Computer Vision and Image Understanding 114 (2010) 835–846

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Multiple objects tracking in the presence of long-term occlusions

Vasilis Papadourakis [a], Antonis Argyros [a,b,*]

[a] *Institute of Computer Science, FORTH, Heraklion, Crete, Greece*
[b] *Computer Science Department, University of Crete, Greece*

ARTICLE INFO

Article history:
Received 7 May 2009
Accepted 8 February 2010
Available online 21 February 2010

Keywords:
Multiple objects tracking
Occlusion
Object permanence

ABSTRACT

We present a robust object tracking algorithm that handles spatially extended and temporally long object occlusions. The proposed approach is based on the concept of "object permanence" which suggests that a totally occluded object will re-emerge near its occluder. The proposed method does not require prior training to account for differences in the shape, size, color or motion of the objects to be tracked. Instead, the method automatically and dynamically builds appropriate object representations that enable robust and effective tracking and occlusion reasoning. The proposed approach has been evaluated on several image sequences showing either complex object manipulation tasks or human activity in the context of surveillance applications. Experimental results demonstrate that the developed tracker is capable of handling several challenging situations, where the labels of objects are correctly identified and maintained over time, despite the complex interactions among the tracked objects that lead to several layers of occlusions.

1. Introduction

Visual tracking of multiple objects is an important problem with instances appearing in several application domains. Despite the huge amount of excellent research in the field, the effective and robust solution to the problem remains challenging in many realistic scenarios and settings. Part of the difficulty of the problem stems from the fact that even simple object interactions may result in full occlusions that last for quite long time periods. An object may totally disappear behind another object and reappear after considerable time, close to it, at a different location. As an example, consider the situation illustrated in Fig. 1 where a person grasps his keys to place them somewhere else. Once the keys are firmly grasped, they totally disappear behind the hand. When the transfer is complete, the same keys reappear. Reasoning about the activities in this scene requires the capability to associate the same label to the object seen before and after manipulation. Clearly, the problem may become much more complicated, for example in scenarios involving bimanual interaction with several objects that may (or may not) differ in shape, size, appearance, etc. Similar kinds of problems can be encountered in other applications, involving, for example, tracking individual persons in crowded scenes. In this work, we present our approach to solving this kind of tracking problems.

A lot of approaches have already been proposed for object tracking in the presence of occlusions. Huang and Essa [6], provide a very informative overview of existing approaches. According to their categorization, several of the existing methods handle occlusions implicitly. In the work of Khan and Shah [10] for people tracking, a person is segmented into classes of similar color using the Expectation Maximization (EM) algorithm. Then, the maximization of the a posteriori probability of these classes drives frame-to-frame tracking. McKenna [13] and Marques [12] employ appearance models of tracked regions to identify people after the occurrence of occlusions but their approach provides limited support of complex object interactions. In [7], Isard introduces a Bayesian filter for tracking a potentially varying number of objects. A particle filter is used to perform joint inference on both the number of objects present and their configurations. Occlusion handling is achieved by incorporating the number of interacting persons into the observation model and inferring it using a Bayes network. Jepson et al. [8] proposes a framework for learning appearance models to be used for motion-based tracking of natural objects. The appearance model involves a mixture of stable image structure, learned over long time courses, along with two-frame motion information and an outlier process. This model is used in a motion-based tracking algorithm to provide robustness in the presence of outliers, such as those caused by occlusions.

Several other methods have been proposed that treat explicitly the problem of tracking in the presence of occlusions. Rehg [15] describe a framework for local tracking of self-occluding motion, in which one part of an object obstructs the visibility of another. His approach uses a kinematic model to predict occlusions and

[*] Corresponding author. Address: N. Plastira 100, Vassilika Vouton, GR-700-13 Heraklion, Crete, Greece. Fax: +30 2810 391609.
E-mail address: argyros@ics.forth.gr (A. Argyros).

1077-3142/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2010.02.003

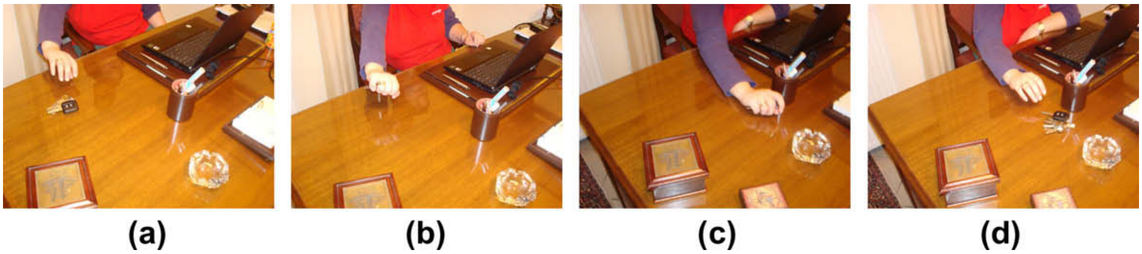

Fig. 1. The need for handling long-term occlusions in the context of tracking. From left to right, a human hand moves towards the keys, grasps and transfers them to a different position. We are interested in a tracking framework which, without a priori information about the tracked objects, will be able to infer that the object disappearing in the second frame, is the same to the one reappearing in the fourth frame.

windowed templates to track partially occluded objects. Brostow [3] present a method to decompose video sequences into layers that represent the relative depths of complex scenes. Activity in a scene is used to extract temporal occlusion events, which are, in turn, used to classify objects on the basis of whether they are occluded by or occlude other objects. Jojic [9] proposes a technique for automatically learning probabilistic 2D appearance maps and masks of moving occluders. The model explains each input image as a layered composition of "flexible sprites". A variational expectation maximization algorithm is employed to learn a mixture of sprites from a video sequence. Tao [18] decomposes video frames into coherent 2D motion layers and introduces a complete dynamic motion layer representation in which spatial and temporal constraints on shape, motion and appearance are estimated using the EM algorithm. His method has been applied in an airborne vehicle tracking system and examples of tracking vehicles in complex interactions are demonstrated. Zhou [22] introduces the concept of background occluding layers and explicitly infer depth ordering of foreground layers. A MAP estimation framework is proposed to simultaneously update the motion layer parameters, the ordering parameters, and the background occluding layers. Wu [19] proposes a dynamic Bayesian network which accommodates an extra hidden process for occlusion. The statistical inference of such a hidden process reveals the occlusion relations among different targets. Yu [20] proposes a framework for treating the general multiple target tracking problem, which is formulated in terms of finding the best spatial and temporal association of observations that maximizes the consistency of both motion and appearance of object trajectories. Leibe [11] considers multi-object tracking as a search for the globally optimal set of space–time trajectories which provides the best explanation for the current image and for all evidence collected so far, while satisfying the constraints that no two objects may occupy the same physical space, nor explain the same image pixels at any point in time. In a recent work, Zhang [21] proposed a network flow based optimization method for data association in multiple objects tracking. The maximum-a-posteriori (MAP) data association problem is mapped into a cost-flow network with a non-overlap constraint on trajectories. The optimal data association is found by a min-cost flow algorithm in the network that is augmented with an explicit occlusion model (EOM) to track long-term occlusions.

The majority of the above methods assume that even partial observations of the occluded objects are possible. As such, they fail to handle total occlusions, especially when they last for considerable amounts of time. The method proposed in this paper is able to handle occlusions that are challenging because of both their spatial extend and duration. The proposed method uses two types of information regarding the scene. The first is the result of scene background subtraction which produces a map showing "where" action takes place in the scene. The second comes from the estimation of several (one per tracked object) Gaussian Mixture Models (GMMs) of color that represent "what" is the appearance of moving objects. The proposed method does not need training to account for the variability in the number of tracked objects, their shape, appearance, or motion characteristics. On the contrary, such information is automatically derived and appropriately updated over time through the use of simple, generic models.

Much of the success of the method depends on a mechanism inspired by the work in [1], that properly associates foreground pixels to different objects. Thus, models of object appearance can be properly maintained and tracked. Occlusion handling is treated through a method founded on the principle of *object permanence* [14,2], which refers to the ability of children to realize that an object exists even when it cannot be seen. Recent studies [2], indicate that infants can reach the object permanence stage at the age of five months, showing the fundamental role of the concept in visual perception.

