Integrating tracking with fine object segmentation

Konstantinos E. Papoutsakis a, Antonis A. Argyros a,b,*

a Institute of Computer Science, FORTH, Greece
b Computer Science Department, University of Crete, Greece

ABSTRACT

We present a novel method for on-line, joint object tracking and segmentation in a monocular video captured by a possibly moving camera. Our goal is to integrate tracking and fine segmentation of a single, previously unseen, potentially non-rigid object of unconstrained appearance, given its segmentation in the first frame of an image sequence as the only prior information. To this end, we tightly couple an existing kernel-based object tracking method with Random Walker-based image segmentation. Bayesian inference mediates between tracking and segmentation, enabling effective data fusion of pixel-wise spatial and color visual cues. The fine segmentation of an object at a certain frame provides tracking with reliable initialization for the next frame, closing the loop between the two building blocks of the proposed framework. The effectiveness of the proposed methodology is evaluated experimentally by comparing it to a large collection of state of the art tracking and video-based object segmentation methods on the basis of a data set consisting of several challenging image sequences for which ground truth data is available.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Visual object tracking and segmentation constitute two fundamental, challenging computer vision problems. Methodologies that aspire to solve these problems, provide the foundations for addressing a number of other interesting problems and applications such as object recognition, 3D tracking, 3D reconstruction, augmented reality, HCI, etc. Object-based tracking and segmentation are highly interrelated problems. An accurate solution to the segmentation problem provides reliable observations to tracking. At the same time, tracking provides input that can guide segmentation, enhancing its performance. Thus, a method that integrates tracking and segmentation is intuitively appealing. Such a method could benefit from the tight coupling of these problems, address their individual challenges and provide reliable and robust solutions to both.

Our approach constitutes an online methodology for integrated tracking and fine segmentation of a previously unseen object in monocular videos captured by a possibly moving camera. No strong constraints are imposed on the appearance and the texture of the target object or the rigidity of its shape. The above may vary dynamically over time under challenging illumination conditions and background appearance.

The main goal of our work is to preclude tracking failures by enhancing its target localization performance through fine object segmentation that is appropriately integrated with tracking in a closed-loop algorithmic scheme. A kernel-based tracking algorithm [60], a natural extension of the popular mean-shift tracker [15,14], is efficiently combined with Random Walker-based image segmentation [25,26] in a closed loop algorithmic scheme. Explicit segmentation of the target region of interest in an image sequence enables effective and reliable tracking and reduces drifting by exploiting static image cues and temporal coherence.

Two algorithms are presented, both based on the proposed integrated tracking and fine object segmentation framework. A preliminary version of the first one, noted as TFOS, has already been reported in [36]. In the current paper the description of TFOS is more detailed. Additionally, we present several novel enhancements of TFOS that give rise to a new variant noted as TFOS+ that is comparatively evaluated with TFOS and other tracking and segmentation methods.

Further than that, we perform (a) an extensive quantitative and qualitative assessment of their performance and a discussion of the additional benefits of them over existing methods, (b) a study on how the defined parameters affect their performance, (c) a comparison with a variety of existing, state of the art tracking and video segmentation methods and (d) an analysis of the proposed methodology and comparison with the two concepts regarding the usage of a different segmentation process other than Random Walks-based method and the usage of no segmentation part in order to assess the additional benefits it introduces and was previously unobtainable by other methods. Experimental results demonstrate that TFOS+ outperforms TFOS. Additionally, both variants outperform most of the existing compared algorithms in
most of the employed test sequences, both in not loosing track of the
target but also in accurately segmenting it from the scene background.

In summary, the study of both TFOS and TFOS+ reveals that the key
benefits of the proposed framework are (i) the enhanced object tracking
and segmentation performance under challenging shape/color variations,
(ii) the capability to track an object from a moving camera (iii) the auto-
matic building and continuous refinement of both the object and the
background appearance models based on Random Walks and its effective
coupling with robust tracking, (iv) the adaptability of the defined param-
eters and algorithms based on partial tracking and segmentation perfor-
mance, (v) its ability to recover from partial failures of segmentation
and drifts of tracking based on the closed-loop interplay between them
and (vi) the modular workflow of the proposed framework that also
eases its extendability by incorporating additional image cues and devel-
opng additional mechanisms to track multiple objects.

The rest of this paper is organized as follows. Section 2 provides an
overview of relevant research efforts. TFOS and TFOS+ are presented in
Section 3. An extensive experimental evaluation of the proposed
methodology has been carried out. The goal was to evaluate TFOS and
TFOS+ quantitatively and qualitatively and to compare their perfor-
mance to state of the art tracking and video-based object segmentation
methods. A large collection of image sequences has been compiled,
representing challenging tracking scenarios involving non-rigid and de-
formable objects, strong illumination changes, moving cameras and
background clutter. The qualitative and quantitative experimental re-
results are reported in Section 4. Finally, Section 5 summarizes the main
conclusions of this work and suggests directions for future research.

2. Related work

Object tracking and segmentation in still images and videos have
been the theme of extensive research in the last couple of decades. A
complete review of the related literature, if possible, is a task beyond
the scope of this paper. In this section we review recent research on
the subject, with emphasis on methods that are related to the proposed
ones.

Despite the important research efforts devoted to the problem, the
development of algorithms for simultaneous tracking and segmentation
of objects in unconstrained videos with no prior knowledge of their ap-
pearance is an open research problem. Moving cameras, appearance
and shape variability of the tracked objects, varying illumination condi-
tions and cluttered backgrounds constitute some of the challenges that
a robust algorithm needs to cope with. In a comprehensive review,
Yilmaz et al. [56] identify state-of-the art tracking methods, cluster
them in classes regarding point tracking, silhouette tracking and kernel
and patch-based tracking methods and thoroughly analyze weaknesses
and powerful aspects of them. We adopt its categorization to support
the analysis of the following methodologies that are related to the pro-
posed system.

Point-tracking methods [30,8] combine tracking and fine object seg-
mentation using multiple image cues. Some of them utilize energy min-
imization techniques, such as Graph-Cuts or Belief Propagation on a
Markov Random Field (MRF) [59] or Conditional Random Field (CRFs)
[58,41] to perform fine segmentation of the target object. An alternative
approach is proposed in [12] where temporally consistent segmentation
of a moving object is performed by clustering long point trajectories
extracted from dense optical flow.

Silhouette-based tracking methods usually evolve an initial contour
to its new position in the current frame. This is performed by utilizing a
state space model [28] defined in terms of the shape and motion param-
meters [37] of the contour that encloses the tracked object or by the mini-
mization of an energy function defined over the object’s contour
[57,9,46]. In [39,50], object tracking is interpreted as a video-based ob-
ject segmentation task. Segmentation is performed by a variational ap-
proach based on a continuous energy formulation using anisotropic
weighted Total Variation (TV) regularization [35]. A recent method,

BHT, for tracking non-rigid objects is presented in [22], providing a
framework for interacting tracking and segmentation. The method ex-
tends the idea of Hough Forests and couples the generalized Hough
transform for object detection with the GrabCut [43] interactive seg-
mentation method. Thus, the method enables simultaneous object
tracking and rough segmentation in each video frame, reducing the
amount of the noisy training samples used for online learning.

Patch-based methods [19,20] are employed to deal with non-rigid,
arbitracted objects and their shape variations. Another recently proposed
approach, SPT [54] formulates a discriminative appearance model based
on mid-level structural information encoded in the superpixels [40].
Tracking based on this representation tolerates occlusions and makes
possible the recovery from drifts. Tracking-by-detection is increasingly
popular over the last years. Many methods exploit online learning tech-
niques to perform robust and unconstrained patch-based object tracking.
The IVT tracker [42] learns incrementally a low-dimensional subspace
representation of the target object. Incremental principal component
analysis is utilized to update the target’s models during the tracking pro-
cess, adapting online to large pose, scale and illumination variations.
The method presented in [3], considers weak classifiers that are combined
into a strong classifier using AdaBoost to tackle object tracking as a bina-
ry classification problem at the pixel level. The ensemble of weak classi-
fiers is trained online to distinguish between the object and the current
background, enabling online learning of object’s appearance. The exact
object location is determined by a subsequent mean-shift tracking proce-
dure. In a similar spirit, an on-line AdaBoost feature selection algorithm
for tracking is proposed in [23], enabling the on-line adaptation of the
classifier while tracking. The algorithm selects the most discriminating
features depending on the appearance of the background. Moreover, a
semi-supervised online boosting algorithm was introduced in [24] to
avoid drifting by performing supervised updates only at the first frame.

