Adaptive Object Tracking by Learning Background Context

Ali Borji* , Simone Frintrop†, Dicky N. Sihite*, and Laurent Itti*

* Department of Computer Science, University of Southern California, Los Angeles, USA
† Institute of Computer Science, Rheinische Friedrich-Wilhelms Universität Bonn, Germany

Supported by the National Science Foundation (grant number BCS-0827764), and the Army Research Office (W911NF-08-1-0360 and W911NF-11-1-0046), and U.S. Army (W81XWH-10-2-0076).

Introduction

One challenge when tracking objects is to adapt the object representation depending on the scene context to account for changes in illumination, coloring, scaling, etc.

We present a solution that is based on particle filters and component-based descriptors. We deal with changing backgrounds by using a quick training phase with user interaction at the beginning of an image sequence. During this phase, some background clusters are learned along with object representations for those clusters. Next, for the rest of the sequence the best fitting background cluster is determined for each frame and the corresponding object representation is used for tracking. Experiments show a particle filter adapting to background changes can efficiently track objects and persons in natural scenes and results in higher tracking results than the basic approach.

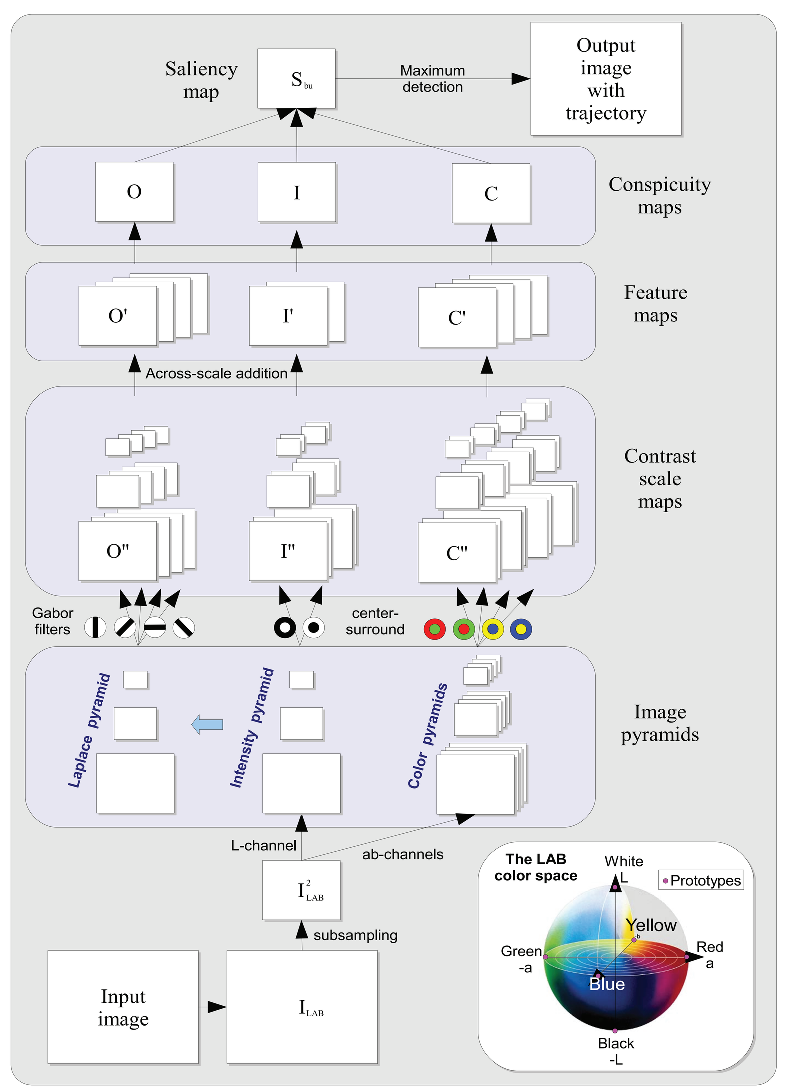
Additionally, using an object tracker to follow the main character in video games, we were able to explain a large amount of eye fixations higher than other saliency models in terms of NSS score proving that tracking is an important top-down attention component.

