International Journal of Computational Intelligence Systems (2022) 15:78
https://doi.org/10.1007/s44196-022-00137-x

RESEARCH ARTICLE

New Aggregation Approaches with HSV to Color Edge Detection

Pablo Flores-Vidal[1] · Daniel Gómez[1] · Javier Castro[1] · Javier Montero[2]

Received: 28 February 2022 / Accepted: 26 August 2022

Abstract
The majority of edge detection algorithms only deal with grayscale images, while their use with color images remains an open problem. This paper explores different approaches to aggregate color information of RGB and HSV images for edge extraction purposes through the usage of the Sobel operator and Canny algorithm. This paper makes use of Berkeley's image data set, and to evaluate the performance of the different aggregations, the *F*-measure is computed. Higher potential of aggregations with HSV channels than with RGB channels is found. This article also shows that depending on the type of image used, RGB or HSV, some methods are more appropriate than others.

Keywords Color edge detection · HSV · Hexcone model · RGB · Pre-aggregation · Post-aggregation

List of Symbols

ED	Edge detection
IP	Image processing
RGB	RGB color space
HSV	HSV color space
I_{gray}	Gray-scale image
I_{soft}	'Soft' image
I_{bin}	Binarized image
I_{RGB}	RGB image
I_{HSV}	HSV image
F_1	F_1 score

Daniel Gómez, Javier Castro and Javier Montero have contributed equally to this work.

✉ Pablo Flores-Vidal
pflores@ucm.es

Daniel Gómez
dagomez@estad.cm.es

Javier Castro
jcastroc@estad.cm.es

Javier Montero
monty@mat.ucm.es

[1] Departamento de Estadística y Ciencia de los datos, Facultad de Estudios Estadísticos, Universidad Complutense de Madrid, Avenida Puerta de Hierro, s/n, 28040 Madrid, Spain

[2] Departamento de Estadística e Investigación Operativa, Facultad de Ciencias Matemáticas, Plaza de las Ciencias, 3, 28040 Madrid, Spain

1 Introduction

Marr [1] suggested that color could play an important role as it 'carries information that often has important biological significance'. This information could help distinguish 'whether a fruit is ripe, whether a leaf is green and supple, whether an insect is likely to be poisonous, and many other things'. Similar to Marr's, McAndrew [2] argued that 'for human beings, colour provides one of the most important descriptors of the world around us'. However, despite the undoubtful importance of color, traditionally, grayscale images have been more widely used when it comes to image processing (IP). This bears particular significance in relation to the edge detection problem, where dealing with color images introduces some complications.

RGBs and HSVs are both among the most important color images types, and are built in the RGB and HSV color space, respectively. The RGB color space is based on human perception. The human eye has three different cone cells, one to capture red luminosity, another to capture green, and the last one to capture blue. Rods, the second kind of cell, processes intensity but not color [3]. As a result, only 'three numerical components are necessary and sufficient to describe a color' as Bogumil has indicated [3]. In this sense, the RGB color space should be ideal for capturing all the color information, as it uses three components for it: red, green and blue. A cube is the common geometric figure to represent RGBs.

On the other hand, HSV was created by graphic designers to mimic the artist process of creating colors [4]. HSV is made up of three channels, similar to RGB images, but color information is instead captured in one single channel: Hue 'H', while the saturation (S) is placed in a different channel. The higher the saturation 'S', the purer the color. A third channel contains the brightness information: the value 'V'. The higher is the value, the whiter the color. HSV color space has recently found new applications as skin detection and face detection [5–7]. The geometric figure that usually represents this color space is a cone (or an hexcone).

Edge detection (ED) is considered one of the main techniques inside Image Processing field (IP) [8–10]. Its importance lies in the fact that most higher-level techniques and algorithms make use of it. In the literature, there is not only one single definition accepted for ED. The most common one is that it is a technique that pursues finding the most important luminosity changes in a digital image [8, 9]. It can be stated that ED mimics the natural process of extracting visual information that is accomplished by human vision. This process has been named *primal sketch* [1]. ED has found applications along with a wide range of tasks and fields as pathological diagnostics in medicine [11], with a special focus in tumoral discovering. As well, in remote sensing [12], which is useful for agriculture and biology, and more recently for research related to climate change. Other relevant fields for ED are: military industry, surveillance [13], and others [14–19].

Some examples of prestigious ED algorithms developed in the literature for working with gray images are Sobel operator [20] and Canny's [21]. The ED algorithms differ from each other mainly in the filters they use. These filters or masks are mathematical computations that perform over the matrix of pixels, i.e., the image, to find significant differences between the pixel values. Other differences between them are based on how they perform along each of the edge detection phases [8, 9, 22], which are commonly divided into conditioning (optional), feature extraction, blending or aggregation of features, and scaling.[1]

Color edge detection is more complex to deal with than grayscale's [23], and due partially to this, there exists commonly a overuse of gray ED, even when the original images are color images that would deserve a more careful treatment to keep its valuable color information. As Koschan and Abidi argue: 'the color edges describe an object geometry in the scene better than the intensity edges, although over 90% of the edges are identical'. Most edge detection algorithms only deal with grayscale images, while a high number of segmentation algorithms deal with color images [10, 24, 25]. One important reason for this is that the distance/dissimilarity between pixels luminosity in one dimension is easier to compute than in a multidimensional case, where the approach for computing the distance/dissimilarity between colors remains an open problem. Therefore, it could be said that the main problem that arises in color ED is how to measure the distance between colors.

In the literature, two main methods for dealing with colors inside the ED problem have been proposed: individual channel [26] and vector-based approaches [27–29]. The first approach seems quite 'natural', and it consists of extracting edges for each channel separately. This approach brings another problem related to the necessity of choosing an appropriate aggregation method for blending the different channels information. And this issue is especially intricate in the case of HSV images due to the different nature of the channels/colors. Another motivation of exploring the possibilities of ED with HSV is that it has been less common in the literature than RGBs.

The proposal of this paper shows that applying different color edge detection algorithms over HSV images making use of aggregation operators inspired by Yager's [30] outperforms RGB based approaches. Moreover, these algorithms are based on 4 different methods for aggregating the RGB and HSV channels.

The rest of this paper is organized as follows: the following Sect. 2 explains some preliminary information including IP basic concepts, color edge detection, HSV images generation from RGB images, relevant aggregation operators concepts and evaluation of ED algorithm's performance using human references. Then, the different approaches for aggregating RGB and HSV channels are presented in Sect. 3. Finally, Sects. 4 and 5 present the comparison, its results, and the conclusions of this research.

2 Preliminaries

This section introduces some important concepts related to IP and aggregation operators.

2.1 Digital Image Processing

Let us denote a digital image by I, and the/its pixel coordinates of the *spatial domain* by (i, j). For clarity's sake, the coordinates are integers, where each point (i, j) represents a pixel with $i = 1, \dots, n$ and $j = 1, \dots, m$. Therefore, the size of an image,

[1] A fifth phase can be as well considered: the thinning (see [22] for more information about this phase).

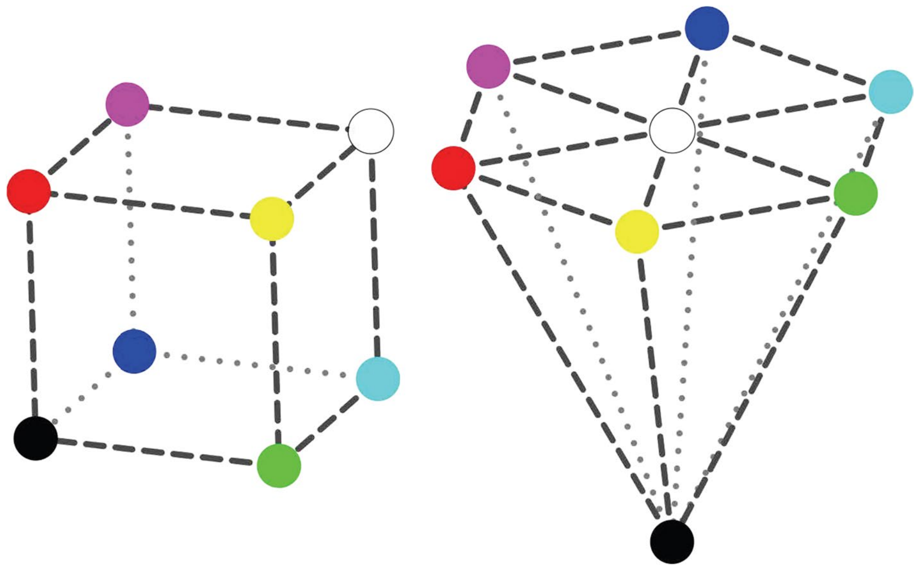
Fig. 1 A visual representation of RGB and HSV (Hexcone model) color spaces. White color is black outlined to contrast it better against the background

