

Article

Class Separation Improvements in Pixel Classification Using Colour Injection

Edward Blanco ¹, Manuel Mazo ², Luis Bergasa ², Sira Palazuelos ^{2*}, Jose Rodríguez ², Cristina Losada ² and Jose Martín ²

¹ Department of Electronics and Electromechanics, Pontificia Universidad Católica Madre y Maestra, 822 Santiago, Dominican Republic; E-Mail: eblanco@pucmmsti.edu.do (E.B.)

² Electronics Department, University of Alcalá Campus Universitario s/n, 28805, Alcalá de Henares, Spain; E-Mails: mazo@depeca.uah.es (M.M.); bergasa@depeca.uah.es (L.B.); jmra@depeca.uah.es (J.R.); losada@depeca.uah.es (C.L.); jlmartin@depeca.uah.es (J.M).

* Author to whom correspondence should be addressed; E-Mail: sira@depeca.uah.es; Tel.: +34-918856581; Fax: +34-918856591.

Received: 25 June 2010; in revised form: 20 July 2010 / Accepted: 4 August 2010 /

Published: 20 August 2010

Abstract: This paper presents an improvement in the colour image segmentation in the Hue Saturation (*HS*) sub-space. The authors propose to inject (add) a colour vector in the Red Green Blue (*RGB*) space to increase the class separation in the *HS* plane. The goal of the work is the development of an algorithm to obtain the optimal colour vector for injection that maximizes the separation between the classes in the *HS* plane. The chromatic Chrominance-1 Chrominance-2 sub-space (of the Luminance Chrominance-1 Chrominance-2 (*YC₁C₂*) space) is used to obtain the optimal vector to add. The proposal is applied on each frame of a colour image sequence in real-time. It has been tested in applications with reduced contrast between the colours of the background and the object, and particularly when the size of the object is very small in comparison with the size of the captured scene. Numerous tests have confirmed that this proposal improves the segmentation process, considerably reducing the effects of the variation of the light intensity of the scene. Several tests have been made in skin segmentation in applications for sign language recognition via computer vision, where an accurate segmentation of hands and face is required.

Keywords: pixel classification; colour clustering; colour segmentation; class separation; colour sub-spaces; colour injection

1. Introduction

In recent years, a significant amount of work has been published in the field of colour segmentation for Human Computer Interfaces (*HCI*). We would like to emphasize those related to the segmentation of the natural colour of skin. In this area, Phung *et al.* [1] proposed a skin segmentation method using a Bayesian classifier, obtaining satisfactory results for different colour spaces such as: *RGB*, Hue Saturation Value (*HSV*), Luminance blue-Chrominance red-Chrominance (YC_bC_r) and Commission Internationale de l'Éclairage's Luminosity a-channel b-channel (*CIE-Lab*), even under adverse illumination conditions. Hsu *et al.* [2] suggested the detection of face skin considering a nonlinear subspace from the YC_bC_r space to partially compensate the luminosity variations.

The robustness of the segmentation against luminosity changes is one of the most desirable features in colour segmentation systems. For this reason, much work on this topic has been focused on minimizing the effects of illumination changes by using colour spaces where the luminance or intensity component can be easily isolated, thus providing chromatic constancy. The actual trend in applications with important time-varying-illumination changes is to use dynamic colour models that can adapt themselves to compensate for variations of the scene illumination. In this area, an extensive overview of previous investigations in the skin colour segmentation field is presented by Sigal *et al.* [3].

The most frequently used colour spaces in these types of applications are *HSV* [3,4] and normalized Red Green (*rg*) [5-7]. The *HSV* space, as well as the Hue Saturation Intensity (*HSI*) and Hue Lightness Saturation (*HLS*) spaces, are widely used in image processing because it is very intuitive for the human brain to interpret the information as it is represented. In some works, only the Hue (*H*) and Intensity (*I*) components are used in the clustering process [8]. In other cases, a threshold value for the Saturation (*S*) of each pixel based on its intensity is defined [9]. This threshold is used before the clustering process to determine if *S* should be replaced by *H* or *I*.

In general, all these segmentation proposals offer good results for objects with significant size in the scene or in cases where the main goal is object tracking, but not in the case of shape recognition. If the goal is to recognize the object shape, the system requirements are higher and very accurate segmentation techniques should be applied. Further difficulties may arise if the images have low quality and spatial resolution. Sign language recognition systems based on computer vision are a good example of these types of applications. In this case, the camera should capture all the upper parts of the speaker's body, implying that the parts to segment (hands and face) constitute a small part of the captured scene. In this field, Habili *et al.* [10] performed a pixel-by-pixel classification of the skin colour with discriminant features of the C_bC_r plane, using the Mahalanobis distance, but they needed a fusion of motion cues to obtain good results. Similar skin segmentation is achieved in the work done by Chai *et al.* [11], where post-segmentation stages were applied, such as morphological operations, in order to surpass the limitations of the segmentation. The YC_bC_r space has been also used [11]. This colour space is one of the most widely used in the segmentation process.

In this field, Ribiero and Gonzana [12] presented hand segmentation in video sequences by means of the Gaussian Mixture Model (GMM) background subtraction algorithm, which is a well-known statistical model for density estimation due to its tractability and universal approximation capability. In this work, [12], an adaptive Gaussian mixture in time is used to model each pixel distribution in *RGB*

space. In Huang and Liu's work [13], clustering of colour images using GMM technique in *HSV* space is performed.

Less common colour spaces are also used in other works: both linear transformation spaces, like Luminance E-channel S-channel (*YES*) [14], and non-linear, spaces like the Uniform Chromaticity Scale (*UCS*) spaces, such as Luminance u-channel v-channel ($L^*u^*v^*$) and its representation in cylindrical coordinates Intensity Hue Saturation (*IHS*) [15], Saturation Tint Value (*STV*) [16] which is a representation of *HSV* space by the normalized *RGB* components. Other spaces used are the Spherical Coordinate Transform (*SCT*) [17] and the geodesic chromaticity space pq [18].

We can also find works related to object/background segmentation with the objective of efficiently delimiting object edges. Some of these publications present the use of graph cuts in N-dimensional images to segment medical images from computed tomography (CT) scanners [19,20], and multilevel graph cuts to accelerate the segmentation and optimize memory use [21]. From our point of view, the main disadvantage of these works is that they are not designed for real-time purposes.

The conclusion of these previous works is that important unresolved problems still exist in order to obtain efficient skin segmentation, especially if we take into account that many applications require real time processing, include complex scenes, are prone to important illumination changes, and the objects to segment (face, arms and hands) are small when compared to the captured scene.

Our contribution to the solution of this segmentation problem is to use an object/background pre-processing technique to enhance the contrast (in the *HS* plane) between the colours corresponding to the objects to segment and the background in each frame. This pre-processing consists of increasing the separation between the object and background classes in the *HS* plane to optimize the segmentation in that plane.

In our proposal, to increase the class separation, a colour vector of components Δ_R , Δ_G , Δ_B , is added to the *R*, *G* and *B* images directly captured from the camera, modifying the value of each pixel (n) to $(R_n + \Delta_R, G_n + \Delta_G, B_n + \Delta_B)$. The objective of this paper is to present the process needed to obtain the values Δ_R , Δ_G and Δ_B that optimize the separation between the classes of interest once the image has been converted to the *HS* plane. This optimization is carried out by means of an algorithm that maximises the Fisher Ratio. We have called the colour vector addition process "colour injection". In our proposal, the colour injection process is achieved using the relationships between the *RGB*, YC_1C_2 [8,22,23] and *HSI* [24-26] colour spaces, and the properties of the C_1C_2 plane.

Our system may be particularized to recognize sign language in real-time and special attention has been paid to the detection of the geometric form of the parts to segment, hands and face edges, in each frame. Our proposal has been thoroughly tested with very good results even with illumination variations, because it isolates the *I* component. We always attempt to work outside the instability or achromatic zone of the *HS* plane, due to the convenient redistribution in the *HS* plane of the existing classes in the colour injected image (seen in [15] for the *IHS* space). In order to perform a comparative qualitative study between the segmentations of the original images and the colour injected images (proposal presented in this paper), a GMM clustering technique in the *HS* classification domain is used. This technique has been used in a similar way for the *HSV* space [13]. In previous works, different formulations for the *HSI* space can be found [22,24-26]. We use the formulation proposed in [26].

This paper has been organized as follows: Section 2 describes the basis of the proposed algorithm to increment the separation between classes. Section 3 presents the criteria considered when separating the classes. Section 4 describes the off-line initialization stage of the proposed algorithm. Section 5 details how to improve the separation between classes in the HS plane starting from their location in the C_1C_2 plane. Section 6 presents the algorithm that performs the optimal class separation. Section 7 describes how to obtain the colour vector for injection, and its effects in the captured images. Section 8 contains the experimental results, and Section 9 provides the conclusions and future work.

2. Overview of the Colour Injection Algorithm

The objective of this work is to improve the segmentation process using colour injection. In order to do that, a colour vector for injection is obtained for each captured image in the RGB space. This colour vector is considered optimal, because it is calculated to maximize the separation between the classes to segment in the HS plane (subspace where the segmentation is performed). For this reason, this colour vector will be called optimal colour vector in this paper and will be denoted by \mathbf{i}_r . It is injected in the RGB space and is calculated starting from significant samples (seeds) from the object to segment and from the part of the scene considered as background. The procedure to obtain the vector \mathbf{i}_r , and the reason why it is optimal is explained in Sections 6 and 7. This optimal colour vector is given by:

$$\mathbf{i}_r = [\Delta_R \ \Delta_G \ \Delta_B]^T \quad (1)$$

where Δ_R , Δ_G and Δ_B are the increments of the colour components R , G and B , respectively.

The optimal colour vector, \mathbf{i}_r , is injected in every frame of an image sequence in real-time applications to segment objects in colour images. Its efficiency has been especially tested in applications where a reduced contrast between the background colour and the colour of the object to segment exists, when there are illumination changes and the size of the object to segment is very small in comparison with the size of the captured scene.

An important property of the perceptual colour spaces (such as the HSI space) is that they produce a maximum disconnection between the chrominance and luminance components. As a result, the luminance can be almost fully isolated, making the segmentation process more invariant to the changes in shades and illumination as in [4]. For this reason, the analysis of the colour injection effects in the HSI space is made only using the H and S chromatic components (HS plane).

As the segmentation is performed using the HS components, we try to separate the representative vectors of the two classes (object and background) in angle (H component) and in magnitude (S component) using colour injection. However, special attention should be paid in this separation process to the variations of the dispersions (reliability) of both classes after the colour injection, because it has a very high incidence in the class separation process.

In short, if the original image is denoted by I , the optimal colour vector to add by \mathbf{i}_r , and the coloured image resulting of the colour injection by I_i , is fulfilled:

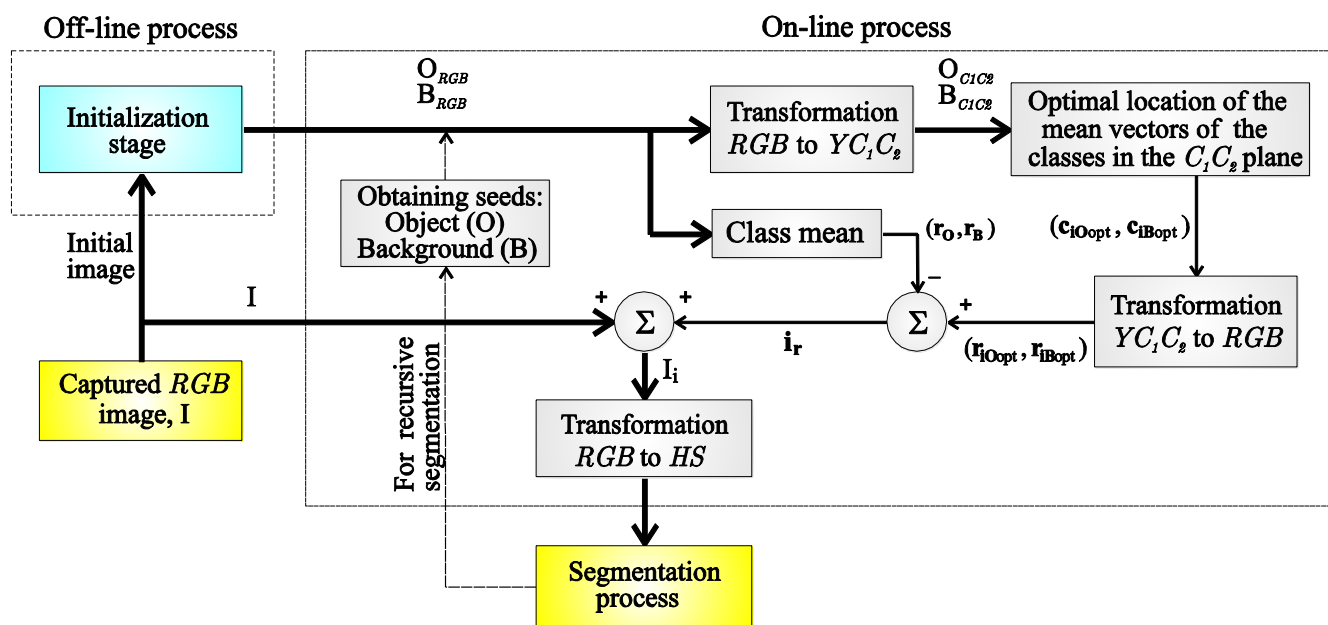
$$I_i = I + \mathbf{i}_r \quad (2)$$

The algorithm proposed in this work is formed by two clearly different stages: an off-line and an on-line stage. The off-line stage is an initialization phase whose objective is to determine the optimal

number of existing classes in the initial frame, and, from that, to obtain the object (O class) and background (B class) classes needed to carry out their separation. The off-line stage is explained in detail in Section 4. The result of this stage is the set of significant pixels (seeds) in the RGB space that represent both classes, identified by: $O_{RGB} = \{\mathbf{r}_{O1}, \mathbf{r}_{O2} \dots \mathbf{r}_{ON}\}$, and $B_{RGB} = \{\mathbf{r}_{B1}, \mathbf{r}_{B2} \dots \mathbf{r}_{BM}\}$, respectively, where \mathbf{r}_{Or} for $r = 1, 2 \dots N$ and \mathbf{r}_{Bq} for $q = 1, 2 \dots M$ refer to the pixel vectors of the object and background classes, respectively.

The on-line stage is the novel contribution of this paper. Its objective is to determine the optimal colour vector to inject (\mathbf{i}_r) for each frame in order to increase, optimally, the separation between classes O and B. The on-line process is executed before the segmentation process for each frame captured in real time. Figure 1 depicts the different phases of a segmentation process that uses the colour injection proposal of this paper.

Figure 1. General block-diagram of the proposed algorithm to obtain the optimal colour vector (\mathbf{i}_r) to be injected to the captured image I . The off-line and on-line processes are grouped by discontinuous lines.



The on-line process consists of the following stages:

(1) For every O_{RGB} and B_{RGB} sample from the captured RGB image I , a transformation to the YC_1C_2 space is done. Considering the chromatic components after the transformation, the resulting classes will be referred to as $O_{C_1C_2} = \{\mathbf{c}_{O1}, \mathbf{c}_{O2} \dots \mathbf{c}_{ON}\}$ for the object class and as $B_{C_1C_2} = \{\mathbf{c}_{B1}, \mathbf{c}_{B2} \dots \mathbf{c}_{BM}\}$ for the background, where the pixel vectors are denoted by “ \mathbf{c} ”.

(2) Using the properties of the C_1C_2 plane and the relationship between the HSI and YC_1C_2 colour spaces, the optimal location of the classes in the C_1C_2 space is obtained by finding the optimal location of their respective mean vectors. The optimal location is the one that maximizes the class separation in the HS plane (maximum distance between the class means and minimum class dispersions). These optimal mean vectors will be referred to as \mathbf{c}_{iOpt} and \mathbf{c}_{iBOpt} . This phase is, undoubtedly, the most important of this work, and will be described in detail in subsequent sections.

(3) From the mean vectors $\mathbf{c}_{iO_{opt}}$ and $\mathbf{c}_{iB_{opt}}$, their corresponding ones in the RGB space, $\mathbf{r}_{iO_{opt}}$ and $\mathbf{r}_{iB_{opt}}$, are calculated.

(4) From the vectors $\mathbf{r}_{iO_{opt}}$ and $\mathbf{r}_{iB_{opt}}$, and the mean vectors of the original classes (O_{RGB} , B_{RGB}) denoted by \mathbf{r}_O and \mathbf{r}_B , the optimal colour vector for injection is obtained. This optimal colour vector can be calculated from one of these expressions:

$$\mathbf{i}_r = \mathbf{r}_{iO_{opt}} - \mathbf{r}_O, \quad \mathbf{i}_r = \mathbf{r}_{iB_{opt}} - \mathbf{r}_B \quad (3)$$

(5) Once the optimal colour vector has been obtained, the new “injected” image I_i can be calculated applying (2).