The proposed algorithm exploits the powerful data association mechanism that has been proposed in Argyros et al. [1], where a method is proposed for tracking multiple skin-colored objects in images acquired by a possibly moving camera. The proposed method encompasses a collection of techniques that enable the detection and modeling of skin-colored objects as well as their temporal association in image sequences. Although not explicitly stated, this tracking algorithm handles occlusions between objects sharing the same color model (skin color). Nevertheless, the method requires prior training to the color model of the objects to be tracked. The approach presented in this paper may handle objects of completely different appearances for which no a priori information is assumed to be known.

In addition to the more complete appearance models, the exploitation of the concept of "object permanence" makes the proposed method much more competent in handling long-term occlusions. Huang et al. [6] also used the concept of "object permanence" to successfully handle long-term occlusions of a varying number of objects over extended image sequences. Their approach incorporates (i) a region-level association process and (ii) a object-level localization process to track objects through long periods of occlusions. Region association is approached as a constrained optimization problem and solved using a genetic algorithm. Objects are localized using adaptive appearance models, spatial distributions and occlusion relationships. The approach in [6] does not explicitly handle interacting objects of similar appearance and is, therefore, expected to fail in tracking them. On the contrary, the proposed method succeeds in treating such cases.

The rest of the paper is organized as follows. Section 2 presents the adopted object representation model. Section 3 describes in detail the proposed tracker and occlusion reasoning. In Section 4, we present results from the application of the proposed methodology in several video sequences that demonstrate important aspects of the performance of the proposed method. Among other things, the method is shown to successfully handle dynamic updating of the object's appearance models, long-term occlusions, layered object occlusions and occlusions among objects of similar appearance. Finally, Section 5 provides the main conclusions of this work as well as extensions that are under investigation.

2. Object modeling

The proposed method is able to detect and track an arbitrary and potentially time varying number of objects. No a priori knowledge regarding the object's 2D or 3D shape, appearance or motion is assumed. To achieve tracking, simple, generic object models are automatically built and maintained.

In the following, we represent an image point as $p = (x, y, c)$ under the convention that it is located at (x, y) and has color c. Each object is represented with a parametric model that takes into account both its spatial layout and its photometric appearance. More specifically, the object model $o \equiv (e, g)$ consists of an ellipse e that accounts for the position and spatial distribution of an object and a Gaussian Mixture Model (GMM) g that represents its color distribution.

The ellipse $e = (c_x, c_y, \alpha, \beta, \theta)$ represents the spatial extend of an object o that is located at (c_x, c_y), has an orientation θ with respect to a local 2D image coordinate frame, and the lengths of its major and minor axes are α and β, respectively. Given a set of image points $\mathcal{P}(o)$ comprising the image of an object, the parameters of e can be computed from the covariance matrix of the locations of pixels in $\mathcal{P}(o)$.

We define the spatial distance $D(p, e)$ of an image point p from ellipse e as in [1]. Intuitively, the ellipse is transformed to a circle of radius equal to one and the same affine transformation is applied to the coordinates of the point p. The distance $D(p, e)$ of p from e is the Euclidean distance of p from the center of ellipse e in this normalized frame. The set $I(e)$ of points p that are interior to the ellipse e can be defined based on $D(p, e)$:

$$I(e) = \{p | D(p, e) \leqslant 1\}. \tag{1}$$

The appearance g of an object o is modeled as a Gaussian Mixture Model (GMM) $g = g(w_k, \mu_k, \Sigma_k), \ 1 \leqslant k \leqslant K$, representing the color (UV components of YUV color space) distribution of the object's pixels. Each of the K triplets (w_k, μ_k, Σ_k) represents the weight, the mean and the covariance matrix of the kth Gaussian component of the mixture. The Expectation Maximization algorithm [4] is employed to determine the parameters of the GMM g for each object o based on the set $\mathcal{P}(o)$ of points that comprise it. We also define the probability that the pixel's color c was drawn from a GMM g as

$$P_A(p, g) = \sum_{k=1}^{K} w_k P(c | \mu_k, \Sigma_k). \tag{2}$$

$P_A(p, g)$ is a measure of the compatibility of p's color with g.

3. Proposed method

Fig. 2 illustrates the information flow of the proposed tracking algorithm. Each frame of the input image sequence is first background subtracted [23] to detect foreground pixels and to form distinct blobs, i.e. regions of connected foreground pixels. Assuming a still camera, background subtraction gives rise to a change mask that can be attributed to the moving objects. A set of objects that must be correctly associated to the pixels of the detected foreground blobs is also maintained. Clearly, even in the simple case of partial occlusions, there is no one-to-one mapping between objects and blobs. Therefore, the goal of the proposed method is to exploit spatial and photometric object information in order to (a) associate foreground blob pixels with objects, (b) investigate occlusion relationships between objects, (c) update the object models and (d) use all extracted information to enable tracking. The rest of this section provides further details on these algorithmic steps.

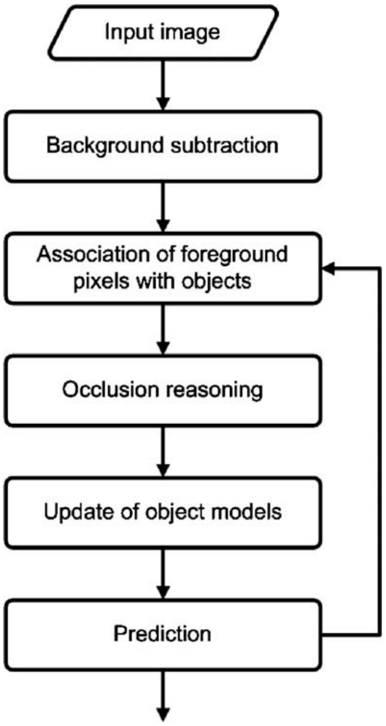

Fig. 2. The flow diagram of the proposed method for tracking multiple objects in the presence of long-term occlusions.

3.1. Associating foreground blob pixels with objects

The aim of this part of the proposed method is to define the set $\mathcal{P}(o)$ of pixels belonging to an object o. It is assumed that at a given moment in time, M foreground blobs $b_j, 1 \leqslant j \leqslant M$ have been detected and that N objects $o_i, 1 \leqslant i \leqslant N$ are already being tracked. A single connected object can give rise to at most one connected blob.[1] However, due to occlusions, two or more different objects may appear as a single connected blob. Thus, it holds that $M \leqslant N$. As a direct consequence, each blob may correspond to one or more objects.

To resolve the data association problem, the method takes into account both the spatial proximity and the appearance similarity between a blob and an object. This is performed in two steps. In the first step, an object is associated with a certain blob. The validity of this algorithmic step stems from the reasonable assumption that a single connected object can give rise to at most one connected blob. In the second step, each object takes its share from the pixels of the blob it is associated with. Fig. 3, graphically illustrates four objects ($o_1, \ o_2, \ o_3$ and o_4, visually represented as the associated ellipses) and three blobs ($b_1, \ b_2$ and b_3, shown as colored image regions).

3.1.1. Associating objects with blobs

For an object $o_i = (e_i, g_i)$ and a blob b_j, the degree $C(b_j, o_i)$ of their association is defined as:

[1] The implicit assumption at this point is that change detection through background subtraction cannot give rise to multiple blobs for a single object. This is safeguarded through morphological filtering applied to the result of background subtraction.

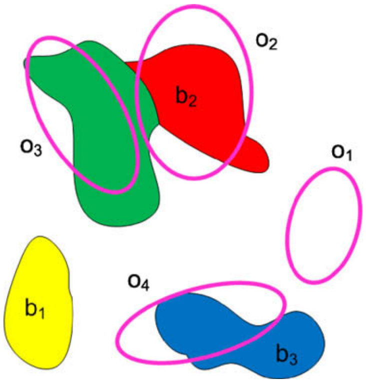

Fig. 3. Possible relations between objects and blobs. For illustration purposes, each object hypothesis is shown as an ellipse and each blob as a monochrome or bi-color image region.

$$C(b_j, o_i) = \sum_{p \in (b_j \cap I(e_i))} P_A(p, g_i). \tag{3}$$

Intuitively, all image points in the intersection of blob b_j with object's ellipse e_i are tested for compatibility with the object's appearance model. Each object is associated with the blob that gives rise to the highest degree of association. More specifically, the blob $\mathscr{B}(o)$ with which object o is associated is defined as:

$$\mathscr{B}(o) = \arg\max_{b_j} C(b_j, o). \tag{4}$$

Thus, an object is associated with only one blob, whereas a blob may be associated with many objects.

3.1.2. Identifying object support regions

After associating objects with blobs, four interesting cases may arise: (a) a blob might be associated with no object, (b) an object might be associated with no blob, (c) an object might be associated with exactly one blob, and (d) a blob might be associated to multiple objects. In the following, we investigate these cases in more detail.

