



HAL
open science

Improvement of moving objects detection in continued all-day illumination conditions using color invariants and color spaces

Julian Murgia, Cyril Meurie, Yassine Ruichek

► To cite this version:

Julian Murgia, Cyril Meurie, Yassine Ruichek. Improvement of moving objects detection in continued all-day illumination conditions using color invariants and color spaces. ITSC 2013, IEEE International Conference on Intelligent Transportation Systems (ITSC), Oct 2013, The Hague, Netherlands. 6p, <https://doi.org/10.1109/ITSC.2013.6728476> . hal-00909451

HAL Id: hal-00909451

<https://hal.science/hal-00909451v1>

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.

Improvement of moving objects detection in continued all-day illumination conditions using color invariants and color spaces

Julian Murgia¹, Cyril Meurie² and Yassine Ruichek¹

Abstract—This paper presents the joint utilisation of the common subtraction technique "Codebook" and color invariants in order to perform a better detection of moving objects while dealing with difficult scenes, thus allowing a good tracking. We will observe the effects of the different invariants we tested on images captured in outdoor scenes at multiple moments of the day when we have to deal with important illumination and color changes. Our objective is to show the effectiveness of the common use of color invariants with a widely-used background subtraction algorithm, and give a preference rank of use.

I. INTRODUCTION

The concept of background subtraction (BGS) regroups multiple methods allowing effective detection of moving objects in an image.

BGS regroups very useful techniques for transports applications such as obstacle detection by intelligent vehicle, environment perception to improve safety at level crossing, detection of suspect persons or potentially dangerous scenes in transport environment [9], [19], [20]. The basic idea of BGS is to determine a background model representing the image parts which do not correspond to moving objects, then to perform the subtraction (or difference) between the current image against the background model.

Many background modeling methods were proposed since the last past years [4], [5], [17]. Simplest methods consist in a single unimodal distribution [15], [23]. But these methods show bad performances on multiple backgrounds (such as waving trees foliages) and fast illumination changes happening in the scene, causing lots of false detections. Another widely used method is the Generalized Mixture Of Gaussians (MOG) [21] which consists in a multi-modal distribution of gaussians for each pixel, which allows the modelisation of more complex backgrounds. This model is able to adapt more-or-less quickly to the background, depending on the learning rate value set. Basically, if the learning rate is too low, the model can't adapt to changes fast enough and be able to detect sudden changes in the background, resulting in a higher false positive rate. On the contrary if learning rate is too high, then slow moving objects may be included in the model, resulting in a high false negative rate. Kim et al. proposed the Codebook algorithm [16] which doesn't

make parametric assumption and which offers usually very good results. That is why, this algorithm (which is recalled in more details in Section II) has been chosen as a base method for our experimentations and study.

BGS methods usually work in 2 phases : learning and updating. The learning phase aims at the creation of the background model that input images will be compared against. The updating phase responds to the need of taking account of background changes over time. Although a lot of these methods now require few memory, updating a model remains a costly phase that we would like to happen as few as possible. Considering this and knowing the colors variations problem related to the fact that we work on image acquired in an outdoor environment and at multiple moments of the day, we propose to address it with the simple use of color invariants.

Objects recognition is highly dependent of viewpoint, object geometry and illumination. Color constancy is a concept that makes use of color invariance. It assumes that the color of an object in an image depends both on intrinsic parameters of its surface and material, and also on the color of the light source. Many research has been done to determine the color information that characterize objects color constancy [14], [22]. Computational models are often described by the determination of the light source color. Theoretically, by filtering the input image with the illuminant color, it is possible to obtain a normalized image representing the objects constant color (assuming light color to be spatially uniform). Section III will briefly present the different color invariants we tested. In Section IV, we present the experimental results of the detection of moving objects on a real database according to the codebook algorithm coupled to the use of color invariants.

II. CODEBOOK

Background modeling of a scene by the Codebook method proposed by Kim et al. [16] became a widely-used reference in the background subtraction for mobile objects tracking field of research. This method is known to be robust and efficient in a large variety of application cases such, including dynamic backgrounds (such as tree foliages, fountains, flags, shores...) and small illumination changes.

Kim et al. observed that false detections generally stick around dark regions of the image. However, the fact is that dark pixels suffer from a higher color ratio uncertainty due to the lower brightness of these pixels.