Finally, the coloured image I_i is transformed from the RGB space to the HS plane, where the segmentation is done, because the colour injection has its effects in the HSI space: the increase in the separation between the classes only happens in the HSI space or HS plane (in the RGB space the colour injection only produces a translation of the classes, keeping the distance between them constant, independently of the colour injection).

The proposed method can be implemented easily and can be used in real-time applications. In the following sections, the process to obtain the optimal vector for injection is presented in detail. At the end of this paper, in order to facilitate its reading, we have included three appendices with aspects related to the relationships between the RGB , the HSI and YC_1C_2 spaces (Appendix A), statistical analysis of vectors in the RGB space and its relationships with the components in the HSI space (Appendix B), as well as the invariants of the mean vectors in the C_1C_2 plane (Appendix C).

3. Criteria for the Separation between Classes

Since the objective of our work is to obtain a higher separation between the classes to facilitate the segmentation, it is necessary to define a measure of the efficiency of our proposal. The Fisher Ratio (FR) is frequently used to measure the efficiency in the class separability in classification systems [6,27,28]. This ratio quantifies simultaneously the inter-class separation and the internal dispersion (reliability) of the classes. For a two-class system, it is interesting to achieve a large distance metric between the class means and a minimum dispersion within each class (leading to a high FR). In this work, the FR is used as a pixel classification measurement index, using as discriminant features the H and S components of each pixel.

In a multi-class system, the generalized Fisher Ratio is expressed by [29]:

$$FR = \text{tr}(M_w^{-1}M_b) \quad (4)$$

where M_b is the inter-class (between class) covariance matrix and M_w is the internal (within class) dispersion matrix of the classes.

Equation (4) cannot be directly applied due to the circular form of the H component trajectory. There are two main reasons for this:

(a) For two-class systems (as our case is), M_b may not represent the real angular distance between the hue means of the classes (the maximum angular distance between two vectors is π radians, even if one of the vectors is in the first quadrant of the HS plane and the other one in the fourth one). These problems have already been studied, for example in [8].

(b) The second reason is the discontinuity of the hue component when it moves from 2π to 0 radians (cyclic property). This implies that M_w matrix does not represent the real hue variance of a class whose mean is close to 0 (2π). The reason is that some of the vectors would have small angles (close to 0), and some others would have very high ones (close to 2π), resulting in a wrong and high variance. The resulting H mean would also be wrong.

For the previous reasons, and supposing that the correlation between H and S is low, a particular FR has been defined. This FR is individually calculated for each component, and, as our space is bi-dimensional, is given by [29]:

$$FR = FR_H + FR_S \quad (5)$$

where FR_H and FR_S represent the Fisher Ratio of the H and S components, respectively, and are given by:

$$FR_H = \frac{\theta_h^2}{\sigma_{HO}^2 + \sigma_{HB}^2}, \quad FR_S = \frac{(S_O - S_B)^2}{\sigma_{SO}^2 + \sigma_{SB}^2} \quad (6)$$

where $S_O - S_B$ is the distance between the saturation means of both classes, σ_{SO} and σ_{SB} are the standard deviations of the saturation component for both classes, θ_h is the separation angle between the hue means of both classes, and σ_{HO} and σ_{HB} are the standard deviations of the H component.

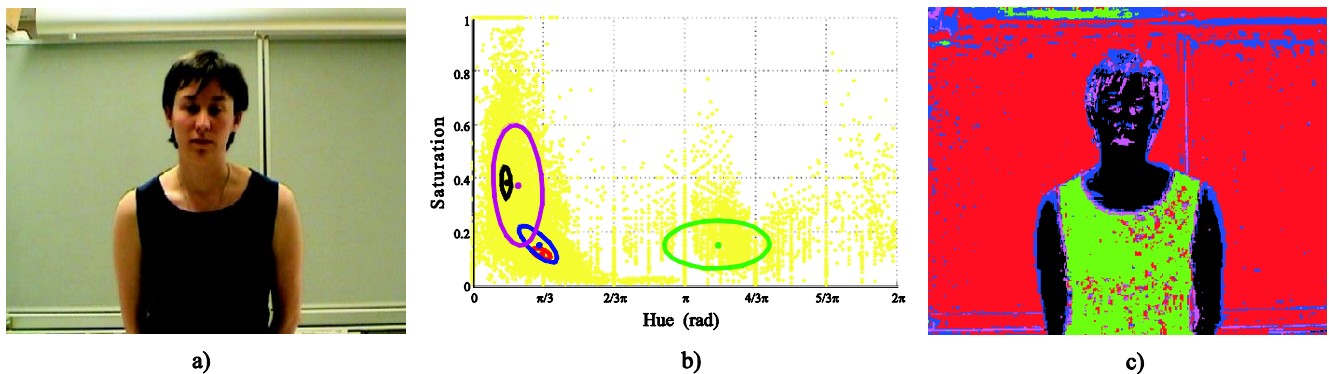
In (6) $\theta_h \in [0, \pi]$ represents the real angular distance between the hue means, because $\theta_h = \cos^{-1}(CC_{OB})$. This avoids the aforementioned problem about the angular distance between the hue means of the mean vectors of both classes in the HS plane. CC_{OB} is the correlation coefficient between the two mean vectors of the RGB components that have generated the mean vectors in the HS plane (Equation C.11) (see Appendix C). In this work, we have performed approximations in the calculation of σ_{HO} and σ_{HB} in order to avoid the problem of the hue discontinuity. Thus, the approximation for σ_{HO} is: $\sigma_{HO}^2 = \sigma_{CHo}^2 + \sigma_{SHo}^2$, where σ_{CHo} is the standard deviation of every $\cos(H_{Or})$ for $r = 1, 2, \dots, N$ and σ_{SHo} is the standard deviation of every $\sin(H_{Or})$. σ_{HB} is calculated with a similar method.

4. Initialization Stage (Off-Line Process)

The first step of the off-line process is the capture of a first frame (initial image). The seed pixels that represent the classes O and B are obtained from this image, by means of any clustering technique used to identify the existing classes of the image, such as K-Means [30], GMM [13], *etc.* In this paper, the GMM technique is used (in the HS domain) because it provides highly reliable classes, and, as a final result, it also provides the mean vectors, the covariance matrixes and the a priori probabilities of the classes. The clustering by means of GMM uses the EM algorithm (Expectation Maximization) to obtain the optimal location and dispersion of a predefined image class number (K), projected in the HS plane. Therefore, a Gaussian model is assumed for each existing class, considering a uniform scene illumination. The GMM algorithm is applied several times, initializing it with different K values, in order to obtain the optimal number of existing classes in the image (K_{opt}). K_{opt} corresponds with the K that produces the smallest error in the log-probability function of the EM algorithm, indicating that it is the best fit between the K Gaussians and the existing classes. Figure 2b shows the K_{opt} Gaussians projected in the HS plane, fitted to the existing classes in the initial image of the example in Figure 2a.

Finally, Figure 2c depicts the original image segmentation as a function of the different existing classes (Figure 2b).

Figure 2. Class segmentation results of the initialization stage. (a) Initial image, (b) the K_{opt} Gaussians fitted to the classes projected in the HS plane, (c) segmented image corresponding with the K_{opt} classes in the Figure 2b. The colours of the different ellipses that represent the Gaussians in the Figure 2b correspond with the colours of the segmented regions in the image of the Figure 2c.



Once the GMM algorithm has converged, the following step is to find out the localization of the object class (O) in the HS plane. This off-line process is carried out easily, because the approximate location of the object class (O) in the HS plane is known at the beginning of the off-line process, as a result of the colour calibration adjustments of the camera. This approximate geometric locus in the HS plane is given by the mean vector \mathbf{h}_{init} . Taking that into account, the detection of the object class (O) is performed by simply selecting the class with the minimum Euclidean distance with \mathbf{h}_{init} . We preferred the Euclidean distance over the Mahalanobis distance because the detection of the object class could be incorrect if \mathbf{h}_{init} is close to a class with high dispersion, and this class is also close to the object class (O). The reason of this effect is the consideration of the class covariance in the Mahalanobis distance.

Once the object class is detected, the next step is to select the background class (B). The background is usually formed by several classes, identified by $\{B_1, B_2, \dots, B_{K_{opt}-1}\}$. Our objective is to select the B_k class that will be considered as representative of the background and that we will be identified simply by B. Among all the classes that form the background, we will select the B_k that:

$$B = \arg \min_{B_k} \left(\frac{P_{FRk}(B_k)}{P_{Bk}^2(B_k)} \right) \quad (7)$$

being P_{FRk} ; $k = 1, 2, \dots, (K_{opt} - 1)$ the Fisher Ratio probabilities between the class O and each B_k , defined by $P_{FRk} = (\sqrt{FR_k} / \sum_{k=1}^{K_{opt}} \sqrt{FR_k})$ where FR_k is the Fisher Ratio described in Section 3, and P_{Bk} ; $k = 1, 2, \dots, (K_{opt} - 1)$ the a priori probabilities of each B_k class to be the background class (B) of the image (given by the GMM algorithm).

Once the classes O and B have been identified, the seed pixels that represent both classes are obtained through an initial segmentation process of both the object class (O) and the background class (B). In this initial segmentation, the *pdf* (probability density function) of both classes in the HS plane are

considered as unimodal bidimensional Gaussians, defined by the parameters obtained by the GMM clustering. This segmentation is carried out by selecting the pixels with higher probability to belong to the corresponding Gaussian. This stage of pixel selection is performed by truncating each class *pdf* with a determined threshold. This threshold corresponds to a percentage of the maximum probability of the corresponding bidimensional *pdf*, P_o , for the class O, and P_b for the class B. The values of $\{P_o, P_b\} \in [1, 0]$, have been experimentally set to $P_o = 0.45$ and $P_b = 0.6$, using the Receiver Operating Characteristic (ROC) curves obtained from the different tests performed with real images. In this case a ROC curve was obtained by each class (O and B), using a set of real images, without and with colour injection. The thresholds P_o and P_b correspond with the nearest values to the elbows of the ROC curves. As the pixel selection is carried out in the *HS* plane, it is necessary to truncate the *pdf* in the Intensity axis, in order to obtain pixel sets that reliably represent the classes O and B in the *RGB* space. The truncation of this unidimensional *pdf* is necessary because the *H* and *I* components are independent (Appendix A) (this is important when the clustering is carried out in the *HS* plane) and this generates correspondence problems when the pixels in *RGB* components are selected from its projections in the *HS* plane. In this second *pdf* truncation, the percentages selected of the maximum value of the *pdf* intensity of each class are P_{fo} for O, and P_{fb} for B. These percentages have been set to $P_{fo} = 0.4$ and $P_{fb} = 0.5$.

Once the previous process is completed, a random sampling is carried out, selecting N samples for the class O and M for the class B, in order to reduce the working space dimension. This is the method to obtain the sets O_{RGB} and B_{RGB} mentioned in Section 2.

5. Separation of the Classes in the *HS* Plane from Their Location in the C_1C_2 Plane

This section details the most important relationships between the statistical mean and variance of the classes in the C_1C_2 and *HS* planes. Also, the effect of adding the same vector (colour injection) to two vectors in *RGB* space on the projections of these vectors in the C_1C_2 and *HS* planes is analyzed. This information is used to define an algorithm to easily calculate the optimal vector to inject in order to obtain the maximum separation between classes in the *HS* plane using translations in the plane C_1C_2 .

5.1. Relationships between the *HS* and C_1C_2 Planes

Given two vectors in the *RGB* space, \mathbf{r}_O and \mathbf{r}_B , the resulting projection vectors in the C_1C_2 plane, \mathbf{c}_O and \mathbf{c}_B , and in the *HS* plane, \mathbf{h}_O and \mathbf{h}_B , fulfil (see Appendix A):

$$\theta_c = \theta_h = \theta \quad (8)$$

$$\|\mathbf{c}_O\| \neq \|\mathbf{h}_O\|, \|\mathbf{c}_B\| \neq \|\mathbf{h}_B\| \quad (9)$$

$$\|\mathbf{d}_c\|^2 = g_1(\mathbf{c}_O, \mathbf{c}_B, \theta_c) = \|\mathbf{c}_O\|^2 + \|\mathbf{c}_B\|^2 - 2\|\mathbf{c}_O\|\|\mathbf{c}_B\|\cos\theta_c \quad (10)$$

$$\|\mathbf{d}_h\|^2 = g_2(\mathbf{c}_O, \mathbf{c}_B, \theta_h, I_O, I_B, f(H)) \quad (11)$$

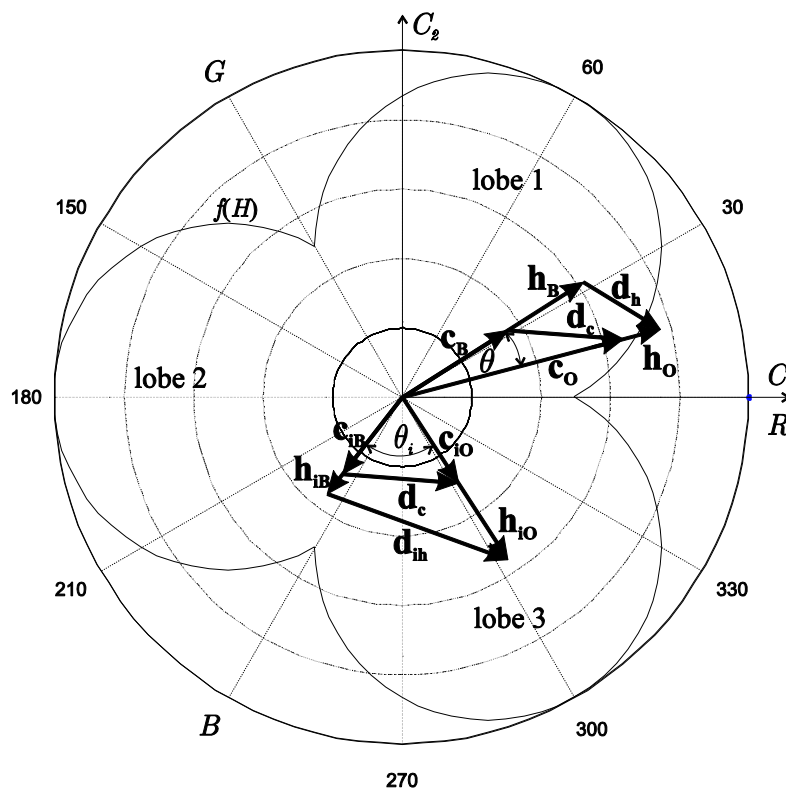
where θ_c is the angle between \mathbf{c}_O and \mathbf{c}_B , θ_h the angle between \mathbf{h}_O and \mathbf{h}_B ; \mathbf{d}_c is the distance vector between \mathbf{c}_O and \mathbf{c}_B , \mathbf{d}_h is the distance vector between \mathbf{h}_O and \mathbf{h}_B ; and I_O, I_B are the intensity means of

both classes, object and background, respectively, corresponding to the \mathbf{h}_O and \mathbf{h}_B vectors. $f(H)$ is a weighting function that depends on the H component. $f(H) \in [1/2, 1]$ (see Appendix B).

It is important to note that, since the C_1C_2 plane is linear, when adding a vector \mathbf{i}_r (injected vector) to both \mathbf{r}_O and \mathbf{r}_B in the RGB space, the distance vector $\mathbf{d}_c = \mathbf{c}_O - \mathbf{c}_B$ in the C_1C_2 plane remains constant. These constant magnitude and orientation values (invariants of the \mathbf{d}_c vector) are denoted by $\|\mathbf{d}_c\|$ and ϕ (see Appendix C). Therefore, colour injections in the C_1C_2 plane result in class translations, as in the RGB space. This effect can be achieved with a translation vector \mathbf{i}_c (corresponding to \mathbf{i}_r) directly added in the C_1C_2 plane.

Moreover, in the case of the C_1C_2 plane, (10) is verified (cosine law). Therefore, given that $\|\mathbf{d}_c\|$ remains constant for the different values of \mathbf{i}_r , the values of θ , $\|\mathbf{c}_O\|$ and $\|\mathbf{c}_B\|$, will be modified as a function of the value of \mathbf{i}_r . In the case of the HS plane, it must be said that if \mathbf{i}_r is added to the vectors \mathbf{r}_O and \mathbf{r}_B (contrary to what happens in the C_1C_2 plane) the difference vector \mathbf{d}_h also varies. The reason is that, according to (11), \mathbf{d}_h depends on the value of I_O and I_B and on the $f(H)$ weighting function. In any case (8) always holds.

Figure 3. Correspondence between the mean vectors in the C_1C_2 plane and the ones in the HS plane. The difference vector \mathbf{d}_c before and after the colour injection is shown.



In short, to calculate the value of the colour vector to be added in the RGB space to obtain a particular separation between the classes in the HS plane, the authors suggest using the relationships between the \mathbf{h} vector components in the HS plane, and their corresponding \mathbf{c} vector components in the C_1C_2 plane, given by (Equation B.12) and (Equation B.13) (see Appendix B), and the relationship between pairs of vectors in these planes, given by (8, 9, 10 and 11).

