Real-Time Background Modeling Based on Classified Dynamic Objects for Human Robot Application

Alessandro Moro, Enzo Mumolo, Massimiliano Nolich, Kenji Terabayashi, Kazunori Umeda

Abstract—**The aim of this paper is to describe a flexible and robust background management algorithm. In these years various techniques were proposed to segment the images from a video stream sequence, and detect interesting dynamic objects. Many works faced the problem to segment the image in indoor environment for human detection and intelligent room applications. In these works, both accuracy and efficiency depend on the background model they used. Specific high performances models suffer of some limitations in chaotic unstructured environments. Long video stream sequences changes in light condition, and object and human displacement. In those environments, dynamic to stable objects and humans can be absorbed in the background, and then become invisible to the system. In this work we propose an approach to combine low level and high level information to improve the background management and to solve unpredictable object dynamic problems. Experimental recall and precision results show improved performances with respect to popular background management algorithms. Finally, a real application is shown and discussed.**

***Index Terms*— Background Management, Computer Vision, Classification, Robotic Application**

I. Introduction

Background maintenance has high importance in many computer vision applications. As a main component of the background subtraction [2, 3], an optimal background maintenance is necessary to perform optimal moving object detection. Many applications are based on background quality, such as shadow detection, extraction of environment, image representations without the effect of the dynamic objects, or scene understanding. For instance, Ehinger et al. [6] proposed a method to classify the scene through the analysis of a single image. However, Ehinger's method suffers of noisy effect due to humans and dynamic objects. An important application field is that of intelligent rooms, such as domotic or Ambient Intelligence (AmI), which are increasing in importance in these days. AmI increased his popularity since 1990 [1, 5, 19], accordingly the number of devices, subsystems, and complexity of proposed paradigms increased. These intelligent environments have multiple purposes, even if a common purpose is the automatic control on multiple devices and assistance, to improve the lifestyle of humans. People suffering of some physical disease receive benefit from those environments.

Vision system devices are one of the fundamental sensors for acquiring the human and object information [15, 20]. Typical approaches to detect and extract the human shape in acquired video streams are based on a complete analysis of the image, or based on change detection algorithms. The first group has the advantage to handle environment changes and occlusion problems together but, on the other hand, cannot detect unknown objects. The second has the advantage to be more flexible and possibly to detect unknown objects.

We opted for change detection algorithm class because we are more interested in detecting and classifying multiple categories of objects, and manage, in the case, the unknown objects.

Change detection algorithms [13] are based principally on subtraction methods (thresholds, stochastic, etc.), and for long video stream acquisition at least one factor is considered: the necessity to manage the background model. The background management algorithms can be classified by the feature they are based on: namely temporal [16], spatial [8, 14] features. Many approaches consider several factors in order to dynamically adjust control parameters: local conditions (i.e. variation on pixel intensity), global conditions (i.e. image histogram). These approaches generally do not handle noise or casted shadow effect. Other approaches try to assume that the background can be modeled as a mixture of multiple backgrounds; in this way light effects, shadow, and small changes are combined in order to increase the image segmentation. However in most of the cases this last approach is unable to return the observed background.

Many solutions reported in the literature are unable to manage environments with chaotic dynamic. Observed dynamic objects like humans have unpredictable behavior; sometimes people move, sometimes people stand for long time in the same position. Also other categories of objects show similar unpredictable behavior in particular environments, such as parking.

In this work we propose an algorithm to solve the unpredictable objects dynamic in background management

A. Moro is with Department of Precision Mechanics, Faculty of Science and Engineering, Chuo University / CREST, JST, 1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan, (e-mail: alessandromoro.italy@gmail.com).

E. Mumolo is with DI3, University of Trieste, Via Valerio 10, 34123 Trieste, Italy, (e-mail: mumolo@units.it).

M. Nolich is with IFACE s.r.l., Via Valerio 10, 34123 Trieste, Italy (e-mail: mnolich@units.it).

K. Terabayashi and K. Umeda are with Department of Precision Mechanics, Faculty of Science and Engineering, Chuo University / CREST, JST, 1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan

978-1-4673-1786-3/12/$31.00 ©2012 IEEE

problem by combining different levels of information: local information, global information and classified objects.

The rest of this paper is organized as follows. In Section II we present and discuss the proposed algorithm. In Section III we describe our proposed background management method and in section IV we give a brief description of the GPU implementation. Experimental results and performances comparison are shown in Section V. Finally, in Section VI some final remarks and conclusions are reported.

II. PROPOSED ALGORITHM

A. General Problem

Image segmentation and classification of objects are currently hot topics in computer vision. In dynamic scenes, segmentation performances are strongly related to the capacity to extract the interesting dynamic elements. An observed scene contains several types of dynamic elements. Humans are the most common, but also objects can be introduced into or removed from the scene. A common problem is how to manage new elements into the scene that move through dynamic to static. When the system should consider the new element as an element of the background? A typical approach is to update the background once a certain event is triggered, for example, applying a temporal threshold on the duration of a certain event.

In this paper we describe an improved background management algorithm that is able to add static, dynamic or unknown objects into the background model.

The main contribution of this work is to obtain a reduced computation time because only the foreground pixels are computed, higher segmentation accuracy with respect to the state of the art algorithms, and unknown objects detection.

