



HAL
open science

All-Day moving objects detection for security at level crossing

Julian Murgia, Cyril Meurie, Yassine Ruichek

► **To cite this version:**

Julian Murgia, Cyril Meurie, Yassine Ruichek. All-Day moving objects detection for security at level crossing. TRA2014 - Transport Research Arena: Transport Solutions: from Research to Deployment - Innovate Mobility, Mobilise Innovation!, Apr 2014, Paris, France. 10p. <hal-01263466>

HAL Id: hal-01263466

<https://hal.science/hal-01263466v1>

Submitted on 27 Jan 2016

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.



HAL Authorization

All-day moving objects detection for security at level crossing

Julian Murgia (a), Cyril Meurie (b), Yassine Ruichek (a)

(a) IRTES-SeT, UTBM, 90010 Belfort Cedex, France

(b) Univ Lille Nord de France, F-59000 Lille, IFSTTAR, LEOST, F59650, Villeneuve d'Ascq, France

Abstract

In this paper, we propose a strategy based on the joint use of background/foreground segmentation methods and colorimetric invariants or color spaces, in order to detect more precisely moving objects at level crossings throughout the day. The proposed strategy is composed of three steps : 1/ apply an adapted colorimetric invariant on the acquired image in order to simplify the image and limit the brightness changes recorded throughout the day, 2/ use a common background subtraction algorithm (Codebook) on simplified images, 3/ track moving objects using a Kalman filter in order to visualize the benefit of this approach at the end of treatment. Results obtained illustrate the common use of color invariants with a Codebook-based background subtraction method in order to provide better segmentations results on images that do not correspond to the current learning state (compared to those obtained without the use of colorimetric invariant/color space). To show the effectiveness of this method, a mobile objects tracking is performed on obtained segmentations.

Keywords: Color invariants ; background subtraction ; tracking ; moving objects detection

Résumé

Dans ce papier, nous proposons une stratégie basée sur l'utilisation conjointe d'une méthode de segmentation fond/forme et d'invariants colorimétriques, afin de détecter plus précisément les objets mobiles dans l'environnement des passages à niveau, tout au long de la journée. La stratégie proposée se compose de 3 étapes : 1/ appliquer un invariant colorimétrique adapté sur l'image acquise afin de simplifier l'image et limiter les changements d'intensités observés au cours de la journée, 2/ utiliser une méthode de soustraction d'arrière-plan existante (Codebook par exemple) sur les images simplifiées (lors de la phase d'apprentissage et de test); 3/ suivre les objets mobiles à l'aide d'un filtre de Kalman afin de visualiser les bénéfices de cette approche sur la fin du traitement. Les résultats obtenus permettent d'illustrer l'intérêt de combiner un invariant colorimétrique avec une méthode de détection fond/forme basée sur l'algorithme Codebook pour permettre une meilleure segmentation des images lorsque celles-ci n'ont pas été acquises dans les mêmes conditions de luminosité que celles utilisées pour l'apprentissage (comparés aux résultats obtenus sans l'utilisation d'un quelconque invariant ou espace couleur). Pour montrer l'efficacité de cette méthode sur une opération de suivi de cibles, un tracking est effectué sur les segmentations obtenues avec et sans la méthode proposée.

Mots-clés : invariants couleur; soustraction d'arrière-plan, suivi, détection objets mobiles.



1. Introduction

Since many years, level-crossings (LC) are considered as sources of many incidents. In 2009, “Réseau Ferré de France” counted 36 fatal accidents (30 dead persons in vehicles and 6 pedestrians killed), and 120 collisions involving trains. In the majority of cases, the accident is caused by carelessness such as the non-compliance with a stop sign, a baffle passage between lowered barriers, rides between lanes in traffic jams implying stationarity on the LC, excessive speed of approach, little visibility around the LC, etc.

Several actions were led to increase security at level-crossing. The ANR project PANsafer 2009-2012 is one of these actions (Salmane et al., 2012a, 2012b). It proposed to develop and evaluate an intelligent, communicant system that allows the detection (thanks to a video perception module) of potentially dangerous situations occurring at level-crossing environment. Then, thanks to an infrastructure/vehicles communication module, it transmits a criticality information to all concerned actors (road users, train, centralized command station, etc).

The video perception module has to be able to work any moment of day. Common foreground segmentation methods from the literature seem to be adapted to this problematic, but they are very sensitive to important luminosity changes, which are amplified when data is acquired on a whole day from daybreak to twilight. These methods generally adapt themselves to the changes in the scene by updating their background model. Even if this update is needed to take into account the presence of mobile objects that actually stay immobile for a certain time (such as parked cars), this operation is costly in resources and time. As an example the Mixture of Gaussians method (Stauffer et al, 1999) needs 12 to 20 seconds to be fully adapted, according to our observations. During this amount of time, resulting segmentations include huge parts of the background as false positives, which make them difficult to use for another process. Other well-known BGS methods suffer from such problems, each one managing them with different efficiencies. In this context, we propose to increase the robustness of the global system by proposing a better mobile objects detection with the joint use of background/foreground segmentation and color invariants and/or color spaces. Resulting segmentations are purposely obtained with mismatching learning images and process images, causing unadapted learnings, in order to simulate the effects that need to be addressed.

