SCAR: Dynamic adaptation for person detection and persistence analysis in unconstrained videos

George Kamberov, Matt Burlick, Lazaros Karydas, and Olga Koteoglou

Department of Computer Science
Stevens Institute of Technology
Hoboken, NJ 07030, USA
gkambero,mburlick,lkarydas,okoteogl@stevens.edu *

Abstract. In many forensic and data analytics applications there is a need to detect whether and for how long a specific person is present in a video. Frames in which the person cannot be recognized by state of the art engines are of particular importance. We describe a new framework for detection and persistence analysis in noisy and cluttered videos. It combines a new approach to tagging individuals with dynamic person-specific tags, occlusion resolution, and contact reacquisition. To assure that the tagging is robust to occlusions and partial visibility the tags are built from small pieces of the face surface. To account for the wide and unpredictable ranges of pose and appearance variations and environmental and illumination clutter the tags are continuously and automatically updated by local incremental learning of the object's background and foreground.

1 Introduction

In this work address a basic persistence analysis task in understanding unconstrained videos: detect if a specific person is present and estimate the length of her presence. The objective is to detect the faces of specific individuals and to retrieve as many as possible of the frames containing at least a part of the individuals' faces. Unconstrained videos often involve illumination and environment clutter, frequent and hard to analyze shot transitions (camera motions and zooms, stitching together multiple camera views and scenes), noisy low quality videos, and lossy compression. In such videos there are many frames where only small and often featureless parts of the face are visible. We do not need to recover the full face, in fact it is often impossible to do that. However, we do need to be able to keep track of the face in the presence of significant occlusions, image deterioration and environmental and scene clutter. The key of the proposed approach is to develop dynamic tags, see Figure 1, that can be used to verify the subjects' presence. Persistence analysis is an important task combining recognition and tracking but, to our knowledge, there is no published work addressing it specifically. The state of the art in face detection, recognition, and tracking provides useful insights and tools but appears inadequate to our persistence analysis task. In a set of experiments reported in Section 3 we evaluated all the state of the art (SOA) tracking systems for which the authors have provided publicly available code: [1,2,4,7,9]. The evaluation shows that the

* This research was partially supported by: the National Science Foundation, Award # 0916610; two Gifts from the Gerondelis Foundation; the Robert Crooks Stanley Fellowship Fund.

Fig. 1. Automatic detection, pose variations, camera motions, exiting and re-entering and occlusions of the tagged persons. Yellow tags indicate initialization/on-line relearning of a new person-specific tag mask and (local) appearance model. Green tags indicate that the persons tag was updated through mean shift. Frame 6: Initial recognition of person 0. Frame 52: Initialization of person 1, continued tracking of person 0. Frame 95: Continued persistence of person 0, a new foreground model and a tag for person 1 is re-acquired upon re-entry in the field of view (FOV) – SCAR detected that person 1 exited the FOV in video frame 71. Frames 138 and 202: pose variations and partial occlusion of person 1. Frame 208: re-initialization (new appearance model) for person 1 while person 0 is still present.

state-of-the-art trackers are struggling when tracking faces in unconstrained videos with difficult backgrounds, fast moving characters, occlusions, exits and re-entries, and camera motion. We also ran the OpenCV CAMSHIFT: not surprisingly on unconstrained videos it is by far the worst performer. The adaptive background enhancement reported in [9] at least gives results in the same ball-park as the rest of the SOA trackers. See Table 1 and Table 2. In addition, we evaluated a baseline method *Picasa BL* for "detection, recognition, and tracking by per-frame detection and recognition". Picasa BL is a straight forward approach based on submitting to Google's Picasa the frames in each video and then considering each set of recognized people's faces as a track. One would have expected that the baseline method should have been severely disadvantaged since it does not use at all the time coherency or the spatial-temporal relationships inherent in video data. The experiments show that this is not the case in videos with substantial clutter (See Table 2.). Still, as Table 1 indicates tracking by per-frame recognition also fails at very high rates on videos involving occlusions and pose and illumination variations. The performance of SOA trackers can be improved sometimes by careful per-video manual tuning. This is not an option in persistence analysis of real world forensics collections of video snippets and collages whose volumes often exceed tera

bytes of data. The evaluation results and the need to develop self-tuning methods show that a new approach is needed to tackle persistence analysis in un-constrained videos.

