AN APPEARANCE BASED APPROACH FOR CONSISTENT LABELING OF HUMANS AND OBJECTS IN VIDEO

Martí Balcells

Daniel DeMenthon

David Doermann

{balcells, daniel, doermann, } @ umiacs.umd.edu

Language and Media Processing Laboratory

CFAR/UMIACS

University of Maryland

College Park, MD, 20742, USA

AN APPEARANCE BASED APPROACH FOR CONSISTENT LABELING OF HUMANS AND OBJECTS IN VIDEO

ABSTRACT

We present an approach for consistently labeling people and for detecting human-object interactions using mono-camera surveillance video. The approach is based on a robust appearance based correlogram model which is combined with histogram information to model color distributions of people and objects in the scene. The models are dynamically built from non-stationary objects which are the outputs of background subtraction, and are used to identify objects on a frame-by-frame basis. We are able to detect when people merge into groups and to segment them even during partial occlusion. We can also detect when a person deposits or removes an object. The models persist when a person or object leaves the scene and are used to identify them when they reappear. Experiments show that the models are able to accommodate perspective foreshortening that occurs with overhead camera angles, as well as partial occlusion. The results show that this is an effective approach able to provide important information to algorithms performing higher-level analysis such as activity recognition, where human-object interactions play an important role.

KEYWORDS: surveillance video, background subtraction, tracking, action recognition, object detection, correlograms.

1. INTRODUCTION

In recent years, automated processing of surveillance video has gained a lot of attention. While applications that involve indexing, retrieval and browsing for obtaining evidence after the occurrence of an event are still necessary, the ultimate goal is to provide real-time surveillance, detection and alerts. At a minimum, when possible threats or suspicious activities have to be detected, components at the lowest level of video analysis will be required to automatically answer questions like "how many people are in the scene?", "what is the identity of each person?", "where were these people before?", and "with whom or what have they interacted". In a teleconferencing environment, similar questions need to be answered if a system is to automatically manage the cameras and decide, for example, which scenes should be shown to each person. There are a number of applications where the availability of such information could be used, for example video retrieval, interfaces to games, etc.

Our system is designed to act as an intermediate level of processing, sitting on top of low level background subtraction, and feeding higher level analysis such as activity recognition, object recognition or person recognition. Our primary objective is to maintain a consistent labeling of humans through a video sequence, maintaining continuous labeling even if a person leaves the scene for a period of time and returns. (In the remainder of the text, when describing our approach we will use the term "consistent labeling" rather than "tracking", as the term "tracking" generally implies a reliance on a motion model or spatial constraints, alone or in combination with visual appearance constraints, while our approach relies entirely on visual appearance constraints.) Simple interactions between humans and objects, such as a person removing an object from the scene or depositing a new object in the scene, should also be detected. The system should be able to handle a wide range of video sequences from many different sources, and the algorithms must work with degraded or compressed data. Furthermore, no specific assumptions should be made about the

camera angle or position. One constraint we impose, however, is that the video be taken with a static camera.

This study is focused on indoor sequences since it is assumed that the humans are relatively large with respect to the artifacts introduced by the background subtraction stage. This is a reasonable assumption for indoor but might not be always valid for outdoor sequences, where scenes often have a wide range of depths and the sizes of humans and objects can be very small. Although there will always be situations under which the performance of the system becomes unacceptable, our experiments show that it can handle a wide range of situations.

The implemented system has two modules. One is responsible for human labeling, and the other is responsible for detecting and labeling objects that are being dropped or taken. In both scenarios, an appearance-based model is used to segment humans and objects and label them, even under partial occlusion. This model is based on a color correlogram consisting of a co-occurrence matrix that gives the probability that a pixel at a distance k from a given pixel of color c_i is of color c_j.

Our system consists of several stages. In the first stage the background is subtracted and the foreground regions are detected. After a cleaning stage, only blobs larger than a specified size are preserved. For each foreground blob, the color model is initialized and a distance measure is used to consistently label the blobs for the rest of the sequence. The model accommodates the presence of partial occlusions, pose variations and illumination changes, thanks to partial updating at every frame. When two blobs merge into one, the color model of each blob is used to classify pixels as belonging to one or the other using a maximum likelihood criterion. A new color model is then initialized for the group and used to label the group until the components split again.

Detection of human-object interaction is also based on background subtraction. When a blob splits into two and one of them is static, we conclude that the static blob is generated by an object that has been deposited or removed from the scene and therefore a human-object interaction will be detected. Image gradients (edges) are then evaluated around the static blob boundaries and if they

are greater in a background frame (a previous frame with no foreground blobs at the region where the static blob has been detected) than in the frame where the interaction was detected, we conclude that the blob corresponds to an object that has been picked up. Otherwise it corresponds to an object that was dropped. A color model is then initialized for the object and it is used, in the case of an object being picked up, to segment and keep track of the object for the rest of the video sequence, or in the case of a deposited object, to keep track of the object backwards in time.

The remainder of this paper is organized as follows. In Section 2, we survey related literature. In Section 3 the background subtraction algorithm used is described briefly. In Section 4, the appearance based model used for consistent labeling is introduced. Methods for updating the model, calculating the distances between two models and segmenting under occluding conditions are also presented. In Sections 5 and 6, the systems used for labeling humans and objects respectively are presented and several experiments are described. Finally, Section 7 presents a summary and conclusions.

2. SURVEY OF RELATED WORK

Over the past several years, a significant number of papers [1-11] have reported progress on detecting and tracking humans. However, because of the inherent complexity of the task, problems are still far from being solved, even with the use of a single static camera.

Important work in the field was carried out by Haritaoglu et al. [1]. They implemented a real-time surveillance system for detecting and tracking multiple humans in an outdoor environment. The system works with high-quality uncompressed gray scale imagery and employs a combination of shape and motion analysis to locate body parts and objects that humans may carry. When humans are isolated, they are tracked using motion models; when they merge into groups, the system is capable of tracking each individual as long as the head is visible. From simple silhouette analysis, objects are also detected and tracked during exchange.

McKenna et al. [2] implemented a complete system for tracking multiple humans in a relatively unconstrained environment. They perform tracking at three levels of abstraction: 1) regions – blobs resulting from a background subtraction algorithm, 2) humans – one or more regions grouped together, and 3) groups – one or more people grouped together. Through the use of color cues they provide estimates of depth ordering and position when humans merge into groups. The tracking is done by matching regions in overlapping bounding boxes. The system is able to detect simple interactions with objects such as removing or depositing an object, but they did not differentiate between different objects and they did not try to track the objects.

In [6], Senior et al. use an appearance based model to track humans through occlusions. The models are used to localize objects during partial occlusions, detect complete occlusions and resolve depth ordering of objects. Tracking is similar to McKenna et al. [2]; foreground regions are matched using a bounding box distance measure. Fuentes and Velastin [8] present a real-time tracking system that matches the blobs from two consecutive frames using what they call direct and inverse matching matrices and they are able to successfully detect the merging and splitting of blobs. In [7], Stauffer and Grimson use a linear predictive multiple hypothesis tracking algorithm to match the connected components obtained from a background subtraction algorithm. The system is able to classify objects belonging to two classes, cars or humans, using aspect ratio. However their approach does not handle occlusion.

In general, appearance based models (as opposed to, for instance, 3D models [9]) have been popular for tracking applications, specifically for those dealing with multiple humans simultaneously. In [10], Elgammal and Davis developed an appearance based model to segment humans under occlusion using a maximum likelihood criterion. Assuming people in an upright position, the researchers estimate color distribution of head, torso and legs using a kernel density estimation method. Color distributions have also been effectively modeled using both histograms [2, 4, 12] and Gaussian mixture models [4, 13] estimated from color data using an expectation-

maximization algorithm. We follow a different approach and model humans and objects appearance using a color correlogram [14, 15]. A color correlogram is a special type of co-occurrence matrix. Co-occurrence matrices were first introduced by Haralick et al. [16] in 1973 to extract features for image classification. In fact, color co-occurrences matrices can be seen as a specific kind of geometric histogram, introduced by Rao et al. in [17]. In [18] color correlograms were used for object recognition and matching and in [14, 19] for image retrieval. In Section 4, color correlograms are described in detail.