To demonstrate the effectiveness of the proposed strategy and go further in the proposed strategy, a moving objects tracking algorithm is applied on resulting segmentations.

2. Method

2.1. Color invariance

Prior to background subtraction, each image is firstly simplified with a colorimetric invariant. Color invariance consists in determining the illuminant color of the image before “normalization” in order to obtain an image that theoretically displays the scene under a canonic illuminant, whatever the time of the day. Multiple methods exist in the literature to perform this task (Obdrzalek et al, 2003): some make use of low-level features (low-level statistic of physics-based), some make use of a learning phase, and finally some are a combination of both. However, each method doesn't produce the same result and none can be considered as “universally true”.

Gijsenij et al (2010) showed that certain operators were performing better on certain images than others. In this way, an algorithm was also proposed by Bianco et al (2010) to detect the best color invariant that fits to a given image. This is the reason why multiple operators were used : Chromaticity space (also known as Normalized RGB), Greyworld normalization (Buchsbbaum, 1980), Comprehensive normalization (Finlayson et al., 1998), Affine normalization (Obdrzalek et al, 2003), c1c2c3, m1m2m3 and l1l2l3 color spaces (Gevers et al. 1999), RGB-Rank (Finlayson et al., 2005), YIQ color space (Buchsbbaum, 1975), YCbCr color space, YCh1Ch2 color space (Carron, 1995) and CIE L*a*b* color space.

2.2. Background subtraction

Background subtraction techniques consist in creating a background model which is then used to perform a comparison between this model and an input image, resulting in a segmentation representing parts of the input



images that do not belong to the background. Multiple techniques exist in the literature (Bouwmans et al, 2010), such as the well-known Mixture of Gaussians algorithm proposed by Stauffer et al (1999) which consists in a multi-modal distribution of gaussians for each pixel, which allows the modeling of more complex backgrounds. This method is parametric, which led us to choose another method to perform our method.

The Codebook algorithm proposed by Kim et al. (2004) is a well-known method that performs a background subtraction on images taken from a still point of view in order to segment moving objects out of the background. It showed very good results on images containing many common difficult environments such as tree foliage, flags, fountains, sea shores as well as small illumination changes. This method works in two distinct phases : learning and processing.

The Learning step builds a model representing the background of learning images. This model consists in N codebooks (1 per pixel) containing various codewords that are computed (or updated) at each learning iteration when the pixel is supposed to be part of the background. More precisely, these codewords are defined by two vectors : the first one contains the RGB colour of the pixel, the second one contains multiple statistical and temporal data (such as minimum and maximum brightness, frequency and time of occurrence, MNRL...).

During this step, a codeword is integrated (or updated) to the model if it satisfies two conditions : a) brightness distortion constraint and b) color distortion constraint.

- a) The intensity of the pixel must lie in the interval $[I_{\min} ; I_{\max}]$ determined from all the minimum and maximum brightnesses 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.
- b) The color distortion *colordist*, calculated from the pixel x_t and the tested codeword v_i from the model, must lie under a given threshold ϵ . Following Formulas (1) and (2) define the calculation of color distortion.

$$p^2 = \frac{\langle x_t, v_t \rangle^2}{\|v_i\|^2} \quad (1)$$

$$\text{colordist}(x_t, v_i) = \sqrt{(x_t)^2 - p^2} \quad (2)$$

At the end of this step, the model is cleaned of the codewords that were most probably belonging to foreground. To do so, the algorithm make use of each codeword's MNRL (Maximum Negative Run-Length) value defined as the longest interval during the learning period that the codeword has not recurred. A low value means that the codeword was frequently observed. A high value means that it was less frequently observed and that it should be removed from the model as it was probably part of foreground. The MNRL is different from the frequency value because even if a pixel has a large frequency and a large MNRL, then it is most probably a foreground object that stayed still for a moment.

The Processing step that follows performs almost the same task as Learning. It simply consists in verifying the existence of a corresponding codeword in the model for the tested pixel of the processed image. To do so, the same procedure is applied to each pixel (application of the two constraints described above). If a match exists for this codeword in the model, then it means that the pixel belongs to the background. If no match is found, it means that this pixel is part of the foreground.