Saliency Modeling

The term "saliency" is often referred to visual attention where some parts of stimuli are selected for further processing.

Selection mechanism could be bottom-up where it is derived by stimuli level competitions or top-down task-relevance mechanisms based on tasks demands.

We follow a different direction than spatio-temporal saliency models by tracking a task-relevant object.



The bottom-up saliency computation of the attention system VOCUS. By Simone Frintrop

Computation of the Target Descriptor

The target descriptor consists of a collection of components that have a strong contrast within a certain feature dimension.

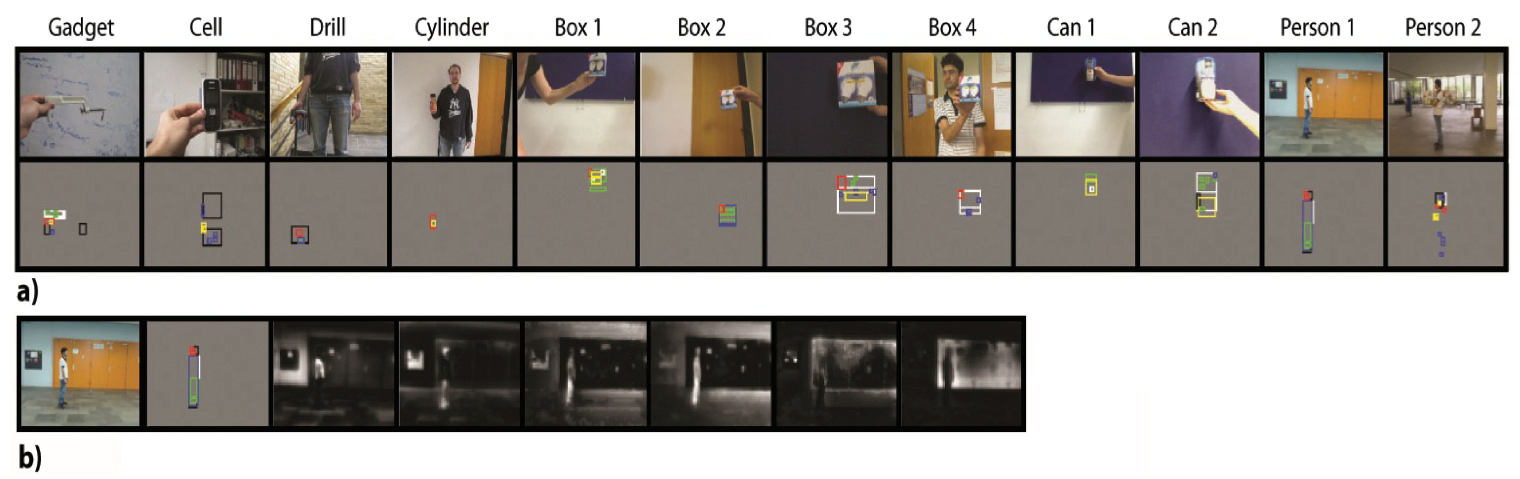
First, six feature maps F_i are computed. They represent intensity and color contrasts based on color-opponent cells of the human visual system.

Second, we compute a component-based target descriptor from the feature maps. A component is a peak in one of the feature maps within the target region.

The positions of the regions $m_{i,j}$ are stored relative to the center of $\vec{R}'$ and represent a template $\vec{M}_{R'}$.

Finally, we describe how the target descriptor $\vec{d}^*$ is matched to an image region $\vec{R}'$ of arbitrary size and dimensions.

Left: An illustration of the template $\vec{M}_{R^*}$ for the target region $\vec{R}^*$. The three colored rectangles denote the $m_{i,j}$. Note that each of them comes from a different feature map which is illustrated here by different colors.
Right: the template $\vec{M}_{R'}$ adapted to region $\vec{R}'$.



a) Used objects in our work and their corresponding component-based descriptors, b) Descriptor and feature maps F_i for Person 1. From left to right: bright-dark, dark-bright, green-red, blue-yellow, red-green and yellow-blue contrasts.

Clustering background contexts

We first learn a number of background clusters from a train image sequence and also their corresponding object descriptors which can successfully detect the object in those backgrounds.

