



**HAL**  
open science

# An Improved colorimetric invariants and RGB-Depth-based Codebook model for background subtraction using kinect

Julian Murgia, Cyril Meurie, Yassine Ruichek

## ► To cite this version:

Julian Murgia, Cyril Meurie, Yassine Ruichek. An Improved colorimetric invariants and RGB-Depth-based Codebook model for background subtraction using kinect. 3th Mexican International Conference on Artificial Intelligence, Nov 2014, Chiapas, Mexico. pp 380-392, 10.1007/978-3-319-13647-9\_35 . hal-01471395

**HAL Id: hal-01471395**

**<https://hal.science/hal-01471395v1>**

Submitted on 15 Oct 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# An improved colorimetric invariants and RGB-Depth-based Codebook model for background subtraction using Kinect

Julian Murgia<sup>1</sup>, Cyril Meurie<sup>2</sup>, and Yassine Ruichek<sup>1</sup>

<sup>1</sup> IRTES-SeT, UTBM, 90010 Belfort Cedex, France  
{julian.murgia, yassine.ruichek}@utbm.fr

<sup>2</sup> Univ Lille Nord de France, F-59000 Lille,  
IFSTTAR, LEOST, F59650, Villeneuve d'Ascq  
cyril.meurie@ifsttar.fr

**Abstract.** *In this paper we propose to join the benefits of multiple invariant information into the well-know background subtraction method "Codebook". Indeed, this method mainly repose on a color model allowing a separate process of color and intensity distortion. In order to manage hard situations involving high illumination changes, we propose to enhance this model with the use of two supplementary steps: 1/ transforming the input color image using a colorimetric invariant in order to obtain a color-invariant image whatever the illumination conditions; 2/ using depth information as a new data inside the Codebook model, thus performing an RGB-D fusion during the segmentation process.*

**Keywords:** Image color analysis, subtraction techniques, segmentation, object detection, colorimetric invariants, RGB-D, fusion

## 1 Introduction

Objects detection is a very common step in severals applications of Intelligent Transportation Systems. Its use is integrated into many applications in various domains, such as videosurveillance, pedestrian and/or vehicle detection, detection of hazardous situations in public transports[7][19][18].

Multiple methods have been proposed to address this issue [3][2][15], from simplest techniques to more sophisticated. Basically, background subtraction consists in modeling the background from a sequence of images, then compare this background model to every new frame obtained from the camera to finally obtain a foreground segmentation. Depending on each method, the model computation can be made at pixel level of image, or at region-level after a simplification of image. The representation of every part of the model is independant.

Among the most known representations, multimodal distribution methods are known to provide the better results, such as the very common method Generalized Mixture Of Gaussians (MOG) [21][22]. In this case, each pixel is modeled by a mixture of  $n$  gaussian distributions depending on the pixel value. Then, a

match is being searched between each pixel of the new frame and the model. If a match is found, then the pixel is background, else it is foreground. This kind of representation allows a modeling of more complex backgrounds but the method depends on a learning parameter.

Kim et al. propose the Codebook algorithm [13] that doesn't make use of a learning rate parameter and provides generally good detection results. Recent results shows that even if the Codebook can adapt to little illumination changes, it is very sensible to strong changes which can occur in the scene when the model is not yet adapted. This paper is an attempt to tackle this kind of problems.

In the past few years, great use was made of RGB-D cameras such as Kinect distributed by Microsoft®, or Xtion Pro Live distributed by Asus®. These sensors provide high-resolution depth maps in real time at a very low cost. Depth information is estimated by the combination of an infra-red emitter and a standard CMOS sensor. The camera determines the quantity of infra-red light reflected by the scene. The closer the object, the greater the quantity of light reflected.

The use of a fourth component in BGS method can be tackled in different ways: considering it as a separate information, thus performing a background subtraction on RGB components in one hand and the new information in the other hand, and only after fuse the results using a simple boolean condition [14], or include the new information into the background model [8]. In most of cases, noise can be observed in resulting segmentations.

The use of colorimetric invariants proposed previously helped the algorithm to provide usable segmentations even when the background model is desynchronized from the current scene [16]. But, as color information is highly dependant of observation conditions (dark, lit, fog, rain, snow...), colorimetric invariance is not necessarily sufficient to provide strong results. To improve further in this direction, additional invariant data can then be used by the algorithm to perform an ever better segmentation. The Codebook representation method is flexible enough to make use of color and brightness information but also any kind of supplementary data for the model can be provided. We then propose in this paper to merge the use colorimetric invariants with RGB-D fusion inside the Codebook algorithm.

In this paper, we will first describe the main components involved in our contribution. That is, a recap of the Codebook method and its improvement for the use of depth information will be done in Part 2. A brief recall of the concept of colorimetric invariants will be made in Part 3, as well as a short presentation of those which turned out to be the most interesting in our cases. Finally, results and observations will be given in Part 4.

## 2 Codebook

Background modelisation is made by the Codebook method proposed by Kim et al [13]. This method has become a reference in many fields of research such as detection and tracking of moving objects. It is robust and efficient in a wide

number of use cases, including dynamic backgrounds (tree foliages, fountains, sea shores, flags...) and little illumination changes.

This algorithm was driven by the following observation: false detections are generally situated in dark zones of the image. Therefore, as the color of pixels define their darkness or brightness, color should be used as an important factor in the comparison of two pixels. This reflection led to the creation of a new color model used inside the algorithm to evaluate separately color and brightness of pixels.