The general algorithm described by Kim et al updates the model during processing step to take into account small illumination changes in the image. In opposition to the general algorithm, we do not update the model during this step because we assume small (and important as well) illumination and color changes to be already dealt by the use of a colorimetric invariant. Thus, our method uses a unique background model to deal with all-day images and spares resources needed by the update of the model.

2.3. Multiple objects tracking

In order to show the advantages gained by the use of our method, we applied a multi-point tracking to the segmentations obtained. Tracked points are the barycenter of each region. This tracker consists in the generation



and/or the update at each frame of a Kalman Filter with the closest detected position to each Kalman Filter's prediction. The tracking system then draws a frame around each successfully tracked region.

Since our objective is not to perform a better tracking but a better detection of moving objects in complex conditions, we do not take into account loss of tracking. As moving objects can be aligned to the camera point of view, this case causes the two regions to be melted into one, which instantly causes loss of tracking. Despite this drawback, one can still easily observe the path of objects at each frame. Results presented in next section will show that in most cases, tracking is extremely poor when our method is not applied.

3. Experimental results

3.1. The first base: an urban scene

This method was tested on a set of images acquired with a video-surveillance objective. These images were taken at different moments of a day, capturing a similar scenario. Some examples are shown in Fig. 1 (from left to right: 7am, 9.30am, 12.00am, 3.00pm and 4.30pm). One can notice that these images were recorded in winter so days are shorter.

As we can see on Figure 1, if no color invariant is used, and when the learning step was made on different images than those processed, segmentation quality is pretty bad. Many regions of the images are erroneously detected as foreground. The actual foreground appears melted in false positive detections. The use of a colorimetric invariant then removes a lot of false positive detections in the background, while good detection is still performed on foreground. Moreover, even if these results appear better, false detections remain in the distant background.

F-Measures were calculated for both classes (foreground and background). Evaluations were done for each colorimetric invariant and color space (Rank line) in order to determine which one of them performs the best for both classes. Results provided in Tables 1 and 2 pointed out RGB-Rank colorimetric invariant to perform the best for this application.

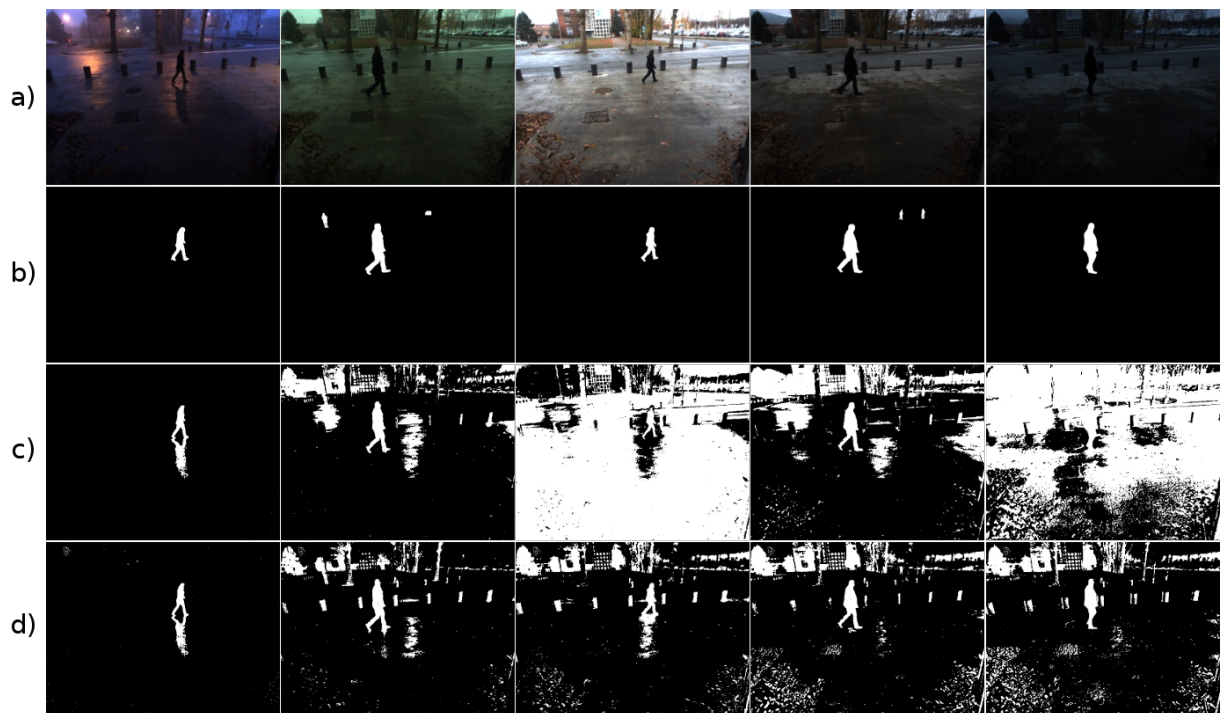


Fig 1: Segmentations obtained images for each time of day (with a learning on 7.30).

a) initial images in RGB color space b) ground truths images c) without colorimetric invariant d) with RGB-Rank



FM bg	Affine Norm	c1c2c3	Chromaticity	Compr. Norm	Greyworld	HSL	L*a*b*	l1l2l3	m1m2m3	none	RGB-rank	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

Table 1: F-Measures on background class for every color invariant/space tested.