Therefore, the proposed algorithm is based on the analysis of the behaviour of the vectors \mathbf{c}_O and \mathbf{c}_B in the C_1C_2 plane and the properties of its difference vector \mathbf{d}_c ($\|\mathbf{d}_c\|$ and ϕ are invariant). These invariants allow us to establish a mathematical relationship between the class mean vectors before and after the colour injection. Thus, for example, the separation angle (hue difference) between two vectors in the HS plane can be easily controlled with the separation angle of the same vectors (\mathbf{c}_O , \mathbf{c}_B) in the C_1C_2 plane, because both angles coincide (8).

In Figure 3, an example of the correspondence between the vectors \mathbf{c}_O and \mathbf{c}_B in the C_1C_2 plane and the vectors \mathbf{h}_O and \mathbf{h}_B in the HS plane is shown. The relationships after performing the colour injection (vectors \mathbf{c}_{iO} , \mathbf{c}_{iB} and \mathbf{h}_{iO} , \mathbf{h}_{iB}) are also shown, as well as the difference vector \mathbf{d}_c before and after the colour injection, where the invariance in magnitude and angle can be observed. From now on, the “i” or “i” subscript refers to “colour injection”.

Figure 3 depicts how the translation of the vector \mathbf{d}_c has favoured the separation of the mean vectors of the classes in both components (H and S), because $\theta_i > \theta$, and $(\|\mathbf{h}_{iO}\| - \|\mathbf{h}_{iB}\|) > (\|\mathbf{h}_O\| - \|\mathbf{h}_B\|)$. An increase in the separation between the vectors after the colour injection can be verified ($\theta_i > \theta$). However, the vector modules (saturation) decrease ($\|\mathbf{h}_{iO}\| < \|\mathbf{h}_O\|$, $\|\mathbf{h}_{iB}\| < \|\mathbf{h}_B\|$), since there is an unavoidable compensation effect given by (10) (notice that for a fixed I , $\|\mathbf{h}\| = \text{const}\|\mathbf{c}\|f(H)$).

We could obtain a great number of class locations within the HS plane relocating \mathbf{d}_c with \mathbf{i}_c all over the C_1C_2 plane. The determination of the optimal location is not a trivial task. In order to obtain an optimal \mathbf{i}_c , it is possible to apply learning techniques, such as fuzzy systems and neural networks, that take as parameters some functions derived from FR 's (6) and the invariants of the vector \mathbf{d}_c .

In the following Section (5.2), an algorithm for the calculation of the optimal \mathbf{i}_r (corresponding to optimal \mathbf{i}_c), conditioned to $\|\mathbf{c}_{iO}\| = \|\mathbf{c}_{iB}\|$, is explained.

5.2. Separation between the Hue Means (Angular Separation)

The separation between the hue means is given by the angular separation between the vectors \mathbf{h}_{iO} and \mathbf{h}_{iB} , which indicate the colour separation. Once the expression of the distance between \mathbf{h}_{iO} and \mathbf{h}_{iB} is obtained, $\|\mathbf{d}_{ih}\|$ (see Figure 3), an optimization process can be applied to it as a function of the RGB components of \mathbf{i}_r in order to obtain the optimal \mathbf{i}_r that produces the maximum separation needed. The problem when calculating the optimal colour vector is that it is not possible to obtain its analytical expression, mainly due to the discontinuities in the function of $\|\mathbf{d}_{ih}\|$ (11).

In order to solve the problem posed by the discontinuities of $\|\mathbf{d}_{ih}\|$ in the HS plane, the authors propose to use the C_1C_2 plane, where the distance function between the vectors \mathbf{c}_{iO} and \mathbf{c}_{iB} , ($\|\mathbf{d}_c\|$) (10) does not present discontinuities, and, as well, remains constant in magnitude and direction for different injections of colour vectors.

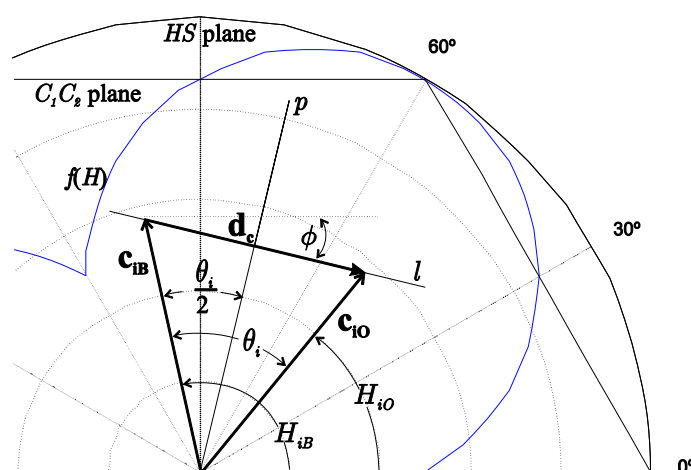
The interrelationship (due to the invariants of the vector \mathbf{d}_c and the relationships between the HS and C_1C_2 planes) between the angle (θ_i) that the vectors \mathbf{c}_{iO} and \mathbf{c}_{iB} form and their modules (10) should be taken into account to obtain the separation angle (hue difference) of two vectors in the HS plane. Therefore, the maximum separation angle between the vectors may imply (due to the compensation effect) a diminution of their modules, and, consequently, the saturation of both vectors. The saturation reduction of the vectors \mathbf{h}_{iO} and \mathbf{h}_{iB} implies that they become closer to the achromatic zone (the origin

of the coordinates system), which means that the colours approximate to gray scale. The consequence of this phenomenon is the loss of discriminating power in the segmentation.

Therefore, the proposed algorithm has been parameterized as a function of the mentioned separation angle θ_i between the vectors \mathbf{c}_{iO} and \mathbf{c}_{iB} . In our case, the optimal angle θ_i is obtained from an observation function that measures the effectiveness of the class separation in different locations in the HS plane. This function will be described in paragraph f of Section 6.

When the angle of separation θ_i reaches a maximum, θ_i coincides with the angle whose bisector is a straight line p , which passes through the origin of coordinates and is perpendicular to the straight line, l , whose director vector is \mathbf{d}_c (Figure 4).

Figure 4. Location of the vectors \mathbf{c}_{iO} and \mathbf{c}_{iB} in the C_1C_2 plane once the colour injection has been performed.



Therefore, the vector for injection (\mathbf{i}_r) that causes the maximum hue difference, causes the modules of both vectors \mathbf{c}_{iO} and \mathbf{c}_{iB} to become equal ($\|\mathbf{c}_{iO}\| = \|\mathbf{c}_{iB}\|$). It also causes the distance between the intersection point of the lines p and l and the extreme of each vector to be $\|\mathbf{d}_c\|/2$. Figure 4 illustrates an example of the location of the vectors \mathbf{c}_O and \mathbf{c}_B after the injection of the colour vector (\mathbf{c}_{iO} and \mathbf{c}_{iB}) with those imposed restrictions.

The authors have given more importance to the angular (H) separation, because increasing both H and S at the same time is not possible. The main reason is that H has a discrimination power higher than S . Besides, the H component is totally uncorrelated to the I component, which does not occur to the S component (see Appendix B). Parameterization only by θ_i implies that we can only control the separation between the hue means. The starting point to obtain the distance between the saturation means is mainly the location of the vectors \mathbf{c}_{iO} and \mathbf{c}_{iB} with respect to the saturation weighting function in the C_1C_2 plane. As can be observed, in this process there is no control on the S component, so its contribution on the class separation will depend on the modification of the statistics of this component with the variation of θ_i .

5.3. Separation between the Saturation Means (Saturation Difference)

In this section, an analysis of the behaviour of the separation between the saturation components of two vectors in the HS plane is performed. Given two vectors, for example \mathbf{h}_{iO} and \mathbf{h}_{iB} , in the HS plane, we analyze how the value of the saturation difference between both vectors $S_O - S_B = \|\mathbf{h}_{iO}\| - \|\mathbf{h}_{iB}\|$ varies. In our case, as $\|\mathbf{c}_{iO}\| = \|\mathbf{c}_{iB}\| = C_i$, then the intensities (I_O, I_B) corresponding to both vectors \mathbf{h}_{iO} and \mathbf{h}_{iB} , and the value of the saturation weighting function $f(H)$ of each one, are the parameters with significant effect in the value of $S_O - S_B$. The reason is that, according to (Equation B.11), the difference $S_O - S_B$ will only have a non-zero value if I and $f(H)$ of both vectors are different (notice that the saturation varies inversely with the intensity, and directly with $f(H)$). As an example, Figure 3 shows vectors \mathbf{h}_{iO} and \mathbf{h}_{iB} (overlapped to their respective vectors \mathbf{c}_{iO} and \mathbf{c}_{iB}), as well as the saturation weighting curve $f(H)$. In the case of Figure 3, the colour injection is done supposing $I_O = I_B$, therefore, the weighting function $f(H)$ is the only responsible for the difference in the module of the vectors \mathbf{h}_{iO} and \mathbf{h}_{iB} , that is, of the separation between the saturation means of both classes. As previously indicated, in our proposal there is no control of the $S_O - S_B$ value, but its behaviour as a function of the colour injections performed, parameterized by θ_i , is known. According to this, it can be said that $S_O - S_B$ is determined, as expressed in (12), by: (a) the intensities of the vectors \mathbf{h}_{iO} and \mathbf{h}_{iB} (I_O, I_B), and (b) the module and angle of \mathbf{d}_c (the invariants) since these determine the location of the vectors \mathbf{h}_{iO} and \mathbf{h}_{iB} along the curve $f(H)$ in the HS plane. In the case of Figure 3, where \mathbf{h}_{iO} is located in the third lobe and \mathbf{h}_{iB} in the second, it is fulfilled:

$$S_O - S_B = k_1 \cot(\theta_i/2) + k_2 \quad (12)$$

where:

$$k_1 = \|\mathbf{d}_c\| (I_B \cos(5\pi/6 + \phi) + I_O \cos(\pi/2 + \phi)) / 3I_O I_B \quad \text{and}$$

$$k_2 = \|\mathbf{d}_c\| (I_O \sin(\pi/2 + \phi) - I_B \sin(5\pi/6 + \phi)) / 3I_O I_B .$$

5.4. Analysis of the Class Dispersion

In order to obtain the optimal vector for injection, \mathbf{i}_r , by means of the suitable election of θ_i , we should take into account not only the information given by the mean vectors \mathbf{c}_O and \mathbf{c}_B in the C_1C_2 plane, but also the dispersion of the distributions of both classes.

In this section we analyze the behaviour of the class dispersions in the HS plane, that is, how the hue and saturation dispersions are affected when the classes are translated in the C_1C_2 plane, as a result of the colour injection. A class separation measurement function will be defined to quantify the effectiveness of the colour injection. This analysis will be necessary to understand how the H and S dispersions are modified with the colour injection, in addition to the performance of the class separation measurement function.

5.4.1. Hue dispersion (Angular dispersion)

The hue dispersion is determined by the effects of the dispersion transformation when passing from the C_1C_2 plane to the HS plane. If \mathbf{R}_o is the $(2 \times N)$ matrix formed by the N vectors of the O class: \mathbf{c}_O ; $r = 1, 2, \dots, N$, before any translation, the parameters of the O class uncertainty ellipse, *i.e.*, the hue dispersion invariants, are obtained from the covariance matrix of \mathbf{R}_o , by:

$$\omega_O = \tan^{-1}(C_{2Ou}/C_{1Ou}) \quad (13)$$

where ω_O is the angle formed by the semi-major axis of the class uncertainty ellipse with respect to the horizontal axis (C_1), and C_{1Ou} and C_{2Ou} are the eigenvector components corresponding to the highest eigenvalue (λ_{Ou}) of the covariance matrix. The semi-major and semi-minor uncertainty ellipse axes, u_O and l_O respectively, which represent the maximum and minimum variance, are given by:

$$u_O = \sqrt{\lambda_{Ou}}, \quad l_O = \sqrt{\lambda_{Ol}} \quad (14)$$

where λ_{Ol} is the minor eigenvalue of the covariance matrix. From these dispersion invariants, it is possible to obtain the model for the hue dispersion. Therefore, our interest is to obtain a correspondence between the hue dispersions in the HS plane by means of the information offered by the angular dispersion in the C_1C_2 plane. Knowing that the variation of the angular dispersion in the C_1C_2 plane corresponds with the variation of the hue dispersion in the HS plane, and since the C_1C_2 plane is a Cartesian plane, the problem is posed in the polar coordinates, taking these two considerations into account:

(a) As previously indicated, in the C_1C_2 plane, the colour injections only produce translations of the classes and, therefore, variations of their mean vector modules ($\|\mathbf{c}_{iO}\|$, $\|\mathbf{c}_{iB}\|$). This causes the modification of the angular dispersions of both classes, because they depend on $C_i = \|\mathbf{c}_{iO}\| = \|\mathbf{c}_{iB}\|$ (distance between the dispersion centre and the origin of the C_1C_2 plane). These effects of the hue dispersion modification have been observed when performing translations of a class by adding Gaussian noise in the RGB space [22,31]. In conclusion, the angular dispersion increases when the magnitude of its respective mean vector decreases due to the increment of the separation angle θ_i , according to:

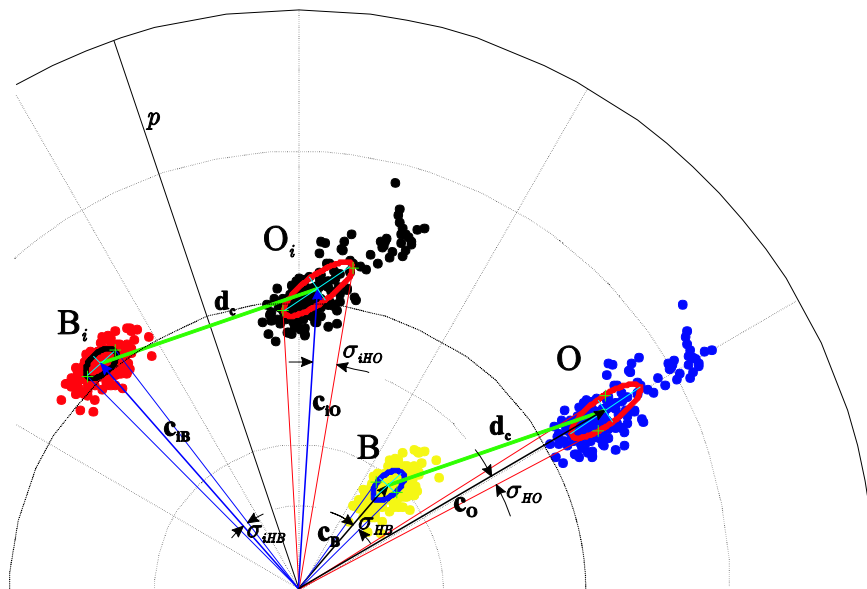
$$C_i = \|\mathbf{d}_c\|/2\sin(\theta_i/2) \quad (15)$$

(b) The geometric forms of the class distributions are not predetermined, but they can vary since they depend on the samples randomly taken from the object and the background. The colour injections produce class translations in the C_1C_2 plane, implying that from the point of view of the HS plane, the dispersion also depends on the geometric form of the classes. The reason is that, for different translations of a class, different orientations between the axis of maximum and minimum dispersion (represented by their uncertainty ellipse in a C_1C_2 plane) with respect to the orientation of their mean vectors (\mathbf{c}_{iO} or \mathbf{c}_{iB}) are generated. Therefore, independently of the class mean vector module, a distance d_a exists that contributes to the angular deviation. This distance d_a depends only on the geometric form and orientation of the dispersion after each translation. Then, d_a , in this case for the O class, will depend on the values of ω_O , u_O and l_O given by (13) and (14). This d_a can be approximated by means of the distance between the centre of the uncertainty ellipse and the intersection point between two right lines: one is the tangent line to the ellipse which at the same time passes through the origin of plane, and the other line is perpendicular to the previous one and it crosses the centre of the ellipse. With d_a and (15) the angular deviation can be approximated by:

$$\sigma_{iH} = \sin^{-1}(d_a/C_i) \quad (16)$$

As an example, in Figure 5 we depict the object class (O) before a translation, for the addition of a vector \mathbf{i}_c in this C_1C_2 plane, or, for the injection of a vector \mathbf{i}_r , directly to the classes in RGB components. Over the object class, its respective uncertainty ellipse is shown.

Figure 5. Uncertainty ellipses of the classes O and B in the C_1C_2 plane: before the colour injection: O (blue) and B (yellow) and after the colour injection: O (black) and B (red). Geometric approximation of the hue deviations of the classes, as a function of the ellipse locations. The different alignments of the axes of the ellipse with respect to the direction of the mean vectors of each class are shown.

