Background foreground segmentation with RGB-D Kinect data: An efficient combination of classifiers

Massimo Camplani , Luis Salgado

A B S T R A C T

Low cost RGB-D cameras such as the Microsoft's Kinect or the Asus's Xtion Pro are completely changing the computer vision world, as they are being successfully used in several applications and research areas. Depth data are particularly attractive and suitable for applications based on moving objects detection through foreground/background segmentation approaches; the RGB-D applications proposed in literature employ, in general, state of the art foreground/background segmentation techniques based on the depth information without taking into account the color information. The novel approach that we propose is based on a combination of classifiers that allows improving background subtraction accuracy with respect to state of the art algorithms by jointly considering color and depth data. In particular, the combination of classifiers is based on a weighted average that allows to adaptively modifying the support of each classifier in the ensemble by considering foreground detections in the previous frames and the depth and color edges. In this way, it is possible to reduce false detections due to critical issues that can not be tackled by the individual classifiers such as: shadows and illumination changes, color and depth camouflage, moved background objects and noisy depth measurements. Moreover, we propose, for the best of the author's knowledge, the first publicly available RGB-D benchmark dataset with hand-labeled ground truth of several challenging scenarios to test background/foreground segmentation algorithms.

1. Introduction

The availability in the market of RGB-D cameras with high performance and a low-price has completely changed the world of computer vision research and applications. Devices such as the Microsoft's Kinect or the Asus's Xtion Pro, that cost less than 200 dollars, can provide real time (≈30 frames per second) depth and color information with good resolution (VGA format); this attractive trade-off between cost and performance is not achievable by any other kind of depth cameras in the market, such as stereo or time of flight cameras. Due to these features, RGB-D cameras have been rapidly employed in several command-less applications such as gaming [1] and human computer interface platforms [2]. Moreover, they have been employed in other computer vision research areas such as human body tracking for elderly activity monitoring [3] and 3D object recognition [4]. Many of these applications, such as gaming or human computer interaction systems, rely on the efficiency of the underlying foreground/background segmentation algorithms where the moving objects are separated from the static environment to be further processed and analyzed.

Dense depth data provided by RGB-D cameras is very attractive for foreground/background segmentation in indoor environments (due to the range camera limitations) since it does not suffer from the challenging issues that affect color based algorithms: light switches or local gradual changes of illumination, shadows cast by the foreground objects such as in the scene presented in Fig. 9, and color camouflage due to similar color of foreground and background regions as in the case of the white moving object shown in Fig. 11. Moreover, depth information is very useful to detect and reduce the effect of moved background object (see figures in Section 4.6). For a complete review about background/foreground algorithms see [5–7].

However, the sole use of depth data presents several problems that bound the efficiency of such systems: depth-based segmentation fails in case of depth-camouflage that appears when foreground objects (or part of them) move towards the modeled background; an example of depth camouflage is reported in Fig. 13. Objects silhouettes are heavily affected by the high level of depth-data noise at object boundaries as shown in [8,9]. Moreover, depth measurements are not always available for all the image's pixels, see [8], due to multiple reflections, scattering in particular surfaces or occlusions. Finally, as shown in [10] the measurements' noise depends on the real-measured distance (with a

quadratic law); this aspect has to be included in the background modeling strategy in order to reduce the detection error due to different levels of noise.

For these reasons, efficient strategies to combine the complementary color and depth features are required to obtain more accurate and reliable foreground/background segmentations, thus reducing the impact of the previously mentioned errors. Although foreground/background segmentation is a mature and advanced topic in computer vision, there is still a lack of testing and investigation for those systems that provide registered color and depth data; for example, in recent review articles such as [5,6] very few works that use depth data – obtained with stereo systems – are mentioned. Moreover, the introduction of the dense depth information poses new challenges in the design and selection of foreground/background segmentation algorithms such as: the efficient integration of the depth and color features, their joint incorporation into well known foreground/background segmentation frameworks, to handle depth camouflage and depth-measurements noise, and the generation of new public databases containing RGB-D sequences for objective testing and comparison of proposed algorithms.

In this paper, we present an efficient foreground/background segmentation strategy based on the combination of two per-pixel statistical classifiers: one classifier based on the depth feature and the other one based on the color features. Their combination is based on a weighted average that allows to adaptively modify the *importance* of each classifier in the ensemble by considering past foreground detections and the detected edges in color and depth images. In this way, it is possible to reduce false detections due to critical issues that can not be tackled by the individual classifiers such as: shadows and illumination changes, color camouflage, moved background object, depth camouflage and noisy depth measurements. Another important contribution of this work is the generation of a new RGB-D dataset (acquired with Microsoft Kinect) with a hand-labeled ground truth for testing the performance of foreground/background segmentation algorithms. The proposed dataset is composed by five sequences of indoor environments, obtained with a static device that registers different demanding situations such as cast shadows, color and depth camouflage and moved background object. Moreover the dataset contains a stereo sequence presented in [11] for which we provide the ground truth. The benchmark database has been used to test the proposed strategy and state of the art algorithms. Results demonstrate that the proposed strategy guarantees more accurate and compact foreground/background segmentations.

The rest of the paper is structured as follows: in Section 2 a review of the most popular pixel-wise background subtraction approaches is given with particular attention to those based on depth; in Section 3 the proposed combination of classifiers strategy is presented; experimental results are introduced in Section 4 and in Section 5 the conclusions are drawn.

2. Pixel-wise foreground/background segmentation

Foreground/background segmentation is a computer vision task that aims at identifying moving objects (foreground *Fg*) in the scene and separate them from the static environment (background *Bg*). Generally applied to video data acquired by a static camera, in both indoor and outdoor scenarios, *Fg*/*Bg* segmentation is a low level fundamental task in many computer vision applications such as: video and traffic surveillance, sport games monitoring, enhanced video conference systems etc. Also known as *background modeling* or *background subtraction*, *Fg*/*Bg* segmentation is generally based on a background model, built by processing a bootstrap sequence, that is iteratively updated; the regions that deviate significantly from the model are identified as part of the foreground *Fg*.

Although many different algorithms have been presented in literature, *Fg*/*Bg* segmentation is still an open problem. Major challenges of this task are (see [5] for more details): stopping foreground objects, multimodal background, and bootstrapping. In addition, also the following issues have to be considered: *color camouflage* due to similar color distributions of the moving foreground and the background (this generally leads to incomplete detected moving object areas); gradual *illumination changes* causing modification of the *Bg* regions that can be erroneously detected as *Fg*; a *light switch*, that is a global or a local sudden change of luminosity; *shadows* projected by the moving objects in the scene. Finally, the problem of *moved background objects* occurs when a background object is moved and the new *empty* space is wrongly detected as foreground.

Among the several background/foreground segmentation algorithms proposed in the literature, the pixel-wise oriented approaches have become the most popular ones. In these approaches each pixel is modeled independently and local spatial relationships are added to the model in post processing stages. The most attractive features of these algorithms are their low computational requirements and easy portability to parallel architectures, and the possibility to locally adapt to each pixel the algorithm response. For a complete review of background modeling and *Fg*/*Bg* segmentation algorithms see [5–7].

Pixel-wise strategies are generally divided into parametric and non-parametric approaches. In the first case, for each pixel a predefined background model is assumed and its parameters are estimated. Non parametric techniques do not assume any fixed model and aims at estimating the background model distribution from the acquired pixels.

One of the first parametric background models, proposed in [12], is based on a single Gaussian distribution where the computed mean value is used as background model. Despite their low complexity, unimodal approaches only work well in controlled environments where the absence of multimodal backgrounds is guaranteed.

Multi-modal background algorithms are proposed to cope with more complex scenes containing quasi static backgrounds (e.g. waving trees). One of the most popular strategies is the Mixture of Gaussian (MoG) approach presented in [13], where each pixel is modeled as a Mixture of Gaussian distribution. The parameters of the distributions (mean μ, standard deviation σ and weight w) are learned and iteratively updated through an online version of the Expectation Maximization algorithm. A variation of the MoG algorithm has been presented in [14], where the recursive equations used to update the parameters of the model are modified to simultaneously select the appropriate number of components employed for each pixel.

A non-parametric approach is presented in [15], where the probability density function of each pixel is estimated online with a kernel density estimator (KDE). This method allows handling complex pixel density distributions that cannot be managed with parametric approaches. However, the main drawback of this type of algorithms is that they require a huge amount of memory to store a sufficient number of samples to accurately estimate the distributions, and are computationally demanding.

Vibe [16], is another interesting non parametric approach in which the pixel model is based on a set of background samples (including past and neighbor pixels), instead of being based on an explicit pixel background distribution model. Each incoming pixel is compared with the closest sample in the background set and those that *match* the background are then included in the set. Background pixels are randomly substituted by new ones independently of the insertion time.

Neural Network models are also proposed to detect foreground objects. In [17] for each pixel a Self Organizing Map (SOM) network is used for the segmentation: the network weights are initialized with the first frame of the sequence, and each incoming pixel is processed by the neurons of the network; if no match is found (large distance from the neurons) the pixel is labeled as a foreground pixel, otherwise the SOM is updated with a *winner takes all* strategy. Neural network approaches do not make any assumption about pixel distributions.

2.1. Foreground/background segmentation with depth data

One of the first proposals based on both color and depth data is presented in [18]; it is an adaptation of the MoG algorithm to color and depth data obtained with a stereo device. Each background pixel is modeled as a mixture of four dimensional Gaussian distributions: three components are the color data (*YUV* space in this case) and the fourth one is the depth data, D. Color and depth features are considered independent and the same updating strategy of the original MoG algorithm is used to update the distribution parameters. The authors propose a strategy where for reliable depth data, depth-based decisions bias the color-based ones: in case that a reliable distribution match is found in the depth component, the color-based matching criterion is relaxed thus reducing the color camouflage errors. On the contrary, in case that the stereo matching algorithm is not reliable, the color-based matching criterion is set to be harder to avoid problems such as shadows or local illumination changes.

MoG is also proposed in [19], where depth and infrared data are combined to detect foreground objects. Two independent background models are built and each pixel is classified as background or foreground only if the two models matching conditions agree. Performance of this approach is limited since a failure of one of the models affects the final pixel classification.

In [20], a multi-camera system combines color data and depth data, obtained with a low resolution ToF camera, for video segmentation. The Vibe algorithm [16] is applied independently to the color and the depth data: the obtained foreground masks are then combined with logical operations and then post processed with morphological operations.

The depth data provided by a ToF camera is used to generate 3DTV contents in [21]. The MoG algorithm is applied on the depth data to obtain foreground regions that are then excluded to the median filtering stage used to improve background depth map accuracy. The foreground objects (usually humans) can be eventually included in virtual scenarios.