$n \times m$, is the number of its horizontal pixels multiplied by its number of vertical pixels. As color images are being dealt with, then a $k = 1, \ldots, \tilde{k}$ index is needed for expressing the number of channels in the image. Let us denote by $I^k_{i,j}$ the spectral information associated with each pixel (i, j) at channel k. The value range of this information is dependent on the digital image type that is being considered:

- *binary map* (I_{bin}) where $I_{i,j} \in \{0, 255\}$ (as well it is usually expressed as $\{0, 1\}$).
- *grayscale image* (I_{gray}), where $I_{i,j} \in \{0, 1, \ldots, 255\}$.
- *soft image* (I_{soft}) where $I_{i,j} \in [0, 1]$. As well it is referred as a normalized grayscale image.
- *RGB image* (I_{RGB}) where $I_{i,j} \in \{0, 1, \ldots, 255\}^3$ (R = *Red*; G = *Green* and B = *Blue*). In this paper, the channels are referred as both, $I^{\text{R}}_{\text{RGB}}, I^{\text{G}}_{\text{RGB}}, I^{\text{B}}_{\text{RGB}}$ or as the more simplified version $I_{\text{R}}, I_{\text{G}}, I_{\text{B}}$.
- *HSV image* (I_{HSV}) composed by three channels ($I^{\text{H}}_{\text{HSV}}, I^{\text{S}}_{\text{HSV}}, I^{\text{V}}_{\text{HSV}}$) being H=*Hue*; S=*Saturation* and V=*Value*), $I^{\text{S}}_{i,j}, I^{\text{V}}_{i,j} \in [0, 1]$ and for $I^{\text{H}}_{i,j}$ there are two possible definitions: That $I^H_{i,j} \in [0^o, 360^o]$ or $I^H_{i,j} \in [0, 1]$ (the one used in this paper). The first one can easily be obtained through multiplying the second one by 360 and changing the scale to degrees. For ease of reference, in this paper the HSV channels are also referred to as $I_{\text{H}}, I_{\text{S}}, I_{\text{V}}$. Moreover, for the use inside a formula by means of numeric index a third kind of expressions are employed: $I^1_{\text{HSV}}, I^2_{\text{HSV}}, I^3_{\text{HSV}}$ where 1 stands for Hue, 2 stands for Saturation and 3 for Value. See Sect. 2.2 for more information about HSV images.

2.2 HSV and Other Color Models

HSV was created for mimicking the process of a painter that starts by choosing a hue, and then adds some white to it to give more light, or some black to darken it. Hue is any pure color that can be represented as a point placed in a disk (or in an hexagon). This color ranges according to its saturation from the purest color, i.e., the maximum saturation that is placed over the disk or hexagon, to white or gray, i.e., the minimum saturation that is situated in the center of the circle or hexagon. This is the reason why hue is expressed in degrees. 'Value' is the third dimension and represents the grade in which this color is non-black (0 = 'black' and 1 = 'white'). Thus, a regular cone or a similar figure is usually employed to represent this color model (see Fig. 1). The less black, the higher the value is.

The literature has proposed color spaces different from RGB and HSV to study the dissimilarities between colors, as in the case of YUV and its variant YCoCg [31], CIELab [32], CYMK among others.

Another color model similar to HSV is HSL. There is a degree of confusion between these two models, and it is important to make a distinction [4]. In HSL, 'L' stands for 'lightness', which is equivalent to whiteness, while the value 'V' in HSV refers to the purity or 'non-blackness' of a color. A practical distinction is that all the pure tones or hues are the same in HSV, and placed in a plane, while in HSL every hue refers to a color that has a different lightness.

The algorithm named 'Hexcone model' [4] allows transforming an RGBs into an HSV image, and its steps are specified in what follows.

1. A normalized RGB image $I_{\text{RGB}} \in [0, 1]^3$ is given;
2. $I_{\text{V}} = \max(I_{\text{R}}, I_{\text{G}}, I_{\text{B}})$;
3. Let $X = \min(I_{\text{R}}, I_{\text{G}}, I_{\text{B}})$;

4. $I_S = \frac{I_V - X}{I_V}$ if $I_V \neq 0$; (else $I_S = 0$ and then the color is pure black);
5. Let $r = \frac{I_V - I_R}{I_V - X}$, $g = \frac{I_V - I_G}{I_V - X}$ and $b = \frac{I_V - I_B}{I_V - X}$;
6. (a) If $I_R = I_V$ then $I_H = 5 + b$ if $I_G = X$ and else $I_H = 1 - g$;
(b) else if $I_G = I_V$ then $I_H = 1 + r$ if $I_B = X$ and else $I_H = 3 - b$;
(c) else $I_H = 3 + g$ if $I_R = X$ and else $I_H = 5 - r$;
7. $I_H = I_H / 6$.

After this algorithm is applied, an HSV image is obtained, with $I_H \in [0, 1]$, $I_S \in [0, 1]$ and $I_V \in [0, 1]$.

This paper is concerned with two different color spaces: RGB and HSV, whose differences can be appreciated in Fig. 1. Previous works have employed a different one: *Super8* [26, 29], and in the previous and shorter version of this research [23], only RGB was employed. In this paper, RGB and HSV are employed by means of individual-channel approach (multi-channel was employed in [29]).

2.3 Aggregation Operators

Aggregation operators (AO) are one of the most important disciplines in information sciences since they are a fundamental part of knowledge acquisition. The process of aggregating the information is a key tool for most knowledge-based systems.

Definition 1 A function $A : [0, 1]^n \rightarrow [0, 1]$ is said to be an n-ary aggregation function if the following conditions hold:

(A1) A is increasing in each argument: for each $i \in \{1, \dots, n\}$, if $x_i \leq y$, then $A(x_1, \dots, x_i, \dots, x_n) \leq A(x_1, \dots, x_{i-1}, y, x_{i+1}, \dots, x_n)$;
(A2) A satisfies the boundary conditions: $A(0, \dots, 0) = 0$ and $A(1, \dots, 1) = 1$.

AO have been employed in different disciplines due to their large number of applications, IP being one of them [33–35].

As this paper deals with information provided by different channels, which belong to RGB or HSV images, the use of AO is justified by the fact that the value of each channel is related to the likeliness of a given pixel to be an edge, or what the literature has called *edginess*. Then, there is a natural connection between the boundary conditions of AO and the potential edginess of a certain pixel. In this sense, the complete lack of edginess can be associated to the concept of minimal boundary. Conversely, the pixel is an edge when the supreme boundary is reached. Another desired propriety is monotonicity, as for a certain pixel an increment in the value of any channel means a higher likeliness for the pixel to be an edge.

The classical definition of AO can be naturally extended by means of replacing the unit interval [0, 1] with a more general lattice, which in the fuzzy area is traditionally assumed to be a complete lattice [36].

Some of these operators allow giving prioritization to some type of data against others. This can be done, for example, dealing with prioritized information, as it happens with the prioritized aggregation operators proposed by Yager in [30], which were generalized by Rojas et al. [37] In the latter, the generalization consisted in the use of general weights acting over the hierarchy and internal aggregation operators that can differ from the minimum, which was employed by Yager [30].

Definition 2 The Yager-inspired hierarchical prioritized aggregation operator is defined as:

$$V(y_1, \dots, y_n) = \sum_{i=1}^{n} w_i \prod_{k=1}^{i} y_{n-k+1}, \qquad (1)$$

where 'V' stands for vertical, as every hierarchized set of data is placed in a different box inside a vertical structure that shows the hierarchy between the clusters 'y_i', and 'n' is the number of different clusters (for further information see [37]).

The *prioritization* of clusters (or 'categories' in Yager's words) enables not only the assignment of a different importance for each one (this could also be done employing weighted operators) but it also enables the use of 'a kind of importance weight in which the importance of a lower priority criteria will be based on its satisfaction to the higher priority criteria' [38].

