Intelligent Collaborative Tracking by Mining Auxiliary Objects

Ming Yang[†], Ying Wu[†]
[†]EECS Dept., Northwestern Univ.
2145 Sheridan Rd, Evanston, IL 60208, US
mya671,yingwu@northwestern.edu

Shihong Lao[‡]
[‡] Sensing & Control Technology Lab, OMRON
9-1 Kizugawadai, Soraku-gun, Kyoto 619-0283, Japan
lao@ari.ncl.omron.co.jp

Abstract

Many tracking methods face a fundamental dilemma in practice: tracking has to be computationally efficient but verifying if or not the tracker is following the true target tends to be demanding, especially when the background is cluttered and/or when occlusion occurs. Due to the lack of a good solution to this problem, many existing methods tend to be either computationally intensive with the use of sophisticated image observation models, or vulnerable to the false alarms. This greatly threatens long-duration robust tracking. This paper presents a novel solution to this dilemma by integrating into the tracking process a set of auxiliary objects that are automatically discovered in the video on the fly by data mining. Auxiliary objects have three properties at least in a short time interval: (1) persistent co-occurrence with the target; (2) consistent motion correlation with the target; and (3) easy to track. The collaborative tracking of these auxiliary objects leads to an efficient computation as well as a strong verification. Our extensive experiments have exhibited exciting performance in very challenging real-world testing cases.

1 Introduction

Although extensive research efforts have been taken, it is still quite difficult in practice to achieve robust and efficient long-duration tracking in unconstrained real-world environments. Most existing methods are in a dilemma: whether to be fast-but-fallible, or to be robust-but-slow.

This dilemma roots in the opposite requirements of the two kinds of image likelihood models: one that tends to be simple for efficient motion estimation and tracking, while the other tends to be sophisticated for comprehensive verification of the presence of the true target. We call them *tracking likelihood model or TLM* and *verification likelihood model or VLM*, respectively. For example, efficient TLMs are generally based on simple image features that are easily accessible, such as contours [12, 4], color histogram [7], or even image templates [5, 10], etc. On the other hand, VLMs are generally associated with classifiers that differentiate the true target from the others, thus they need to either extract the unique invariants or model the variations of the target's appearance. As a result, they tend to be computationally demanding or difficult to model. The ultimate verification is target recognition. A tracker can have a separate TLM and VLM. It is also possible to use one likelihood model for the two purposes, *e.g.*simply use the same TLM or VLM for both tasks, or even a smart combination like the SVM tracker [2]. The computational cost of either approach is bounded by the complexity of the VLM.

An effective target verification is important, since it tells if the tracker is following the true target or not. For a fast tracking, we prefer to use simple TLMs. But many real-world complications such as clutters, illumination and view changes, low image quality, motion blur and partial occlusions, all may invalidate these simple TLMs, such that these simple TLMs are likely to give a large number of false positive detections with a high confidence level.

Without a strong verification, the tracker is vulnerable. In addition, an effective verification is very important for adaptation. In practice, the visual appearances of the target and/or the dynamic environment may exhibit non-stationary characteristics. Unless visual invariants can be identified or the variability can be learned off-line (*e.g.*, by learning and switching multiple TLMs [10, 16, 14]), the tracker needs to adapt its TLM to the environment (*e.g.*, by selecting the discriminant features on-line [6, 22] or adapting the TLM [3, 23]). But adaptation is not trivial and it is risky, as its nature is a chicken-and-egg problem [23] and it generally lacks a mechanism to prevent model drifting, unless a strong verification that provides confident supervision can be asserted for the target in order to keep the risk minimum.

As a common practice, the VLM for verification is designed or trained off-line, by learning the possible variabilities of the target. Various levels of verification can be designed [21]. As mentioned above, the ultimate level is target recognition. It can be extremely difficult in general, if not impossible, because the target may have very different vi-

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE

sual appearances due to the changes of view, lighting, etc. Therefore, a natural question to ask is: can we have more efficient but still effective verification?

In this paper, we propose a novel solution to the aforementioned problem by taking the advantage of *auxiliary objects* that are automatically discovered on the fly in an unsupervised fashion by using data mining techniques. Auxiliary objects are those that have strong motion correlation to the target such that the correlation can be employed to improve the tracking and to provide a computationally efficient but powerful verification. Specifically, an auxiliary object should satisfy 3 properties at least in a short time interval: 1) persist co-occurrence with the target, 2) consistent motion correlation with the target, and 3) easy to track.

Figure 1. Some sample auxiliary objects to the target head.

