# Attention-based Target Localization using Multiple Instance Learning[*]

Karthik Sankaranarayanan and James W. Davis

Dept. of Computer Science and Engineering
Ohio State University
Columbus, OH, USA
{sankaran,jwdavis}@cse.ohio-state.edu

**Abstract.** We propose a novel Multiple Instance Learning (MIL) framework to perform target localization from image sequences. The proposed approach consists of a softmax logistic regression MIL algorithm using log covariance features to automatically learn the model of a target that persists across input frames. The approach makes no assumptions about the target's motion model and can be used to learn models for multiple targets present in the scene. The learned target models can also be updated in an online manner. We demonstrate the validity and usefulness of the proposed approach to localize targets in various scenes using commercial-grade surveillance cameras. We also demonstrate its applicability to bootstrap conventional tracking systems and show that automatic initialization using our technique helps to achieve superior performance.

## 1 Introduction

Object tracking is one of the most studied problems in the field of computer vision. Tracking of pedestrians is an end application in itself, or can be used for collecting trajectories for higher level behavior analysis. Traditional techniques for following particular targets in surveillance settings which use *active* PTZ cameras involve a security operator driving a joystick (to control the PTZ camera) and adjusting it to follow the target of interest. For handing off the task to an automatic tracker, the operator would have to stop and select the target (e.g., with a bounding box) for the tracker to initialize. While on the one hand, learning the target model from a single frame is not robust, requiring the operator to click on multiple successive frames is also very impractical. Moreover, due to the fast and real-time nature of these tasks, such initializations are both unreliable and require significant training. They also require the ability to control multiple input devices for driving the camera (joystick) and clicking on the video feed (mouse). In this situation, an automatic technique to learn target models that is both robust for tracking and intuitive to operate is needed.

In a typical surveillance setting, an intuitive way for an operator to initialize a tracker would be to use the joystick to loosely follow the person of interest and

[*] Appears in *International Symposium on Visual Computing*, November 2010.

then have the system *automatically* learn the intended target, and then continue tracking the target. Since the system now has a robust model that has been automatically learned using the target's persistent appearance features, it can use this model to reliably track the target in future frames. Towards this end, we propose an attention-based technique to learn target models using a machine learning approach. (Note that the attention nature of this method allows the target to be anywhere in the scene/image and therefore does not assume any particular motion model on the part of the target). A limitation of supervised learning is that it requires labeling of the individual instances, which are typically hard to obtain (this is also the case in our setting, as explained later). Multiple instance learning (MIL) is a variant of supervised learning where it relaxes the granularity at which the labels are available. Therefore, in MIL settings, the instances are grouped into "bags" which may contain any number of instances and the labeling is done at the bag level. A bag is labeled positive if it contains at least one positive instance in it. On the other hand, a bag is labeled negative if it contains all negative instances. Note that positive bags may contain negative instances.

In our problem formulation, we model images containing the target of interest as bags, and regions within the images as instances. The MIL framework is therefore well suited to this task because it is guaranteed that at least one instance in the bag/image contains the object of interest, and it is much more efficient (faster and cheaper) to label bags/images instead of individual image regions. Therefore, we take image patches from areas of motion in the image (when the camera is not moving) and create a positive bag using all of these instances. At the same time, we sample image patches from the non-moving areas and create a negative bag from these instances. (Another option is to collect all patches from the input image and build a positive bag, and use patches from a background image of the same/similar region to construct a negative bag.) By repeating this for every frame, we are guaranteed that every positive bag contains at least one patch containing the target of interest and at the same time ensures that the target is absent from all the negative bags. By training a MIL algorithm based on logistic regression, we learn a target model that can be used to classify every instance (image patch) from a new incoming frame as target or not with a certain probability. We then use this probability map over the image and threshold it to update the model for the target.