3. CODEBOOK-BASED BACKGROUND SUBTRACTION

A common approach for detecting moving objects from a video sequence captured with a static camera is to use a background subtraction algorithm. In our system we use the algorithm developed by Kim et al. [20] that models the background using quantization/clustering techniques [21]. The algorithm encodes the time series of the observed color values for each pixel location over time into one or more codewords. The number of codewords required for each pixel is variable, but rarely exceeds six. These codewords are obtained by a vector quantization technique, which essentially operates a clustering of the observed time series of color values at each pixel location. The similarity measure for this clustering is based on a color distortion measure and a brightness range.

Once the background has been encoded, to detect foreground objects in a new frame, the algorithm looks at each pixel location of the frame, and for each of these locations it compares the pixel color and brightness to those stored in the codewords describing the background at this location. A pixel is classified as a background pixel if it satisfies two conditions – (1) the color distortion of the pixel with respect to some codeword is less than a detection threshold, and (2) its brightness lies within the brightness range of that codeword. Otherwise, it is classified as foreground. For a comparison of this technique and other background subtraction techniques, see Kim et al. [20]

4. APPEARANCE BASED MODEL

As our main objective is to consistently label humans from frame to frame in video, the model used for representing people should capture relevant information to differentiate between different humans and recognize humans that have been seen by the system before. Unfortunately, humans are complex deformable entities and it is difficult to find a single good model that can handle all possible situations. A good model should be invariant to rotation, translation and changes in scale, and should also be robust to partial occlusions, deformations and changes in illumination. The model should allow segmentation of objects from people and consistently label humans even under occlusion. We use a color correlogram as the main model to represent humans and objects appearance. This model is used for labeling humans from frame to frame or for correctly identifying people reentering the scene. However when people merge into groups and we need to segment each individual, or when someone picks up an object and we want to differentiate the object from the person, besides the color correlogram, we also use histogram information, as described in Section 4.3.

Color histograms have been widely used for appearance modeling [2, 14, 22] as they are relatively invariant to changes in object orientation, scale, partial occlusion, viewing position and object deformation. However, color histograms capture only the color distribution in an image and do not include any spatial correlation information and therefore have limited discriminative power. A color correlogram [14], on the other hand, is a co-occurrence matrix $\gamma_I(c_i, c_j, k)$ [16] that gives the probability that a pixel at a distance k from a given pixel of color c_i is of color c_j. In [14] Huang et al. used color correlograms for image indexing and other applications (image subregion querying, subimage localization and cut detection) and their experiments showed that this model is able to outperform both the traditional histogram and the histogram refinement method for image indexing/retrieval introduced by Pass and Zabih in [23]. This model is robust enough to handle

common deformations and partial occlusions and at the same time is simple enough so that it is fast to update and therefore can adapt easily to illumination or pose changes.

Formally, histogram and correlogram can be defined in the following way:

Notation. Let I be an image, quantized into m colors $c_1, ..., c_m$. Let $|I|$ denote the size of the image. For a pixel $p = (x, y) \in I$, let $I(p)$ denote its color and $p \in I_c$ be synonymous with $p \in I$, $I(p) = c$. For convenience we use the L_∞ norm to measure the distance between the pixels, i.e. for pixels $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$, we define $|p_1 - p_2| = \max\{|x_1 - x_2|, |y_1 - y_2|\}$. We denote the set $\{1, 2, ..., n\}$ by $[n]$.

Histogram. The color histogram h of image I is defined for $i \in [m]$ such that $h_I(c_i)$ gives for any pixel in I, the probability that the color of the pixel is c_i. Given the count

$$H_I(c_i) \equiv \left| \{ p \in I_{c_i} \} \right| \tag{1}$$

it follows that

$$h_I(c_i) = \frac{H_I(c_i)}{|I|} \tag{2}$$

Correlogram. Let a distance set D be defined *a priori*. Let $d = |D|$. Then the correlogram of I is defined for $i, j \in [m], k \in D$, such that $\gamma_I(c_i, c_j, k)$ gives the conditional probability that a pixel at a given distance k from a given pixel of color c_i, is of color c_j. Therefore, we need to locate the pixels of color c_i in the image, and for each of these, we need to accumulate the pixel counts of all the pixels of color c_j at distance k. The resulting count can be formally written

$$\Gamma_I(c_i, c_j, k) \equiv \left| \{ p_1 \in I_{c_i}, p_2 \in I_{c_j} \,|\, |p_1 - p_2| = k \} \right| \tag{3}$$

Then the correlogram value for color c_j, given that c_j is a color that is at distance k from another pixel of color c_i, is the quantity obtained by normalizing the accumulated pixel count of Eq. (3), so

it is interpreted as a conditional probability; it sums up to 1 when integrated over all possible colors c_j. Its expression is

$$\gamma_I(c_i, c_j, k) = \frac{\Gamma_I(c_i, c_j, k)}{8k\, H_I(c_i)} \tag{4}$$

where the denominator is the total number of pixels at distance k from any pixel of color c_i. The $8k$ factor is the number of pixel neighbors at distance k from a given pixel. Due to properties of the L_∞ norm, at distance 1 a pixel has 8 neighbors, at distance 2 it has 16, and so on. Therefore if we multiply the total number of occurrences of pixels of color c_i by the number of neighbors that each of these pixels has, we obtain the total number of pixels at distance k from any pixel of color c_i.

In the experiments we are using 512 bins both for histograms and correlograms, and for correlograms, we extract statistics for a set of eight distances. Thus the histogram is an array of 512 elements and the correlogram model is composed of eight 512×512 arrays.

4.1 UPDATING THE MODEL

A drawback of appearance based methods is their sensitivity to lighting conditions. Rotation might be a problem too if the color distribution of the object is not uniform within the entire object area. In order to handle these situations, models are updated at every frame. Model updating can be done in two ways. If a stationary distribution is assumed, the model should be updated cumulatively. Otherwise, the model should be updated adaptively. In our case it is more appropriate to consider a non-stationary distribution and adapt the model to the local changes that occur at every frame. Histogram updating is carried out as follows:

$$h_I(c_i, t) = \alpha\, h_I(c_i, t-1) + (1-\alpha)\, h_I^{new}(c_i, t) \tag{5}$$

In other words, the model at time *t* is the weighted sum of the color histogram of the current observation and the histogram of the stored model. The parameter α is a constant between 0 and 1

that determines the rate of the updating process. If α takes values close to 1, the new information will be slowly incorporated. If α takes values close to 0, the old information will be forgotten rapidly. In our system, α was set to 0.9. Similarly, the correlogram is updated by

$$\gamma_I(c_i, c_j, k, t) = \alpha\, \gamma_I(c_i, c_j, k, t-1) + (1-\alpha)\, \gamma_I^{new}(c_i, c_j, k, t) \tag{6}$$

4.2 SIMILARITY BETWEEN MODELS

Once we have a representation for observed regions, we need to find a good distance measure between models that allows us to classify an observation given a set of models. For this purpose we only use correlogram information. If we interpret correlograms as feature vectors whose components are the normalized bin counts described in Eq. (4), then the distance measure between models can be determined by the distance between the corresponding feature vectors. There are many different distance measures that can be used. L_1 and L_2 norms have been commonly used, and depending on the application one or the other performs better. In our system, the normalized L_1 distance is used, since it is simple and statistically more robust to outliers than the L_2 measure. Therefore, given two images I and I', our distance measure is defined as

$$D_\gamma(I, I') \equiv \left| I - I' \right|_{\gamma, d_1} \equiv \frac{\sum_{\forall i,j \in [m], k \in [d]} \left| \gamma_I(c_i, c_j, k) - \gamma_{I'}(c_i, c_j, k) \right|}{2\, m\, d} \tag{7}$$