Auxiliary objects can be in various forms, *e.g.* solid semantic objects which bear intrinsic relations with the target, or some image regions that happen to have motion correlation with the target for a short period. They may reliably associate with the target for a long duration, or only exist in a short time interval, or not exist at all. Thus it is impossible to determine auxiliary objects off-line, but they have to be discovered on the fly, which is fundamentally different from [24]. For example, in Figure 1, the targets of interest are the heads in solid-yellow boxes, and the image regions in dash-red boxes are the auxiliary objects discovered automatically. We resort to data mining techniques for discovering auxiliary objects by learning their co-occurrence associations to the target. Data mining methods originated from text information processing and relational databases [1], and found their uses in extracting video objects [19, 20, 15]. To the best of our knowledge, this paper presents an original attempt, if not the first, of combining visual tracking and data mining in a collaborative tracking framework.

This new approach has the following advantages. Firstly, it is computationally efficient. Because the auxiliary objects by definition are those easy to track (*e.g.* color regions), tracking them does not incur significant computational costs. Secondly, it outputs more accurate tracking results. The new method tracks the target and the set of auxiliary objects as a random field in a collaborative manner. It is provably correct that the uncertainty of the motion estimation is reduced. Thirdly, it also provides an effective verification, because the learned motion and/or geometric correlations among the target and the auxiliary objects serve as a strong cue for verification. Last but not the least, it is intelligent and robust. All the auxiliary objects and the motion correlation (*i.e.*, the random field) are automatically discovered on the fly. The robust fusion embedded can handle partial occlusions and even camouflages. Our extensive tests on real-world data give quite exciting performance in dealing with challenging cases including large scale changes, partial occlusions and complicated clutter backgrounds.

2 Intelligent Collaborative Tracking (ICT)

The new approach called *intelligent collaborative tracking* or ICT addresses the following three important issues:

- **Mining auxiliary objects** (in Sec. 2.1): the methods of extracting object candidates and learning the associations will be discussed. This step not only identifies a set of auxiliary objects, but also learns the random fields among them;
- **Collaborative tracking** (in Sec. 2.2): both the target and the set of auxiliary objects need to be tracked in ICT. Because they are not independent, the tracking is formulated on a random field and is achieved efficiently by the collaborations among all the individual trackers where an individual tracker influences other trackers as well as receives influence from others;
- **Robust fusion** (in Sec: 2.3): for an individual tracker, there may exist inconsistency among the set of influences it receives and its own image measurements. Handling inconsistency is fundamental and critical.

2.1 Mining auxiliary objects

2.1.1 Auxiliary objects

Auxiliary objects (AOs) are those that can help the target tracker. We abuse a little bit the term "object". In fact, it is not necessary for an AO to be a semantic object. In the tracking scenario, it refers to an informative image region or image feature that has the following three properties:

1. Frequent co-occurrence with the target;
2. Consistent motion correlation with the target;
3. Suitable for tracking.

Although this definition may cover a large variety of image regions or features, not all of them are appropriate for balancing the complexity and generality. Since the prior knowledge about the target and the environments are

in general not accessible, it is preferable to choose simple, generic and low-level auxiliary objects, such as image regions or feature points. Feature points are geometrically significant and provide the most localized information. There are some outstanding work on invariant feature points, *e.g.* [17, 18, 8]. Although feature points may be salient and therefore suitable for object recognition, because they can be easily localized, they are in general prone to occlusion, lighting and local geometry changes. Thus they are not always stable in video. In addition, extracting invariant features needs a good amount of computation, which makes it hard to achieve real-time performance. Therefore, although the tracking of feature points can be quite efficient, we generally do not use feature points as auxiliary objects.

Instead, we choose to use significant image regions. Different from localized image feature points, image regions reflect the image property in a neighborhood, and they tolerate more occlusions and local geometry changes. More importantly, image regions, if selected properly, can be reliably and efficiently tracked, for example, by the mean-shift algorithm [7]. Although texture regions may have invariants and can be very significant, our current implementation does not use them because it takes more computation to spot them than color regions. Therefore, our current treatment for data mining is to discover a set of color regions that are temporally stable and spatially correlated with the target in a video sequence in an unsupervised way.

2.1.2 Item candidate generation

To follow data mining's conventions and manifest the discussion, we define these terms in video data mining task.

Definition 1 *We denote an* item candidate *by s which is a particular image feature obtained by low-level image processing; an* item *by I which is a quantized item candidate by a* vocabulary $\mathcal{V} = \{I_1, \ldots, I_N\}$ *which is learned by clustering all item candidates; an* itemset *by* $\mathbf{I}$*, set of items; and a transaction by T, the itemset at a neighborhood R.*

The set of candidate AOs, denoted by C, is a subset of $\mathcal{V}$ that are frequently co-occurrent with the target. In our implementation, an item candidate is a rough color segment with its motion parameters, and an item is defined by $I = \{H(I), \mathbf{x}_I\}$, where $H(I)$ is the average color histogram of the item and $\mathbf{x}_I$ is the motion parameters and respective covariances.