Thereafter we assume that the blob extraction and classification of objects is a mature technology. We considered the high performances of today's algorithms in

Fig. 1 Example of complex scenario. On the top left a captured frame from a video sequence. This scene contains several typical computer vision hard task. Occlusion, point of view, illumination on the scene change not uniformly and dynamic objects (automatic door). People behavior is not predictable because they can stand, walk, leave objects. The other pictures show the background reconstructed by EHB, MoG, and our proposed method.

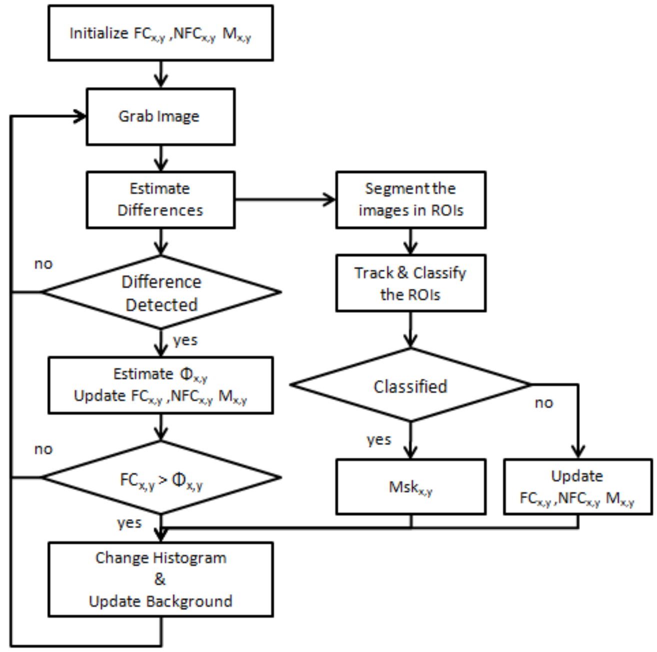

Fig. 2 Block diagram of the proposed algorithm.

blobs extraction and classification, as key factor to improve the background management algorithm.

We propose to combine low level information with high level information, obtained from the work described in [9], in order to improve the background management. In this paper we define low level information as data from not aggregated structures, such as pixels or the whole image.

High level information is the data obtained by aggregated structures, such as tracked regions or the regions labeled as Regions of Interest (ROIs).

We also defined the local information as the data which belong to a single pixel, and global information the data from the whole image (i.e. global intensity). Our method also solves the problems of sparse changing light condition and the chaotic dynamic of the moving elements in the scene. The scheme of the proposed algorithm is shown in Figure 2. In this paper we briefly resume the main components: subtraction stereo, noise reduction, image segmentation, classification.

Image segmentation and object extraction require a reliable background model. We chose a pixel-wise background management algorithm because it has the benefit that the background image obtained from the background model has a higher quality, compared with other background management algorithms, for example statistical [9].

We extended the work described in [11, 16] which shown high performances in dynamic environment, to study the pedestrian flux, but which suffer of absorbtion effect when the objects stay for a long time. We propose a method to improve pixel-wise histogram based algorithms. The key of the proposed method is the combination of histogram based algorithm with the classification of objects.

The histogram based algorithm better reconstructs the background image which improves the performance of other algorithms applied later. Objects classification solves the problem of unpredictability of permanence time.

B. Conceptual Scheme

Proposed conceptual scheme is shown in Figure 2. Compared to other works, we considered two main factors:

- dynamic change of the background model based on the local and global information;
- regions stabilizer based on the regions of interest (ROIs) detected and classified.

Foreground is important in human and object detection.

People move in a given environment. It is possible to extract information about their movements by analyzing the temporal dynamic.

In the literature several works have been proposed about change detection. Some of these works have good performance in specific field but our aim is to develop a flexible system which can be easily adopted in different indoor or outdoor environments.

C. Object Extraction and Classification

The technique used to extract, track, and classify the detected objects have been described in [11]. In this work we will just give a brief explanation about the main components and how they will be used in the background management context. For detailed information please referred to [11].

Candidate dynamic objects of interest are segmented by detection of moving pixels in a video stream sequence. We used a commercial stereo camera system which provided the estimated distance between the camera and observed points as additional information. A change detection algorithm to detect candidate foreground pixel is used [18]. It is called subtraction stereo because the stereo matching between the images from left and right camera is performed after the foreground extraction.

Once the stereo matching is performed, the candidate foreground pixels are refined by noise and casted shadow removal which use the background image to model the shadow. The remaining foreground pixels detected are grouped in clusters by a method proposed by Ubukata et. al. [17], which exploits stereo information. Regions of interest (ROIs) are tracked by a Kalman filter which exploits the stereo information in the transition matrix. Because we consider an indoor environment, the movement of the detected object is considered constant. To classify the detected objects we use a K binary neural network, with a final decision module which selects the final classification results based on the output of all the neural networks. In order to describe the detected ROIs, we used Histograms of Oriented Gradients (HOG), which compared to other features, better describe the human structures and increase the classifier performances.