Note that this measure is symmetric. Also, this measure lies within [0, 1] because of the normalization factor of the denominator. The paramter m is the number of color bins (512 in our experiments) and d the size of the distance set D (8 in our case). Therefore, a similarity measure can be simply derived from $D_\gamma(I, I')$ by taking its complement to 1:

$$S_\gamma(I, I') \equiv 1 - D_\gamma(I, I') \tag{8}$$

4.3 SEGMENTATION UNDER OCCLUSION

Occlusion is one of the most challenging problems when tracking humans. The problem arises because, when a person is partially occluding another person, the background subtraction algorithm will detect them as a single connected component so that it is impossible to know where each person is by just looking at the background subtraction results. The same will happen when a person picks up an object from the scene. For many applications it is important to be able to differentiate each person in the group or localize the object. This is possible if before occlusion we have the models of each entity involved in the occlusion process. In our system we use a variation of the method developed by Huang et al. [14] for subimage localization. We classify each pixel as belonging to one of the possible models.

Formally, the solution can be described in the following way:

Let p be a pixel located in a blob G, and let $G(p)$ denote its color. Let $\Pi_p(G \mid M_m)$ be the likelihood of pixel p belonging to model M_m. Then p will be labeled as belonging to model M_m if and only if

$$m = \arg\max_i \Pi_p \left(G \mid M_i\right) \tag{9}$$

where

$$\Pi_p(G \mid M_i) \equiv \beta \pi_{G_{(p)},h}(G \mid M_i) + (1-\beta)\pi_{p,\gamma}(G \mid M_i) \tag{10}$$

with

$$\pi_{G(p),h}(G \mid M_i) \equiv \min\left\{\frac{\mathrm{H}_{\mathrm{M_i}}(G(p))}{\mathrm{H}_{\mathrm{G}}(G(p))}, 1\right\} \tag{11}$$

$$\pi_{p,\gamma}(G \mid M_i) \equiv 1 - D_\gamma(M_i, \{p\}) \tag{12}$$

Thus, in Eq. (10) two likelihood measures are combined. $\pi_{G(p),h}(G \mid M_i)$ is the histogram backprojection ratio introduced by Swain and Ballard in [22] and $\pi_{p,\gamma}(G \mid M_i)$ is the correlogram correction factor that Huang et al. added in [14]. Note that the histogram backprojection ratio does

not take into account any local information, since it gives the same likelihood to all pixels of the same color in blob G. As explained in [14], the correlogram correction factor adds discriminating power by introducing local spatial correlation information. These two factors are weighted by β and $1-\beta$ respectively, where β is a constant that can take values in the interval $[0,1]$. In our experiments, as in [14], β is set to 0.5. In Eq. (11), $H_{M_i}(G(p))$ is the count corresponding to the histogram of the model M_i. In Eq. (12), $\{p\}$ represents the set composed of pixel p and its neighbors (the neighborhood of p depends on the set of distances where the correlogram is defined). Then $D_\gamma(M_i,\{p\})$ is the distance between the local correlogram centered at pixel p and the part of correlogram for M_i that corresponds to color G_p, that is, $\gamma_{M_i}(G_p,c_j,k)$. Note that $\gamma_{M_i}(G_p,c_j,k)$ is an $m\times d$ array (512×8 in our experiments).

If we use Eq. (9) directly at each pixel location, it can happen that a pixel is classified as belonging to model M_i even though all its neighbors have been classified as belonging to model M_j. Since we expect segmentation to be smooth (i.e. neighbor pixels will usually belong to the same model), at each pixel location p the likelihood measure of Eq. (9) of the neighboring pixels in a specific window W can be added. In this way, in order to classify a pixel, besides its likelihood, we will also take into account the likelihoods of its neighbors. Thus we have:

$$p \in M_m \Leftrightarrow m = argmax_i \sum_{\forall p \in W} \Pi_p(G \mid M_i) \tag{13}$$

In our experiments the window size is set to the maximum distance of the distance set D (25 pixels for our correlogram definition).

5. LABELING OF PEOPLE

5.1 ALGORITHM

The objective of the human labeling module is to label each individual consistently throughout the video sequence. The first problem to be solved when labeling people is detecting them. For the

static camera case, detection is usually accomplished by using some kind of background subtraction method [20, 24]. Although the idea behind background subtraction is simple, the problem is very challenging. Changes in illumination conditions, movements of tree branches waving in the wind, camera noise, shadows or reflections can all affect the detection performance.

We use the background subtraction algorithm described in Section 3 and developed by Kim et al. [20]. After this process, the system expects each isolated person to be segmented into an isolated blob. Thus it assumes that in an indoor environment each detected blob will correspond to a person. For instance, if there is a part of the body that has been segmented into a different blob after the background subtraction stage, and this blob has not been removed during the thresholding process, the system will fail and detect that blob corresponding to a part of the body as a new person. It is also assumed that when a person enters the scene he/she will be isolated. If two or more people enter together and are segmented into the same blob by the background subtraction algorithm, the system will treat them as though they were a single person until they split. At this point the system will begin to keep track of each individual separately, assigning the label of the group to one of the components of the group and new labels for the rest. There would be the possibility for the system to go back in time and segment the people while they were together, but this was not implemented.

As soon as a person enters the scene, a model for that person is generated and stored. As explained in Section 4, a model consists of both histogram and correlogram information, even though the histogram is only used in the segmentation process. In the next frame, another model is built for each of the foreground blobs. Then the similarity measure between all the stored models and all the models of blobs in the current frame is calculated using the definition in Eq. (8). The most similar blobs are matched as long as the similarity measure is above a certain threshold. However, if a blob in the current frame is at a significant distance from all the stored models, a new model is initialized. The threshold (which can take values between 0 and 1) was trained using a training set

of ten sequences and its value was set to 0.55. Once the matching stage is completed, all the models are updated using Eqs. (5) and (6) and the next frame is processed.

Of course, people can form groups that the background subtraction algorithm will segment into a single blob. It is desirable to be able to track these people even when they are in the group or are partially occluded by other people or objects. The system detects that two or more people have merged into a group when the total number of blobs in the frame has decreased and two or more blobs in the previous frame overlap with a single blob in the current frame. When merging is detected, a model for the group is initialized. This model consists of a histogram and a correlogram for the group, but it also contains the models of each of the people that form the group. The correlogram of the group is used to consistently label the group for the rest of the sequence in the same way that the other blobs are labeled. The models of the components of the group are used to segment the group into each of the people that form it, classifying each pixel as belonging to one of the models using the method described in Section 4.3. However, since segmentation is not perfect, during occlusion the individual models of each of the components of the group are not updated. If a person joins an existing group, a new component is added to the group.

Just as several people can merge into a group, the group can also be split. This event is detected when the total number of blobs in the frame has increased and several blobs in the current frame overlap with a group blob in the previous frame. Of course a group can split in many different ways. For instance, a group of four people can split into a group of three people and a single person, or into two groups of two persons. In order to assign labels after a split, each blob involved in the splitting is segmented as if it was still the group with all the components. Assuming that each person can only be present in one of the blobs involved in the splitting process, it is concluded that a person is present in the blob that contains the largest number of pixels labeled with that person's label. For instance, consider a case of a group of four people (A , B , C and D) that has split in two groups, G_1 and G_2, of two people each. The system would segment each of these two blobs as if

they were still the group with four people each. Then, if we consider the case for person A, the system would count the number of pixels that have been labeled with label A in blob G_1 and in blob G_2. If blob G_1 has 100 pixels labeled with label A and blob G_2 has 1000, the system would conclude that person A is present in blob G_2. Once the system has deduced in which group is present each person, it repeats the segmentation of each group but this time taking into account only the individuals that from each group.