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

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

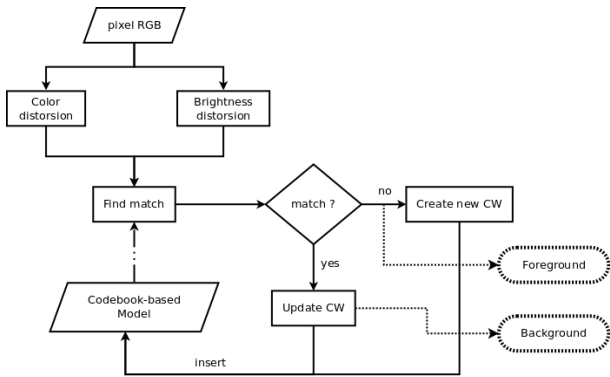


Fig. 1. Flow chart of the Codebook algorithm (plain lines correspond to learning phase while dotted lines correspond to foreground detection phase).

Therefore, brightness should be used as a factor in the comparison of color ratios. Another observation shows that for one pixel of fixed value, observer under variable illuminations over time, the observed values for this pixel are mostly distributed along a line aligned with the origin point (0,0,0). These considerations motivated the creation of a new color model allowing a separate evaluation of color distortion and brightness distortion as shown in Figure 1.

A. Learning step

The codebook method consists in an image partitioning in order to build a background model during a learning period. This model is represented by a list of codebooks (1 per pixel) containing N codewords. A codeword is created (or updated if the pixel appears close enough to an existing codeword) at each learning iteration. It is defined by two vectors containing respectively RGB values of the codeword, and multiple other data such as minimum and maximum brightness values, temporal variables and occurrence frequency of the codeword.

During learning step, a new codeword obtained from a given pixel is integrated in the background model if it satisfies two conditions: 1) brightness distortion constraint and 2) color distortion constraint. Precisions can be found in authors' article.

In the end of this step, the background model is "cleaned" of codewords that were most probably belonging to foreground objects during learning. This cleaning makes use of the maximum negative run-length (MNRL) value defined as the longest interval during the training period that the codeword has NOT recurred. That is, if the a codewords's MNRL is low, it means that this pixel is frequently observed. On the contrary, a high MNRL means that the pixel is less frequently observed (and it is probably not part of the background) and should then be filtered out from the model. The difference of this value from the frequency value is that even if a pixel has a large frequency AND a large MNRL means that it is

most probably a foreground object that stayed still for a moment.

B. Foreground detection

The background model obtained represents the parts of the image that do not move. It is then possible to use it to characterize each pixel of a new image. More simply, we verify the existence of codeword that can match this pixel in this pixel's codebook. This match test is done with the same constraints tests 1 and 2 described above, though the color distortion distance threshold ϵ can be set to another value for this step (which we did during our tests). If a match exists, then the pixel is classified as background (in black color in 2) and the matching codeword is then updated with new temporal and frequency values. If no match exists, then the pixel is classified as foreground (in white color in 2).

III. COLOR INVARIANTS

Color constancy considers that we can perceive the same color of an object, despite differences in the color of the light source. This technique requires to estimate the color of the light source (illuminant estimation). Existing methods to perform color constancy are generally sorted into three groups :

- 1) Using low-level features: these methods are either based on low-level statistics or physics-based dichromatic reflection model;
- 2) Using a learning phase: the illuminant is determined using a model learned on training data;
- 3) Using a combination of methods

All color constancy algorithms do not result in the same image. They often need to create a new color model adapted to features. Nevertheless, none of these algorithms can be considered as universal. Many works consist in a combination and a selection of multiple strategies: [13] showed that different color constancy algorithms perform better on different types of images ; [3] proposed an algorithm to select the best color invariant for one image using a CART methodology and a decision forest. Our work currently tries to observe the effects of a certain number of low-level features invariants on the segmentations obtained from Codebook algorithm.

13 color invariants were tested : Chromaticity space (also known as Normalized RGB), Greyworld normalization [6], Comprehensive normalization [11], Affine normalization [18], c1c2c3, m1m2m3 and l1l2l3 color spaces [12], RGB-Rank [10], YIQ color space [7], YCbCr color space [1], [2], YCh1Ch2 color space [8] and CIE L*a*b* color space.

