A Nonparametric Treatment for Location/Segmentation Based Visual Tracking

Le Lu*
Integrated Data Systems Department
Siemens Corporate Research

Gregory D. Hager
Computer Science Department
Johns Hopkins University

Abstract

In this paper, we address two closely related visual tracking problems: 1) localizing a target's position in low or moderate resolution videos and 2) segmenting a target's image support in moderate to high resolution videos. Both tasks are treated as an online binary classification problem using dynamic foreground/background appearance models. Our major contribution is a novel nonparametric approach that successfully maintains a temporally changing appearance model for both foreground and background. The appearance models are formulated as "bags of image patches" that approximate the true two-class appearance distributions. They are maintained using a temporal-adaptive importance resampling procedure that is based on simple nonparametric statistics of the appearance patch bags. The overall framework is independent of an specific foreground/background classification process and thus offers the freedom to use different classifiers. We demonstrate the effectiveness of our approach with extensive comparative experimental results on sequences from previous visual tracking [1, 12] and video matting [4] work as well as our own data.

1. Introduction

Visual tracking is an important computer vision problem that has received intensive study over the past two decades. At a general level, tracking involves two inter-related tasks: localizing a target in a frame of video given an appearance model and location estimates at previous frames, and adjusting the model of object appearance given its location. Despite extensive research, tracking remains difficult in cases where the foreground and background are similar in appearance , where there is a rapidly changing or deforming object appearance, and when the background is highly variable.

An even more challenging task is to accurately segment the target region from the background through a video sequence. This is known as video cutout and matting [4, 25, 17, 16, 23] in computer graphics community. Whereas localization can often rely on an approximate model of object shape (e.g. a rectangular region of interest), video cutout requires classification of then entire image as either foreground or background.

In this paper, we present a non-parametric framework for modeling the evolving appearance of image regions using nonparametric K-nearest-neighbor (*KNN*) [6] statistics, and we use this modeling framework to solve both localization and segmentation as a sequential binary classification problem. As such, our work is closely related to recent work on ensemble tracking [1], online density based tracking [8] and foreground-background texture discriminative tracking [20].

In our approach, we rely on descriptive (rather than descriminative) models. Descriptive appearance models are expected to capture all two-class image variations throughout the video volume. The dominating methods are either based on segmenting [25, 17, 16] using interactive 2D or 3D graph-cut technique [2] or interactive matting over hand-drawn trimaps [4] propagated by optical flow. Both methods involve a tremendous amount of manual interaction. Our approach formulates the two-class figure/ground appearance models in a nonparametric form of "bags of image patches." The method performs temporal-adaptive importance resampling procedure for both models, including a novel, robust process of bidirectional consistency checking from *KNN* statistics. For segmentation, we utilize the concept of "superpixels" [22, 16, 9, 10] to spatial-adaptively sample visually representative two-class random patches from a given image frame. We demonstrate that our proposed method can also provide a practically feasible, and fully automatic solution for the video object cutout or matting problem.

The remainder of this paper is organized as follow. In section 2, we address the differences and advantages of our method compared with previous work in location and segmentation based tracking. The proposed algorithm is then described in section 3, with details on learning nonparametric discriminative or descriptive appearance models for the two tasks respectively. Extensive experimental results and comparison to the state-of-the-art algorithms are provided

*The work was done when the first author was a graduate student in Johns Hopkins University.

1-4244-1180-7/07/$25.00 ©2007 IEEE

later by using the videos from [1, 12, 4] and our own data. Finally we conclude the paper and discuss several open issues and possible extensions.

2. Related work

In this paper, we address localization-based visual tracking [1, 8, 20] and segmentation-based video cutout [4, 23, 16, 17, 25]. Our methods are also related to, and relevant for, background subtraction [24, 19, 21] under static and dynamic environments.

Avidan [1] presents a method using an ensemble of simple weak classifiers for the binary foreground/background appearance model maintenance and tracking. Each weak classifier is trained online from a specific frame, and the ensemble is collected from a predefined range of recent frames. By design the ensemble is designed to capture the recent fixed-length foreground/background appearances. Not surprisingly, the tracker will fail when an extended occlusion happens, unless particle filtering or other temporal filtering methods are applied [1]. Our temporal appearance model operates directly on fine-grained data samples, ie. pools of simple color-texture features of sampled image patches. As such, model adaptation is driven by the feature matching and feature distinctiveness, not time. Therefore our model can handle arbitrarily long occlusions while rejecting new, unfamiliar observations in the occluded region. Another important point is that fitting a discriminative classifier as a representation of image appearance [1] may introduce bias directly into the appearance model. In our approach, appearance model maintenance is performed independent of the classification procedure. Indeed, several commonly used classification algorithms can be integrated with our method.