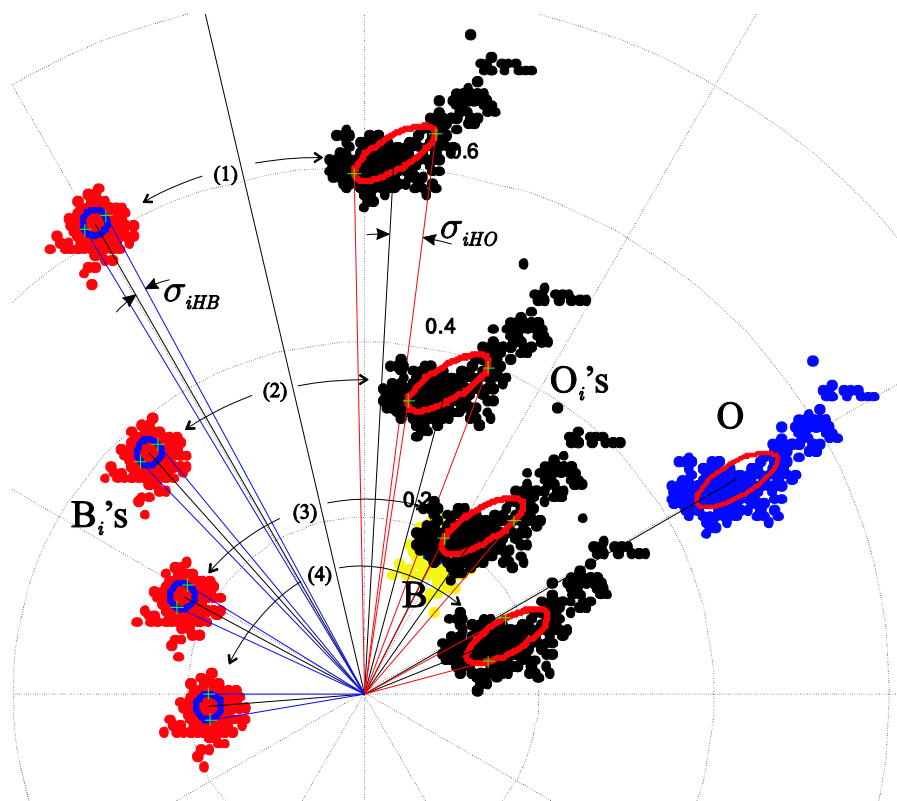


In Figure 5, we can observe that the semi-major axis of the ellipse is relatively aligned to the mean vector \mathbf{c}_O of the class, causing the perception of the minimum angular dispersion of that class. It can also be observed that the module of this mean vector, \mathbf{c}_O , before the injection is greater than the module of the vector after it has been injected, \mathbf{c}_{iO} , which, therefore, is also perceived as a minor angular dispersion by this effect. We may conclude then, that the initial location of this class in the HS plane represents a very favourable case, since the angular deviation before the colour injection is small.

Nevertheless, for the background class (B) before the colour injection, certain alignment between the mean vector \mathbf{c}_B and the axis of greater dispersion of this class can also be observed, implying a reduced angular deviation. However, the problem is that the module of the vector \mathbf{c}_B is reduced and, therefore, the angular deviation increases. In this case, it can be observed in Figure 5 that after the colour injection, the angular dispersion of the class B_i is smaller, since the module \mathbf{c}_{iB} is greater.

Figure 6 depicts another example, with the different class locations after four colour injections. The modifications of the angular deviations σ_{iHO} and σ_{iHB} of the object and background classes as a function of the orientation of their respective uncertainty ellipses and the modules of their respective mean vectors can be observed.

Figure 6. Location of the classes for 4 different separation angles (θ_i) in the polar HS plane: (1) $\theta_i = 33^\circ$; (2) $\theta_i = 57^\circ$; (3) $\theta_i = 97^\circ$; (4) $\theta_i = 163^\circ$. The original classes (O and B) and the injected classes are shown for the 4 colour injections (O_i's and B_i's).



5.4.2. Saturation dispersion

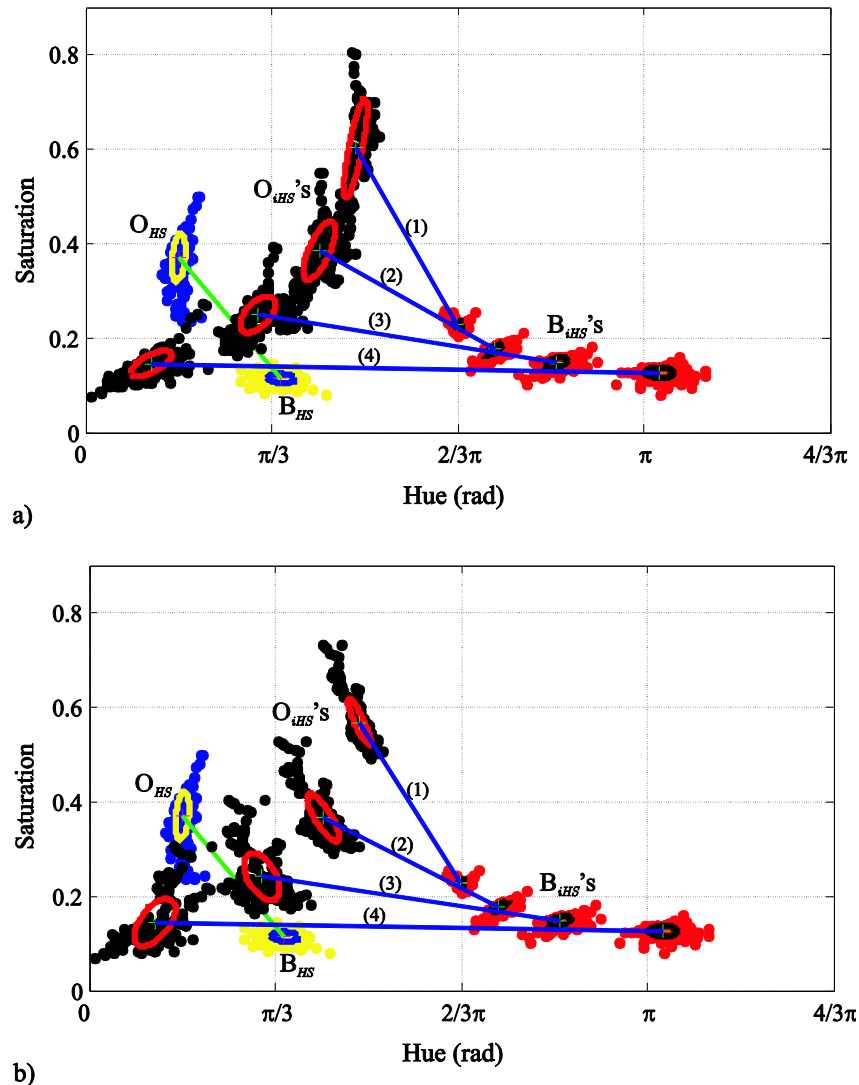
The dispersion of the saturation component is not directly affected by the class translations (due to the colour injections) in the C_1C_2 plane, if all the class vectors have the same intensity. The reason is that the saturation is a linear function of the C_1 and C_2 components. The expression of the saturation for lobe 1 of $f(H)$ is (Equation B.13) (see Appendix B):

$$S = \frac{C_1}{3I} + \frac{C_2}{\sqrt{3}I} \quad (17)$$

This characteristic of linearity in the C_1C_2 plane makes the deviation of the saturation (σ_S) constant, since the distance between vectors in the C_1C_2 plane remains constant, independent of the colour injection. Nevertheless, in the HS plane σ_S will be different for each lobe of $f(H)$ but will stay constant within each lobe. Evidently, if the class vectors have different intensity, the dispersion of the saturation will not be constant for each location, not even within the lobes (there is a greater variation of σ_S when the dispersion of the intensity component is greater).

Figure 7 illustrates how the hue and saturation dispersions are modified for the four colour injections of Figure 6.

Figure 7. Classes projected in an *HS* Cartesian plane, corresponding with the example of the Figure 6, where the dispersion variation of both classes with the colour injections is observed. (a) A variable *S* deviation is shown because the classes keep their original intensities: $\sigma_{IO} \neq \sigma_{IB} \neq 0$, (b) A constant *S* deviation is shown because each class intensity is equal to its respective intensity mean, *i.e.*, $\sigma_{IO} = \sigma_{IB} = 0$.



In this case, the locations of both classes are projected in an *HS* Cartesian plane. The magnitude of the *H* and *S* deviation can be appreciated by means of the projections of the corresponding uncertainty ellipses of both classes in the axes *H* and *S*. In Figure 7a we can observe a diminution of the *H* deviation and the increase of the *S* deviation when the angle θ_i between the classes decreases, because the modules of the mean vectors of both classes increase. It can also be observed how the *S* deviation of the O_i class, is modified more than the deviation of B_i , because the *I* dispersion of O_i is greater. Figure 7b shows the same example as Figure 7a, but with the intensities of the class vectors equal to its intensity mean, *i.e.*, $I_{O1} = I_{O2} = \dots = I_{ON} = I_O$, and $I_{B1} = I_{B2} = \dots = I_{BM} = I_B$, implying that $\sigma_{IO} = \sigma_{IB} = 0$. Then, we can see how the *S* deviation of O_i , remains constant for each colour injection.

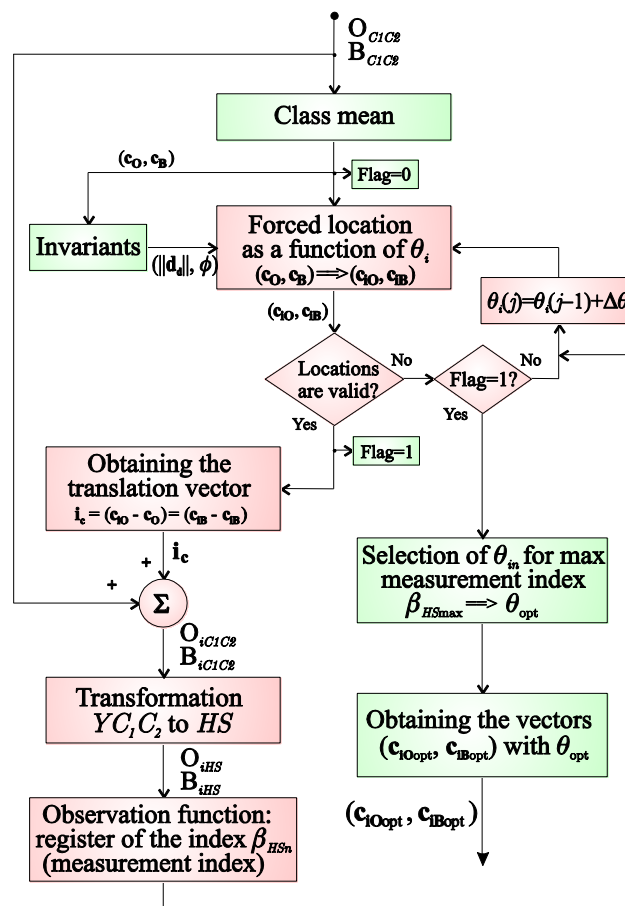
However, our interest in this paragraph is to understand how the colour injections affect the saturation dispersion. This is the reason why in our algorithm the S deviations of both classes are obtained considering their original intensities.

6. Algorithm for the Optimal Location of the Mean Vectors of Both Classes in C_1C_2 Plane

This section presents the strategy used to obtain, in the C_1C_2 plane, the mean vectors that maximize the separation between the classes in the HS plane. This section constitutes the main stage in Figure 1: “Optimal location of the mean vectors of the classes in the C_1C_2 plane”. As shown in Figure 1, for each captured image, an algorithm to obtain the optimal location in the C_1C_2 plane of the mean vectors of both classes (object and background) is executed. From these optimal vectors, $\mathbf{c}_{iO_{opt}}$ and $\mathbf{c}_{iB_{opt}}$, and once the transformation to the RGB space is performed ($\mathbf{r}_{iO_{opt}}$, $\mathbf{r}_{iB_{opt}}$), the optimal vector to inject, \mathbf{i}_r , is obtained using (3).

The proposal to obtain these optimal vectors, $\mathbf{c}_{iO_{opt}}$ and $\mathbf{c}_{iB_{opt}}$, consists of different phases, and its general block diagram is depicted in Figure 8.

Figure 8. Functional diagram to obtain the optimal location of the mean vectors in the C_1C_2 plane: $\mathbf{c}_{iO_{opt}}$ and $\mathbf{c}_{iB_{opt}}$.



As can be observed, the proposal includes an iterative algorithm to obtain a set of locations for the mean vectors of the classes (\mathbf{c}_{iO} and \mathbf{c}_{iB}) in the C_1C_2 plane. The location of each vector will be parameterized by the angle formed between both vectors, θ_i . Therefore, we try to obtain a set of

θ_{in} ($\theta_{i1}, \theta_{i2}, \dots$). Each of them will have associated a measurement index of separation between classes that we will identify by β_{HSn} ($\beta_{HS1}, \beta_{HS2}, \dots$). From the function $\beta_{HSn} = f(\theta_{in})$, the value of θ_{in} that produces the maximum separation between classes is obtained, θ_{in} optimal: θ_{opt} .

The process begins obtaining the mean vectors of each class in C_1C_2 plane. These mean vectors will be,

$$\mathbf{c}_O = E\{\mathbf{c}_{O1}, \mathbf{c}_{O2}, \dots, \mathbf{c}_{ON}\}, \quad \mathbf{c}_B = E\{\mathbf{c}_{B1}, \mathbf{c}_{B2}, \dots, \mathbf{c}_{BM}\} \quad (18)$$

From the vectors \mathbf{c}_O and \mathbf{c}_B , its difference vector, \mathbf{d}_c , is obtained. As previously indicated, the magnitude, $\|\mathbf{d}_c\|$, and angle, ϕ , of the vector \mathbf{d}_c are invariant against translations in the C_1C_2 plane. Their values are given by equation (19):

$$\|\mathbf{d}_c\| = (d_{C1}^2 + d_{C2}^2)^{1/2}, \quad \phi = \begin{cases} \cos^{-1}(d_{C1}/\|\mathbf{d}_c\|); & d_{C2} \geq 0 \\ 2\pi - \cos^{-1}(d_{C1}/\|\mathbf{d}_c\|); & d_{C2} < 0 \end{cases} \quad (19)$$

where $d_{C1} = C_{1O} - C_{1B}$ and $d_{C2} = C_{2O} - C_{2B}$, such that (C_{1O}, C_{2O}) and (C_{1B}, C_{2B}) are the components of the vectors \mathbf{c}_O and \mathbf{c}_B , respectively.

The iterative process consists of the following six steps:

(a) Forced location of the mean vectors in the C_1C_2 plane

The original vectors \mathbf{c}_O and \mathbf{c}_B are relocated (forced) in the C_1C_2 plane using the invariants ($\|\mathbf{d}_c\|, \phi$), obtaining the new vectors (\mathbf{c}_{iO} and \mathbf{c}_{iB}). Each location of the vectors (\mathbf{c}_{iO} and \mathbf{c}_{iB}) should fulfil the following geometric restriction: the straight line that passes through the origin of the C_1C_2 plane and is perpendicular to the vector \mathbf{d}_c should intersect this last one in $\|\mathbf{d}_c\|/2$. As previously indicated, this implies that:

$$C_i = \|\mathbf{c}_{iO}\| = \|\mathbf{c}_{iB}\| = \|\mathbf{d}_c\|/(2\sin(\theta_i/2)) \quad (20)$$

This θ_i is the parameter to vary in order to obtain the different locations of the vectors \mathbf{C}_{iO} and \mathbf{C}_{iB} , and, therefore, of the locations of the classes in the C_1C_2 plane.

The Cartesian components of these vectors (Figure 4), particularized for the vector \mathbf{C}_{iO} , are given by:

$$C_{i1O} = C_i \cos(H_{iO}), \quad C_{i2O} = C_i \sin(H_{iO}) \quad (21)$$

where h_{iO} is the angle of the vector that can be expressed by:

$$H_{iO} = \pi/2 + \phi - \theta_i/2 \quad (22)$$

Similar expressions can be obtained for \mathbf{C}_{iB} .

The iterative algorithm is initialized with an θ_i equal to θ (θ is the angle formed by the vectors \mathbf{c}_O and \mathbf{c}_B). In each iteration (j) of the algorithm, the value of θ_i is increased: $\theta_i(j) = \theta_i(j-1) + \Delta\theta$.

We should also take into account that θ_i represents the hue distance between the mean vectors (\mathbf{H}_{iO} and \mathbf{H}_{iB}) of the classes in the HS plane. This indicates that a direct relationship exists between the class translations in the C_1C_2 plane and the hue separation distance between the class means in the HS plane.

(b) Verification of the validity for the locations of the \mathbf{C}_{iO} and \mathbf{C}_{iB} vectors

For each increase of θ_i , the validity of the locations of the vectors \mathbf{C}_{IO} and \mathbf{C}_{IB} is verified. In case they are valid, the value of θ_i is included in the set θ_{in} . The validity of \mathbf{C}_{IO} and \mathbf{C}_{IB} (validity of θ_i) is tested by checking if the components of the corresponding vectors in *RGB* space (\mathbf{R}_{IO} , \mathbf{R}_{IB}) fulfil the limitations imposed by this space, *i.e.*, the values are in the range $[0, 1]$, because they are normalized with respect to 255.