An alternative formulation of object tracking was recently proposed
in [4] using online multiple instance learning. The main idea is to select
elements of object and non-object patches on or near the current target
object position. In [29], an adaptive Lukas–Kanade tracker is combined
with a proposed P-N learning procedure to build an incremental binary
classifier. A robust object detector is established (TLD) able to avoid
drifting and to re-initialize tracking in case of a tracking failure. The
PROST method [45] exploits the concept of online random forests [44],
employing an on-line learning method with complementary tracking
approaches. These refer to a simple template model as a non-adaptive
stable component, a novel optical-flow-based mean-shift tracker and
an on-line random forest appearance-based learner, combined in a
cascade.

The method presented in [32] obtains samples from both the states
of the target and the trackers themselves using Markov Chain Monte
Carlo sampling. A four-dimensional tracker sampling space is defined,
consisting of appearance models, motion models, state representation
types, and observation types. Therefore, the sampled trackers perform
efficiently under challenging appearance variations and environmental
conditions.

Overall, methods based on online learning achieve fairly robust
tracking of previously unknown objects. However, most approaches
are limited to a bounding-box representation of the tracked object’s
shape, that is often of fixed aspect ratio and orientation. This coarse fore-
ground representation also includes background information, increas-
ing the amount of noise that is introduced during the online learning
and tracker update processes. As a result, they fail to handle challenging
tracking scenarios involving non-rigid objects.

While point and patch-based methodologies perform efficiently and
cover a broad range of appearance variations of the target object, their
weaknesses are mostly originated by inaccurate separation of object and
background pixels that further introduce erroneous information to the
tracking routine and benefit drifts towards the target object. To circum-
vent these problems, silhouette-based methods rely on accurate object-
background segmentation that further provides reliable observations to
tracking. The closed-loop coupling of tracking and fine object segmentation is also a key-point in our proposed methodology.

Various efficient methodologies for object segmentation/matting in images and videos tackle this problem explicitly and provide notable insights regarding the usage of visual cues and the extraction of features as well as techniques that can be adapted and incorporated into novel tracking methods. As such, Random Walker-based image segmentation is a method introduced for user interaction-based segmentation on still images [26,48] and is adapted and utilized in our proposed methodology for online simultaneous tracking and segmentation of an object. We extend our analysis of related work to a short list of methodologies that perform fine object segmentation or matting in still images or videos based on semi-automated or automated algorithmic schemes [53].

One of the first such methods was introduced in [13] building upon techniques in natural image matting, optical flow computation and background estimation. User interaction is exploited to compute keyframe segmentations corresponding to “foreground”, “background” and “unknown” masks, defining a “trimap”. The trimaps are interpolated across the video volume using forward and backward optical flow to compute high-quality alpha matte and foreground from a video sequence. Other proposed methods include the extension of image matting and 2D graph cut segmentation directly to 3D video volumes [5,2,11,52]. Thus, a combination of global and local color models with gradient and coherent image cues is formulated, treating video segmentation as a global optimization problem [33,38].

An alternative work interpolates partial boundary splines using rotoscopic techniques [1], whereas a highly efficient semi-automated segmentation method, introduced in [55], is based on minimization of the geodesic active contour energy incorporating a shape prior. The latter can also perform real-time visual object tracking. Two more prominent methods use adaptive local classifiers for segmenting dynamic video objects. They use a set of overlapping local classifiers, each integrating adaptively multiple local image features such as shape and local color and motion, which are propagated across frames by exploiting SIFT features [6,7]. The proposed interactive video segmentation method, called Video SnapCut (VSC) has been incorporated in the Adobe After Effects CS5, as a major new feature, the Roto Brush effect tool [49].

3. Proposed framework

The proposed framework for joint Tracking and Fine Object Segmentation (TFOS) is schematically illustrated in Fig. 1. To further ease understanding, Fig. 10 provides intermediate sample results of the main algorithmic steps.

It is assumed that at a certain time \( t \), an image frame \( I_t \) becomes available, while a binary object mask \( M_{t-1} \) is provided as the segmentation of the target foreground object from the scene background (see Fig. 10) at the previous time step \( t = 1 \). In general, \( M_t \) has a value of \( L_0 \) at object points of \( I_t \) and a value of \( L_0 \) at background points of \( I_t \). For time \( t = 1, M_0 \) should be provided by the user. The goal of the proposed method is to produce \( M_t \) for each frame \( I_t \).

Towards this end, prior spatial information is computed for the object represented by the 2D spatial mean and covariance matrix of the

![Fig. 1. Outline of the proposed joint object segmentation and tracking framework.](image-url)

![Fig. 2. Object tracking and affine propagation of its prior shape. (a) The prior object contour \( C_{t-1} \) and shape parameters \((\mu_{t-1}, \Sigma_{t-1})\) (red color) superimposed on \( I_t \). (b) Updated parameters \((\mu_t, \Sigma_t)\) (blue color) computed by the EMT [60] algorithm. (c) Affine propagation of \( C_{t-1} \) to \( C_t \) (blue color, Eq. (3)).](image-url)
image points belonging to the foreground object in $M_{t-1}$. This information can be interpreted as an ellipsoidal region coarsely representing the object’s position, scale, orientation and shape in $I_{t-1}$. Additionally, prior color information is computed based on an adaptive appearance model of the object, that is, a color histogram of the known, segmented foreground object region in $I_{t-1}$. This is computed using a Gaussian kernel function over that image subregion. The computed prior spatial and appearance models are exploited to initialize an iterative, kernel-based tracking algorithm [60] based on the Expectation Maximization method [17]. The iterative tracking thus performed, results in a prediction of the 2D spatial position and covariance of the tracked object in frame $I_t$.

The 2D affine transformation of the ellipsoid between times $t-1$ and $t$, is applied to the foreground object mask $M_{t-1}$ to yield a propagated object mask $M_t$. The Hausdorff distance between the object contour point sets of $M_{t-1}$ and $M_t$ is then computed and a shape band (or trimap) $B_t$ around the $M_t$ contour point set is determined, as in [57,58]. The width of $B_t$ is equal to the computed Hausdorff distance of the two contour point sets, determining a search area where the actual contour of the tracked object may be localized in $I_t$.

Subsequently, $M_t$ is distance transformed, resulting in pixel-wise spatial likelihoods on the predicted position and shape of the object in $I_t$. Moreover, pixel-wise color likelihoods are computed based on the appearance model of the tracked object (image pixels under $M_t$). Bayesian inference is applied to fuse the computed prior spatial and color image cues, estimating the PDFs of the object and the background regions. Given these PDFs, a Random Walker-based segmentation algorithm is finally employed to obtain an accurate segmented image $M_t$ for $I_t$.

In the following sections, the above steps comprising TFOS are presented in more detail. Moreover, extensions and improvements to this basic framework resulting in the TFOS+ variant are also identified.

3.1. Object tracking

3.1.1. Color-based object tracking

We employ the kernel-based tracking method [60] (denoted as EMT) that is closely related to the mean-shift tracking (MST) algorithm [15,14]. EMT represents the object’s 2D position, scale, orientation and shape by a 2D ellipsoid region after its center of mass $\mu_t$ and covariance matrix $\Sigma$. The appearance model of the tracked object is a color histogram $H_{t-1}$ of the image points corresponding to the 2D ellipsoid region $\{\mu_t, \Sigma\}$, and is computed using a Gaussian weighting kernel function. The histogram $H_{t-1}$ can be computed given $M_{t-1}$ and $I_{t-1}$ as well as parameters $\{\mu_{t-1}, \Sigma_{t-1}\}$. For each new image $I_t$, this prior information is utilized to initialize an iterative tracking process that evolves the ellipsoid region determining the image area in $I_t$ that best matches the appearance model $H_{t-1}$ of the tracked object in terms of a Bhattacharyya coefficient-based color similarity measure. This computes the parameters $\{\hat{\mu}_t, \hat{\Sigma}_t\}$, yielding a prediction on object’s position and covariance in $I_t$.