As it can be seen, for the best of authors' knowledge, so far very few works have been devoted to the analysis and development of RGB-D foreground/background modeling techniques. In fact, as they have been mostly developed for stereo-based or ToF technologies, they do not consider the noise characteristics of the depth data provided by low-cost RGB-D cameras. Moreover, the few examples based on Kinect devices, such as [3], only use depth information to extract the foreground silhouettes, thus providing very limited results affected by the depth-data noise particularities previously described in the introduction. Therefore, new solutions are required to efficiently integrate color and depth data to improve the performance of the foreground/background segmentation.

3. RGB-D data segmentation with combination of classifiers

Color and depth features present different problems that can affect the performance of foreground/background segmentation algorithms; in particular color features are not robust to modification of illumination conditions and shadows cast by the moving objects. Furthermore, similar colors between background and foreground lead to the well known problem of color camouflage. Moreover, depth data (provided by RGB-D cameras) presents other challenging issues: pixels for which no depth measurements are provided (pixels *nmd*), noisy and irregular object boundaries and distance dependent measurement noise.

The simple binary combination of the resulting *foreground masks* obtained by using two independent color based and depth based segmentation algorithms, that is often proposed in literature, produces poor results since individual segmentation errors affect the final segmentation result.

In this paper, we propose a novel *Fg/Bg* segmentation strategy that is based on a combination of two per-pixel statistical classifiers; the scheme of the proposed combination of classifier is shown in Fig. 1. Color classifier CL_C is based on the color features C, the depth classifier CL_D is based on the depth feature D. The combination of the two classifiers' output, respectively d_C and d_D, is obtained through a weighted average combiner. For each pixel the support of each classifier to the final classification ($L(t)$) is obtained by considering the global edge-closeness probability P_g^e and the classification labels obtained in the previous frame $L(t-1)$. CL_C will be more *important* in the final ensemble decision near object borders, thus reducing the problem of noisy depth measurements at object boundaries. On the contrary, the CL_D will have a greater influence on the final ensemble decision for those pixels located in low gradient areas of the depth map; thus guaranteeing compact detection and reducing the influence of shadows and illumination changes. The *Fg* regions detected in the previous frames are used to check the reliability of the depth data: the lower is the distance between *Fg* and *Bg* depth values, the higher is the influence of the color classifier in the ensemble decision. In this way, it is reduce the effect of depth camouflage since detected foreground objects that move towards the background can be still segmented thanks to the support of CL_C. $L(t-1)$ information is also used to detect moved background objects as it will be explained in Section 3.4. It has to be highlighted that the depth based classifier has to be designed in order to reduce the distance-dependent noise, as shown in detail in Section 3.3.1.

Furthermore, thanks to the analysis of the color and depth edges and that of the evolution of previous detections, the proposed combination of classifiers allows to include in the segmentation process also the spatial pixel relationship that is not considered by the individual classifiers.

3.1. Combining color and depth based classifiers

Classifiers combination is a successful pattern recognition approach to solve particularly complex problems: instead of training and building a monolithic complex classifier that processes different features, the problem is decomposed into more simple and *local* problems that are tackled by *weak* classifiers. This research area

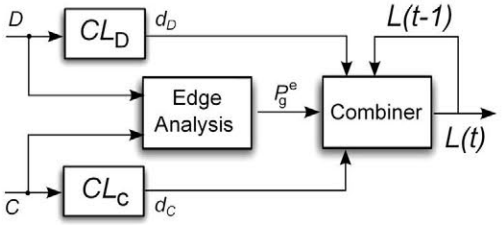

Fig. 1. Scheme of the proposed combination of classifiers. CL_C classifier based on color features C, CL_D classifier based on the depth data D. Output of the classifiers d_C and d_D, global edge-closeness probability P_g^e. Classification Labels obtained by the Combiner at time $t-1$ and t : $L(t-1)$ and $L(t)$.

has been extensively treated in the literature, for a complete review see [22] and [23, Ch. 9].

Before describing the proposed method, we want to remark some aspects of the problem at hand that justify and support our choices and formulation. The problem of background/foreground segmentation can be viewed as a binary classification problem in which a foreground label L_{fg} or a background label L_{bg} has to be assigned to each pixel. In particular, the feature set is composed by the color and depth data that can be considered uncorrelated. As previously mentioned, these features are, individually, well suited for this task, both showing different – but to some extent complementary – advantages and drawbacks. Hence, the main challenge is to find an efficient solution to combine these features in order to build an accurate system for foreground and background segmentation.

The proposed method is based on the use of two per-pixel weak statistical classifiers CL_D and CL_C built independently on the two different feature sets: depth data D and color data C. Each classifier CL_i produces a vector $d_i = [d_{i,bg}, d_{i,fg}]$ in which the value d_{ij} is the support for the hypothesis that the measured data $\mathbf{x}$ (C and D value of a pixel) belongs to the background class ω_{bg} and to the foreground class ω_{fg}. In fact, we are considering classifiers that produce a continuous output such as MoG or KDE introduced in Section 2 (a detailed description of the selected classifiers is given in Section 3.3). As far as the color classifier is concerned, the symbol C can be interpreted without loss of generality as a set of three features containing the color information such as the *RGB* space or any other color space as for example *YCbCr*, *HSI*, etc., or as a single feature containing the luminance data Y; as will be presented in Section 4 the proposed CL_C is based on the features space *YCbCr*.

Let us consider the pixel s at position (x, y), and the corresponding measured data $\mathbf{x}_s = [D, C], \Omega$ the set of c classes $\{\omega_1, \ldots, \omega_c\}$ and CL the set of l classifiers $\{CL_1, \ldots, CL_l\}$. Each classifier CL_i gives as output a support d_{ij} for all the c classes. The support d_{ij} can be considered as an estimation (obtained with the ith classifier) of the posterior probability that $\mathbf{x}_s$ belongs to the jth class. The decision profile DP [22] for $\mathbf{x}_s$ is:

$$DP(\mathbf{x}_s) = \begin{bmatrix} d_{1,1} & \cdots & d_{1,j} & \cdots & d_{1,c} \\ \vdots & \ldots & \vdots & \ldots & \vdots \\ d_{i,1} & \cdots & d_{i,j} & \cdots & d_{i,c} \\ \vdots & \ldots & \vdots & \ldots & \vdots \\ d_{l,1} & \cdots & d_{l,j} & \cdots & d_{l,c} \end{bmatrix} \quad (1)$$

where the ith row represents the output of the classifier CL_i, and the jth column represents the overall support $M_j(\mathbf{x}_s)$ of the classifiers to the class ω_j. The decision profile $DP(\mathbf{x}_s)$ can be used to find the overall support $\mathbf{M}(\mathbf{x}_s)$ for all the classes; the label is assigned to the class with the greatest $M_j(\mathbf{x}_s)$.

Two different combination techniques of the data contained in $DP(\mathbf{x}_s)$ can be considered: *class conscious* approaches, in which each column is processed independently without combining the $M_j(\mathbf{x}_s)$ of different classes together; on the contrary, in *class indifferent* approaches the obtained $M_j(\mathbf{x}_s)$ are used to generate new features processed by another classifier that completes the final classification task. *Class conscious* combiners are non trainable combiners since the overall supports $M_j(\mathbf{x}_s)$ are obtained with arithmetic operations; hence, the computational and memory cost of the classifiers' combination is kept low. On the contrary, *class indifferent* combiners are more complex since they add new parameters in the classification system that have to be tuned and initialized after an additional training phase. For these reasons, in our approach we select a *class conscious* combiner to reduce its complexity and computational requirements. In these combiners the overall support for the class ω_j is obtained with a combination function $f(\bullet)$:

$$M_j(\mathbf{x}_s) = f(d_{1,j}, \ldots, d_{l,j}) \quad (2)$$

The class label assigned to $\mathbf{x}_s$ corresponds to the class with the maximum value of $M_j(\mathbf{x}_s)$. Common choice for the function $f(\bullet)$ are: simple average, maximum, median, etc.; for a complete review see [22, Ch. 5]. These approaches have been successfully used in many pattern recognition problems and also in computer vision for background subtraction problems, such as in [24] where the simple average has been used to fuse 13 classifiers based on 13 different visual cues such as color, gradient and Haar features. However, the use of simple combination functions, such as average, assigns to all the features the same support to the final ensemble decision. This aspect is not suitable for our problem, since it does not allow to exploit the different information that derives from the feature sets D and C in different regions of the image or in particular conditions.

For these reasons, we select a more complex combination of the classifier supports d_{ij} that allows to adapt efficiently the *contribution* of each classifier to the final classification: we propose to use a weighted average combiner with l weights such that $M_j(\mathbf{x}_s)$ is estimated as:

$$M_j(\mathbf{x}_s) = \sum_{i=1}^{l} W_i(\mathbf{x}_s) d_{i,j} \quad (3)$$

As previously anticipated, the weight W_i is chosen as a function of the input $\mathbf{x}_s$ in order to increase the support of the most reliable classifier.

3.2. Classifiers' weights selection

The proposed weights selection strategy is a cascade of three steps that allows to properly calculating the classifiers weights according to the depth and color data. The scheme of the weights selection strategy for a pixel at position s is presented in Fig. 2. In the first step, it is checked if x_s is a *nmd* pixel. For all the *nmd* pixels, the weights of the classifiers are set such that $W_D(\mathbf{x}_s) = 0$ and $W_C(\mathbf{x}_s) = 1$. It is clear that in the case that either the depth measurement or the depth-based background model is not available, only CL_C is taken into account for the final pixel classification.

For all those pixels that do not belong to the *nmd* set, the second and the third steps are necessary to calculate the classifiers' weights. The aim of the second phase is to assign the weights for those pixels that belong to depth-image regions that contain edges, thus limiting the effect of noisy depth values at object boundaries. In particular, the depth (D) and color data (I) are analyzed and the global edge-closeness probability $P_g^e(x_s)$ is estimated (see Section 3.2.1 for more details). For those pixels that likely belong to an edge region, such that $P_g^e(x_s)$ is greater than a threshold W_{low}, the weights are assigned as $W_C(\mathbf{x}_s) = P_g^e(x_s)$ and $W_D(\mathbf{x}_s) = 1 - W_C(\mathbf{x}_s)$.