## 2 Related Work

While there has been much work in the areas of pedestrian detection and visual tracking separately, not much work has been done in automatic localization from the point of view of initialization of a tracker based on visual attention. Pedestrian detection approaches such as [1] are generally view specific and are therefore unsuitable for our domain since PTZ cameras overlooking a large area can have a wide range of pedestrian views (from fronto-parallel to top-down). Moreover, such approaches are not applicable for finding the most *persistent* pedestrian in the scene (the target to be tracked), which is what is required to initialize an object tracker in our operator-joystick setting. In the area of object tracking, popular approaches include appearance-based techniques such as Mean-shift [2] and Covariance tracking [3], and filtering and association-based approaches such as a Kalman filter [4] and particle filter [5]. All of these approaches require good target localization and initialization for them to work well, and assume that a good initial target model is provided.

In the area of Multiple Instance Learning, the original work of [6] proposed to learn axis-parallel rectangles for modeling target concepts. Since then there have been various algorithms proposed including Diverse Density (DD) [7], EM-DD [8], and SVM techniques [9]. More recently, in a comparison study of MIL algorithms, Multiple instance logistic regression [10] has been shown to be the state-of-the-art MIL algorithm, especially for image retrieval tasks. Our contribution is an adaptation of such a logistic regression based MIL algorithm based on the *softmax* function and a new application problem.

## 3 Multiple Instance Learning

In order to learn the target model from a sequence of images and localize the target of interest within a new image, we wish to build a discriminative classifier which can output the probability $p(y = 1|x)$ indicating the posterior probability that the target is present $(y = 1)$ in the image patch $x$. In a MIL framework, the input data is obtained in the form of positive bags $(B^+)$ and negative bags $(B^-)$ containing instances. More formally, the input is presented as $\{(X_1, y_1), (X_2, y_2), ..., (X_n, y_n)\}$ where $X_i = \{x_{i1}, x_{i2}, ..., x_{im}\}$ denotes bag $i$ containing $m$ instances and has a corresponding bag label $y_i \in \{0, 1\}$. Each instance $x_{ij}$ is a feature vector calculated for an image patch $j$ from bag $i$. The bag labels are obtained from the instances in the bags. More specifically, a bag is labeled positive if it contains *at least one* positive instance. A bag is labeled negative if it contains *all* negative instances.

Using a likelihood formulation, the correct bag classifier/labeler will maximize the log likelihood of labels over all the bags (given the MIL constraints)

$$\log \mathcal{L} = \sum_{i}^{n} \log p(y_i|X_i) \tag{1}$$

where $p(y_i|X_i)$ is the probability of the bag $i$ (given its instances) having label $y_i$. As we can see, since the above likelihood formulation is expressed in terms of bag probabilities and what we want is to learn an instance-level classifier (for an instance/patch $x$), we will use a combining function to assemble instance-level probabilities into a bag probability. This is done using the *softmax* combining function as follows.

From the definition of positive and negative bags, we can formally express the notion of bag label in terms of its instance labels as

$$y_i = \max_{j}(y_{ij}) \tag{2}$$

which states that the label of a bag is the label of the instance within it which has the highest label (i.e.,$\{0, 1\}$). Notice how this formulation conforms to the definition of positive and negative bags and encodes the multiple instance assumption. Here, we incorporate a probabilistic approximation of the max operator called *softmax*, in order to combine these instance probabilities in a smoother way, so as to allow all instances to contribute to the bag label. This *softmax* function is defined as: $\text{softmax}(a_1, ..., a_m) = \sum_{j=1}^{m} (a_j \exp(\alpha a_j)) \, / \sum_{j=1}^{m} \exp(\alpha a_j)$. $\alpha$ is a constant that controls the weighting within the *softmax* function such that *softmax* calculates mean when $\alpha$=0 and max when $\alpha \to \infty$.
The bag-level probabilities for positive and negative bags are now defined as

$$p(y_i = 1|X_i) = \text{softmax}(t_{i1}, ..., t_{im}) \, , \quad p(y_i = 0|X_i) = 1 - p(y_i = 1|X_i) \qquad (3)$$

where $t_{ij} = p(y_{ij} = 1|x_{ij})$ are the instance level probabilities being combined to obtain the bag probabilities $p(y_i|X_i)$. Thus, if one of the instances is very likely to be positive, the nature of the *softmax* combining function is such that its estimate of the bag's "positive-ness" will be very high, since it gives an exponentially higher weight to such an instance, and consequently the weighted average of all the instances will also be high. Here, $\alpha$ controls the proportion of instances in the bag that influence the bag label. Therefore, if one has an estimate of the proportion of positive to negative instances in the positive bags (noise-level), one can appropriately tune $\alpha$ to reflect this, and hence learn more robust models than by simply using the max operator.