![Fig. 3. Illustration of the fixed width shape band (a)–(c) and the varying width shape band (d)–(f). Due to the articulated 3D motion of the arm, the area around thumb is expected to exhibit larger displacement than the lower part of the palm. In (a)–(c) the shape band is of equal width in each image. In (d)–(f) the shape band is adaptively defined.](image)

![Fig. 4. Shape band and spatial prior image cues. (a) The outer and inner outlines of $B_t$. (b), (c) The probability maps $P(I_{t, x_i})$ and $P(I_{t, x_i})$, respectively (blue color indicates probability near zero, red color indicates probability towards 1).](image)
3.1.2. Affine propagation of object shape

The employed object tracking algorithm represents coarsely the scale and the shape of the target object with a 2D ellipsoid. Since the shape of an object is not precisely described by an ellipse, the computed appearance model is forced to incorporate color information of background pixels, causing tracking to drift. One of the primary goals of this work is to prevent tracking drifts by integrating tracking with fine object segmentation. Therefore, only pixels actually belonging to the object should contribute to the appearance model used for object tracking.

Towards this end, a first step is to propagate the prior spatial information regarding the object’s shape and position to the new frame.

The object contour $C_{t-1}$ in $M_{t-1}$ is propagated to the current frame $l_t$, exploiting a transformation that is suggested by the computed parameters $\mu_{t-1}$, $\Sigma_{t-1}$ and $\Sigma_t$. An affine transformation is defined between these two 2D spatial distributions. More specifically, a linear $2 \times 2$ affine transformation matrix $A_t$ is computed based on the square root $(\Sigma^{1/2})$ of $\Sigma_{t-1}$ and $\Sigma_t$. A $2 \times 2$ covariance matrix $\Sigma$ is a square, symmetric and positive semi-definite matrix, therefore its square root can be calculated by diagonalization as:

$$\Sigma^{1/2} = QA^{1/2}Q^{-1},$$

where $Q$ is a square $2 \times 2$ matrix whose $i$th column is the eigenvector $q_i$ of $\Sigma$ and $A^{1/2}$ is a diagonal matrix, the diagonal elements of which represent the square values of the corresponding eigenvalues. The inverse of the $Q$ matrix is equal to the transpose matrix $Q^T$, therefore $\Sigma^{1/2} = QA^{1/2}Q^T$. Accordingly, we compute the transformation matrix $A_t$ as:

$$A_t = Q_{2 \times 2}A^{1/2}_{t-1}Q^{-1}_{t-1}Q^T_{t-1}.$$  

Finally, $C_t$ is derived from $C_{t-1}$ based on the following transformation

$$C'_t = A_t(C_{t-1}-\mu_{t-1}) + \mu_t.$$  

The propagated contour $C'_t$ is then used to define the propagated object mask $M'_t$ that serves as a prediction for the position, scale, orientation and shape of the tracked object in $l_t$ (Fig. 2).

3.2. Object segmentation

3.2.1. Shape band

$C_t$ represents a prediction of the position, scale, orientation and shape of the object in $l_t$, approximating its unknown, actual contour $C$. This is based on the assumption that the object’s appearance and shape do not undergo large variations between consecutive frames. To estimate $C_t$ given $C'$, we utilize the concept of object shape band employed in [57,58]. The shape band is an image subregion defined around $C_t$, based on the “trimap” used for object segmentation (see Section 3.1.2). The object shape band $B_t$ is considered as an area in which $C_t$ is to be found. We consider two variants of the shape band. In the first one, employed in the TFOS algorithm, the shape band has a fixed width. In the second, employed in the TFOS+ algorithm, a shape band of varying width is defined.

3.2.1.1. Fixed width shape band. The width $w$ of the shape band $B_t$ is set equal to the 2D Hausdorff distance $H$ between the contour point sets $C_{t-1}$ and $C'_t$ (see Section 3.1.2). In notation,

$$w = H(C'_t, C_{t-1}).$$

(4)

The calculated width $w$ is not allowed to exceed a maximum value $w_{max}$ and to be less than that a minimum value $w_{min}$, both of which are fractions of the distance (in pixels) of the object’s center of mass from the furthest contour point in $C_t$. An example of $B_t$ thus defined is shown in Fig. 3(a).

3.2.1.2. Varying width shape band. The fixed width shape band is rather conservative. Experiments have shown that this results in a large area within which the actual object contour can be localized. An alternative shape band definition considers individual width values at each point of $C'_t$ (see Section 3). More specifically, the width at each point is calculated as a function of its individual displacement between $C_{t-1}$ and $C'_t$. Thus, the shape band has smaller width around contour points that move slower compared to those that exhibit larger displacements.

To do this, we first introduce the measure $\lambda_{ht}$ which is defined as

$$\lambda_{ht} = \frac{1}{S} ||\mu_{t-1} - \mu_{t-1}||_2.$$  

(5)

In Eq. (5), $S$ is a scale factor which guarantees that $0 < \lambda_{ht} \leq 1$. Essentially, $\lambda_{ht}$ is the normalized Euclidean distance of the 2D position of the object as it has been predicted based on tracking and the final 2D position of the object as it has been computed after segmentation, both at the previous time step. As such, $\lambda_{ht}$ quantifies the discrepancy between the tracking solution and the segmentation solution. Thus, the smaller $\lambda_{ht}$ is, the larger our confidence on the tracking result.

Having defined $\lambda_{ht}$, we compute an intermediate object contour $C''_t$ as in Eq. (3) and we derive a shape band of varying width around $C''_t$.  

---

Fig. 5. Probabilistic fusion of prior image cues. (a) Input image $l_r$. (b) $P(l_r, c)$, (c) $P(l_r, x)$. (d) $P(l_r, c, x)$ (based on Eq. (8)).

Fig. 6. Adaptive attenuation of prior color information. (a) Input image. The object shape band is indicated in blur. (b) The wrong segmentation result provided by TFOS. (c) Points within the shape band for which $\lambda = 0.5$ are shown in green color. (d) The improved segmentation result provided by TFOS+. 
The Distance Transform is computed starting from the outer contour $B_t$. A second 2D Distance Transform based on the inner contour of $P$ we set to $B_t$. The appearance model of the background is represented by the color histogram $H_B$, computed over image points in $I_{t-1}$ labeled as $L_B$ in $M_{t-1}$, encoding the conditional probability $P(c|L_B)$. The TFOS variant represents color-based priors by employing the aforementioned approach. In TFOS+, we consider the more elaborate approach of adaptive attenuation for the prior color cue, as described in Section 3.2.4.

### 3.2.4. Probabilistic fusion of prior image cues

Object segmentation can be considered as a pixel-wise binary classification problem. Our goal is to generate the posterior probability distribution for each of the labels $L_O$ and $L_B$ based on the computed prior image cues. This information further guides the Random Walker-based image segmentation. Using Bayesian inference, we combine the available prior color and spatial image cues, computed as described in Sections 3.2.2 and 3.2.3. Considering the pixel color $c$ as evidence and conditioning on image point location $x$ in $I_t$, the posterior probability distribution for label $L_i$ is given by:

$$P(L_i|c,x) = \frac{P(c|L_i,x)P(L_i|x)}{\sum_{L_i} P(c|L_i,x)P(L_i|x)}. \quad (7)$$

The conditional probability $P(c|L_i,x)$ encodes the probability of an image point having color value $c$ given its label $L_i$ and its location $x$. Assuming the independence of position $x$ and color $c$, the probability of color $c$ is only conditioned on the prior knowledge of its class $L_i$. Thus, $P(c|L_i,x) = P(c|L_i)$. Given this, Eq. (7) can be rewritten as

$$P(L_i|c,x) = \frac{P(c|L_i,L_t|x)P(L_t|x)}{\sum_{L_i} P(c|L_i,L_t|x)P(L_t|x)}. \quad (8)$$

The conditional color probability $P(c|L_i)$ for the class $L_t$ is computed based on the region-based color histogram $H_c$, whereas the conditional spatial probability $P(L_t|x)$ is computed based on the Distance Transform measure calculation, as described in Section 3.2.2.

