270	29	.255	29	.260	23	.252	30
106025	.719	34	.733	43	.699	34	.729	47
108041	.290	30	.283	41	.283	40	.279	41
108073	.389	36	.375	37	.394	35	.370	60
109034	.282	88	.328	90	.260	40	.341	88
112082	.432	52	.388	53	.431	50	.373	53
113009	.733	24	.728	29	.704	23	.711	38
113016	.741	49	.714	76	.711	49	.700	78
113044	.578	61	.635	68	.556	56	.626	70
117054	.506	54	.504	66	.497	65	.501	66
118020	.454	53	.468	60	.446	32	.467	60
118035	.793	48	.800	67	.789	44	.789	67
12003	.597	46	.585	61	.591	43	.572	68
12074	.592	89	.550	89	.586	61	.512	83
122048	.530	74	.518	80	.500	61	.519	81
124084	.621	27	.706	48	.648	24	.674	47
126039	.601	30	.624	37	.594	22	.616	35
130034	.321	65	.337	74	.294	54	.343	74
134008	.296	39	.286	43	.280	33	.285	43
134052	.524	61	.529	68	.499	58	.509	72
135037	.742	47	.722	47	.711	29	.713	47
135069	.934	61	.933	78	.931	53	.916	80
138032	.346	78	.345	79	.354	57	.341	79
138078	.766	50	.785	54	.721	50	.789	60
140055	.309	27	.399	34	.319	26	.389	38
140075	.595	47	.647	54	.590	36	.642	54
144067	.321	80	.328	75	.320	77	.333	75
145014	.501	80	.520	86	.497	36	.523	87
145053	.515	48	.527	61	.531	48	.510	61
147021	.451	42	.433	42	.449	51	.438	44
147062	.527	67	.526	74	.510	63	.508	72
15004	.459	57	.482	65	.438	55	.472	65
15088	.553	68	.537	73	.536	68	.569	78
151087	.736	49	.732	55	.707	43	.745	55
153077	.532	45	.547	52	.533	39	.535	57
153093	.651	44	.668	50	.635	38	.678	50
155060	.634	54	.645	58	.619	55	.643	56
156079	.558	46	.557	59	.549	41	.545	53
157036	.806	71	.814	75	.796	64	.814	75
159029	.383	63	.391	70	.401	66	.384	69
159045	.481	50	.504	52	.488	36	.504	56
159091	.629	66	.648	77	.613	58	.623	78
16052	.456	50	.437	54	.456	40	.426	55
Min	.270	24	.255	27	.260	22	.252	27
Mean	.542	51	.549	58	.533	44	.542	59
Max	.934	89	.933	90	.931	77	.916	88

Table 2: Aggregating color channels using Canny algorithm

Comparison	R^+	R^-	p-val
Post-aggregation vs. pre-aggregation			
Method B vs. Method A	932.0	343.0	**0.0044**
Method C1 vs. Method A	1275.0	0.0	**0**
Method C2 vs. Method A	1275.0	0.0	**0**
Fuzzy vs. Crispy			
Method C2 vs. Method B	1275.0	0.0	**0**
Method C1 vs. Method B	1270.0	5.0	**0**
Between fuzzy aggregations			
Method C2 vs. Method C1	843.0	432.0	**0.0467**

Table 3: Wilcoxon Test to compare the different aggregation methods for the Sobel's algorithm.

Comparison	R^+	R^-	p-val
RGB vs. Grey Scale			
Method B vs. Method A	778.0	497.0	0.173478
Method A vs. Method C1	1061.0	214.0	**0.000043**
Method A vs. Method C2	676.0	599.0	0.706562
Fuzzy vs. Crispy			
Method B vs. Method C1	1033.0	242.0	**0.000132**
Method B vs. Method C2	1021.0	254.0	**0.00021**
Between fuzzy aggregations			
Method C2 vs. Method C1	837.0	438.0	0.053526

Table 4: Wilcoxon Test to compare the different aggregation methods for the Canny's algorithm.

In the Canny's case, which is showed in Table 4, both Method A and B outperformed "fuzzy" methods (C1 and C2). This results are commented in Section 5.

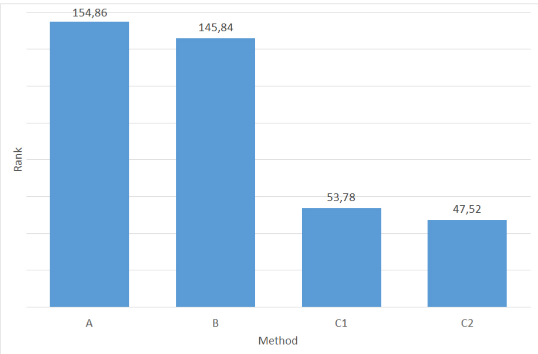

Figure 2: Rankings of the three compared methods employing Sobel's algorithm.

Finally, inside the soft ("fuzzy") frame, there is a significant improvement ($p - value = 0.0467$) between the results reached by the maximum aggregation and those obtained by using the mean operator.

The Friedman aligned rank test (see Figure 2) obtains a low p-value (< 0.0001) when comparing the four methods at the same time, which implies that there are significant differences between the results provided by each method. In the case of Canny (see Figure 3) we find as well differences but this time is Method B best ranked compared with C1's and C2's, and Method A is better compared with C1's.

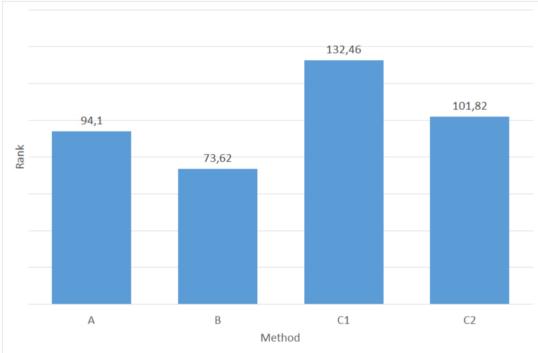

Figure 3: Rankings of the three compared methods employing Canny's algorithm.