## 2.1 Learning

Codebook algorithm consists in a clustering of the image to build a background model from a learning period. This model is represented by a list of Codebooks (1 per pixel) each containing a certain number of codewords. A codeword is created (or updated if the observed pixel's representation is similar to an existing codeword) for each pixel of every frame of the learning sequence. A codeword contains two vectors, respectively R,G and B values of the pixel, and other data such minimum and maximum observed brightness, temporal data and frequency of occurrence. During this phase, a new codeword obtained for a given pixel is integrated in the background model if it satisfies two conditions:

1. brightness constraint: the intensity must lie in the interval  $[L_{low}; L_{high}]$  determined from all the minimum  $L_{min}$  and maximum  $L_{max}$  brightness observed for this pixel. This range of brightness delimits the range under which a codeword is considered as shadow, and above which it is considered as highlight. For each codeword, we have :

$$\begin{aligned} L_{low} &= \alpha L_{min}, \\ L_{high} &= \min(\beta L_{max}, \frac{L_{min}}{\alpha}) \end{aligned} \quad (1)$$

where  $\alpha < 1$  and  $\beta > 1$  are fixed parameters of the algorithm.

2. color distortion constraint: the color distortion  $\delta$  of the pixel and the codeword must lie under a given threshold  $\epsilon$ . This  $\delta$  value is calculated from input pixel RGB values and the tested codeword.

After the background model construction, the algorithm optimizes its size by determining Codewords corresponding to pixels erroneously integrated as background pixels. A Codeword  $m$  is defined by a pair of vectors  $V_m = [\bar{R}, \bar{G}, \bar{B}]$  and  $Aux_m = [I_{min}^m, I_{max}^m, f_m, \lambda_m, t, l]$ .  $V_m$  defines the average value of each component.  $I_{min}^m$  and  $I_{max}^m$  define minimum and maximum brightness respectively, of all observations that match to codeword  $m$ .  $\lambda_m$  denoted as Maximum Negative Run-Length (MNRL) is defined as the longest interval of time during which the codeword  $m$  has not been updated.  $p$  and  $q$  define first and last times of update of codeword  $m$ , respectively.

## 2.2 Foreground/background detection

The final background model represents the parts of an image that does not move. It is then possible to compare each Codebook to determine whether an observed pixel belongs to background or not. More simply, the existence of a similar Codeword in the model for this pixel's Codebook is tested, using the same constraints quoted before. If a Codeword matches this pixel, then it belongs to background (black in segmentations). If no match is found, this pixel is marked as foreground (white in segmentations). After foreground detection phase is executed for each pixel, the algorithm updates the background codebooks model.

## 2.3 Codebook RGB-D

Disparity information can be used into the Codebook algorithm by incorporating a new data obtained from the disparity map. As disparity is a 1D information, it can be treated the same way as brightness. The modification made to the algorithm is the addition of a new value in the first vector of Codewords, containing the disparity of the corresponding pixel, that is:  $V_m = [\bar{R}, \bar{G}, \bar{B}, \bar{D}]$ . Also, new values are added into the second vector of Codeword: low disparity  $D_{low}$  and high disparity  $D_{hi}$  values, representing the disparity range allowed for input values, as defined in Equation 2.

$$\begin{aligned} D_{low} &= \alpha_D D_{max} \\ D_{hi} &= \min\{\beta_D D_{max}, \frac{D_{min}}{\alpha_D}\} \end{aligned} \quad (2)$$

having  $D_{min}$  and  $D_{max}$  respectively the minimum and maximum disparities observed for this pixel.  $\alpha_D$  and  $\beta_D$  are thresholds in the depths distortion, defined the same way as for brightness distortion.  $\alpha_D$  value is typically between 0.4 and 0.7,  $\beta_D$  between 1.1 and 1.5. Then, the new disparity distortion function described in Equation 3 is used during both learning and process phase:

$$disparityDist(D, \langle D_{min}, D_{max} \rangle) = \begin{cases} true & isInvalid(D) \vee (D_{hi} \leq D \leq D_{low}) \\ false & otherwise \end{cases} \quad (3)$$

$D$  is the tested disparity, and  $isInvalid()$  function determines whether  $D$  is erroneous. Typically, a disparity value of 0 denotes an invalid disparity. A matching between the observed pixel and an existing codeword means it satisfies those three conditions: color, brightness and disparity distortions. The decision function determining whether a pixel  $x$  matches a codeword  $c_m$  is described in Equation 4. If a pixel matches a codeword in the background model, it is also considered as background (BG), else as foreground (FG).

$$decision(x) = \begin{cases} BG & \text{if } (colorDist(x) \leq \epsilon_1 \vee \\ & (\epsilon_1 < colorDist(x, c_m) \leq \epsilon_2 \\ & \wedge disparityDist(D, \langle D_{min}, D_{max} \rangle))) \wedge \\ & brightnessDist(I, \langle I_{min}, I_{max} \rangle) \wedge \\ & disparityDist(D, \langle D_{min}, D_{max} \rangle) \\ FG & \text{otherwise} \end{cases} \quad (4)$$



**Fig. 1.** Color invariants and color models used. From left to right and top to bottom: original RGB, Greyworld, RGB-Rank,  $L^*a^*b^*$ , HSL, Opposite Colorspace, 11l2l3, c1c2c3, YIQ, YCbCr

Equation 4 describes the pixel classification decision. If the disparity attached to this pixel is valid and comprised between acceptable thresholds, the decision function takes it into account. Otherwise, only color and brightness distortions are considered. If a matching occurs, the pixel is set as background. Otherwise, it is classified as foreground.