FM fg	Affine Norm	c1c2c3	Chromaticity	Compr. Norm	Greyworld	HSL	L*a*b*	l1l2l3	m1m2m3	none	RGB-rank	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%	2,9%	2,5%	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%	2,9%	16,7%	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 2: F-Measures on foreground class for every color invariant/space tested.

This method has been tested on a second base of images sequences acquired at different moments of one day. These images, of size 640x480 were acquired in the PANsafer project with the use of a camera jAI CV-M9CL equipped with a 3-CCD sensor.

Recorded sequences consist in different scenarios reproducing critical situations involving dangerous users behaviours or unintentional blocking on level-crossings. Some of these scenarios consist for example in a vehicle parking on the level-crossing because of a blocking of one or more vehicles before it, or a chicane passage between closes barriers.



One can observe small color changes and illumination variations on these images ; unfortunately, the acquisitions were unable to show huge changes in the images. Nevertheless, we chose three images sequences that seemed to show enough differences to apply our method. Thus, the first images base (taken at 12am) named P1 shows a brighter aspect than the second one (taken at 4pm) named P2, and the third one P3 (taken around the same time, but the scene appears darker because of a cloud passing by the sun). From these three processing bases, we define three learning bases of 75 images each, respectively named L1, L2 and L3. From this naming rule, a test involving Learning base L1 and Process base P2 is called "L1_P2". Each test was processed multiple times with different color invariants operators. It is important to recall that for each test case, the same color invariant operator was used for both learning images and processed images. Also, the learning base is never updated after the Codebook Learning step.

Each test was driven with four steps:

- Apply a colorimetric invariant operator on learning base ;
- Build the Codebook model with learning images ;
- Process background subtraction after applying the same colorimetric invariant to processing base ;
- Perform tracking operator on segmented images.

Parameters used for the Codebook algorithms are the following : $\alpha = 0.4$, $\beta = 1.2$, $\epsilon_1 = 0.2$, $\epsilon_2 = 100$. These parameters are empiric values determined by Codebook algorithm authors and that provide good results most of the time, whatever the images.

3.2. Evaluation criteria

Resulting segmentations were evaluated with the use of ground truths segmentations. For each segmentation, 4 pixel cases are considered:

- True Positive (TP) : the pixel is correctly classified as foreground ;
- True Negative (TN) : the pixel is correctly classified as background ;
- False Positive (FP) : the pixel is incorrectly classified as foreground ;
- False Negative (FN) : the pixel is incorrectly classified as background.

F-Measure (FM_c) value was then calculated for both classes foreground (fg) and background (bg) (Formula (3)).

$$\begin{aligned} Recall_{fg} &= \frac{TP}{TP+FN}, Recall_{bg} = \frac{TN}{TN+FP} \\ Precision_{fg} &= \frac{TP}{TP+FP}, Precision_{bg} = \frac{TN}{TN+FN} \\ FM_c &= \frac{2 \cdot (Recall_c \cdot Precision_c)}{Recall_c + Precision_c} \end{aligned} \quad (3)$$

Having c either bg or fg . F-Measure Fm_c is the harmonic mean of Recall and Precision. The greater the value, the better the segmentation quality. Since our method is meant more to reduce the False Positive Rate (so reducing the number of regions detected as foreground) that to augment the True Positive Rate, we wanted to observe its effects on both classes separately.

3.3. Background subtraction results

Since three images sequences seemed to be interesting enough for the tests to be driven, we denote three different tests with a different name for each, which we present the results in Tables 3 and 4, the same way they were presented in Part 3.1.. These results are averages obtained with a comparison against ground truth references made at hand by a specialist. A high value denotes a good segmentation quality, 100% being the exact same quality as the reference (for the given class only).

Table 3 shows F-Measures obtained for background. One can observe that, similarly to previous results, Chromaticity space color invariant shows best results. But, a look at values in table 4 denotes very poor detection rates in foreground class, which means that segmented images are almost all black. Interesting results lie in HSL



column. Forgetting uninteresting results (Chromaticity, c1c2c3, L*a*b, YCbCr), HSL shows better or equivalent results than no-operator used in the case of different learning base and processing base (L_xPy, with x ≠ y). The same way, RGB-Rank performs very good as well, even if its ranks for background (6th, 9th, 6th) and foreground (2nd, 5th, 5th) make it appear in the middle of the rankings, its evaluations lay above 80% for background and close to 5% for foreground.