III. Background Management

Core of all the proposed method is the management of the background. Originally based on the Efficient Histogram Based (EHB) [10], and successively extended in [16], we proposed an adaptive threshold for the update of the background model, which had the advantage to represent the background of dynamic scenes and to be ported in a parallel structure. However, it could not solve the problems of absorption due to dynamic objects which stand for a long time in same place. A common approach is to tune a time variable to adapt the problem to the dynamic of the scene. However this method can be applied to a structured scene, where the dynamic objects keep a similar behavior. On the contrary it cannot manage unstructured or chaotic dynamics. In this work we propose a solution which improves the dynamically updating time based on local and global information. To manage the populated areas which can be cause of ghost regions, we propose a top-down approach which gives higher weight to the detected objects instead to the temporary information, in order to reduce the relevance of the tuning and to follow the dynamic of the scene.

A. Preliminary considerations

Let us call I the current acquired image: in this image the pixel (x, y) at time t is a RGB vector denoted as $I_{x,y}(t)$. Similarly, B is the background image as produced by a background model. The goal of background maintenance is to estimate, at each time t, the background model which produces the background image B.

Furthermore, we call MO the set of pixels of the current image I which reports the Moving Objects and G the set of pixels which appears in motion but does not correspond to any moving object; G is called Ghost image. The ghost is due to the delay introduced by the background model in reconstructing the background image. In the images$I_{x,y}(t)$, $B_{x,y}(t)$, $MO_{x,y}(t)$ and $G_{x,y}(t)$, the pixel (x, y) at time t is a RGB vector. ROIs are composed both by subsets of MO and G. A ROI is perfect if it contains only MO’s pixels.

Once the background image is computed, the moving objects can be detected by subtracting the background B from the current image I, resulting in the difference image

$$D_{x,y} = I_{x,y} - B_{x,y} = \begin{cases} MO_{x,y} - B_{x,y} & \text{if } (x,y) \in MO \\ B_{x,y} - G_{x,y} & \text{if } (x,y) \in G \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

The background image must be reconstructed from a model because the MO makes not visible the real background. $B_{x,y}(t)$ is reconstructed by computing, at time instant t, its RGB values, according to the following equation:

$$B_{x,y}(t) = \begin{cases} B_{x,y}(t - \Delta t_{x,y}) & \text{if } (x,y) \in MO \\ MO_{x,y}(t) & \text{if } (x,y) \in G \\ \text{unchanged} & \text{otherwise.} \end{cases} \quad (2)$$

where the time interval $\Delta t_{x,y}$ is estimated such that no moving objects correspond to the pixel (x, y).

Given a certain amount of frames, a pixel in a certain position will assume several intensity values. The color distribution at each pixel location (x, y) can be described with the histogram $H^c_{x,y}(\cdot)$ which represents the distribution of the intensity value of each color c,($c \in \{red, green, blue\}$), for a given pixel in the background. In the following we consider foreach color channel a color depth of 8 bits. The histogram is updated

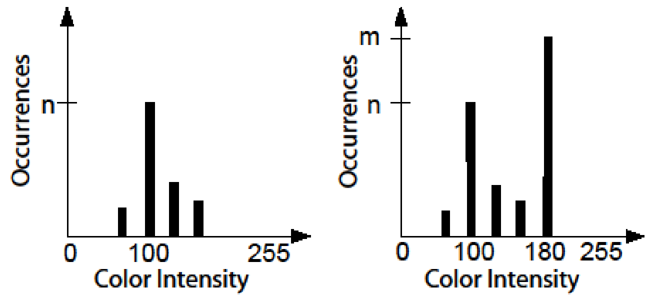

Fig. 3 Example of intensity histogram. On the left, at time t', on the right at time t''.

considering the intensity value of the color c in the current image I^c in the same pixel (x,y), as follows:

$$H^c_{x,y}(t+1,a) = H^c_{x,y}(t,a) + \delta|I^c - a|, \qquad 0 \le a \le 255 \tag{3}$$

$$\delta|p-q| = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$$

where $\delta(\cdot)$ is the Dirac delta function. The histogram is periodically rescaled in order to avoid saturation.

In other words, if the pixel (x, y) of the current image represents always the same, fixed, point of an object, its histogram continuously increases at each frame. For example, Fig 3 left panel, is the histogram of a pixel pointing to an object with color intensity equal to 100 after *n* frames, assuming that initially is equal to 0. If however an object with color intensity equal to180 covers the same pixel for *m* frames (m > n), its histogram becomes as in Fig.3 right panel. $H^c_{x,y}(\cdot)$ is a model for the current background. From this model a background image is reconstructed by analysis of the histogram, since the height of the peak represents the color intensity which characterized the pixel for the longest time. In [15] it has been described an algorithm, called Histogram Based (HB) in the following, where the background image is obtained extracting the peak value of the histograms: a pixel is considered foreground if it is significantly different from the current background estimation. In [17], an improvement called Efficient Histogram Based (EHB) update the background not periodically, but only when changes occur for a certain amount of time. To adapt to the changes in the scene, the number of Found Changes (FC) of each pixel is computed. If a color intensity variation is frequently detected in a period of time and FC is above a given threshold, the background image is updated.

From histogram model a background image is reconstructed by choosing the color with the highest value in the histogram, since the height of the peak represents the color intensity which characterized the pixel for the longest time. If B^c represents the intensity of the background image,

$$B^c_{x,y}(t) = \operatorname{argmax}(H^c_{x,y}(t)) \tag{4}$$

if the moving object is not seen by the pixel anymore, still its color continues to remain the highest peak for some time, until other peaks become higher due to its increasing. If we reconstruct the background image, thus it still appears belonging to a moving object although the moving object is not there anymore: this is called a ghost.