Then over a test sequence, for each frame, first we find its background cluster and then apply the descriptor of that cluster to the frame.

For image representation, we partition the image and use the average of the feature maps:

$$\vec{e_i} = (E_i), E_i = \begin{bmatrix} F_i^{11} & F_i^{12} & \dots & F_i^{1m} \\ F_i^{21} & F_i^{22} & \dots & F_i^{2m} \\ & \dots & & \\ F_i^{n1} & F_i^{n2} & \dots & F_i^{nm} \end{bmatrix}, i = 1..6.$$

where element F^{pq} of E_i matrix is the normalized mean of $F_i((p-1)w : pw, (q-1)h : qh)$ region of map F_i and generates a row vector of matrix E_i:

$$F_i^{pq} = \frac{Avg(F_i((p-1)w : pw, (q-1)h : qh)) - m_{i1}^{pq}}{m_{i2}^{pq} - m_{i1}^{pq}}$$

$$\vec{e}_{C_k}^{\,new} = \frac{(n_{C_k^{new}} - 1)\vec{e}_{C_k}^{\,old} + \vec{e}_z}{n_{C_k^{new}}}$$

We then use BSAS algorithm to generate a number of background clusters.

Particle filter based tracking

The tracker employs the standard Condensation algorithm which maintains a set of weighted particles over time using a recursive procedure based on the following three steps:

First, the system draws particles randomly from the particle set of the previous time step, where each particle is drawn with a probability proportional to the associated weight of the particle.

$$\pi_t^j = c \cdot e^{\lambda \cdot T(\vec{d}^*, \vec{d}_t^j)},$$

weight of a particle is based on the distance of the descriptor it represents and the target template descriptor.

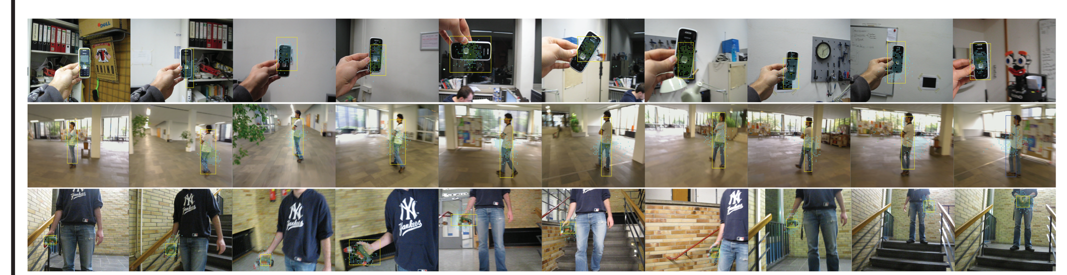
Second, the particles are transformed (predicted) according to a motion model.

Finally, all particles are assigned new weights according to an observation model and the object state is estimated.