IV. EXPERIMENTAL RESULTS

A. Experimental protocol

As explained in Section I, we propose to improve the segmentation results based on the Codebook algorithm by using a specific color invariant on outdoor images taken at different moments of the day. A street scene was

shot with a stereoscopic camera (BumbleBee 2) which only right side images were used, resized at 640x480 for processes. Multiple acquisitions at different times of the day (see the first line of Figure 2) were done in order to observe very different and high color and intensity variations. It is important to precise that the camera was set to disable automatic camera gain corrections: images of one base were recorded with fixed values. However, these values could not be set once and for all times of the because of the difficulty to have one setting allowing a good perceptions at all cases - in particular, differents values were set for dark times of day.

More accurately, shutter and gain parameters had to be set to values that would stay as static as possible all the day long. All other parameters (sharpness, hue, gamma...) were disabled.

To show the effectiveness of this method, results were calculated with very dark scenes as well as lit scenes. Times of shooting are 7.30am (daybreak), 9.30am, 12.00am (zenith), 3.00pm and 4.30pm (twilight). One can notice that our recordings were done in winter so days are shorter.

Each base of images shows people walking through the scene according to a scenario involving occlusions between people and/or background, people staying immobile for some time, people moving away and approaching the camera. At some times, one can observe vehicles driving along the street and other people wandering here and there (who were not involved in the scenario). Processings were done after all bases were recorded.

Each base was considered as a learning base for this one base and also for the following bases. That is, a learning applied to 12.00am base will be used to process 12.00am base and following on to 4.30pm, but not before. This way, experimental tests are led to simulate an on-line functionment. Let us denote chronological learning bases from L1 to L5 (from 7.30am to 4.30pm) and chonological process bases from P1 to P5. Thus, a test involving a learning on base L2 and processing base P4 is named L2_P4, and so on.

Parameters of the Codebook algorithm are: 50 learning images per base, $\alpha = 0.4$, $\beta = 1.2$, $\epsilon_1 = 0.2$, $\epsilon_2 = 100$. Each test was processed in 3 steps:

- 1) Apply a color invariance operator on learning base;
- 2) Build the Codebook model on this learning base;
- 3) Process each test base with the same color invariance operator as learning base, starting from learning base.

The background model obtained at step 2 is never updated for our tests at any moment. We count on color invariant features to remove the effects of the differences between learning images and tested images.

B. Evaluation criteria

The result segmentations images were evaluated according to ground truths reference segmentations made by an expert. Considering True Positives (TP) the pixels

correctly classified as foreground, True Negatives (TN) the pixels correctly classified as background, False Positive (FP) the pixels erroneously classified as foreground and False Negative (FN) the pixels erroneously classified as background. The commonly used measure value F-Measure (FM_c) was calculated for both foreground (fg) and background (bg), which is as follows (Eq. 1):

$$\begin{aligned} Recall_{fg} &= \frac{TP}{TP + FN}, \\ Recall_{bg} &= \frac{TN}{TN + FP}, \\ Precision_{fg} &= \frac{TP}{TP + FP}, \\ Precision_{bg} &= \frac{TN}{TN + FN}, \\ FM_c &= \frac{2.(Recall_c.Precision_c)}{Recall_c + Precision_c} \end{aligned} \quad (1)$$

Where c is either bg (background) or fg (foreground). F-Measure FM_c is the harmonic mean of Recall and Precision. The greater the F-Measure, the better the segmentation quality. This value was determined for the two classes foreground and background separately because this method shall perform more a better background segmentation than a better foreground segmentation. As the results following will demonstrate, learning and processing different bases at the time without using any color invariant causes very bad detection, as a huge part of the image is classified as foreground. Thus, it is presumed this method was more likely to reduce false detection rate than increasing good detection rate.

Having these values for each invariant, the efficiency of one color invariant with another needs to be determined. To do so, the global mean F-Measure $\overline{FM_{ci}}$ value is calculated for each color invariant ci using both foreground F-Measure FM_{fg} and background F-Measure FM_{bg} (Eq. 2). That is :

$$\overline{FM_{ci}} = \frac{\overline{FM_{bg,ci}} + \overline{FM_{fg,ci}}}{2} \quad (2)$$

Obviously, good results are expected in the case the same base is used for processing as for learning (Lx_Px). To avoid bias in the results, these values were not taken into account for the mean F-Measure calculation, and only results for Lx_Py where $x \neq y$ were used.