B. *Proposed enhancement*

The histogram based algorithms have suitable characteristics for parallelization, because each pixel or block of pixel can be associated to a separate thread. Also they have good quality of the resulting background, as shown in the experimental results (see Sec. V). On the other hand, the methods in literature propose a fixed threshold value for the management of the background applied equally to all the pixel of the image. The proposed algorithm, instead, estimates an adaptive threshold different for each pixel. In principle, this lead to increasing the background image quality at the cost of increasing computational burden; however, we expect that this burden is absorbed in an eventual GPU implementation.

The difference vector Δ is calculated as follows:

$$\Delta = \left[\left|I^R_{x,y} - B^R_{x,y}\right|, \left|I^G_{x,y} - B^G_{x,y}\right|, \left|I^B_{x,y} - B^B_{x,y}\right|\right] \tag{5}$$

where (x, y) is the pixel position, I^c the intensity of the current image for the channel c, c = (Red; Green; Blue), B^c the intensity of the background image and $\tau = [\tau^R, \tau^G, \tau^B]^T$ is a vector of thresholds used to detect changes in each channel.

For each imageI^c, at each frame t (with t > 1), the color distribution for each pixel x, y is calculated using histogram analysis:

$$H(t, I^c) = \begin{cases} H(t-1, I^c) + 2\delta|I^c - B^c| & \text{if } \Delta \ge \tau \\ H(t-1, I^c) + \delta|I^c - B^c| & \text{otherwise.} \end{cases} \tag{6}$$

Additionally the changes are marked into a binary matrix M as follows:

$$M_{x,y}(t) = \begin{cases} 1 & \text{if } \Delta \ge \tau \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

At each frame t, the total of FC and Not Found Changes (NFC) are updated as shown in (8) and (9). The parameter *U* is the time parameter that is associated to the update rate of the background model. Its value depends on the type of scene observed. A scene with few changes (almost static scene) will have a very high U value. On the other hand, U will assume a low value for scene with high dynamic (for example Highway). In our test, we empirically estimated a value of 100.

$$FC_{x,y}(t) = \begin{cases} FC_{x,y}(t-1) + 1 & \text{if } \Delta \ge \tau \quad (5) \\ 0 & \text{if } \Delta \ngeq \tau \wedge NFC_{x,y}(t) = U \\ FC_{x,y}(t-1) & \text{otherwise.} \end{cases} \tag{8}$$

$$NFC_{x,y}(t) = \begin{cases} NFC_{x,y}(t-1) + 1 & \text{if } \Delta \ngeq \tau \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

FC and NFC are used to trigger the background updating phase. The main goal of FC is to update the background model and image once the observed background is estimated as changed. Instead NFC is used to correct small changes in long period of observation. It is a different background maintenance problem and used to refine the background image. Over long acquisition time, if a pixel has small variations under the thresholdτ, it can have changed its value.

So, periodically, the background image is reconstructed from the histograms model even for unchanged pixels.
Introducing a weight $\alpha_{x,y}$ on the variability of the intensity of the pixel x, y:

$$\alpha_{x,y} = \frac{1}{\max(1,\sigma(x,y))} \cdot \left(1 - \gamma \frac{\sum_{i=1}^{T} M_{x,y}(i)}{T}\right) \tag{10}$$

and a weight $\beta_{x,y}$ on the number of changed pixels:

$$\beta_{x,y} = (1-\gamma) \cdot \left(\frac{\sum_{x,y} M_{x,y}(t)}{\# \text{Pixels}} + 1\right) \tag{11}$$

we compute the threshold $\phi_{x,y}$ as

$$\phi_{x,y} = \left(\alpha_{x,y} - \beta_{x,y}\right) \cdot U \tag{12}$$

Where γ weight the contribution given by local pixel (α) and the whole image (β), and typically is equal to $\frac{1}{2}$, and ϕ represents the local dynamic threshold in order to update the background model, and background image. This value is important to follow the different dynamics inside an observed scene. A value of $\beta > \alpha$ mean that the scene is greatly changed compared to the model, and that the background model, and background image should be updated to follow the dynamic of the scene. This event happens when, for example, the light conditions on the entire image suddenly change.
Thus, if $FC_{x,y} > \phi_{x,y}$ the pixel in the background is considered to be changed and hence its histogram model is updated as follow:

$$H_{x,y}^{c}(t,a) = \begin{cases} \frac{H_{x,y}^{c}(t,a)}{2} & \text{if } a = \operatorname{argmax}\left(H_{x,y}^{c}(t)\right) \wedge FC_{x,y} > \phi_{x,y} \\ H_{x,y}^{c}(t,a) & \text{otherwise.} \end{cases} \tag{13}$$

C. *Known Object Management*

Environments where the dynamic of the scene can be unpredictable, such for instance living rooms, mines the effectiveness of the background management. We propose to use the high level information obtained from the detection and classification procedure to refine the background model in order to increase the performance of the complete system. If a detected ROI is associated to a known object, the pixels which belong to the classified objects will be not used to model the background for an extended period. Eq. 8 will be extended as follow:

$$FC_{x,y}(t) = \begin{cases} FC_{x,y}(t-1) - 1 & \text{if } \Delta \geq \tau \wedge \text{classified} \\ FC_{x,y}(t-1) & \text{if } \Delta < \tau \wedge \text{classified} \\ FC_{x,y}(t-1) + 1 & \text{if } \Delta \geq \tau \wedge \overline{\text{classified}} \\ 0 & \text{if } \Delta \ngeq \tau \wedge NFC_{x,y}(t) = U \wedge \overline{\text{classified}} \\ FC_{x,y}(t-1) & \text{otherwise} \end{cases} \tag{14}$$

where the updating time for the pixels which belongs to a classified objects is temporary frozen.