An illustration of the probabilistic fusion of prior image cues is shown in Fig. 5. The example shows a hand that needs to be tracked in a cluttered background containing other skin-colored objects. Therefore, non-separable colors are present in both the foreground and the background (Fig. 5(b)). The pixel-wise spatial likelihood around the predicted hand area (Fig. 5(c)), is strong enough to resolve the ambiguity that arises in the presence of non-separable colors (see Fig. 5(d)).

### 3.2.4.1. Adaptive attenuation of prior color cue

The representation of color priors in TFOS (Section 3.2.3) leads to reduced segmentation performance when the color distributions of the tracked foreground object and the background have a considerable overlap. In such cases, spatial information provided by shape propagation might not suffice to prevent from tracking drifts. An example segmentation failure is shown in Fig. 6(a)–(b), where the object of interest, a clown fish, is to be segmented within a cluttered background containing similar colors. A solution to this problem is to attenuate the significance of color likelihoods, based on a measure of similarity of the object and background prior color distributions. This mechanism is incorporated in TFOS+. More specifically,

![Fig. 8](image-url)
given the normalized prior color likelihoods \( P(c|l_i) \), we compute the residual \( r(c) = P(c|l_0) - P(c|l_2) \) that is utilized to compute a color confidence factor \( \lambda(c) \) as:

\[
\lambda(r(c)) = \begin{cases} 
1 - a e^{-r(c)^2/\alpha^2}, & |r(c)| \geq \sigma \sqrt{\ln\left(\frac{\alpha}{1-\alpha}\right)} \\
\delta, & \text{otherwise}
\end{cases}
\]  

(9)

We set \( \delta = 0.5 \), whereas \( \alpha \in [1, 5] \) and \( \alpha \in [0.1, 1] \). A plot of function \( \lambda(r(c)) \) is illustrated in Fig. 7.

An attenuated prior color likelihood value, \( P_\lambda(c|l_i) \), is computed as

\[
P_\lambda(c|l_i) = \begin{cases} 
\lambda(r(c))P(c|l_i), & \text{if } \lambda(r(c)) > 0.5 \\
\lambda(r(c))P(c|l_i) + (1 - \lambda(r(c)))P(c|l_j), & \text{if } \lambda(r(c)) = 0.5
\end{cases}
\]  

(10)

In Eq. (10), \( c \) is the colors of image points in the shape band \( B_i \), \( l \in \{l_0, l_2\} \), and \( l' = l_0 \) else \( l' = l_2 \). Higher \( r(c) \) values result in higher color confidence \( \lambda \). Therefore, a color value \( c \) that appears frequently in both the background and the tracked object will be assigned a minimum confidence value \( \lambda = 0.5 \). In this case, the attenuated prior color likelihoods \( P_\lambda(c|l_0) \) and \( P_\lambda(c|l_2) \) are set equal to each other.

The adaptive attenuation of the color cue prior enables the prior shape cue to prevail in the probabilistic integration of the cues represented by the joint likelihoods \( P(L|c, x) \) (see Eq. (8)). Fig. 6(c) provides an example of the effectiveness of this mechanism. Image points in green color within the shape band have \( \lambda(c) = 0.5 \), expressing that colors in these regions might belong either to the tracked object or the background. For points within this subregion of the shape band, the prior shape information will prevail over the color. The segmentation result, shown in Fig. 6(d), highlights the enhanced segmentation performance, annihilating the degradation of the segmentation performance in the presence of non-separable color distributions of the tracked object and the background.

3.2.5. Object segmentation based on Random Walks

The resulting posterior probability distributions \( P(L|c, x) \) on image points \( x \in l \), for labels \( l \in \{l_0, l_1\} \) are used to guide a Random Walker-based image segmentation of the tracked object in \( l \). Random Walks for image segmentation was introduced in [27] as a method to perform \( K \)-way graph-based image segmentation. The method assumes that for each of the \( K \) disjoint image segments there is at least one class member, i.e., pre-labeled image point. The principal idea behind the method is that one can analytically determine the real-valued probability of a random walker starting at each unlabeled image point to first reach any of the pre-labeled points. The random walker-based framework bears some resemblance to the graph-cuts framework for image segmentation, as they are both related to spectral clustering [51], but they also exhibit significant differences [26].

The algorithm is formulated on a discrete weighted undirected graph \( G = (V, E) \), where nodes \( u \in V \) represent the image points and the positive-weighted edges \( e \in E \subseteq V \times V \) indicate their local connectivity. The solution is calculated analytically by solving \( K - 1 \) sparse, symmetric, positive-definite linear systems of equations, for \( K \) labels.
In order to represent the image structure by random walker biases in TFOS, color information is exploited. We map the edge weights to positive weighting scores computed by a Gaussian weighting function over the normalized Euclidean distance of the color intensities between two adjacent pixels, practically the color contrast. Texture-based image information is also exploited in conjunction with the Gaussian weighting function in the TFOS+ variant, as described later in this section.

The Gaussian weights \( w_{ij} \) of the edges between points \( i, j \in V \) are defined as:

\[
   w_{ij} = e^{-k \left\| c_i - c_j \right\|^2} + \epsilon, \tag{11}
\]

where \( c_i \) stands for the vector containing the color channel values of image point/node \( k \), \( k = 10^{-6} \) and \( \rho \) is a normalizing scalar defined as

\[
   \rho = \max_{i,j \in V} \left\{ \left\| c_i - c_j \right\| \right\}. \tag{12}
\]

The parameter \( \beta \) is user-defined and modulates the spatial random walker biases. The posterior probability \( P(L|c, x) \) represents the probability of label \( L \) for a color \( c \) at image location \( x \). Therefore, we consider pixels of highest posterior probability values for label \( L \) as pre-labeled/seed pixels for that label in the formulated graph.

An alternative formulation of the Random Walker-based image segmentation is presented in [25], incorporating non-parametric probability models, hence prior belief on label assignments. In this formulation, the \( \gamma \) scalar weight parameter is introduced to adjust the contribution of the prior belief to the belief computed by the random walks in the image. The extended formulation of using both seeds and prior belief is compatible with our approach, considering the obtained posterior probability distributions \( P(L|c, x) \) for the two image labels \( L = \{ L_0, L_p \} \). The Random Walker formulations using prior models, suggest for a graph construction similar to the graph-cut algorithm [10]. In graph cuts terminology, the edge weights of the constructed graph can be seen as the \( N \)-links or \( link \)-terms and the prior belief values of the graph nodes for any of the potential labels can be considered as the \( T \)-links or the \( data \)-terms.

For any of the formulations of Random Walker-based segmentation, the primary output consists of \( K \) probability maps, that is, a soft image segmentation result per label. By assigning each pixel to the label for which the largest probability is calculated, a \( K \)-way segmentation is obtained. This process gives rise to the object mask \( M_t \) at image frame \( t \). Fig. 8 provides an illustration of intermediate results pertaining to the Random Walker-based segmentation methodology.

3.2.5.1. Integration of texture-based image cues. On top of the color and shape cues integrated to TFOS, TFOS+ also incorporates texture-based image cues. The proposed addition exploits brightness changes between neighboring image pixels to map the image structure into random walker biases. A filter-based representation [31] of image structure is considered taking into account the response values of a set of 2D filters adjusted to various orientations and spatial frequencies [34] in the neighborhood of each image pixel. The response of this filtering consists of a vector of scalar values for each pixel corresponding to responses of the applied filters. This vector is used in Eq. (11) similarly to color contrast information.

Various frameworks for texture-based image analysis exist. We adopt the steerable pyramid approach [47,21]. An example of the steerable pyramid computed on an input image for six orientations and two scale levels is shown in Fig. 9.

The convolution of the steerable filters and the scale levels results in a response vector, where each value is a scalar that corresponds to filter response for each scale level. The response vector is computed at each pixel \( x \) of \( I \) and utilized in Eq. (11), replacing the color information \( c \) of a point \( x \) and mapping the texture profiles of neighboring pixels to random walker biases. The resulting values of Eq. (11) are further utilized to populate the Laplacian matrix of the Random Walker [26].