Next, to model these instance-level probabilities $t_{ij}$, we employ a logistic formulation given as

$$t_{ij} = p(y_{ij} = 1|x_{ij}) = \frac{1}{1 + \exp(-\mathbf{w} \cdot x_{ij})} \qquad (4)$$

where the parameter vector $\mathbf{w}$ (to be learned) models the target of interest, so that the probability $p(y_{ij} = 1|x_{ij})$ calculated with Eqn. 4 would be high for an image patch $x_{ij}$ that contains the target, and low for a patch that does not contain the target.

Now, using Eqns. 4 and 3 in Eqn. 1 along with a regularization term on $\mathbf{w}$, we can express a maximum likelihood formulation (in terms of the parameter vector $\mathbf{w}$ to be learned) as

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} \sum_{i \in B^+} \log \left( \frac{\sum_{j=1}^{m} t_{ij} \exp(\alpha t_{ij})}{\sum_{j=1}^{m} \exp(\alpha t_{ij})} \right) + \sum_{i \in B^-} \log \left( 1 - \frac{\sum_{j=1}^{m} t_{ij} \exp(\alpha t_{ij})}{\sum_{j=1}^{m} \exp(\alpha t_{ij})} \right) - \frac{\lambda}{2} \mathbf{w}^{\mathrm{T}} \mathbf{w} \qquad (5)$$

where the regularization term is obtained by using the following prior on the parameter vector $\mathbf{w}$,

$$p(\mathbf{w}) \sim \mathcal{N}(0, \lambda^{-1} \mathbf{I}) \, , \quad \lambda > 0 \qquad (6)$$

to give better generalization performance [11]. To optimize Eqn. 5, we use a gradient-based optimization technique using the BFGS method [12].

The parameter vector $\hat{\mathbf{w}}$ obtained from the optimization algorithm thus represents the learned target model. Therefore, when presented with a new image, the probability that an image patch within it (with feature vector $x$) contains the learned target can be calculated from Eqn. 4 using $\hat{\mathbf{w}}$.

## 4 Image Features and Instance Model

We adopt a variation of covariance matrix features to obtain a feature vector $x_{ij}$ for each image patch instance in our formulation. Covariance features have been shown to be robust appearance-based descriptors for modeling image regions [3]. The covariance matrix representation $C_R$ for a given image patch $R$ of size $W \times H$ in our framework is calculated as

$$C_R = \frac{1}{WH} \sum_{k=1}^{WH} (f_k - \mu_R)(f_k - \mu_R)^T \tag{7}$$

where $f_k = [x\ y\ r\ g\ b\ I_x\ I_y]$ is a 7 dimensional feature vector using a combination of position, color, and gradient values at each pixel location in the image patch $R$, and $\mu_R$ is the mean feature vector within the image patch.

We require a Euclidean distance based feature representation for Eqn. 4 whereas distances between covariance matrices are based on their eigenvalues [3]. Therefore, we use the property from [13] that eigenvalues of matrix logarithm of a covariance matrix $C_R$ are equal to logarithms of eigenvalues of $C_R$. Therefore, the covariance matrix descriptor can be transformed to a feature vector representation by first calculating the matrix logarithm of $C_R$ to obtain $C_l$ and then stringing out the elements of the matrix $C_l$ to obtain a vector $C_v$ [13]. Moreover, since the matrix logarithm $C_l$ is a symmetric matrix, it is fully specified by its bottom triangular part. Therefore, the feature vector $C_v$ only needs to have the bottom triangular part of $C_l$, with the off-diagonal elements scaled by $\sqrt{2}$ to compensate for their double presence in the matrix. In our case, the 7x7 dimensional covariance matrix reduces to a 28 dimensional feature vector. We then use these log covariance-based features to model the instances $x_{ij}$ corresponding to each image patch.