C. Results

Figure 2 illustrates respectively the segmentation results for every process base (7.30am, 9.30am, 12.00am, 3.00pm, 4.30pm) when learning is previously done on base 1 (7.30am). Each line shows the result among a selection of different color invariant (RGB-rank, L*a*b*, comprehensive normalization and without color invariant).

In this figure, one can notice that RGB-Rank (line 3) gives the best results, both for foreground and background detection. Comprehensive Normalization (line

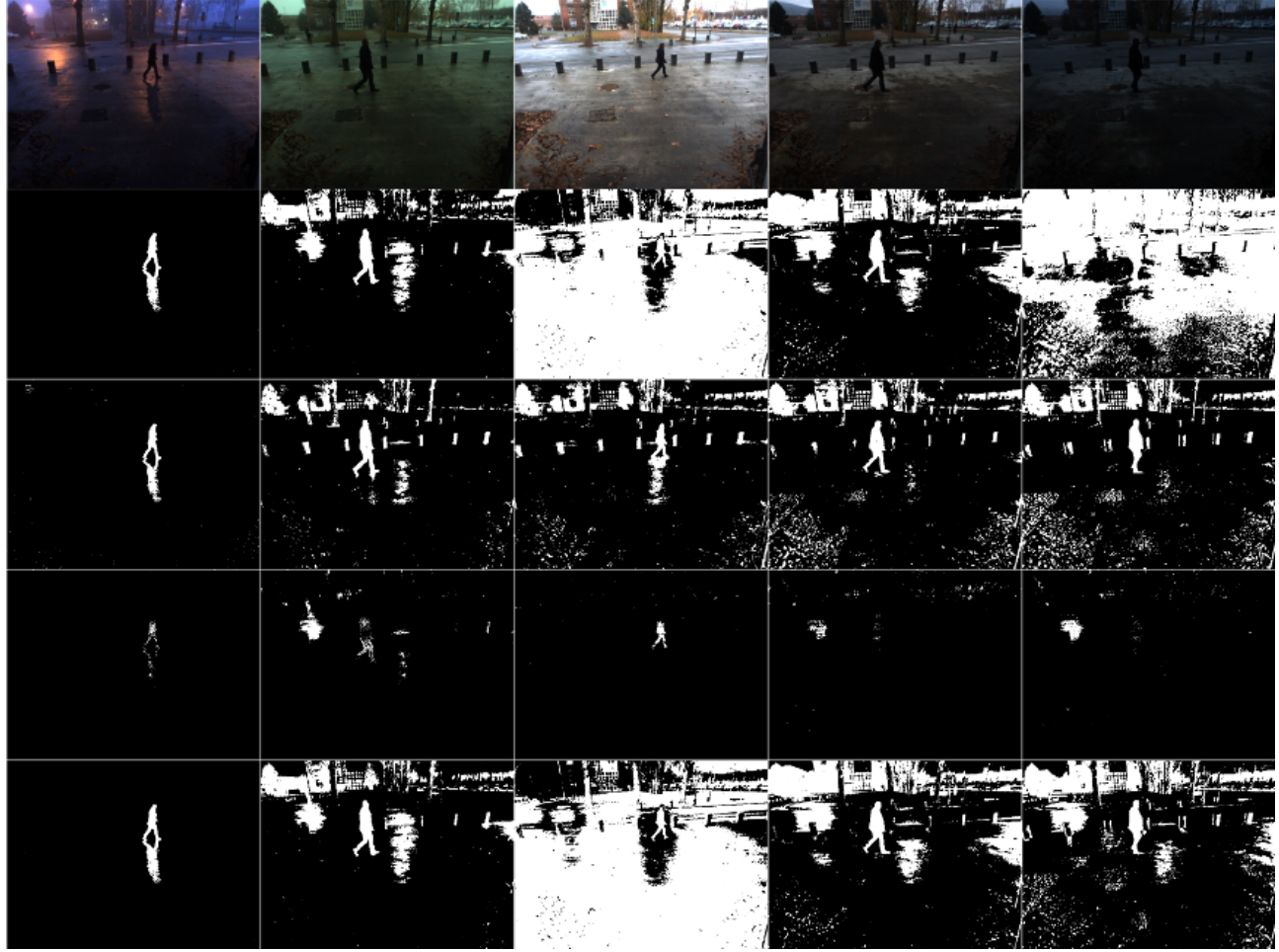


Fig. 2. Codebook segmentations (Learning base 1). From left to right : test images from base 1 to base 5 (7am to 4.30pm). Line 1 : RGB images; Line 2 : segmentations without color invariant ; Line 3 : RGB-Rank ; Line 4 : L*a*b* color space ; Line 5 : Comprehensive Normalization

TABLE I
F-MEASURES FOR FOREGROUND PIXELS (FM_{fg})

Base	Affine	c1c2c3	Chroma	Compr	GW	HSL	L*a*b*	l1l2l3	m1m2m3	none	RGBrank	YCbCr	YIQ
L1_P1	33,6%	0,6%	0,0%	63,8%	64,6%	1,6%	1,9%	8,5%	6,3%	62,7%	56,6%	0,4%	0,4%
L1_P2	6,2%	3,1%	2,0%	13,2%	12,6%	4,5%	10,8%	4,2%	4,2%	13,0%	16,9%	2,0%	1,9%
L1_P3	8,4%	5,5%	6,0%	4,9%	4,6%	3,6%	14,5%	6,5%	7,6%	4,6%	18,2%	4,7%	4,3%
L1_P4	5,7%	4,1%	3,0%	8,7%	9,0%	1,0%	13,2%	2,6%	3,4%	9,2%	13,3%	1,3%	1,9%
L1_P5	5,6%	6,1%	3,5%	8,7%	3,7%	2,4%	12,4%	3,1%	4,2%	3,8%	14,5%	2,4%	1,9%