For this paper, and within the HSV context, the y_1 = 'value' channel is in the top box, followed by the y_2 = 'saturation' channel in the middle, with the 'least important' the y_3 = 'Hue' channel at the bottom. The assumption that the 'value' channel carries more information for edge extraction purposes, i.e, it is more important than the other two, has been made. In fact, using only this channel allows an edge extraction that works perfectly (in practice, this happens because this channel is equal to the maximum color intensity of the RGB image's version as can be seen in Hexcone model algorithm). Another reason for using this prioritization is that first comes the lightness (the 'value' channel), after comes the saturation of the color (the color purity), and finally the hue, what theoretically would be the least important of the three channels for ED purposes. Finally, to satisfy the conditions of Definition 1, $\sum_{i=1}^{n} w_i = 1$ and $w_i \in [0, 1]$.

2.4 Color Edge Detection

From a mathematical point of view, an edge detection algorithm is a function that converts a digital image into a binary image. The literature on the topic offers two main approaches to color edge detection:

1. *Individual channel*: The edges are extracted for each channel. This is the approach this paper employs.
2. *Vector-based approach*: An aggregation function is applied: for example, a median filter [27], a range operator [28], or other statistical aggregation methods [39]. This approach was employed in [29].

Other alternative approaches for working with color edge detection have been proposed. For example, in [40, 41] where this problem is solved by means of working with gradients that result from combining different colors.

2.5 Performance in Edge Detection Problems

Evaluating an edge detection algorithm cannot be considered a trivial task. For managing this task there exist many different approaches. In this paper, is followed the boundary-based evaluation methodology developed in [42, 43]. The methodology for benchmarking boundary detection algorithms developed by [43] is used on the Berkeley Segmentation Dataset (BSDS500). Nevertheless, this dataset of images was not created specifically for edge detection, but it has been widely used for edge detection comparisons [18, 22]. This dataset consists of 500 natural images that are divided into a training set of 200 images, a test set of 200 images and a validation set of 100 images. Each image of BSDS is accompanied by a set of four to seven human-made reference boundary maps (an example of this 'Humans ground-truth' is shown in Fig. 4) that serve as ground-truth for evaluating the automatic boundary maps that constitute the output of an edge detection technique [42].

Given an image I and to compare an edge detection solution I_{bin} (a binary image) for this image with the result given by a human ground-truth, a matching algorithm is developed to find the true positive values needed to build the confusion matrix. In this matching algorithm a distance threshold δ is defined to specify the tolerance level to small boundary localization errors. Then, an unmatched automatic boundary pixel that lies closer than a distance δ from a human boundary pixel is counted as a true positive (TP). Otherwise, unmatched automatic boundary pixels are counted as false positives (FP). And unmatched human boundary pixels are counted as false negatives (FN). Once these values are obtained, the confusion matrix can be built as well as other accuracy measures as the *precision* (Prec), *recall* (Rec) and also the F_β-measure. These constitute the most accepted alternative in recent years [18, 42, 44] to evaluate the performance of each one-to-one comparison.

Formally, given a candidate automatic boundary map I_{bin} and a ground-truth human boundary map I_{gt}, its comparison's F_β is computed as follows:

$$F_\beta(I_{\text{bin}}, I_{\text{gt}}) = (1+\beta^2)\frac{\text{Prec}(I_{\text{bin}}, I_{\text{gt}}) \cdot \text{Rec}(I_{\text{bin}}, I_{\text{gt}})}{\beta^2 \text{Prec}(I_{\text{bin}}, I_{\text{gt}}) + \text{Rec}(I_{\text{bin}}, I_{\text{gt}})}, \tag{2}$$

where an harmonic mean is obtained for $\beta = 1$ and

$$\text{Prec}(I_{\text{bin}}, I_{\text{gt}}) = \frac{\text{TP}}{\text{TP}+\text{FP}}, \tag{3}$$

$$\text{Rec}(I_{\text{bin}}, I_{\text{gt}}) = \frac{\text{TP}}{\text{TP}+\text{FN}}. \tag{4}$$

3 Aggregating Channels in Color Edge Detection

This paper proposes 4 different methods for aggregating the RGB and HSV channels. The Sobel operator [20, 45] and Canny algorithm [21] are both used for these methods specified below. Moreover, all these methods make use of the individual channel approach[2].

1. *Crisp pre-aggregation (method A)*: This method aggregates the different components/channels into one single channel. The exact procedure depends on the color space that is being used. In the RGB case, this means that the three channels/colors are aggregated into one single gray channel employing the mean. Then, an edge detection algorithm is applied over the resulting grayscale image. Canny's and Sobel's are applied to this paper. This method can be regarded as the classic method, as it is the most common procedure employed in the literature.

$$I_{\text{gray}} = \frac{I_R + I_G + I_B}{3}. \tag{5}$$

Equation 5 can be easily extended to more channels simply by considering the mean or weighted mean, which has already been applied, for example, in [26, 39]. This approach allows assigning variable importance to each color/channel.

When aggregating the three channels of an HSV image the specific nature of each channel has to be taken into

[2] Multichannel approach was employed in [29].

account. To deal with the information provided by I^1_{HSV}, I^2_{HSV} and I^3_{HSV}, corresponding to hue, saturation and value respectively, this paper proposes what it has been termed *Yager aggregation* of channels, which is inspired by [30], employed in [37] and explained in Sect. 2.3. Three different weights: w_1, w_2, w_3 are applied hierarchically as shown below:

$$I_{\mathrm{gray}} = \sum_{i=1}^{3} w_i \prod_{k=1}^{i} I^{4-k}_{\mathrm{HSV}} \tag{6}$$

expression that can also be expressed as:

$$I_{\mathrm{gray}} = w_1 \cdot I_{\mathrm{V}} + w_2 \cdot I_{\mathrm{V}} \cdot I_{\mathrm{S}} + w_3 \cdot I_{\mathrm{V}} \cdot I_{\mathrm{S}} \cdot I_{\mathrm{H}}. \tag{7}$$

Once the HSV image has been aggregated into a single grayscale channel, which can be expressed as Yager$(I_{\mathrm{HSV}}, w_1, w_2, w_3)$, an edge detection operator is applied following standard procedures: $I_{\mathrm{soft}} = edge(I_{\mathrm{gray}})$ as a general formula, and in the case of this paper being $I_{\mathrm{soft}} = \mathrm{Sobel}(I_{\mathrm{gray}})$ or $I_{\mathrm{soft}} = \mathrm{Canny}(I_{\mathrm{gray}})$ as these two edge detection algorithms (operator in the case of Sobel) are the ones employed.

As the final ED step, the binarized image I_{bin} is produced. This image results from the soft image after a threshold value is applied (for a detailed explanation of the different phases of edge detection task see [22, 46]): $I_{\mathrm{bin}} = \mathrm{threshold}(I_{\mathrm{soft}}, \alpha)$, where this means that an α threshold is applied over the soft image. Normally, this α value ranges from 0 to 100% or [0, 1]. The w_i weights are chosen to satisfy the conditions of Definition 1.

Two schemes for this method, one for RGB and the other for HSV, can be seen at the top of Figs. 2 and 3.

2. *Crisp post-aggregation (method B)*: Applying an edge detection operator over each channel separately, which produces $\tilde{K}$ different edges maps ($\tilde{K} = 3$ in the case of RGB and HSV). Then, all $\tilde{K}$ resulted binarized images will be aggregated into a single one. We can see this methodology in Algorithm 1:

   In the case of HSV images, the three channels differ in their nature, as it was pointed out in Sect. 2. Therefore, to adapt the I^k_{soft} images, these three images resulted as follows:

   (a) $I^{\mathrm{YagerV}}_{\mathrm{soft}} = \mathrm{Sobel}(w_1 \cdot I_{\mathrm{V}})$;
   (b) $I^{\mathrm{YagerS}}_{\mathrm{soft}} = \mathrm{Sobel}(w_2 \cdot I_{\mathrm{V}} \cdot I_{\mathrm{S}})$;
   (c) $I^{\mathrm{YagerH}}_{\mathrm{soft}} = \mathrm{Sobel}(w_3 \cdot I_{\mathrm{V}} \cdot I_{\mathrm{H}} \cdot I_{\mathrm{S}})$.