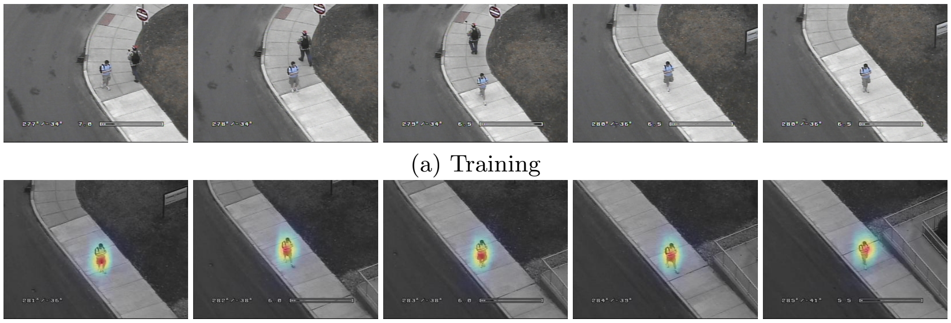
## 5 Target Localization Algorithm

The first step in our localization approach is to extract image patch instances from a sequence of images and use them to construct positive and negative bags. Given an input image sequence such as Fig. 1(a), we first detect regions of motion in each image by standard frame differencing (with the assumption that the target is moving). For each image, we then extract image patches from a reasonably large sample of the pixel locations marked as belonging to the motion region (the patch size can be predetermined or multiple sizes/aspect-ratios can be used). We construct a positive bag for this image using these instances since it is guaranteed to have at least one instance patch containing the desired target. Note that with this technique, instances corresponding to other parts of the scene in motion (trees, cars, noise pixels, etc.) would also be added, but that is acceptable since a positive bag can contain negative instances. At the same time, we sample a similarly large number of pixel locations from the (non-moving) background and extract image patches from these locations to construct the corresponding negative bag. Notice that this method ensures that no instance in this bag will contain the target. We similarly repeat this process for each of the input frames. This way it is guaranteed that at least one instance corresponding to the desired target is present in *each* of the positive bags and at the same time, absent from *all* the negative bags, thus satisfying our Multiple Instance assumption.

Once the positive and negative bags are constructed, we train the MIL classifier to learn the target concept by using the aforementioned optimization method. We initialize the weight parameter vector $\mathbf{w}$ uniformly at random between (-1, 1). The algorithm converges when the maximum change in the weight vector is less than a fixed small threshold $\epsilon$.

**Online Update:** An important aspect of the proposed learning approach is that the learned target concept can be updated in an online manner with each new incoming frame instead of having to retrain the classifier using all the positive and negative bags collected from the beginning. We use the assumption that the target appearance does not change much with the new frame and hence would result in only a small change in the target model. Once we receive a new incoming frame, we create a positive and negative bag using the method described in Sect. 5 and run the gradient optimization algorithm, but this time with the weight vector initialized to the previously learned target model. Once the algorithm converges, we obtain the new weight vector reflecting the updated target model.

**Multiple Targets:** Another advantage of the proposed approach is its ability to learn multiple target concepts (if present across the input bags). Since it could be the case that more than one target is present across all input frames, there could potentially be multiple target models to be learned by the algorithm. Therefore, the proposed approach tests for multiple targets by using the first learned concept to remove all corresponding target instances from the positive bags, and then retrains to learn the next strongest concept, and so on for each remaining target. More specifically, once the algorithm converges and learns the first target model, it then uses Eqn. 4 with this model for every instance from every positive bag to calculate its probability of being the target. It then classifies each instance as positive (target) if this probability lies above a fixed threshold $\sigma$. We then update every positive bag by removing from it all the instances classified as being positive. This ensures that none of the positive bags contain even a single instance corresponding to the learned target. It is important that the threshold $\sigma$ be picked conservatively so as to eliminate every true positive



(a) Training

(b) Testing

**Fig. 1.** (a) Image sequence input to MIL. (b) Probability surfaces overlaid on incoming frames showing target localization (best viewed in color).