In other work, [8] utilizes mean shift mode-seeking algorithm [5] to maintain an online Gaussian mixture density model. The meanshift density model has potential difficulties with high dimensional image features which potentially limits its applicability. Nguyen and Smeulders [20] describe a classification-based object tracking approach relying on the online construction of target/background texture discriminant functions. However, as suggested above, the choice of discrimant functions (linear or nonlinear)may influence the tracker performance and introduce bias. Furthermore, a parametric formulation using a mean feature vector and covariance matrix to represent figure/ground appearances will have limited usefulness when figure/ground regions contain a large variety of visual patterns.

Interactively extracting a foreground object from an image [23, 16], or segmenting a moving object from a video sequence [17, 25] remains a difficult computer graphics task. State-of-the-art methods [23, 16, 17, 25] employ an interactive graph-cut algorithm [2] as a Markov random field solver to assign pixels with figure/ground labels using color cues. Such approaches still need a large amount of manual interaction and usually assume the camera is fixed. Our approach provides an automatic means to propagate segmentation labels over images by nonparametric appearance modeling. Recently, [13] proposed a hierarchical model switching method for unsupervised video segmentation. The methods involves variational inference over many conditional switching and conditional hidden variables. It is very computationally expensive and depends on creating a complex switching process among different global shape/appearance models.

Dynamically changing backgrounds render many of the above methods ineffective. In recent work, [24, 19] describe pixel-wise foreground detection algorithms to handle a quasi-static[1] background. This work relies on a local smoothing process on the pixels occupied by dynamic textures using a kernel density estimator in the joint spatial-color space. However, the approach does not handle the change in background due to a moving camera. Motion segmentation is another approach to find independent moving objects by computing an exact model of background motion [21]. Unfortunately it is only effective for segmenting small moving objects from a dominant background motion, mostly for aerial visual surveillance applications. By comparison, our treatment of image segments (instead of pixels) as the elements of foreground/background classification avoids the need for motion assumptions across images.

3. Algorithms

In this section, we present two slightly different tracking algorithms. Both use a model updating process with two parts: 1) classifying image patches and regions or segments with model matching and 2) updating models from newly classified image patches. We summarize these processes in Algorithms 1 and 2.

3.1. Location Tracking

Algorithm 1 performs location tracking by the steps of: 1) *image sampling* to generate figure/ground appearance representatives, 2) class-conditional *image-model matching* to generate a likelihood or confidence map, 3) *tracking* by high confidence/likelihood mode seeking [5, 1], and 4) *bidirectional consistency checking* and *resampling* for nonparametric appearance model updating.

We make use of the following notation. Let p denote an image patch, $\mathcal{P}$ denote a set of patches sampled from an image, and Ω denote a patch model. We use subscripts to denote time, and superscript F and B to denote foreground (target) and background. Thus, $\mathcal{P}_t^F$ denotes a set of patches sampled from the image at time t from the foreground. $\Omega_t^{F|B}$ represents the joint foreground/background model at

[1] A static scene with periodically changing objects, such as a running river, waving trees, or ocean waves and so on.

time t. Given a set of patches, $\mathcal{P}$, we define $knn(p, \mathcal{P})$ as the kth nearest neighbor of p in $\mathcal{P}$, where k is an $apriori$ fixed parameter. Finally, we denote a negative exponential function as $g(x; s) = exp(-x^2/s^2)$.

Algorithm 1 (Nonparametric Location Video Tracking Algorithm)

inputs: Images $\mathbb{X}_t, t = 1, 2, ..., T$; Location $\mathbb{L}_1$
outputs: Locations $\mathbb{L}_t, t = 2, ..., T$; 2 "bags of patches" appearance model for foreground/background $\Omega_T^{F|B}$