Blobs not associated to objects: Consider a blob b such that

$$\forall o_i, b \cap I(e_i) = \emptyset \Rightarrow \forall o_i, C(b, o_i) = 0. \tag{5}$$

Eq. (5) implies that none of the existing object hypotheses explains the existence of this blob. Thus, this has to be a new object, an object that has just appeared in the scene for the first time. In the example of Fig. 3, b_1 is such a blob. In this case, a new object is generated and its set $\mathscr{P}(o)$ becomes equal to b.

Objects not associated to blobs: Consider the case of an object o such that

$$\left(\cup_{j=1}^{M} b_j\right) \cap I(e) = \emptyset \Rightarrow \forall b_j, C(b_j, o) = 0. \tag{6}$$

In this case, the hypothesis for an object o is not supported by any foreground blob pixel observations. Thus, o has disappeared and must be removed from further consideration. In the example of Fig. 3, object o_1 satisfies the criterion of Eq. (6).

Blobs in one-to-one correspondence with objects: In case that a single object o is associated to a single blob b, the set $\mathscr{P}(o)$ becomes equal to b. This is the case with object o_4 and blob b_3 in Fig. 3.

Blobs associated to multiple objects: As discussed earlier, the correspondence between blobs and objects is not necessarily one-to-one. Two objects in an occlusion relationship will give rise to a single image blob. Consider, for example, the relevant situation in Fig. 3. Objects o_2 and o_3 must "compete" for the pixels of blob b_2. Having already associated an object o with the blob $\mathscr{B}(o)$ (see Eq. (4)), we search for the set $\mathscr{P}(o)$ of pixels to be associated with object o only within blob $\mathscr{B}(o)$. Equivalently, each pixel p of such a blob is associated with the object o^* defined as:

$$o^* = \arg\max_{o} \frac{P_A(p, g)}{D(p, e)}. \tag{7}$$

Intuitively, Eq. (7) assigns blob pixels p to the object o^* that minimizes spatial distance and maximizes appearance compatibility.

The approach described so far assigns image points to objects with distinct appearance models. Still, in several tracking tasks, interacting objects of similar appearance are frequently encountered. For such cases, an approach similar in spirit to that of [1] has been adopted. As a first step, it is required to quantitatively characterize the appearance similarity of two objects. Having represented an object's appearance with a GMM, this boils down to employing a criterion measuring the similarity between two GMMs. For this purpose, a Bhattacharyya-based distance has been employed. More specifically, the distance $\Delta(g, g')$ between two mixtures of Gaussians g and g', is given by [16]:

$$\Delta(g, g') = \sum_{i=1}^{K} \sum_{j=1}^{K} w_i w'_j B(g_i, g'_j). \tag{8}$$

In Eq. (8), both g and g' are composed of K kernels, g_i and g'_j denote the corresponding kernel parameters and w_i, w'_j the mixing weights. $B(\cdot,\cdot)$ denotes the Bhattacharyya distance between two Gaussian kernels, defined as [5]:

$$B(g, g') = \frac{1}{8}(\mu - \mu')^T \frac{(\Sigma + \Sigma')}{2}^{-1} (\mu - \mu') + \frac{1}{2} \ln \frac{\left|\frac{\Sigma + \Sigma'}{2}\right|}{\sqrt{|\Sigma||\Sigma'|}}. \tag{9}$$

Let a number of objects of the same appearance compete for the pixels of the same blob. Having already associated an object o to the blob $\mathscr{B}(o)$ (see Eq. (4)), pixels $\mathscr{P}(o)$ defining object o will be searched only within blob $\mathscr{B}(o)$. Let also $\mathscr{P}(o)$ be initialized to the empty set, that is, $\forall o \mathscr{P}(o) = \emptyset$. Then, the rules governing the assignment of blob points to objects are the following:

- If for a pixel $p \in \mathscr{B}(o)$ of the blob associated to an object $o = (e, g)$ it holds that $p \in I(e)$, then $\mathscr{P}(o) := \mathscr{P}(o) \cup \{p\}$. Note that this way, p may be assigned to several different objects having the same appearance models.
- If a pixel $p \in \mathscr{B}(o)$ does not belong to any of the ellipses of the competing object models, then $\mathscr{P}(o^*) := \mathscr{P}(o^*) \cup \{p\}$ where o^* is the object defined as:

$$o^* = \arg\min_{o} D(p, e). \tag{10}$$

Intuitively, Eq. (10) assigns blob pixels p outside any object ellipse, to an object that minimizes the spatial distance to it.

3.2. Object models update

Once each and every blob pixel has been assigned to some object, point sets $\mathscr{P}(o_i)$ have been computed for all objects o_i. Then, an update of the objects $o_i = (e_i, g_i)$ can be performed based on the sets $\mathscr{P}(o_i)$. As stated earlier, e_i can be computed from the spatial distribution of points in $\mathscr{P}(o_i)$. Additionally, each object's area is defined as $A_i = |\mathscr{P}(o_i)|$. The appearance model g_i is computed through the application of Expectation Maximization algorithm [4] over the colors of the image points in $\mathscr{P}(o_i)$.

The appearance model of an object is updated only for objects that are in one-to-one correspondence with an image blob. In fact, and as it will become more clear in Section 3.3, this is equivalent to updating an object's appearance model when it is observed in isolation, without any occlusions occurring. Having two objects competing for the pixels of a single blob signals occlusion. In that case, the appearance models of the corresponding objects are stopped from being updated. We also denote with A'_i the area of object o_i at the last frame in which this object appeared in isolation.

3.3. Object visibility and occlusion handling

Occlusion reasoning is based on both the spatial and the appearance components on an object's model. As an example, consider the situation graphically illustrated in Fig. 4. Fig. 4a shows two objects (a hand and a pincer) prior to occlusion.

At this time, each of the objects is associated with its own blob. As long as occlusion occurs (Fig. 4b–e), the two objects compete for the points of a single blob. The blob pixels that are compatible to the appearance of the occluder (hand) will be assigned to it, so no significant changes in its area will be observed. The occluded object (pincer), will appear to shrink, since fewer and fewer image points will be assigned to it (Fig. 4b and c). Thus, for the occluded object a significant decrease of its area will be observed as soon as occlusion starts. Therefore, the occlusion ratio R_i of an object o_i is defined as:

$$R_i = \frac{A_i}{A'_i}. \tag{11}$$

The occlusion ratio R_i is measured for objects o_i sharing the points of a blob with other objects. A small R_i indicates that its currently observed size is small compared to the area of the same object before occlusion started. Thus, R_i can be used to quantitatively characterize a certain occlusion. In fact, in case that

$$R_i \leqslant T, \tag{12}$$

object o_i is declared as disappeared because of a full occlusion (e.g., Fig. 4c).

Occlusion reasoning does not only require understanding whether an object is occluded or not but also requires the identification of the occluder. In case that only two objects compete for the points of a blob, the situation is straightforward. In case that more than two objects compete for the pixels of the same blob, the definition of the occluder needs more attention. The occluder should be an object that lies in the close proximity of the occluded object o_B and has recently occupied a portion of the occluded object's image. Formally, for each possible occluder o_i, the number of pixels p in $\mathscr{P}(o_i) \cap I(e_B)$ is calculated. The object o_i that produces the largest such number of pixels is defined as the object occluding o_B.

Objects reported as fully occluded according to the definition of Eq. (12) are treated as suggested by the object permanence principle. This means that, until the object appears again (i.e., $R_i > T$), it is assumed to be behind its occluder and to move with it. The object is excluded from the association of objects to blobs (Section 3.1.1). Instead, it inherits the associations of its occluder. In the pixel assignment part (Section 3.1.2), the occluded object is assumed to share the same ellipse with its occluder. This allows the occluded object to continuously claim pixels that are compatible to its color appearance model and lie in the proximity of its occluder.

When a previously occluded object reappears ($R_i > T$) in the proximity of its occluder, the two objects are dis-associated and the image points assigned to the occluded object are used to construct a new spatial model. As the object emerges (see, for example, Fig. 4e), the spatial model grows smoothly through frames and accurately encapsulates the object's shape. As discussed earlier, the appearance model of the object will be updated only when the occluded object appears isolated (Fig. 4f), that is, in a one-to-one correspondence with a blob.