Algorithm	Hypothesis	APV
Method A	Rejected for Method C2	**0***
Method B	Rejected for Method C2	**0***
Method C1	Not rejected for Method C2	0.5887

Table 5: Holm test to compare Method C2 against the others.

After the Sobel's results we can apply a Holm test by using the best approach (the one with lower ranking) as control method and computing the adjusted p-value (APV) for the remaining three methods.

Table 5 clearly reflects the statistically significant differences between the control method (Method C2) and each one crispy based methods A or B. This is not enough evidence to reject the hypothesis of equivalence of both fuzzy approaches when considering a multiple comparison scheme.

5 Conclusions

As a remarkable novelty of this paper, we have clarified different methodologies for aggregating the color information of RGB images in the edge detection problem. Moreover, we have found differences in the quality of the comparatives depending of which methodology is applied. In the case of Canny's we could not find improvements for the soft approach as we did with Sobel's. This is due to the fact that the *hysteresis* is affected in a complex way by the aggregation function employed. We believe that this aspect deserves a deeper research, and this will show a connection between global evaluation approach (see [9]) and color edge detection.

Future research could point out the use of different thresholds for each channel. This seems a natural way to continue with this research. This is not been explored here as the number of possibilities and casuistics grows significatively, which leads towards a much longer paper. As well, more research is needed for working with new aggregations, i.e the weighted sum of the color channels. We think that this aggregation function could result in better comparatives against humans due to the slightly different weight of each color in human vision. As well, different aggregation functions could be applied depending of the edge detection algorithm employed. Finally, a more complex line for aggregating colors could be based on the use of other color space different from the RGB's.

Acknowledgement

This research has been partially supported by the Government of Spain, grant TIN2015-66471-P.

For the conducting of this research, the code created by Kermit Research Unit has been strongly helpful [3].

References

[1] S. Bogumil, Color image edge detection and segmentation: A comparison of the vector angle and the euclidean distance color similarity measures, Ph.D. thesis, University of Waterloo (1999).

[2] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis y Machine Intelligence PAMI-8 (6) (1986) 679–698.

[3] B. de Baets, C. López-Molina, The kermit image toolkit (kitt), Ghent university, www.kermitimagetoolkit.net.

[4] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine learning research 7 (Jan) (2006) 1–30.

[5] S. Di Zenzo, A note on the gradient of a multi-image, Computer vision, graphics, and image processing 33 (1) (1986) 116–125.

[6] S. Dutta, A color edge detection algorithm in rgb color space, 2009, pp. 337–340.

[7] P. Flores-Vidal, N. Martínez, D. Gómez, Post-processing in edge detection based on segments, in: Proceedings of the 13th International FLINS Conference on Data Science and Knowledge Engineering for Sensing Decision Support, World Scientific Proceedings Series on Computer Engineering and Information Science (Belfast), World Scientific, 2018, p. 11.

[8] P. A. Flores-Vidal, J. Montero, D. Gómez, G. Villarino, A new edge detection method based on global evaluation using supervised classification algorithms, International Journal of Computational Intelligence Systems 11 (1).

[9] P. A. Flores-Vidal, P. Olaso, D. Gómez, C. Guada, A new edge detection method based on global evaluation using fuzzy clustering, Soft Computing (2018) 1–13.

[10] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, Information Sciences 180 (10) (2010) 2044–2064.

[11] C. Guada, D. Gómez, J. T. Rodríguez, J. Yáñez, J. Montero, Classifying image analysis techniques from their output, International Journal of Computational Intelligence Systems 9 (2016) 43–68, cited By :4.

[12] C. Guada, D. Gómez, J. T. Rodríguez, J. Yáñez, J. Montero, Fuzzy image segmentation based on the hierarchical divide y link clustering algorithm, in: Proceedings - The 2015 10th International Conference on Intelligent Systems y Knowledge Engineering, ISKE 2015, 2016, pp. 12–17.

[13] M. Heath, S. Sarkar, T. Sanocki, K. Bowyer, Comparison of edge detectors: A methodology and initial study, Computer Vision and Image Understanding 69 (1) (1998) 38–54.

[14] S. Holm, A simple sequentially rejective multiple test procedure, Scandinavian journal of statistics (1979) 65–70.

[15] N. P. Jacobson, M. R. Gupta, Design goals and solutions for display of hyperspectral images, IEEE Transactions on Geoscience and Remote Sensing 43 (11) (2005) 2684–2692.

[16] D. Marr, Vision: a Computational Investigation into the Human Representation y Processing of Visual Information, 1982.

[17] D. R. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images y its application to evaluating segmentation algorithms y measuring ecological statistics, in: Proceedings of the IEEE International Conference on Computer Vision, Vol. 2, 2001, pp. 416–423.

[18] M. A. Ruzon, C. Tomasi, Color edge detection with the compass operator, in: Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), Vol. 2, 1999, pp. 160–166 Vol. 2.

[19] I. Sobel, History and definition of the so-called "sobel operator", more appropriately named the sobel-feldman operator.

[20] P. E. Trahanias, A. N. Venetsanopoulos, Color edge detection using vector order statistics, IEEE Transactions on Image Processing 2 (2) (1993) 259–264.

[21] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics bulletin 1 (6) (1945) 80–83.

[22] Y. Yang, Colour edge detection and segmentation using vector analysis, 1995.