1. Sample image patches $\mathcal{P}_1$ from image $\mathbb{X}_1$.
2. Construct 2 bags of patches $\Omega_1^{F|B}$ for using patches $\mathcal{P}_1$ with labels inferred from their position relative to given foreground/background windows about $\mathbb{L}_1$; set $t = 1$.
3. Train a binary classifier (see text for examples) $\mathcal{C}_t$ with probability or confidence output using $\Omega_t^{F|B}$.
4. Matching and Tracking:
   (a) Sample image patches $\mathcal{P}_{t+1}$ from image $\mathbb{X}_{t+1}$;
   (b) Input $\mathcal{P}_{t+1}$ into $\mathcal{C}_t$ and output the normalized positive-class (foreground) confidence map;
   (c) Run meanshift [5] on the confidence map from $\mathbb{L}_t$ to locate the position of converged peak as $\mathbb{L}_{t+1}$.
5. Bidirectional Consistency Check and Model Update:
   (a) Classify $\mathcal{P}_{t+1}$ against $\Omega_t^{F|B}$ and filter by rejecting ambiguous, redundant, and outlier patch samples.
   (b) Incorporate the filtered $\mathcal{P}'_{t+1}$ into $\Omega_t^{F|B}$ producing $\Omega_{t+1}^{F'|B'}$
   (c) Evaluate the "probability of survival" $Pr_s^{F|B}$ for all patches $p' \in \Omega_{t+1}^{F'|B'}$ relative to $\mathcal{P}_{t+1}$.
   (d) Resample $\Omega_{t+1}^{F'|B'}$ according to the "probability of survival" to generate $\Omega_{t+1}^{F|B}$.
6. Update $t = t + 1$; If $t = T$, output $\mathbb{L}_t, t = 2, ..., T$ and $\Omega_T^{F|B}$; exit. If $t < T$, go to (3).

Sampling: In the case of location tracking, we model the appearance of the target within a fixed foreground window, and the appearance of the background within a surrounding "context window." As illustrated in figure 1, in the first image we extract image patches $\mathcal{P}_t^F$ and $\mathcal{P}_t^B$ from the figure and background[2] regions, respectively. We then create initial foreground and background "appearance bags" to initialize an appearance model $\Omega_1^{F|B}$. In subsequent frames $t > 1$, we sample within the outer context rectangle predicted from the previous frame to produce a mixed sample set $\mathcal{P}_t$. We have found that evenly scanning or randomly sampling patches gives very similar final tracking performance provided the sampling rate (the ratio between the number of sampled patches and the number of all spatially eligible patches) is similar (normally $2\% \sim 8\%$).

[2]An image patch sampled across the figure/ground boundary is placed into a class one of the two classes only when more than 70% percent of its area is contained in a given class; it is otherwise rejected.

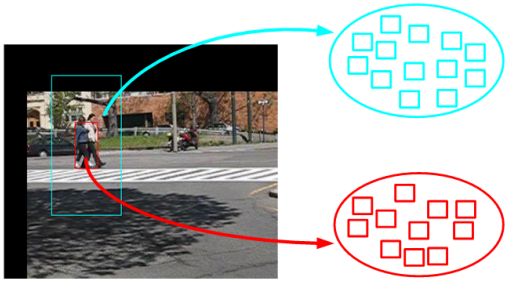

Figure 1. Image patch sampling from foreground/background regions. The red/cyan patches and rectangles represent the figure/ground patches and regions, respectively.

Matching: Given a model $\Omega_t^{F|B}$ at time t we train a figure/ground binary classifier $\mathcal{C}_t$, and use this to classify patch samples $\mathcal{P}_{t+1}$. Obtaining $\mathcal{C}_t$ from $\Omega_t^{F|B}$ is not dependant on the model updating process itself or on previous classifiers $\mathcal{C}_{t'}, t' < t$; all appearance model history is contained in $\Omega_t^{F|B}$. For location estimation, we do not make hard decisions over $\mathcal{P}_{t+1}$, but instead use a measure of classification confidence. Thus, any classification algorithm with reasonable performance and which produces confidence outputs can be employed.

In this paper, we present results from three different[3] classification algorithms: *KNN*, *PCA+KDE*, and *SVM*. The confidence values are computed as follows. For *KNN*, for each image patch $p \in \mathcal{P}_{t+1}$, we define

$$d_p^x = \|p - knn(p, \Omega_t^x)\|, \quad x \in \{F, B\} \tag{1}$$

as the distance from p to its k-th nearest neighbor in the bag of patches Ω_t^F and Ω_t^B, respectively. Let σ_d denote the standard deviation of the values $\{d_p^F, d_p^B | p \in \mathcal{P}_{t+1}\}$. The normalized foreground likelihood value of $p \in \mathcal{P}_{t+1}$ is then defined as

$$l_p^F = \frac{g(d_p^F; \sigma_d)}{g(d_p^F; \sigma_d) + g(d_p^B; \sigma_d)} \tag{2}$$