Therefore, two connected pixels of the graph-based representation of the image with similar texture profiles will result in a weak random walker bias, thus high probability to be assigned to the same segmentation label. In our implementation, an existing toolbox for multi-scale image processing1 was used.

Fig. 10 recapitulates the flow of information of the proposed methodology by also illustrating representative intermediate results.

4. Experimental evaluation

The evaluation of the proposed methodology incorporates both a quantitative and a qualitative assessment of TFOS and TFOS+, together with comparisons against a variety of state of the art methods for video-based object segmentation and visual object tracking. Moreover, we assess the performance of the proposed methodology to

---

1. [http://www.cns.nyu.edu/~eero](http://www.cns.nyu.edu/~eero)
The performance of the EMT tracker [60] used in TFOS and TFOS+ depends on the representation of the color information. For example, skin colored objects are better discriminated in HSV or in YCbCr compared to other color-spaces. By discarding the V (brightness value) or Y (luminance component), the skin-color representation becomes more robust to illumination changes. Still, aiming at generality, in our experiments we choose the RGB color space for all sequences. The appearance model of the tracked object consists of a color histogram of 8 bins per dimension of the RGB space.

The performance of the tracking component of the method also depends on the convergence criterion used in its internal EM algorithm. Alternatives include a fixed number of EM iterations or a stopping threshold value reflecting the percentage of image points added/removed in the ellipsoid tracking region in two consecutive iterations. We selected the convergence criterion to be a combination of both criteria. Thus, tracking is terminated in 20 iterations, or earlier if less than 5% of the image points were added/removed in the ellipsoid tracking region. In the performed experiments, the mean value of EM iterations was measured equal to 10.

Regarding segmentation, one issue affecting the performance of the Random Walker (see Section 3.2) is the graph connectivity policy that adjusts the sparsity of the Laplacian matrix and the weighting function that is chosen to compute the weights $w_{ij}$ that populate it. We employ 4-connectivity of the corresponding image points. Regarding the weighting function, a Gaussian is utilized (Eq. (11)). This introduces a single parameter $\beta$ that adjusts the variance of the Gaussian function, thus the severity of the random walks biases on the graph (see Fig. 11). In our experiments, the values of $\beta$ parameter ranges within the interval $[1\ldots200]$.

A second issue regards the selection among the three Random Walker variants presented in Section 3.2.5. These refer to the usage of seeds (pre-labeled graph nodes), prior values (probabilities/beliefs on label assignments for some graph nodes), or a combination of them. In the variants using seeds, each image point $x$ with $P(L|x) \geq 0.9$ is considered as a seed pixel for the label $L_x$. All other points with $P(L|x) < T$ provide prior values for label $L_x$. The $T$ parameter stands for a constant value that is usually selected equal to 0.9. In the variant using seeds and prior values, a parameter $\gamma$ is introduced to adjust the degree of authority of the prior beliefs towards the definite label-assignments expressed by the seed pixels of the image. The $\gamma$ parameter ranges within $[0.05\ldots0.5]$.

Finally, a third issue affecting the segmentation performance regards the appearance models utilized to capture the region-specific color information of the segmented objects. A color histogram per label (object,
Table 2
Percentage (Cbox criterion, see Section 4.3.1) of correctly tracked frames by each tracking method on each image sequence. The best and second best results are shown in boldface and in italic font in parentheses, respectively.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>IVT</th>
<th>TLD</th>
<th>SPT</th>
<th>HBT</th>
<th>Tfos</th>
<th>EMt</th>
<th>MST</th>
<th>BBT</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand-1</td>
<td>80</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>99</td>
<td>1</td>
<td>36</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td>Hand-2</td>
<td>55</td>
<td>37</td>
<td>85</td>
<td>100</td>
<td>100</td>
<td>58</td>
<td>1</td>
<td>68</td>
<td>67</td>
</tr>
<tr>
<td>Grasp-1</td>
<td>17</td>
<td>3</td>
<td>4</td>
<td>100</td>
<td>100</td>
<td>13</td>
<td>2</td>
<td>22</td>
<td>40</td>
</tr>
<tr>
<td>Grasp-2</td>
<td>44</td>
<td>77</td>
<td>62</td>
<td>98</td>
<td>100</td>
<td>73</td>
<td>1</td>
<td>34</td>
<td>65</td>
</tr>
<tr>
<td>John</td>
<td>48</td>
<td>100</td>
<td>48</td>
<td>11</td>
<td>12</td>
<td>100</td>
<td>1</td>
<td>11</td>
<td>48</td>
</tr>
<tr>
<td>Anto</td>
<td>92</td>
<td>55</td>
<td>87</td>
<td>97</td>
<td>55</td>
<td>25</td>
<td>10</td>
<td>55</td>
<td>59</td>
</tr>
<tr>
<td>Foreman</td>
<td>94</td>
<td>62</td>
<td>96</td>
<td>100</td>
<td>100</td>
<td>93</td>
<td>1</td>
<td>34</td>
<td>76</td>
</tr>
<tr>
<td>Fish</td>
<td>92</td>
<td>8</td>
<td>98</td>
<td>80</td>
<td>100</td>
<td>86</td>
<td>2</td>
<td>5</td>
<td>69</td>
</tr>
<tr>
<td>Book-1</td>
<td>56</td>
<td>51</td>
<td>95</td>
<td>48</td>
<td>100</td>
<td>65</td>
<td>4</td>
<td>46</td>
<td>63</td>
</tr>
<tr>
<td>Book-2</td>
<td>85</td>
<td>3</td>
<td>98</td>
<td>59</td>
<td>97</td>
<td>100</td>
<td>83</td>
<td>2</td>
<td>41</td>
</tr>
<tr>
<td>Cliff-dive</td>
<td>34</td>
<td>45</td>
<td>92</td>
<td>93</td>
<td>45</td>
<td>64</td>
<td>14</td>
<td>71</td>
<td>64</td>
</tr>
<tr>
<td>Caterpillar</td>
<td>14</td>
<td>11</td>
<td>50</td>
<td>99</td>
<td>100</td>
<td>37</td>
<td>5</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Mean Cbox (%)</td>
<td>59</td>
<td>41</td>
<td>76</td>
<td>82</td>
<td>(84)</td>
<td>93</td>
<td>62</td>
<td>8</td>
<td>40</td>
</tr>
</tbody>
</table>

background) is utilized in the proposed methodology (see Section 3.2.3).

For both histograms, 16 bins were considered per dimension of the color space.

4.2. Test image sequences

A large set of test image sequences representing challenging tracking and segmentation scenarios has been compiled to form the basis of the evaluation of Tfos and Tfos +. Each sequence illustrates a single object, undergoing challenging color appearance variations, scale changes and shape deformations. The selected sequences also differ with respect to the camera motion and the illumination conditions.

Fig. 12 provides a single frame from each of the sequences. 2 Table 1 provides further details regarding their content and the main challenges each of them introduces. Ground truth data were also compiled, consisting of (a) manually extracted, detailed segmented binary masks of the target object’s area for each frame of all sequences and (b) the horizontal bounding boxes of the object of interest in each frame.

Eight of these twelve test sequences (Fig. 12(a)–(h)) were either grabbed at our lab or were already available in the web. These videos represent a large variety of challenging conditions regarding the camera motion, target object’s appearance variations and illumination changes.

The four other videos (Fig. 12(i)–(l)) have already been employed in previous, related, scientific publications for the evaluation of tracking algorithms. More specifically, sequences in Fig. 12(i),(j) were taken from [16], based on the free dataset of the i2i – 3D Visual Communication available from Microsoft. 3 The Cliff-dive sequence in Fig. 12(k) was used in [22] and is available online. 4 Finally, the Foreman sequence illustrated in Fig. 12(l) is also available online (YUV Video Sequences). 5

4.3. Quantitative results

We conducted a two-fold quantitative assessment of the performance of the proposed methodology, regarding the individual segmentation and the tracking performance on all the test image sequences. The two proposed methods are compared to state-of-the-art object trackers and to methods performing video-based object segmentation.