After the segmentation process is done, we applied a chain of morphological operators in order to remove false positives remaining in the background, and holes filling to get plain silhouettes when possible.

### 3 Color invariance

Prior to any Codebook-related action, we propose to apply a colorimetric invariant to modify the color aspect of the image to give it the aspect it would have under a canonical illuminant. This concept assumes that one can perceive the color of an object, whichever the color of the illuminant. To perform this task, it is necessary to estimate this illuminant color. Multiple methods can achieve this, which can be sorted into 3 categories:

1. Use of low-level characteristics: these methods are based on low-level statistics or a dichromatic reflection model, physics-based
2. Use of a learning phase: the illuminant is determined using a model obtained from a learning dataset
3. A combination of these two methods.

Every colorimetric invariance method does not produce identic images. They often need to create a new model, adapted to the characteristics they make use of. Therefore, no method can be considered as "universal". Some works also consist in a combination of multiple strategies: Gijssen et al. [12] showed that different colorimetric invariance algorithms perform better on different types of images; Bianco et al. [1] proposed a CART-based algorithm to choose the best colorimetric invariant for a given image, using a decision forest technique. In this paper, we prefer testing several colorimetric invariants: Greyworld [4], Affine Normalization [17], Chromaticity Space, Comprehensive Normalization [10], Opposite Colorspace, Reduced Coordinates, m1m2m3, 11l2l3 and c1c2c3

[11], YIQ [5], YCh1Ch2 [6], YCbCr, RGB-Rank [9], CIE  $L^*a^*b^*$ , HSL color models [20].

Figure 1 shows some invariant images tested from one RGB image and used as input images in the Codebook algorithm.

## 4 Results and observations

### 4.1 Dataset, tests and metrics

Images acquisitions have been made in a room six times at different moments of day, implying different lighting conditions. The scenario for all bases is the same: a person enters the room, walk before his desk, then after, sits down on a chair, stays for a while, then stands up and walks out. Figure 2 shows six images taken from each database. In certain cases, the curtains are closed and light can be turned off, in order to simulate a very difficult case when the room is very dark.



**Fig. 2.** Original color images from images L1 to L6

These six cases provide different tests to determine the benefits of the RGB-D fusion inside the Codebook when applied to different bases and not up-to-date learning bases, because illuminations and colors are very different from one base to another. That is, learnings are applied on 50 frames (RGB and disparity maps obtained from the Kinect) from each learning base, with every tested colorimetric invariant or color model. From each of these learnings, we compute segmentations for all six processing bases, using corresponding colorimetric invariant / color model as well. Thus, a naming convention was created for each test performed: we call  $Lx$  the learning bases and  $Px$  the processing bases, where  $x$  is the number of the corresponding base as enumerated before. Consequently, a test named L1\_P3 defines a test where the learning was done with base 1 and the process was done with base 3. Of course, a colorimetric invariant name (or color model) is attached to every test.

The use of color-invariant images is supposed to increase the robustness of the Codebook model. As a consequence, the codebook algorithm was also modified to avoid the model update occurring after the detection phase of the Codebook process. This way, we were able to observe the effects of colorimetric invariants on adaptative as well as fixed RGB and RGB-D codebook model. Non-adaptative codebooks correspond to algorithms which do not update their background model while adaptative codebooks do update their model. This detail is critical in the way to compare results, since these two modes do not correspond to the same algorithm anymore. Therefore, adaptative Codebooks results can be compared only with each other, as non-adaptative Codebooks can be compared only amongst them.

To evaluate our method and be able to perform a quantitative analysis, hand-segmented ground truths were produced. Relative measures Recall and Precision were then calculated using true/false positives and true/false negatives (TP, FP, TN, FN) for *both* foreground and background classes. This choice is motivated by the fact that considering the effectiveness of the Codebook (RGB) algorithm, the improvements the new method including disparity will mostly be situated in the background zones of the images. These measures are then combined into an accuracy metric, F-Measure  $FM$  to evaluate the quality of the segmentations. Recall, Precision and F-Measure are defined as follows:

$$\begin{aligned} Recall_{fg} &= \frac{TP}{TP+FN}, & Recall_{bg} &= \frac{TN}{TN+FP}, \\ Precision_{fg} &= \frac{TP}{TP+FP}, & Precision_{bg} &= \frac{TN}{TN+FN}, \\ FM_c &= \frac{2.(Recall_c.Precision_c)}{Recall_c+Precision_c} \end{aligned} \quad (5)$$

Tests were performed with a learning base image L1 constituted of 50 images. Codebook values used are:  $\alpha_c = 0.4, \beta_c = 1.7, \alpha_D = 0.75, \beta_D = 1.25, \epsilon_1 = 10, \epsilon_2 = 16$ . These values were determined empirically to allow the algorithm to produce interesting enough results with every tested database.

## 4.2 Results and analysis

Tables 1 and 3 display mean F-measures corresponding to Codebook RGB algorithms, non-adaptative and adaptative respectively. Positive (bold values) and negative values describe respectively improvement and deterioration, referring to values obtained without any use of colorimetric invariants or color spaces. More precisely, Chromaticity Space, RGB-Rank, c1c2c3, L\*a\*b\* and HSL provided the best results amongst all tested colorimetric invariants and color models. Red and green values indicate for each line the maximum and minimum value, respectively. In the case of non-adaptative Codebook RGB, gains appear important. Chromaticity Space especially provide the best results improvements, between +22% and +77% for background regions and between 0% and +46% for foreground regions, even though improvements are brought by every colorimetric invariant displayed in Table 1.