$$\vec{x}_t = \sum_{j=1}^{J} \pi_t^j \cdot \vec{s}_t^j.$$

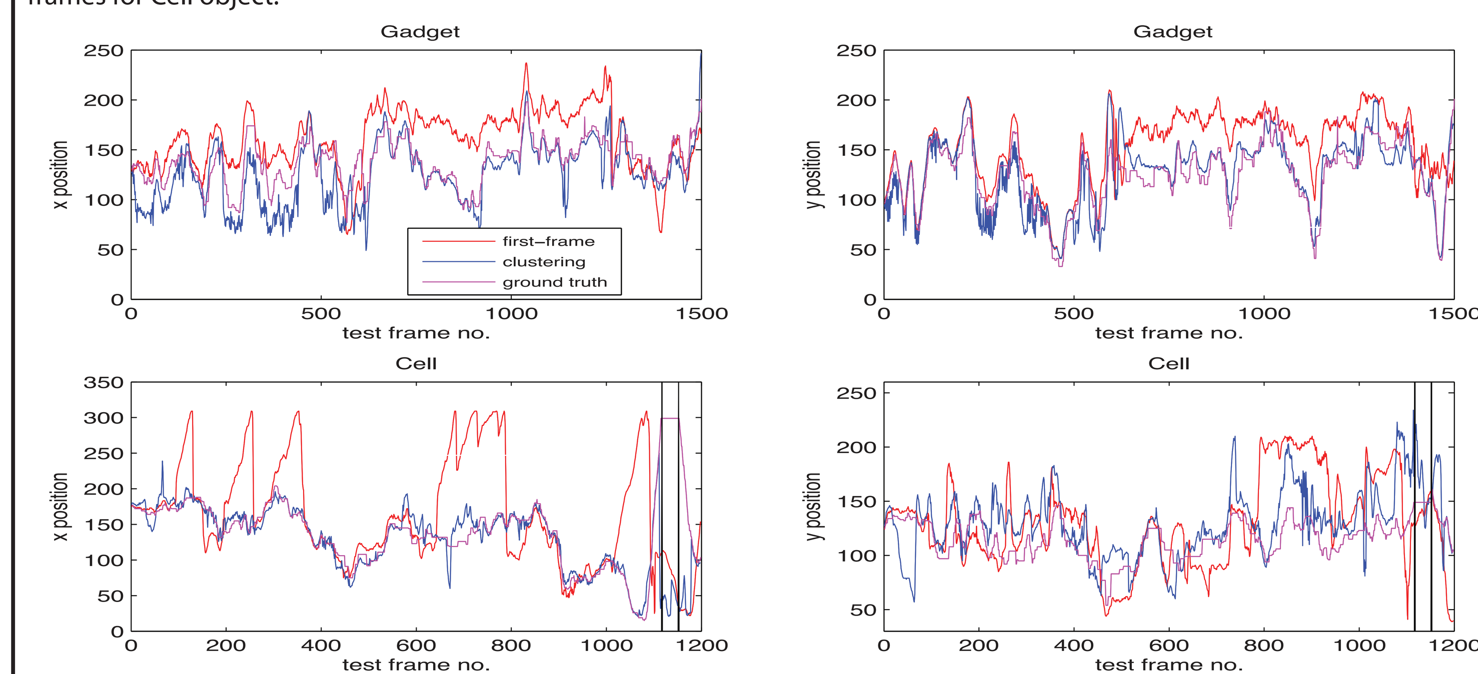
To adapt the particle tracking to account for background changes, for each frame in a sequence we find its cluster among the learned background clusters from training frames and then use the descriptor of that cluster.

Object Tracking Experiments



Sample frames from Cell, Person2 and Drill test sequences and estimated target rectangles. Green dots: particles that matched to target, cyan dots: particles that did not match. Yellow (blue) rectangles mean high (low) confidence.

Traces of object position in x and y dimensions for Gadget and Cell objects. Black vertical bars show the occluded frames for Cell object.



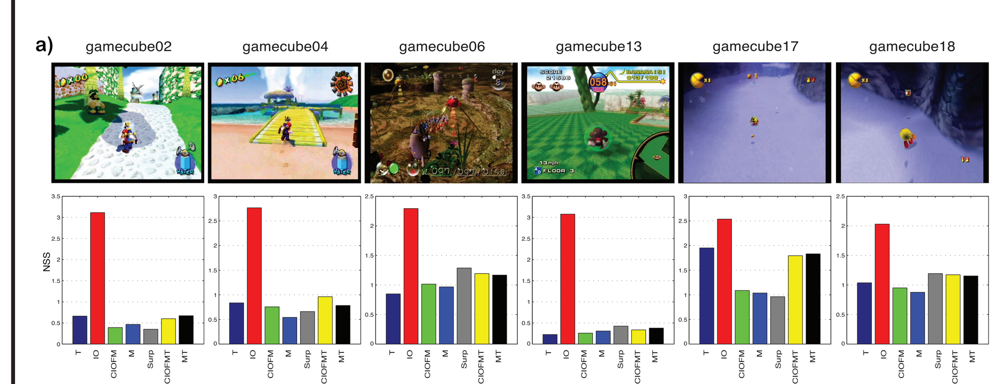
Detection results for both first-frame and clustering cases.