For the remaining pixels, i.e. these that do not belong to edge-regions ($P_g^e(x_s) < W_{low}$), weights are assigned to reduce the effect of depth camouflage (see for more details Section 3.2.2). The previous classification label for pixel s is considered: if $L(x_s, t-1) = \omega_{bg}$ the weights calculated in the second phase are used; otherwise if

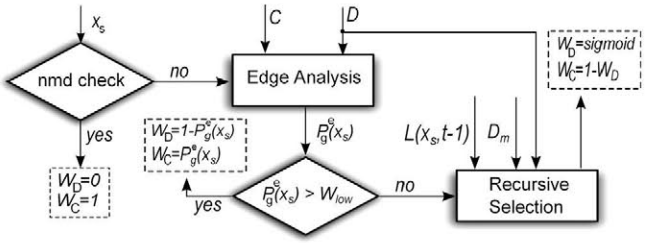

Fig. 2. Weights selection process.

$L(x_s, t-1) = \omega_{fg}$ the corresponding model D_m is taken into account and the weights are assigned by using the sigmoid function presented in Eq. 5.

3.2.1. Edge regions weights

For each pixel, $W_i(\mathbf{x}_s)$ are selected as a function of its edge-closeness probability in the depth and color images; the idea is to give a higher weight to the color based classifier CL_C for those pixels that belong to edge regions of the depth map and are close to edges in the color domain.

Let us consider the pixel s at the position (x, y) in the image; $P^e_D(\mathbf{x}_s)$ and $P^e_C(\mathbf{x}_s)$ are respectively the edge-closeness probability estimated for s in the depth and color image plane; the global edge-closeness probability is obtained as the product of the two probabilities $P^e_G(\mathbf{x}_s) = P^e_D(\mathbf{x}_s) * P^e_C(\mathbf{x}_s)$. The obtained global edge-closeness probability map P^e_G identifies the regions in which edges in the depth domain are *supported* by corresponding edges in the color domain; in these regions, color features are more reliable for the classification than the depth feature, hence, a higher weight has to be assigned to CL_C to increase the detection accuracy. At the same time, the product between the two edge-closeness probability functions allows to exclude those regions that do not contain a significant edge in the color domain. In this case the color of the region around the detected depth-border is homogeneous, CL_C could be affected by the color camouflage problem and, hence, it could wrongly classify all pixels as belonging to the background. Therefore in this case, the weight assigned to CL_C is lower than that of CL_D. In conclusion, for each processed frame, P^e_G is calculated as presented above and $W_i(\mathbf{x}_s)$ are assigned as follows:

$$\begin{cases} W_C(\mathbf{x}_s) = P^e_G(\mathbf{x}_s) \\ W_D(\mathbf{x}_s) = 1 - P^e_G(\mathbf{x}_s) \end{cases} \tag{4}$$

The values of the weights are limited to a minimum and a maximum value W_{min} and W_{max}; in this way it is guaranteed that all the classifiers' support are included in the final classification stage (see Eq. 3). The complete exclusion of a classifier from the final classification (i.e. in the case of $P^e_G(\mathbf{x}_s) = 1$) can lead to wrong labeled pixels. In our implementation, these values are $W_{min} = 0.1$ and $W_{max} = 0.9$. It is worth noting that, to improve the compactness of the detection, the weights of the pixel corresponding to the interior of the objects detected by CL_D are set to $W_{min} = 0.1$.

An example of the proposed weights selection strategy in the edge regions is shown in Fig. 3. The color at the lower parts of the moving object (see Fig. 3(a)) is very similar to the background, thus generating a very low P^e_C (Fig. 3(c)). On the contrary in the depth data (Fig. 3(b)) a clear discontinuity is present between the moving object and the background, thus obtaining the P^e_D values shown in Fig. 3(d). The final values of P^e_G are reported in Fig. 3(e): a low probability is assigned to the pixels in the lower part of the object since there are no color-edges there. In Fig. 3(h) the final foreground mask is reported: the object silhouette has been strongly refined with respect to the one generated by CL_D (Fig. 3(g)) in the regions where P^e_G is high (more *influence* for CL_C in the final classification). On the contrary, far from the edge regions, higher weights for CL_D guarantee a more compact foreground reducing the problem of color camouflage that affects CL_C in lower parts of the object (Fig. 3(f)). It is worth noting that the use of P^e_G is fundamental for an accurate detection, in fact, by using only P^e_D to assign the classifier's weights the edges of areas misclassified by CL_C (due to color camouflage) will be completely distorted. It has to be noticed that the range of the reported weights in Fig. 3 has been expanded into improve image quality; moreover *nmd* pixels are marked in red.

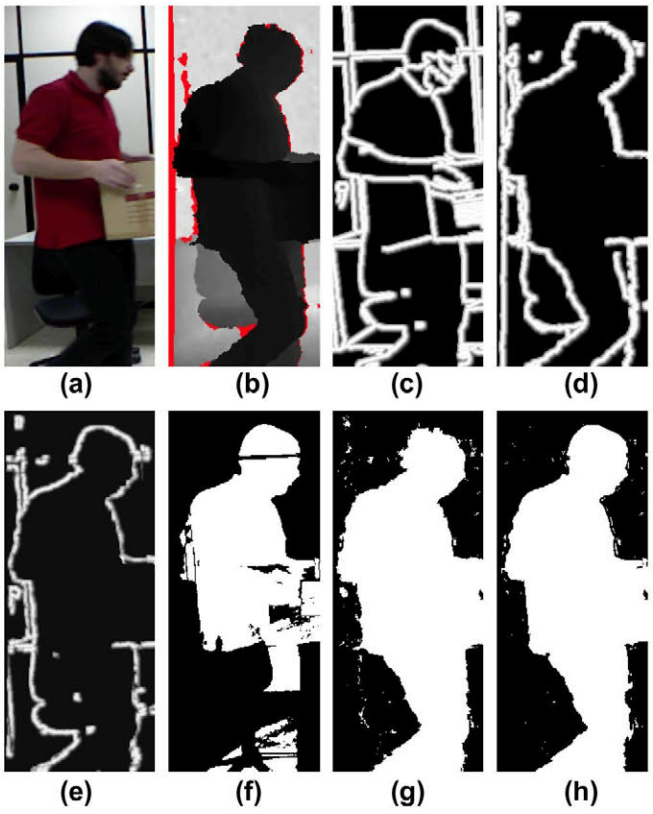

Fig. 3. Example of weights selection strategies in the edge regions: color data (a), depth data (b), (c) P^e_C, (d) P^e_D, (e) P^e_G, (f) CL_C foreground detection, (g) CL_D foreground detection, (h) proposed combined foreground detection.

3.2.2. Recursive weights selection

The color information is fundamental to avoid errors due to depth camouflage, these errors are due to foreground objects that move towards the background, thus resulting in foreground regions having the same depth of the background. The proposed approach tackles this problem by using previous foreground detections to estimate the distance (in the depth domain) between *Fg* pixels and the *Bg* model, and consequently increase the *influence* of CL_C when this distance is reduced.

For those pixels for which $P^e_G(\mathbf{x}_s) < W_{min}$ (pixels far from depth edges) the classifier weights are calculated as follows. Let us consider the pixel s located at position (x, y) of the frame acquired at time t; this pixel has been classified (by using the Eq. (3)) as foreground such that $L(\mathbf{x}_s, t) = L_{fg}$. The normalized depth distance $\delta(\mathbf{x}_s, t)$ (described in detail in the following paragraphs) between the foreground pixel and the corresponding pixel in the background model is computed and used to set the weights for the pixel classification in the following frame $(\mathbf{x}_s, t+1)$. In particular, the weight of CL_D is estimated as:

$$W_D(\mathbf{x}_s, t+1) = W_{min} + \frac{W_{max} - W_{min}}{\left(1 + Qe^{-B(\delta(\mathbf{x}_s,t)-M)}\right)^{1/v}} \tag{5}$$

The weight of CL_C is calculated as $W_C(\mathbf{x}_s, t+1) = 1 - W_D(\mathbf{x}_s, t+1)$. The function in Eq. (5) is the generalized logistic function [25], where W_{min} and W_{max} are the upper and lower asymptotes, B is the growth rate, M is the point of maximum growth, v determines which is the closest asymptote to the maximum growth point and Q is related to the curve value in the origin.

Regarding the normalized distance δ, it is defined as:

$$\delta(\mathbf{x}_s, t) = |(D(\mathbf{x}_s, t) - \mu_s(t)|/\sigma_s(t) \tag{6}$$

where $D(\mathbf{x}_s, t)$ is the depth value of the pixel, $\mu_s(t)$ is the depth value of the most representative sample of the corresponding background model, and $\sigma_s(t)$ its interval of confidence. In our implementation, as it will be explained in detail in Section 3.3.1, the background model is based on a mixture of Gaussian distributions and μ and σ represent respectively the mean and the standard deviation of the most probable Gaussian of the background model. It is worth noting that this definition of δ allows to adapt the weights in Eq. (5) to the time varying characteristics of the background distributions.

The logistic curve has been selected because it allows to smoothly select classifiers' weights and bound their values to the two extrema W_{min} and W_{max}: in this way, as in the case of the weight selection presented in the previous section, it is guaranteed that all the classifiers' support are included in the final classification stage. The parameters of the logistic curves need to be selected such that a low (high) weight is assigned to the CL_D (CL_C) for those foreground pixels that are very close to the background model and vice versa, thus limiting the effect of depth camouflage. The choice of the curve parameters and profile is strictly related to the used background model. In our implementation where a Gaussian model is used, it is reasonable to consider a normalized distance of 2.5 (used for matched component identification in MoG models [26]) as the distance for which CL_C is considered a very reliable classifier, reaching the maximum weight W_{max} for $\delta(\mathbf{x}) = 2$. On the other side, as $\delta(\mathbf{x})$ increases, W_C should decrease and consequently increase the weight W_D of the depth classifier. From $\delta(\mathbf{x})$ above 5.5, i.e. there is no possible depth camouflage, the W_C is set to be almost minimum and, therefore, W_D set to maximum as CL_D becomes the most reliable classifier. The curve meeting the previous considerations is shown in Fig. 4, obtained with $B = 1.8$, $M = 3.3$, and Q and v set to typical values 1 and 0.5.

It has to be highlighted that in the case that $L(\mathbf{x}_s, t) = L_{bg}$, the weights W_i are computed using Eq. (4).

An example of the proposed recursive weights selection strategy is shown in Fig. 5. As it can be noticed in Fig. 5(b) the moving hand depth values are very similar to the background ones generating very low P_G^e values (Fig. 5(c)) since no depth discontinuity is detected. Moreover the foreground mask obtained by CL_D is clearly incomplete due to the depth camouflage problem (see Fig. 5(g)). On the contrary the color information allows to easily segment the hand; the foreground mask obtained by CL_C is reported in Fig. 5(f). By calculating the classifiers' weights as in Eq. (5) it is possible to obtain the final weights masks for CL_D (Fig. 5(d)) and CL_C (Fig. 5(e)). As it can be noticed, higher weights are assigned in the hand region to CL_C, thus guaranteeing an improved detection in the final classification step (see Fig. 5(h)).