FM bg	Affine Norm	c1c2c3	Chromaticity	Compr. Norm	Greyworld	HSL	L*a*b*	I1I2I3	m1m2m3	none	RGB-rank	YcbCr	YIQ
L1_P1	93,01%	94,12%	94,12%	94,34%	94,34%	94,19%	93,35%	94,02%	92,78%	94,34%	94,30%	48,53%	91,59%
L1_P2	81,63%	96,43%	96,43%	58,05%	58,66%	96,35%	95,69%	83,31%	91,52%	61,04%	92,50%	25,00%	95,45%
L1_P3	78,06%	94,27%	94,30%	75,37%	78,44%	90,46%	93,28%	87,56%	84,53%	79,50%	83,70%	20,41%	90,27%
Mean L1	79,8%	95,4%	95,4%	66,7%	68,5%	93,4%	94,5%	85,4%	88,0%	70,3%	88,1%	22,7%	92,9%
Rank L1	9	2	1	12	11	4	3	8	7	10	6	13	5
L2_P1	78,73%	94,13%	94,12%	93,49%	93,59%	93,87%	89,48%	88,88%	91,62%	93,58%	90,30%	3,90%	19,85%
L2_P2	95,51%	96,43%	96,43%	96,69%	96,68%	96,67%	95,60%	96,53%	95,00%	96,68%	96,55%	49,39%	88,25%
L2_P3	78,11%	94,30%	94,33%	91,75%	91,88%	90,98%	91,07%	78,81%	85,76%	91,99%	84,06%	6,52%	68,00%
Mean L2	78,4%	94,2%	94,2%	92,6%	92,7%	92,4%	90,3%	83,8%	88,7%	92,8%	87,2%	5,2%	43,9%
Rank L2	11	2	1	5	4	6	7	10	8	3	9	13	12
L3_P1	81,15%	94,01%	94,07%	91,17%	91,27%	92,02%	92,19%	86,21%	83,54%	91,41%	84,73%	12,39%	84,89%
L3_P2	82,63%	96,14%	96,35%	76,74%	76,80%	94,77%	94,89%	66,22%	83,67%	78,24%	85,50%	6,76%	92,87%
L3_P3	85,66%	94,35%	94,36%	91,51%	92,82%	94,14%	93,59%	92,90%	92,33%	93,16%	93,87%	55,81%	94,18%
Mean L3	81,9%	95,1%	95,2%	84,0%	84,0%	93,4%	93,5%	76,2%	83,6%	84,8%	85,1%	9,6%	88,9%
Rank L3	11	2	1	9	8	4	3	12	10	7	6	13	5

Table 3: F-Measures values for the background class

FM fg	Affine Norm	c1c2c3	Chromaticity	Compr. Norm	Greyworld	HSL	L*a*b*	I1I2I3	m1m2m3	none	RGB-rank	YcbCr	YIQ
L1_P1	14,64%	0,85%	0,28%	31,12%	31,29%	23,51%	2,87%	28,07%	9,11%	31,80%	28,71%	4,63%	10,79%
L1_P2	7,69%	2,33%	1,44%	3,49%	3,50%	25,80%	4,74%	8,36%	11,82%	3,60%	14,06%	5,14%	24,56%
L1_P3	3,55%	5,99%	4,33%	4,09%	4,42%	9,29%	8,25%	5,61%	3,23%	4,62%	4,28%	2,77%	9,55%
Mean L1	5,6%	4,2%	2,9%	3,8%	4,0%	17,5%	6,5%	7,0%	7,5%	4,1%	9,2%	4,0%	17,1%
Rank L1	7	8	13	12	10	1	6	5	4	9	3	11	2
L2_P1	6,18%	4,15%	1,32%	35,02%	36,27%	34,70%	9,78%	9,35%	9,84%	35,66%	13,36%	3,42%	2,20%
L2_P2	18,41%	1,79%	1,15%	29,11%	28,99%	21,47%	9,55%	25,41%	14,88%	28,53%	26,28%	5,99%	6,93%
L2_P3	4,53%	5,72%	3,05%	15,93%	16,32%	11,07%	9,69%	3,04%	3,55%	16,76%	4,92%	2,53%	3,94%
Mean L2	5,4%	4,9%	2,2%	25,5%	26,3%	22,9%	9,7%	6,2%	6,7%	26,2%	9,1%	3,0%	3,1%
Rank L2	9	10	13	3	1	4	5	8	7	2	6	12	11
L3_P1	4,91%	1,87%	0,45%	14,78%	14,96%	15,79%	5,16%	7,86%	2,97%	15,44%	6,86%	3,56%	6,11%
L3_P2	8,46%	2,18%	1,60%	5,19%	5,18%	19,36%	3,43%	5,09%	6,28%	5,40%	7,52%	4,66%	14,78%
L3_P3	3,81%	1,23%	0,77%	15,50%	18,99%	18,97%	2,54%	11,21%	5,44%	20,28%	21,54%	3,78%	20,11%
Mean L3	6,7%	2,0%	1,0%	10,0%	10,1%	17,6%	4,3%	6,5%	4,6%	10,4%	7,2%	4,1%	10,4%
Rank L3	7	12	13	5	4	1	10	8	9	3	6	11	2