5.2 PSEUDOCODE

We can now summarize the description of the system for consistent labeling of humans with the following pseudocode:

- Step 1: Read frames and subtract the background until one or multiple foreground objects are detected.
- Step2: Build an appearance model for each detected blob.
- Setp3: Read the next frame.
    - Subtract the background.
    - If several people have merged into a group.
        - Detect what people/models form the group.
        - Segment the group into its individuals.
        - Build a model for the group.
    - Match group models from previous frame with blobs in current frame.
        - Update group models for each group in the current frame.
    - Detect if a group in the previous frame has split in the current frame.
        - Delete the group model.
        - Detect if the group has split into single individuals or subgroups.
        - Segment possible subgroups and build a new model for each of them.

- Match remaining blobs with the stored individual models.
    - If the distance of one blob in the current frame to all stored models is higher than a threshold, build a new model.
    - Update the other models.
- Step 4: Go to step 3.

5.3 RESULTS

The system has been tested on a test set of ten uncompressed indoor sequences containing various numbers of people (from two to five people) for different camera positions (side view and 45 degrees top view). All free parameters were trained using a 10 sequence training set (different from the test set). In Figure 1, sample frames of a sequence of 1500 frames are shown. The first column shows the input images, the second column shows the images after background subtraction and the third column shows the output images. In the output images, pseudo-colors are used to represent the identity of each person. The color associated to each person is chosen arbitrarily: the system always paints in red the first person to enter the scene, in green the second one, and so on. In the experiment shown in Figure 1, images were quantized into 512 colors and the correlogram was calculated at distances {1,3,5,7,10,15,23,25}. The sequence consists of five persons moving in a room. In the first sample frame in Figure 1a, the system detects a person and associates the red color to his identity. In the next sample frame, another person enters the scene and the system assigns to him the green color. The next two sample frames show a new person entering the scene while the other two have formed a group. A merge has been detected and the system has segmented the group into each of the individuals that form it, using the maximum likelihood pixel classification described in Section 4.3. In the following frames a new person enters the scene and the four people in the room merge into a single group. In frame number 597, a person leaves the group and identities are preserved. Note that during occlusion the appearance models of each

individual have not been updated but still the system is able to identify the person splitting from the group. This person leaves the scene and in the first sample frame in Figure 1b a new individual that has not been seen before enters the scene. The system correctly assigns a new identity to him. Then,

(a) (b)

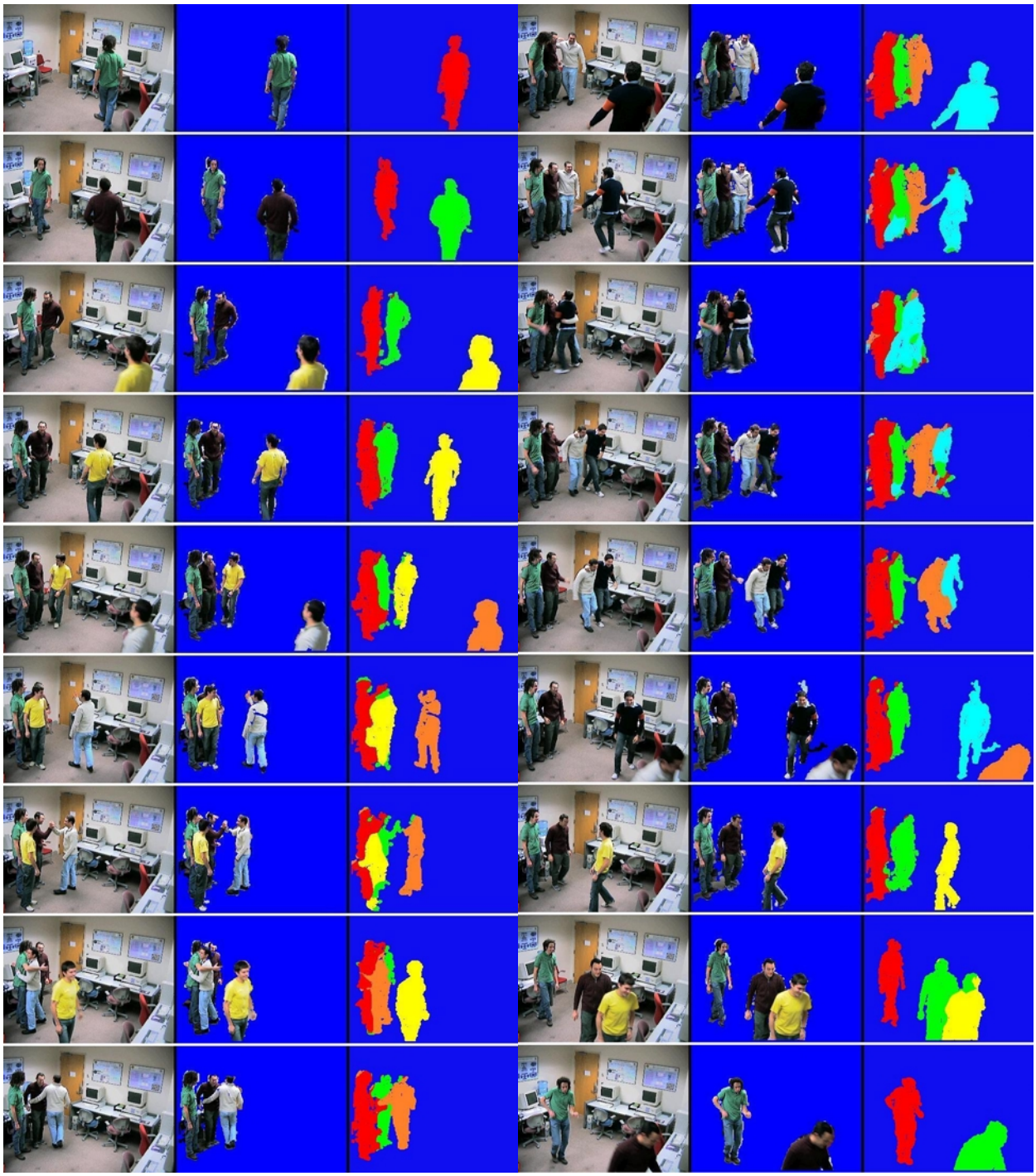

Figure 1: Sequence with 5 people. **a** Sample frames 126, 236, 293, 346, 425, 508, 543, 597 and 681. **b** Sample frames 711, 762, 809, 895, 908, 988, 1165, 1230 and 1265.

again, a group of four humans is formed. This time, however, the group splits in a different way than before, forming two groups of two people each. Again identities are correctly differentiated. In the following frames two individuals leave the scene and in sample frame number 1165 the individual with the yellow label reenters again into the room. The system properly identifies him. Finally, in the last frames all individuals progressively leave the scene. In all sequences from the testing set, the system always correctly detected the total number of people in the scene and always labeled isolated persons consistently. However we did not measure the number of correctly labeled pixels in groups of people.

In order to test the identification capabilities of the system we performed experiments with a dataset of five individuals. Each of them entered the scene twice in a random order and always wearing the same clothing. The system was able to properly identify all individuals. Results are shown in Figure 2. The first frame shows the first person to enter the scene. The system builds a model for him and labels the individual with the red color. Then three more humans enter the scene. The system detects each of them as people that have not been seen before and therefore builds a new model for each person and assigns a different label to each of them. The fifth person to enter, however, is a person that had already been modeled by the system. The system detects this facte and properly associates him as the person who had entered the scene in the third place. The same happens with the next person, who is confirmed as the first person to enter the scene and is labeled with the red color. The following person is a new person and the rest of the individuals to enter the scene are all individuals who were seen before. As shown in Figure 2, all of them are correctly associated by the system.