Mean L1	11,9%	3,9%	2,9%	19,8%	18,9%	2,6%	10,6%	5,0%	5,1%	18,6%	23,9%	2,2%	2,1%
Rank L1	5	9	10	2	3	11	6	8	7	4	1	12	13
L2_P2	36,2%	7,5%	2,2%	74,4%	74,5%	19,5%	20,0%	56,6%	20,2%	74,7%	48,0%	5,2%	5,1%
L2_P3	8,0%	6,9%	3,3%	4,7%	4,3%	5,9%	17,0%	5,4%	8,6%	4,3%	21,2%	7,6%	4,8%
L2_P4	5,3%	9,2%	5,0%	7,6%	10,8%	1,4%	17,8%	2,2%	4,7%	11,2%	13,0%	2,0%	2,1%
L2_P5	5,8%	8,7%	2,5%	7,8%	2,5%	2,9%	15,7%	2,7%	9,7%	2,9%	12,8%	1,4%	1,9%
Mean L2	13,8%	8,1%	3,3%	23,6%	23,1%	7,3%	17,6%	16,7%	10,8%	23,3%	23,7%	4,1%	3,5%
Rank L2	7	9	13	2	4	10	5	6	8	3	1	11	12
L3_P3	48,5%	6,5%	3,6%	65,0%	64,8%	16,1%	44,5%	27,3%	26,4%	64,9%	52,7%	33,4%	29,9%
L3_P4	4,4%	8,3%	4,4%	2,2%	2,2%	2,2%	46,3%	2,2%	3,4%	2,2%	10,7%	8,5%	7,3%
L3_P5	5,1%	7,7%	2,6%	2,7%	2,7%	2,8%	28,0%	2,6%	4,1%	2,7%	8,5%	5,1%	6,9%
Mean L3	19,3%	7,5%	3,5%	23,3%	23,2%	7,0%	39,6%	10,7%	11,3%	23,2%	24,0%	15,7%	14,7%
Rank L3	6	11	13	3	5	12	1	10	9	4	2	7	8
L4_P4	45,0%	7,7%	3,8%	78,9%	78,8%	26,8%	15,7%	10,3%	21,2%	78,8%	46,8%	1,4%	1,4%
L4_P5	8,3%	6,7%	2,4%	17,8%	2,9%	3,5%	22,5%	2,5%	6,0%	16,7%	2,1%	2,1%	1,7%
Mean L4	26,7%	7,2%	3,1%	48,4%	40,8%	15,1%	19,1%	6,4%	13,6%	40,8%	31,7%	1,7%	1,5%
Rank L4	5	9	11	1	2	7	6	10	8	3	4	12	13
L5_P5	33,2%	2,6%	0,5%	52,8%	78,9%	9,2%	16,0%	12,9%	19,0%	78,6%	56,5%	4,2%	2,5%
Rank L5	5	11	13	4	1	9	7	8	6	2	3	10	12