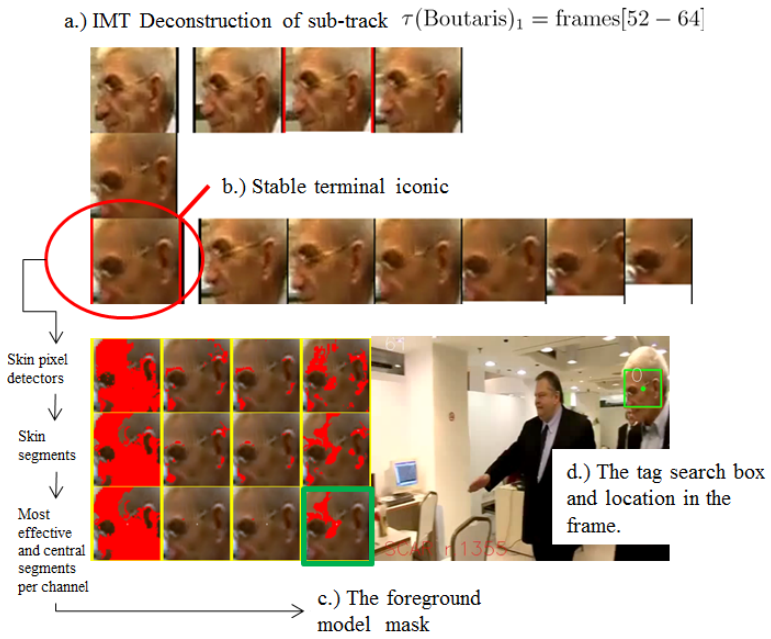

Fig. 2. The IMT re-initialization pipeline: a.) History decomposition into initial, mature, and terminal tracklets; b.) Stable terminal iconic used for the re-initialization model; c.) The mask used for building the re-initialization foreground model; d.) The tag location and its search box in the iconic's frame.

2 The SCAR Algorithm

We propose a new approach based on the following novel ideas and observations: **(1.)** Use small, informative, central descriptors: that is, build descriptors from small informative central patches instead of elaborate representations encapsulating the whole object and its surrounding supporters/distractors. See Figure 2 c.) and d.). Using small central patches makes the tracking more robust to occlusions and also to the effects of camera motions. **(2.)** Use just enough context: couple the foreground object descriptor patch with some background context but use as little as possible context to flesh out the object while encompassing only a few possible distracters. See Figure 5. This approach is also aimed at minimizing the probability of occlusion. **(3.)** Detect when the current foreground model is not adequate or when a contact is occluded or exits from the FOV. **(4.)** Contact re-acquisition is a hard cognitive task and it is handled by either: (i) Detection and recognition using recognition memory and a high level recognition engine (we

use a plug-in face detection and recognition engine); (ii) If the recognition fails because the engine is not trained to handle the hard poses, partial views, or illumination clutter, then engage a secondary mechanism called IMT-based re-initialization which looks at the track up to date and extracts a probable foreground model. See Figure 2. **(5.)** Use an information theoretic approach to handle occlusion resolution even in the absence of scene depth and elaborate scene context and semantics models.

The proposed method applies to persistence analysis of arbitrary objects. The only place where we specialize to faces is where we select the actual masks (for intialization/reintialization) for building the foreground models. Different color models can be used but for faces we will use 1D Hue.

2.1 Dynamic Tags and Persistence Analysis

In our framework, in each frame f the tag of an object o is built from an appearance model for the object and an axis aligned square bounding an object segment. The tag is marked on the video frame as the center of the axis aligned bounding square. The appearance model is represented as a class conditional distribution. Following [3] we call the tag square the search box and denote by $P(C|o;f)$ the class conditional distribution representing the probability for any pixel color C given that the pixel belongs to the object segment enclosed by the search box in this frame. To study the persistence of an object (person) we build an initial tag from a crop and then update the tag, the search box, and the supporting foreground appearance model dynamically.