3.4. Layered occlusions

The term layered occlusions is used to describe situations where multiple objects participate in an occlusion relationship. The proposed method forms and maintains dependencies between occluded objects and their occluders. For a set of objects in a layered occlusions relation, there will always be the foremost occluder and a number of occluded ones behind it. All occluded objects declare all other objects as potential occluders. The reappearance of one of these has the following implications:

- The remaining occluded objects will be searched not only in the proximity of the original occluder, but also in the proximity of the newly reappeared object.
- The label of the reappeared object will be removed from the list of all of its potential occluders.

As an example, consider object X which occludes object Y. Let object Z be occluded by the constellation of X and Y. Then, if Y appears, Z has to be searched around both X and Y. Simultaneously, Y should stop from being searched around X. This could lead to a fast grow of the number of alternative hypotheses that need to be monitored and maintained. On the other hand, for all practical pur-

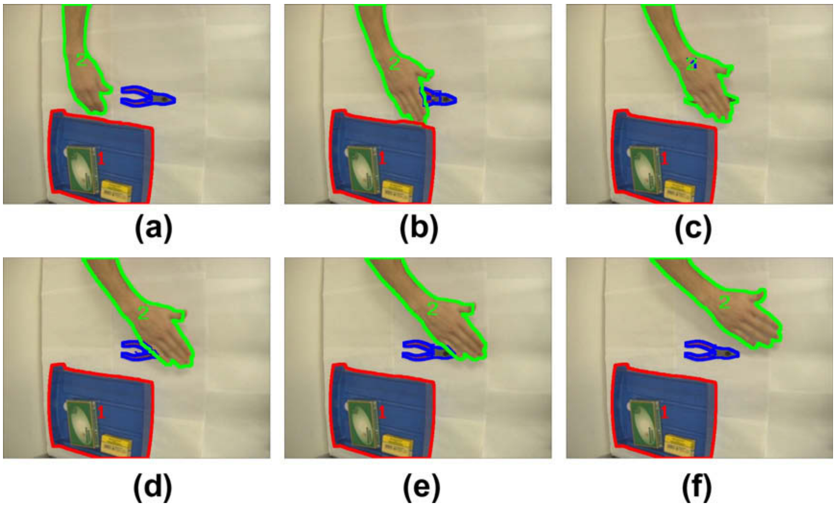

Fig. 4. The size of an object being occluded decreases considerably as occlusion progresses.

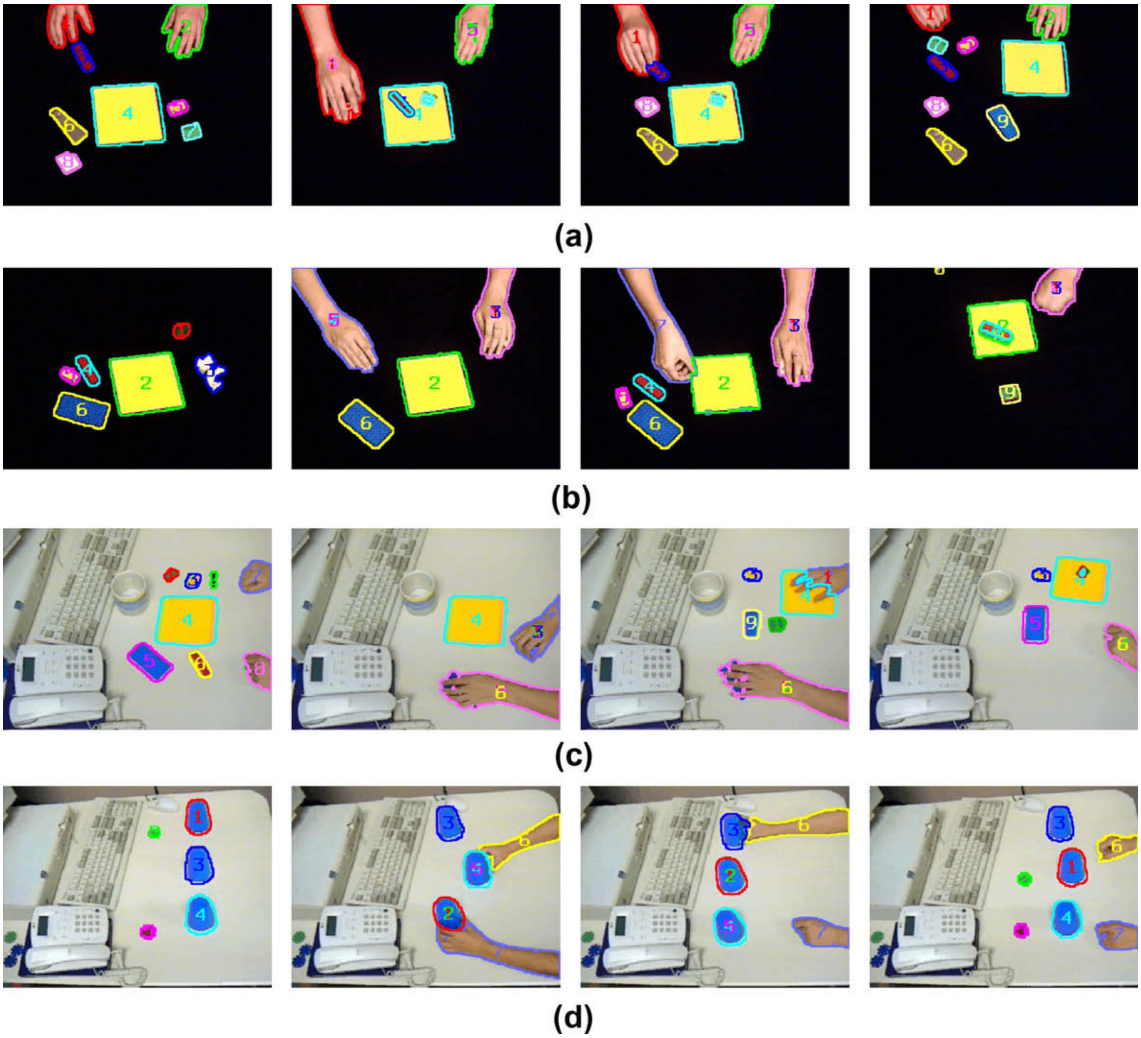

Fig. 5. Characteristic snapshots from the tracking experiments with the sequences tested in [6]. Rows (a)–(d) correspond to datasets “lego 1”, “lego 2”, “lego 3” and “shellgame”, respectively.

poses, the adopted convention performs well in realistic depths of layered occlusions and number of objects involved.

3.5. Linear prediction and object model propagation

In the process described so far, data association is based on the relations of an object’s spatial distribution (represented as the ellipse e in the object’s model $o = (e, g)$) with the detected blobs. However, instead of using the ellipses as those were computed in the previous frame, we may use a prediction about the position of an object’s ellipse based on its recent motion. Assuming that the immediate past is a good prediction for the immediate future, a simple linear scheme is used to predict the object’s ellipse position in the current frame. Blob and blob pixel associations with objects (described in Sections 3.1.1 and 3.1.2, respectively) is then based on these predicted ellipse positions.

4. Experimental results

The proposed method has been tested and evaluated in a series of image sequences demonstrating challenging tracking scenarios. Results from several representative input video sequences are presented in this paper. Videos demonstrating tracking results are available online.[2] In all experiments, input sequences are composed of images of VGA resolution (480×640).

The first set of experiments has been carried out to assess the performance of the proposed method in the image sequences[3] employed in [6]. The four sequences (“lego 1”, “lego 2”, “lego 3” and “shellgame”) consist of 309, 398, 412 and 460 frames, respectively. No background model for these sequences is available. Sequences “lego 1” and “lego 2” are pre-segmented, i.e., foreground colors appear on a black background. For the rest two sequences, a background model is built based on frames in which no foreground object appears. Each row of images in Fig. 5, provides characteristic frames from object tracking in each of these sequences. Individual objects are identified through the use of different colors for their contours and through arithmetic labels located on object centroids. Thus, an object is successfully tracked if it maintains the same color and label in all of its occurrences. Overall, the proposed method managed to successfully track all objects in all of these videos.

To evaluate the proposed method in even more challenging situations, several other videos have been recorded and used for test-

[2] Supplementary video material showing tracking results can be retrieved at the following web address: http://www.ics.forth.gr/~argyros/research/occlusions.html.