   After this, the binarized images of each channel are created through applying a threshold (see 4th line of Algorithm 1). Then, they are aggregated (5th line of Algorithm 1). The maximum aggregation function $\Theta() = \max()$ has been employed:

   $I_{\mathrm{bin}} = \max(I^1_{\mathrm{bin}}, \dots, I^{\tilde{k}}_{\mathrm{bin}})$

   In the case of RGBs, this method B with the max() aggregation function has been employed by [2].

3. *Fuzzy post-aggregation (method C)*: In this case the aggregation function is not applied over the already binarized image, but it is instead applied in the previous step over the soft image corresponding to each color channel. This soft image is made of what are termed as *candidates to be edge* pixels (see [22]). At the last step of the Algorithm 2, the binarized image is produced, following a soft approach that we have called 'fuzzy' approach.

   One more algorithm, the Algorithm 3, has been developed going beyond what was done in [23]. In this new approach, which has been named Method C2, the aggregation function is applied before combining the edge information, aka as edginess, of different directions/features, which are two directions, horizontal and vertical, in the case of Sobel's. On the other side, when the aggregation of channels is made after the blending of directional information, this has been named Method C1.

Algorithm 1 Crisp post-aggregation (method B)

```
1: procedure (Crisp post-aggregation of $\tilde{k}$ channels)
2:     for $k = 1, \dots, \tilde{k}$ do
3:         $I^k_{soft} = edge(I^k_{RGB}$ or $I^k_{HSV})$
4:         $I^k_{bin} = threshold(I^k_{soft}, \alpha)$
5:     end for
6:     $I_{bin} = \Theta(I^1_{bin}, \dots, I^{\tilde{k}}_{bin})$
7: end procedure
```

Algorithm 2 'Fuzzy' post-aggregation (method C1)

```
1: procedure ('Fuzzy' aggregation of channels)
2:     for $k = 1, \dots, \tilde{k}$ do
3:         $I^k_{soft} = edge(I^k_{RGB}$ or $I^k_{HSV})$
4:     end for
5:     $I_{soft} = \Theta(I^1_{soft}, \dots, I^{\tilde{k}}_{soft})$
6:     $I_{bin} = threshold(I_{soft}, \alpha)$
7: end procedure
```

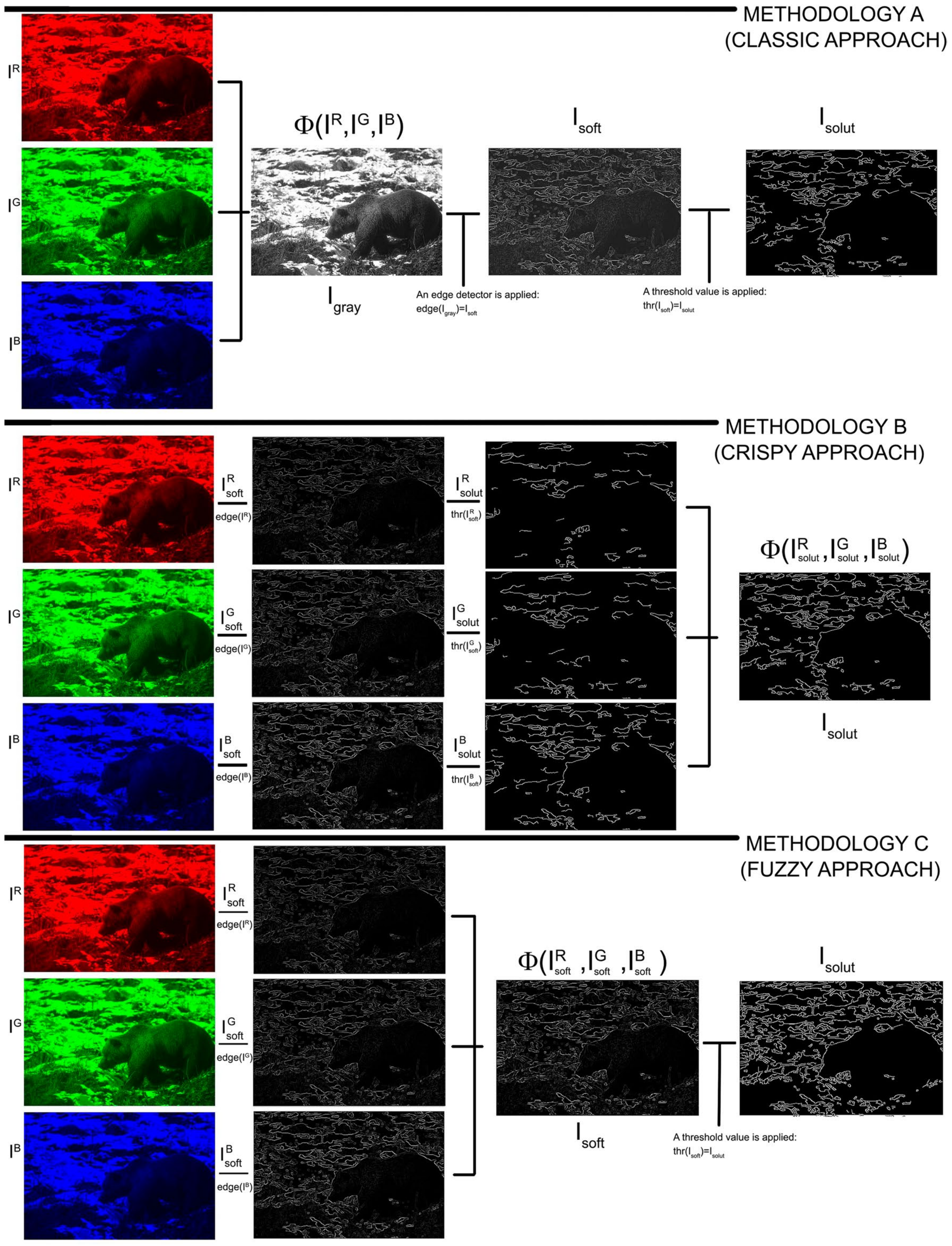
METHODOLOGY A
(CLASSIC APPROACH)
I^R
I^G
I^B
$\Phi(I^R,I^G,I^B)$
I_{gray}
An edge detector is applied: $edge(I_{gray})=I_{soft}$
I_{soft}
A threshold value is applied: $thr(I_{soft})=I_{solut}$
I_{solut}
METHODOLOGY B
(CRISPY APPROACH)
I^R_{soft} $edge(I^R)$
I^G_{soft} $edge(I^G)$
I^B_{soft} $edge(I^B)$
I^R_{solut} $thr(I^R_{soft})$
I^G_{solut} $thr(I^G_{soft})$
I^B_{solut} $thr(I^B_{soft})$
$\Phi(I^R_{solut},I^G_{solut},I^B_{solut})$
I_{solut}
METHODOLOGY C
(FUZZY APPROACH)
$\Phi(I^R_{soft},I^G_{soft},I^B_{soft})$
I_{soft}
A threshold value is applied: $thr(I_{soft})=I_{solut}$
I_{solut}