Moreover, regarding the tracking performance, we compare to the tracking only EM-Shift algorithm of the proposed framework and the original MeanShift method.

Finally, to assess the importance of our choice of Random Walker as the segmentation algorithm employed in the proposed framework, we implemented another variant (BBT) that is identical to Tfos + except for the fact that segmentation is performed by thresholding the probability values computed by Eq. (8).

4.3.1. Evaluation criteria

Three criteria are selected to evaluate quantitatively the performance of tracking. Given the bounding box predicted by the tracker in a frame and the ground truth bounding box of the object in the same frame, their overlap ratio or F-measure can be computed as the area of their intersection over the area of their union. As in [18], an overlap ratio greater than 50% signifies a correct tracking result, otherwise the frame is considered as a tracking failure/miss. A first tracking metric is the percentage Cbox of correctly tracked frames for a sequence. A second metric is the mean F-measure Fbox for an image sequence, computed only over the successfully tracked frames. Finally, the mean distance Dbox between the centers of the estimated and the ground truth bounding boxes is computed in a sequence, again only for successfully tracked frames.

For the problem of segmentation, metrics Cseg, Fseg analogous to those used for tracking are defined, with the exception that instead of bounding boxes, the binary segmentation masks are used. More specifically, a frame is considered as a segmentation hit if the overlap ratio of the estimated and the ground truth object mask is at least 0.5. Segmentation hits are used to define the percentage of correctly segmented frames Cseg in a sequence. Additionally, the average F-measure between the estimated and the ground truth object masks for all frames of a sequence that constitute segmentation hits is also computed giving rise to the metric Fseg.

4.3.2. Object tracking

A detailed quantitative assessment of the tracking performance of the proposed methods is reported based on the selected image sequences. We compare Tfos and Tfos + to a variety of existing tracking algorithms including IVT [42], TLD [29], SPT [54], BHT [22], MST [14], EMT [60] and BBT. Other well known, efficient tracking methods were excluded from this comparative evaluation, as their available implementation did not support scale adaptation of the tracking bounding box to the actual size of the target object.

Each algorithm is initialized using the ground truth bounding box on the first frame of each sequence. The Tfos, Tfos +, MST and EMT tracking methods report an elliptical target region containing the object in each frame. To become comparable with other tracking methods that do not provide this information, the minimum horizontal bounding box to this elliptical region is computed as a tracking result.

For a fair comparison, we adjusted the parameters of every tracking method (as provided by the available implementations), towards achieving optimal performance according to the selected evaluation metrics. The IVT algorithm was tested on gray-level versions of the test image sequences. A common parameter configuration was utilized for the MST and EMT trackers and the tracking part of Tfos and Tfos +. Regarding BBT, we set the same parameter configuration as in EMT for tracking, whereas for segmentation we chose a threshold value per test sequence that maximized its performance with respect to the computed results.

Table 2 presents the Cbox metric (percentage of correctly tracked frames as defined in Section 4.3.1) for each algorithm and image sequence. The superior performance of the EMT tracker compared to the MeanShift method is clear, as the latter drifts away from the target object into background regions in the very first frames for almost all sequences and was unable to recover the target.

We observe that the BBT method does not achieve satisfactory scores for the Cbox metric. However, as shown in Tables 3 and 4 its tracking accuracy based on Fbox and Dbox criteria is superior to EMT for the proportion of successfully tracked frames in each test sequence. Therefore, EMT could enhance its performance and benefit from a more successful fine segmentation algorithm that can separate an object from
its background and facilitate robust refinement of appearance models and accurate initialization for tracking in the next frame. This justifies our choice to build upon EMT in our framework and combine with Random Walks-based image segmentation method. The performance of TFOS+ (best) and TFOS (second best) shows the benefits of the proposed approach based on the closed-loop coupling between tracking and segmentation. Additionally, TFOS+ clearly outperforms TFOS relying on the proposed enhancements described above.

The IVT tracker performs well on specific sequences (Hand-1, Fish, Anto, Foreman), where the target object’s area is rigid and of limited extent with respect to the frame size and adapts well to extensive illumination changes. However, a moderate performance is reported in the presence of large non-rigid object and scale variations (Grasp-1, Cliffs, Dive, Caterpillar sequences). A similar behavior is demonstrated by the TLD tracker. However, lower performance scores are reported, as it fails to track over the 50% frames in the majority of sequences. Both methods are heavily affected by the background regions introduced in the bounding boxes of the objects in each frame and transferred to their learning-update procedures. This effect is pronounced in the case of a time varying background.

For the successfully tracked frames, Table 3 shows the $F_{BOX}$ of the estimated bounding boxes and Table 4 the distances $D_{BOX}$ of the estimated bounding box centroids from their ground truth positions. The TFOS+ algorithm resulted in the highest average scores on both metrics for most of the test image sequences, followed by the TFOS method. More specifically, for TFOS+ $F_{BOX} = 0.94$ and $D_{BOX} = 9$ pixels, compared to $F_{BOX} = 0.92$ and $D_{BOX} = 14$ pixels, respectively, achieved by TFOS. In comparison, the EMT tracker achieves $F_{BOX} = 0.78$ and $D_{BOX} = 32$ pixels, while the integration with segmentation resulting BBT outperforms it with $F_{BOX} = 0.86$ and $D_{BOX} = 22$. The TLD and IVT trackers scored lower in the $F_{BOX}$ (0.82 and 0.84, respectively) and $D_{BOX}$ (21 and 27, respectively) metrics. MST scores are not taken into consideration given the poor performance and low scores based on the $C_{BOX}$ criterion in 2.

Interestingly, the five highest average scores are achieved by TFOS+, TFOS, HBT, SPT and BBT methods that combine a rough or fine segmentation of the object area to accurately update the appearance models of the tracked objects. This observation further supports our proposed concept for integrating tracking and segmentation in a tight coupling to enhance performance of each other.

4.3.3. Object segmentation

The quantitative assessment of the segmentation performance of the proposed algorithms had two distinct goals (a) to assess quantitatively the performance of the Random Walker-based image segmentation algorithm [26] used in the proposed framework as a function of its parameters and (b) to compare the overall segmentation performance of TFOS and TFOS+ to existing, state of the art algorithms.

4.3.3.1. Performance of the Random Walker-based segmentation as a function of its parameters. Experiments were conducted with TFOS for a variety of parameter configurations of the Random Walker-based segmentation algorithm. The Grasp-1 and Grasp-2 image sequences were employed in this assessment. The influence of the parameters $\beta$ and $\gamma$ on the segmentation performance is assessed, based on all three variants of the Random Walker method formulation, as described in Section 3.2.5.

We initially assessed the performance of the variant utilizing only seed points ($\gamma = 0.0$). The $F_{SEG}$ metric is computed for various values of $\beta$ in $[1, 20, 50, 100, 200]$. The scores in Table 5 demonstrate the overall high performance of the proposed framework that exceeds 0.90 for all tested $\beta$ values. The results in this table also show that $F_{SEG}$ remains practically constant for a particular sequence, despite the large variations of $\beta$. By setting $\gamma = 0.05$, prior information is introduced in