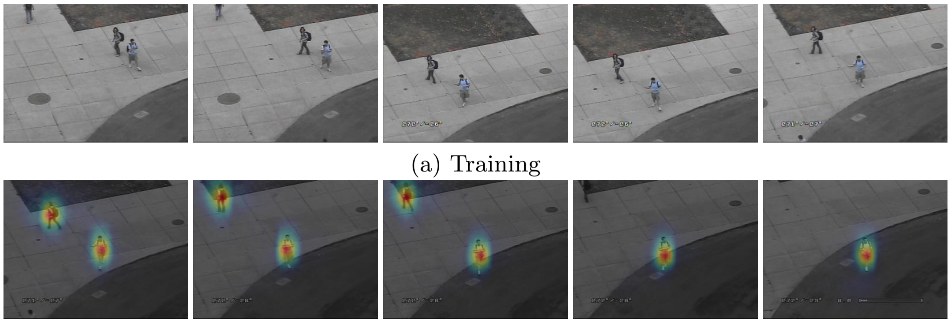
instance, even at the cost of eliminating a few false positives if necessary (we set $\sigma$=0.75). Since all instances corresponding to the first target have now been removed, re-running the optimization algorithm learns the next strongest target concept (if such a valid target is present). This process is repeated until all valid target concepts have been learned. To test whether a valid target concept has been learned or not, we calculate the log likelihood of the learned model on the input set of positive and negative bags. An extremely low value of the combined likelihood over all the bags indicates that the learned model is degenerate and there was no target concept left to learn.

## 6 Experiments

In this section, we present various experiments to demonstrate the proposed approach for target localization, apply it to a unique target detection and auto-locking system, demonstrate its sufficiency for learning appearance-based models to bootstrap object tracking systems, and finally, compare its performance with other manual initialization techniques.

### 6.1 Automatic Target Localization

Given a sequence of input images, we first constructed a set of positive and negative bags according to the technique described in Sect. 5 with patch size of 75x25 pixels. We then ran the MIL algorithm to learn a concept corresponding to a target that was common across all positive bags and absent in each of the negative bags. The parameters of the learning algorithm were set as $\alpha$=3, $\epsilon$=$10^{-5}$, and $\lambda$=$10^{-2}$. The validity of the learned target concept was checked by calculating the likelihood of the learned model across all input bags. Next, for every new incoming frame, the target model was evaluated against the input image at every possible location. This results in a probability surface corresponding to



(a) Training

(b) Testing

**Fig. 2.** (a) Input image sequence containing 2 valid targets. (b) Probability surfaces for incoming frames showing unique target detected at a later instant.

the new incoming frame indicating the probability of the target being present at each particular location. Figure 1(a) shows the sequence of input frames used to learn a target model. Figure 1(b) shows the probability "heatmap" overlaid on new input frames, representing the probability surface $p(y = 1|x)$ for each patch $x$ across the image. As seen in the figure, the target (in blue) present across all input images was detected and localized by the algorithm. After this, a new pair of positive and negative bags was created for the new input frame using the online update technique described in Sect. 5. The target model was then updated using the MIL algorithm and the updated model was used to evaluate the next incoming frame. This process was repeated with every new frame. Figure 1(b) shows the results of target localization using model update in the new input frames. Notice also that the other person (wearing black pants) was not learned by the algorithm since that person was not present in all of the frames, thus not satisfying the MIL criterion.