D. *Unknown Object Management*

Detected blobs are not always classified because they do not belong to any known object or because errors due to occlusions effects, orientations, or errors of the classifier. In most of the cases big structures would represent objects, however it is important to update the background if they represent changes of environment conditions.
We used a sigmoid function to estimate the time extension before a certain point of the image is updated. Objects detected and classified represent stability to the system. In Eq. 15 the sigmoid function used

$$\varsigma_{x,y} = \frac{1}{1+e^{-\frac{c}{n}}}, \ c \leq n \tag{15}$$

where n are the number of frames the object is tracked and c the total number of frames the object is classified. The new computed threshold to update the background becomes as follow:

$$\phi_{x,y} = \left(1 + \varsigma_{x,y}\right) \cdot \left[\left(\alpha_{x,y} - \beta_{x,y}\right) \cdot U\right] \tag{16}$$

ς works as temporal extension before to update the background model. It guarantees that an object classified as *unknown* is available for a longer period. On the other hand also that in case of errors a detected object is removed.

IV. GPU Implementation

Graphics Processing Units (GPU) computing turns the massive floating-point computational power of a modern graphics accelerator's shader pipeline into general-purpose computing power. When utilizing a GPU, there are several things that must be considered, as the internal structure of the GPU is completely different from the internal structure of CPUs. Originally designed for rasterization and graphics primitives, nowadays are more likely fast multi-core processors capable of performing complex mathematical task.

Each GPU contains multiprocessors (MPs) which each contains several Scalar Processors (SPs, usually eight) and additional memory. Typically GPU are designed to optimize SIMD-type processing, and modern GPU can have several hundred of stream processors. Our approach is implemented using Nvidia's CUDA programming model. The CUDA programming model, mapping the software CUDA block to a

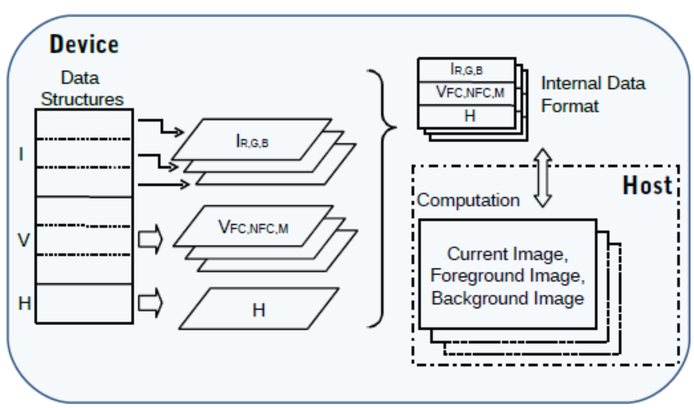

Fig. 4 Data structure used in the parallelized algorithm.

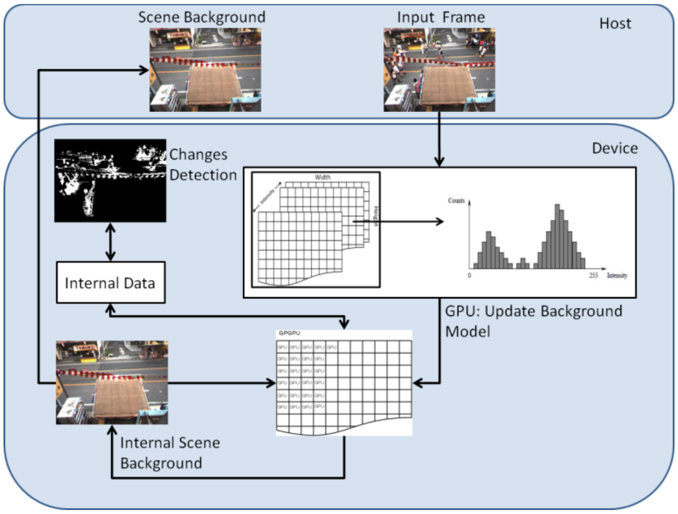

Fig. 5 Data management in the parallelized algorithm.

hardware CUDA multiprocessor. A number of blocks can be assigned to a multiprocessor and they are time-shared internally by the CUDA programming environment

The GPU can manage, schedule and execute many threads in hardware to avoid high thread management overhead. These threads are organized as a large grid blocks. Each block is a 3D structure that can contain up to 512 threads with maximum size of 512x512x64, and the grid is a 2D structure with maximum size of 65535x65535.

The threads are executed by assigning thread blocks to MPs.

The multiprocessor will split the thread blocks into sets of 32 threads known as warps. When executing instructions, the MP will select a warp that is ready to execute ad issue it an instruction.