In practice, we choose $k = 2$ or 3 and k is insensitive within $2 \sim 10$. For KDE, we first perform dimensional reduction using Principal Component Analysis (PCA) [6] to map the patches in $\Omega_t^{F|B}$ into a lower dimensional subspace suitable for kernel density estimation (KDE) [11]. We then build figure and ground KDEs from PCA mapped features. Let $kde(\cdot; \Omega_t^x)$ denote the likelihood function for foreground ($x = F$) and background ($x = B$), respectively. As above,

[3]In fact, in addition to these three we have experimented with *LDA+KDE* and *NDA+KDE* however there were no significant differences [18] which focused on evaluating object-level image matching across multiple viewpoints.

we then compute the normalized foreground likelihood of $p \in \mathcal{P}_{t+1}$ as

$$l_p^F = \frac{kde(p;\Omega_t^F)}{kde(p;\Omega_t^F) + kde(p;\Omega_t^B)} \quad (3)$$

Finally, following [3], a Support Vector Machine (SVM) is trained by using $\Omega_t^{F|B}$, and tested over all image patches at $t+1$ to produce class labels and confidence values. Since the SVM produces both positive and negative values, we compute likelihood by truncating negative confidences (background) to zero and rescaling positive confidence values in the range of $[0, 1]$.

Tracking: As in [1], we map each patch foreground likelihood/confidence value l_p^F onto p's image coordinates to create a confidence response map (CRM) (figure 2 (c),(d)). We then run the mean-shift algorithm [5] from $\mathbb{L}_t$ to locate the mode of this map and assign it as the object position $\mathbb{L}_{t+1}$

Model Updating: Patches from $\mathcal{P}_{t+1}$, are integrated, using knn distances, to compute an updated model $\Omega_{t+1}^{F|B}$ as follows. First, ambiguous patch samples, defined as all $p \in \mathcal{P}_{t+1}$ such that $0.8 \leq d_p^F/d_p^B < 1/0.8$ are discarded. The remaining patches are retained and classified as foreground or background yielding sets $\mathcal{P}_{t+1}^F$ and $\mathcal{P}_{t+1}^B$, respectively. These sets are trimmed by requiring, for each $p \in \mathcal{P}_{t+1}^{F|B}$, that

$$\overline{d}_{t+1}^x - \lambda_1 * \sigma_{t+1}^x \leq d_p^x \leq \overline{d}_{t+1}^x + \lambda_2 * \sigma_{t+1}^x \;\; x \in \{F, B\} \quad (4)$$

where $\overline{d}_{t+1}^x$ and σ_{t+1}^x denote the mean and standard deviation of the knn distances of all patches in $\mathcal{P}_{t+1}^x$. Patches with small distances are very similar to the current model $\Omega^{F|B}$ (and are thus redundant), while patches with large distances are likely to be outliers. λ_1 and λ_2 (both $1.0 \sim 2.0$ in our experiments) control the *model rigidity* (ie. variation tolerance during model's temporal evolution). After filtering, the resulting patch sets are denoted as $\mathcal{P}_{t+1}^{F'|B'}$, respectively, and are referred to as the "filtered" sets.

Finally, an initial updated model is computed as $\Omega_{t+1}^{F'|B'} = \Omega_{t+1}^{F|B} \cup \mathcal{P}_{t+1}^{F'|B'}$. This new model is resampled to form $\Omega_{t+1}^{F|B}$. To do so, for each patch $p' \in \Omega_{t+1}^{F'}$, we first compute its knn distance *back to* the unfiltered $\mathcal{P}_{t+1}^F$

$$d_{p'}^F = dist(p', knn(p', \mathcal{P}_{t+1}^F)) \quad (5)$$

A small distance thus indicates the patch is present in the current image. We again convert this distance to a probability as

$$p_{p'}^F = g(d_{p'}^F; \sigma'^F)/w \quad (6)$$

where σ'^F is the standard deviation of distance over all $p' \in \Omega_{t+1}'^F$ and $w = \sum_{p'} g(d_{p'}^F; \sigma'^F)$. Finally, we sample each $p' \in \Omega_{t+1}'^F$ with probability $\min(m \times p_{p'}^F, 1)$ (denoted as Pr_s^F for all probilities), given a fixed nominal model size m. The resulting samples form Ω_{t+1}^F. Similarly, we obtain Pr_s^B and Ω_{t+1}^B.