The item candidates, *i.e.*, the color segments in our case, are the inputs for mining. Efficient segmentation is more preferred than a delicate but expensive one since exact boundaries of the segments are not necessary for mining and tracking. In our current implementation, we employ the classical split-merge quad-tree color segmentation [13]. The image is recursively split to the smallest possible homogenous color regions, and then the adjacent regions with similar appearances are merged gradually. The most prominent advantage of this method is the computational efficiency. Some segments are not appropriate for tracking, so we employ some heuristics to prune them, *e.g.* segments that are too large (the area over $1/2$ of the entire image) or too small (the area less than 64 pixels), and concave segments (the area less than $1/2$ of the bounding box) are excluded. Figure. 2 shows some typical segmentation results.

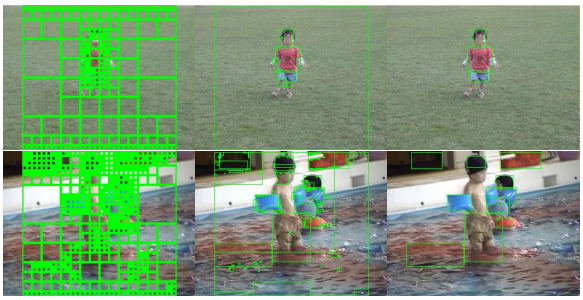

Figure 2. Illustration of the quad-tree color segmentation. (left) input frame, (middle) over-segmentation, (right) pruned segmentation.

2.1.3 Transaction generation

To build the vocabulary $\mathcal{V}$ so as to construct transactions, we need to quantize the space of the item candidates. We define the similarity of two item candidates by the Bhattacharyya coefficient [7] ρ of the histograms of the segments, and a k-means clustering is used to generate the vocabulary. Then each image segment (*i.e.* item candidate) can be quantized. For the item inside the region of interest in each frame, we collect the item and the items that are in the neighborhood of this item to form a transaction. There are different choices of the neighborhood. We can use the item itself (*i.e.* use a 0 neighbor) or use the item and those that are spatially adjacent to this item. Thus, for the image sequence, we can build a transaction database for mining.

2.1.4 Frequent itemset mining

Among the transaction database, an itemset which has a high co-occurrent frequency will be chosen as a candidate auxiliary object. There have been extensive research on this topic in data mining research. We use a modified FP-growth algorithm [9] which is computationally efficient for this purpose. Because mining is performed online, we need to take into account the importance of the history images. We maintain an M frames slide window and count the itemset frequency $f(i_n)$ with a forgetting coefficient $\beta = 0.9$. One special case is to treat an item as a transaction, and what we find are frequent items. If image segmentation does not end

up with too many small segments, this special treatment is good enough for identifying candidate auxiliary object. But the segmentation tends to over-segment and produces too many small segments, then we cannot use 0 neighbor for constructing transactions. Using the nearby items to form transaction can identify co-occurrent patterns that merge the adjacent small segments. This is another reason that it is fine for image segmentation step to be imperfect.

Finding such frequent itemsets only spots the candidate auxiliary objects that are frequently co-occurrent with the target, but they are not necessarily to have strong motion correlations. We need to check if these candidates satisfy other properties of an auxiliary object. For each candidate, we can initialize a mean-shift tracker to check if it can find its correspondence candidate in the successive image frames. If this tracker loses track for 4 frames in a row, we assert that this candidate is not suitable for tracking and remove it. Otherwise, we can form the motion trajectories over the frames in the slide window for a set of candidate auxiliary objects. Then, for each such motion trajectory, we check its correlation with the motion trajectory of the target. The ones with high correlation will be kept and are used as the final set of auxiliary objects.

Such a mining process is meaningful, because it has learned a random field. We denote the motion of the target by $\mathbf{x}_0$ and those of the auxiliary object by $\mathbf{x}_k, k = 1, \ldots, K$, where K is the number of auxiliary objects. They constitute a random field. The pair-wise potentials $\psi_{k0}(\mathbf{x}_k, \mathbf{x}_0)$ are actually learned as a by-product of data mining. In many cases, auxiliary objects share almost the same motion as the target, *e.g.*, the torso and the target head. Therefore, we can use a Gaussian distribution to characterize those potentials.

2.2 Collaborative tracking

Certainly, in the tracking scenario, such a random field is hidden and they need to be inferred from image evidence. We formulate this problem under a Markov network with a special topology, as shown in Figure. 3, where we only assume pair-wise connections between the target $\mathbf{x}_0$ and the auxiliary object $\mathbf{x}_k$ and there are no connections among auxiliary objects. Each of them is associated with its image evidence $\mathbf{z}_k$. We denote $\mathbf{Z} = \{\mathbf{z}_k, k = 0, \ldots, K\}$. The core of tracking is to estimate the posteriors $p(\mathbf{x}_0|\mathbf{Z})$ of the target and $p(\mathbf{x}_k|\mathbf{Z}), k = 1, \ldots, K$, for the auxiliary objects.