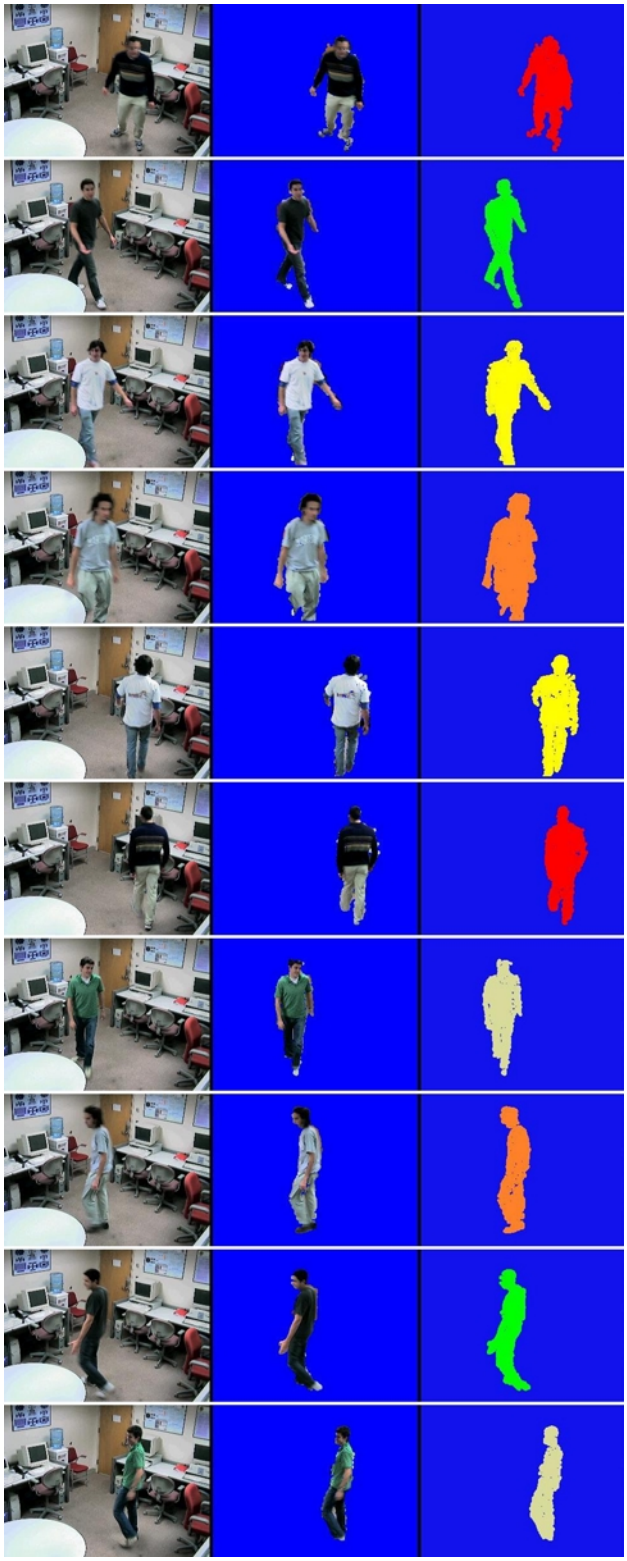

Figure 2: Experiment testing the identification capabilities of the system. Five humans enter the scene twice in a random order and the system properly identifies them.

In another experiment we tested the modeling capabilities of the correlogram in comparison to histogram modeling. Figure 3 shows an increase in performance introduced by the correlogram modeling in a case where segmentation is difficult, since the colors of each person are very similar. The third column in Figure 3 shows the results of the segmentation when the system just uses histogram information. The next column shows the segmentation results when both histogram and correlogram information are used. Although the computational cost of using correlograms is higher than when just histograms are used, the number of correctly labeled pixels is increased and better results are obtained. Note that in the last frame, the group has already split and therefore there is a single label for each individual.

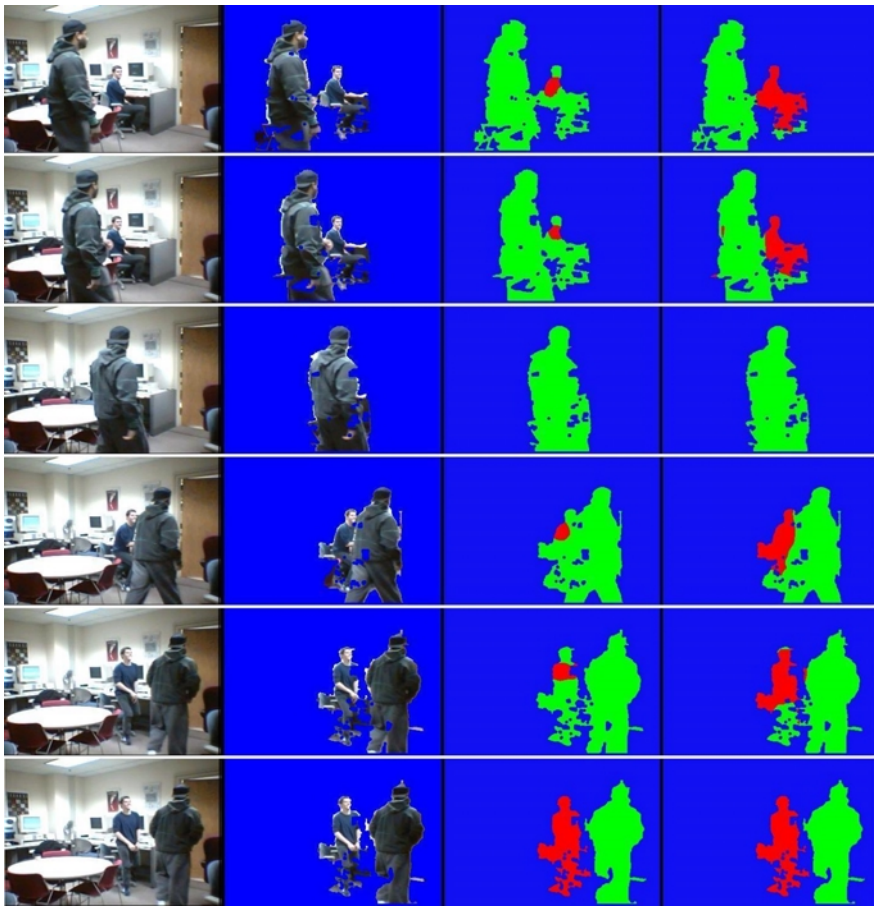

Figure 3: Performance comparison: histogram vs. correlogram. The third column shows the segmentation using only histogram information, and the fourth column shows the results using both histogram and correlogram. Note that the in the last frame the group has already split and therefore there is a single label for each individual.

6. INTERPRETATION OF HUMAN-OBJECT INTERACTIONS

6.1 ALGORITHM

Detecting interactions between humans and objects is of fundamental importance in automatically inferring human activities. A video, for example, could be summarized by showing only the frames where people are interacting with objects. In surveillance applications, it is obviously useful to determine when someone is interacting with objects, to know, for instance, if anybody is stealing anything, or to know if anybody has left a suspicious object.

Figure 4: Background subtraction mask showing a human-object interaction. From the mask it cannot be decided whether the object has been dropped or picked up.