### 6.2 Auto-locking for Unique Target Detection

A useful feature of the proposed approach with the online update is its ability to continue updating all the target models (if there are multiple targets present in the scene satisfying the MIL criterion), and continue this process until a *unique* target is detected and localized in the scene. This is possible because the likelihood evaluation from the optimization algorithm can be used to identify the number of target models learned. Thus, this feature is useful in a system which can continuously update multiple target models until only the single most persistent target remains in the scene and then use that model for active tracking. We demonstrate this ability here.

As seen in Fig. 2(a), there were 2 targets present in the scene satisfying the MIL criterion. Therefore, the algorithm learned 2 corresponding target models and these were then used for localization with the incoming frames as shown in

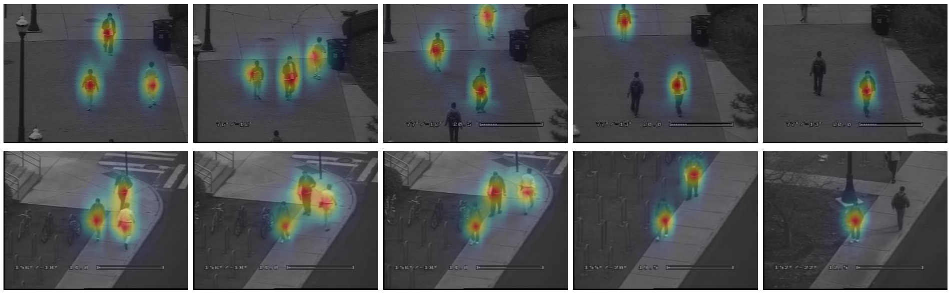


**Fig. 3.** Localization results from two other sequences involving multiple targets.

Fig. 2(b) (note that a probability surface corresponding to each learned model is obtained separately, but they are shown here overlaid on the same image for compactness). Further, with each new incoming frame, the target models were updated online and eventually, when incorporating a frame where one of the targets is no longer present, the MIL criterion is violated and consequently the only single remaining target model was updated, as shown in the last two frames in Fig. 2(b). Even if the other target later re-enters the scene, they will not be localized, as the MIL criterion requires the target to be persistent across all frames from the beginning. Figure 3 shows correct localization results from 2 other sequences involving multiple targets and distractors.

### 6.3 Sufficiency of Learned Models for Tracking

Once a unique target is detected, the next step is to use the probability surface for localizing the target and build an appearance model that can then be used for tracking. We present experiments that demonstrate that the target models learned using the proposed approach can be used with commonly used tracking methods such as covariance and mean-shift tracking.

We use the probability surface generated by the MIL algorithm and threshold it to extract the extents of the target which can then be used to build an appearance-based model for tracking. Figure 4(a) shows the probability surface for a particular input frame. We then picked a threshold of 0.5 on the probability surface and used this to extract the area of the learned target model. Figure 4(b) shows the thresholded area of the target corresponding to Fig. 4(a), and the associated image chips in Fig. 4(c) show the results for various other input frames. Note that they all roughly correspond to the same area (target's torso and legs).

We then calculated the width and the height extents of the thresholded area and fit a bounding box around the region. This bounding box was then used to learn an appearance-based model of the target and bootstrap different trackers. This bounding box (mostly around the target's torso and legs) captured the appearance features that remained most persistent across input frames, as opposed to a larger bounding box around the entire body (includ-