TABLE II
F-MEASURES FOR BACKGROUND PIXELS (FM_{bg})

Base	Affine	c1c2c3	Chroma	Compr	GW	HSL	L*a*b*	l1l2l3	m1m2m3	none	RGBRank	YCbCr	YIQ
L1_P1	91.2%	91.4%	91.4%	91.6%	91.6%	91.2%	91.4%	90.9%	89.8%	91.6%	91.5%	91.4%	91.4%
L1_P2	78.3%	87.9%	90.4%	86.6%	86.1%	81.2%	90.9%	74.4%	79.4%	86.3%	88.1%	90.3%	90.2%
L1_P3	66.0%	89.0%	90.9%	34.8%	29.3%	66.5%	91.4%	69.2%	77.6%	27.8%	87.9%	81.3%	78.3%
L1_P4	76.9%	91.5%	92.5%	83.1%	83.6%	69.7%	92.6%	70.1%	77.4%	84.0%	88.0%	92.2%	91.8%
L1_P5	74.6%	92.1%	92.5%	71.4%	46.6%	88.3%	92.0%	69.7%	77.5%	47.3%	87.3%	92.0%	91.4%
Mean L1	77.4%	90.4%	91.6%	73.5%	67.4%	79.4%	91.7%	74.8%	80.3%	67.4%	88.6%	89.5%	88.6%
Rank L1	9	3	2	11	12	8	1	10	7	13	6	4	5
L2_P2	90.5%	91.1%	91.1%	91.4%	91.4%	91.1%	91.2%	91.3%	89.8%	91.4%	91.0%	91.1%	91.1%
L2_P3	66.2%	91.2%	91.2%	23.6%	13.8%	79.9%	91.4%	27.4%	80.7%	13.6%	88.5%	87.9%	83.5%
L2_P4	77.0%	92.8%	92.8%	81.1%	85.7%	69.3%	92.8%	26.9%	82.6%	86.2%	87.8%	92.6%	92.6%
L2_P5	73.9%	92.6%	92.6%	60.1%	19.8%	84.6%	92.6%	22.2%	83.4%	20.4%	86.1%	91.5%	86.2%
Mean L2	76.9%	91.9%	91.9%	64.0%	52.7%	81.2%	92.0%	42.0%	84.1%	52.9%	88.4%	90.8%	88.4%
Rank L2	9	2	3	10	12	8	1	13	7	11	6	4	5
L3_P3	90.2%	91.2%	91.9%	91.8%	91.8%	89.6%	91.9%	90.8%	90.2%	91.8%	89.2%	91.5%	91.5%
L3_P4	68.0%	92.8%	91.2%	2.7%	2.2%	74.4%	93.0%	26.9%	76.7%	2.2%	86.6%	87.8%	89.6%
L3_P5	66.4%	92.6%	92.8%	3.5%	0.1%	81.4%	87.5%	74.2%	68.9%	0.2%	81.5%	66.8%	76.8%
Mean L3	74.9%	92.2%	92.0%	32.7%	31.4%	81.8%	90.8%	64.0%	78.6%	31.4%	85.8%	82.0%	86.0%
Rank L3	9	1	2	11	12	7	3	10	8	13	5	6	4
L4_P4	92.6%	92.8%	92.8%	93.2%	93.2%	92.7%	92.8%	90.8%	91.8%	93.2%	92.7%	92.8%	92.8%
L4_P5	80.3%	92.6%	92.6%	67.7%	21.1%	79.3%	92.6%	80.0%	81.4%	21.3%	88.3%	91.0%	91.8%
Mean L4	86.5%	92.7%	92.7%	80.5%	57.1%	86.0%	92.7%	85.4%	86.6%	57.2%	90.5%	91.9%	92.3%
Rank L4	8	3	2	11	13	9	1	10	7	12	6	5	4
L5_P5	86.1%	92.7%	92.6%	61.2%	93.1%	92.0%	92.7%	90.5%	90.8%	93.1%	92.8%	92.6%	92.6%
Rank L5	12	5	6	13	2	9	4	11	10	1	3	7	8