Fig. 2 Scheme of methods A, B and C with RGB images

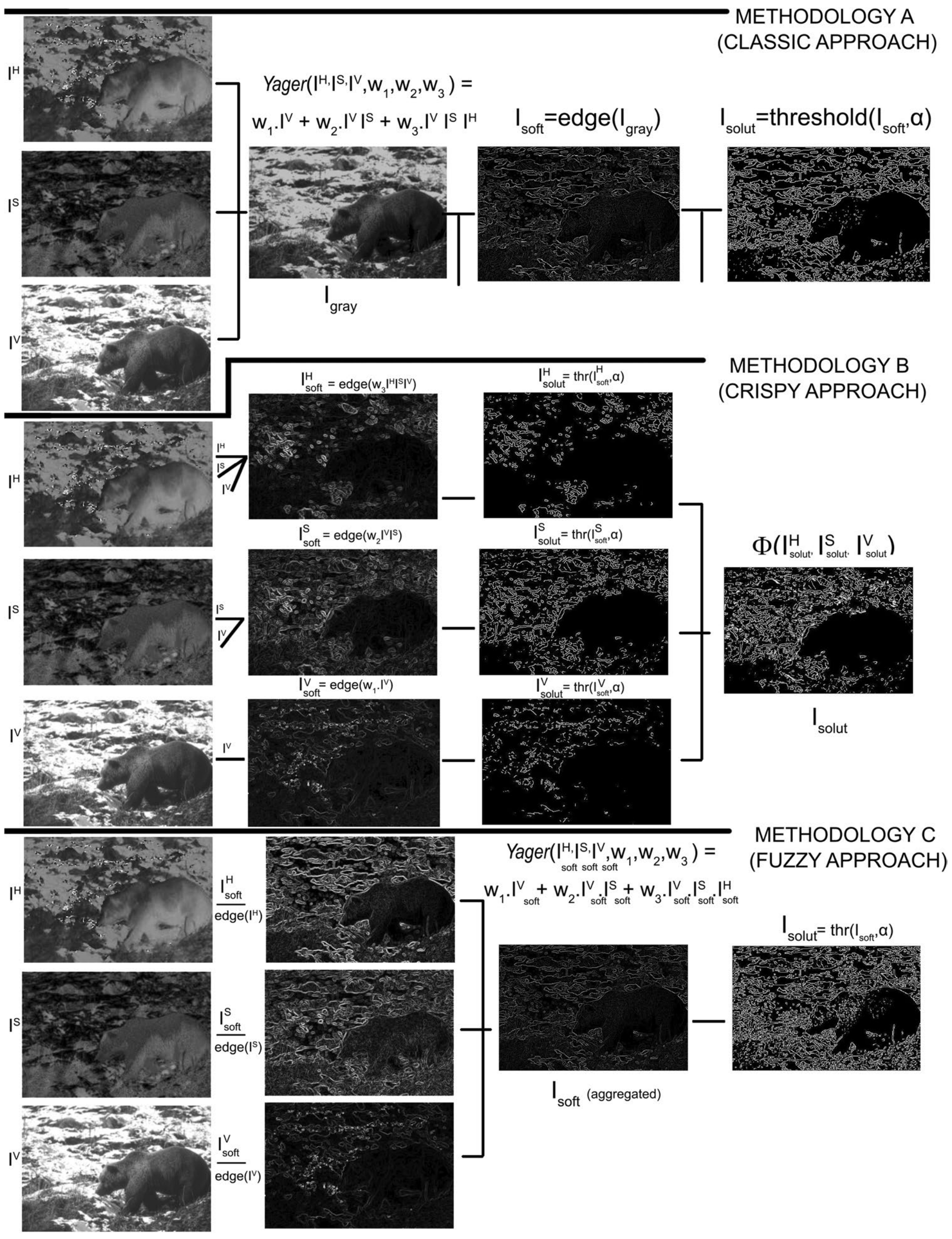
METHODOLOGY A
(CLASSIC APPROACH)
I^H
I^S
I^V
$Yager(I^H, I^S, I^V, w_1, w_2, w_3) = w_1.I^V + w_2.I^V I^S + w_3.I^V I^S I^H$
I_{soft}=edge(I_{gray})
I_{solut}=threshold(I_{soft},α)
I_{gray}
METHODOLOGY B
(CRISPY APPROACH)
I^H_{soft} = edge($w_3 I^H I^S I^V$)
I^H_{solut}= thr(I^H_{soft},α)
I^S_{soft} = edge($w_2 I^V I^S$)
I^S_{solut}= thr(I^S_{soft},α)
I^V_{soft} = edge($w_1.I^V$)
I^V_{solut}= thr(I^V_{soft},α)
$\Phi(I^H_{solut}, I^S_{solut}, I^V_{solut})$
I_{solut}
METHODOLOGY C
(FUZZY APPROACH)
$Yager(I^H_{soft}, I^S_{soft}, I^V_{soft}, w_1, w_2, w_3) = w_1.I^V_{soft} + w_2.I^V_{soft}.I^S_{soft} + w_3.I^V_{soft}.I^S_{soft}.I^H_{soft}$
I^H_{soft} edge(I^H)
I^S_{soft} edge(I^S)
I^V_{soft} edge(I^V)
I_{solut}= thr(I_{soft},α)
I_{soft} (aggregated)

Fig. 3 Scheme of methods A, B and C with HSV images

Table 1 Sobel evaluated measures for HSV and RGB algorithms (average of 10 cross-validation sets)—closest human

Algorithms (Method)	Color	Precision	Recall	F_1	Std. dev	Best threshold and w_i
Sobel RGBMean (A)	RGB	0.45	0.66	0.535	0.023	–
Sobel RGB 'Crisp' Max (B)	RGB	0.47	0.66	0.547	0.025	–
Sobel RGB 'Fuzzy' Max (C)	RGB	0.43	0.63	0.510	0.030	–
Sobel HSV Yager (A)	HSV	0.47	0.67	**0.555**	0.018	0.30; $\{0.3, 0.6, 0.1\}$
Sobel HSV Yager Max (B)	HSV	0.43	0.73	0.546	0.027	0.30; $\{0.1, 0.9, 0.0\}$
Sobel HSV 'Fuzzy' Yager (C1)	HSV	0.47	0.67	0.554	0.021	0.15; $\{0.4, 0.6, 0.0\}$
Sobel HSV 'Fuzzy' Yager (C2)	HSV	0.50	0.63	**0.555**	0.021	0.10; $\{0.3, 0.7, 0.0\}$

Bold value indicate method reached the highest value

Table 2 Sobel evaluated measures for HSV and RGB algorithms (average of 10 cross-validation sets)—average human

Algorithms (method)	Color	Precision	Recall	F_1	Std. dev	Best threshold and w_i
Sobel RGBMean (A)	RGB	0.34	0.57	0.430	0.022	–
Sobel RGB 'Crisp' Max (B)	RGB	0.36	0.56	0.437	0.019	–
Sobel RGB 'Fuzzy' Max (C)	RGB	0.31	0.55	0.397	0.024	–
Sobel HSV Yager (A)	HSV	0.40	0.56	**0.464**	0.011	0.30; $\{0.3, 0.6, 0.1\}$
Sobel HSV Yager Max (B)	HSV	0.32	0.68	0.436	0.031	0.30; $\{0.1, 0.9, 0.0\}$
Sobel HSV 'Fuzzy' Yager (C1)	HSV	0.36	0.57	0.441	0.018	0.10; $\{0.4, 0.6, 0.0\}$
Sobel HSV 'Fuzzy' Yager (C2)	HSV	0.38	0.55	0.447	0.017	0.25; $\{0.5, 0.2, 0.3\}$

Bold value indicate method reached the highest value

Table 3 Canny evaluated measures for HSV and RGB algorithms (average of 10 cross-validation sets)—closest human

Algorithms (method)	Color	Precision	Recall	F_1	Std. dev	Best threshold and w_i
Canny RGBMean (A)	RGB	0.46	0.68	0.550	0.019	–
Canny RGB 'Crisp' Max (B)	RGB	0.47	0.70	0.559	0.039	–
Canny RGB 'Fuzzy' Mean (C)	RGB	0.46	0.67	0.548	0.029	–
Canny RGB 'Fuzzy' Max (C)	RGB	0.43	0.75	0.545	0.016	–
Canny HSV Yager (A)	HSV	0.45	0.73	0.559	0.025	0.50; $\{0.7, 0.3, 0.0\}$
Canny HSV Yager Max (B)	HSV	0.48	0.68	0.566	0.021	0.50; $\{1.0, 0.0, 0.0\}$
Canny HSV 'Fuzzy' Yager (C1)	HSV	0.50	0.66	**0.570**	0.026	0.20; $\{0.3, 0.7, 0.0\}$
Canny HSV 'Fuzzy' Yager (C2)	HSV	0.48	0.69	0.567	0.017	0.10; $\{0.1, 0.5, 0.4\}$

Bold value indicate method reached the highest value

Table 4 Canny evaluated measures for HSV and RGB algorithms (average of 10 cross-validation sets)—average human