Without loss of generality we may assume that for each object o the class conditional distribution $P(C|o;f)$ is locally stable with respect to time. Therefore, the sequence of video frames $f_1,\dots,f_T$ is broken into consecutive intervals of adjacent frames $\tau(o)_1,\dots,\tau(o)_{m(o)}$, $m(o)\leq T$, so that

$$P(C|o;f)\approx \text{const}_j, \forall f\in\tau(o)_j, j=1,\dots,m(o). \tag{1}$$

In fact this is always true, even for the most violent and unstable appearances in which case each subinterval will contain a single frame. The fundamental observation that underlines our method is that within a CAMSHIFT framework we can estimate this temporal partition of the video for each object. (See Figure 3.) Indeed for each frame f and object tag o let $P(o;f)=1/\,r(o;f)$ be the prior probability that a pixel belongs to the object and so let $P(b;f)=(r(o;f)-1)/\,r(o;f)$ be the prior probability that a pixel belongs to the background. In order to deal successfully with both background changes and object appearance changes we will have to estimate $r(o;f)$ for each frame and for each object. In the original CAMSHIFT both priors were postulated to be constant and equal to 0.5. As the authors point out this is the least informed choice and it is not surprising that it can not account for most object appearance changes and environment clutter. As observed in [9] the selection of $r(o;f)$ should be connected to the relative sizes of the area

$$s(o;f)=\text{Area of the search box}(o;f) \tag{2}$$

and the area $\beta(o;f)$ of the *calculation region* for the object in the frame, [3]. The calculation region is an axes-aligned bounding box where the object is supposed to

reside. It is centered at the center of the selection box but is slightly larger so that it encompasses some background pixels. See Figure 5. Thus following [9] we define

$$r(o;f) = \frac{\beta(o;f)}{s(o;f)} > 1. \tag{3}$$

The video partition can be chosen so that $r(o;f)$ is locally constant (in time). That is, both (1) and

$$r(o;f) = \text{const}_j^*, \forall f \in \tau(o)_j, j = 1,\ldots,m(o) \tag{4}$$

hold.

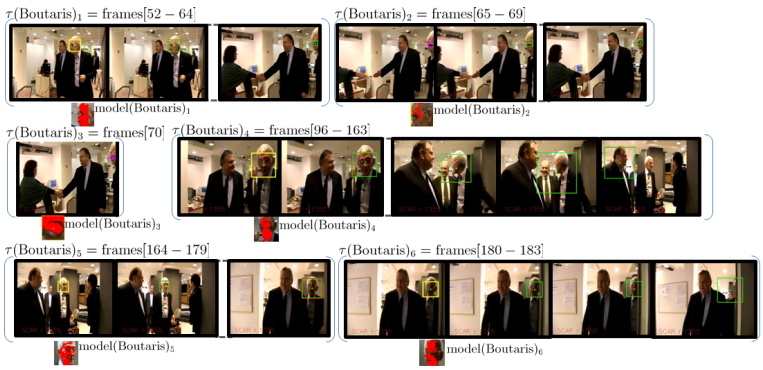

Fig. 3. Detecting and tagging of a contact (Yiannis Boutaris) in a 184 frames video. The contacts' search boxes are color coded to distinguish model initialization/reinitialization (yellow and using recognition, pink when using the IMT mechanism) vs CAMSHIFT with background adaptation (green). $\tau(\text{Boutaris})_1$: SCAR automatically builds an initial tag based on a face crop produced by Picasa in frame 52; it then dynamically updates the tag until frame 64 by mean shifting with a continuous background adaptation. $\tau(\text{Boutaris})_2$ and $\tau(\text{Boutaris})_3$: New IMT based foreground models are built in frame 65 and then again in frame 70 to cope with the progressive occlusion as the contact exits camera FOV. SCAR detects the contact's ultimate exit from the FOV in frame 71. $\tau(\text{Boutaris})_4$: The contact is re-acquired in frame 96 after it re-enters the FOV; $\tau(\text{Boutaris})_5$ and $\tau(\text{Boutaris})_6$: new foreground models are reacquired as needed to account for the pose variations and the severe visibility occlusions. The appearance model for Yiannis Boutaris for the time interval $\tau(\text{Boutaris})_i$ is built using the crop and the mask shown in $\text{model}(\text{Boutaris})_i$, $i-1,\ldots,6$.

The basic steps in the SCAR framework are illustrated in Figure 3. They are: (i) generation of temporally stable foreground models and initial tags from image crops (See Section 2.2.); (ii) Tag and foreground models dynamic updates including: (ii.a) tag adaptation using the temporally stable foreground models and locally constant object priors and mean shifting with background adaptation, (ii.b) re-initialization and contact reacquisition (See Section 2.3.); (iii) Handling contact overlaps. (See Section 2.4.)