Depth information usage in non-adaptative Codebook RGB-D to improve these results showed interesting results as well, as showed in Table 2. In this table, F-Measures appear a little lower than in Table 1, even though improvements still remain situated between +33% and +70% in background, and between +13% and 42% in foreground. Again, Chromaticity Space showed the best results in this case.

Figures 3, 4 and 5 show segmentations obtained when the learning was performed on L1 (first database, corresponding to the top-left corner image) with three algorithm versions presented, respectively adaptative Codebook RGB, non-adaptative Codebook RGB-D and adaptative Codebook RGB-D. Tests were processed on all databases, from L1 to L6 in the following order, with every colorimetric invariant and color model separately. Learning and process phases were done using the same colorimetric invariant or color model for both phases.



**Table 1.** F-Measures gains and deteriorations for non-adaptative Codebook RGB

	cs		rgb-r		c1c2c3		hsl		yiq		ych1ch2		lab	
	bg	fg	bg	fg	bg	fg	bg	fg	bg	fg	bg	fg	bg	fg
L1_P1	<b>45.5%</b>	<b>45.7%</b>	<b>31.8%</b>	<b>20.2%</b>	<b>43.5%</b>	<b>39.6%</b>	<b>31.2%</b>	<b>15.3%</b>	<b>26.7%</b>	<b>22.6%</b>	<b>8.2%</b>	<b>9.2%</b>	<b>41.1%</b>	<b>45.0%</b>
L1_P2	<b>57.3%</b>	<b>43.6%</b>	<b>40.7%</b>	<b>12.5%</b>	<b>52.4%</b>	<b>32.3%</b>	<b>31.0%</b>	<b>6.3%</b>	<b>40.9%</b>	<b>15.3%</b>	<b>10.2%</b>	<b>1.0%</b>	<b>51.2%</b>	<b>37.0%</b>
L1_P3	<b>60.2%</b>	<b>39.6%</b>	<b>43.5%</b>	<b>11.9%</b>	<b>45.0%</b>	<b>23.8%</b>	<b>20.4%</b>	<b>4.9%</b>	<b>12.9%</b>	<b>3s,4%</b>	<b>6.4%</b>	<b>0.4%</b>	<b>39.4%</b>	<b>22.2%</b>
L1_P4	<b>34.7%</b>	<b>31.6%</b>	<b>23.3%</b>	<b>16.8%</b>	<b>33.0%</b>	<b>25.7%</b>	<b>21.1%</b>	<b>8.3%</b>	<b>19.6%</b>	<b>10.7%</b>	<b>-10.8%</b>	<b>-1.6%</b>	<b>36.2%</b>	<b>41.9%</b>
L1_P5	<b>22.9%</b>	<b>0.4%</b>	<b>2.9%</b>	<b>0.2%</b>	<b>21.1%</b>	<b>0.3%</b>	<b>14.5%</b>	<b>0.5%</b>	<b>22.7%</b>	<b>0.2%</b>	<b>12.3%</b>	<b>0.4%</b>	<b>6.2%</b>	<b>0.3%</b>
L1_P6	<b>76.8%</b>	<b>14.8%</b>	<b>43.6%</b>	<b>4.2%</b>	<b>57.2%</b>	<b>8.0%</b>	<b>23.9%</b>	<b>2.0%</b>	<b>9.4%</b>	<b>1.1%</b>	<b>19.0%</b>	<b>1.4%</b>	<b>8.4%</b>	<b>1.2%</b>
Average	<b>49.6%</b>	<b>29.3%</b>	<b>31.0%</b>	<b>11.0%</b>	<b>42.0%</b>	<b>21.6%</b>	<b>23.7%</b>	<b>6.2%</b>	<b>22.0%</b>	<b>8.9%</b>	<b>7.5%</b>	<b>1.8%</b>	<b>30.4%</b>	<b>24.6%</b>