[3] Available at http://www.cc.gatech.edu/cpl/projects/occlusion/.

ing. In all reported experiments, input sequences consisted of standard VGA resolution images (640×480) acquired at 20 Hz. Background subtraction has been performed with Zivkovic's improvements [23] of the Stauffer and Grimson's method [17]. The U, V components of the YUV color space has been used for building the GMMs g of object appearances. For each GMM, the EM algorithm had to estimate the parameters of $K = 10$ components. The threshold T on the occlusion ratio (Eq. (12)) signaling full object occlusion was set to 35%. The selection of this threshold value is related to the robust handling of the re-emergence and the subsequent tracking of previously occluded objects. Setting the threshold value T close to 0%, would mean that a minor color misclassification would suffice to falsely signal the reappearance of an occluded object. Additionally, the detection of very small visible parts of partially occluded objects necessitates their subsequent tracking. This can be error-prone if these parts are very small.

The first such image sequence ("objects" sequence) consists of 1280 frames and shows a person manipulating several objects on a tabletop. Characteristic snapshots demonstrating tracking results are shown in Fig. 6. The sequence scenario is as follows. Initially, a hand brings into the scene a basket containing several objects. Then, he empties the basket, interacts with the objects, fills the basket again and finally empties it once more. At the beginning of the experiment, the system has no a priori knowledge about the type, size, color, shape or motion of the objects to be observed. At the end of the experiment the proposed method has been able to track individual objects and has built a model of their color appearance.

More specifically, Fig. 6a shows the empty desktop on which the experiment is performed and of which a background model has been built. In Fig. 6b, the human hand has already brought into the scene a box containing a few objects. Having no a priori knowl-

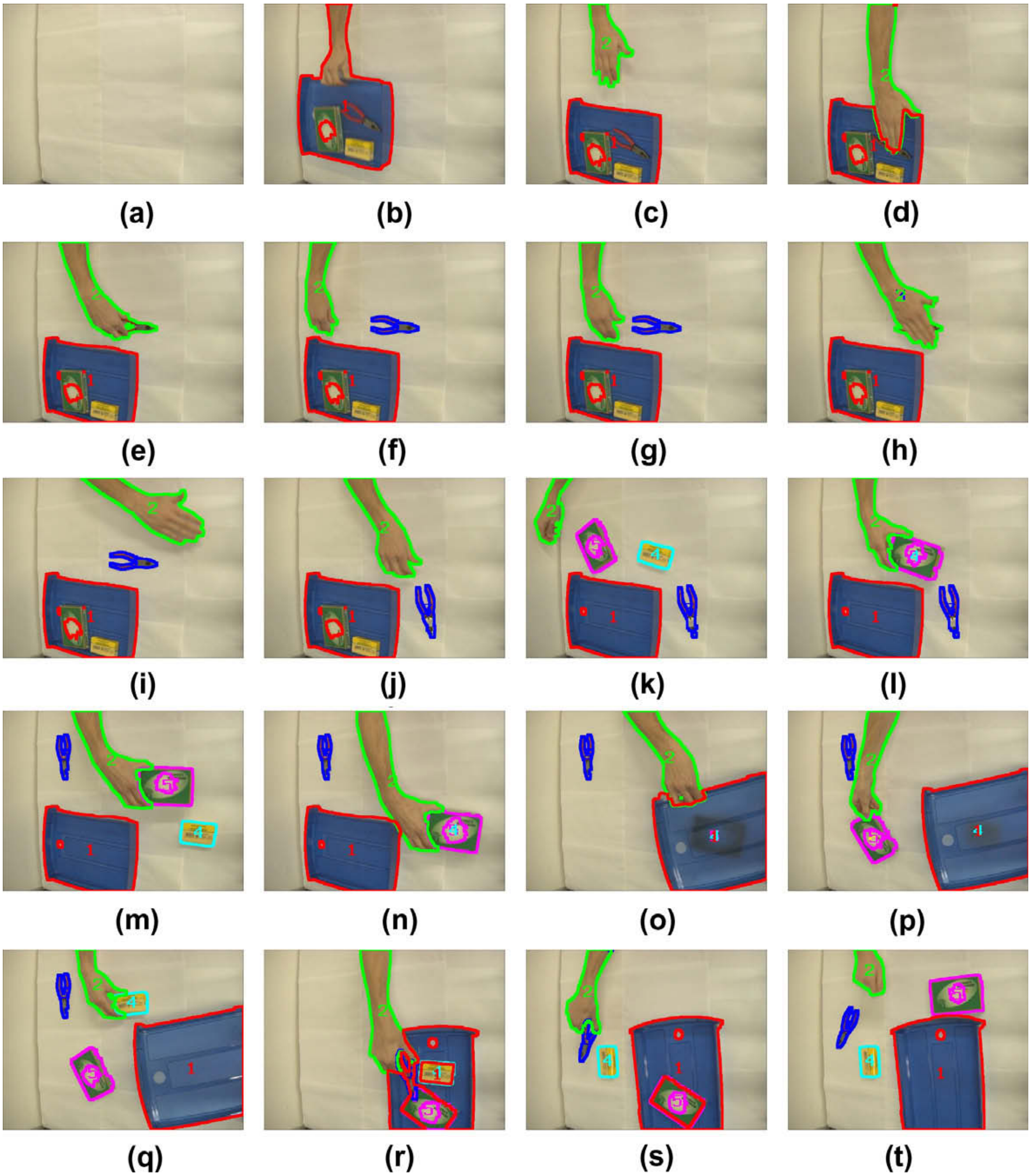

Fig. 6. Characteristic snapshots from the tracking experiment on the "objects" image sequence.

edge about the scene other than a background model of it, the system identifies the constellation of the hand, the blue box and the rest of the objects as a single multicolor object, for which it builds a single object model. As soon as the hand leaves the box on the table (Fig. 6c), the originally connected set of pixels becomes disconnected. The original object hypothesis (red contour) is assigned to the blue box because this is more similar to the previous box/hand constellation. Another object (hand, green contour) is automatically generated. For the next frames, the hand color appearance model is updated. The same happens also to the appearance model of the blue box, in which the components corresponding to the previously joined hand, now vanish. The hand interacts with the box again (Fig. 6d). Now, the color models built assist the method in correctly assigning the pixels of the single connected blob to the two object hypotheses (hand, box). In Fig. 6e, the hand has taken the pincer off the blue box and moves it to another position on the table. For the moment, the method interprets this as a change in the appearance of the hand and, at that stage, the pincer appears as part of the hand object. This is because the pincer has never been observed in isolation but only as part of another object (box). As soon as the hand leaves the pincer on the table, the pincer is understood as an individual object (Fig. 6f, blue contour). The identity of the pincer object is not lost even when the hand passes several times over it, grasps it and moves it to another place on the table (Fig. 6g–j). In a similar manner, the hand empties the basket. As shown in Fig. 6k, the hand, the box and the pincer maintain their original identity, while the two other objects have acquired their own object identities. In Fig. 6l, the hand has grasped the object with the purple contour and has used it to completely occlude the one with the cyan contour. The full occlusion has been signaled and both object hypotheses are maintained and tracked together with the observed region of the occluder. Both objects are transfered to a new position, the hand removes the occluding object (Fig. 6m) and the correct identity for the occluded object is still maintained. The purple object is again brought on top of the cyan one, fully occluding it once more. This time, the big box is also brought on top of the purple object creating a layered occlusion (Fig. 6o). When the hand brings the purple object again in sight dragging it under the big box, the purple object still maintains its original identity (Fig. 6p). The same happens to the cyan object (Fig. 6q). The manipulation of objects continues; the hand brings all objects again into the blue basket and starts moving the latter around (Fig. 6r). The experiment ends with the hand emptying the basket once more (Fig. 6s and t). Correct object identities are still maintained.

Another experiment was performed on the ‘‘lemons” sequence (350 frames in total, presented in Fig. 7), demonstrating that the method succeeds in handling occlusions when tracking objects of similar appearance. In a scene setting that is similar to the previous one, two hands appear in front of a camera (Fig. 7a) and are assigned two different object identities. The hand appearing at the left (green contour) holds two lemons. As soon as lemons appear in isolation (Fig. 7b and c) they get their own object labels. Then, each hand grasps a lemon (Fig. 7d), fully occludes it (Fig. 7e) and then reveals it (Fig. 7f). Lemon identities have been maintained. The two hands grasp the two lemons totally occluding them and then cross (Fig. 7g). Hands reveal what they carry (Fig. 7h and i), showing that despite the complex interaction of two objects of similar color appearance (arms) with two other objects of similar color appearance (lemons) and the simultaneous presence of two full occlusions, the identities of the lemons are correctly tracked. The experiment ends after the hands leave the objects they hold on table (Fig. 7j–l).