Table 4

<table>
<thead>
<tr>
<th>Sequence</th>
<th>IVT</th>
<th>TLD</th>
<th>SPT</th>
<th>HBT</th>
<th>TFOs</th>
<th>TFOS+</th>
<th>EMT</th>
<th>MST</th>
<th>BBT</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand-1</td>
<td>25</td>
<td>32</td>
<td>16</td>
<td>13</td>
<td>13</td>
<td>31</td>
<td>34</td>
<td>37</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Hand-2</td>
<td>56</td>
<td>70</td>
<td>28</td>
<td>14</td>
<td>5</td>
<td>48</td>
<td>46</td>
<td>31</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>Grasp-1</td>
<td>31</td>
<td>33</td>
<td>12</td>
<td>10</td>
<td>2</td>
<td>3</td>
<td>37</td>
<td>16</td>
<td>30</td>
<td>19</td>
</tr>
<tr>
<td>Grasp-2</td>
<td>22</td>
<td>17</td>
<td>26</td>
<td>6</td>
<td>43</td>
<td>24</td>
<td>8</td>
<td>19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>John</td>
<td>19</td>
<td>40</td>
<td>38</td>
<td>4</td>
<td>42</td>
<td>3</td>
<td>13</td>
<td>7</td>
<td>41</td>
<td>23</td>
</tr>
<tr>
<td>Anto</td>
<td>5</td>
<td>22</td>
<td>6</td>
<td>14</td>
<td>28</td>
<td>23</td>
<td>41</td>
<td>10</td>
<td>26</td>
<td>19</td>
</tr>
<tr>
<td>Foreman</td>
<td>12</td>
<td>30</td>
<td>19</td>
<td>18</td>
<td>6</td>
<td>6</td>
<td>30</td>
<td>19</td>
<td>13</td>
<td>17</td>
</tr>
<tr>
<td>Fish</td>
<td>10</td>
<td>12</td>
<td>20</td>
<td>16</td>
<td>11</td>
<td>6</td>
<td>21</td>
<td>22</td>
<td>19</td>
<td>15</td>
</tr>
<tr>
<td>Book-1</td>
<td>22</td>
<td>30</td>
<td>23</td>
<td>20</td>
<td>6</td>
<td>3</td>
<td>42</td>
<td>10</td>
<td>24</td>
<td>20</td>
</tr>
<tr>
<td>Book-2</td>
<td>30</td>
<td>5</td>
<td>226</td>
<td>28</td>
<td>10</td>
<td>33</td>
<td>6</td>
<td>33</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Cliffs</td>
<td>11</td>
<td>18</td>
<td>14</td>
<td>15</td>
<td>13</td>
<td>14</td>
<td>12</td>
<td>15</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>Caterpillar</td>
<td>10</td>
<td>11</td>
<td>19</td>
<td>5</td>
<td>5</td>
<td>34</td>
<td>6</td>
<td>7</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Mean $D_{BOX}$</td>
<td>27</td>
<td>27</td>
<td>20</td>
<td>15</td>
<td>14</td>
<td>32</td>
<td>18</td>
<td>22</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Table 5

<table>
<thead>
<tr>
<th>$F_{BOX}$</th>
<th>Grasp-1</th>
<th>Grasp-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta = 1$</td>
<td>$\gamma = 0.0$</td>
<td>$\gamma = 0.05$</td>
</tr>
<tr>
<td>$\beta = 20$</td>
<td>$\gamma = 0.90$</td>
<td>$\gamma = 0.988$</td>
</tr>
<tr>
<td>$\beta = 50$</td>
<td>$\gamma = 0.910$</td>
<td>$\gamma = 0.907$</td>
</tr>
</tbody>
</table>

4.3.3.2. Performance of Random Walker-based segmentation as a function of its parameters. The segmentation performance of the proposed framework that exceeds 0.90 for all tested $\beta$ values. The results in this table also show that $F_{SEG}$ remains practically constant for a particular sequence, despite the large variations of $\beta$. By setting $\gamma = 0.05$, prior information is introduced in

$F_{BOX}$ of the estimated bounding boxes and Table 4 the distances $D_{BOX}$ of the estimated bounding box centroids from their ground truth positions. The TFOS+ algorithm resulted in the highest average scores on both metrics for most of the test image sequences, followed by the TFOs method. More specifically, for $F_{BOX} = 0.94$ and $D_{BOX} = 9$ pixels, compared to $F_{BOX} = 0.92$ and $D_{BOX} = 14$ pixels, respectively, achieved by TFOS. In comparison, the EMT tracker achieves $F_{BOX} = 0.78$ and $D_{BOX} = 32$ pixels, while the integration with segmentation resulting BBT outperforms it with $F_{BOX} = 0.86$ and $D_{BOX} = 22$. The TLD and IVT trackers scored lower in the $F_{BOX}$ (0.82 and 0.84, respectively) and $D_{BOX}$ (21 and 27, respectively) metrics. MST scores are not taken into consideration given the poor performance and low scores based on the $C_{BOX}$ criterion in 2.

Interestingly, the five highest average scores are achieved by TFOS+, TFOS, HBT, SPT and BBT methods that combine a rough or fine segmentation of the object area to accurately update the appearance models of the tracked objects. This observation further supports our proposed concept for integrating tracking and segmentation in a tight coupling to enhance performance of each other.

The quantitative assessment of the segmentation performance of the proposed algorithms had two distinct goals (a) to assess quantitatively the performance of the Random Walker-based image segmentation algorithm [26] used in the proposed framework as a function of its parameters and (b) to compare the overall segmentation performance of TFOS and TFOS+ to existing, state of the art algorithms.

4.3.3.1. Performance of the Random Walker-based segmentation as a function of its parameters. Experiments were conducted with TFOS for a variety of parameter configurations of the Random Walker-based segmentation algorithm. The Grasp-1 and Grasp-2 image sequences were employed in this assessment. The influence of the parameters $\beta$ and $\gamma$ on the segmentation performance is assessed, based on all three variants of the Random Walker method formulation, as described in Section 3.2.5.

We initially assessed the performance of the variant utilizing only seed points ($\gamma = 0.0$). The $F_{SEG}$ metric is computed for various values of $\beta$ in $[1, 20, 50, 100, 200]$. The scores in Table 5 demonstrate the overall high performance of the proposed framework that exceeds 0.90 for all tested $\beta$ values. The results in this table also show that $F_{SEG}$ remains practically constant for a particular sequence, despite the large variations of $\beta$. By setting $\gamma = 0.05$, prior information is introduced in
segmentation besides the seed points. Table 5 provides the obtained scores for $\gamma = 0.05$ for both test image sequences. The highest score is obtained for values of $\beta = 20$ and $\gamma = 0.05$, whereas the scores for the rest of the configurations are notably lower but still exceed 0.90.

Next, keeping the value $\beta = 20$, we assess in more detail the incorporation of prior information by running experiments with varying $\gamma$. Table 6 provides the resulting scores while $\gamma$ ranges within $[0...0.5]$. The highest $F_{SEC}$ scores of 0.988 is achieved for $\gamma = 0.05$, for the Grasp-1 test sequence, whereas 0.940 is achieved for $\gamma = 0.005$ in the Grasp-2 sequence. The same table also shows the $F_{SEC}$ scores for various values of $\gamma$ and for $\beta = 1.0$. As it can be observed, the performance is, on average, unaffected for Grasp-1 and degrades by a factor of approximately 5% for Grasp-2.

Finally, for each test image sequence, we performed experiments with using priors, only. The performance of all three variants for the two sequences is summarized in Table 7. The reported parameters are those maximizing the obtained $F_{SEC}$ scores.

Although all three variants perform satisfactorily, the usage of both seeds and priors improves the segmentation performance in both test image sequences.

4.3.2.2. TFOS vs object-based video segmentation. The final part of the quantitative evaluation regards the performance of TFOS, TFOS+ and its BBT variant with two state of the art methods performing visual object tracking combined with fine segmentation and video segmentation, respectively.

The first existing method to compare with is the Video-SnapCut (VSC) algorithm for video segmentation [7]. The algorithm is available in Adobe After Effects (AE) CSS, as the RotoBrush tool/effect [49]. Concerning the experimental results extracted by this tool, an initial foreground annotation was introduced by the user, forming the accurate area of the object that is to be tracked. The procedure was then initiated to compute the segmentation results for each image throughout the sequence with no additional user interaction.

We also compared our approach to the (HBT) algorithm [22]. It is a bounding box-based tracking-by-detection method, which also computes binary masks of the tracked object’s shape. A rough foreground/background separation is performed by the method for each frame, using the GrabCut [43] segmentation algorithm. The ground truth bounding box of the object in the first frame is required to initialize the method for each test image sequence rather than an accurate binary mask like the rest of the compared methods.

The BBT method consists of the same parts as the proposed algorithms except the segmentation part, where we exploit the Bayesian fusion of image cues computed by Eq. (8) to perform binary pixel-wise classification on image pixels based on a single threshold, separating the target object and its background. We manually set the threshold value within a range [0.5–0.7] to obtain the best overall result for each test image sequence.