(c) Calculation of the class translation vector in the C_1C_2 plane

The translation vector \mathbf{i}_c is obtained for each value of θ_{in} . This vector \mathbf{i}_c is responsible for the class translations from its original position to the forced location defined by θ_{in} . The translation vector \mathbf{i}_c in the C_1C_2 plane corresponds to the vector to inject \mathbf{i}_r in the *RGB* space. This translation vector can be calculated from any of the following expressions:

$$\mathbf{i}_c = \mathbf{c}_{IO} - \mathbf{c}_O, \quad \mathbf{i}_c = \mathbf{c}_{IB} - \mathbf{c}_B \quad (23)$$

(d) Translation of the classes in the C_1C_2 plane

The class translations in the C_1C_2 plane are performed with the value of \mathbf{i}_c that has been calculated. Therefore, each vector \mathbf{c} belonging to the object and background classes are increased by \mathbf{i}_c :

$$\mathbf{O}_{iC1C2} = \{\mathbf{c}_{Or} + \mathbf{i}_c\}; \quad r = 1, 2, \dots, N, \quad \mathbf{B}_{iC1C2} = \{\mathbf{c}_{Bq} + \mathbf{i}_c\}; \quad q = 1, 2, \dots, M \quad (24)$$

(e) Class transformation from the C_1C_2 plane to the *HS*

The classes in the *HS* plane (\mathbf{o}_{HS} and \mathbf{b}_{HS}) are obtained from the translated classes \mathbf{O}_{iC1C2} and \mathbf{B}_{iC1C2} , using (Equation B.12), (Equation B.13) and (Equation B.14).

(f) Observation function: calculation of the class separation measurement index (β_{HSn}) in the *HS* plane

As the class separation observation function, a normalized measurement index has been defined (β_{HS}) from the *FR* described in (5). It has been normalized to obtain $\beta_{HSn} = 1$ when the class separation is maximum. To obtain the β_{HSn} corresponding with each θ_{in} , we consider the mean and the dispersion of *H* and *S* of the classes, according to (6). Therefore, two class separation measurement indexes as a function of θ_{in} have been defined, one for each component:

$$\beta_{Hn} = (\sqrt{FR_H} - 1) / \sqrt{FR_H}, \quad \beta_{Sn} = (\sqrt{FR_S} - 1) / \sqrt{FR_S} \quad (25)$$

The final class separation measurement index is given by:

$$\beta_{HSn} = k_h \beta_{Hn} + (1 - k_h) \beta_{Sn} \quad (26)$$

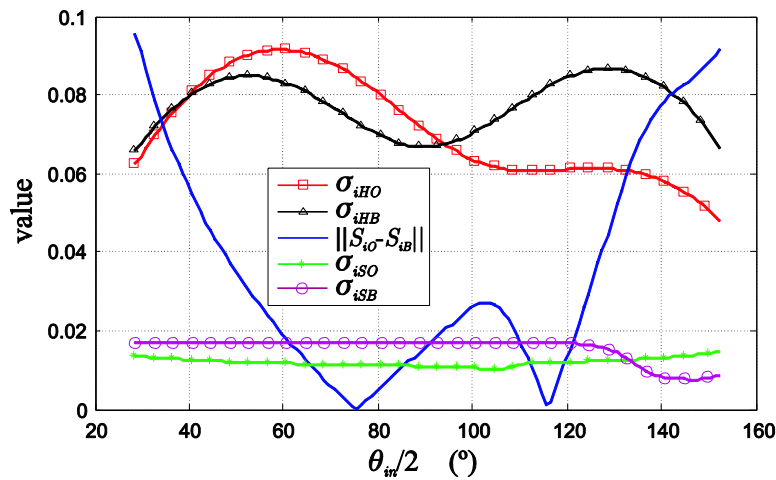
where k_h is a weighting factor between β_{Hn} and β_{Sn} . The value of $k_h \in [0, 1]$ is chosen depending on the prominence we want to give *H* or *S* in the segmentation process. Taking into account that *H* has a greater discriminating power than *S*, $k_h > 1/2$ should be fixed.

This iterative process is repeated until the first non valid value of θ_{in} is generated, and the pairs (β_{HSn} , θ_{in}) are registered to obtain the function $\beta_{HSn} = f(\theta_{in})$ afterwards.

Once the set of pairs (β_{HSn} , θ_{in}) is obtained, the θ_{in} that produces the maximum class separation measurement index is selected. A cubic interpolation is performed around that local maximum to

obtain the maximum of the interpolation index, β_{HSmax} , and its associated angle, θ_{opt} . Finally, with this θ_{opt} , the \mathbf{c}_{iOopt} and \mathbf{c}_{iBopt} vectors are obtained using (20), (21) and (22).

Figure 9. Hue and saturation deviation of both classes as a function of $\theta_{in}/2$. The difference between the saturation means of both classes is shown too.



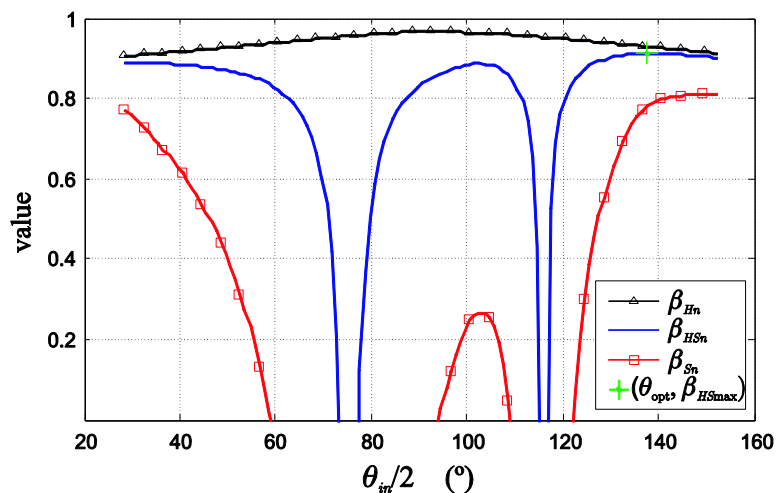
As an example, Figure 9 shows the variation curves, as a function of $\theta_{in}/2$, of the statistical data: deviations of hue (σ_{iHO} and σ_{iHB}), deviation of saturation (σ_{iSO} and σ_{iSB}), and difference between the saturation means $\|S_{iO} - S_{iB}\|$ needed to obtain the different class separation measurement indexes (25).

7. Calculation of the Optimal Colour Vector to Add and the Effects that it Produces on the Images

The calculation of the optimal colour vector to add, \mathbf{i}_r , is the goal of our proposal, because this vector changes the colours of the captured image in a suitable manner, so that the classes separate and, therefore, the object class can be more easily segmented.

Figure 10 shows the values of β_{Hn} , β_{Sn} and β_{HSn} obtained from the values of the statistical data depicted in Figure 9. The values of $(\theta_{opt}, \beta_{HSmax})$ obtained by interpolation are also shown.

Figure 10. Measurement indexes as a function of $\theta_{in}/2$.



As depicted in the block diagram of Figure 1, once the vectors $\mathbf{c}_{iO_{opt}}$ and $\mathbf{c}_{iB_{opt}}$, that represent the optimal location of the classes in the HS plane, are obtained, the vectors $\mathbf{r}_{iO_{opt}}$ and $\mathbf{r}_{iB_{opt}}$ can be calculated. Thus, for instance, for the object class, O: if $C_{1O_{opt}}$ and $C_{2O_{opt}}$ are the C_1 and C_2 components of the vector $\mathbf{c}_{iO_{opt}}$ respectively, the vector $\mathbf{r}_{iO_{opt}}$ in RGB space is obtained by:

$$\mathbf{r}_{iO_{opt}} = \mathbf{Q}^{-1} [Y_{iO} \quad C_{1O_{opt}} \quad C_{2O_{opt}}]^T \quad (27)$$

where \mathbf{Q} is the transformation matrix (Equation A.2) and Y_{iO} is the intensity mean of the object class translated in the C_1C_2 plane. The \mathbf{i}_r vector is obtained with this $\mathbf{r}_{iO_{opt}}$ applying (3). Considering that the colour injection can be made without modifying the mean intensity of the class after the injection of \mathbf{i}_r , $Y_{iO} = I_O$ holds. Although it is possible to modify the saturation mean varying the intensity mean, in this case, we want the saturation mean to be only affected by the $f(H)$ value and the Chroma component (C). Therefore, the vector to inject, \mathbf{i}_r , should have zero mean ($E\{\mathbf{i}_r\} = 0$). The fact that $E\{\mathbf{i}_r\} = 0$ implies that the intensity mean of the original image (I) and the injected one (I_i) are equal.

The effect of injecting the vector \mathbf{i}_r to the original image in the new image, I_i , is a greater concentration of the pixel colours around the mean colour of each one of the two classes. That is, the colour injection contributes to the histogram equalization of the captured image in the HS plane. This equalization has a concentration effect on each class, and, therefore, the injection of \mathbf{i}_r contributes to approaching the class distributions to a Gaussian shape. As an example, Figure 11 shows the 2D histograms of image I (Figure 11a) and of the coloured image resulting from the colour injection I_i , (Figure 11b), for a particular case (Figure 12 images).

Figure 11. Example of 2D histograms in the HS plane for the original image I and injected image I_i . (a) Original image histogram, (b) injected image histogram. The class redistribution in the injected image when compared with the original image can be observed. In the image I_i , an isolation of the main scene classes (O and B) can be visually appreciated, as well as a shape closer to the Gaussian form.

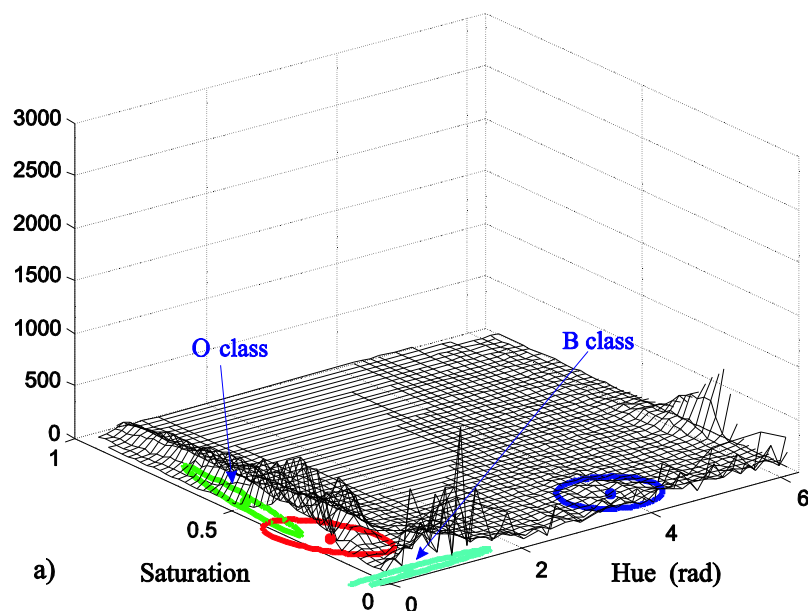
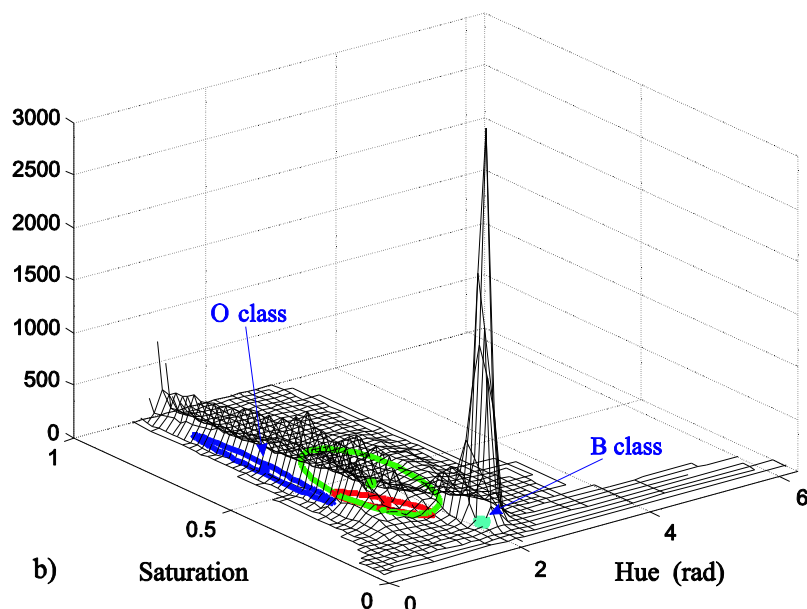
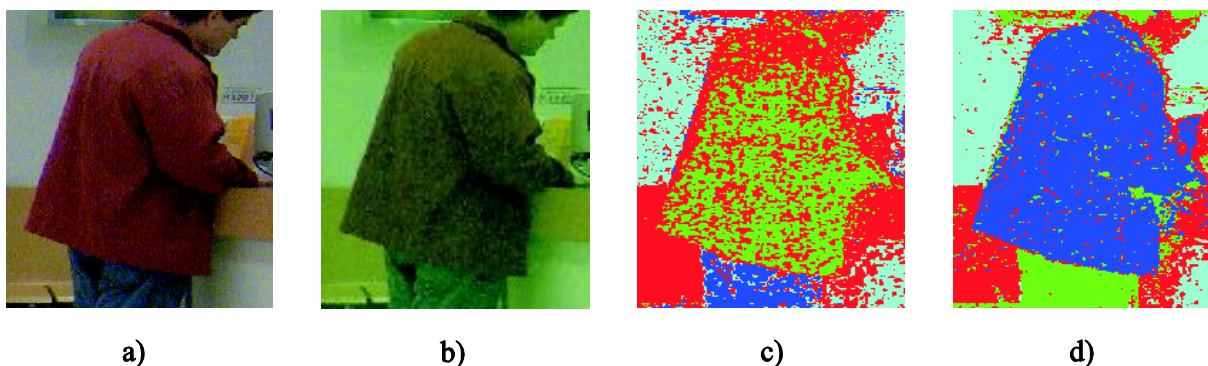


Figure 11. Cont.



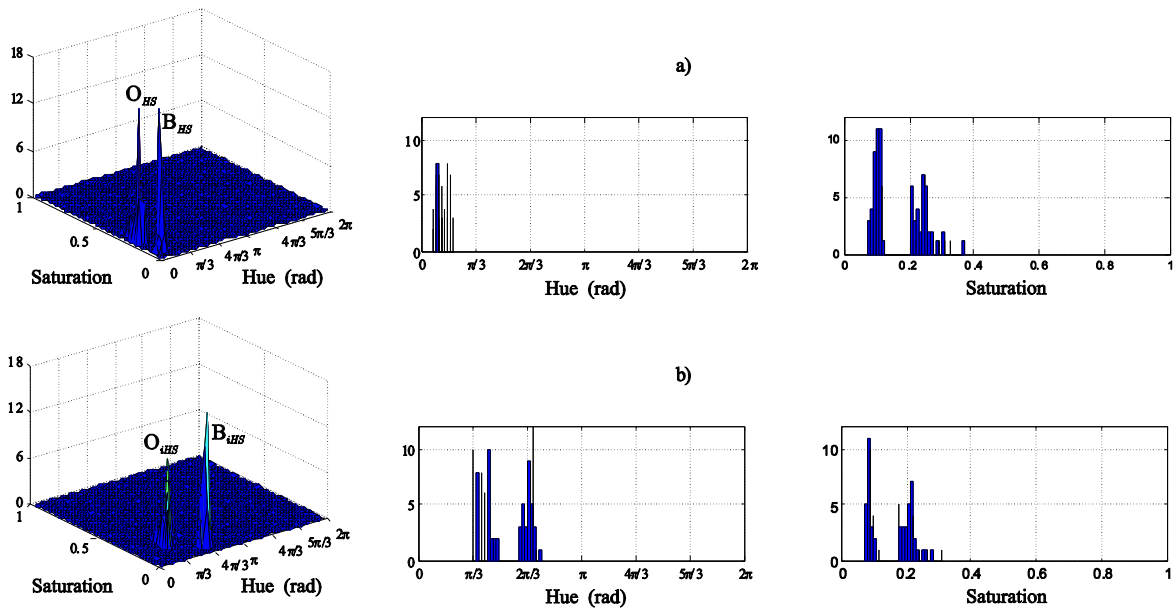
In these figures (11a and 11b), the equalization of the histogram produced by the effect of the colour injection can be clearly observed. The segmentation of both images is shown in Figure 12c and 12d respectively. In this example, $K_{opt} = 4$, the O class corresponds with the jacket and the B class with the wall.

Figure 12. Reference images for the example of Figure 11. Example of original and injected images segmentations: (a) original image, I , (b) injected image, I_i , (c) original image segmentation according to the projected classes in Figure 11a and (d) injected image segmentation according to the projected classes in Figure 11b.



The effect of the class separation between O and B classes can also be directly seen, analyzing the class locations before and after the colour injection in their histograms. Figure 13 shows the histograms corresponding to the sets O_{HS} and B_{HS} in (a), and the sets O_{iHS} and B_{iHS} in (b). A remarkable increase in the hue component separation can be observed in the histograms of Figure 13b due to the colour injection.