2.2 Initial Tag Selection

To obtain a tag from a facial crop we combine appearance models (skin models in the case of faces, using the skin detectors introduced in [5]), fragment-based shape cues; and centrality cues expressed in the weights of the fragments. To obtain the fragments encoding the shape of the person's facial area detected in a given frame we need to segment the image crop enclosing the facial area. The shape fragments are just the segments of the segmented facial area. So we will refer to fragments as segments. In a unconstrained video the facial crop can cover only small and/or severely occluded parts of the face. Further complications can be caused by environmental clutter, compression, and abnormal viewing angles. To deal with segmentation bias we use an automatically selected approximation of the central segmentation of the crop. The central segmentation (CS) was introduced in [10]. By design this is the segmentation whose entropy equals the mean of the entropies of all possible segmentations. It is obtained through gradient descend. In practice we can only compute approximations of the CS by following a fixed number of steps down the gradient descent path. Instead of manually selecting thresholds to stop the gradient descend we chose as approximation the segmentation whose most effective segments explain most of the content of the crop. To do this we measure the effectiveness $p(\sigma, S)$ of a fragment σ in a segmentation S of a crop $\mathcal{C}$ as the relative area of the fragment. Thus

$$p(\sigma, S) = \text{Area}(\sigma)/\,\text{Area}(\mathcal{C}).$$

Under the single hypothesis that the crop *does enclose just the object of interest* the quality of the segmentation is measured by its *effective number of different fragments*:

$$R(S) = \text{int}(1/\sum_{\sigma\in S} p(\sigma, S)^2).$$

To encode the shape of the object (facial area) enclosed by a crop we chose an approximate central segmentation S whose top $R(S)$ effective segments cover most of the crop area.

Once an effective approximate segmentation is chosen we can proceed to build a facial tag by ranking segments according to their effectiveness and centrality. In particular, the weight of a fragment σ, is $\omega(\sigma) \propto e^{-\text{dist}(\sigma,P)}\,\text{Area}(\sigma)$, where P is the centroid of the object or object tag crop. See Figure 4.

The crops are selected by one of two mechanisms – recognition when possible, or the memory-based IMT mechanism (see Figure 2).

2.3 Foreground Models and Tag Dynamics

The first question we should address is how do we recognize a good tag initialization – i.e., a mask identifying a selected set of pixels that belong to the object o. Without loss of generality we may assume that this has happened at frame 0. Can we now mean-shift to the next frame? If the initialization is good, then we should be able to select a tight calculation box around the mask which also contains some background pixels, otherwise it is known that a mean shift algorithm will fall in an unstable mode. (The

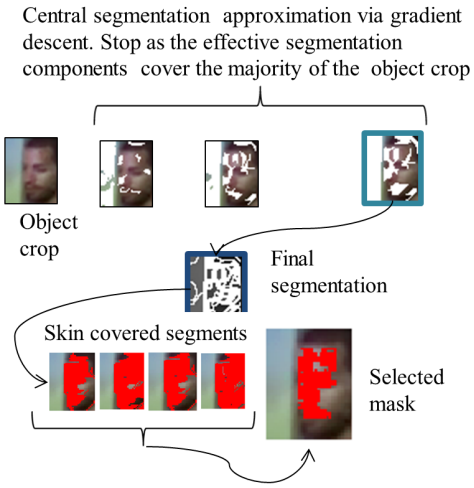

Fig. 4. Automatic mask generation combining appearance models (skin models in the case of faces), and fragment-based shape cues and centrality cues expressed in the weights of the segments. The final tag is a minimal axis aligned square bounding box of the mask.

extreme case is when the object is flat color and takes the whole image.) Thus if the initialization is good, one must be able to choose a calculation region such that the object and the objects background distributions are different $P(C|o;0)$ and $P(C|b;0)$, that is

$$\text{dist}(P(C|o;0), P(C|b;0)) > 0. \tag{5}$$

Theoretically it does not matter what metric is used in Equation (5). Recall that for a chosen $r = r(o;0) > 1$, that is, a chosen calculation region, we have

$$\begin{aligned} P(C;0) &= P(C|o;0)P(o;0) + P(C|b;0)P(b;0) \\ &= \frac{P(C|o;0)}{r} + \frac{(r-1)P(C|b;0)}{r}. \end{aligned}$$