The proposed approach has been implemented on GPU: each acquired image is divided into 8x8 pixel blocks and for each block a thread pool of independent threads is instantiated. A big amount of memory is required because, for each pixel, inside the GPU, for each concurrent thread several data structure have to be stored, namely the three histogram H_c, M, Msk, FC and NFC. A schema of the data structure is represented in Fig. 4. Each thread updates the model of a single pixel of the background. As the pixels are update by independent threads, this approach does not require inter-thread communication to synchronize the thread operations. A schematic representation of the overall parallelized algorithm is reported in Fig. 5.

V. Experimental Results

We have evaluated our proposed method to manage the background with a dataset of 10000 frames at 20 frames/s with a resolution of 320x240 pixels, obtained with a stereo camera Bumblebee2, where multiple objects and humans are inside the scene (Fig. 7 and 8). These results have been computed on one core of an Intel Core 2 Quad Q9550 CPU running at 2.83 GHz. Computation time is about 340ms for CPU implementation and 25ms for GPU implementation. In our experiment we considered a simple environment where the dynamic known objects were limited to humans, chair, and balls. We used 200 images to train each model, and 400 images for the "no classified" models. Generally the quality of a background model, using background subtraction approaches, is measured through the segmentation of the foreground objects. We have considered the following measures:

Similarity. If A is an extracted foreground region and B is the corresponding ground truth region, the similarity S between A and B is defined as [13]:

$$S(A,B) = \frac{A \cap B}{A \cup B} \tag{17}$$

This nonlinear measure approaches to one (the maximum value) if A and B are the same and approaches to 0 when A and B are completely different. Thus false positive and negative errors are integrated.

Recall-Precision. These are two widely used metrics for evaluating the correctness of a pattern recognition algorithm. They can be seen as extended versions of accuracy, a simple metric that computes the fraction of instances for which the correct result is returned. Recall and precision are defined in [9] as:

$$\text{Recall} = \frac{\#\,\text{Correct Foreground Pixels Identified}}{\#\,\text{Foreground Pixels in Ground Truth}} \tag{18}$$

$$\text{Precision} = \frac{\#\,\text{Correct Foreground Pixels Identified}}{\#\,\text{Foreground Pixels Identified}} \tag{19}$$

We can qualify how well a background model works by matching its results to the ground-truth. When using precision

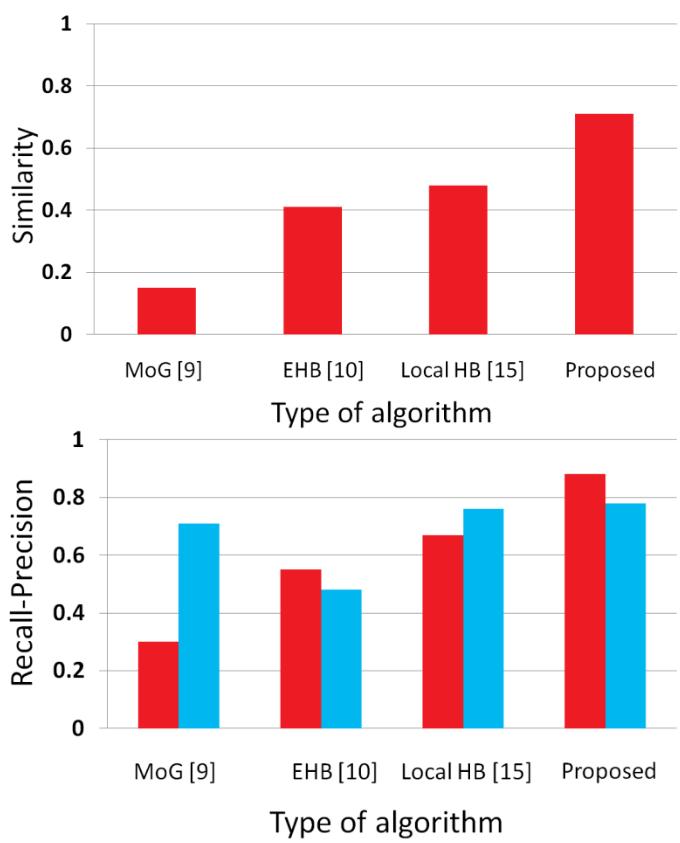

Fig. 6 Similarity and Recall-Precision results.

TABLE I
CLASSIFICATION RATE

Objects	Human	Chair	Ball	NC
Human	0.92	0.03	0.01	0.04
Chair	0.05	0.90	0.01	0.04
Ball	0	0	1	0

and recall, the set of possible labels for a given instance is divided into two subsets, one of which is considered ”relevant” for the purposes of the metric. Recall is then computed as the fraction of correct instances among all instances that actually belong to the relevant subset, while precision is the fraction of correct instances among those that the algorithm believes to belong to the relevant subset. Precision can be seen as a measure of exactness or fidelity, whereas recall is a measure of completeness.

Clearly, a high quality measure is when the values of precision and recall are both high.

In our experiments the proposed algorithm reconstruct the background environment with the accuracy of 99.5% compared with [13] which could reach only the 93.5%.

There is a constant presence of humans or objects in the scene which can be distributed as follow: 7000 frames with humans, 4000 chairs, 2000 balls.