Figure 13. Histograms of the O_{HS} and B_{HS} sets: (a) before the colour injection, (b) after the colour injection.

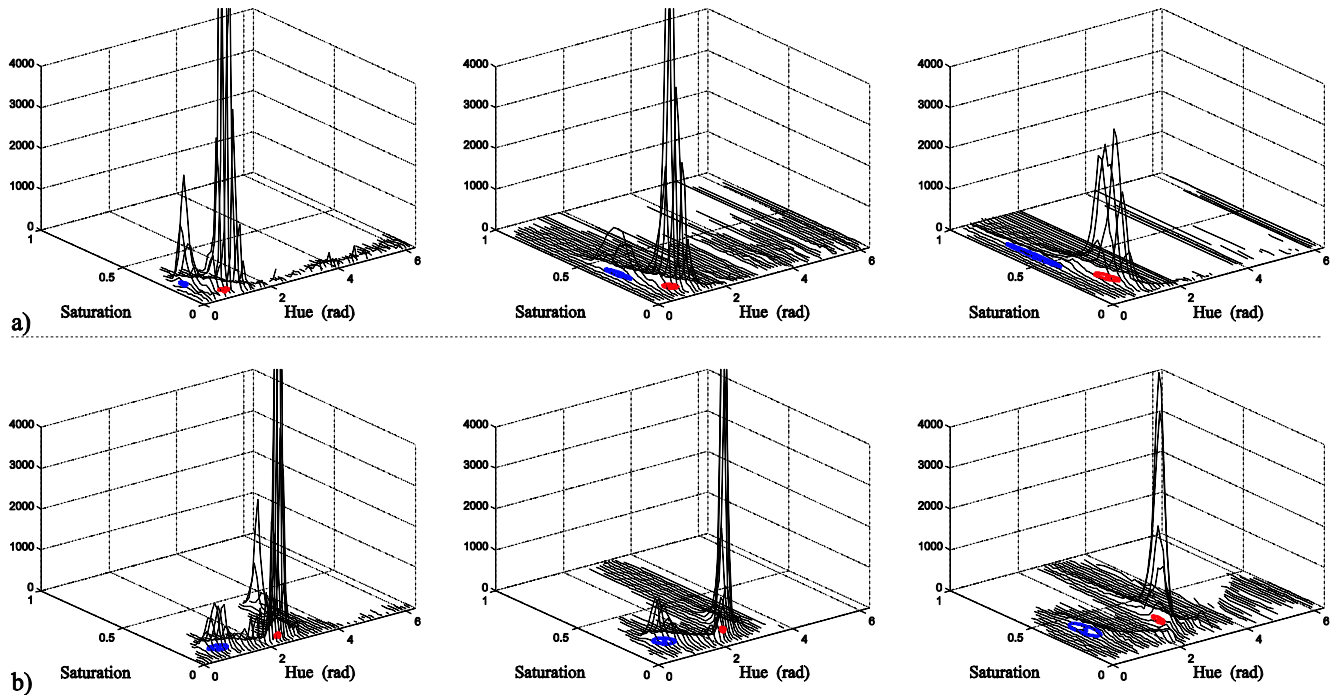


The rest of the image classes different from B, $B_x \neq B$; $x = 1, 2 \dots K_{opt} - 2$, are also affected by the effects of the colour injection. In this sense, as the class selected as B is the closest to the class O that also has a high probability to be the image background (fulfils equation (7)), when the separation between the classes O and B increases, the classes B_x also increase their separation with the class O. However, the colour injection decreases the separation between the class O and those classes B_x (B'_x ; $x = 1, 2 \dots$) that are closer than B to the class O but that were not selected as class B because they had a lower a priori probability. The consequence is that these classes (B'_x) can be considered as class O, producing false positives in the object pixel classification.

Another effect of the colour injection is the automatic compensation of the illumination changes. That is, due to the equalization and the separation of the classes O and B in the injected image, there is a minimization of the problems produced by the illumination changes. The reason is that the main colour component affected by the illumination changes is S , and, as previously explained, our algorithm gives more importance to the separation of the most discriminant component, H . Then, both classes, O and B, always keep a certain separation, independently of the parameter variation of both distributions, and mainly when the mean and variance of the S component vary due to changes in the luminous intensity.

Next, in Figure 14, three histograms are presented, for the original and the injected image. All of them have been obtained with the different mean luminous intensity of the image ($I_m = E\{I\} = E\{I_i\}$): ($I_{m1} = 0.70$, $I_{m2} = 0.45$ and $I_{m3} = 0.21$). The illumination compensation effects mentioned above can be observed in this figure.

Figure 14. Three 2D Histograms for three intensity mean values for each image, I and I_i : $I_{m1} = 0.70$, $I_{m2} = 0.45$ and $I_{m3} = 0.21$. (a) histograms of the image I , (b) histograms of the image I_i . We can observe how the distribution statistics of both classes of the image I_i are less affected by the illumination changes than the ones of image I .



8. Experimental Results

A bank of real images from different scenes has been used in a first phase of the practical tests, in order to evaluate the effectiveness of the proposed method. Here, a Gaussian classifier has been used as a segmentation technique, supposing a unimodal Gaussian model for the respective object and background class-conditional *pdfs*, *i.e.*, $p(\mathbf{h}_i|O_i)$ and $p(\mathbf{h}_i|B_i)$. Thus, $p(\mathbf{h}_i|O_i) = g(\mathbf{h}_i; \mathbf{h}_{iO}, \Sigma_{iO})$ is given by:

$$p(\mathbf{h}_i|O_i) = \frac{1}{2\pi|\Sigma_{iO}|^{1/2}} \exp\left\{-\frac{1}{2}d_m\right\}; \quad d_m = (\mathbf{h}_i - \mathbf{h}_{iO})^T \Sigma_{iO}^{-1} (\mathbf{h}_i - \mathbf{h}_{iO}) \quad (28)$$

where \mathbf{h}_i represents each pixel of the image I_i , and Σ_{iO} is the covariance matrix of the injected object class in the *HS* plane. The segmentation is performed by thresholding the *pdf* (28) with a T_h value. This threshold is obtained knowing that we want to segment the class O_i taking the background class B_i as reference, so, T_h corresponds to the value of *pdf* (28) when: $d_m = \frac{1}{2}(\mathbf{h}_{iB} - \mathbf{h}_{iO})^T (\Sigma_{iO} + \Sigma_{iB})^{-1} (\mathbf{h}_{iB} - \mathbf{h}_{iO}) = \frac{1}{2}\text{tr}(M_w^{-1}M_b)$. Therefore, T_h is given by:

$$T_h = \frac{1}{2\pi|\Sigma_{iO}|^{1/2}} \exp\left\{-\frac{1}{4}\text{tr}(M_w^{-1}M_b)\right\} \quad (29)$$

The problems derived from the cyclical nature of the hue in the segmentations have been solved via software, using the convention introduced by Zhang and Wang [8].

In the evaluation, the same number of samples (seeds) for the object class (O) and the background (B) has been taken, $M = N$, in order to ensure that the difference between their statistical data is for

intrinsic reasons, and not for differences in the sample space dimension. In the tests, the following data have been used: samples number: $M = N = 50$. Other tests with a higher number of seeds ($M = N = 100, 200, 400, 800$ and $1,600$) have also been carried out, providing similar qualitative results in all of them, but with an increase in the iterative process computational cost. The increase of θ used in the algorithm shown in Figure 8: $\Delta\theta = 5^\circ$; interpolation interval $\Delta\Theta = \pm 3\Delta\theta$ and the weighting factor k_h in (26) has been experimentally selected for each experiment, always fulfilling $k_h \in [0.75, 0.97]$. In this stage, the experimental results have been quantified by means of the FR defined in (5). Table 1 shows the values of FR for 14 cases of the bank of images used in the tests.

Table 1. FR results for 14 cases in this work.

Case	FR	FR (Injected)	%Increase
1	49.15	112.32	128.53
2	74.67	246.82	230.52
3	11.18	21.84	95.31
4	68.08	1,826.02	2,581.97
5	100.82	214.46	112.71
6	96.27	173.50	80.22
7	209.62	735.81	251.01
8	23.91	63.16	164.15
9	123.15	2,277.07	1,749.00
10	9.49	44.52	369.11
11	126.27	946.48	649.56
12	21.02	197.71	840.32
13	65.13	74.68	14.66
14	1.604	5.57	247.70

Fourteen examples of segmentation can be seen in Figure 15 (figures a, b, c, d, e, f, g, h, i, j, k, l, m and n) that correspond with the 14 cases of FR calculated in Table 1. Four images are shown in each column (from up to down): the upper image is the original one (I), the second one is the coloured image (I_i), the third one, the results of the segmentation of the original image (I segmentation), and the lower one, the results of the segmentation of the injected image (I_i segmentation). The segmented images show the object pixels in green colour. For the figures between Figure 15a and Figure 15m, the object class (O) is the skin, and for Figure 15n, the object class is a jacket.

As can be observed, our proposal to inject a colour vector allows the attainment of remarkable improvements in the segmentation process, even with a segmentation technique as studied and effective as the Gaussian classifier.

Figure 15. Segmentation results for objects in different environments.



As a second phase of the experimental tests, and in order to quantify the improvement in the segmentation of the injected image with respect to the original image, an analysis, pixel by pixel, has been made, comparing with the manually segmented reference images for the 14 cases. The data generated in this analysis, without noise added, are shown in Table 2. The performance of the segmentation has been measured taking into account the classification Correct Detection Rate (*CDR*) and False Detection Rate (*FDR*) and the total Classification Rate (*CR*). *CDR* is the percentage of object pixels correctly classified, *FDR* is the percentage of background pixels incorrectly classified and *CR* is the total percentage of correctly classified pixels. Table 2 also shows the number (*K*) of Gaussians used

by the GMM algorithm, the *FR* obtained by the statistics given by GMM for both classes, as well as the k_h used for each image.

Table 2. Comparative analysis of the segmentations for the 14 example images shown in Figure 15, without noise added.

Case	Reference pixels*	CDR (%)		FDR (%)		CR (%)		Angle (°)		i_r vector			k_h	K	Fisher Ratio	
		I	I_i	I	I_i	I	I_i	θ	θ_i	[R	G	B]			FR	FR _i
a	10,503	91	97	33	33	67	67	25	179	[-24	0	24]	0.97	5	23	39
b	18,340	94	97	22	10	78	90	32	116	[-23	4	18]	0.93	5	16	55
c	5,749	96	89	76	58	24	42	7	282	[-15	-4	19]	0.65	11	6	9
d	13,460	93	97	13	11	87	89	4	258	[-15	1	15]	0.65	5	6	84
e	10,097	84	97	26	10	74	90	23	246	[-26	-1	27]	0.87	5	47	157
f	12,666	97	97	47	24	53	76	56	179	[-4	-8	13]	0.9	7	15	122
g	13,775	98	91	28	13	72	87	18	171	[-31	14	16]	0.93	5	48	1,118
h	12,231	91	96	115	29	-15	71	14	158	[-21	6	15]	0.95	6	5	77
i	9,063	97	95	34	22	66	78	6	126	[-30	20	10]	0.93	5	11	139
j	12,497	89	97	100	55	0	45	20	167	[-13	2	10]	0.85	5	3	17
k	12,512	97	95	22	14	78	86	3	193	[-27	7	20]	0.95	5	19	176
l	23,102	69	94	59	17	41	83	17	23	[-39	44	-6]	0.87	10	5	24
m	20,176	99	96	31	12	69	88	26	56	[-16	15	1]	0.6	5	25	27
n	38,629	67	82	35	19	65	81	6	298	[-13	-19	33]	0.55	4	5	21
Average		90	94	46	23	54	77	18	175	[-21	6	15]	0.83	6	17	148

*Segmented Object Reference Image, I original image, I_i injected image

Table 3 shows the results of the comparison of the same images, but contaminated by additive zero-mean Gaussian noise. As can be seen, results obtained with the colour injection technique for both tests are better than those obtained using only a Gaussian classification.

As a third phase of the tests, an example of image sequence segmentation is presented. In this case, each frame illumination has been modified before the segmentation process, in order to verify the advantage of our proposal against illumination changes. The illumination is applied to each frame in a uniform way. A zero-mean Gaussian noise with standard deviation $n_p = 0.15I_O$ was also added to the pixels of the images. Moreover, a sinusoidal time variation in the luminous intensity has been set up.

With this example, we try to show the improvements in the segmentation phase when the colour injection preprocessing step proposed in this paper is used before the segmentation. In this example, the GMM technique is used as an on-line segmentation technique (the same used in the off-line process). The original image I is identified by I^k for each captured frame in time kT ($k = 1, 2... \text{ and } T = \text{time between consecutive images}$), and its corresponding image after the colour injection, I_i^k , are segmented using the optimal class number obtained as a result of the off-line stage. In this case $K_{opt} = 5$.

For the images I^k and I_i^k , the GMM segmentation process is applied recursively using the a priori probabilities, means and variances obtained in the images I^{k-1} and I_i^{k-1} , respectively. For the segmentation of the image I_i^k , the next steps are also added: (a) we obtain the pixels (seeds) in the

RGB space of the object and background of the image I_i^{k-1} , (b) the vector \mathbf{i}_r is subtracted from them, (c) they are transformed to the HSI space, (d) the truncation process described in Section 4 is applied, and, finally, (e) the sets O_{RGB}^k and B_{RGB}^k are obtained. These steps (a, b, c, d and e) represent the block: “Obtaining seeds: Object (O) and Background (B)”, for recursive segmentation in the block-diagram of the Figure 1.

Table 3. Comparative analysis of the segmentations for the 14 example images shown in Figure 15, contaminated these images by additive zero-mean Gaussian noise.

Case	Reference Pixels*	CDR (%)		FDR (%)		CR (%)		Noise [†] (σ^2)	Angle (°)		\mathbf{i}_r vector			k_h	K	Fisher Ratio	
		I	I_i	I	I_i	I	I_i		θ	θ_i	[R	G	B]			FR	FR _i
a	10,503	51	76	86	52	14	48	0.8×10^{-3}	2	74	[-20	2	18]	0.97	5	3	4
b	18,340	67	84	44	30	56	70	2.0×10^{-3}	35	78	[-24	13	11]	0.93	5	4	10
c	5,749	89	74	140	117	-40	-17	0.5×10^{-3}	2	27	[-15	-4	19]	0.65	7	4	5
d	13,460	62	70	42	38	58	62	1.0×10^{-3}	12	272	[-20	0	21]	0.95	6	4	28
e	10,097	57	85	47	23	53	77	1.5×10^{-3}	40	262	[-31	0	31]	0.95	5	4	17
f	12,666	48	60	70	51	30	49	0.5×10^{-3}	158	202	[-5	-8	13]	0.9	7	3	6
g	13,775	79	84	30	26	70	74	1.5×10^{-3}	10	256	[-30	0	30]	0.93	5	8	16
h	12,231	70	80	64	62	36	38	0.8×10^{-3}	1	94	[-26	14	12]	0.95	8	8	17
i	9,063	77	92	42	25	58	75	1.5×10^{-3}	128	265	[-27	-5	31]	0.93	5	5	13
j	12,497	64	75	115	74	-15	26	1.0×10^{-3}	14	344	[-13	2	10]	0.85	10	3	5
k	12,512	89	89	33	21	67	79	1.0×10^{-3}	1	291	[-20	-8	27]	0.9	6	5	13
l	23,102	80	81	90	29	10	71	0.5×10^{-3}	12	14	[-49	75	-26]	0.87	7	5	7
m	20,176	62	76	52	38	48	62	1.3×10^{-3}	19	20	[-9	26	-17]	0.88	5	2	6
n	38,629	49	90	72	40	28	60	1.0×10^{-3}	11	43	[-23	34	-11]	0.6	4	4	17
Average		67	80	66	45	34	55	1.0×10^{-3}	32	160	[-22	10	12]	0.88	6	4	12

*Segmented Object Reference Image, I original image, I_i injected image, [†]Additive Gaussian noise: $N(0, \sigma^2)$

In image sequence segmentation, as this example, the iterative process (seen in Section 6) has been slightly modified in order to reduce the processing time and to increase the stability of the colour injection in time. The first modification, is to use θ_{opt}^{k-1} (θ_{opt} of the previous frame) as a starting point to obtain θ_{opt}^k , thus reducing the search interval to: $[\theta_{opt}^{k-1} - \theta_f, \theta_{opt}^{k-1} + \theta_f]$. In the example of Figure 16, we have fixed experimentally $\theta_f = 12^\circ$, $\Delta\theta = 1^\circ$ and $k_h = 0.91$. The second modification in the iterative process is that the optimal colour vector to inject \mathbf{i}_r^k is obtained recursively, using for the calculation of θ_{opt}^k the following expression: $\theta_{opt}^k = k_t \theta_{opt}^k + (1 - k_t) \theta_{opt}^{k-1}$, where $k_t \in [1, 0]$ is the constant fixed to obtain a proper smoothing of the evolution of the different parameters involved in the colour injection. k_t has been fixed experimentally to 0.1.