Thus we have,

$$P(C|b;0) = \frac{P(C;0)r - P(C|o;0)}{r-1}. \tag{6}$$

Once $r = r(o;0)$ is chosen, we compute $P(C;0)$ and $P(C|o;0)$ by computing the color histograms in the calculation window and the color histogram of the object. We reached to the criterion for a good initialization: we must be able to chose $r > 1$ such that

$$0 \leq P(C;0)r - P(C|o;0) \leq r - 1 \tag{7}$$

and (5) hold. If such an $r > 1$ cannot be chosen, we can not use this initialization to start a mean-shifter. It is informative to think how can we go about to find r and what could go wrong. The obvious procedure is, start with $r = 1$, in which case (5) fails, and keep

increasing r until the inequalities (5) and (7) hold. If the initialization is badly chosen, then this may not happen – eventually r will become so large that the calculation box will overflow the video frame.

On the other hand, suppose that the initialization is chosen well and so we can find the appropriate $r = r(o;0) > 1$. At this point we can start mean shifting, which gives us the location of the object center in the next frame. First we need to check whether the model of the object appearance is still valid. That is, we check if

$$0 \leq P(C;1)\, r(o;0) - P(C|o;0) \leq r(o;0) - 1 \tag{8}$$

holds. If it does, then frames 0 and 1 are in the time interval where both $r(o;1) = r(o;0)$ and $P(C|o;0) \approx P(C|o;1)$ and we proceed with adapting the object background as in [9]. We will discuss how to do this in Section 2.3. If this condition fails, this means that the object appearance model is no longer valid, that is the object has changed too much. Essentially either the object is not visible anymore or we must find a way to compute the new model. Next we will discuss the methods to deal with this situation. The important moral is that we have gleaned a method to diagnose substantial changes in the appearance models and that this is done without using any hand tuned thresholds, or making the assumption that we can somehow estimate and list all possible appearances of an object or person.

Updating the Foreground and Background Models: At the beginning of the processing of each new frame f and for each tracked object we must: (i) check if the object tag appearance model is still valid in the new frame; (ii) update the object tag background model.

The discussion in the previous section dealt with the case $f = 1$. In the general case we have to check if

$$0 \leq P(C;f)r - P(C|o;f-1) \leq r - 1, \tag{9}$$

where $r = r(o;f-1)$. As long as (9) holds the foreground model is valid, and following [9], the background model is automatically updated since

$$P(C|b;f) = \frac{P(C;f)r - P(C|o;f)}{r-1}.$$

The only subtlety is that to account for computational precision when verifying and adjusting the axes aligned boxes so that the calculation box contains enough background pixels we check that

$$\text{dist}(P(C|o;f), P(C|b;f)) > \tau(o),$$

where

$$\tau(o) = \text{dist}(P(C|\text{object mask};0), P(C|\text{selection box}';0)),$$

where

$$\text{selection box}' = \text{selection box}(0) \setminus \text{object mask}.$$

The object tag mask is the initial skin pixels mask. In contrast, in [9] the threshold $\tau(o)$ was hand-tuned. If (9) fails, then the object might have become invisible or we may have to re-build its appearance model, i.e., re-initialize the object.

Re-acquisition: The optimal method to re-initialize an object/person model is to have a good detection and recognition engine which will find a correct mask for the object. When this is impossible we attempt to re-initialize using the incremental knowledge acquired up to date. We perform recursive initial-mature-terminal appearance analysis (IMT) [6] using the Hellinger distance on the tracklet terminating at the frame where the object appearance model became invalid. The tracklet is split into three clusters (the initial, the mature, and the terminal clusters, respectively). And we use the stable final appearance in the terminal tracklet of the object/person to generate the re-initialization mask. The stable final appearance is obtained by selecting the centroid of stable sub-cluster of the terminal portion of the tracklet. See Figure 2.

Sometimes a tracked object/person becomes invisible. It could have left the FOV, or it could be occluded or the tracker may have lost it. Such tracks could still be re-activated as the object/person is recognized in a future frame. To reactivate tracks we keep a log of the suspended tracks and the individuals associated with them. Each future frame is checked by a plug in recognition engine and if the individual object is found, then the track is reactivated using a re-initialization mask.