Table 4: F-Measures values for the foreground class

One can also observe good results obtained with no invariant used (“none” column). F-Measures remain close to HSL ones, which we can explain by the relative similarity of tested images from one base to another. Codebook algorithm uses its own color model (color distortion and brightness distortion), which is also very effective for limited illumination changes. Even with this important quality, one can notice that using Codebook alone does not perform the best at all time, according to our results, which does not invalidate our observations. Color invariant also do not seem to degrade the quality of segmentations, regarding HSL and RGB-Rank results. For any test, F-Measure results with an (interesting) color invariant are whether close to none (with a difference of ±5%), whether more than 15% better.

Figure 2 shows some examples of segmentations, all obtained with a learning made on base 1 (L1). First line shows results for an image of processing base 1 (P1), second line shows an image from processing base 2 (P2) and last line an image from base 3 (P3). From left to right, the original image, the ground truths images, a segmentation performed without any colorimetric invariant applied, a segmentation with HSL conversion and lastly a segmentation using RGB-Rank.

First line of Figure 2 of course shows the most accurate segmentations. This is normal, since the learning was made on images of the same base as those processed. One can notice that the barriers were not well segmented. Second and third lines are more interesting. It is clearly visible that a segmentation obtained without any colorimetric invariant shows a very high false detection rate. Lots of parts of the background are incorrectly detected as foreground. On the contrary, HSL and RGB-Rank allow the algorithm to reduce dramatically the false positive rate. Of course, these segmentations cannot be perfect, but segmentations like these can now be



used for other processes such as tracking, without having to assume as much errors in it. A mathematical morphology post-processing could also be applied to resulting segmentation to remove noise and close holed foreground regions.

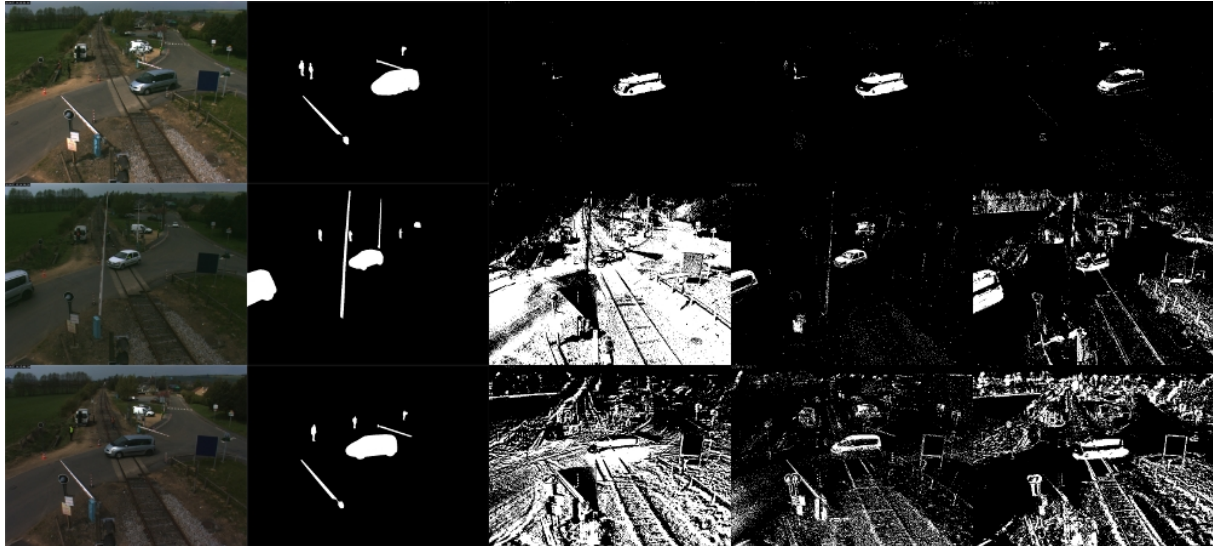


Fig 2: Segmentations obtained with a learning base L1. From left to right : original image, ground truths images, without colorimetric invariant, with HSL, with RGB-Rank

3.4. Tracking results

This section aims at showing the influence of the colorimetric invariants on the tracking of moving objects. It is important to note that the main objective of our contribution is not to determine which tracking method suits better to a use, but to point out the improvements of a better segmentation for any tracking method. To do so, mathematical morphology operators are applied in order to remove noise and recover full foreground regions in resulting segmentations, then we applied tracking. Such methods need parameters, that depend directly on the image aspect. Because of the inconstancy of the resulting noises from one test to another, it was difficult in certain cases to obtain the best results. Yet, we believe it is possible to obtain segmentations that can give better results. Nevertheless, one can determine the paths of interesting objects.