The GMM technique is used in these tests, mainly to obtain a better adjustment of the K_{opt} Gaussians in each frame, and, therefore, to obtain the maximum quality in the object segmentation. Then, a reliable comparison of the segmentation quality between the segmentation of the images I and I_i in the time can be carried out, and the compensation effects in the segmentation against illumination changes, applying the colour injection or not, can be verified. However, as it is known, this technique may have a relatively high computational cost due to the convergence iterations of the EM algorithm,

so, its use in video segmentation is sometimes limited. For the consecutive segmentation of an image sequence in real time, our proposal in this work is to track recursively the parameters that define each Gaussian: O and B , using the optimal estimation provided by the Kalman filter, tracking technique widely studied in the image processing field.

Figure 16 shows the results of the segmentation of the images I^k and I_i^k of the example sequence, for the frames captured in $k = 21, 42, 63, 84, 105, 126, 147, 168, 189$ and 210 . The respective mean intensities of these frames are: $I_m^{21} = 0.530$, $I_m^{42} = 0.613$, $I_m^{63} = 0.672$, $I_m^{84} = 0.698$, $I_m^{105} = 0.658$, $I_m^{126} = 0.572$, $I_m^{147} = 0.510$, $I_m^{168} = 0.403$, $I_m^{189} = 0.307$ and $I_m^{210} = 0.224$.

Figure 16. Segmentation results for 10 frames of an image sequence of a person generating sign language with big temporal illumination changes.

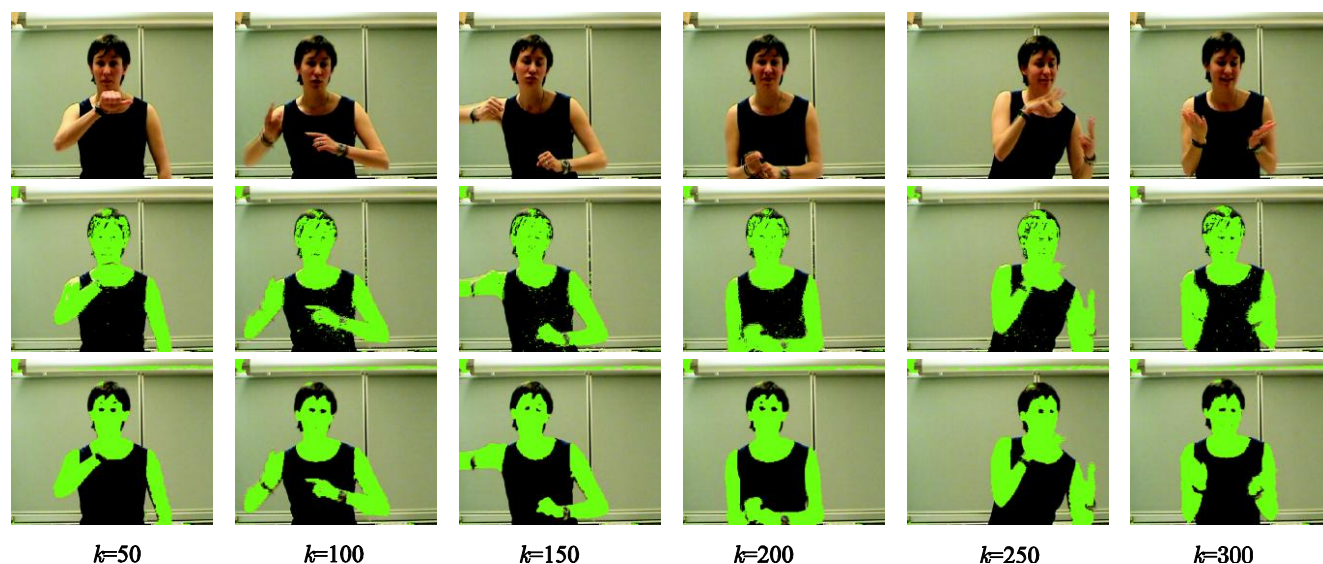


However, if the variation of the parameters of the different classes of the scene in the image sequence is very small, that is, when the scene is relatively uniform in the time with small illumination changes, the colour injection can be carried out applying the same colour vector \mathbf{i}_r to each frame in the time kT , with no need to recalculate it. This is possible due to the illumination compensation effect previously mentioned in Section 7. In this sense, the segmentation can be carried out keeping the parameters of both Gaussians as fixed values in all the sequence. Then, the computational cost is noticeably reduced.

Figure 17 depicts the results of the segmentation of the images I^k and I_i^k corresponding to the instants: $k = 50, 100, 150, 200, 250$ and 300 of the previous example image sequence, but, this time, without recalculating the colour vector \mathbf{i}_r and with fixed parameters for both Gaussians. The objective of this example is to appreciate the improvement in the segmentation of the sequence of colour injected images, although the same colour vector \mathbf{i}_r is used in the injection.

In Figure 17, the upper row depicts the I^k images, the central row shows the results of the segmentation without colour injection (I^k segmentation), and the lower one contains the results of the segmentations after the colour injection proposed in this work (I_1^k segmentation). The segmented images show the object pixels in green colour. The segmentation process used in this phase has been used in the first stage of the experimental tests.

Figure 17. Segmentation results of 6 frames of an image sequence of a person generating sign language with small temporal illumination changes.



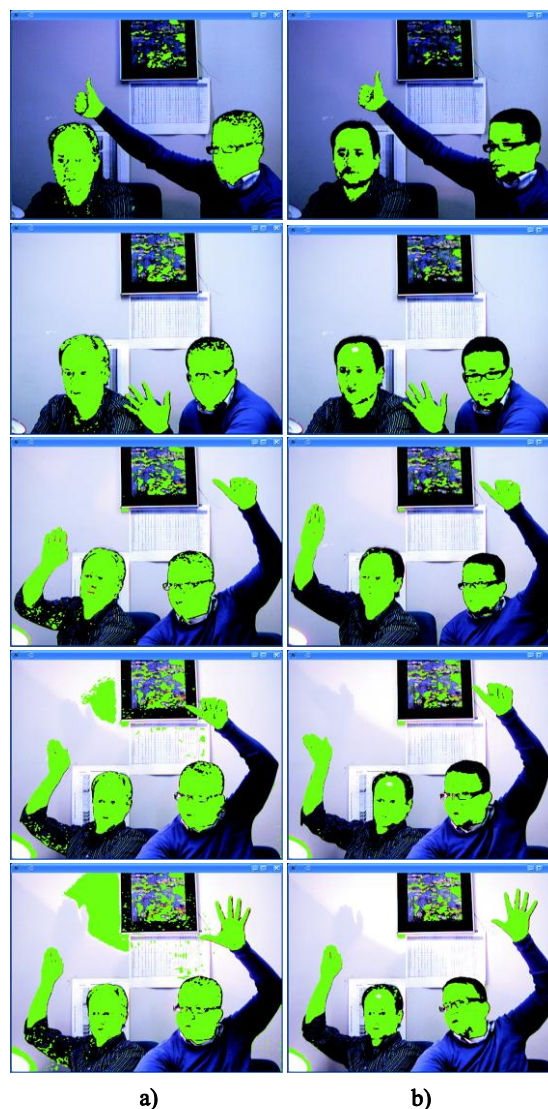
As a reference, the average execution time (T_p) in Matlab of the on-line process for different $M = N$ values is approximately: $T_p = 74.9$ ms. for $N = 50$, $T_p = 80.0$ ms. for $N = 100$, $T_p = 85.4$ ms. for $N = 200$, $T_p = 95.9$ ms. for $N = 400$, $T_p = 117.3$ ms. for $N = 800$ and $T_p = 160.4$ ms. for $N = 1,600$. The tests have been made with the following configuration: $\theta_f = 12^\circ$ in the recursive process, 10% of pixels segmented in the previous frame are used to obtain the O_{RGB}^k and B_{RGB}^k sets, and the image size is 346×421 pixels. The image size affects the execution time of the injection of \mathbf{i}_r to the original image, and the conversion of the injected image to the HS plane for its posterior segmentation. The tests have been carried out in a PC with a processor Intel Core 2 Duo with a 2.4 GHz frequency.

Finally, we show some results of the real-time segmentation of images captured in a scene with significant illumination changes. These results highlight again the advantages of using the colour injection proposal presented in this paper. Figure 18 depicts the comparative results in two columns: the left column (a) shows the segmented images without the colour injection, and in the right one (b) the images segmented after applying the colour injection.

The segmentation has been performed by thresholding the *pdf* of the skin class seen in (28), once all the classes have been obtained with the GMM algorithm. In this case, $K = 10$ predefined classes were used. In order to demonstrate the robustness to illumination changes, an incandescent light bulb has been used (that produces a hue change in the whole image that tends to yellow) to really change the illumination of the scenes. The different luminous Intensity levels have been quantified with the mean intensity of the image normalized between $[0, 1]$. The corresponding Intensity levels for the five

images of each column of Figure 18, starting from the image above, are: $I_1 = 0.351$, $I_2 = 0.390$, $I_3 = 0.521$, $I_4 = 0.565$, $I_5 = 0.610$.

Figure 18. Results of the real-time segmentations with different illumination levels: (a) segmentation of original images captured directly from the camera, (b) segmentation of the images after the colour injection.



In the performance of this last practical test, a PC with an Intel Quad Core Q6600 @ 2.4 GHz processor and 2 GB SDRAM @ 633 MHz memory has been used. Although it is a last generation PC with four processing cores (CPUs), our application has only used a single CPU. A Fire-Wire video camera with a 1/2" CCD sensor with a 640×480 spatial resolution and an image capture rate of 30 fps (*RGB* without compression) was used. The optic used is a C-Mount of 3/4" with a focal length of $f = 12$ mm. The different algorithms of our proposal (GMM, colour injection and segmentation) have been developed in C, under Linux OpenSuSE10.3 (x86_64) operating system. With this configuration, the average processing time of the on-line process (T_p) is approximately 2 ms for $N = 50$.

9. Conclusions

A method to increase the separation between two classes in a pixel classification process has been proposed. The experimental results demonstrate that injecting colour in the captured image guarantees good results in maximizing the class separation, implying that class distributions adopt more Gaussian shapes, and, therefore, the segmentation of the desired object improves.

Its practical implementation results are simple and the process time is small. Even though the algorithm needs to calculate both class deviations in each iteration, these are easy to obtain, considering that classes are formed by a limited number of samples (N), the increase of θ_i is not very small, $\Delta\theta = 1^\circ$; and the search interval is not very wide: $\theta_f = 12^\circ$. This implies that calculations are relatively fast.

In this work, the expressions of interest to understand the vector's behaviour in the *HSI* space have been demonstrated from the respective statistics in the *RGB* space. Moreover, the equations to convert directly from YC_1C_2 colour space to *HSI* space have been obtained.

Finally, we should indicate that the images have been directly obtained from the classification process without any other auxiliary stage, such as morphological operations.

For future work, our research is currently focused on the injection of a vector \mathbf{i}_r with a non-zero mean, as a function of the intensity mean desired in the image, to increase the compensation of the effects caused by the illumination changes. We are also developing the hue and saturation dispersion model when the classes are translated all over the *HS* plane, using the C_1C_2 plane, (similar to the hue and saturation deviation estimation that is made in [31] for the *HSI* space defined in [24]). This will diminish the processing time, because it will not be necessary to calculate the hue and saturation variances of both classes in each iteration. Finally, we are doing research into the class separation applying higher order transformations that imply scales and rotations of the classes in the C_1C_2 plane. This could solve part of the intrinsic limitations of the colour injections for just adding the vector \mathbf{i}_r .

Acknowledgements

This work has been supported by the Spanish Ministry of Science and Innovation under projects VISNU (Ref. TIN2009-08984) and SD-TEAM (Ref. TIN2008-06856-C05-05). The authors would also like to thank the Vice-rectory of Research, Innovation and Inter-Institutional Relations (VRIII) from PUCMM, and the Ministry of Higher Education, Science and Technology (MESCYT) from the Dominican Republic through the FONDOCYT program.

References

1. Phung, S.L.; Bouzerdoum, A.; Chai, D. Skin segmentation using color pixel classification: Analysis and comparison. *IEEE Trans. Patt. Anal. Mach. Int.* **2005**, *27*, 148-154.
2. Hsu, R.L.; Abdel-Mottaleb, M.; Jain, A.K. Face detection in color images. *IEEE Trans. Patt. Anal. Mach. Int.* **2002**, *24*, 606-706.
3. Sigal, L.; Sclaroff, S.; Athitsos, V. Skin color-based video segmentation under time-varying illumination. *IEEE Trans. Patt. Anal. Mach. Int.* **2004**, *26*, 862-877.

4. Zhu, X.; Yang, J.; Waibel, A. Segmenting Hands of Arbitrary Color; In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, 28–30 March 2000; pp. 446-453.
5. Fritsch, J.; Lang, S.; Kleinhagenbrock, A.; Fink, G.A.; Sagerer, G. Improving Adaptive Skin Color Segmentation by Incorporating Results from Detection, In *Proceedings of IEEE 11th International Workshop on Robot and Human Interactive Communication*, Berlin, Germany, 25–27 September 2002; pp. 337-343.
6. Bergasa, L.M.; Mazo, M.; Gardel, A.; Sotelo, M.A.; Boquete, L. Unsupervised and adaptive gaussian skin color model. *Image Vision Comput.* **2000**, *18*, 987-1003.
7. Soriano, M.; Martinkauppi, B.; Huovinen, S.; Laaksonen, M. Skin Detection in Video under Changing Illumination Conditions; In *Proceedings of IEEE 15th International Conference on Pattern Recognition*, Barcelona, Spain, 3–7 September 2000; Volume 1, pp. 839-842.
8. Zhang, C.; Wang, P. A New Method of Color Image Segmentation Based on Intensity and Hue Clustering; In *Proceedings of IEEE 15th International Conference on Pattern Recognition*, Barcelona, Spain, 3-7 September, 2000; Volume 3, pp. 613-616.
9. Sural, S.; Qian, G.; Pramanik, S. Segmentation and Histogram Generation Using the HSV Color Space for Image Retrieval; In *Proceedings of IEEE 15th International Conference on Image Processing*, Rochester, New York, NY, USA, 24–28 June 2002; Volume 2, pp. 589-592.
10. Habili, N.; Lim, C.; Moini, A. Segmentation of the face and hands in sign language video sequences using color and motion cues. *IEEE Trans. Circ. Syst. Video T.* **2004**, *14*, 1086-1097.
11. Chai, D.; Ngan, K.N. Face segmentation using skin-color map in videophone applications. *IEEE Trans. Circ. Syst. Video T.* **1999**, *9*, 551-564.
12. Ribeiro, H.L.; Gonzaga, A. Hand Image Segmentation in Video Sequence by GMM: A Comparative Analysis; In *Proceedings of Brazilian Symposium on Computer Graphics and Image Processing SIBGRAPI '06*, Manaus, Amazonas, Brazil, 8–11 October 2006; pp. 357-364.
13. Huang, Z.K.; Liu, D.H. Segmentation of Color Image Using EM algorithm in HSV Color Space; In *Proceedings of International Conference on Information Acquisition, ICIA'07*, Jeju Island, Korean, 8–11 July 2007; Volume 8, pp. 316-319.
14. Fu, H.C.; Lai, P.S.; Lou, R.S.; Pao, H.T. Face Detection and Eye Localization by Neural Network Based Color Segmentation; In *Proceedings of Neural Networks for Signal Processing; IEEE Signal Processing Society Workshop*, Sydney, Australia, 11–13 December 2000; Volume 2, pp. 507-516.
15. Tseng, D.C.; Chang, C.H. Color Segmentation Using Perceptual Attributes; In *Proceedings of IEEE 11th International Conference on Pattern Recognition*, The Hague, The Netherlands, 30 August–1 September 1992; Volume 3, pp. 228-231.
16. Terrillon, J.C.; David, M.; Akamatsu, S. Detection of Human Faces in Complex Scene Images by Use of a Skin Color Model and of Invariant Fourier-Mellin Moments; In *Proceedings of IEEE 14th International Conference on Pattern Recognition*, Brisbane, Australia, 16–20 August, 1998; Volume 2, pp. 1350-1355.