5) also provides interesting results (except for base P3=12.00am illustrated in the third column) though it still shows many false positives. These results are better than those obtained without color invariant illustrated in line 2 (for example: the segmentation results of the base P5=4.30pm given in the column 5). L*a*b* color space gives much less interesting results because of the poor foreground detection (though the following statistical analysis denotes its qualities), but the background appear much less noised than other results. One can then hypothesize on its usage to allow noise suppression in the background. In comparison, the non-use of any color invariant (line 2) gives very bad results for P3=12.00am and P5=4.30pm, showing many false positives in the background. Interestingly enough, the quality of the segmentations obtained with RGB-Rank looks similar for each test base, with false positives always situated in the same regions of the images.

Let us compare the results shown on Tables I and II which illustrate the F-Measure values obtained respectively for foreground classifications (FM_{fg}) and for background classifications over time (FM_{bg}). A separate observation is needed because in the case of L1, F-Measure means are calculated for much more lighting conditions (with 7.00pm + 9.00pm + 12.00am + 15.30am + 16.30am) than L4 for example (with 15:30am + 16:30am). Thus, to compare the efficiency of each invariant, the mean F-Measure was calculated for each test base and for each color invariant, that is: one mean value per L_x group. These values appear after each group of tests. Then, as the higher the F-Measure, the better the segmentation, these values were ranked so we are able to compare each color invariant's ranking.

Considering foreground F-Measures given in Table I, one can observe that RGB-Rank color invariant shows the best performances in L1 and L2 learnings (23.9% and 23.8% respectively). L*a*b* is first for L3 (39.6%) while RGB-Rank is second (24%). Then for L4, Comprehensive Normalization shows the better performances (48.4%)

compared to Greyworld and without invariant (40.8%) and RGB-Rank (31.7%) follow. Finally for L5, Greyworld offers the best performances (78.9%), but this case is tested with the same base as for learning so these good results are not surprising and implies a improvement of segmentation results which is limited in comparison to the results obtained without invariant. Though, it is important to notice that for all these top 4 color invariants, the range of mean F-Measures was pretty small in each case.

Comparing the best color invariants of this ranking together, it appears that RGB-Rank was classified 1st two times on five and was never classified under rank 4. Comprehensive Normalization was classified 1st one time, but in all other cases it was under RGB-Rank. The same observation is made for Greyworld and L*a*b color space, which allows us to think that RGB-Rank shows the better foreground detection most of the time.

Visually, Figure 2 effectively shows that foreground segmentation appear very similar (considering only foreground) for RGB-Rank and Comprehensive Normalization. But, the differences lie more in background segmentations and the false positives.

Now considering background F-Measures given in Table II, the hypothesis is that a high F-Measure shows a low False Positive Rate. It appears that L*a*b* color space is ranked 1st three times on five with all of its values never lying under 89%. This confirms the visual observation of the good background segmentation it can obtain. It is followed by Chromaticity Space and c1c2c3, which also show very high F-Measures for background (all around 90%), close to those of L*a*b* color space. But, Chromaticity Space and c1c2c3 show poor F-Measures for foreground, which don't confirm them to be interesting enough. In the contrary, Comprehensive Normalization which performed interesting foreground detection, is ranked in the bottom of the table. To conclude on this background F-Measures, RGB-Rank is placed in the middle of the rankings.

All these results allow us to confirm the usage of color invariants to perform background segmentations during all day with only one images learning base. Even if these segmentations are not perfect, they remain very interesting while providing in the best cases, much less false detections with a color invariant than without. In most cases, one can observe that false positives lie in background part that are subject to occasional illuminations: metal stakes and streelights are partially lit because of car headlights. Their illumination also change between the different moments of the day, which cause them to be detected as foreground. Finally, one can conclude that it is useful to involve RGB-rank as color invariant to the codebook algorithm in order to obtain the best detection of moving object for continued all-day.