In all the experiments reported so far, a person manipulates certain objects in front of a visually simple background. Although that background model building and maintenance is not the main focus of this paper, it is interesting to verify the performance of the proposed approach in cases where background modeling and foreground detection is performed in more realistic conditions. Towards this goal, two image sequences have been recorded in a

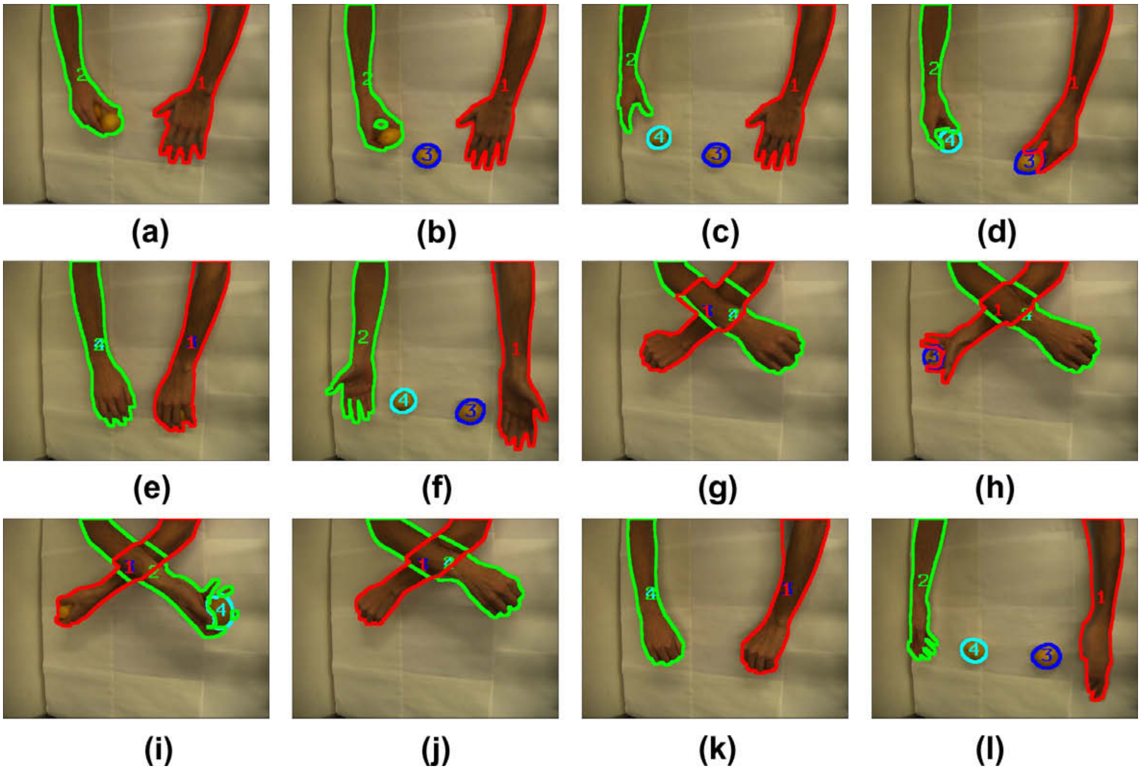
(a) (b) (c) (d)
(e) (f) (g) (h)
(i) (j) (k) (l)

Fig. 7. Characteristic snapshots from the tracking experiment on the ‘‘lemons” image sequence.

room that is monitored by several cameras. The proposed tracking approach has been employed to monitor the activity of humans interacting in this room.

Fig. 8 shows characteristic snapshots of the first such sequence ("bag" sequence), which consists of 287 frames. Fig. 8a shows the appearance of the space where the experiments have been conducted; the background model has been built for this environment appearance. Fig. 8b shows the first person that has been detected and tracked (red contour, person 1). This human is successfully tracked as he approaches the camera (Fig. 8c) and rotates around his vertical axis (Fig. 8d). In the meantime (Fig. 8c) a new person (green contour, person 2) enters the room holding a bag. The bag is identified as a new object as soon as the person holding it leaves it on the floor (Fig. 8e). After this, both humans move around the bag occluding it as well as occluding each other (Fig. 8f–g). At a certain point (Fig. 8h), person 2 has grasped again the bag and hands it to person 1. Despite the complex humans/object interaction, all objects maintain correct identities. In Fig. 8i, person 1 holds the bag, although totally occluding it. Both persons continue to move occluding each other (Fig. 8j) until the one holding the bag leaves it again on the floor (Fig. 8k). Finally, both persons are successfully tracked until they exit the room at Fig. 8m and p, respectively. Overall, the proposed tracking method was able to detect and track correctly all the individual objects moving in the scene. It should be stressed that this has been achieved without any kind of a priori known object models. The achievement of object tracking in such complex situations is difficult even if somebody takes into account additional context dependent knowledge such as the fact that there are persons walking on a ground floor, etc. Clearly, accommodating such important additional cues and knowledge can only improve tracking.

In another experiment ("bucket" sequence, 398 frames), an even more challenging situation is encountered. This image sequence involves two persons that interact with two almost identical looking objects. Snapshots from this sequence are provided in Fig. 9. In Fig. 9a, a person enters a room holding, in each of his hands, a red bucket. The person leaves the two buckets on the floor (Fig. 9b and c) and starts moving around them (Fig. 9d and e). Then, he stands in front of one of the buckets occluding it, grasping it with his right hand and then passing it to his left hand behind his back (Fig. 9f and g). He then grasps the second bucket with his right hand and then starts hiding each of the buckets from the camera (Fig. 9i–k). At some point in time, he leaves both objects on the floor again (Fig. 9l). Right after, another person appears (Fig. 9m). While various types of occlusions continue to occur, each person grasps a bucket (Fig. 9n) and start moving around in the room. Between frames corresponding to Fig. 9p and r, the two persons move around each other holding the buckets, thus creating several layered occlusions. Finally, the two persons leave the buckets again on the floor and exit the room (Fig. 9s and t). Throughout