As in the labeling system described in the previous sections, the system for detecting human-object interactions starts with a background subtraction stage. When a person deposits an object in the scene, it will subsequently be detected as a foreground region because the object has not been modeled in the learning stage of the background subtraction algorithm. The same will happen if a person takes an object from the scene; when the object is removed a new foreground region will appear because part of the scene that was behind the object was not modeled,. Therefore, the first objective of the system is to provide a mechanism for detecting such events. The system will detect a human-object interaction (that is, a new object being deposited or an object being been removed) when it detects that a blob splits into two blobs and one of them is static. “Static” means that the centroid of the blob does not move more than two pixels from its initial position during a certain

period of time (in the system this period was set to 10 frames, for a rate of approximately 12 frames per second). In the top two images of Figure 5, an example of an object being removed can be seen. However, from the foreground blobs obtained by background subtraction alone, it cannot be determined whether the object has been removed or deposited in the scene, since in both cases the foreground blobs would look exactly the same. This can be observed in the sequence of three frames in Figure 4: although it may seem that an object has been deposited, actually an object has been picked up in this sequence. Therefore, in order to differentiate the event of taking an object from the event of depositing an object, the images have to be analyzed more carefully. Edges play a key role in this analysis.

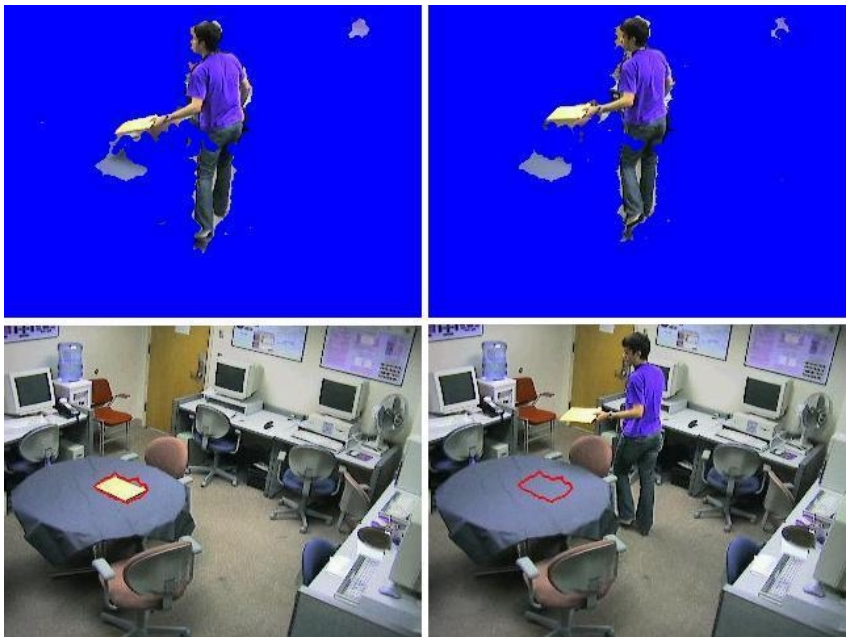

Figure 5: Example of an object being picked up. The top images show the background subtraction mask before and after the object-human interaction has been detected. The bottom images show how the gradient is calculated at object boundaries in the background frame and in the frame where the interaction has been detected.

Heuristically, if edges around the boundaries of the object before the human-object interaction is detected are stronger than after the interaction is detected, this suggests that the object was present before the interaction and it is not present anymore. On the other hand, if edges are stronger after

the interaction than before the interaction, this implies that an object was deposited. More precisely, if the sum of the gradient magnitudes at the boundaries of the foreground blob is greater in the background frame (a previous frame with no foreground blobs present in the region where the static blob has been detected) than in the frame where the interaction was detected, it is concluded that the object was in the background and it is not present anymore. Therefore the object has been picked up (see Figure 5). Otherwise it is decided that the object has been deposited.

Formally, this concept can be expressed in the following way:

Let $R(p)$ be the gradient response at pixel p. Considering the Sobel mask in each direction:

$$\nabla_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \qquad \nabla_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \tag{14}$$

Then,

$$R(p) = \sqrt{\nabla_x^2(p) + \nabla_y^2(p)} \tag{15}$$

Let B be the set of pixels lying at the two pixel wide 8-connected boundary of the object (the static blob), and let $B_i = (x, y)$ be the i^{th} element of this set. Let I_B be a background frame, i.e. a frame captured prior to the frame where the interaction was detected and in which no foreground blobs are present in the region where the static blob has been detected. Let I_D be the frame where the interaction has been detected. Let r_B and r_D be the sums of the gradient magnitudes along the edges of a candidate blob object in the frames I_B and I_D:

$$r_B \equiv \sum_{\forall p \in B} R(I_B(B_p)) \tag{16}$$

and

$$r_D \equiv \sum_{\forall p \in B} R(I_D(B_p)) \tag{17}$$

Then, if $r_B > r_D$, there were edges and they disappeared; therefore the system will decide that the object has been taken. Otherwise, edges were absent and they appeared; therefore the system will decide that the object has been deposited.

After a human-object interaction is detected, if the object has been picked up, both the person and the object will appear connected within the same foreground blob in subsequent frames. In order for the system to segment each of them, the models for the object and the person need to be initialized before the interaction occurred. Therefore the system looks back in time at prior frames until it finds a frame where the object and person blob do not overlap and builds a model for each of them. These models are then used to segment the person and the object in the frames after the pickup event occurred using the segmentation method described in Section 4.3. In the case of an object being dropped, the models are built after the interaction is detected and from that point, the system analyzes individual frames while rewinding the video, i.e. analyzes frames backwards in time, to keep track of the object and the person independently until the moment when the person first entered the scene.

6.2 PSEUDOCODE

We can now summarize the system for interpreting human-object interactions with the following pseudocode:

- Step 1: Read frames and subtract the background until an interaction is detected.
- Step 2: Detect whether the object has been removed or deposited.
    - If the object has been removed:
        - Find a previous frame where the person and the object did not overlap.
        - Build a model for the object and the person.
        - Segment for the rest of the sequence the person from the object.
    - If the object has been deposited:
        - Build a model for the object and the person.

- Rewinding the video sequence, segment the person from the object before the person had dropped the object.

6.3 RESULTS

We performed a large number of experiments with sequences of people dropping and removing objects from the scene. In all of the cases, the algorithm is able to differentiate between an object being dropped and an object being picked up and the segmentation results for the task of distinguishing objects from the persons carrying them are good. In Figure 6, sample frames of a sequence of a person picking up a folder are shown. Between the first frame in Figure 6 and the second one, the system detects that an object is being picked up, builds the models for the person and the object, and starts the segmentation, painting in red the pixels classified as "person" and in green the pixels classified as "object". The segmentation of the folder is very good since its color is very different from the color of the clothes.

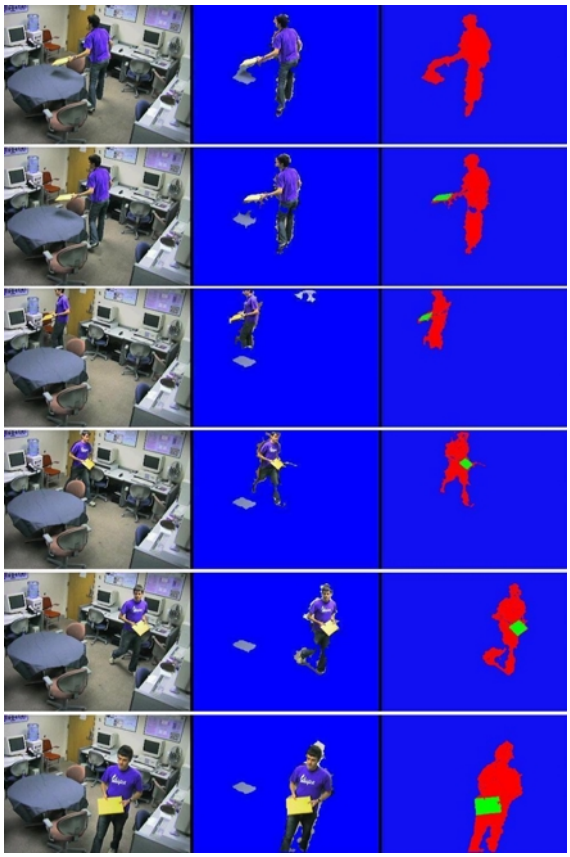

Figure 6: Sequence showing a folder being picked up. Sample frames 213, 216, 266, 343, 385 and 441.