For such a singly connected network, a belief propagation algorithm with 2-step message passing gives the exact estimates of the posteriors. Denote $p(\mathbf{z}_i|\mathbf{x}_i)$ is the local likelihood and $\psi_{k0}(\mathbf{x}_k, \mathbf{x}_0)$ is pair-wise potential learned in the data mining step, $\phi_k(\mathbf{x}_k)$ is the local prior such as the dynamics prediction prior for $\mathbf{x}_k$, and $m_{ij}(\mathbf{x}_j)$ is the message passed from $\mathbf{x}_i$ to $\mathbf{x}_j$, and it is a function of $\mathbf{x}_j$. At the first iteration step the target $\mathbf{x}_0$ receives all the messages m_{k0} from every auxiliary object $\mathbf{x}_k$, then propagates the message back to them at the second iteration.

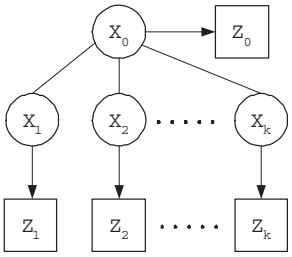

Figure 3. Star topology random field.

This message passing mechanism implies a collaborative way for tracking. Notice that if the target and the auxiliary objects are independent, their independent motion estimates are $\hat{p}_k(\mathbf{x}_k|\mathbf{Z}) \propto \phi_k(\mathbf{x}_k)p(\mathbf{z}_k|\mathbf{x}_k), k = 0, \ldots, K$. The relation between the true estimates and independent estimates is simply captured in a fixed-point equation of the messages:

$$p(\mathbf{x}_0|\mathbf{Z}) \propto \hat{p}_0(\mathbf{x}_0|\mathbf{Z}) \prod_k m_{k0}(\mathbf{x}_0), \quad (1)$$

$$m_{k0}(\mathbf{x}_0) = \int_{x_k} \hat{p}_k(\mathbf{x}_k|\mathbf{Z})\psi_{k0}(\mathbf{x}_k, \mathbf{x}_0)d\mathbf{x}_k, \quad (2)$$

$$p(\mathbf{x}_k|\mathbf{Z}) \propto \hat{p}_k(\mathbf{x}_k|\mathbf{Z})m_{0k}(\mathbf{x}_k) \quad k = 1, \ldots, K, \quad (3)$$

$$m_{0k}(\mathbf{x}_k) = \int_{x_0} \hat{p}_0(\mathbf{x}_0|\mathbf{Z}) \prod_{\mathbf{x}_i \setminus \mathbf{x}_k} m_{i0}(\mathbf{x}_0)d\mathbf{x}_0. \quad (4)$$

This suggests that we can use individual tracker for the target and auxiliary objects. These individual trackers need to combine their local estimates and the messages from others, and iterate. Such a collaborative mechanism leads to a very efficient solution to tracking the random field. Thus, even if our new approach involves the tracking of a set of auxiliary objects that are tracked by mean-shift, the computation is manageable because of the efficiency of the collaborative way and the efficiency of the mean-shift tracker.

Compared with a single tracker for the target, the involvement of auxiliary objects can reduce the uncertainty of the motion estimate of the target and thus make the tracking more confident. We can prove this in a special case when setting both the potential $\psi_{k0}(|\mathbf{x}_k - \mathbf{x}_0|)$ to be a Gaussian $N(\mu_{k0}, \Sigma_{k0})$ and the local likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$ to be a Gaussian $N(\hat{\mu}_k, \hat{\Sigma}_k)$ (we ignore the local prior without losing generality). Under this setting, the closed-form belief propagation gives:

$${\Sigma_0}^{-1} = \hat{\Sigma}_0^{-1} + \sum_{k=1}^{K} (\hat{\Sigma}_k + \Sigma_{k0})^{-1}, \quad (5)$$

$$\mu_0 = \Sigma_0(\hat{\Sigma}_0^{-1}\hat{\mu}_0 + \sum_{k=1}^{K} (\hat{\Sigma}_k + \Sigma_{k0})^{-1}(\hat{\mu}_k + \mu_{k0})), \quad (6)$$

where (μ_0, Σ_0) is the target's posterior when tracking the random field. If we assume the local priors to be Gaussian,

this result still hold but now $(\hat{\mu}_k, \hat{\Sigma}_k)$ refers to the independent (or local) posterior.

Eq. 5 makes it clear that Σ_0 is always less than $\hat{\Sigma}_0$, meaning that the confidence of the collaborative estimate of the target is higher than that produced by a single target tracker.

2.3 Robust fusion and verification

The closed form analysis for the collaborative tracking can be explained in the view of information fusion. When the connection potentials of the target and the auxiliary objects are set to be extremely tight, *i.e.*, the covariance of Σ_{k0} is 0, this belief propagation is equivalent to the best linear unbiased estimator (BLUE) for $\mathbf{x}_0$; if they are extremely loose, *i.e.* $\Sigma_{k0} = \infty$, it becomes independent estimation; otherwise, it is similar as covariance intersection.