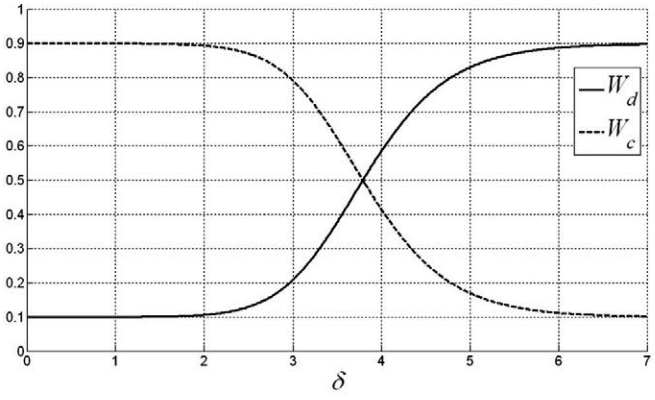

Fig. 4. Logistic function to calculate classifiers' weights as a function of the normalized distance δ: solid line W_D and dashed line W_C. The parameters used to obtain this specific logistic function are: $Q = 1$, $B = 1.8$, $M = 3.3$, $v = 0.5$.

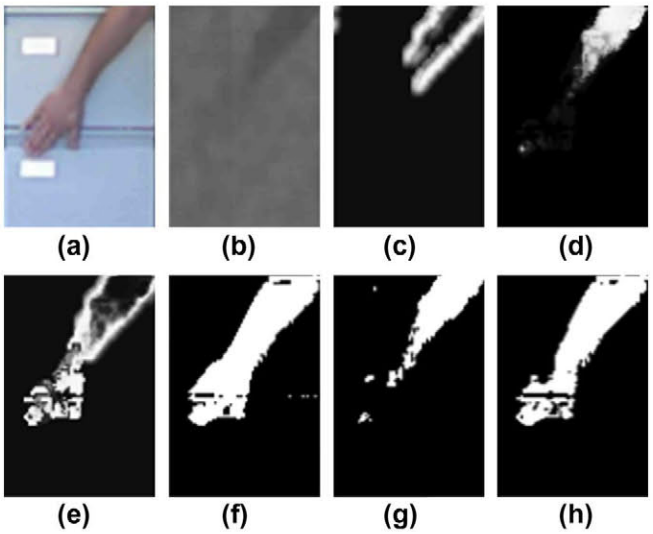

Fig. 5. Example of weights selection strategies in the edge regions: color data (a), depth data (b), (c) P_G^e, (d) W_D, (e) W_C, (f) CL_C foreground detection, (g) CL_D foreground detection, (h) proposed combined foreground detection.

3.3. Statistical classifiers

In the proposed approach, the independent pixel-wise Bayesian classifiers CL_C and CL_D estimate for each sample its a posteriori probability to belong to the foreground or background class. The estimated a posteriori probabilities are then inserted in the decision profile DP and subsequently processed by the classifier combiner as shown in Eq. (3).

Let us consider $\tilde{\mathbf{x}}_s$ as the value of the pixel s at position (x, y) in one of the two features spaces, C or D. Each Bayesian classifier classifies $\tilde{\mathbf{x}}_s$ as belonging to the class that maximizes the probability of the pixel to belong to the class ω_i when its value is $\tilde{\mathbf{x}}_s$. This probability is the so-called a posteriori probability $P(\omega_i|\tilde{\mathbf{x}}_s)$, that can be written as:

$$P(\omega_i|\tilde{\mathbf{x}}_s) = P(\omega_i)p(\tilde{\mathbf{x}}_s|\omega_i)/p(\tilde{\mathbf{x}}_s) \tag{7}$$

where $P(\omega_i)$ is the prior class probability, $p(\tilde{\mathbf{x}}_s|\omega_i)$ its likelihood, and $p(\tilde{\mathbf{x}}_s)$ is the prior probability of measure $\tilde{\mathbf{x}}_s$ (also called evidence factor). The latter one in a classification problem containing c classes can be written as:

$$p(\tilde{\mathbf{x}}_s) = \sum_{i=1}^{c} p(\tilde{\mathbf{x}}_s|\omega_i)P(\omega_i) \tag{8}$$

As can be noted in Eq. (7), the evidence factor is a scale factor that does not affect the classification. Moreover, in our implementation we consider the two classes having the same prior probabilities. Hence, the classifiers are based on the Maximum Likelihood (ML) criteria, where s is classified as belonging to the class ω_{ML} that maximizes $p(\tilde{\mathbf{x}}_s|\omega_i)$:

$$\omega_{ML} = argmax_i\{p(\tilde{\mathbf{x}}_s|\omega_i)\} \tag{9}$$

When a pixel has been classified as belonging to ω_{ML}, the model of this class is updated. In the following sections the parametric models used for ω_{bg} and ω_{fg} are presented.

3.3.1. Background modeling

The likelihood function of the background used in both classifiers CL_C and CL_D is based on a mixture of Gaussians model proposed in [13]. As mentioned in Section 2 MoG background model has been widely used in literature. The general features of the MoG

are very attractive for our application: it allows to accurately estimating quasi-static backgrounds, adapting to new background configurations and gradual changes, and each pixel is modeled independently as a mixture of Gaussian distributions. In particular, this choice is very appropriate for the depth-based background model, since it is possible to efficiently adapt for each pixel the MoG parameters to the distance-dependent noise that affects the depth data.

The likelihood function of the background model is:

$$P(\tilde{\mathbf{x}}_{s,t}|\omega_{bg}) = \sum_{i=1}^{K} v_{i,t} \cdot \eta\left(\tilde{\mathbf{x}}_{s,t}, \mu_{i,t}, \boldsymbol{\Sigma}_{i,t}\right) \tag{10}$$

where K is the number of Gaussians, $v_{i,t}$ is the weight associated to the ith Gaussian η at the time t with mean $\mu_{i,t}$ and covariance matrix $\boldsymbol{\Sigma}_{i,t}$. For the depth classifier, the Gaussians have a single dimension; for the color based classifier, Gaussians have three components that are assumed to be independent, thus reducing the computational cost of the algorithm as widely considered in the literature [6,26]. Therefore, the covariance matrix $\boldsymbol{\Sigma}$ is a diagonal matrix containing the variances of the three color components. The weight of a Gaussian measures the accuracy with which it models the value of the corresponding pixel.

In the case of a single dimension feature space (i.e. depth), when a pixel has been classified as background the parameters of the matching distribution are updated as proposed in [13]:

$$v_{i,t+1} = v_{i,t}(1-\alpha) + \alpha \tag{11}$$

$$\rho = \alpha \cdot \eta(\tilde{x}_{s,t}, \mu_{i,t}, \sigma_{i,t}) \tag{12}$$

$$\mu_{i,t+1} = \mu_{i,t}(1-\rho) + \rho\tilde{x}_{s,t+1} \tag{13}$$

$$\sigma^2_{i,t+1} = \sigma^2_{i,t}(1-\rho) + \rho(\tilde{x}_{s,t+1} - \mu_{s,t+1})^2 \tag{14}$$

where α is the so called *learning rate* that determines the speed of adaptation to changes in the scene and the speed of the incorporation of foreground objects to the background. The learning rate indicates the influence that the last data have on the Gaussian distribution parameters. For the unmatched Gaussians all the parameters remain unchanged except their weight:

$$v_{i,t+1} = v_{i,t}(1-\alpha) \tag{15}$$

The distributions are continuously ordered by considering the ratio between their weight and the standard deviation ($r_i = w_{i,t}/\sigma_{i,t}$): a high value for $r_{i,t}$ means that the ith distribution has modeled very well the pixel in the past ($w_{i,t}$ is high), that its value has a low variability (low $\sigma_{i,t}$) and is close to the mean ($\mu_{i,t}$) of the Gaussian. The pixel is assigned to the first of the K distributions for which the following equation is met:

$$(\tilde{\mathbf{x}}_{s,t+1} - \mu_{i,t}) < \lambda\sigma_{i,t} \tag{16}$$

A pixel is considered belonging to a Gaussian if it falls in an interval of λ times the standard deviation from the mean of the Gaussian. A typical value used in the literature for λ is 2.5.

We propose to modify the ranking parameter r for the depth-based background model. In fact, as mentioned in the introduction, the standard deviation of the depth measurements is related to the measured depth value: pixels corresponding to object points close to the camera always show smaller $\sigma_{i,t}$ values than those for points located further, thus introducing a possible bias in the ranking procedure. For this reason, to limit this bias, we propose normalizing $r_{i,t}$ with a parameter σ_{min} that is selected according to the quadratic relationship between distance $\mu_{i,t}$ and noise dispersion [10]. This parameter represents the minimum allowed standard deviation for a Gaussian distribution modeling the background in the depth domain.

It is worth noting that very stable measurements can lead to very small values of Gaussian standard deviation, thus limiting the adaptability of the model to gradual and small measurements variations. In general, to avoid this problem, the estimated standard deviation needs to be lower-bounded. The selection of the value of this lower bound is straightforward (and fixed for every pixel) if the MoG is used in a color space (as we do for the color-based model). The choice of this value is more critical in the case of the depth-based model due to the distance-dependent noise; hence it is set to the value of σ_{min} previously introduced. In this way also the MoG algorithm is dynamically adapted to the different levels of noise affecting the depth data.

3.3.2. Foreground modeling

Modeling the foreground process is a non trivial task when considering only single pixel information and when no specific assumptions on foreground object characteristics are made. In the proposed approach we use a uniform distribution to model the foreground process (as proposed in [27]): $p(\tilde{\mathbf{x}}_s|\omega_{fg}) = \frac{1}{(2^{nbit})^N}$. Where 2^{nbit} is the number of discrete values that the foreground pixel can take: in our case, 8-bit per pixel for each feature is considered; it is worth noting that the depth values are properly scaled in this image format. N is the dimension of each feature set: $N = 1$ for depth data and $N = 3$ for color data. This model has been selected since it outperforms, when used in a statistical framework, common background-exception based classification strategies, as demonstrated in [28].