Figure 7 shows results for a sequence of a person leaving a jacket on a chair. When the system detects that the jacket has been dropped, at frame 846, it analyzes the frame in decreasing time order to segment the object and the person in the previous frames. Note that in Figure 7 the frames are shown in the order in which they are analyzed, which is not their natural temporal order. Therefore the frames following the third frame are displayed in backward order. The segmentation of the object within the silhouette of the person is successful, despite some small misclassified areas mainly due to the strong highlighting effects that occur in the hair. Indeed we are able to keep track of the object from the beginning of the sequence to the end. For surveillance applications the accuracy of the boundary detection for carried objects is much less important that the detection of the presence of a carried object.

7. DISCUSSION AND CONCLUSIONS

A system for consistent labeling of human and object has been described. An appearance model using color correlograms is applied for segmenting and keeping track of humans and objects even under occlusion. The system performs well and is able to handle common occluding situations that occur in indoor environments. No assumptions are made about human pose or camera points of view. In the object labeling case, no assumptions are made about the shape of the object. Since the system uses an appearance based model, performance depends on the appearance of objects and humans. If several humans wear similar clothes or the object colors are very similar to the clothes of the person, the performance of the segmentation will degrade. Identification seems to be robust when individuals are isolated. However, the performance could possibly be improved by using Kalman filtering [25] or a particle filtering algorithm [26]. If several people enter together into the scene, the system will not detect them as different individuals until the group splits. As future work the system could at this point go backwards in time to track and segment the group in a similar way the system did in the case of an object being dropped. Adaptive modeling allows the system to

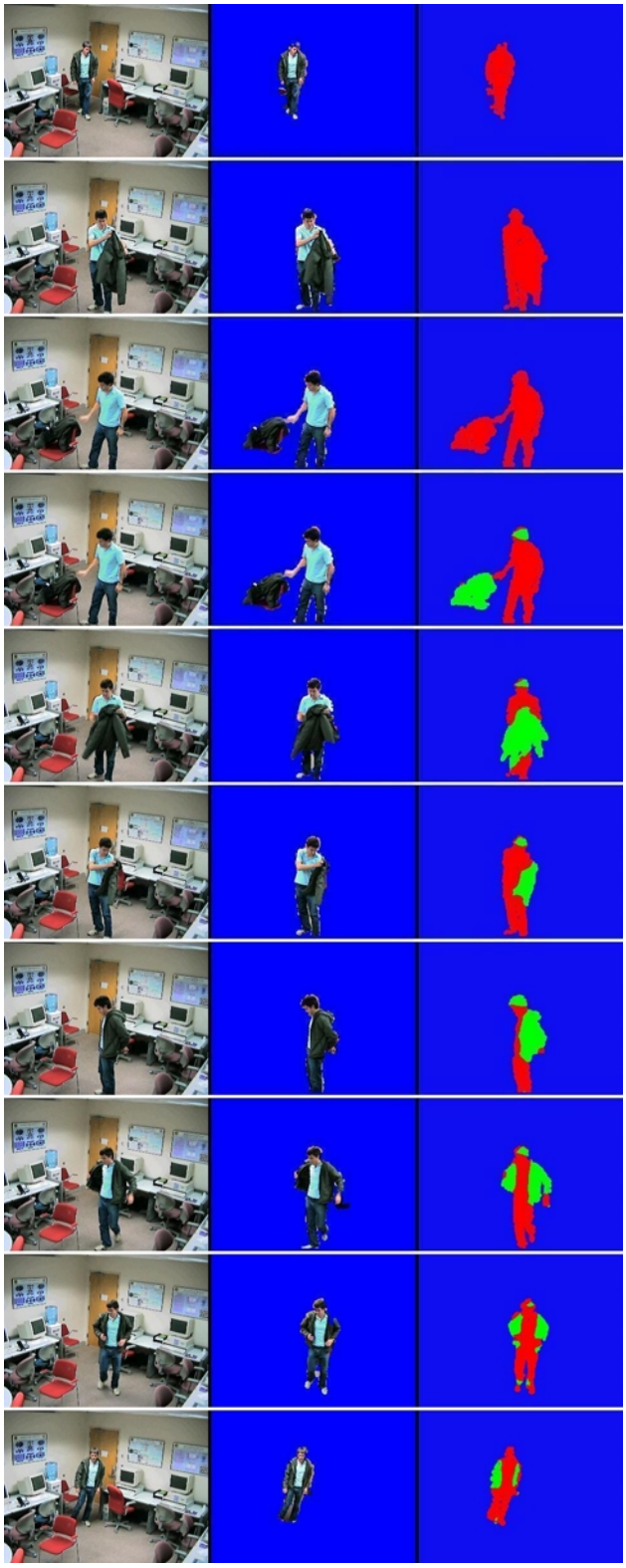

Figure 7: Sequence showing a jacket being deposited on a chair. Sample frames 455, 714, 845, 845, 731, 688, 567, 531, 514 and 472. Object deposit is detected in frame 846, then time order is reversed.

overcome illumination variations. Under occlusion, however, if the models have not been updated for a long period of time, illumination changes may have a more significant effect on segmentation, thus degrading the results. The object labeling module is more sensitive to illumination variations, since the objects are usually relatively small (and therefore have less color information) and the models are initialized using just one frame. In general, segmentation degrades when there are dark colors in shadowed areas or clear colors in highlighted areas. The system has other weak points as well. Although the way for detecting people reentering the scene works well with a small number of individuals, it may not be reliable in all situations and the performance could degrade if the system has to handle a large number of individuals. Therefore, this should be an area of further investigation. Multiple cues (face or gait recognition, etc.) could be combined in order to properly address this problem. Similar issues can be raised with the merging and splitting rule described in Section 5.1. Although these rules performed well in our experiments, one can always think of situations usually improbable, in which these rules would fail. An error in the detection of the foreground blobs would also be critical to the performance of the system. For instance, if a person is divided into multiple foreground blobs, the system would treat each blob as if it belonged to a single individual.

However, overall, the system works fairly well when assumptions made in the designing stage hold. Appearance based modeling using correlograms presents a good combination of flexibility and robustness: it can handle partial occlusion or variations in pose and at the same time is able to produce remarkably good segmentation results under occlusion. The price to pay for adopting correlogram instead of using only histograms is a higher computational load, but as already suggested by Huang et al. [14], dynamic programming techniques can be used to reduce computational complexity.

Applications of the techniques presented in the paper are numerous – from surveillance to video retrieval or video summarization. As an example, we could summarize a video by showing only the

moments where there are several people in the scene or when an interaction with an object has been detected. The system could also be useful in analyzing common paths followed by people in indoor halls or in providing information to surveillance algorithms, passing to these algorithms specific regions where more complex models would be applied. Applications requiring higher level analysis, such as activity recognition, could also take advantage of the information generated by the system.