2.4 Handling Overlaps

To handle target overlaps when tagging multiple objects we propose a mechanism for early detection of possible overlaps and a related mechanism for adjusting the target masks to separate the objects if possible. A possible target overlap is detected when the search area of a one tracked object o_i intersects with the calculation area of another tracked object o_j. Without actual depth ordering it is not clear who is the occluding and who is the occluded. To reduce the chance for an person-person occlusion we try to locate a sub-area of o_i that is still representative of the object appearance. The selection area of o_i is shrunk until the it does not overlap o_j's calculation region. To avoid the loss of appearance information we need to compare the total information loss as we shrink the search area and the object mask information loss. The resizing is invalid if full info loss $<$ mask info loss. To compute these losses we use object specific masks $\sigma(S(o))$. For a face tracker $\sigma(S(o))$ is the collection of skin pixels inside the search box of the object $S(o) = \text{scale}(\frac{1}{r}, \frac{1}{r})C(o)$ where $C(o)$ is the calculation box of our object in this frame. Thus for each object o and for each frame we have the chain $\sigma(S(o)) \subset S(o) \subset C(o)$. And the detection of a possible occlusion event involving o_i and o_j in frame f is equivalent to $S(o_i) \cap C(o_j) \neq \emptyset$. If this event is detected we begin to shrink the search (and hence the calculation box) of o_i to $S'(o_i)$ such that $S'(o_i) \cap C(o_j) = \emptyset$. We use the Kullback-Leibler divergence to model information loss

$$\text{full info loss} = \text{KL}(P(C|S(o_i))||P(C|S'(o_i)))$$

$$\text{mask info loss} = \text{KL}(P(C|\sigma(S(o_i)))||P(C|\sigma(S'(o_i)))).$$

If the resizing is invalid, i.e., full info loss $<$ mask info loss, the object is considered occluded.

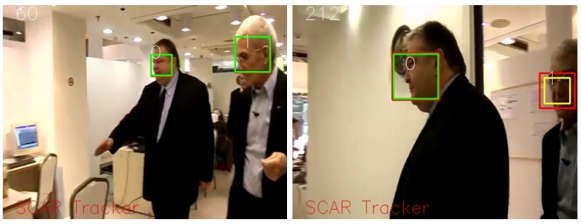

Fig. 5. The search boxes - marked by the inner yellow or green squares, and the corresponding calculation regions – marked by the outer, red rectangles.

3 Evaluation

As person trackers have improved there has been additional interest in *unconstrained*, *ad-hoc* data sets. Several of these do exist [11], however they tend to focus on easier examples: talking heads style interviews and fixed camera surveillance videos. The SCAR framework is evaluated on a set of videos involving large camera transitions, low resolution imagery (or distant subjects), and complicated scenarios involving object occlusions and fast appearance changes. The data set consists of both single-subject videos as well as multiple-person videos. Additionally it includes videos where a subject may leave and then re-enter the scene. The videos are:

The full Trellis and David Indoor: http://www.cs.toronto.edu/~dross/ivt;
Clips 0:0-1:19 from http://www.youtube.com/watch?v=Bl5lHtRvJcI;
Clip 0:0:05.75-1:17 from http://www.youtube.com/watch?v=tC3Q5mOGlDA.

The results for the tagging task are shown in Table 1 and Table 2. For reference we also show the performance of several state-of-the-art trackers [9,4,1,7,2] and the baseline Picasa tagging-by-per-frame-recognition. To avoid bias and implementation-based variations the evaluation tables include only comparisons with systems whose code was made available by the original authors. For all trackers the original code was used. The default parameters were used for all runs. For each video and for each person in the video the trackers and Picasa BL are seeded with the first recognition of the person in the video. Typically tracker and face detection performance measures involve the distance between the centers of the tracker/detection box $D_k(t)$ and the hand-labeled ground truth box $G_k(t)$. For testing tagging performance this measure is not relevant since the SCAR tags are already small by design and can cover very small patches of the face. We consider a valid detection to occur in frame t if the two windows intersect. Indeed, this choice gives a slight advantage to the competing trackers since SCAR tends to have very small boxes. In fact, in the evaluation tests the median size of the SCAR detection boxes is 1.4% of a video frame. As a face recognition engine we used a plug-in to Picasa.