3.4. Classifiers training and initialization

As mentioned in the introduction, one of the main problems of background/foreground segmentation algorithms is the initialization of the background model. In fact, an empty scene without moving objects for building a reliable background model is seldom available. The proposed classifiers are based on a parametric mixture of Gaussian model for the background likelihood term (see Eq. (10)) that has to be initialized with the very first frames. In particular for CL_C the likelihood parameters μ and $\boldsymbol{\Sigma}$ are initialized with the first frame by analyzing neighbor pixels and supposing that they share the same distribution as proposed in [29] and successfully used in [16]. Also in the case of CL_D the parameters of the one-dimensional likelihood function are initialized with the first frame: the value of μ corresponds to the measured depth value and the value of σ is selected according to the distance-noise quadratic law (see [10]), in particular for each measured depth we theoretically know its dispersion (σ_{noise}). A value of σ equal to $2.5\sigma_{noise}$ has been found suitable for the model initialization. Once the likelihood models are initialized they are continuously updated with new samples as shown in Section 3.3.1. It is worth noting that the foreground likelihood model has not to be trained. The proposed initialization is quite robust and reliable, but the presence (in the very first frames) of moving objects in the scene can still lead to a wrong selection of the likelihood parameters. For this reason, as proposed in [30] we use an adaptive learning rate α that is initially set to $1/N_{frame}$, where N_{frame} is the number of processed frames, until a minimum value of α is reached. Therefore, during the initialization the first frames have a great impact on the distributions' parameters, thus preventing from incorporating the moving objects to the background model.

In order to avoid the moved background objects problem, we propose an efficient per-pixel strategy based on the depth feature. In particular, for all those pixels detected as *Fg*, the likelihood models are re-initialized if the moved background object condition is reached: the foreground depth is greater than $2.5\sigma_{noise}$ times then the one of the background. In this way the pixels relative to the

new background, previously occluded by the removed object, are rapidly incorporated in the *BG*.

3.5. Estimating edge-closeness probability

The edge-closeness probabilities of the depth map and the color image are estimated in order to modify the combiner's weights as described in Section 3.2.1. As the same approach is used for both types of data, in the following paragraphs we use the word image without distinction between depth and color data. The main idea of the proposed strategy is to identify the edges present in the image and assign to each pixel of the image an edge-closeness probability value $P^e(\tilde{\mathbf{x}}_s)$ that depends on the distance between the pixel and the closest edge.

Let us consider an image I and any edge detector function $F(\bullet)$ such that the binary mask $B = F(I)$ can be estimated: B highlights the image pixels that belong to detected edges. For a pixel s at position (x, y), $P^e(\tilde{\mathbf{x}}_s)$ is computed as follows:

$$P^e(\tilde{\mathbf{x}}_s) = \max\{B(E) * G\} \tag{17}$$

where E is a defined neighborhood of s, G is a Gaussian kernel of size E and standard deviation equal to σ_G. The $P^e(\tilde{\mathbf{x}}_s)$ value is then equal to one for those pixels that belong to the identified edge ($B(s) = 1$); for the other ones, the probability value is assigned as a function of the distance between the closest edge and the pixel. The value $P^e(\tilde{\mathbf{x}}_s)$ is set equal to the maximum value of G response over the binary mask B: the higher is the response of the filter the closer is the edge to the pixel s. It is worth noting that $P^e(\tilde{\mathbf{x}}_s)$ decays like a Gaussian with the increase of the edge-pixel distance. In our implementation we used the Canny edge detector [31] to obtain the binary mask B.

4. Benchmark data and results

The results have been obtained using as benchmark five different indoor sequences acquired in our laboratory with the Microsoft Kinect. For each sequence, freely available on the Internet,[1] we provide a hand-labeled ground truth. For the best of authors' knowledge, we propose the first RGB-D benchmark dataset with hand-labeled ground truth that includes sequences with different challenging scenarios for *Fg*/*Bg* segmentation. Moreover, we tested the proposed algorithm on the stereo data presented in [11] for which we provided a ground truth.

The sequence *GenSeq* is used to test the overall performance of the algorithm in case of complex scenarios taking into account all the possible error contributions in the scene. On the contrary, the other sequences are conceived to highlight the impact of one particular issue to the algorithm under test. The sequence *ShSeq* helps to analyze the impact of shadows on the *Fg*/*Bg* segmentation algorithms. The sequence *DCamSeq* is used to analyze the performance of the algorithm when depth camouflage occurs. The goal the sequence *ColCamSeq* is to test the performance of the algorithms when the color camouflage problem occurs. The sequence *MoveBGSeq* helps to analyze the impact of the moved background object problem. *StereoSeq* is a stereo sequence of an outdoor environment. It is worth noting that for *DCamSeq* and *ColCamSeq*, the ground-truth and the algorithms performance tests have been conducted considering only those regions in the images where each particular type of problem is present. This procedure guarantees that other sources of errors do not bias the algorithms performance comparison with respect to the considered error factor.

As a measure of the algorithm performance, we compare the following values: False Positive (FP) the fraction of the *Bg* pixels that are marked as *Fg*; False Negative (FN) the fraction of *Fg* pixels that are marked as *Bg*; Total Error (TE) the total number of misclassified pixels, normalized with respect to the image size. Moreover, we consider also the similarity measure S defined in [32]. It is a non-linear measure that fuses FP and FN and it is close to 1 if detected *Fg* regions correspond to the real ones, otherwise its value is close to 0. To analyze also the errors close to the moving objects boundaries, we propose using also the similarity measure S_B. It is calculated as S, but considering only the regions of the image surrounding the ground-truth object boundaries: a region of 10 pixels is considered.

Finally, we use the overall metric to rank the accuracy of the analyzed methods that has been proposed in [7]. We combine the performance across different metrics and sequences into a single rank that is indicative of how well a method performs with respect to the other studied methods by calculating an average ranking RC across all categories. Let us define $rank_i(m, sq)$ as the rank of the ith method for the performance metric m in the sequence sq, the average ranking of the method i in the sequence sq is calculated as:

$$RM_i = \frac{1}{N_m}\sum_m rank_i(m, sq) \tag{18}$$

where N_m is the number of performance metrics. The overall ranking across categories RC_i of ith method is computed as the mean of the single RM_i across all the sequences. In the following paragraphs we compare the performance of the following algorithms: the proposed adaptive weighted classifier CL_W; the two weak classifiers CL_C and CL_D; the MoG algorithm proposed in [18] MOG_{RGB-D}; and the binary combinations of the foreground masks obtained by two independent modules based on depth and color data as proposed in [19] (by using MoG) and in [20] (by using ViBe), we refer to these algorithm as MOG_{Bin} and $Vibe_{Bin}$. Finally we adapt to the RGBD feature space the neural networks algorithm proposed in [17] (*SOM*) and the modified MoG algorithm proposed in [14] (MoG_{Ziv}).It has to be noted that no post-processing stages, such as morphological filtering, are applied to the resulting foreground masks.

4.1. GenSeq results

Sequence *GenSeq* is an indoor sequence acquired at a frame rate of 30 fps; it is composed by 300 frames and the corresponding ground truth it is composed by 39 frames spanning 115 frames of the sequence where the moving object is present (one every three frames has been labeled). This sequence combines different challenging situation for foreground/background segmentation algorithms: shadows of the moving objects, color camouflage, noisy depth data and interaction between foreground objects and the background. As previously mentioned this sequence is used to test the overall performance of the algorithm in case of complex scenarios taking into account all the possible error contributions in the scene. In Fig. 6 the color data and the depth data of the scene acquired with the Kinect are reported. Many of parameter used (on this and the other sequences) for the proposed algorithms has been introduced and motivated along the paper. The number of Gaussians for the *Bg* likelihood function has been set to $K = 2$, such that combined with the uniform distribution for the *Fg* the proposed algorithm is similar to a MoG with $K = 3$ that is a typical parameter for indoor applications. The size of E is set to 11 and it has been selected by considering the ROC curve in Fig. 8. For the others approaches the parameters have been selected according to the optimal values reported in [7].

The results obtained with this sequence are reported in Table 1. As it can be noticed, the proposed approach CL_W guarantees the highest values of S and a low percentage of *FP* and *FN*; furthermore,

[1] http://www.gti.ssr.upm.es/mac/.

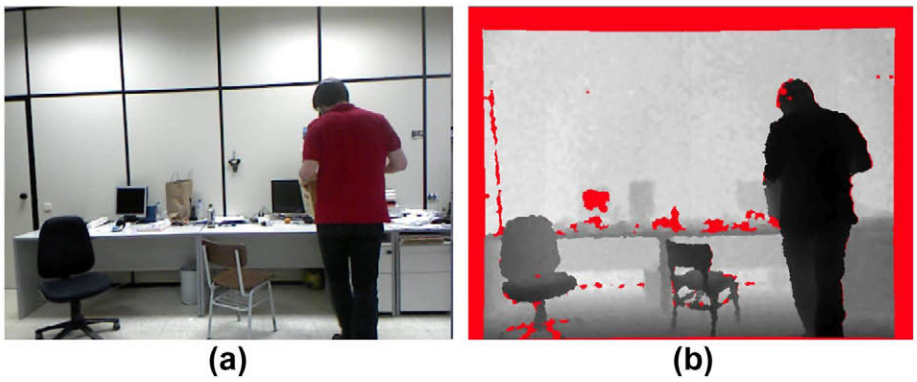

Fig. 6. *GenSeq* sequence: color data (a), depth data (b).

Table 1
Detection accuracy obtained by analyzing the *GenSeq*.

	TE		*FN*		*FP*		*S*		S_B		*RM*
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	
CL_C	2.38	1.20	16.38	0.30	0.63	0.02	0.72	0.23	0.55	0.16	4.60
CL_D	2.06	0.48	1.77	0.03	2.09	0.01	0.78	0.21	0.42	0.12	5.40
CL_W	1.30	0.42	1.49	0.02	1.27	0.01	0.83	0.21	0.53	0.14	2.60
MoG_{Bin}	2.03	1.20	17.01	0.26	0.16	10^{-3}	0.74	0.24	0.61	0.17	3.80
MoG_{RGB-D}	1.93	0.66	0.63	0.01	2.09	0.02	0.79	0.20	0.45	0.13	3.40
SOM	1.91	1.00	1.34	0.03	1.98	0.03	0.80	0.19	0.48	0.14	3.20
mog_{ZIV}	3.12	1.03	1.21	0.02	3.35	0.03	0.72	0.20	0.35	0.12	6.20
$ViBe_{Bin}$	12.39	1.15	0.65	0.03	13.85	0.03	0.44	0.16	0.12	0.05	6.80

the approach proposed in this paper allows to obtain the lowest value for *RM*. The classifiers CL_D and CL_C, used in the classifier combination, lead individually to inaccurate results: CL_C is affected dramatically by the color camouflage problems that lead to a high value of *FN*, on the contrary, the CL_D leads to a high value of *FP*, that is due mainly to the noisy boundaries. It is worth noting that the proposed approach allows to obtain a value of S_B very close to the one obtained with CL_C. MoG_{RGB-D}, *SOM* and MoG_{ZIV} guarantee a lower value of *FN* but a higher value of *FP* and this is caused mainly by shadows and irregular boundaries of the segmented foreground object. However, the value of *TE* is higher with respect to the proposed approach CL_W and also a lower value of *S* is obtained. It is worth noting that in this case the worst segmentation performances are obtained with $ViBe_{Bin}$ since the errors of the two independent models affect negatively the obtained final segmentation.