We have presented a system to consistently label humans and objects in an indoor environment. As an appearance model, we believe that the correlogram is a powerful representation that could be used in many other applications. The segmentation and labeling methods used have also been shown to be effective and in keeping track of people and objects even under partial occlusion. We also have presented methods for detecting human-object interactions and, more importantly, algorithms for differentiating between the actions of picking up and dropping an object. To our knowledge, this problem had not been addressed before and, despite being a simple idea, it is a valuable contribution of our work. Detailed analysis of human-object interactions, once they have been detected by a system such as ours, merits further studies since their interpretation is important for surveillance applications.

8.ORIGINALITY AND CONTRIBUTIONS

Many systems have been designed for tracking humans in videos using an appearance based approach. Previous approaches, however, use models that are not robust to variations in viewpoint or to partial occlusion, and they assume an upright posture. We use color correlograms for representing object and human appearance, for segmenting and labeling objects handled by people, and for keeping track of people even when they merge into groups or separate from groups. Although the correlogram is a well-known technique for image indexing and retrieval, to our knowledge it has not been used for segmenting and labeling humans and objects. The fact that

correlograms model local feature proximity results in stable labeling performance even during changes in view and with partial occlusion.

For detecting human-object interactions, little work has been reported. Haritaoglu et al. [1] were able to segment objects that were carried by people using a simple symmetry constraint on the shape of the people, but the approach requires a full silhouette. McKenna et al. [2] have also addressed the problem of detecting the placement or removal of an object in the scene, but we have gone several steps further: our system is able to differentiate between the two events, to segment the object and consistently label it from frame to frame.

The system facilitates activity recognition and can provide useful information to other systems performing higher-level analysis. It can be used as a first stage to localize people and then fit a more complex model. In a surveillance environment it can be used to trigger alarms when, for example, an object has been removed or deposited, and can be useful for indexing and browsing hours of surveillance video and searching for specific events.

9.ABOUT THE AUTHORS

Martí Balcells received the BS degree from the Technical University of Catalonia (UPC), Barcelona, Spain, in 2001, and the MS degree from University of Maryland in 2002, both in Electrical Engineering. He is currently a researcher at the Prous Institute for Collaborative Biomedical Research, Barcelona, Spain. His research interests include pattern recognition, machine learning, video analysis and data mining.

Daniel DeMenthon is a research scientist with the Laboratory for Language and Media Processing (LAMP) at University of Maryland, College Park, USA. He graduated from Ecole Centrale de Lyon, a French ``Grande Ecole''. After additional graduate degrees in Applied Mathematics from France and Engineering from U.C. Berkeley, he received a PhD in Computer Science from Universite Joseph Fourier in Grenoble, France, in 1993, ith research on a 3D mouse tracked in

space by a video camera. His present research interests include object recognition, augmented reality and video analysis.

David Doermann is co-director of the Laboratory for Language and Media Processing in the University of Maryland's Institute for Advanced Computer Studies and an adjunct member of the Graduate Faculty. He received a B.Sc. degrees in Computer Science and Mathematics from Bloomsburg University in 1987, an M.Sc. degree in 1989 in the Department of Computer Science at the University of Maryland and continued his studies in the Computer Vision Laboratory, where he earned a Ph.D. His research centers widely around the topics of document image analysis and multimedia information processing. Recent interests include mobile applications of document image and video analysis.

Dr. Doermann has served on numerous board and program committees and is an editor of the newly formed International Journal on Document Analysis and Recognition. In addition he the principal investigator on a number of contracts to government agencies and corporations.

10. REFERENCES

[1] I. Haritaoglu, D. Harwood and L. S. Davis. W4: Real-time surveillance of people and their activities. IEEE Transactions on Pattern Analysis and Machine Intelligence 2000, 22(8): 809-830.

[2] S. J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. Tracking groups of people. Computer Vision and Image Understanding 2000, 80(1): 42-56.

[3] A. Mittal and L. S. Davis. M2Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo. In Proceedings of the 7th European conference on computer vision, vol. 1, 2002, pp 18-36.

[4] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for EasyLiving. In Proceedings of the 3rd IEEE International workshop on visual surveillance, 2000, pp 3-10.

[5] C. Wren, A. Azarbayejani, T. Darrel, and A. Pentland. Pfinder: Real-time tracking of the human body. IEEE Transactions on Pattern Analysis and Machine Intelligence 1997, 19(7):780-785.

[6] A. Senior, A. Hampapur, Y. Tian, L. Brown, S. Pankanti and R. Bolle. Appearance models for occlusion handling. In Proceedings of the 2nd IEEE International workshop on PETS, 2001.

[7] C. Stauffer, and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In Proceedings of the IEEE Conference on computer vision and pattern recognition, 1999, pp. 246-252.

[8] L. M. Fuentes, and S. A. Velastin. People tracking in surveillance applications. In Proceedings of the 2nd IEEE International workshop on PETS, 2001.

[9] H. Moon, R. Chellappa, and A. Rosenfeld. 3D object tracking using shape-encoded particle propagation. In Proceedings of the IEEE International Conference on Computer Vision, 2001, pp. 307-314.

[10] A. M. Elgammal, and L. S. Davis. Probabilistic framework for segmenting people under occlusion. In Proceeding of the IEEE 8th International conference on computer vision, vol. 2, 2001, pp. 145-152.

[11] V. Philomin, L. Davis and R. Duraiswami. Tracking humans from a moving platform. In Proceedings of the IEEE 15th International Conference on Pattern Recognition, 2000, pp. 4171-4179.

[12] C. Nakajima, M. Pontil, B. Heisele, T. Poggio. People recognition in image sequences by supervised learning. In Proceedings of IEEE-INNS-ENNS International joint conference on neural networks, 2000.

[13] Y. Raja, S. J. McKenna, and S. Gong. Segmentation and tracking using color mixture models. In Proceedings of the Asian Conference on Computer Vision, vol. 1, 1998, pp. 601-614.

[14] J. Huang, S. R. Kumar, M. Mitra, and W. Zhu. Spatial color indexing and applications. International Journal of Computer Vision 1999, 35(3):245-268.

[15] J. Li, C. S. Chua, and Y. K. Ho. Color Based Multiple People Tracking. In Proceedings of the Seventh International Conference on Control, Automation, Robotics and Vision, 2002, pp. 309-314.

[16] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics 1973, vol. SMC-3(6): 610-621.

[17] A. Rao, R. K. Srihari, Z. Zhang. Geometric histograms: a distribution of geometric configurations of color subsets. In Proceedings of SPIE: Internet imaging, 2000, 3964:91-101.

[18] V. Kovalev and M. Petrou. Multidimensional co-occurrence matrices for object recognition and matching. Graphical Models and Image Processing 1996, 58(3):187-197.

[19] V. Kovalev and S. Volmer. Color co-occurrences descriptors for querying-by-example. In Proceedings of the 5th International conference on multimedia modeling. 1998, pp. 32-38.

[20] K. Kim, T. H. Chalidabhongse, D. Harwood and L. Davis. Background Modeling by Codebook Construction. In Proceedings of the IEEE International Conference on Image Processing, 2004.

[21] T. Kohonen. Learning vector quantization. In Neural Networks 1988, 1:3-16.

[22] M. Swain, and D. Ballard. Color indexing. International Journal of Computer vision 1991, 7(1): 11-32.

[23] G. Pass and R. Zabih. Histogram Refinement for Content-Based Image Retrieval. ACM Journal of Multimedia Systems 1999, 7(3): 234-240.

[24] T. Horprasert, D. Harwood and L. S. Davis. A robust background subtraction and shadow detection. In Proceedings of the Asian Conference on Computer Vision 2000.

[25] R.E. Kalman. A new approach to linear filtering and prediction problems. Transactions of the ASME – Journal of Basic Engineering 1960, 82(Series D):35-45.

[26] M. Isard and A. Blake. Condensation – Conditional Density Propagation for Visual Tracking. International Journal on Computer Vision 1998, 29(1):5-28.