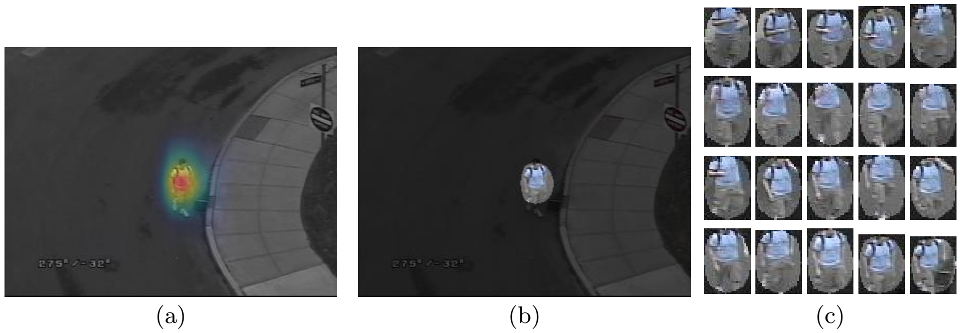
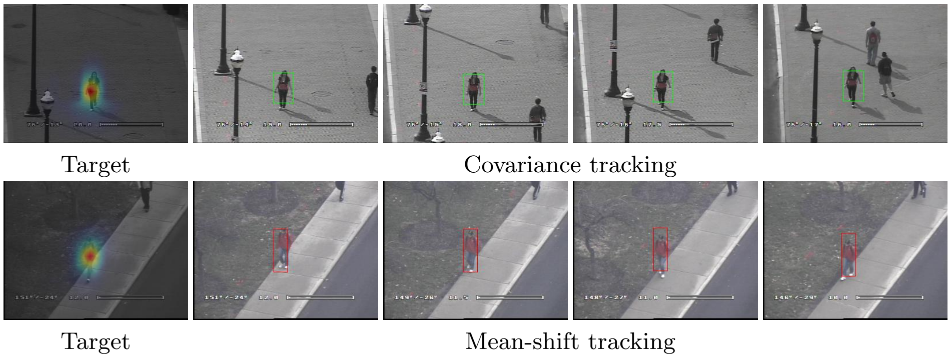

(a) (b) (c)

**Fig. 4.** (a) Probability surface. (b) Thresholded surface showing identified target area. (c) Image chips showing region used to learn appearance model for target tracking.


Target

Covariance tracking

Target

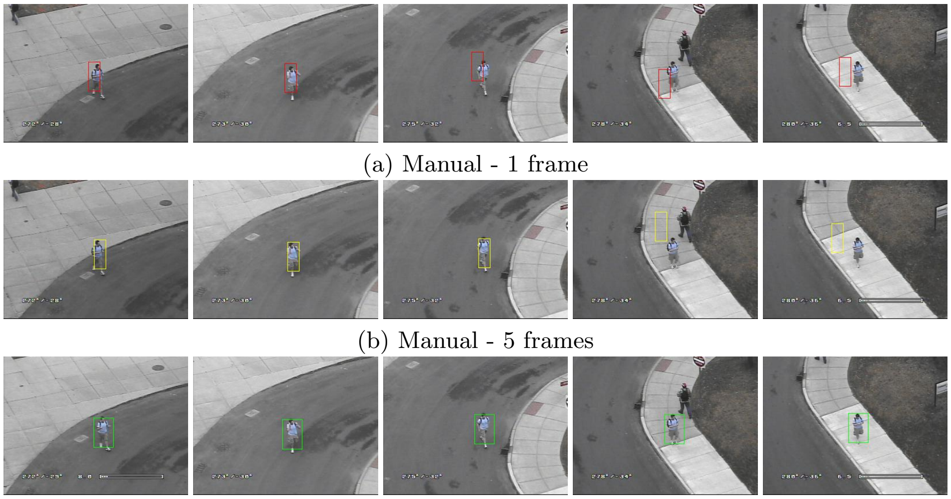
Mean-shift tracking

**Fig. 5.** Results from bootstrapping (a) covariance tracker (b) mean-shift tracker using a model learned from the proposed approach.

ing head, hands, and feet), which could potentially include several background pixels. We evaluated this approach with both covariance and mean-shift trackers. Figure 5(top) shows a few frames from tracking a target using a covariance tracker, and Fig. 5(bottom) shows the results using the mean-shift tracker. In both cases, the appearance model learned using the proposed approach is reliable and sufficient to bootstrap standard object trackers.