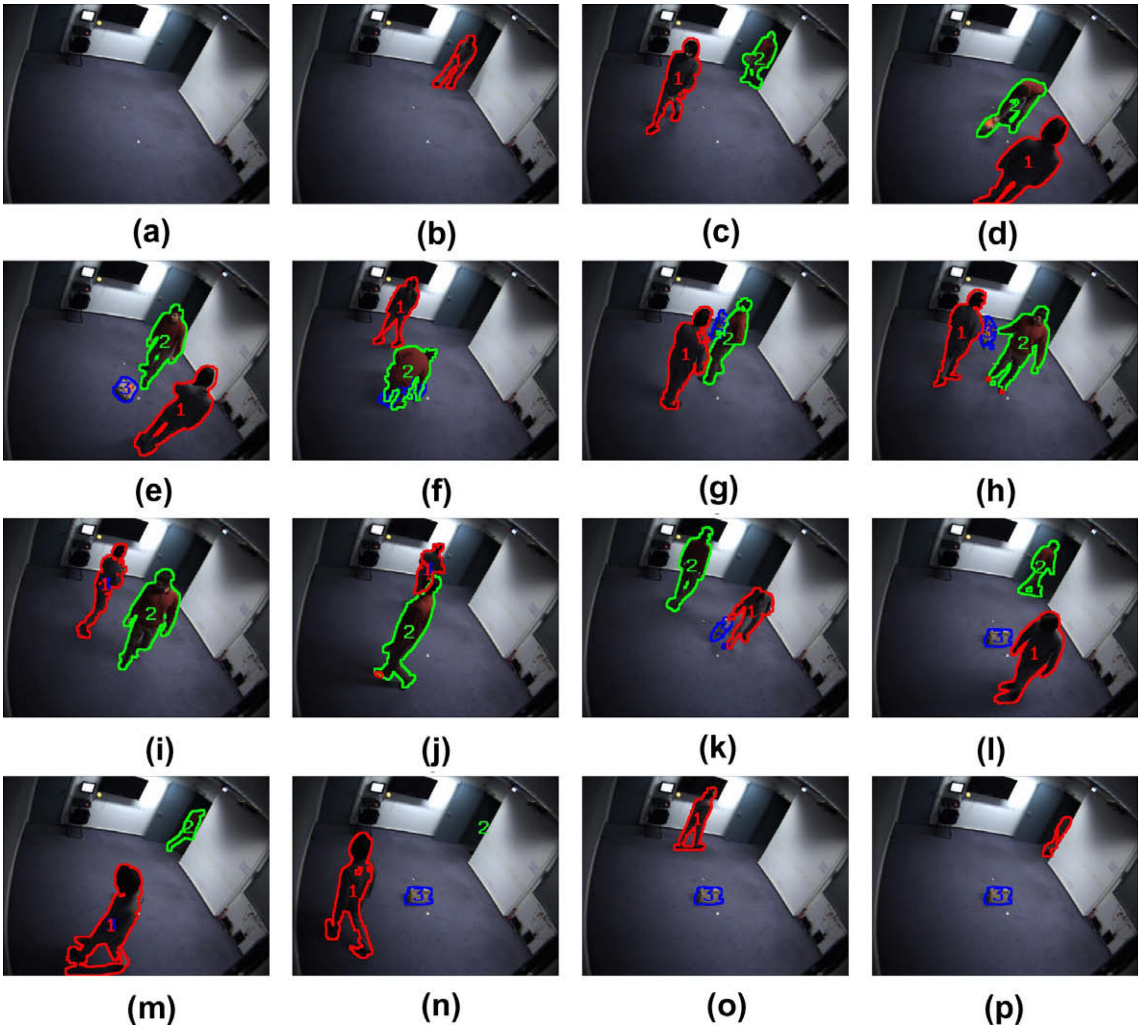

Fig. 8. Characteristic snapshots from the tracking experiment on the "bag" image sequence.

the whole sequence, object identities are correctly assigned and propagated in time.

In the above sequences, foreground detection was adequately accurate and tracking was not affected by errors in background subtraction. To investigate the influence of background subtraction on tracking, we performed a number of tests where we forced background subtraction to produce poor results. More specifically, we varied a parameter of the employed background subtraction method [23] that affects the true positives and false negatives of the method. As a test case, we considered the "buckets" sequence (Fig. 9). For a broad range of parameter values, tracking produced identical results. In extreme cases, the amount of false positives (or false negatives) produced by background subtraction affected the correctness of the performance of the method.

When background subtraction produces many false positives, several background pixels are labeled as foreground. Lenient background subtraction may produce hypotheses for non-existent objects. Additionally, foreground blobs are larger than the real objects. This results in the generation of wrong object hypotheses. Different objects appear connected in the foreground masks and object hypotheses are built for constellations of objects rather than for individual objects. The built appearance models might also be inaccurate because they are affected by the colors of the background pixels falsely identified as foreground, leading to inaccuracies and false similarities between objects.

When background subtraction produces many false negatives, several foreground pixels are labeled as background. Clearly, an object will not be tracked if it cannot account for a sufficiently large blob in the foreground mask. More often, a single object will give rise to multiple separate blobs. This violates the basic assumption that a single object gives rise to a single blob. As a result, multiple object hypotheses will be generated for a single object.

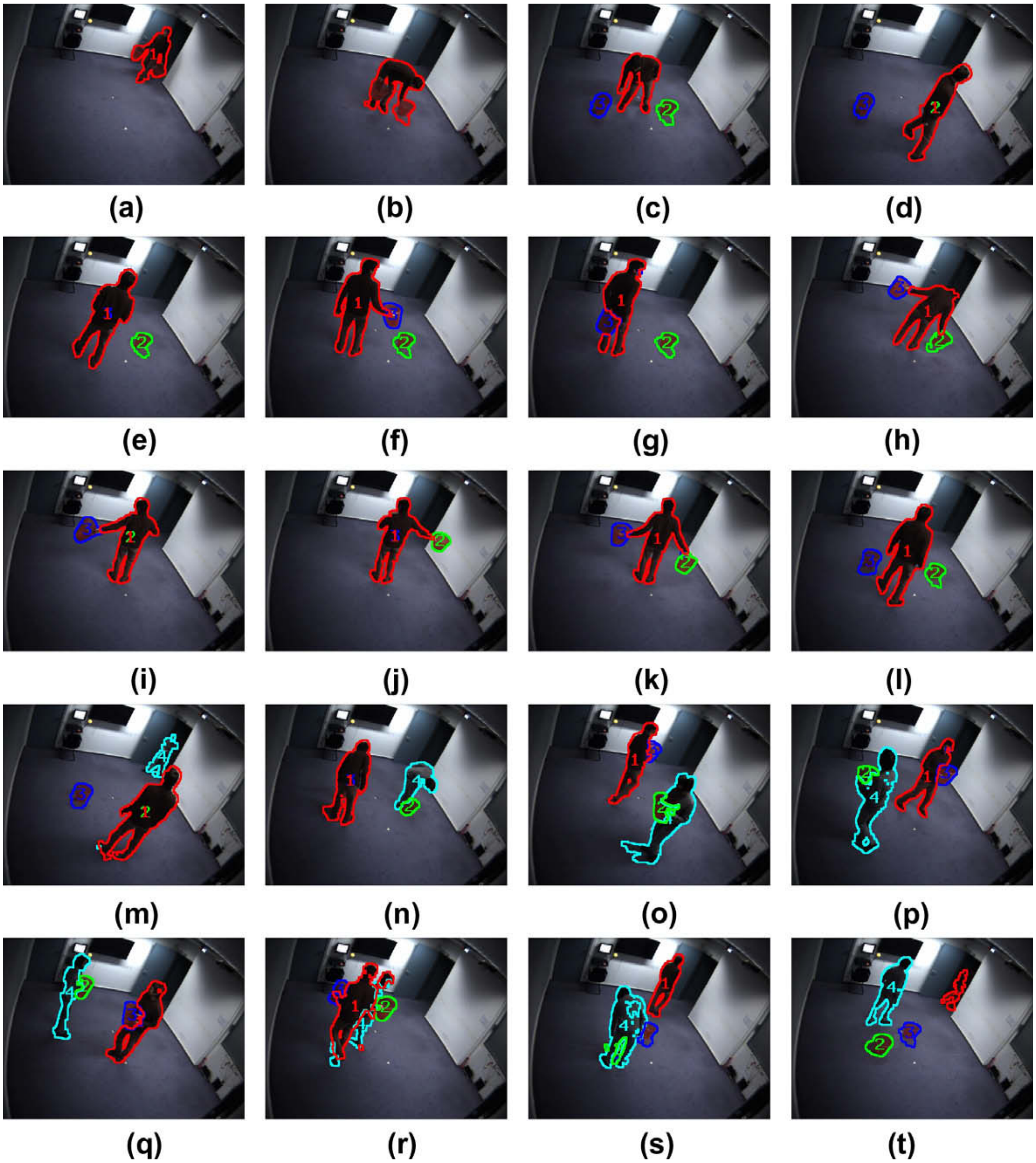

Fig. 9. Characteristic snapshots from the tracking experiment on the "buckets" image sequence.

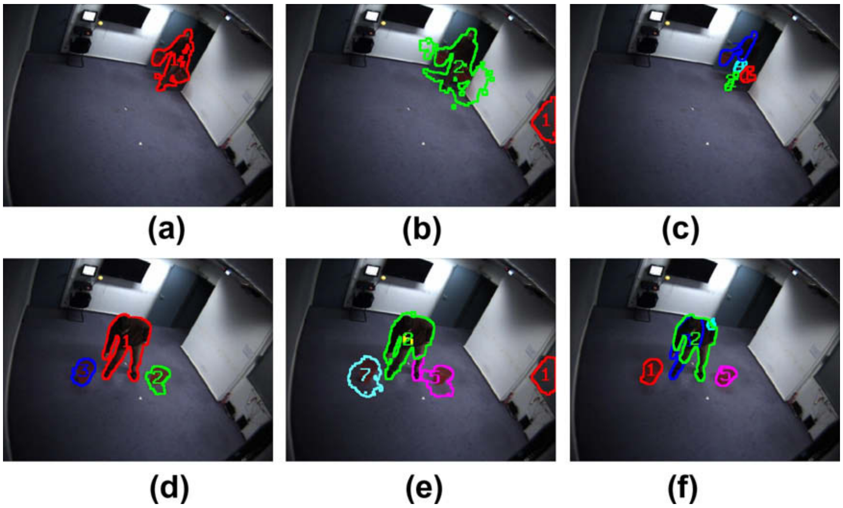

Fig. 10. Influence of background subtraction on the results of tracking for the case of the "buckets" sequence. The three columns show indicative tracking results for the cases of accurate, lenient and strict background subtraction.