In Fig. 7 the foreground/background segmentation results for one frame of the *GenSeq* sequence are reported. As shown in Fig. 7(c), the color classifier CL_C is affected by shadows and color camouflage problems thus resulting in a fragmented foreground object mask. On the contrary, the CL_D (Fig. 7(c)) classifier allows to obtain a more compact silhouette and to reduce the effect of shadows; however, irregular borders and several false positive detections in the wall are present. The proposed approach CL_W (Fig. 7(e)) efficiently combines depth and color data thus reducing the effect of color camouflage and shadows; furthermore, false positive detections in the wall are reduced and the foreground object silhouette accuracy is dramatically increased. The approach *SOM* (Fig. 7(f)) improves the detection with respect to the color and depth classifiers but it is still affected by the shadows effect and the problem of noisy depth boundaries.

Fig. 8 shows the ROC curves considering *FP* and *FN* and obtained varying the fixed a priori probabilities of the two classes. In Fig. 8 are reported different curves obtained with the proposed algorithm by using different size of *E* for the P_g^e estimation. Satisfactory results are obtained with a *E* size of 11 pixels (solid curve with asterisk marker), in the plot we report also the point corresponding with the methods MoG_{RGBD}, *SOM* and MoG_{ZIV}, as reported in Table 1 to a lower *FN* rate corresponds a higher value for *FP*.

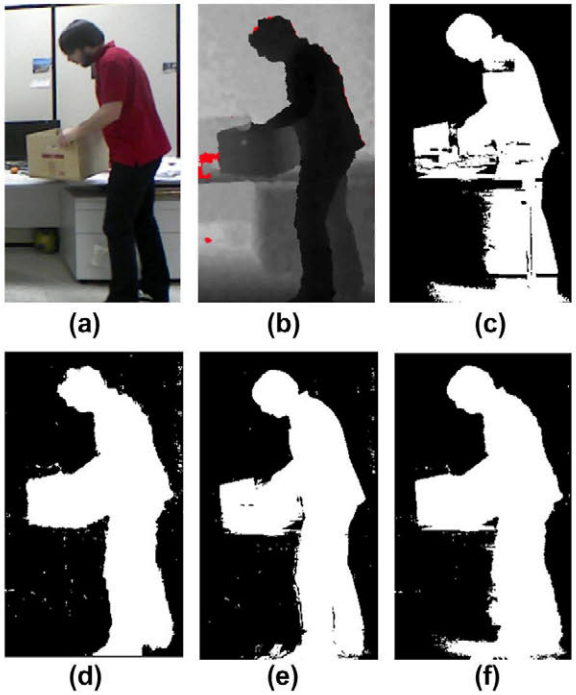

Fig. 7. Frame 1014 of the *GenSeq* sequence: color data (a), depth data (b), CL_C output (c), CL_D output (d), CL_W output (e), MoG_{RGB-D} output (f).

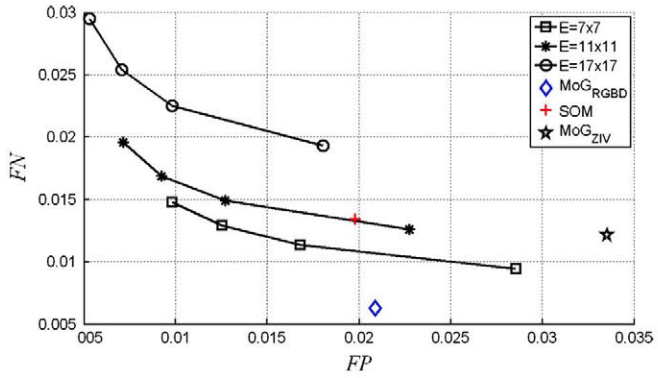

Fig. 8. ROC curves for the *GenSeq* sequence obtained by varying the a priori class probabilities for CL_W. Curves obtained with different size of E are presented: $E = 7 \times 7$ solid line and square marker, $E = 11 \times 11$ solid line and asterisk marker, $E = 17 \times 17$ solid line and circle marker. The values of *FN* and *FP* for the others algorithms are reported: MoG_{RGBD} diamond, *SOM* cross and MoG_{ZIV} star.

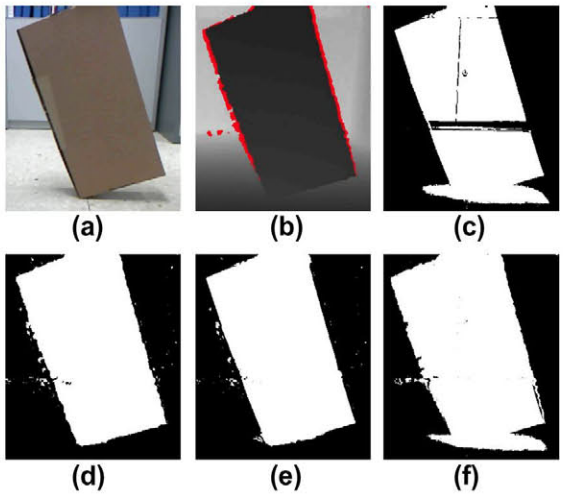

Fig. 10. Frame 446 of the *ShSeq* sequence: color data (a), depth data (b),CL_C output (c), CL_D output (d), CL_W output (e), MoG_{ZIV} output (f).

4.2. ShSeq results

Sequence *ShSeq* is an indoor sequence acquired at a frame rate of 30 fps; it is composed by 250 frames and the corresponding ground truth is composed by 25 frames spanning 120 frames of the sequence where the moving object is present (one every five frames has been labeled). The goal of this sequence is to highlight the impact that shadows projected moving object have on the foreground/background segmentation algorithm. In particular it is a close-distance sequence (maximum depth ≈3.5 m) that contains a moving box that projects strong and light shadows on the floor. In Fig. 9 the color data and the depth data of the scene acquired with the Kinect are reported.

Table 2 reports the results obtained by processing the *ShSeq*. Let us consider the column of *FP*: as it can be noticed, CL_C presents a high value of *FP* with respect to CL_D due to the presence of the shadows. All the others methods are affected by the presence of the shadows, due to the fact that color and depth features have the same influence to the final classification that is in fact biased by the misleading color data. The MoG_{bin} reduces drastically the percentage of *FP* since the shadows are correctly classified by the depth based module: however, the binary combination of the depth based and color based foreground masks leads to a high percentage of *FN* and a low value of *S*. The combination of classifiers proposed in this paper, CL_W, is the most reliable classifier in this sequence since it allows to obtain a very low value of *FP*, similar to the one obtained by CL_D and, at the same time, a low value of *FN*. Moreover, it guarantees the highest value of *S* and S_B with respect to the other classifiers and it obtains also the lowest value of *RM*. In Fig. 10 the results obtained with the two weak classifiers (Fig. 10(c)

Fig. 9. *ShSeq* sequence: color data (a), depth data (b).

Table 2
Detection accuracy obtained by analyzing the *ShSeq*.

	TE		*FN*		*FP*		*S*		S_B		*RM*
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	
CL_C	5.37	2.88	18.20	0.58	3.23	0.05	0.67	0.16	0.63	0.10	5.60
CL_D	0.98	0.33	0.95	0.04	0.98	0.02	0.93	0.03	0.67	0.06	2.80
CL_W	0.81	0.35	1.60	0.05	0.68	0.02	0.94	0.04	0.71	0.07	2.20
MoG_{Bin}	3.43	2.38	23.51	0.66	0.08	10^{-3}	0.75	0.17	0.58	0.11	4.80
MoG_{RGB-D}	3.94	1.54	0.59	0.02	4.50	0.07	0.77	0.09	0.66	0.05	4.20
SOM	5.75	1.80	0.05	10^{-3}	6.70	0.08	0.71	0.10	0.57	0.03	5.80
mog_{ZIV}	3.19	0.79	0.61	0.03	3.62	0.04	0.81	0.07	0.59	0.04	4.00
$ViBe_{Bin}$	7.15	1.49	0.01	10^{-3}	8.34	0.07	0.66	0.09	0.54	0.02	6.60

Fig. 11. *ColCamSeq* sequence: color data (a), depth data (b).

and (d)) and the proposed approach (Fig. 10(e)) are shown. As it can be noticed, the CL_W correctly classifies the shadows as background, similarly to the results obtained by CL_D, and, additionally, it refines and improves the accuracy of the detected object silhouette. On the contrary MoG_{ZIV}, see Fig. 10(f), (the second best algorithm for *ShSeq*) is not able to eliminate the false detection due to the presence of the shadows.

4.3. ColCamSeq results

Sequence *ColCamSeq* is an indoor sequence acquired at a frame rate of 30 fps; it contains 360 frames and the ground truth is composed by 45 frames that cover ≈240 frames of the sequence that are the one where the moving object is present. In this case one every six frames have been labeled. This sequence aims at testing the performance of the algorithms when the color camouflage problem occurs. In the scene a white box is moved in front of a white panel that is part of the background. In Fig. 11 the color data and the depth data of the scene acquired with the Kinect are reported. The results obtained by processing the *ColCamSeq* are reported in Table 3. As it can be noticed the percentage of *FN* obtained with CL_C is very high due to the color camouflage problem. Also the performance of MoG_{bin} and MoG_{ZIV} is affected by this problem since the color features bias negatively the final results. On the contrary the CL_D classifier guarantees a low value of *FN*. The proposed classifier CL_W guarantees an efficient combination of the depth and color data, thus obtaining a low value of *FN* and *FP*. However the absence of useful color information do not allow to improve the silhouette of the detection, so the obtained values for *S* and S_B are very close to the ones obtained by CL_D. In *ColCamSeq* sequence the lowest values of *FN* and *FP* are obtained with MOG_{RGB-D}. In Fig. 12 the foreground masks obtained by analyzing the *ColCamSeq* sequence are shown. As it can be noticed in Fig. 12(c), the color features are not useful for the segmentation, hence the foreground object can not be correctly segmented by

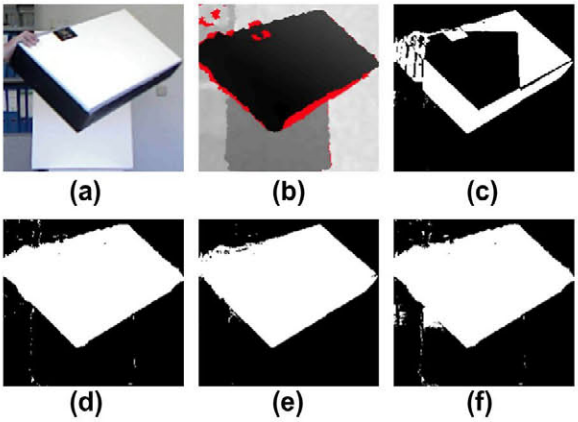

Fig. 12. Frame 934 of the *ColCamSeq* sequence: color data (a), depth data (b), CL_C output (c), CL_D output (d), CL_W output (e), MoG_{RGB-D} output (f).