**Table 2.** F-Measures gains and deteriorations for non-adaptative Codebook RGB-D

	cs		rgb-r		c1c2c3		hsl		yiq		ych1ch2		lab	
	bg	fg	bg	fg	bg	fg	bg	fg	bg	fg	bg	fg	bg	fg
L1_P1	<b>34.9%</b>	<b>38.1%</b>	<b>26.4%</b>	<b>33.8%</b>	<b>33.9%</b>	<b>31.8%</b>	<b>22.1%</b>	<b>6.9%</b>	<b>13.5%</b>	<b>6.7%</b>	<b>-2.7%</b>	<b>-4.2%</b>	<b>28.2%</b>	<b>31.9%</b>
L1_P2	<b>51.9%</b>	<b>41.3%</b>	<b>43.2%</b>	<b>25.6%</b>	<b>48.5%</b>	<b>30.8%</b>	<b>11.5%</b>	<b>1.1%</b>	<b>15.8%</b>	<b>4.1%</b>	<b>-4.5%</b>	<b>-1.1%</b>	<b>40.3%</b>	<b>31.5%</b>
L1_P3	<b>50.1%</b>	<b>41.5%</b>	<b>41.9%</b>	<b>23.7%</b>	<b>45.8%</b>	<b>29.8%</b>	<b>15.4%</b>	<b>4.5%</b>	<b>-4.9%</b>	<b>2.3%</b>	<b>-4.7%</b>	<b>-0.8%</b>	<b>25.6%</b>	<b>20.0%</b>
L1_P4	<b>18.7%</b>	<b>22.2%</b>	<b>13.0%</b>	<b>19.5%</b>	<b>16.7%</b>	<b>15.3%</b>	<b>7.2%</b>	<b>0.4%</b>	<b>-9.7%</b>	<b>-5.7%</b>	<b>-31.3%</b>	<b>-12.1%</b>	<b>19.7%</b>	<b>29.1%</b>
L1_P5	<b>14.4%</b>	<b>0.6%</b>	<b>-1.6%</b>	<b>0.0%</b>	<b>13.8%</b>	<b>0.6%</b>	<b>9.1%</b>	<b>0.4%</b>	<b>33.6%</b>	<b>3.0%</b>	<b>5.5%</b>	<b>0.1%</b>	<b>3.6%</b>	<b>0.3%</b>
L1_P6	<b>70.3%</b>	<b>13.5%</b>	<b>51.7%</b>	<b>6.1%</b>	<b>56.0%</b>	<b>8.5%</b>	<b>24.7%</b>	<b>2.2%</b>	<b>2.0%</b>	<b>0.6%</b>	<b>19.0%</b>	<b>1.4%</b>	<b>3.4%</b>	<b>0.6%</b>
Average	<b>40.0%</b>	<b>26.2%</b>	<b>29.1%</b>	<b>18.1%</b>	<b>35.8%</b>	<b>19.4%</b>	<b>15.0%</b>	<b>2.6%</b>	<b>8.4%</b>	<b>1.8%</b>	<b>-3.1%</b>	<b>-2.8%</b>	<b>20.1%</b>	<b>18.9%</b>

**Table 3.** F-Measures gains and deteriorations for adaptative Codebook RGB

	cs		rgb-r		c1c2c3		hsl		yiq		ych1ch2		lab	
	bg	fg	bg	fg	bg	fg	bg	fg	bg	fg	bg	fg	bg	fg
L1_P1	<b>3.0%</b>	<b>-19.2%</b>	<b>1.8%</b>	<b>0.3%</b>	<b>2.6%</b>	<b>-16.8%</b>	<b>3.3%</b>	<b>10.3%</b>	<b>3.2%</b>	<b>0.2%</b>	<b>1.5%</b>	<b>3.5%</b>	<b>3.0%</b>	<b>-15.5%</b>
L1_P2	-0.1%	<b>0.0%</b>	<b>0.6%</b>	<b>0.0%</b>	<b>0.0%</b>	<b>0.0%</b>	<b>-0.2%</b>	<b>0.0%</b>	<b>-0.4%</b>	<b>0.0%</b>	<b>-0.8%</b>	<b>0.0%</b>	<b>0.3%</b>	<b>0.0%</b>
L1_P3	-0.2%	<b>5.7%</b>	<b>0.6%</b>	<b>3.7%</b>	<b>-0.1%</b>	<b>-4.7%</b>	<b>0.8%</b>	<b>14.7%</b>	<b>-0.5%</b>	<b>-0.3%</b>	<b>-3.1%</b>	<b>1.2%</b>	<b>0.3%</b>	<b>-2.3%</b>
L1_P4	<b>0.4%</b>	<b>7.0%</b>	<b>0.3%</b>	<b>4.7%</b>	<b>-0.2%</b>	<b>-2.0%</b>	<b>0.4%</b>	<b>8.9%</b>	<b>0.6%</b>	<b>1.0%</b>	<b>0.8%</b>	<b>7.7%</b>	<b>-0.1%</b>	<b>-1.7%</b>
L1_P5	<b>-1.9%</b>	<b>-11.3%</b>	<b>-7.8%</b>	<b>16.2%</b>	<b>-3.3%</b>	<b>-16.5%</b>	<b>-0.5%</b>	<b>-2.1%</b>	<b>-5.2%</b>	<b>-13.0%</b>	<b>1.5%</b>	<b>-6.7%</b>	<b>-1.5%</b>	<b>-6.9%</b>
L1_P6	<b>-1.2%</b>	<b>-52.1%</b>	<b>-0.7%</b>	<b>-11.5%</b>	<b>-1.6%</b>	<b>-50.9%</b>	<b>-0.8%</b>	<b>-19.5%</b>	<b>-0.5%</b>	<b>-8.4%</b>	<b>-1.4%</b>	<b>-10.5%</b>	<b>-0.8%</b>	<b>-37.3%</b>
Average	<b>0.0%</b>	<b>-11.7%</b>	<b>-0.8%</b>	<b>2.2%</b>	<b>-0.4%</b>	<b>-15.2%</b>	<b>0.5%</b>	<b>2.1%</b>	<b>-0.5%</b>	<b>-3.4%</b>	<b>-0.2%</b>	<b>-0.8%</b>	<b>0.2%</b>	<b>-10.6%</b>