Algorithms (method)	Color	Precision	Recall	F_1	Std. dev	Best threshold and w_i
Canny RGBMean (A)	RGB	0.34	0.61	0.440	0.014	–
Canny RGB 'Crisp' Max (B)	RGB	0.37	0.58	0.450	0.022	–
Canny RGB 'Fuzzy' Mean (C)	RGB	0.35	0.58	0.440	0.022	–
Canny RGB 'Fuzzy' Max (C)	RGB	0.32	0.67	0.432	0.017	–
Canny HSV Yager (A)	HSV	0.38	0.57	0.453	0.014	0.45; $\{0.4, 0.6, 0.0\}$
Canny HSV Yager Max (B)	HSV	0.37	0.58	0.453	0.020	0.50; $\{1.0, 0.0, 0.0\}$
Canny HSV 'Fuzzy' Yager (C1)	HSV	0.37	0.59	**0.456**	0.020	0.20; $\{0.3, 0.7, 0.0\}$
Canny HSV 'Fuzzy' Yager (C2)	HSV	0.37	0.58	0.453	0.015	0.15; $\{0.0, 0.9, 0.1\}$

Bold value indicate method reached the highest value

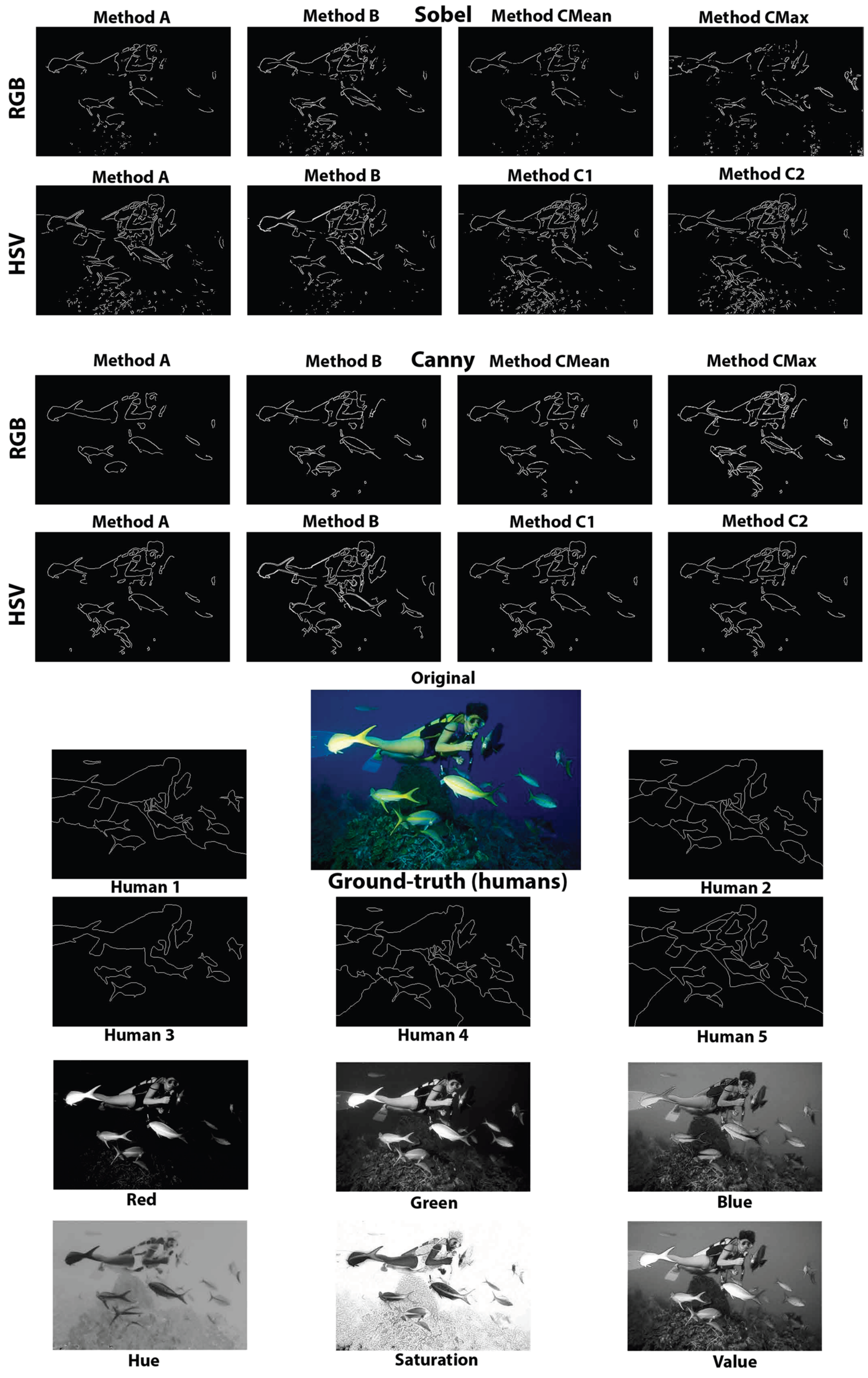

Fig. 4 Best outputs for all methods for an example's image

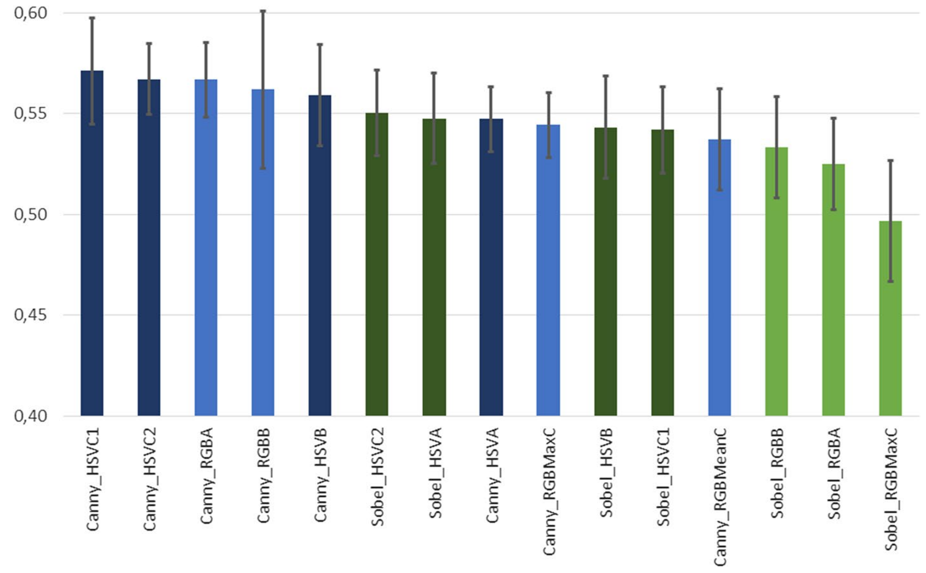

Fig. 5 F_1 medians and standard errors for all methods (closest human)

Table 5 Wilcoxon signed rank test with continuity correction between the aggregation methods (**Sobel**)

Methods that are being compared		Differences	*P* value	Test direction
RGB A vs	**RGB B**	Yes	0.0273	RGB B > RGB A
RGB A vs	RGB C Max	Yes	0.0020	RGB A > RGB C Max
RGB B vs	RGB C Max	Yes	0.0020	RGB B > RGB C Max
HSV Yager A vs	HSV Yager B	Yes	0.0020	HSV Yager A > HSV Yager B
HSV Yager A vs	HSV Yager C1 Max	Yes	0.0020	HSV Yager A > HSV Yager C1 Max
HSV Yager A vs	HSV Yager C2 Max	Yes	0.0039	HSV Yager A > HSV Yager C2 Max
HSV Yager B vs	HSV Yager C1 Max	No	–	–
RGB B (best RGB) vs	**HSV Yager A** (best HSV)	Yes	0.0020	HSV Yager A > RGB B

Bold value indicate method reached the highest value

Table 6 Wilcoxon signed rank test with continuity correction between the aggregation methods (**Canny**)

Methods that are being compared		Differences	*P* value	Test direction
RGB A vs	RGB B	No	0.0644	–
RGB B vs	RGB C Max	Yes	0.0273	RGB B > RGB C Max
RGB B vs	RGB C Mean	Yes	0.0136	RGB B > RGB C Mean
RGB C Max vs	RGB C Mean	No	0.8457	–
HSV Yager A vs	HSV Yager B	No	0.2753	–
HSV Yager A vs	HSV Yager C1	No	0.2324	–
HSV Yager B vs	HSV Yager C1	No	0.1601	–
RGB B (best RGB) vs	**HSV Yager C1** (best HSV)	Yes	0.0019	HSV Yager C1 > RGB B