4 Summary, Conclusions, and Future Work

We described a new fully automatic approach to address the persistence analysis task. The development of a new approach is motivated by: (i) The need to obtain high accu-

Tracker	*Precision*	*Recall*	*Accuracy*
SCAR	97.9%	100.0%	97.9%
IVT[7]	92.9%	100.0%	92.9%
USC_CT[4]	87.0%	100.0%	87.0%
MIL[2]	82.4%	100.0%	82.4%
FaceTrack[8]	77.9%	99.2%	77.7%
Picasa BL	**100%**	**69.8%**	**69.8%**
FRAGTrack[1]	86.0%	53.4%	49.5%
abcSHIFT[9]	85.1%	28.4%	25.8%

Table 1. Tracker Evaluation on Single-Person Videos Trellis and David Indoor

Tracker	*Precision*	*Recall*	*Accuracy*
SCAR	65.6%	81.8%	66.4%
Picasa BL	**100%**	**49.5%**	**58.6%**
IVT[7]	55.2%	68.0%	44.6%
MIL[2]	43.8%	100.0%	43.8%
FRAGTrack[1]	41.0%	71.1%	34.4%
USC_CT[4]	34.2%	100.0%	34.2%
abcSHIFT[9]	55.8%	31.4%	24.8%

Table 2. Evaluation while tracking the persistence of single individuals in unconstrained videos involving multi person scenarios, occlusions, pose and illumination variations, large camera transitions

racy rates; (ii) The need to develop a completely self-tuning framework that does not require manual tweaking and tuning. The contributions in this paper are: (i) formulation the persistence analysis task as a tagging problem; (ii) a novel method to select a representative tag from an object (facial) crop box in an image; (iii) a method to evaluate whether an initial tag can be used to initiate a dynamic sequence of continuously varying tags that can be used to mark the individual in subsequent frames; (iv) a CAMSHIFT based method to update a tag; (v) a method to detect abrupt changes in the scene (e.g., the person is occluded or exits the field of view) or in the person's appearance which is equivalent to detecting that the current foreground model is no more valid; (vi) a dual-mechanism method to re-acquire contacts and in effect re-initialize persons' tags; (vii) a method to resolve person-person occlusions. An evaluation on publicly available data and code shows that SCAR outperforms the current SOA methods. The proposed method applies to persistence analysis of arbitrary objects. The only place where we specialize to faces is when we select the actual mask (for intialization/reintialization), in which case we use skin color models (and a face recognition engine if possible). In our future work we will extend the tagging framework to tackle a wider category of objects.

5 Acknowledgements

We are very grateful to the authors of [1,2,4,7,9,10] for providing their code.

References

1. A. Adam, E. Rivlin, and I. Shimshoni. Robust Fragments-based Tracking using the Integral Histogram. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR*, volume 1, pages 798–805, 2006. 1, 10, 11, 12
2. B. Babenko, M.-H. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. In *CVPR*, 2009. 1, 10, 11, 12
3. G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, (Q2), 1998. 4
4. T. Dinh, N. Vo, and G. Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1177–1184, 2011. 1, 10, 11, 12
5. G. Gomez and E. F. Morales. Automatic feature construction and a simple rule induction algorithm for skin detection. In *In Proc. of the ICML Workshop on Machine Learning in Computer Vision*, pages 31–38, 2002. 6
6. G. Kamberov, M. Burlick, B. Luczinski, L. Karydas, and G. Kamberova. Collaborative track analysis, data cleansing, and labeling. In *Internatioal Symoposium on Visual Computing*. Springer Lecture Notes in Computer Science, 2011. 9
7. J. Lim, D. Ross, R.-S. Lin, and M.-H. Yang. Incremental learning for visual tracking. In *Advances in Neural Information Processing Systems*, pages 793–800, 2005. 1, 10, 11, 12
8. J. Saragih, S. Lucey, and J. Cohn. Deformable model fitting by regularized landmark Mean-Shift. *International Journal of Computer Vision*, 91(2):200–215, Jan. 2011. 11
9. R. Stolkin, I. Florescu, and G. Kamberov. An adaptive background model for camshift tracking with a moving camera. In *Proceedings of the 6th International Conference on Advances in Pattern Recognition*, pages 147–151, 2007. 1, 2, 4, 5, 8, 10, 11, 12
10. H. Wang and J. Oliensis. Rigid shape matching by segmentation averaging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(4):619–635, 2010. 6, 12
11. L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 529 –534, June 2011. 10