Figure 3 shows the results of the objects tracking for segmentations obtained on base P1 with a learning on base 1 (L1). The first column shows results on images without colorimetric invariant, second column images are converted to HSL, and third column images use the RGB-Rank colorimetric invariant. For each image, one color corresponds to one tracker that followed an object. Figure 4 shows the same results, but for segmentations obtained on base P2, also with a learning on base 1 (L1).

First, to make these images more understandable, we must describe the paths of objects for each sequence. The first and third sequences show closed barriers. Then, a car arrives from the road, top right of the image, turns on the level-crossing and stops on it. On the left of the images, two people wander. The second sequence shows three cars following each other from the horizon, top right. The first car passes the level-crossing, followed by the two others. The last car stops on the rail-crossing as fences go down.

Although, it is very difficult to read these images with the critically high number of active trackers in very noisy segmentations. One can determine certain paths and by comparison, notice that certain cases simply do not work. Let us focus our attention on the points that should theoretically track the car.

On Figure 3, first image (without colorimetric invariant) shows the path of the car (in white) arriving from the top right road and stopping in the middle of the scene. The second image (in HSL color space) shows two paths following the car. This is caused by the fact that the segmentation of the car is decomposed into two regions. The third image (with RGB-Rank) shows a trajectory, but with a color that changes, and which means the tracking was lost and recovered after a while. One can point out of this image that in this specific case (learning and



process bases are the same), the tracking quality is not very affected by the use of a color invariant or not, which means we do not degrade or lose information with this technique.

On Figure 4, first image (without colorimetric invariant) shows only a few regions in the left could be tracked, but no car could be since we can't determine any path in the center of the scene. On the second and third images, though, one can observe many points in curves on the theoretic path of cars, so these ones could be tracked.

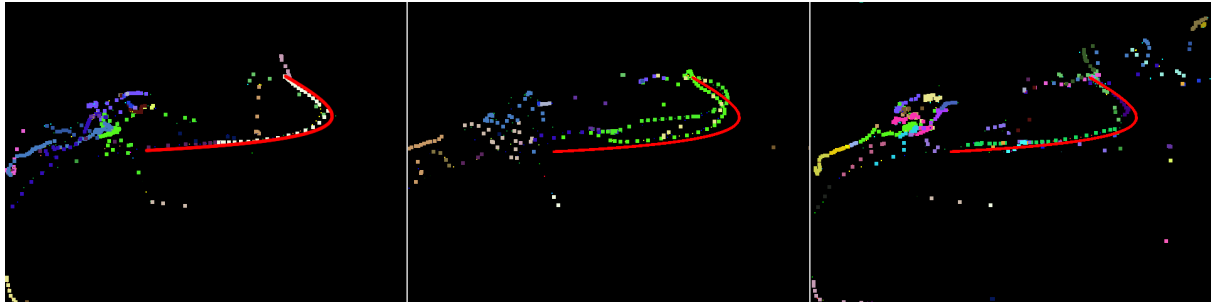


Fig 3: Results of objects tracking on segmentations from base P1 with learning on base L1, without invariant (column 1), HSL (column 2) and RGB-Rank (column 3). Red lines correspond to reference path.

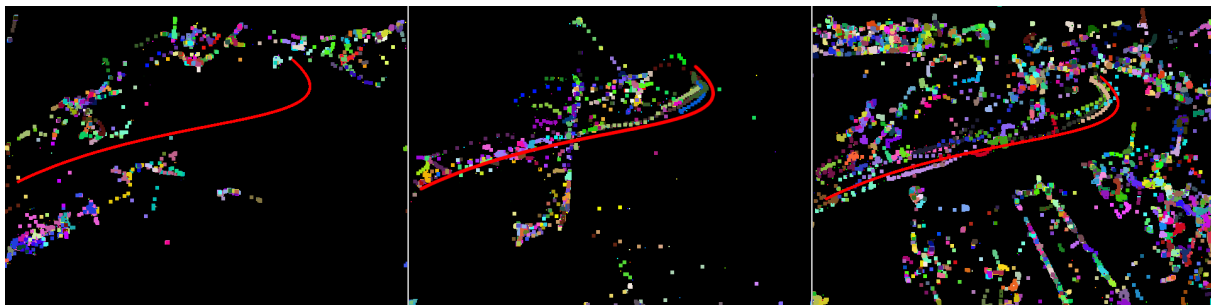


Fig 4: Results of objects tracking on segmentations from base P2 with learning on base L1, without invariant (column 1), HSL (column 2) and RGB-Rank (column 3). Red lines correspond to reference path.