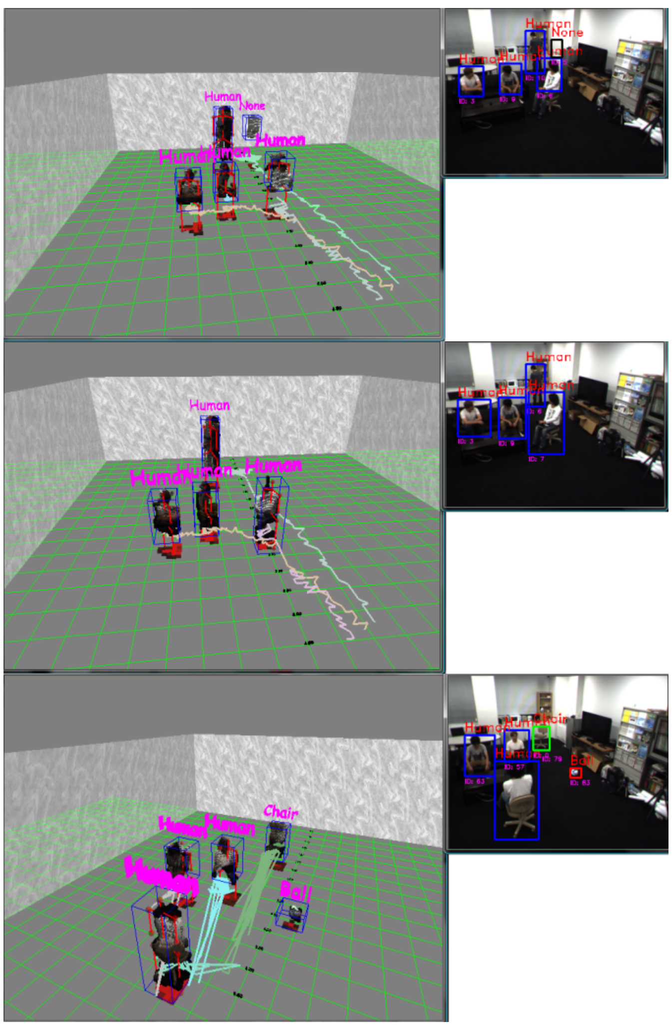

Fig. 7 Example of ROIs detected and classified. On the top a pixel wise algorithm [16], and our proposed method.

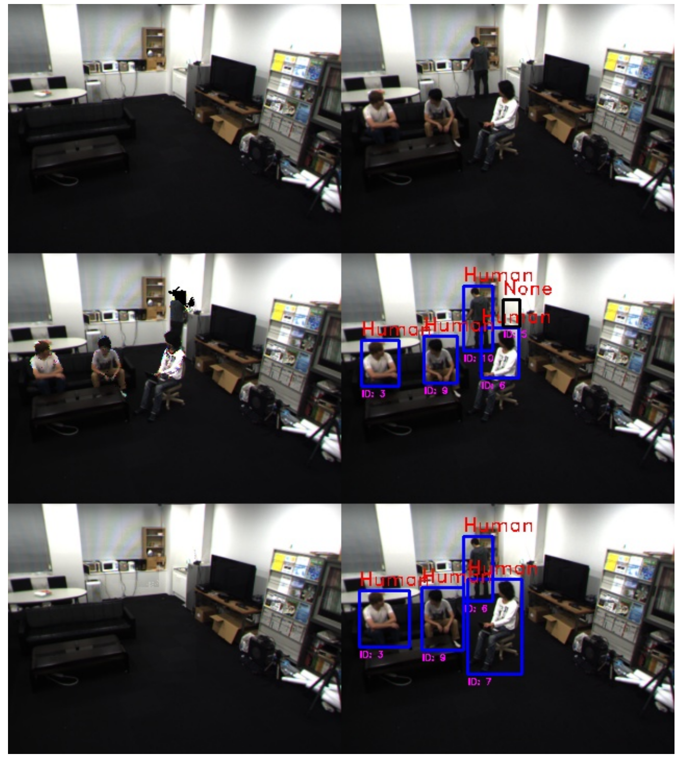

Fig. 8 Reconstructed background. On the left side the background, on the right side the scene after 2 minutes. From top to bottom. Raw scene, ROIs classified with pixel wise update [16], and with proposed method.

A. Case of study

We developed some applications of the algorithm described in this paper. The applications perform visual scene segmentation, moving objects detection and classification and object position estimation.

In this section we report some qualitative results obtained from the following application. According to what described in [12], a human operator, looking at the visual scene acquired by a fixed camera in an indoor environment, interacts with a mobile robot moving in the same environment using commands embedded in natural language phrases such as: "go to the human on the left" or "go to the human in the middle".

Such phrases are processed by a left-to-right (LR) parser which extracts the commands from the phrase. For example, from the first phrase the following keywords are extracted using the LR parser: "go human left" and used to identify the moving human on the left first, to evaluate its position in the world space and to direct a mobile robot to that position.

In Fig. 8 a pixel-wise background maintenance algorithms is used to extract moving objects from the scene. An example of a common problem of this type of applications is represented by the absorption of human due to the prolonged permanence in the same position. Our proposed method allows to properly detect users, dynamic objects, and the extract the necessary information to control a domestic robot.