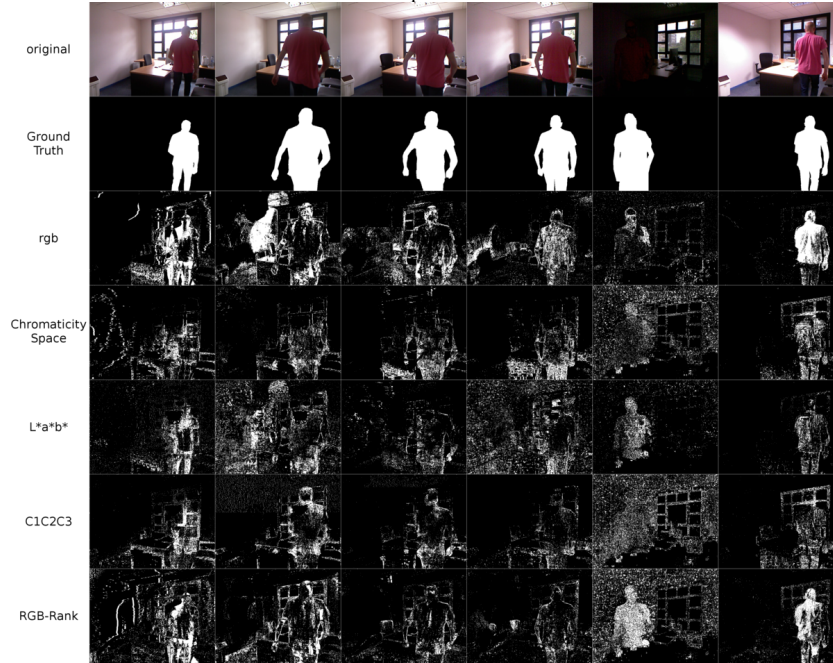
**Table 4.** F-Measures gains and deteriorations for adaptative Codebook RGB-D

	cs		rgb-r		c1c2c3		hsl		yiq		ych1ch2		lab	
	bg	fg	bg	fg	bg	fg	bg	fg	bg	fg	bg	fg	bg	fg
L1_P1	<b>1.4%</b>	<b>10.6%</b>	<b>-0.2%</b>	<b>0.2%</b>	<b>1.4%</b>	<b>13.3%</b>	<b>0.3%</b>	<b>12.1%</b>	<b>-1.3%</b>	<b>1.8%</b>	<b>-0.9%</b>	<b>4.9%</b>	<b>-0.4%</b>	<b>0.6%</b>
L1_P2	<b>-1.4%</b>	<b>12.0%</b>	<b>0.5%</b>	<b>4.4%</b>	<b>0.7%</b>	<b>17.5%</b>	<b>-0.3%</b>	<b>15.8%</b>	<b>-0.7%</b>	<b>6.5%</b>	<b>-0.4%</b>	<b>6.9%</b>	<b>0.2%</b>	<b>5.3%</b>
L1_P3	<b>-0.1%</b>	<b>8.6%</b>	<b>0.6%</b>	<b>4.9%</b>	<b>0.4%</b>	<b>14.1%</b>	<b>0.5%</b>	<b>12.7%</b>	<b>-0.8%</b>	<b>3.1%</b>	<b>-1.4%</b>	<b>4.8%</b>	<b>0.1%</b>	<b>1.1%</b>
L1_P4	<b>0.8%</b>	<b>11.5%</b>	<b>0.3%</b>	<b>4.2%</b>	<b>0.7%</b>	<b>14.1%</b>	<b>0.6%</b>	<b>13.4%</b>	<b>0.9%</b>	<b>6.2%</b>	<b>0.8%</b>	<b>10.3%</b>	<b>0.1%</b>	<b>2.5%</b>
L1_P5	<b>-1.6%</b>	<b>17.9%</b>	<b>-5.8%</b>	<b>24.3%</b>	<b>-4.3%</b>	<b>8.5%</b>	<b>0.2%</b>	<b>14.0%</b>	<b>-5.3%</b>	<b>-9.6%</b>	<b>0.4%</b>	<b>-3.8%</b>	<b>-2.0%</b>	<b>-3.9%</b>
L1_P6	<b>-0.4%</b>	<b>-29.9%</b>	<b>-0.1%</b>	<b>-4.6%</b>	<b>-0.4%</b>	<b>-23.1%</b>	<b>0.2%</b>	<b>4.7%</b>	<b>-0.6%</b>	<b>-6.5%</b>	<b>-0.8%</b>	<b>-0.9%</b>	<b>0.1%</b>	<b>-12.2%</b>
Average	<b>-0.2%</b>	<b>5.1%</b>	<b>-0.8%</b>	<b>5.6%</b>	<b>-0.2%</b>	<b>7.4%</b>	<b>0.2%</b>	<b>12.1%</b>	<b>-1.3%</b>	<b>0.2%</b>	<b>-0.4%</b>	<b>3.7%</b>	<b>-0.3%</b>	<b>-1.1%</b>

The quality of segmentations obtained with adaptative Codebook RGB (Figure 3) is low as no average F-Measure came upper than 2.2% and 0.5% improvement in foreground and background classes respectively. This did not allow a post-processing operation to provide better results. Noise is observed in the background parts of images and foreground silhouettes were not homogen enough to allow a proper reconstruction.

Figures 4 and 5, in another hand, display results making use of depth information. In these two cases, a post-processing was applied consisting in performing morphological operations (closing, opening, holes filling) in order to remove noise and false positives in the background as well as to improve silhouettes shapes.