Bold value indicate method reached the highest value

Algorithm 3 'Fuzzy' post-aggregation (method C2)

```
1:  procedure ('Fuzzy' aggregation of channels directions)
2:      for $k = 1, \ldots, \tilde{k}$ do
3:          $I^k_{soft_x} = edge_x(I^k_{RGB}$ or $I^k_{HSV})$
4:          $I^k_{soft_y} = edge_y(I^k_{RGB}$ or $I^k_{HSV})$
5:      end for
6:      $I_{soft_x} = \Theta(I^1_{soft_x}, \ldots, I^{\tilde{k}}_{soft_x})$
7:      $I_{soft_y} = \Theta(I^1_{soft_y}, \ldots, I^{\tilde{k}}_{soft_y})$
8:      $I_{soft} = sqrt(I^2_{soft_x} + I^2_{soft_y})$
9:      $I_{bin} = threshold(I_{soft}, \alpha)$
10: end procedure
```

4 Comparisons

To test edge extraction quality for the different aggregation methods specified in the previous section, Berkeley's images data set (BSDS500) [43] was used. More specifically, the first 50 images sorted by number were employed (from '100075.jpg' to '16052.jpg'). This sample was divided into training and test, which allowed learning the best parameters. Following a tenfold cross-validation method, the 50 images were divided into 5 blocks of 10 images each, which resulted in 10 different combinations of training sets of 30 images and validation sets of 20 images. Following this learning approach meant another improvement compared to what was done in previous research [23].

The comparisons were conducted for both versions of the image, RGB and HSV. The RGB version was the original included in Berkeley's dataset, while the HSV version was computed by means of the rgb2hsv(I_{RGB}) function of *Matlab2020b* program. This function transforms an RGB image into HSVs applying the algorithm known as the hexcone model [4].

Moreover, 10 different w_i values were allowed for each HSV channel (0.1 by 0.1), but this was made respecting the boundary condition of Definition 1. Therefore, in resume, in this paper were compared 2 different color spaces (RGB and HSV) with 4 methods and 3 smoothing values ($\sigma_{\mathrm{smooth}} = \{0, 0.6, 1.0\}$), with 66 different weights (the number of w_i combinations that result when steps of 0.1 by 0.1 are being considered), 2 different edge detection algorithms (with 3 different Gaussian values in the case of Canny's: $\sigma_{\mathrm{Canny}} = \{1, 1.5, 2.0\}$), and all this was made employing 19 different superior threshold values for hysteresis process [21] (ranging from 0.05 to 0.95 and stepping 0.05 by 0.05). Lower threshold was set up as $\mathrm{Thr}_{\mathrm{low}} = 0.4\mathrm{Thr}_{\mathrm{sup}}$ as this relationship between both thresholds had been previously discovered in previous researches. All these combinations made a total of almost 3 million and a half binarized images to be evaluated. An ACER Intel(R) Core(TM) i7-8550U CPU with 1.80 GHz and 2.00 GHz with 20 GB RAM was employed for the computational analysis, supported occasionally by an OMEN HP Intel(R) Core(TM) i7-7700HQ CPU with 2.80 GHz and 2.81 GHz with 16 GB RAM.

After the matching process between the outputs of each algorithm with human's ground-truth was performed, Precision, Recall and two different F_1 were computed for the 10 cross-validated folds. The first F_1 was computed for the average human, and the second one for the most similar human. This was another improvement compared to what was done in [23], where only the average human F_1 was computed). Finally, the average Precision and Recall for the 10 cross-validation folders was employed to compute the overall F_1 for both types of images, HSV and RGB, and method applied: A, B and C.

Focusing first in Sobel's results, in Tables 1 and 2 is shown the superiority of HSV aggregations against RGBs. Among the HSV aggregations, the Yager-Inspired aggregations, whose scheme is shown in Fig. 3, based on method C ('fuzzy' approach) and method A, reached the highest F_1 value (0.555). Method A proved its superiority for both, the average human and the closest human. Among the RGB aggregations, the method B reached the best performance.

It was considered as the best parameter's value the one that reached the highest F value for a certain cross-validation folder. The smoothing best parameter was equal to 1 ($\sigma_{\mathrm{smooth}} = 1$) in all the methods except in the case of Sobel's method C with maximum aggregation where it was 0. In the case of σ_{Canny} the best value for all the Canny methods was 2. In relation to the w_i, i.e., the Yager-inspired weights, the most frequent among the 10 cross-validation folders best parameter values are shown at the last column of the tables

In the case of the Canny algorithm, Tables 3 and 4 show again (as in Sobel's case) that the HSV Yager-inspired aggregations outperformed the RGB results.

In Fig. 4 are shown through an example image all the outputs, human references and original images: HSV and RGB.

A battery of Wilcoxon signed rank test with continuity correction was performed. This test pursued to find differences in F_1 median values (average human) for a 10 training-test set fold between methods. The results of all these tests can be seen in Tables 5[3] and 6, and the medians with their standard errors are shown in Fig. 5. The tests showed that the best method for HSV outperformed the best method for RGB, which means

[3] It could seem strange the repetition of p-value = 0.0020, but it was doubled-checked employing two different software. The explanation is that only 10 pairs of medians are being compared, so it is not difficult that they result in the same signs ranks sum.

that the F_1 median in HSV images was significantly higher than the F_1 median in RGB images. As well, more Wilcoxon rank tests were performed separately for RGB and HSV to seek differences between A, B and C methods. Method B was the best between RGB methods, and method HSV Yager A was found to be the best between HSVs.

5 Conclusions and Future Research

This paper yields two main findings. The first is based on the potential of Yager-inspired aggregations with HSV channels, which showed to be closer to human vision (as they reached highest F_1 values) than its equivalent RGB aggregations. Moreover, HSV aggregations presented lower standard deviation than RGBs which shows that these methods were more robust. This was tested for both Canny's and Sobel's. The superiority of HSV aggregations is remarkable, and it seems slightly counterintuitive that the same image containing the same information, but differently arranged, could result in having such a different potential for edge extraction purposes. The second finding is related to what we have called Yager-inspired aggregation. This aggregation proved to be an efficient approach for HSV image analysis. In relation to the w_i Yager-inspired weights, which are applied over the HSV images, it came out that w_1 and w_2, that perform over the 'value' and 'saturation' channels, respectively, were more relevant for the edge extraction process than w_3.

Future research can explore Canny's aggregations more in depth, due to the fact that Canny's brings more complexity as its scaling phase is based on two different thresholds in a process known as hysteresis. Then, when the scaling is made separately for each channel the resulting segments built by hysteresis tend to be shorter than when hysteresis is performed over the information provided by the three channels combined.

We are also considering the possibility of adapting these aggregation methods to edge detection based on the global evaluation with segments that was explored in [22]. Moreover, it would be worthwhile to continue exploring these aggregation methods, or equivalent ones, with other color spaces such as HSL (similar to HSV), CYMK, CIELAB, and even Super 8 [29]. As well, other interesting possibility would be comparing the proposed HSV aggregations with other HSV aggregations proposed in the literature. Finally, the approach based on mixing different channels from different color spaces using deep learning techniques seems like a promising and natural next step.

Acknowledgements For conducting this research, the code created by Kermit Research Unit has been helpful [47].

Funding This research has been partially supported by the Government of Spain, grant PGC2018-096509-B-I00.

Availability of Data and Materials The needed code to replicate the proposed methods is available at Github (https://github.com/pabloarcadio/Image-Processing.git). To run this code, the Berkeley's segmentation data set (BSDS500) is required. This data set of images can be found in the links provided in [43]. One more repository could be needed if the same Sobel and Canny functions are used[4]. This last code that was used in this research can be found in [47].

Declarations

Ethics Approval and Consent to Participate Not applicable.

Consent for Publication Not applicable.

Conflict of Interest The authors declare that they have no competing interests.

Author Contributions Conceptualization, DG, PF, and JC; methodology, JC and DG; software, PF; investigation, DG; writing—original draft preparation, PF; writing—review and editing, JM and DG; supervision, DG and JM; project administration, JM; funding acquisition, DG and JM. All authors have read and agreed to the published version of the manuscript.