The evaluation metrics that were utilized to assess the segmentation performance of each algorithm are those presented in Section 4.3.1. The results are reported in Table 8. As it can be verified, VSC results in ideal scores, with TFOS+ following. The proposed methods clearly outperform BBT.

Secondly, the $F_{SEC}$ value is calculated for each frame over the estimated and the ground truth binary masks. The mean $F_{SEC}$ is finally computed for each image sequence, only for the successfully segmented frames. We note that the $t_{HBT}$, $VSC$ and $BBT$ compared algorithms were carefully adjusted to the best parameter configuration in order to maximize the selected evaluation metrics for each test image sequence.

The results are listed in Table 9. The VSC algorithm succeeds the highest $F_{SEC}$ scores in four of the sequences (Hand-1, Anto, Foreman, Caterpillar) and the second highest average score. It reports lower $F_{SEC}$ scores than the proposed methods for sequences containing abrupt illumination changes and strong object shape deformations (Hand-1, Grasp-1 and Book-1,2).

The TFOS method does not perform satisfactorily for sequences containing abrupt scale object changes and/or highly overlapping object/background colors (John, Anto, Fish, Clif-dive). However, its performance is already higher by 8% in average compared to the basic BBT method for pixel-wise classification based on Bayesian-

<table>
<thead>
<tr>
<th>Table 6</th>
<th>$F_{SEC}$ scores of the Random Walker-based segmentation using different values of $\gamma$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{SEC}$</td>
<td>Grasp-1</td>
</tr>
<tr>
<td>$\beta = 1.0$</td>
<td>$\beta = 20.0$</td>
</tr>
<tr>
<td>$\gamma = 0.500$</td>
<td>0.910</td>
</tr>
<tr>
<td>$\gamma = 0.250$</td>
<td>0.930</td>
</tr>
<tr>
<td>$\gamma = 0.050$</td>
<td>0.910</td>
</tr>
<tr>
<td>$\gamma = 0.025$</td>
<td>0.900</td>
</tr>
<tr>
<td>$\gamma = 0.005$</td>
<td>0.900</td>
</tr>
<tr>
<td>$\gamma = 0.000$ (seeds only)</td>
<td>0.900</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 7</th>
<th>Quantitative assessment of segmentation performance for the three variant formulations of the Random Walker image segmentation method on the Grasp-1 and Grasp-2 test sequences.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{SEC}$</td>
<td>Grasp-1</td>
</tr>
<tr>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>Priors, only</td>
<td>0.937</td>
</tr>
<tr>
<td>Seeds, only</td>
<td>0.932</td>
</tr>
<tr>
<td>Seeds &amp; priors</td>
<td>0.992</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 8</th>
<th>Object-based video segmentation results. Percentage of frames ($C_{SEC}$) correctly segmented by each method (see text for details).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>TFOS</td>
</tr>
<tr>
<td>Hand-1</td>
<td>100</td>
</tr>
<tr>
<td>Hand-2</td>
<td>100</td>
</tr>
<tr>
<td>Grasp-1</td>
<td>100</td>
</tr>
<tr>
<td>Grasp-2</td>
<td>100</td>
</tr>
<tr>
<td>John</td>
<td>16</td>
</tr>
<tr>
<td>Anto</td>
<td>23</td>
</tr>
<tr>
<td>Foreman</td>
<td>100</td>
</tr>
<tr>
<td>Fish</td>
<td>100</td>
</tr>
<tr>
<td>Book-1</td>
<td>100</td>
</tr>
<tr>
<td>Book-2</td>
<td>100</td>
</tr>
<tr>
<td>Cliff-dive</td>
<td>74</td>
</tr>
<tr>
<td>Caterpillar</td>
<td>100</td>
</tr>
<tr>
<td>Mean $C_{SEC}$ (%)</td>
<td>83</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 9</th>
<th>Object-based video segmentation results. The mean $F_{SEC}$ is reported for each test image sequence on the successfully segmented frames by each method, calculated between the estimated and the ground truth segmentation masks.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>TFOS</td>
</tr>
<tr>
<td>Hand-1</td>
<td>0.922</td>
</tr>
<tr>
<td>Hand-2</td>
<td>(0.941)</td>
</tr>
<tr>
<td>Grasp-1</td>
<td>(0.989)</td>
</tr>
<tr>
<td>Grasp-2</td>
<td>0.974</td>
</tr>
<tr>
<td>John</td>
<td>0.734</td>
</tr>
<tr>
<td>Anto</td>
<td>0.571</td>
</tr>
<tr>
<td>Foreman</td>
<td>0.925</td>
</tr>
<tr>
<td>Fish</td>
<td>0.896</td>
</tr>
<tr>
<td>Book-1</td>
<td>0.915</td>
</tr>
<tr>
<td>Book-2</td>
<td>(0.880)</td>
</tr>
<tr>
<td>Cliff-dive</td>
<td>0.747</td>
</tr>
<tr>
<td>Caterpillar</td>
<td>0.867</td>
</tr>
<tr>
<td>Mean $F_{SEC}$</td>
<td>0.863</td>
</tr>
</tbody>
</table>
based binary thresholding technique. Furthermore, TFOS + enhances the performance in the aforementioned sequences, providing more accurate segmentation of the target objects near 90%.

4.4. Qualitative results

A qualitative assessment of the tracking and segmentation performance of the proposed algorithms was also performed. Initially, we compare TFOS with EMT [60], in order to demonstrate the effectiveness and the key role of our methodology towards robust and drift-free tracking. Fig. 13 illustrates representative snapshots of the comparative tracking results for some of the test image sequences. The performance of the stand-alone EMT tracking method degrades in the presence of extensive variations of the colors and shape of the object. TFOS provides stable tracking and adaptation of the tracking kernel size to object’s shape changes exploiting the noise-free color and shape information provided by the segmentation phase of the previous frame.

Additional qualitative results regard representative snapshots obtained by the proposed methods and the Video SnapCut method [7]. They are presented in Fig. 14 for eight of the test image sequences. A single frame is shown for some of the image sequences in each row. The ground truth object mask and the segmentation result from the compared algorithms are presented in each column. Video material resulting from this qualitative evaluation is also available online.6

4.5. Computational performance issues

The EMT method performs in real-time. The implementation of the Random Walker-based image segmentation method is based on the Graph Analysis Toolbox of Matlab that is available online.7 The most computationally demanding part of the proposed methodology is the solution of the large system of the sparse linear equations of the Random Walker formulation. The run-time performance of the current Matlab implementation varies between 2 to 4 s per 640 × 480 frame. The reported experiments were conducted on a PC equipped with an Intel i7 CPU and 4 GB of RAM memory. However, a near real-time performance should be feasible with code optimization and GPU implementation.

5. Summary and future work

We presented a new approach to the problem of model-free, joint segmentation and tracking of a potentially non-rigid object in a monocular video. The location and the shape of the target object in the first frame of an image sequence is the only required input. As such, the proposed methodology addresses several challenges with respect to tracking that are associated to the appearance and shape of the tracked object, the motion of the camera and the scene illumination conditions. It also aspires to increase target localization performance through fine object segmentation.

---

6 http://users.ics.forth.gr/argyros/research/trackingsegmentation.html
7 http://cns.bu.edu/~lgrady/software.html
The key contribution of the proposed framework is the efficient integration of EM-based object tracking and Random Walker-based image segmentation in a closed loop scheme, enhancing their individual effectiveness, thus the overall performance of the methodology. Two variants of the approach (TFOS and TFOS+) have been proposed. Both of them achieve high tracking and segmentation accuracy, compared to state of the art methods. Their performance has been assessed in a series of challenging image sequences, for which ground truth data were manually extracted. The experimental results show that TFOS+ demonstrates superior overall segmentation as well as tracking performance for the test image sequences, compared to TFOS. As a general conclusion, it has been shown that explicit fine segmentation of the target object facilitates drift-free tracking.

The proposed framework provides a conceptually simple and effective workflow. Various extensions are currently being considered. As an example, motion and depth information can provide important
additional input to the existing framework. Additionally, extensions of the existing framework towards achieving multi-object segmentation and tracking would be of great theoretical and practical importance.

Acknowledgments

This work was partially supported by the IST-FP7-IP-288533 project Robohow.Cog.

References