However, there is a hidden assumption under this conclusion, *i.e.*, the information from all the sources must be consistent. In a simple term, they must more or less agree with each other. But in reality, this assumption may not be valid, when the estimate from the individual tracker may be completely different or inconsistent for many reasons. If we use the above mentioned method to fuse these inconsistent estimates, we amount to end up with a false estimate whose confidence is rather high. Thus, it is desirable to have a mechanism to detect the inconsistency and identify the outliers for a robust fusion.

Our study [11] gives a new theorem to handle the inconsistency. Due to the page limit, we give two criteria that are very useful for detecting the pair-wise inconsistency.

Theorem 1 *Considering two Gaussian sources $N(\mu_1, \Sigma_1)$ and $N(\mu_2, \Sigma_2)$, where $\mu_1, \mu_2 \in R^n$, the two sources are inconsistent if:*

$$\frac{1}{n}(\mu_1 - \mu_2)^T(\Sigma_1 + \Sigma_2)^{-1}(\mu_1 - \mu_2) \geq 2 + \sqrt{C_p} + \frac{1}{\sqrt{C_p}},$$

where C_p is the 2-norm conditional number of $\Sigma_1 + \Sigma_2$, and they are consistent if:

$$\frac{1}{n}(\mu_1 - \mu_2)^T(\Sigma_1 + \Sigma_2)^{-1}(\mu_1 - \mu_2) < 4.$$

Although these are sufficient conditions in general cases, they are actually also necessary conditions when $n = 1$. These criterion enable simple and quick detection of inconsistency. In addition, the estimation that is inconsistent with all the others will be regarded as an outlier. The outlier can be the target or the AOs. If the target is an outlier, we assert that the target is experiencing occlusion or drift, and stop the mining process temporarily. But we can give an estimation of the target purely based on the predictions from the auxiliary objects, and search for the image evidence. If the outlier is an auxiliary object, we simply exclude this auxiliary object for fusion. After excluding the outliers, we perform belief propagation again on the rest of the network. When the majority are not consistent which means the target estimate can not be verified, the tracking failure is asserted.

3 Experiments

3.1 Experiment settings

We substantialized and implemented the proposed ICT algorithm in a head tracking system, where the head tracker is a contour-based elliptical tracker similar to [4], and the auxiliary trackers are mean-shift trackers. Since fixed number of edge points along the ellipse are matched, the single head tracker is quite computationally efficient and runs at over 50 fps. Although the single head tracker is relatively robust to illumination and view changes, it is vulnerable to the clutter background, motion blur and occlusions. In our experiments, we compare the proposed ICT algorithm with the single head tracker in a large number of real-world sequences captured in unconstrained environments including both indoor and outdoor scenes. These extensive experiments and exciting results have demonstrated the advantages of the ICT algorithm.

The motion parameter $\mathbf{x} = \{u, v, s_u, s_v\}$ to be recovered includes the location (u, v) and the scales s_u and s_v. The color segmentation and the mean-shift tracker work in the normalized R-G color space with 32×32 bins. Without code optimization, our C++ implementation of ICT comfortably runs around 10 fps on average on Pentium 3G for 320×240 images depending on the number of auxiliary objects.

3.2 Quantitative experiments

For a quantitative evaluation, we manually labeled the ground truth of the sequences `kid in yellow` and `dancing girl` for 1200 and 1600 frames respectively. The evaluation criterion of tracking error are based on the relative position errors between the center of the tracking result and that of the ground truth, and the relative scale normalized by the ground truth scale. Ideally, the position differences should be around 0, and the relative scales 1.

As shown in Figure. 4 and Figure. 5, the position differences of the results in the ICT are much smaller than that of the single head tracker and the relative scales have much less fluctuations around 1, which demonstrate the advantages of the ICT, *i.e.* reducing the false alarm rate and the estimation covariance. Note that at the end of the sequence `kid in yellow`, the single tracker happens to track the head by chance after the drift. Although the ICT tracker loses track at around frame 1100 for several frames, it is able to recover promptly because of the auxiliary objects.

Some key frames are shown in Figure. 6. The first row shows the results of the single head tracker where the highlighted solid-yellow box indicates the location of the head. The second row is the segmentation and mining results, where each green rectangle indicates an item in the current frame. The blue numbers at the corner show the item labels of the candidate auxiliary objects. The third row illustrates the fusion results. Each blue box is the estimate of the head

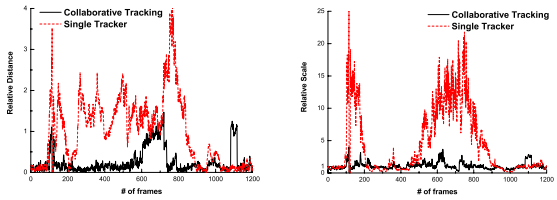

Figure 4. Quantitative comparison: (left) position errors, (right) scale errors, [`kid in yellow,1200 frames`].