CL_C. On the contrary, CL_D (Fig. 12(d)) guarantees a more compact foreground object; however, its borders are irregular and noisy due to the irregular depth measurements on the objects boundaries. The proposed approach CL_W (Fig. 12(e)) guarantees a compact silhouette and refined boundaries where the color data is useful (black part of the box). It is worth noting that in this case the CL_D obtains the lowest value of *RM* showing that the depth information is fundamental where the problem of color camouflage occurs. The performance of $ViBe_{Bin}$ and *SOM* are affected by a high level of *FP*.

4.4. DCamSeq results

The indoor sequence *DCamSeq* helps to analyze the performance of the algorithm when depth camouflage occurs. It contains inter-

Table 3
Detection accuracy obtained by analyzing the *ColCamSeq*.

	TE		*FN*		*FP*		*S*		S_B		*RM*
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	
CL_C	39.02	23.12	82.27	1.11	2.27	0.06	0.22	0.22	0.37	0.20	6.40
CL_D	2.47	2.35	2.58	0.05	2.38	0.10	0.91	0.10	0.78	0.11	2.40
CL_W	3.20	2.77	3.52	0.09	2.92	0.10	0.89	0.15	0.77	0.16	3.60
MoG_{Bin}	38.47	22.98	82.87	1.10	0.75	0.04	0.22	0.19	0.35	0.17	6.40
MoG_{RGB-D}	3.49	3.40	0.38	0.02	6.13	0.14	0.91	0.09	0.81	0.08	3.00
SOM	6.49	5.60	0.25	0.01	11.80	0.23	0.84	0.16	0.76	0.08	4.20
mog_{ZIV}	32.89	22.64	69.37	1.11	1.88	0.06	0.34	0.30	0.48	0.30	5.20
$ViBe_{Bin}$	6.94	4.13	0.17	0.01	12.69	0.17	0.81	0.18	0.74	0.06	4.80

Fig. 13. *DCamSeq* sequence: color data (a), depth data (b).

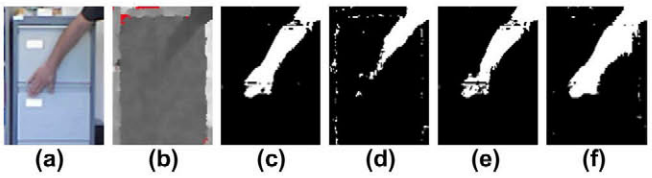

Fig. 14. Frame 1086 of the *DCamSeq* sequence: color data (a), depth data (b),CL_C output (c), CL_D output (d), CL_W output (e), *SOM* output (f).

actions between foreground and background elements of the scene: in particular a person moves towards a file cabinet belonging to the background and interact with it. This sequence has been acquired at a frame rate of 30 fps; it contains 670 frames and the ground truth is composed by 102 frames that cover ≈400 frames of the sequence where the moving object is present, in this case one every four frames has been labeled. In Fig. 13 the color data and the depth data of the scene acquired with the Kinect are reported. Let us consider the depth information provided by the Kinect in the *DCamSeq* and presented in Fig. 14(b). As it can be noticed, the depth data corresponding to the hand is very similar to the drawers' depth values. This is a clear example of depth camouflage and, as it is shown in Fig. 14(d), it dramatically affects CL_D that it is not able to correctly segment the moving hand. On the contrary, the color features enable the color-based classifier CL_C to easily identify the foreground object (Fig. 14(b)). The proposed approach CL_W is able to efficiently combine the color and depth information thus allowing to obtain an accurate detection of the foreground object, as it can be noticed in (Fig. 14(e)). The results obtained with the different algorithms are reported in Table 4. As expected, the depth based classifier CL_D and is affected by a high percentage of *FN* due to the depth camouflage problem. The proposed algorithms CL_W allows to reduce the percentage of *FN* and *FP* with respect to the CL_D and, at the same time, the value of *S* is also increased. The *SOM* algorithm allows to obtain a very low value of *FN* at cost of a higher level of *FP*. In fact, as shown in Fig. 14(f), the segmentation obtained with *SOM* completely detect the hand but it introduces several false positive caused by the hand shadows. Also the performance of the algorithms MoG_{bin} and MoG_{ZIV} are affected by a very high level of *FN* due to the depth camouflage problem.

4.5. StereoSeq results

The outdoor sequence *StereoSeq* is used to test the proposed approach with stereo disparity data; it is composed by 297 frames and the corresponding ground truth is composed by 33 frames spanning the entire sequence where the moving object is present (one every five frames has been labeled). For this sequence, only luminance data is available. In the scene several people are walking on outdoor stairs, a keyframe of this sequence is presented in Fig. 15.

As it can be noticed in Fig. 16(e), the proposed approach guarantees an efficient combination of the disparity and luminance data. Also in this case, CL_C (Fig. 16(c)) is severely affected by color camouflage, even if the borders have been well detected. On the contrary CL_D results in a compact but noisy silhouette (Fig. 16(d)). The segmentation obtained with MoG_{RGB-D} is very compact but the borders are not properly refined (see Fig. 16(f)). The results obtained with the different algorithms are reported in Table 5. As expected the best performance have been obtained with MoG_{RGBD} that was originally proposed to process stereo data; the lowest values for *RM* is obtained with this method. However, CL_W guarantees the lowest value of *TE* and the obtained value of *S* is comparable with the one obtained with MoG_{RGBD}. Also in this case binary combinations of foreground masks lead to poor results. For this sequence also MoG_{ZIV} and *SOM* are severely affected by the low contrast of the luminance data.

Table 4
Detection accuracy obtained by analyzing the *DCamSeq*.

	TE		*FN*		*FP*		*S*		S_B		*RM*
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	
CL_C	1.78	1.47	15.60	0.09	0.95	0.01	0.67	0.13	0.62	0.10	2.80
CL_D	3.38	2.19	48.49	0.44	0.64	10^{-3}	0.40	0.20	0.39	0.18	5.20
CL_W	2.46	1.82	32.21	0.26	0.66	0.01	0.55	0.14	0.51	0.12	4.20
MoG_{Bin}	3.57	2.76	60.87	0.54	0.09	10^{-3}	0.32	0.22	0.27	0.18	6.20
MoG_{RGB-D}	2.11	1.29	15.25	0.09	1.31	0.02	0.61	0.14	0.61	0.11	3.60
SOM	2.11	2.00	2.98	0.02	2.05	0.02	0.70	0.12	0.70	0.07	2.40
mog_{ZIV}	3.33	1.83	45.98	0.32	0.74	0.01	0.36	0.18	0.39	0.16	5.60
$ViBe_{Bin}$	9.31	1.30	5.48	0.04	9.55	0.02	0.30	0.15	0.60	0.07	6.00

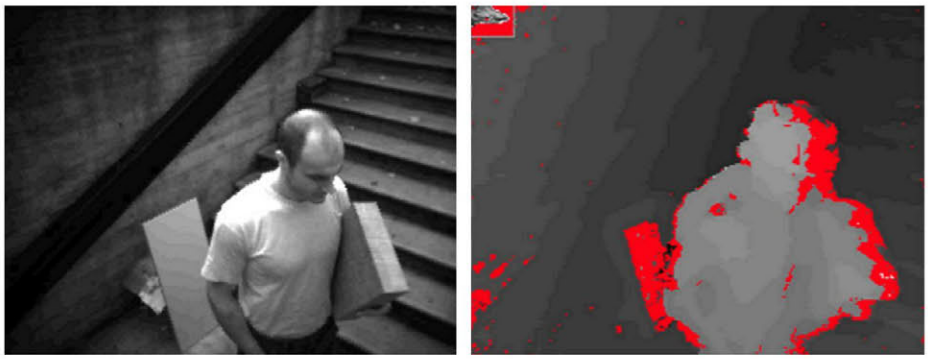

Fig. 15. *StereoSeq* sequence: color data (a), disparity data (b).

4.6. MoveBGSeq results

Sequence *MoveBGSeq* is an indoor sequence acquired at a frame rate of 30 fps; it contains 250 frames. The objective of this sequence is to highlight the impact of the moved background object problems on the algorithms' performance. In the scene there are present two static bags on the floor, that are rapidly removed from their position after ≈130 frames. In Fig. 17 the color data and the depth data of the scene acquired with the Kinect are reported.

In this case only qualitative results are reported since the *new background* is incorporated at different speed by the different algorithms (depending on the corresponding learning rates). In the proposed re-initialization approach it is immediately incorporated to the background, on the contrary for the other algorithm the incorporation is slower. It is clear that different speeds will strongly bias the qualitative comparison. In Fig. 18 the detection of the moved background object is reported. As it can be noticed, the proposed method (Fig. 18(c)) allows to rapidly solve the moved background object problem, in particular the background previously covered by the two bags is correctly classified and not detected as foreground, on the contrary the other approaches (see for example the *SOM* detection results in Fig. 18(d)) wrongly classifies these pixels as foreground.

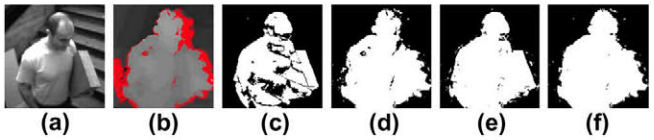

Fig. 16. Frame 139 of the *StereoSeq* sequence: color data (a), depth data (b), CL_C output (c), CL_D output (d), CL_W output (e), MoG_{RGB-D} output (e).

Table 5
Detection accuracy obtained by analyzing the *StereoSeq*.

	TE		*FN*		*FP*		*S*		S_B		*RM*
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	
CL_C	5.73	2.58	26.02	0.35	2.96	0.08	0.59	0.16	0.64	0.13	4.80
CL_D	4.25	1.12	25.78	0.30	1.31	0.03	0.59	0.24	0.42	0.22	4.80
CL_W	3.70	1.54	21.55	0.33	1.26	0.03	0.64	0.25	0.53	0.24	3.00
MoG_{Bin}	6.28	2.03	51.49	0.50	0.09	0.01	0.40	0.26	0.35	0.26	6.20
MoG_{RGB-D}	3.89	1.93	11.98	0.22	2.78	0.06	0.69	0.18	0.62	0.11	2.40
SOM	5.27	2.42	12.66	0.31	4.26	0.08	0.62	0.18	0.58	0.09	4.00
mog_{ZIV}	6.46	3.22	46.36	0.83	1.00	0.01	0.46	0.22	0.46	0.17	5.80
$ViBe_{Bin}$	6.50	1.88	4.87	0.13	6.73	0.08	0.60	0.12	0.58	0.06	5.00

Fig. 17. *MoveBGSeq* sequence: color data (a), depth data (b).