V. CONCLUSIONS & FUTURE WORK

In this paper, we have proposed to combine the use of color invariants and color spaces with a codebook-based background subtraction method. This proposed strategy permits better segmentations results on images that do not correspond to the current learning state, allowing the codebook algorithm to rarefy the learning update phase and increase the detection of moving objects. Three color invariants come out on top: RGB-Rank, Greyworld, Comprehensive Normalization, and $L^*a^*b^*$ which will be used in future works to reduce noise in background detections.

These results illustrate the common use of color invariants with a codebook-based background subtraction method to provide better segmentations results on images that don't correspond to the current learning state. This allows the codebook algorithm to rarefy the learning update phase. However, some situations need a proper background model update, especially to take care of mobile objects that stay still for a long time (parked cars, stationary pedestrians...).

Though, our experiments showed that background subtraction is possible in the case of an on-line process at different moments of the day, allowing the process to work during all-day (7.00pm to 4:30am in this paper) and do not require an important processing time. Finally, we conclude that RGB-Rank is the best color invariant to use with the codebook algorithm in order to increase the detection of moving objects whatever the illuminations changes of the day.

These observations motivate us to improve the codebook method with a different way to deal with illumination and color changes. The color model used by the Codebook method proposed by [16] can be modified to apply one or multiple color invariants according to the scene situation. Additionally, a comparison of this method with other background subtraction methods used with color invariants would support our present results. Future works should also demonstrate the influence of these works on tracking methods.

REFERENCES

- [1] Recommendation itu-r bt.601-5, 1982-1995.
- [2] Recommendation itu-r bt.709-5, 1990-2002.
- [3] S. Bianco, G. Ciocca, C. Cusano, and R. Schettini. Automatic color constancy algorithm selection and combination. *Pattern Recognition*, 43(3):695 – 705, 2010.
- [4] Thierry Bouwmans and Fida El Baf. Statistical background modeling for foreground detection: A survey. *Handbook of Pattern Recognition and Computer*, 2010.
- [5] Thierry Bouwmans, Fida El Baf, Bertrand Vachon, et al. Background modeling using mixture of gaussians for foreground detection-a survey. 2008.
- [6] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310(1):1 – 26, 1980.
- [7] Walter H. Buchsbaum. *Color TV Servicing, third edition*. Englewood Cliffs, NJ: Prentice Hall, ISBN 0-13-152397-X, 1975.
- [8] T Carron. *Segmentation d'images couleur dans la base Teinte Luminance Saturation : approche numerique et symbolique*. PhD thesis, Universite de Stanford, 1995.
- [9] 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.
- [10] Graham D. Finlayson, Steven D. Hordley, Gerald Schaefer, and Gui Yun Tian. Illuminant and device invariant colour using histogram equalisation. *Pattern Recognition*, 38(2):179–190, 2005.
- [11] Graham D. Finlayson, Bernt Schiele, and James L. Crowley. Comprehensive colour image normalization. 1998.
- [12] Theo Gevers and Arnold W.M. Smeulders. Color-based object recognition. *Pattern Recognition*, 32(3):453 – 464, 1999.
- [13] A. Gijsenij and T. Gevers. Color constancy using natural image statistics and scene semantics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(4):687 –698, april 2010.
- [14] A. Gijsenij, T. Gevers, and J. van de Weijer. Computational color constancy: Survey and experiments. *Image Processing, IEEE Transactions on*, 20(9):2475 –2489, sept. 2011.
- [15] Thanarat Horprasert, David Harwood, and Larry S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. *IEEE Frame-Rate Applications Workshop Kerkyra, Greece*, pages 1–19, 1999.
- [16] K. Kim, T. H. Chalidabhongse, D. Hanuood, and L. Davis. Background modeling and subtraction by codebook construction. 2004.
- [17] Alan M. Mcivov. Background Subtraction Techniques. 2000.
- [18] 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.
- [19] 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.
- [20] 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.
- [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] J. van de Weijer, T. Gevers, and A. Gijsenij. Edge-based color constancy. *Image Processing, IEEE Transactions on*, 16(9):2207 –2214, sept. 2007.
- [23] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfunder: real-time tracking of the human body. pages 51 –56, oct 1996.