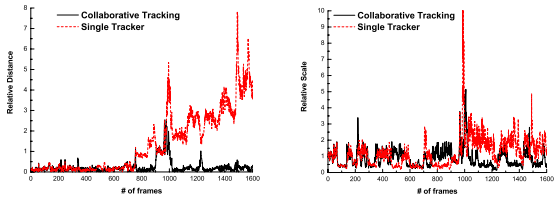

Figure 5. Quantitative comparison: (left) position errors, (right) scale errors, [`dancing girl,1600 frames`].

from difference sources (*i.e.* auxiliary objects or the target). The white box indicates that estimate is regarded as an outlier. The dark red box is the final result of the fusion. The corresponding labels of the auxiliary objects are shown at the bottom-right corner. The final tracking results of ICT are shown in the 4-th row as highlighted solid-yellow box, and the dash-red boxes are the auxiliary object trackers.

3.3 Occlusion and drift

Figure. 6 samples the results on the sequence `kid in yellow` which is very challenging due to a serious occlusion, target out-of-range and the clutters. When the head moves outside the upper boundary at frame 113, the single head tracker drifts to a false positive in the clutter background and is unable to recover. On the contrary, the ICT tracker asserts the occlusion and keeps tracking correctly. It freezes the head tracker temporarily and re-initializes it based on the predictions provided by the auxiliary objects. When the kid is walking in front of the bush, the background is so clutter that it causes big problems to the edge-based tracker. On the other hand, ICT discovers several auxiliary objects, *i.e.* the shirt and short pant, which are quite stable and provide roughly correct estimates of the head location and rescue the head tracker from the drift at frame 736.

3.4 Quick movement and camouflage

As shown in Figure. 7, the sequence `dancing girl` presents quick movements and camouflage. All the girls are similar in terms of the appearances. This is extremely difficult for a single head tracker to work, but ICT comfortably handles such a challenge. During the dancing, ICT gradually discovers the spatial relations of the target (a girl of interest) to the adjacent regions *e.g.* other girls' shirts, although such relations are only valid in short time interval. At frame 757, the single head tracker is trapped by the shoulder of the girl and unable to recover. At frame 758, the ICT tracker identifies this false alarm and pulls back the head tracker with the help of the predictions of the AOs that are still close to the true target. At frame 1234, the girl of interest suddenly gets down, ICT detects the tracking failure and resumes tracking quickly. ICT can comfortably track over 1600 frames for this highly dynamic sequence until the target moves outside the left boundary for several seconds.

3.5 Scale and view changes

We show the tracking performance when the target undergoes large scale and view changes and demonstrate the transition of the auxiliary objects in the sequence `kid&dad` (Figure. 8). For the single head tracker, when the scale of the head becomes very small, it drifts to the torso of the kid from frame 69 and loses track. During the first 300 frames, the dad walks with the kid with quite stable motion correlation. This is discovered by ICT and the region of dad's shirt is mined as the auxiliary object to help track the kid's head. When they move close to the camera, the scale and the view change dramatically so that the learned relation between dad's shirt and the kid's head no longer holds. Fortunately, ICT spots that the hat to be one good auxiliary object at large scale and guides the tracking. At the end of the sequence, the head is completely occluded by the hat for several seconds. Although this is impossible to recover, ICT detects and reports the tracking failure, while the single head tracker tends to drift to a false positive without notice.

3.6 Clutter background

As shown in Figure. 9 (`swimming boy`), the background is quite cluttered due to the texture of water and other people, which makes the single head tracker hopeless. On the other hand, ICT discovers the two blue life buoys and the swimming hat and uses them as the auxiliary objects. The single head tracker is easily distracted by the edges in the background and drifts away. When the boy jumps towards his mom's arms, ICT uses the life buoys as well as the orange box on the bank to help locate his head accurately, which is difficult for the single head tracker. Note that at the end of this sequence, the kid's head is occluded by his mum's head and ICT switches to the mom. This is reasonable because the auxiliary objects can not differentiate the two heads at the same location.