### 6.4 Comparison with Manual Initialization

Using a standard test sequence, we next performed three experiments to compare the performance of our automatic localization and tracker initialization with typical manual initialization techniques to demonstrate the applicability of our approach. In the first experiment, we initialized a covariance tracker by manually specifying the location of the target and the size of the bounding box around the target in the first frame. In the second experiment, instead of manually marking



(a) Manual - 1 frame

(b) Manual - 5 frames

(c) Proposed

**Fig. 6.** Tracking results for (a) manual initialization with just 1 frame, (b) manual initialization using 5 frames, and (c) automatic initialization using proposed approach.

the target in only the first frame, we marked its locations in each of the first 5 frames and computed a manifold mean of the covariance matrix representations from all 5 frames, and then used this model to manually initialize the covariance tracker. Even though this scenario is unrealistic in practical settings (where one cannot expect to manually select the target in every frame), we performed this experiment for a fair comparison with the proposed approach since our automatic localization uses an initial set of frames. (Note that the covariance tracker used a model update every frame). The third experiment involved tracking the target after automatically learning the target model using the proposed approach.

The results of each of the three experiments are shown in Fig. 6. As seen in Fig. 6(a), the manual initialization with a single frame performs poorly and loses the target within a few frames (as expected). The second experiment produces better results since it computes an average model across the 5 frames and consequently the model learned is less noisy. However, even in this case, we can see that the target is lost after a few frames (see Fig. 6(b)). In the third experiment (where the target is automatically localized), the target is tracked the longest (see Fig. 6(c)). The strength of employing a MIL formulation here (as opposed to a supervised approach) is that the task of identifying the best representation of the target present in the frame (and one that is also the most persistent across all frames) is ambiguous, and hence is pushed into the MIL framework. Consequently, our MIL approach outperforms the alternate initialization methods.

## 7 Summary and Future Work

We proposed a novel MIL framework to perform target localization from image sequences in a surveillance setting. The approach consists of a softmax logistic regression MIL algorithm using log covariance features to automatically learn the model of a target that is persistent across all input frames. The learned target model can be updated online and the approach can also be used to learn multiple targets in the scene (if present). We performed experiments to demonstrate the validity and usefulness of the proposed approach to localize targets in various scenes. We also demonstrated the applicability of the approach to bootstrap conventional tracking systems and showed that automatic initialization using our technique helps achieve better performance. In future work, we plan to explore the applicability of this approach to multi-camera systems for learning models from multiple views of targets.

**Acknowledgement:** We gratefully acknowledge the support of the U.S. Department of Energy through the LANL/LDRD Program under LDRD-DR project 'RADIUS' for this work.

## References

1. Triggs, B.: Camera pose and calibration from 4 or 5 known 3d points. In: ICCV. (1999)
2. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean-shift. In: Proc. IEEE CVPR. (2000)
3. Porikli, F., Tuzel, O., Meer, P.: Covariance tracking using model update based on means on Riemannian manifolds. In: Proc. IEEE CVPR. (2006)
4. Kalman, R.E.: A new approach to linear filtering and prediction problems. Transactions of the ASME–Journal of Basic Engineering **82** (1960) 35–45
5. Arulampalam, A., Clapp, T.: A tutorial on particle filters for online nonlinear bayesian tracking. IEEE Transactions on Signal Processing **50** (2002) 174–188
6. Dietterich, T., Lozano-Perez, T.: Solving the multiple-instance problem with axis-parallel rectangles. In: Artificial Intelligence. (1997)
7. Maron, O., Lozano-Perez, T.: A framework for multiple instance learning. In: Proc. NIPS. (1998)
8. Zhang, Q., Goldman, S.: Em-dd: An improved multiple instance learning technique. In: Proc. NIPS. (2001)
9. Andrews, S., Hofmann, T.: Support vector machines for multiple instance learning. In: Proc. NIPS. (2003)
10. Settles, B., Craven, M., Ray, S.: Multiple-instance active learning. In: Proc. NIPS. (2007)
11. Nigam, K., Lafferty, J., McCallum, A.: Using maximum entropy for text classification. In: Proc. IJCAI. (1999)
12. Fletcher, R.: Practical methods of optimization. In: Unconstrained Optimization Chap. 3 Vol. 1. (1980)
13. Jost, J.: Riemannian geometry and geometric analysis,. In: Berlin: Springer-Verlag. (2002)