Fig. 10 provides evidence for the behavior of the tracker with respect to the performance of background modeling and subtraction. Fig. 10a and d shows the results of the tracker for two frames of the "buckets" sequence (Fig. 9). In this particular experiment, background subtraction gives fairly accurate results. As a consequence, when the person enters the room (Fig. 10a) the person and the buckets he holds are identified as a single connected blob and object. Later on (Fig. 10d), the person leaves the two buckets on the floor, which results on their identification as two new objects. Fig. 10b and e shows the same frames in the case of lenient background subtraction. In Fig. 10b, background subtraction has already contaminated the object model of Fig. 10a with background pixels. Additionally, a wrong object hypothesis (object 1) has been created. When the person leaves the buckets on the floor (Fig. 10e), one of them is correctly identified, but the second one, although clearly separated from the person's body, has been assigned a considerable part of the background and of the person's foot. Fig. 10c and f shows the same frames in the case of strict background subtraction. When the person first appears in the scene, several individual objects are identified as a result of the disconnectedness of the foreground mask. When the person leaves the buckets on the floor, the buckets are correctly identified, but the originally detected and propagated object hypotheses still live on the region of the person's body.

5. Discussion

We presented a method for tracking multiple objects in the presence of occlusions with long temporal duration and large spatial extends. The proposed method can cope successfully with multiple objects dynamically entering and exiting the field of view of a camera and interacting in complex patterns. Towards this end, simple models of object shape, appearance and motion are dynamically built and used for supporting tracking and occlusion reasoning. Tracking is performed by systematically assigning pixels of foreground blobs to simple geometrical models of objects, taking into account object's appearance. Occlusion reasoning is based on the concept of "object permanence".

Our method is based on the approach proposed in [1]. As already stated in Section 1, the tracker proposed in [1] successfully tracks multiple skin color objects in images acquired by a possibly moving camera and can handle partial occlusions and short term full occlusions. Prior training is required to obtain the color model of the objects to be tracked. In this paper, we use background subtraction to find the image regions that are occupied by moving objects and thus we are able to use color information to track objects of different colors and to explicitly reason about occlusions. Therefore, our method assumes a steady camera for image acquisition but can handle many more cases of object appearance and interactions than [1].

Inspired by [6], our approach reasons about occlusions by relying on the concept of "object permanence". The authors in [6] use background subtraction, color, shape and spatial distribution to track objects in the presence of occlusions. Despite the fact that the two approaches share some methodological aspects, some key features allow our method to cope with a broader spectrum of situations. First of all, the work in [6] employs distinct region and object level tracking mechanisms. The evaluation of the region level correspondences is based solely on the shape and the displacement of the candidate regions (blobs). In our method, we omit this level by assigning blobs (regions) to object hypotheses in a direct manner that makes use of predicted object displacement, shape and color. By taking into account richer information about objects, errors in blob association are avoided. The major difference between the two methods is the treatment of objects of similar appearance. If two similar objects share the same blob, the method in [6] is forced to assign each pixel to a single object by using information about color and distance. This hard decision is bound to misclassify pixels and eventually distort the object models. The longer similar objects share the same blob the harder it gets to obtain the correct object shape and to acquire the correct object to region association when the region splits again. On the contrary, our method detects objects of similar appearance and uses the data association mechanism of [1]. Thus, depending on the spatial and appearance proximity of pixels to object models, pixels may be assigned to more than one object hypotheses.

The proposed method was successfully tested in the complete data set of [6]. Given the fact that the aforementioned data set does not contain even small interactions between objects of similar appearance, we tested our method on additional image sequences showing complex interaction between such objects ("lemons" and "buckets" sequences). The obtained experimental results demonstrate that the developed tracking methodology can successfully handle occlusions in challenging situations. The tracker incorporates and maintains very simple models of object shape, appearance and motion. This makes the tracker simple, fast and generic in the sense that no strong assumptions are imposed on the characteristics of the tracked objects. Our approach is expected to fail when objects to be tracked have too complex shapes and appearance or move with irregular motion patterns. Moreover, in our approach, successful background subtraction is an important factor that affects tracking. This is because background subtraction determines where in the scene action takes place and, therefore, what needs to be represented, modeled and associated between consecutive frames. If background subtraction has many false negatives, a single object may appear as a set of disconnected foreground blobs. This violates the main assumption, that a single object can give rise to a single blob. As a result, more than one object hypotheses will be generated for a single object. On the other hand, if background subtraction results in too many false positives, objects will be mixed with the background and their appearance models may drift and fail to accurately represent them. Towards removing these drawbacks, future research will consider the use of more elaborate spatial and appearance models that will provide more accurate object representations. Additionally, tracking results might be improved by a soft assignment of foreground pixels to object hypotheses as opposed to the current approach which bases this assignment on the strict notion of blob connectedness.

Acknowledgment

This work was partially supported by the IST-FP7-IP-215821 project GRASP.

References

[1] A.A. Argyros, M.I.A. Lourakis, Real-time tracking of multiple skin-colored objects with a possibly moving camera, in: European Conference on Computer Vision (ECCV), 2004, pp. 368–379.
[2] B. Baillargeon, E.S. Spelke, S. Wasserman, Object permanence in five-month-old infants, Cognition 20 (3) (1985) 191–208.
[3] G.J. Brostow, I. Essa, Motion based decompositing of video, in: International Conference on Computer Vision (ICCV), 1999, pp. 8–13.
[4] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, Series B (Methodological) 39 (1) (1977) 1–38.
[5] K. Fukunaga, Introduction to Statistical Pattern Recognition, second ed., Academic Press Inc., San Diego, CA, USA, 1990.
[6] Y. Huang, I. Essa, Tracking multiple objects through occlusions, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005, pp. 1051–1058.
[7] M. Isard, J. Maccormick, Bramble: a bayesian multiple-blob tracker, in: International Conference on Computer Vision (ICCV), vol. 2, 2001, pp. 34–41.
[8] A.D. Jepson, D.J. Fleet, T.F. El-Maraghi, Robust online appearance models for visual tracking, IEEE Transactions on PAMI 25 (10) (2003) 1296–1311.
[9] N. Jojic, B.J. Frey, Learning flexible sprites in video layers, in: IEEE Computer Vision and Pattern Recognition (CVPR), vol. 1, 2001, p. 199.
[10] S. Khan, M. Shah, Tracking people in presence of occlusion, in: Asian Conference on Computer Vision (ACCV), 2000, pp. 132–1137.
[11] B. Leibe, K. Schindler, L. Van Gool, Coupled detection and trajectory estimation for multi-object tracking, in: International Conference on Computer Vision (ICCV), 2007, 1–8.
[12] J.S. Marques, P.M. Jorge, A.J. Abrantes, J.M. Lemos, Tracking groups of pedestrians in video sequences, in: IEEE International Conference on Pattern Recognition (CVPR), 2003, p. 101.
[13] S.J. Mckenna, S. Jabri, Z. Duric, H. Wechsler, A. Rosenfeld, Tracking groups of people, Computer Vision and Image Understanding 80 (2000) 42–56.
[14] J. Piaget, The construction of reality in the child, Basic Books, San Diego, CA, USA, New York, 1937/1954.
[15] J.M. Rehg, T. Kanade, Model-based tracking of self-occluding articulated objects, in: International Conference on Computer Vision (ICCV), 1995, pp. 612–617.
[16] G. Sfikas, C. Constantinopoulos, A. Likas, N.P. Galatsanos, An analytic distance metric for Gaussian mixture models with application in image retrieval, in: ICANN (2), Lecture Notes in Computer Science, vol. 3697, Springer, 2005, pp. 835–840.
[17] C. Stauffer, W.E.L. Grimson, Adaptive background mixture models for real-time tracking, in: IEEC Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, 1999, p. 2246.
[18] Hai Tao, H.S. Sawhney, R. Kumar, Object tracking with bayesian estimation of dynamic layer representations, IEEE Transactions on PAMI 24 (1) (2002) 75–89.
[19] Y. Wu, T. Yu, G. Hua, Tracking appearances with occlusions, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, 2003, p. 789.
[20] Q. Yu, G. Medioni, I. Cohen, Multiple target tracking using spatio-temporal Markov chain Monte Carlo data association, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8.
[21] L. Zhang, Y. Li, R. Nevatia, Global data association for multi-object tracking using network flows, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008, pp. 1–8.
[22] Y. Zhou, H. Tao, A background layer model for object tracking through occlusion, in: International Conference on Computer Vision (ICCV), 2003, p. 1079.
[23] Z.Zivkovic, Improved adaptive gaussian mixture model for background subtraction, in: Proceedings of the International Conference on Pattern Recognition (ICPR), 2004.