Two hypothesis come out from these observations. First, the quality of the segmentation obtained is critical for the quality of the tracking. A noisy segmentation can cause poor tracking as can be seen in Figure 4, 3rd image. Second, we point the decisive effect of the use of a colorimetric invariant for the final goal of getting a tracking. As seen on Figure 4, 1st image, no tracking was possible if no colorimetric invariant was used. On the contrary, the use of either HSL or RGB-Rank enabled the tracker to draw the trajectory of moving objects, even if the number of false detections remained high.

4. Conclusions & Future work

In this paper, we showed the usefulness of the joint use of a colorimetric invariant with a codebook-based background subtraction system which is adapted for fixed-camera applications. This strategy allows in one hand rarefy the expensive background model update operation of the background subtraction algorithm, and in second hand to provide better results with a colorimetric invariant than without. Two color invariants behaved particularly better than other for this application : HSL and RGB-Rank. Multiple conclusions can be done from this observation: first, RGB-Rank is a good choice as a color invariant for this kind of process as it performed well in almost all cases; second, the difference between previous and current results for HSL color space confirm that results of this methods are highly dependent of the type of camera and the time of observation.

Plus, this method was successfully used with a simple multi-point tracking in order to follow targets during time. Even if this tracker was not the most elaborated one, a tracking could be done in most cases on resulting images. This is a very interesting result for level-crossing surveillance and dangerous situations detection, since a tracking of mobile objects can be performed successfully with better chances of success, especially when the background model is not yet adapted to an important change in the scene. We showed that even a very simple tracking method is able to provide at least the shape of objects trajectories when segmentations are obtained



through our method, where it is unable to do so when they are not. Applied to existing dangerousness-estimation methods involving even better tracking solutions, this technique may provide very good results for a lower cost. These observations are motivating to improve the codebook with different ways to deal with illumination and color changes according to any situation. Images in this paper were just different enough to confirm the efficiency of this method, but other images with specific, very dark or rainy situations should be interesting to test as well.

Future works on the problematic of detection can be done, especially on a new integrated color model included in the Codebook algorithm. Also, tests should be done on images showing darker and/or different meteorologic conditions to observe the behavior of this method on harder conditions of observation.

References

- S. Bianco, G. Ciocca, C. Cusano & R. Schettini (2010). Automatic color constancy algorithm selection and combination. *Pattern Recognition*, 43(3):695 – 705, 2010.
- Thierry Bouwmans & Fida El Baf (2010). Statistical background modeling for foreground detection: A survey. *Handbook of Pattern Recognition and Computer*, 2010.
- G. Buchsbaum (1980). A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310(1):1 – 26, 1980.
- Walter H. Buchsbaum (1975). *Color TV Servicing*, third edition. Englewood Cliffs, NJ: Prentice Hall, ISBN 0-13-152397-X, 1975.
- T. Carron (1995). Segmentation d'images couleur dans la base Teinte Luminance Saturation : approche numerique et symbolique. *PhD thesis, Universite de Stanford*, 1995.
- Graham D. Finlayson, Bernt Schiele & James L. Crowley (1998). Comprehensive colour image normalization. 1998.
- Graham D. Finlayson, Steven D. Hordley, Gerald Schaefer & Gui Yun Tian (2005). Illuminant and device invariant colour using histogram equalisation. *Pattern Recognition*, 38(2):179–190, 2005.
- Theo Gevers and Arnold W.M. Smeulders (1999). Color-based object recognition. *Pattern Recognition*, 32(3):453 – 464, 1999.
- A. Gijzenij & T. Gevers (2010). Color constancy using natural image statistics and scene semantics. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on, 33(4):687 –698, april 2010.
- K. Kim, T. H. Chalidabhongse, D. Hanuood & L. Davis (2004). Background modeling and subtraction by codebook construction. 2004.
- Stephan Obdrzalek, Jiri Matas & Ondrej Chum (2003). On the interaction between object recognition and colour constancy. *Proc. International Workshop on Color and Photometric Methods in Computer Vision*, 2003.
- Houssam Salmane, Yassine Ruichek & Louahdi Khoudour (2012a). 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.
- Houssam Salmane, Yassine Ruichek & Louahdi Khoudour (2012b). 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.
- C. Stauffer & W.E.L. Grimson (1999). Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:2246, 1999.