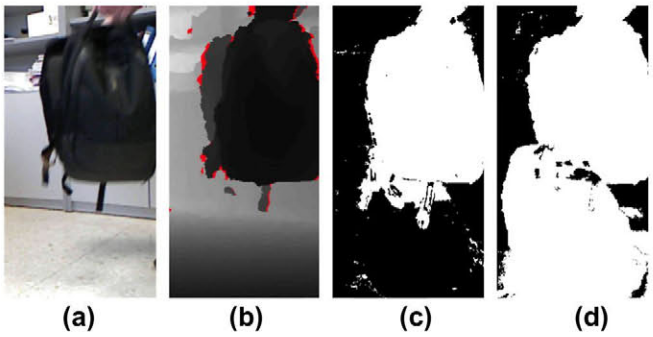

Fig. 18. Frame 467 of the *MoveBGSeq* sequence: color data (a), depth data (b), CL_W detection (c), *SOM* detection (d).

Table 6
RC values for the analyzed algorithms.

	CL_C	CL_D	CL_W	MoG_{Bin}	MoG_{RGB-D}	*SOM*	MoG_{ZIV}	$ViBe_{Bin}$
RC	4.85	3.95	3.15	5.30	3.55	3.90	5.25	6.05

5. Results summary and conclusion

In this paper we present a novel foreground/background segmentation algorithm based on combination of classifiers that allows improving background subtraction accuracy with respect to state of the art algorithms by jointly considering color and depth data. The combination of the two classifiers' output is obtained through a weighted average that adapts, for each pixel, the support of each classifier to the final classification by considering depth and color images edges and the previous foreground detections. In particular, the color based classifier has a greater influence to the final ensemble decision near object borders, thus reducing the problem of noisy depth measurements at object boundaries. On the contrary, the depth based classifier has a greater influence on the final ensemble decision for those pixels located in low gradient areas of the depth map guaranteeing compact detected foreground regions and reduced errors due to shadows and illumination changes. Moreover, the results of the foreground regions obtained in the previous frames are used to check the reliability of the depth based model in order to modify the influence of the color classifier in the ensemble decision and, consequently, to reduce the depth camouflage errors. The proposed method allows also identifying rapidly the pixels belonging to moved background objects and consequently reducing the classification error.

The results section shows that the proposed classifier CL_W results to be the most reliable and accurate one with respect to the other state of the art algorithms. In fact, we have demonstrated that CL_W allows to efficiently combine the independent statistical classifiers improving the overall performance of the *Fg*/*Bg* segmentation. This result is also supported by the *RC* data reported in Table 6, where it is shown how CL_W guarantees the lowest value of *RC*, thus highlighting the robustness of the proposed approach with different benchmark sequences. The other algorithms can eventually lead to a better result for one of the benchmark sequences, but at the same time obtaining not satisfactory results when applied to another sequence. Furthermore, the accuracy and reliability of the proposed classifier CL_W is also apparent considering a qualitative analysis of the examples presented in the previous sections. Further improvements of the proposed algorithm can be obtained if a more sophisticated dynamic model of the foreground is used. In particular, the depth camouflage problem can be still reduced if a RGB-D tracking system is used. In our approach we are using the previous detections to identify the regions where this problem can occur, but this hypothesis is not always true for very fast movements. This problem can be solved by combining the proposed approach with a dynamic model used to predict the objects positions in the next frames. As far as the computational requirements of the proposed strategy are concerned, the most computational demanding block of the proposed system is the estimation of the background likelihood parameters. In our approach we use the Canny algorithm for edge detection, but a less demanding algorithm such as a Sobel filter can be used. The complexity of the edge regions weights calculation is similar to the one of a standard Gaussian filtering. The recursive weights selection and the logistic function can be easily calculated with a LUT; few operations are required to obtain the value of δ. Also the classifiers combination requires few per pixels operations as stated in Section 3.1. For these reasons the complexity of the proposed approach is very similar to the one of others parametric algorithms such as MoG. The pixel-wise approach guarantees an optimized and efficient parallel implementation of the proposed method.

Finally, another important point of this work is the first (for the best of the author's knowledge) publicly available RGB-D benchmark dataset for testing background/foreground segmentation algorithms. The dataset is provided with hand-labeled ground truth; the sequences contain different challenging situations such as cast shadows, color and depth camouflage for the segmentation algorithms.

Acknowledgments

This work has been partially supported by the Ministerio de Economía y Competitividad of the Spanish Government under the project TEC2010-20412 (Enhanced 3DTV). M. Camplani would like to acknowledge the European Union and the Universidad Politécnica de Madrid (UPM) for supporting his activities through the Marie Curie-Cofund research grant.

References

[1] M. Roccetti, G. Marfia, A. Semeraro, Playing into the wild: a gesture-based interface for gaming in public spaces, Journal of Visual Communication and Image Representation 23 (2012) 426–440.
[2] J.P. Wachs, M. Kölsch, H. Stern, Y. Edan, Vision-based hand-gesture applications, Communications of the ACM 54 (2011) 60.
[3] E. Stone, M. Skubic, Evaluation of an inexpensive depth camera for in-home gait assessment, Journal of Ambient Intelligence and Smart Environments 3 (2011) 349–361.
[4] A. Janoch, S. Karayev, J.T. Barron, M. Fritz, K. Saenko, T. Darrell, A category-level 3-D object dataset: putting the Kinect to work, in: International Conference on Computer Vision Workshops (ICCV Workshops), IEEE, 2011, pp. 1168–1174.
[5] M. Cristani, M. Farenzena, D. Bloisi, V. Murino, Background subtraction for automated multisensor surveillance: a comprehensive review, EURASIP Journal on Advances in Signal Processing 2010 (2010) 1–24.
[6] T. Bouwmans, Recent advanced statistical background modeling for foreground detection – a systematic survey, Recent Patents on Computer Science 4 (2011) 147–176.
[7] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, P. Ishwar, changedetection.net: A new change detection benchmark dataset, in: Workshop on Change Detection at CVPR, IEEE, 2012.
[8] M. Camplani, L. Salgado, Efficient spatio-temporal hole filling strategy for Kinect depth maps, Three-Dimensional Image Processing (3DIP) and Applications II, vol. 8290, SPIE, 2012, p. 82900E.
[9] M. Camplani, T. Mantecon, L. Salgado, Accurate depth-color scene modeling for 3D contents generation with low cost depth cameras, in: International Conference on Image Processing, IEEE, 2012, pp. 1741–1744.
[10] K. Khoshelham, S.O. Elberink, Accuracy and resolution of Kinect depth data for indoor mapping applications, Sensors 12 (2012) 1437–1454.
[11] L.D. Stefano, M. Marchionni, S. Mattoccia, A fast area-based stereo matching algorithm, Image and Vision Computing 22 (2004) 983–1005.
[12] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland, Pfinder: real-time tracking of the human body, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1997) 780–785.
[13] C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking, in: Conference on Computer Vision and Pattern Recognition, IEEE, 1999, pp. 246–252.

[14] Z. Zivkovic, F. van der Heijden, Efficient adaptive density estimation per image pixel for the task of background subtraction, Pattern Recognition Letters 27 (2006) 773–780.
[15] A. Elgammal, R. Duraiswami, D. Harwood, L. Davis, Background and foreground modeling using nonparametric kernel density estimation for visual surveillance, Proceedings of the IEEE, vol. 90, IEEE, 2002, pp. 1151–1163.
[16] O. Barnich, M. Van Droogenbroeck, ViBe: a universal background subtraction algorithm for video sequences, IEEE Transactions on Image Processing 20 (2011) 1709–1724.
[17] L. Maddalena, A. Petrosino, A self-organizing approach to background subtraction for visual surveillance applications, IEEE Transactions on Image Processing 17 (2008) 1168–1177.
[18] G. Gordon, T. Darrell, M. Harville, J. Woodfill, Background estimation and removal based on range and color, Conference on Computer Vision and Pattern Recognition, vol. 2, IEEE, 1999.
[19] A. Stormer, M. Hofmann, G. Rigoll, Depth gradient based segmentation of overlapping foreground objects in range images, in: Conference on Information Fusion, IEEE, 2010, pp. 1–4.
[20] J. Leens, O. Barnich, S. Piérard, M. Droogenbroeck, J.-M. Wagner, Combining color, depth, and motion for video segmentation, in: Computer Vision Systems, Lecture Notes in Computer Science, vol. 5815, Springer, Berlin, Heidelberg, 2009, pp. 104–113.
[21] A. Frick, F. Kellner, B. Bartczak, R. Koch, Generation of 3d-tv LDV-content with time-of-flight camera, in: 3DTV Conference, IEEE, 2009, pp. 1–4.
[22] L. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.
[23] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley-Interscience, 2001.
[24] B. Klare, S. Sarkar, Background subtraction in varying illuminations using an ensemble based on an enlarged feature set, in: Conference on Computer Vision and Pattern Recognition Workshops, IEEE, 2009, pp. 66–73.
[25] F.J. Richards, A flexible growth function for empirical use, Journal of Experimental Botany 10 (1959) 290–301.
[26] T. Bouwmans, F.E. Baf, Background modeling using mixture of Gaussians for foreground detection – a survey, Recent Patents on Computer Science 3 (2008) 219–237.
[27] Y. Sheikh, S. Member, M. Shah, Bayesian modeling of dynamic scenes for object detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 1778–1792.
[28] J.-L. Landabaso, A unified framework for consistent 2D/3D foreground object detection, Ph.D. Thesis, Image Processing Department, Technical University of Catalunya, 2008.
[29] P.-M. Jodoin, M. Mignotte, J. Konrad, Statistical background subtraction using spatial cues, IEEE Transactions on Circuits and Systems for Video Technology 17 (2007) 1758–1763.
[30] P. KaewTraKulPong, R. Bowden, An improved adaptive background mixture model for real-time tracking with shadow detection, in: European Workshop on Advanced Video Based Surveillance Systems, Kluwer Academic Publishers, 2001, pp. 149–158.
[31] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (1986) 679–698.
[32] L. Li, W. Huang, I.Y.-H. Gu, Q. Tian, Statistical modeling of complex backgrounds for foreground object detection, IEEE Transactions on Image Processing 13 (2004) 1459–1472.