B. *Some remarks on the proposed method*

In this paper we propose a method to improve moving objects extraction from the visual scene using background subtraction. The method has been used in human-robot interaction applications with phrases written in natural language. The goals of the developed applications are to count or track the moving objects. We obtain high performances by developing a stable and robust background. With the proposed method, we improved the objects extraction of elements which, because the behavior is unpredictable, can stand for long time, preserving the capability of the image subtraction algorithms to detect unknown objects. We used an RGB color space to represent the image, because discrete representations can be mapped inside a fixed amount of memory. Even if our method can be used with other color spaces, we took in consideration only discrete representations.

VI. CONCLUSION

In this paper, we described a robust and parallelizable background management algorithm. The main contribution of this paper concerns the background managing based on low and high level information extracted from the visual scene and combined together. Using recall-precision tests we experimentally show that the proposed method leads to significant improvements in object extraction in a complex scenario. The proposed model has been efficiently implemented on a GPU. A case study based on the application of the described method in a human-robot interaction problem is described.

REFERENCES

[1] Altamirano, I.C.; Henriquez, F.R.; , "A scalable intelligent room based on wireless sensor networks and RFIDs," *Electrical Engineering Computing Science and Automatic Control (CCE), 2010 7th International Conference on* , vol., no., pp.434-439, 8-10 Sept. 2010

[2] T. Bouwmans, "Recent Advanced Statistical Background Modeling for Foreground Detection: A Systematic Survey", Recent Patents on Computer Science, Volume 4, No. 3, pages147-176, September 2011.

[3] T. Bouwmans, F. El Baf, B. Vachon, "Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey", Recent Patents on Computer Science, Volume 1, No 3, pages 219-237, November 2008.

[4] Chen Rui, Hou Yi-bin, Huang Zhang-qin, Zhang Yong, Li Hui. "A Framework for Local Ambient Intelligence Space: The AmI-Space Project". Computer Software and Applications Conference, 2007. COMPSAC 2007 - Vol. 2. 31st Annual International. 2007(2): 95-100.

[5] Dertouzos M., "The Future of Computing," *Scientific American*, vol. 281, no. 2, 1999, pp. 52-55

[6] Ehinger, K.A., Xiao, J., Torralba, A., & Oliva, A.. "Estimating scene typicality from human ratings and image features", in L. Carlson, C. Hölscher, & T. Shipley (Eds.), Proceedings of the 33rd Annual Conference of the Cognitive Science Society (pp. 2562-2567), 2011.

[7] Elgammal A., Duraiswami E., Harwood D., Davis L., Background and foreground modeling using nonparametric kernel density estimation for visual surveillance, Proceedings of the IEEE 90 (7) (2002) 1151 – 1163.

[8] Huwer S., Niemann H., Adaptive change detection for real-time surveillance applications, in: Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on, 2000, pp. 37 – 46

[9] Li L., Huang W., H. Gu I. Y, Tian Q.;, Statistical modeling of complex backgrounds for foreground object detection, Image Processing, IEEE Transactions on 13 (11) (2004) 1459 – 1472.

[10] C.-M. Kuo, W.-H. Chang, S.-B. Wang, C.-S. Liu;, An efficient histogram-based method for background modeling, in: Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on, 2009, pp. 480 – 483

[11] Moro A., Mumolo E., Nolich M., Umeda K., "Real-Time GPU Implementation of an Improved Cars, Pedestian and Bicycles Detection and Classification System", IEEE Conf. Intelligent Transportation Systems 2011, pp.1343-1348, Washington D.C., October 2011

[12] Moro A., Terabayashi T., Umeda K., Mumolo E., "A Framework for the Detection and Interaction with Pedestrian and Objects in an Unknown Environment", IEEE International Conference on Networked Sensing Systems 2010, pp.257-260, June 2010

[13] Radke, R.J.; Andra, S.; Al-Kofahi, O.; Roysam, B.; , "Image change detection algorithms: a systematic survey," *Image Processing, IEEE Transactions on* , vol.14, no.3, pp.294-307, March 2005

[14] Stauffer C., Grimson W., Adaptive background mixture models for real-time tracking, in: Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on., Vol. 2, 1999, pp. 22 – 29.

[15] Trivedi, M.M.; Huang, K.S.; Mikic, I.; , "Dynamic context capture and distributed video arrays for intelligent spaces," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* , vol.35, no.1, pp. 145- 163, Jan. 2005

[16] Terabayashi K., Moro A., Umeda K., "Nonparametric approach to background updating using a GPU," Proceedings of The 2010 IEEJ Conference on Electronics, Information and Systems, TC10-7, pp.398-399, 2010.9.

[17] Ubukata T., Terabayashi K., Moro A., Umeda K.: Multi-Object Segmentation in a Projection Plane Using Subtraction Stereo, Int. Conf. Pattern Recognition (ICPR), pp.3296-3299, August 2010

[18] Umeda K., et al., Subtraction Stereo - A Stereo Camera System that focuses on Moving Regions, Proc. Of SPIE-IS&T Electronic Imaging, 7239 Three-Dimensional Imaging Metrology (2009).

[19] Weber W., et al. "Ambient intelligence-key technologies inthe information age", *IEEE International conference on Electron Devices*, 8-10 Dec.2003, pp111 - 118.

[20] Weerachai, S.; Mizukawa, M.; , "Human behavior recognition via top-view vision for intelligent space," *Control Automation and Systems (ICCAS), 2010 International Conference on* , vol., no., pp.1687-1690, 27-30 Oct. 2010