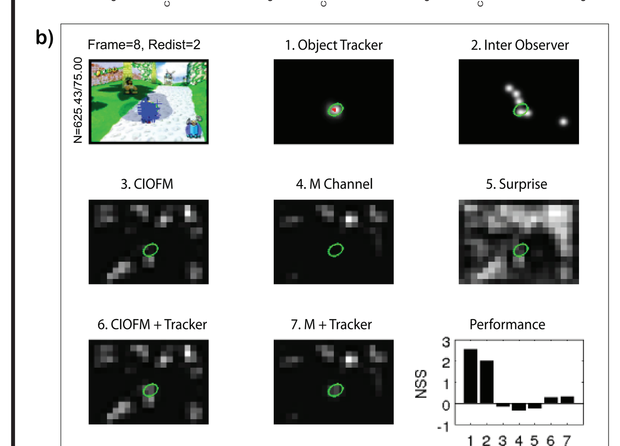
	Object	# off rames in train set	# off rames in test set	# of Clusters	Detection rate (Train)		Detection rate (Test)	
					first-frame case	clustering case	first-frame case	clustering case
1	Gadget	1200	1500	3	91.6	97.6 (6)	66.6	96.7 (30.1)
2	Cell	2400	1200	8	82.9	95.6 (12.7)	49.6	95 (45.4)
3	Drill	2400	3300	10	67.1	93.5 (26.4)	71.3	80.3 (9)
4	Cylinder	2665	2100	13	78.3	82.6 (4.3)	36.8	69.2 (32.4)
5.a	Box[1]	75	75	2	61.3	100 (38.7)	50.6	88 (37.4)
5.b	-	-	53	-	-	-	40	61.3 (21.3)
5.c	-	-	43	-	-	-	28	45.3 (17.3)
6	Box[2]	100	100	3	63	100 (37)	67	90 (23)
7	Box[3]	97	101	3	52.5	96.9 (44.4)	40.6	92 (51.4)
8	Box[4]	50	65	2	100	100 (0)	100	100 (0)
9	Can[1]	70	70	3	61.4	100 (38.6)	44.3	82.8 (38.5)
10.a	Can[2]	75	100	3	64	100 (36)	57	78 (21)
10.b	-	-	100	-	-	-	67	90 (23)
11	Person[1]	84	100	3	96.4	100 (3.6)	86	100 (14)
12	Person[2]	158	161	5	41.8	93 (51.2)	87.5	92.5 (5)

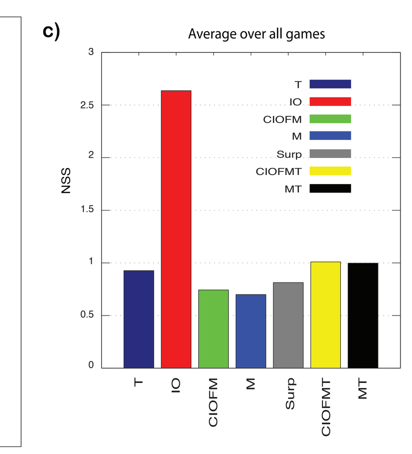
Detection rate (percentage of frames with the object correctly detected) and detection enhancement rate (in parentheses). In both train and test cases except the Box3 (since it was a small easy case and without large changes in background) we observed an increase in detection rate of clustering compared to the first-frame case. An object is considered as detected if the center of the rectangle M^o proposed by the tracker is on the manually tagged target region M^t.

Saliency Modeling Experiments

a) Sample frames from 6 game stimuli used in the experiments: Super Mario Sunshine (left two), Pikmin, Super Monkey ball, PacMan World (last two). Below each frame is the average NSS score over 1668, 1082, 2483, 687, 1863, and 1548 frames, respectively for several models.







b) A sample frame of Mario Sunshine game with particles overlaid. Sample saliency maps of models are also shown. The panel at the bottom-right is the instantaneous NSS score for this frame. Since subjects did not agree much in this frame NSS score for the IO model is smaller than Tracking model. NSS scores for CIOFM, M and Surprise are negative indicating that bottom-up salient stimuli do not capture task-relevant attention, however when adding saliency map of Tracking model to this models NSS score increased to above 0.

c) Average NSS score over all six games. As it shows CIOFM + Tracking model achieved the best score followed by Motion + Tracking. Tracking alone is higher than other pure bottom-up saliency models indicating that subjects most of the time tracked the main character in these games. There is still a big difference in performance of models and Inter-Observer model (more than 1.5 difference in NSS score).