Figure 5 shows results with the adaptative Codebook as it provided the best results (with post-processing). These segmentations lead to some first qualitative observations: when processing original images (line 3) depending on the moment of day, hollowed foreground appear and noised background remain in results. Tests driven with the use of colorimetric invariants helped the filling of fore-

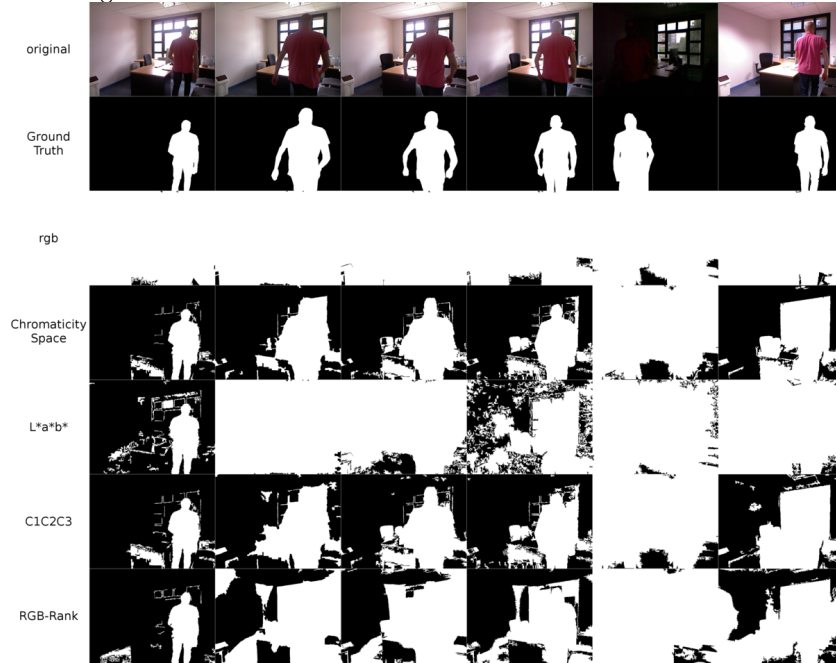


**Fig. 3.** Segmentations obtained with algorithm adaptative Codebook RGB with a learning on L1. From left to right: P1 to P6 processes. Bottom lines display most interesting colorimetric invariants and color models according to our results.

ground silhouettes, especially in L1\_P5 case: at the price of a higher background noise (-0.2%), c1c2c3 increased the foreground F-Measure by 7.4%. In certain test cases though (L1\_P4), the improvement is less visible.

Certain colorimetric invariants such as YCbCr, YCh1Ch1 or m1m2m3 do not converge towards one similar image. Depending on the moments of the day, two images which look originally similar can look very different when using one colorimetric invariant on both. Moreover, certain colorimetric invariants (ie: l1l2l3, Opposite colorspace, HSL...) do not output "real world color" images. This state of fact explains some results less interesting than others, depending on databases.

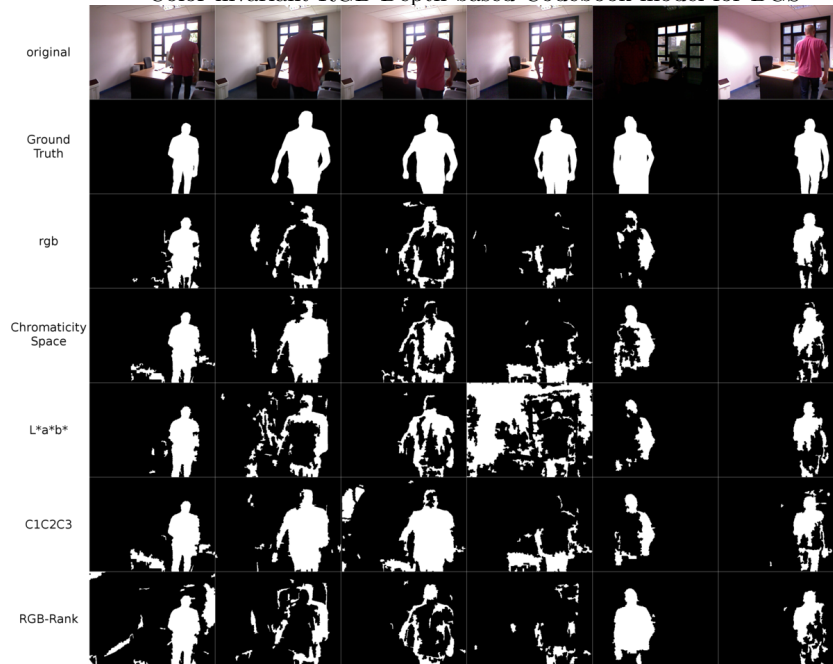
Previous qualitative observations are confirmed by F-Measure values. Results obtained in [16] with Codebook RGB are very similar, as we obtain slightly better results with certain colorimetric invariants increasing F-Measures. Non-adaptative Codebooks (RGB and RGB-D) effectively improve more importantly the F-Measure values for both foreground and background classes than adaptative Codebooks do. Table 1 demonstrates these improvements as F-Measure increase up to 78% for background and 45% foreground (with Codebook RGB and Chromaticity space color invariant). Table 2 also shows improvements as F-Measure increase up to 70% for background and more than 40% in foreground (with Codebook RGB-D and Chromaticity space color invariant). More globally, these results confirm the interest of the use of colorimetric invariant to improve the segmentation of moving objects.



**Fig. 4.** Segmentations obtained with algorithm non-adaptative Codebook RGB-D with a learning on L1. From left to right: P1 to P6 processes. Bottom lines display most interesting colorimetric invariants and color models according to our results. Segmentations are filtered with morphological operators.