17. Hyams, J.; Powell, M.W.; Murphy, R. Cooperative Navigation of Micro-Rovers Using Color Segmentation; In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA '99*, Monterey, CA, USA, 8–9 November 1999; pp. 195-201.
18. Kehtarnavaz, N.; Monaco, J.; Nimtschek, J.; Weeks, A. Color Image Segmentation Using Multi-Scale Clustering, In *Proceedings IEEE Southwest Symposium Image Analysis and Interpretation*, Tucson, AZ, USA, 5–7 April 1998; pp. 142-147.
19. Boykov, Y.; Jolly, M.P. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images, In *Proceedings of International Conference on Computer Vision*, Vancouver, BC, Canada, 7–14 July 2001; Volume 1, pp. 105-112.
20. Boykov, Y.; Funka-Lea, G. Graph cuts and efficient N-D image segmentation. *Int. J. Comput. Vision* **2006**, *70*, 109-131.
21. Lombaert, H.; Sun, Y.; Grady, L.; Xu, C. A Multilevel Banded Graph Cuts Method For Fast Image Segmentation; In *Proceedings of IEEE International Conference on Computer Vision*, Beijing, China, 17–21 October 2005; Volume 1, pp. 259-265.
22. Carron, T.; Lambert, P. Color Edge Detector Using Jointly Hue, Saturation and Intensity; In *Proceedings of IEEE International Conference on Image Processing*, Austin, TX, USA, 13–16 November 1994; Volume 3, pp. 977-981.
23. Carron, T.; Lambert, P. Symbolic Fusion of Hue-Chroma-Intensity Features for Region Segmentation; In *Proceedings of IEEE International Conference on Image Processing*, Laussane, Switzerland, 16–19 September 1996; Volume 1, pp. 971-974.
24. Smith, A.R. Color Gamut Transform Pairs; In *Proceedings of Conference SIGGRAPH'78*, New York, NY, USA, 23–25 August 1978; Volume 12, pp. 12-19.
25. Kay, G.; Jager, G. A Versatile Colour System Capable Of Fruit Sorting and Accurate Object Classification; In *Proceedings of IEEE Symposium Communications and Signal Processing COMSIG '92*, South Africa, 11 September 1992; pp.145-148.
26. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 2nd ed.; Prentice-Hall Inc.: New Jersey, NJ, USA, 2002; p. 299.
27. Vandenbroucke, N.; Macaire, L.; Postaire, J.G. Color Pixels Classification in a Hybrid Color Space; In *Proceedings of IEEE International Conference on Image Processing ICIP 98*, Chicago, IL, USA, 4–7 October 1998; Volume 1, pp. 176-180.
28. Vandenbroucke, N.; Macaire, L.; Postaire, J.G. Color Image Segmentation by Supervised Pixel Classification in a Color Texture Feature Space; Application to Soccer Image Segmentation, In *Proceedings of IEEE 15th International Conference on Pattern Recognition*, Barcelona, Spain, 3–7 September 2000; Volume 3, pp. 621-624.
29. Theodoridis, S.; Koutroumbas, K. *Pattern Recognition*; Academic Press: San Diego, CA, USA, 1999; pp. 155-157.
30. Lee, D.; Baek, S.; Sung, K. Modified K-means algorithm for vector quantizer design. *IEEE Signal Process. Let.* **1997**, *4*, 2-4.
31. Roman í S.; Sobrerilla, P.; Montseny, E. On the Reliability Degree of Hue and Saturation Values of a Pixel for Color Image Classification; In *Proceedings of IEEE 14th International Conference on Fuzzy Systems FUZZ '05*, Reno, NV, USA, 22–25 May 2005; pp. 306-311.

Appendix A

In this appendix, the relationships between the RGB, the HSI and YC₁C₂ spaces are shown.

Given a vector $\mathbf{r} = [R \ G \ B]^T$ located in the RGB space, a vector $\mathbf{c}' = [Y \ C_1 \ C_2]^T$ in the YC₁C₂ space can be calculated using the following expression [8,22,23]:

$$\begin{bmatrix} Y \\ C_1 \\ C_2 \end{bmatrix} = \mathbf{Q} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{A.1})$$

where \mathbf{Q} is the space transformation matrix, given by:

$$\mathbf{Q} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \quad (\text{A.2})$$

From (Equation A.1) the components C_1 and C_2 of the vector $\mathbf{c} = [C_1 \ C_2]^T$ are:

$$C_1 = R - 1/2G - 1/2B, \quad C_2 = \sqrt{3}/2G - \sqrt{3}/2B \quad (\text{A.3})$$

From the last equation, the module (Chroma component, C) and angle, H' , of the vector \mathbf{c} in the plane C_1C_2 are given by:

$$C = \|\mathbf{c}\| = (C_1^2 + C_2^2)^{1/2} = (R^2 + G^2 + B^2 - RG - GB - BR)^{1/2} \quad (\text{A.4})$$

$$H' = \begin{cases} \alpha, & B \leq G \\ 2\pi - \alpha, & \text{otherwise} \end{cases}; \quad \alpha = \cos^{-1} \left(\frac{R - 1/2G - 1/2B}{(R^2 + G^2 + B^2 - RG - GB - BR)^{1/2}} \right) \quad (\text{A.5})$$

On the other hand, the components of a vector $\mathbf{h}' = [H \ S \ I]^T$ in the HSI space [26] are given by:

$$H = \begin{cases} \gamma, & B \leq G \\ 2\pi - \gamma, & \text{otherwise} \end{cases}; \quad \gamma = \cos^{-1} \left(\frac{R - 1/2G - 1/2B}{(R^2 + G^2 + B^2 - RG - GB - BR)^{1/2}} \right) \quad (\text{A.6})$$

$$S = \left(1 - \frac{3 \min(R, G, B)}{R + G + B} \right) \quad (\text{A.7})$$

$$I = (R + G + B)/3 \quad (\text{A.8})$$

We can observe that the angles of the vectors $\mathbf{c} = [C_1 \ C_2]^T$ and $\mathbf{h} = [H \ S]^T$ coincide (vectors superimposed on the HS plane). Therefore, it has been demonstrated that a vector \mathbf{r} in the RGB space can be projected in the HS and C_1C_2 planes with the same phase shift but a different module, that is: $H = H'$ but $S \neq C$. The relationship between S and C is shown the next appendix.

Appendix B

In this appendix, the relationships between the statistics of vectors in RGB components, \mathbf{r} , and its relationship with the components of their respective vectors, \mathbf{h} , in the HSI space or HS plane are shown.

Intensity (I): If μ is defined as the mean of the vector \mathbf{r} , whose expression is:

$$\mu = (R + G + B)/3 \quad (\text{B.1})$$

Then, knowing the expression of the intensity component in the *HSI* space (Equation A.8), the intensity of the vectors \mathbf{h}' and \mathbf{c}' is given by:

$$I = Y = \mu \quad (\text{B.2})$$

Hue (H): Using (Equation B.1), the expression of the variance of the vector \mathbf{r} is given by:

$$\sigma^2 = \frac{2}{9}(R^2 + G^2 + B^2 + RG + GB + BR) \quad (\text{B.3})$$

Relating the Chroma (*C*) expression (Equation A.4) with (Equation B.3), we can conclude:

$$\|\mathbf{c}\| = C = (C_1^2 + C_2^2)^{1/2} = \sqrt{9/2}\sigma \quad (\text{B.4})$$

Then, the equations of the angles of \mathbf{c} in the C_1C_2 plane and \mathbf{h} in the *HS* plane can be rewritten as:

$$H = \begin{cases} \delta, & B \leq G \\ 2\pi - \delta, & \text{otherwise} \end{cases}; \quad \delta = \cos^{-1}\left(\frac{R - 1/2G - 1/2B}{\sqrt{9/2}\sigma}\right) \quad (\text{B.5})$$

Therefore, for any two vectors $\mathbf{r}_1, \mathbf{r}_2$, in *RGB* components and with similar deviations (σ), only six possible values of *H* exist in all the range of *H* in the *HS* plane where the two vectors overlap. This is fulfilled, independently of the intensity of each one, because they are uncorrelated. Since C_1 can be expressed only as a function of σ without $\mu(I)$, the previous equation (Equation B.5) shows the independence between the components *H* and *I*.

Saturation (S): The analysis of the saturation component should be made between the colour sectors of the *HS* plane, *i.e.*, $(0-2\pi/3)$, $(2\pi/3-4\pi/3)$ and $(4\pi/3-2\pi)$, because these ranges delimit the three discontinuities of the saturation function. Knowing that C_1C_2 plane has the same colour ranges that *HS* plane, considering the colour sector $(0-2\pi/3)$ and supposing that \mathbf{c} is within this sector, its angle can be expressed by any or the following expressions:

$$H = \cos^{-1}\left(\frac{R - 1/2G - 1/2B}{\sqrt{9/2}\sigma}\right), \quad H = \sin^{-1}\left(\frac{\sqrt{3}/2G - \sqrt{3}/2B}{\sqrt{9/2}\sigma}\right) \quad (\text{B.6})$$

The *B* component is the minimum one for this colour sector, therefore, the saturation equation (Equation A.7) is given by:

$$S = \frac{(R + G - 2B)}{(R + G + B)}. \quad (\text{B.7})$$

Multiplying by $2\sqrt{9/2}\sigma$ numerator and denominator of (Equation B.7), and applying (Equation B.1), the following expression is obtained:

$$S = \frac{\sqrt{2}\sigma}{\mu} \left(\frac{1}{2} \frac{(R - 1/2G - 1/2B)}{\sqrt{9/2}\sigma} + \frac{\sqrt{3}}{2} \frac{(\sqrt{3}/2G - \sqrt{3}/2B)}{\sqrt{9/2}\sigma} \right) \quad (\text{B.8})$$

Substituting (Equation B.6) in (Equation B.8), the equation of this colour sector saturation is obtained:

$$S = \sqrt{2} \frac{\sigma}{\mu} \cos(H - \pi/3). \quad (\text{B.9})$$

Therefore, the general equation of the saturation is given by:

$$S = \sqrt{2} \frac{\sigma}{\mu} f(H); f(H) = \begin{cases} \cos(H - \pi/3); & (0 < H \leq 2\pi/3) \\ \cos(H - \pi); & (2\pi/3 < H \leq 4\pi/3) \\ \cos(H - 5\pi/3); & (4\pi/3 < H \leq 2\pi) \end{cases} \quad (\text{B.10})$$

where $f(H)$ is a weighting function that takes values in the range from $1/2$ to 1. This function $f(H)$ generates a three lobe curve in the HS plane delimited by the discontinuities corresponding to the three colour sectors of the plane: $(0-2\pi/3)$, $(2\pi/3-4\pi/3)$ and $(4\pi/3-2\pi)$.

From (Equation B.10) we can conclude that the saturation component of a vector in the HSI space varies directly with the standard deviation of the RGB vector that produces it, and inversely proportionally with its mean.

In case we want to control the saturation of a colour keeping the same intensity, only the standard deviation (σ) of the RGB vector needs to be controlled, forcing its mean not to vary, *i.e.*, $I = \text{constant}$. Therefore, relating (Equation B.4) with (Equation B.10) the following expression is obtained:

$$S = K C f(H) = K \|\mathbf{c}\| f(H) \quad (\text{B.11})$$

where $K = 3/(2I)$.

Equation (Equation B.11) represents the relationship between saturation S and the vector \mathbf{c} associated in the C_1C_2 plane. As we can observe, the saturation can be controlled varying the magnitude of the vector \mathbf{c} (C), which is achieved by modifying the standard deviation (σ) of the vector \mathbf{r} . Due to H is also function of σ (Equation B.5), the effect of controlling S by means of \mathbf{c} is also determined by the weighting function $f(H)$.

Performing some operations in (Equation B.10), a new form to express the saturation component from its components C_1 and C_2 is obtained. Therefore, (Equation A.5), (Equation B.2) and (Equation B.10) may be used to obtain new expressions for the space transformations from YC_1C_2 to HSI , given by:

$$H = \begin{cases} \psi, & C_2 \geq 0 \\ 2\pi - \psi, & \text{otherwise} \end{cases}; \quad \psi = \cos^{-1} \left(\frac{C_1}{(C_1^2 + C_2^2)^{1/2}} \right) \quad (\text{B.12})$$

$$S = \begin{cases} \frac{C_1 + \sqrt{3}C_2}{3Y} \Rightarrow & 0 < H \leq 2\pi/3 \\ -\frac{2C_1}{3Y} \Rightarrow & 2\pi/3 < H \leq 4\pi/3 \\ \frac{C_1 - \sqrt{3}C_2}{3Y} \Rightarrow & 4\pi/3 < H \leq 2\pi \end{cases} \quad (\text{B.13})$$

$$I = Y. \quad (\text{B.14})$$

Appendix C

In this appendix, invariants of the mean vectors in the C_1C_2 plane are shown.

The invariants of the mean vectors are determined by the mean vectors \mathbf{c}_1 and \mathbf{c}_2 in the C_1C_2 plane, or directly by the mean vectors \mathbf{r}_1 and \mathbf{r}_2 in the RGB space, and have the property of keeping constant independently of the colour injection performed. In order to demonstrate it, we define a vector \mathbf{d} , which represents the distance between them, whose expression is given by:

$$\mathbf{d} = \mathbf{c}_1 - \mathbf{c}_2 = [d_{C1} \quad d_{C2}]^T \quad (\text{C.1})$$

where

$$d_{C1} = C_{11} - C_{12} = d_R - 1/2d_G - 1/2d_B, \quad d_{C2} = C_{21} - C_{22} = \sqrt{3}/2d_G - \sqrt{3}/2d_B \quad (\text{C.2})$$

where d_R , d_G and d_B represent the R , G and B components, respectively, of the difference of the vectors \mathbf{r}_1 and \mathbf{r}_2 , and (C_{11}, C_{21}) and (C_{12}, C_{22}) the components of the vectors \mathbf{c}_1 and \mathbf{c}_2 . If a vector $\mathbf{i}_r = [\Delta_R \quad \Delta_G \quad \Delta_B]^T$ is injected in the RGB space, the new difference vector (\mathbf{d}_i) between the injected vectors (\mathbf{c}_{i1} and \mathbf{c}_{i2}) is given by:

$$\mathbf{d}_i = \mathbf{c}_{i1} - \mathbf{c}_{i2} = [d_{iC1} \quad d_{iC2}]^T \quad (\text{C.3})$$

where the d_{iC1} and d_{iC2} components are formed by the d_{iR} , d_{iG} and d_{iB} components according to (Equation C.2), where the subscript “i” indicates they have already been injected with the respective component of \mathbf{i}_r . As $d_{iR} = (R_1 + \Delta_R) - (R_2 + \Delta_R) = R_1 - R_2 = d_R$, also $d_{iG} = d_G$ and $d_{iB} = d_B$, therefore:

$$\mathbf{d}_i = \mathbf{d} \quad (\text{C.4})$$

In correspondence with \mathbf{i}_r , if a translation vector in the C_1C_2 plane is defined, such as $\mathbf{i}_c = [\Delta_{C1} \quad \Delta_{C2}]^T$, the effect produced in the difference vector \mathbf{d}_i , when \mathbf{i}_c is added to the vectors \mathbf{c}_1 and \mathbf{c}_2 , is the same as the effect produced by \mathbf{i}_r , because $d_{iC1} = (C_{11} + \Delta_{C1}) - (C_{12} + \Delta_{C1}) = d_{C1}$ and also $d_{iC2} = d_{C2}$, therefore, (Equation C.4) is fulfilled.

From (Equation C.4) we can conclude that this difference vector is not affected by the injection of the vector \mathbf{i}_r in the RGB space, that is, by the addition of \mathbf{i}_c in the C_1C_2 plane, remaining as invariant factors both its magnitude and its orientation. Therefore, the invariants are: $\|\mathbf{d}\|$ and ϕ , whose expressions are given by:

$$\|\mathbf{d}\| = (d_{C1}^2 + d_{C2}^2)^{1/2}, \quad \phi = \begin{cases} \cos^{-1}(d_{C1}/\|\mathbf{d}_c\|); & d_{C2} \geq 0 \\ 2\pi - \cos^{-1}(d_{C1}/\|\mathbf{d}_c\|); & d_{C2} < 0 \end{cases} \quad (\text{C.5})$$

The angle θ formed between both vectors is obtained with:

$$\cos\theta = \frac{\mathbf{c}_1^T \cdot \mathbf{c}_2}{\|\mathbf{c}_1\| \cdot \|\mathbf{c}_2\|} \quad (\text{C.6})$$

where:

$$\mathbf{c}_1^T \cdot \mathbf{c}_2 = (R_1R_2 + G_1G_2 + B_1B_2) - (R_2G_1 + R_2B_1 + G_2R_1 + G_2B_1 + B_2R_1 + B_2G_1)/2 \quad (\text{C.7})$$

Considering the covariance, Cov_{12} , of the vectors \mathbf{r}_1 and \mathbf{r}_2 in the RGB space, and relating it with (Equation C.7), the following expression is obtained:

$$Cov_{12} = (2/9)\mathbf{c}_1^T \cdot \mathbf{c}_2 \quad (\text{C.8})$$

Knowing that the Chroma (C) expression of the vector \mathbf{c} in the C_1C_2 plane is:

$$\|\mathbf{c}\| = C = \sqrt{9/2}\sigma \quad (\text{C.9})$$

Substituting (Equation C.9) and (Equation C.8) in (Equation C.6), we finally obtain the angle expression:

$$\theta = \cos^{-1}(CC_{12}) \quad (\text{C.10})$$

where CC_{12} is the correlation coefficient of the vectors \mathbf{r}_1 and \mathbf{r}_2 , whose expression is given by:

$$CC_{12} = (Cov_{12}/\sigma_1\sigma_2) \quad (\text{C.11})$$

We can conclude from (Equation C.11) that the angle between two vectors in the HS plane is the $arccos$ of the correlation coefficient between both vectors in the RGB space, and from (Equation C.10) we conclude that this angle has a range between 0 and π radians.

© 2010 by the authors; licensee MDPI, Basel, Switzerland. This article is an Open Access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).