References

1. Marr, D.: Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. MIT Press (1982)
2. McAndrew, A.: An Introduction to Digital Image Processing with Matlab Notes for scm2511 Image Processing, vol. 264(1), pp. 1–264. School of Computer Science and Mathematics, Victoria University of Technology, Footscray (2004)
3. Bogumil, S.: 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)
4. Smith, A.R.: Color gamut transform pairs. ACM Siggraph Comput. Gr. **12**(3), 12–19 (1978)

[4] Not significatively differences from the results of this research are expected if other versions of Canny's or Sobel's are employed instead, overall if a thinning is applied in the case of Sobel.

5. Shaik, K.B., Ganesan, P., Kalist, V., Sathish, B., Jenitha, J.M.M.: Comparative study of skin color detection and segmentation in hsv and ycbcr color space. Procedia Comput. Sci. **57**, 41–48 (2015)
6. Sandeep, K., Rajagopalan, A.: Human face detection in cluttered color images using skin color, edge information. In: ICVGIP (2002)
7. Lee, D., Wang, J., Plataniotis, K.N.: Contribution of skin color cue in face detection applications. In: Emre Celebi, M., Smolka, B. (eds.) Advances in Low-Level Color Image Processing, pp. 367–407. Springer (2014)
8. López-Molina, C.: The breakdown structure of edge detection: analysis of individual components and revisit of the overall structure. Ph.D. thesis (2012)
9. González, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Prentice Hall, Upper Saddle River, USA (2008)
10. Guada, C., Gómez, D., Rodríguez, J.T., Yáñez, J., Montero, J.: Classifying image analysis techniques from their output. Int. J. Comput. Intell. Syst. **9**, 43–68 (2016). https://doi.org/10.1080/18756891.2016.1180819
11. Sonka, M.: IEEE transactions on medical imaging statement of editorial policy. IEEE Trans. Med. Imaging **33**(4) (2014)
12. Campbell, J.B., Wynne, R.H.: Introduction to Remote Sensing. Guilford Press, New York (2011)
13. Rosin, P.: Thresholding for change detection. In: Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271), pp. 274–279 (1998). IEEE
14. Monga, O., Deriche, R., Malyain, G., Cocquerez, J.P.: Recursive filtering y edge tracking: two primary tools for 3d edge detection. Image Vis. Comput. **9**(4), 203–214 (1991). https://doi.org/10.1016/0262-8856(91)90025-K
15. Fathy, M., Siyal, M.I.: An image detection technique based on morphological edge detection y background differencing for real-time traffic analysis. Pattern Recogn. Lett. **16**(12), 1321–1330 (1995). https://doi.org/10.1016/0167-8655(95)00081-X
16. Zielke, T., Brauckmann, M., Vonseelen, W.: Intensity y edge-based symmetry detection with an application to car-following. CVGIP Image Understying **58**(2), 177–190 (1993). https://doi.org/10.1006/ciun.1993.1037
17. Pal, S.K., King, R.A.: On edge detection of X-ray images using fuzzy sets. IEEE Trans. Pattern Anal. Mach. Intell. PAMI **5**(1), 69–77 (1983). https://doi.org/10.1109/TPAMI.1983.4767347
18. Perfilieva, I., Hodáková, P., Hurtík, P.: Differentiation by the f-transform y application to edge detection. Fuzzy Sets Syst. **288**, 96–114 (2016)
19. Daňková, M., Hodáková, P., Perfilieva, I., Vajgl, M.: Edge detection using f-transform. In: 2011 11th International Conference on Intelligent Systems Design Y Applications, pp. 672–677 (2011). https://doi.org/10.1109/ISDA.2011.6121733
20. Sobel, I.: History and definition of the so-called "Sobel operator", more appropriately named the Sobel–Feldman operator (2014)
21. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. PAMI **8**(6), 679–698 (1986). https://doi.org/10.1109/TPAMI.1986.4767851
22. Flores-Vidal, P.A., Olaso, P., Gómez, D., Guada, C.: A new edge detection method based on global evaluation using fuzzy clustering. Soft Comput. **23**(6), 1–13 (2018)
23. Flores-Vidal, P.A., Gómez, D., Villarino, G., Castro, J., Montero, J.: A new approach to color edge detection. In: Atlantis Studies in Uncertainty Modelling, 2019 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (EUSFLAT 2019), pp. 376–384 (2019)
24. Guada, C., Gómez, D., Rodríguez, J.T., Yáñez, J., Montero, J.: 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, pp. 12–17 (2016). https://doi.org/10.1109/ISKE.2015.89
25. Mega, K.W., Yu, X., Li, J.: Comparative analysis of color edge detection for image segmentation. In: Proceedings of the 2018 International Conference on Computing and Pattern Recognition, pp. 93–101 (2018). ACM
26. Flores-Vidal, P.A., Gómez, D., Castro, J., Montero, J.: The different importance of each color in edge detection. In: Developments of Artificial Intelligence Technologies in Computation and Robotics—Proceedings of the 14th International FLINS Conference (FLINS2020), pp. 931–938 (2020)
27. Trahanias, P.E., Venetsanopoulos, A.N.: Color edge detection using vector order statistics. IEEE Trans. Image Process. **2**(2), 259–264 (1993)
28. Yang, Y.: Colour Edge Detection and Segmentation Using Vector Analysis. University of Toronto (1996)
29. Flores-Vidal, P.A., Gómez, D., Castro, J., Montero, J.: A new approach to color edge detection by means of transforming rgb images into an 8-dimension color space. In: Proceedings of the EEE 14th International Conference on Intelligent Systems and Knowledge Engineering (ISKE 2919), pp. 1140–1147 (2020)
30. Yager, R.R.: Prioritized aggregation operators. Int. J. Approx. Reason. **48**(1), 263–274 (2008)
31. Gnanatheja, R., Reddy, T.S.: Ycocg color image edge detection. Int. J. Eng. Res. Appl. **2**(2), 152–156 (2012)
32. Macedo-Cruz, A., Pajares, G., Santos, M., Villegas-Romero, I.: Digital image sensor-based assessment of the status of oat (*Avena sativa* L.) crops after frost damage. Sensors **11**(6), 6015–6036 (2011)
33. Bouchon-Meunier, B.: Aggregation and fusion of imperfect information. Physica **12** (2013)
34. Beliakov, G., Bustince, H., Paternain, D.: Image reduction using means on discrete product lattices. IEEE Trans. Image Process. **21**(3), 1070–1083 (2011)
35. Bustince, H., Fernández, J., Kolesárová, A., Mesiar, R.: Generation of linear orders for intervals by means of aggregation functions. Fuzzy Sets Syst. **220**, 69–77 (2013)
36. Goguen, J.A.: L-fuzzy sets. J. Math. Anal. Appl. **18**(1), 145–174 (1967)
37. Rojas, K., Gómez, D., Montero, J., Rodríguez, J.T., Valdivia Barrios, A., Paiva, F.: Development of child's home environment indexes based on consistent families of aggregation operators with prioritized hierarchical information. Fuzzy Sets Syst. **241**, 41–60 (2014)
38. Yager, R.R.: Modeling prioritized multicriteria decision making. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **34**(6), 2396–2404 (2004)
39. Dutta, S.: A Color Edge Detection Algorithm in rgb Color Space, pp. 337–340. IEEE Computer Society, Los Alamitos, CA (2009)
40. Ruzon, M.A., Tomasi, C.: Color edge detection with the compass operator. In: Proceedingsof the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), vol. 2, pp. 160–1662 (1999)
41. Di Zenzo, S.: A note on the gradient of a multi-image. Comput. Vis. Gr. Image Process. **33**(1), 116–125 (1986)
42. Arbeláez, P., Fowlkes, C., Maire, M., Malik, M.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **33**, 898–916 (2011)
43. Martin, D., Fowlkes, C., Tal, D., Malik, J.: 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, pp. 416–423 (2001)

44. Estrada, F.J., Jepson, A.D.: Benchmarking image segmentation algorithms. Int. J. Comput. Vis. **85**(2), 167–167181181 (2009). https://doi.org/10.1007/s11263-009-0251-zpmid
45. Sobel, I.: Camera models and machine perception. Technical report, Computer Science Department, Technion (1972)
46. Flores Vidal, P.A., Villarino, G., Gómez, D., Montero, J.: A new edge detection method based on global evaluation using supervised classification algorithms. Int. J. Comput. Intell. Syst. **12**(1), 367–378 (2019)
47. de Baets, B., López-Molina, C.: The kermit image toolkit (kitt), Ghent University. www.kermitimagetoolkit.net (2016). Accessed 29 Aug 2019