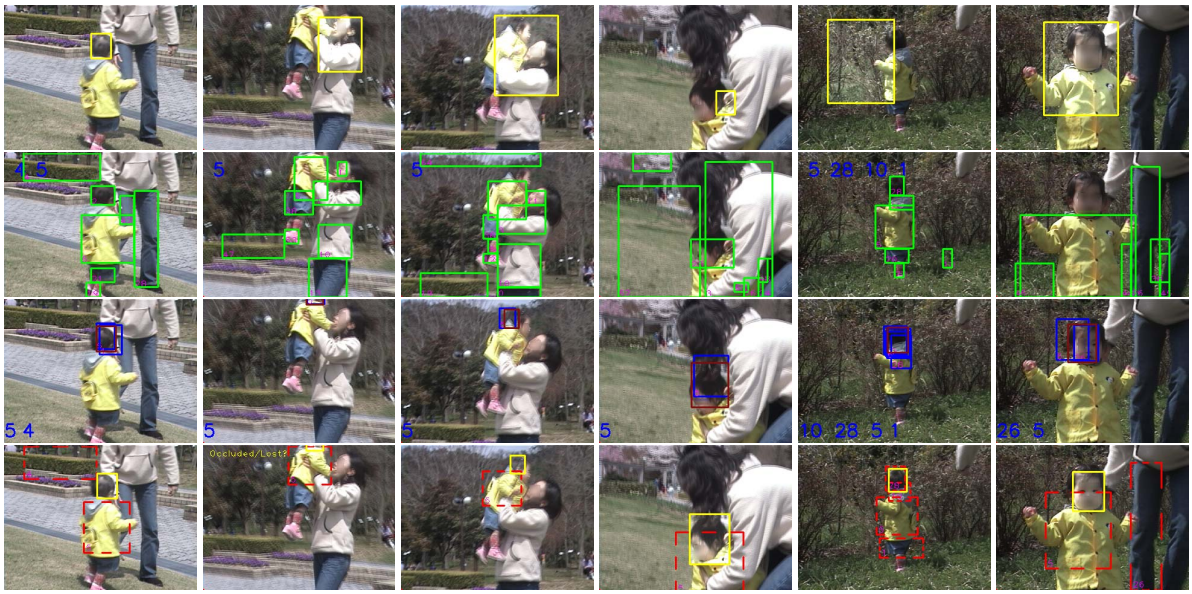

Figure 6. Frame # 50, 113, 124, 229, 736 and 866 of `kid in yellow,1200 frames`. (1st row) the head tracker, (2nd row) the mining results, (3rd row) the fusion results,(4th row) the ICT tracker.

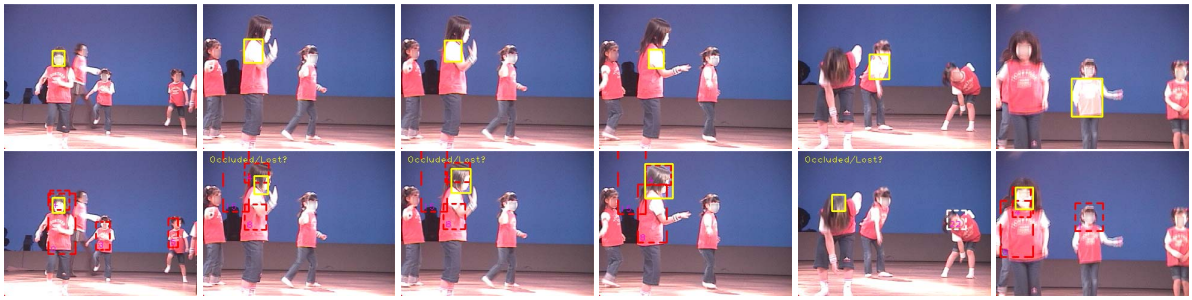

Figure 7. Frame # 67, 757, 758, 764, 1234 and 1372 of `dancing girl,1600 frames`. (top) the head tracker, (bottom) the ICT tracker.

3.7 Discussions

As illustrated in the challenging sequences, there are two primary reasons why the auxiliary objects greatly help the tracking: 1) some auxiliary objects have persistent relations with the target and present fairly accurate estimates although these relations may not be foreseen; 2) a number of auxiliary objects have transitional relations with the target and the majority of them can give rough correct estimates in a short time interval. In the cases of occlusion or drift, it is not likely that all the auxiliary objects are occluded or all auxiliary trackers lose track at the same time, since the auxiliary objects may not locate in a close neighborhood of the target. The mechanism of robust fusion can identify the inconsistency induced by occlusions or drifts. Although there are some too difficult cases, *e.g.* the target is occluded for long time, ICT fails reasonably because on-line data mining may not be invoked at all. The advantage of ICT is the ability to detect and report the failure, and leave the system to other means of re-initialization, while the single tracker tends not to report the failure but keep working aimlessly.

4 Conclusions

We proposed a novel solution to robust long-duration tracking by integrating the auxiliary objects discovered on the fly by data mining. The auxiliary objects provide extra measurements to the target and reduce the uncertainty of the estimation. In addition, the learned motion correlations among the auxiliary objects and the target serve as a strong cue to verify the tracking result to avoid drifting due to short-term occlusion or tracking lost. The auxiliary objects are automatically discovered without supervision and do not incur much extra computation, which makes the approach generally applicable to different tracking scenarios.

Figure 8. Frame # 52, 69, 70, 313, 555 and 616 of `kid&dad,617 frames`. (top) the head tracker, (bottom) the ICT tracker.