As stated previously, depth inclusion into the codebook algorithm globally improves segmentations quality by +7% for foreground class and +5% for background class when using original RGB images jointly with depth values in  $L_x-P_x$  cases (learning and processing bases are the same).

When learning and process times are different ( $L_x-P_y$ ) cases, of course F-Measure decrease since these cases are more difficult. When no colorimetric invariant is used, foreground F-Measures decrease below 22% and 50%. But even then, colorimetric invariants such as c1c2c3, Chromaticity space, RGB-Rank, HSL and YCh1Ch2 improve foreground F-Measure in certain cases by more than 40%. One can also notice that these average values are widely reduced by L1\_P6 average F-Measures. This denotes a limit of the method: very time-shifted databases are difficult for the algorithm to manage, and colorimetric invariants are not sufficient enough to limit this effect, which denote the need of a proper learning update even if it can be shifted. When no improvement is brought, no noticeable deterioration exists either. This means that for best colorimetric invariants, results are at least very close to results obtained when using original images, at best they are better. Little deteriorations are observed in background class while this method interestingly improves detection quality of foreground class. These results confirm the benefits of the joint use of depth in a background subtraction technique as well as the use of modified images with colorimetric invariants or different color spaces.



**Fig. 5.** Segmentations obtained with algorithm adaptive Codebook RGB-D with a learning on L1. Bottom lines display most interesting colorimetric invariants and color models according to our results. Post-processing is applied after segmentation.

## 5 Conclusion

In this paper, we have proposed a new solution for foreground detection methods involving colorimetric invariance and RGB-D fusion. Depth information collected from active sensors were used together with different colorimetric invariant modified images, thus occasionally providing better segmentation results. This method opens a door for future works dealing with difficult, indoor and outdoor environments which often cause detection problems because of important lighting changes. Future works should propose a different way to manage the color constancy inside the background model, and determine the best colorimetric invariance method according to each situation.

## References

1. S. Bianco, G. Ciocca, C. Cusano, and R. Schettini. Automatic color constancy algorithm selection and combination. *Pattern Recognition*, 43(3):695 – 705, 2010.
2. Thierry Bouwmans and Fida El Baf. Statistical background modeling for foreground detection: A survey. *Handbook of Pattern Recognition and Computer*, 2010.
3. Thierry Bouwmans, Fida El Baf, Bertrand Vachon, et al. Background modeling using mixture of gaussians for foreground detection-a survey. 2008.
4. G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 1980.

5. Walter H. Buchsbaum. *Color TV Servicing, third edition*. Englewood Cliffs, NJ: Prentice Hall, 1975.
6. T Carron. *Segmentation d'images couleur dans la base Teinte Luminance Saturation : approche numerique et symbolique*. PhD thesis, Universite de Stanford, 1995.
7. D.-N. Truong Cong, L. Khoudour, C. Achard, C. Meurie, and O. Lezoray. People re-identification by spectral classification of silhouettes. *Signal Processing*, 90(8):2362 – 2374, 2010. Special Section on Processing and Analysis of High-Dimensional Masses of Image and Signal Data.
8. Enrique J. Fernandez-Sanchez, Javier Diaz, and Eduardo Ros. Background subtraction based on color and depth using active sensors. *Sensors*, 13(7):8895–8915, 2013.
9. Graham D. Finlayson, Steven D. Hordley, Gerald Schaefer, and Gui Yun Tian. Illuminant and device invariant colour using histogram equalisation. *Pattern Recognition*, 2005.
10. Graham D. Finlayson, Bernt Schiele, and James L. Crowley. Comprehensive colour image normalization. 1998.
11. Theo Gevers and Arnold W.M. Smeulders. Color-based object recognition. *Pattern Recognition*, 1999.
12. A. Gijsenij and T. Gevers. Color constancy using natural image statistics and scene semantics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2010.
13. K. Kim, T. H. Chalidabhongse, D. Hanuood, and L. Davis. Background modeling and subtraction by codebook construction. 2004.
14. Alex Leykin. Robust multi-pedestrian tracking in thermal-visible surveillance videos. In *in and Beyond the Visible Spectrum Workshop at the International Conference on Computer Vision and Pattern Recognition, 0:136, 2006*. 88, pages 0–136, 2006.
15. Alan M. Mcivor. Background Subtraction Techniques. 2000.
16. J. Murgia, C. Meurie, and Y. Ruichek. Improvement of moving objects detection in continued all-day illumination conditions using color invariants and color spaces. 2013.
17. Stephan Obdrzalek, Jiri Matas, and Ondrej Chum. On the interaction between object recognition and colour constancy. *Proc. International Workshop on Color and Photometric Methods in Computer Vision*, 2003.
18. Houssam Salmane, Yassine Ruichek, and Louahdi Khoudour. Gaussian propagation model based dense optical flow for objects tracking. In *International Conference on Image Analysis and Recognition (ICIAR'2012), LNCS*, pages 234–244, Aveiro, Portugal, 2012.
19. Houssam Salmane, Yassine Ruichek, and Louahdi Khoudour. Using hidden markov model and dempster-shafer theory for evaluating and detecting dangerous situations in level crossing environments. In *Mexican International Conference on Artificial Intelligence (MICAI'2012), LNCS*, San Luis Potosi, Mexico, 2012.
20. Alvy Ray Smith. Color gamut transform pairs. *SIGGRAPH Comput. Graph.*, 1978.
21. C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:2246, 1999.
22. Z. Zivkovic and F. van der Heijden. Recursive unsupervised learning of finite mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(5):651–656, May 2004.