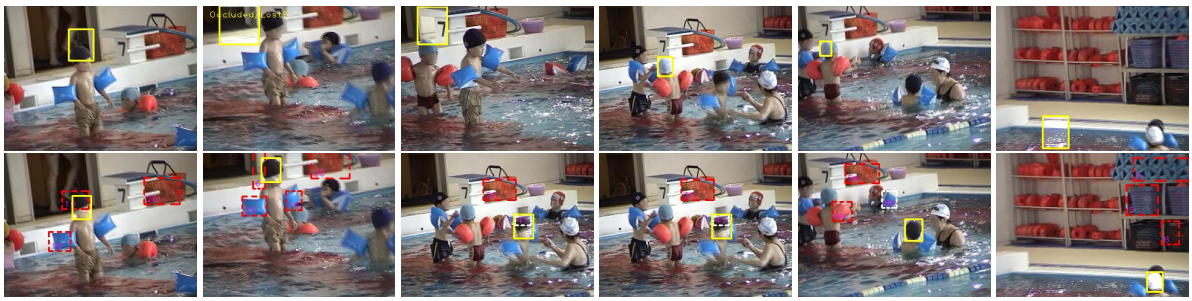

Figure 9. Frame # 87, 131, 334, 526, 578 and 848 of `swimming boy,900 frames`. (top) the head tracker, (bottom) the ICT tracker.

Acknowledgement

This work was supported in part by NSF IIS-0347877, IIS-0308222 and Northwestern faculty startup funds.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *VLDB'94*, pages 487 – 499, Chile, 1994.

[2] S. Avidan. Support vector tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 26:1064–1072, Aug. 2004.

[3] S. Avidan. Ensemble tracking. In *CVPR'05*, volume 2, pages 494 – 501, San Diego, June 2005.

[4] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *CVPR'98*, pages 232 – 237, Santa Barbara, CA, June 1998.

[5] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *ECCV'96*, pages 329–342, Apr. 1996.

[6] R. T. Collins and Y. Liu. On-line selection of discriminative tracking features. In *ICCV'03*, volume 2, pages 346–352, Nice, France, Oct. 2003.

[7] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *CVPR'00*, volume 2, pages 142–149, Hilton Head Island, SC, June 2000.

[8] A. Fitzgibbon and A. Zisserman. On affine invariant clustering and automatic cast listing in movies. In *ECCV'02*, volume 3, pages 304 – 320, 2002.

[9] G. Grahne and J. Zhu. Fast algorithms for frequent itemset mining using fp-trees. *IEEE Trans. Knowledge Data Eng.*, pages 1347 – 1362, 2005.

[10] G. Hager and P. Belhumeur. Real-time tracking of image regions with changes in geoemtry and illumination. In *CVPR'96*, pages 403–410, San Francisco, June 1996.

[11] G. Hua and Y. Wu. Measurement integration under inconsistency for robust tracking. In *CVPR'06*, June 2006.

[12] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *ECCV'96*, 1996.

[13] R. Jain, R. Kasturi, and B. G. Schunck. *Machine Vision*. McGrawHill, Inc, 1995.

[14] K.-C. Lee and D. Kriegman. Online learning of probabilistic appearance manifolds for video-based recognition and tracking. In *CVPR'05*, volume 1, 2005.

[15] M. Leordeanu and R. Collins. Unsupervised learning of object features from video sequences. In *CVPR'05*, volume 1, pages 1142 – 1149, San Diego, CA, June 2005.

[16] J. Lim, D. Ross, R.-S. Lin, and M.-H. Yang. Incremental learning for visual tracking. In *NIPS'04*, pages 801–808, Vancouver, Canada, Dec. 2004.

[17] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV'99*, Sept. 1999.

[18] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *ICCV'01*, volume 1, pages 525 – 531, Vancouver, Canada, July 7 - 14 2001.

[19] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *ICCV'03*, volume 2, pages 1470 – 1477, Nice, France, Oct. 2003.

[20] J. Sivic and A. Zisserman. Video data mining using configurations of viewpoint invariant regions. In *CVPR'04*, volume 1, pages 488 – 495, Washington, DC, Jun.27-Jul.2 2004.

[21] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR'01*, volume 1, pages 511 – 518, Hawaii, Dec. 2001.

[22] J. Wang, X. Chen, and W. Gao. Online selecting discriminative tracking features using particle filter. In *CVPR'05*, volume 2, pages 1037 – 1042, San Diego, June 2005.

[23] M. Yang and Y. Wu. Tracking non-stationary appearances and dynamic feature selection. In *CVPR'05*, volume 2, pages 1059 – 1066, San Diego, June 2005.

[24] T. Yu and Y. Wu. Collaborative tracking of multiple targets. In *CVPR'04*, volume 1, pages 834 – 841, Washington, DC, Jun.27-Jul.2 2004.