By approximately fixing the model size m, the expected number of image patches retained from time t in the model decreases exponentially over time, thus allowing the model to adapt to new appearance. Although this also leads to potential confusion between foreground and background, we have not found this to be the case, as shown in Section 4. This is in part because the algorithm effectively evaluates new patches against the model, and the model against new patches, retaining only those that are clearly classified and mutually consistent. We refer to this step as the *bidirectional consistency check*.

3.2. Segmentation Tracking

Segmentation-based tracking differs in that we now attempt to clearly and accurately demarcate the target region in the image. We assume we are supplied with one (or more) annotated frames, and our goal is to propagate these labels to segment and classify other occurrences of figure/ground in the video. The critical difference is that segmentation tracking must maintain a complete (for accuracy) appearance representation for all possible complex visual patterns appearing in the foreground or background region. Some examples are shown in figures 6,7,8.

Algorithmically, rather than operating on individual pixels, we first partition each video frame into segments or "superpixels" [22, 16, 9] using a standard algorithm [7]. We then pose the tracking problem as one of classifying the resulting segments.

Sampling: We denote an image segment as $\mathcal{S}_t^i$ where i is its index within the image $\mathbb{X}_t$. Let $\mathcal{P}_t^i$ represent a set of random image patches sampled from $\mathcal{S}_t^i$. The number of all possible image patches of an image segment $\mathcal{S}_t^i$, denoted $\mathcal{N}_t^i$, typically ranges from dozens to thousands. However, given these are the output of a segmentation algorithm, small or large segments are expected to have roughly the same amount of visual uniformity. Therefore the size of $\mathcal{P}_t^i$ is fixed as the smaller of a fixed proportion ($1\% \sim 6\%$) of $\mathcal{N}_t^i$ or a predefined limit ($150 \sim 250$). In practice, this adaptive spatial sampling strategy is sufficient to represent image segments of differing sizes while keeping the sizes of "bags of image patches" manageable. By comparison, directly scanning or random sampling as in algorithm 1 is less likely to be representative of appearance since it "wastes" samples in large areas of low texture and may lack representatives from small, uniquely appearing image regions.

Matching: We classify any new image segment $\mathcal{S}_{t+1}^i$ based on the classification result of its representatives $\mathcal{P}_{t+1}^i$ to the figure/ground appearance models $\Omega_t^{F|B}$. To do so, for each patch $p \in \mathcal{P}_{t+1}^i$, we calculate its KNN distance d_p^F and d_p^B to $\Omega_t^{F|B}$. The decision of assigning $\mathcal{S}_i$ to F

Figure 2. Image-model matching with confidence response output and tracking using mode seeking [5, 1]. (a) Frame t, (b) Frame t+1, (c) Confidence response map (CRM) within the searching window at t+1 by SVM matching [3], (d) Confidence response map (CRM) of the final figure/ground window after mean-shift tracking [5]. CRM is coded in colored 3D mesh in (c) and intensity in (d). More red color or brighter intensity represents higher confidence/likelihood, and vice versa.

Algorithm 2 (Nonparametric Segmentation Video Tracking Algorithm)

inputs: Pre-segmented Images $\mathbb{X}_t, t = 1, 2, ..., T$; Label $\mathbb{L}_1$
outputs: Labels $\mathbb{L}_t, t = 2, ..., T$; 2 "bags of patches" appearance model for foreground/background $\Omega_T^{F|B}$

1. Sample segmentation-adaptive random image patches $\mathcal{P}_1$ from image $\mathbb{X}_1$.
2. Construct 2 new bags of patches $\Omega_1^{F|B}$ for foreground/background using patches $\mathcal{P}_1$ and label $\mathbb{L}_1$; set $t = 1$.
3. Sample segmentation-adaptive random image patches $\mathcal{P}_{t+1}$ from image $\mathbb{X}_{t+1}$; match $\mathcal{P}_{t+1}$ with $\Omega_t^{F|B}$ and classify segments of $\mathbb{X}_{t+1}$ to generate label $\mathbb{L}_{t+1}$ by aggregation.
4. Perform bidirectional consistency check to get $\Omega_{t+1}^{F'|B'}$.
5. Perform the random partition and resampling process according to the probability of survival $Pr_s^{F|B}$ (integrated with a partition-wise sampling rate γ) inside $\Omega_{t+1}^{F'|B'}$ to generate $\Omega_{t+1}^{F|B}$.
6. Update $t = t + 1$. If $t = T$, output $\mathbb{L}_t, t = 2, ..., T$ and $\Omega_T^{F|B}$; exit. If $t < T$, go to (3).

or B, is made by comparing the mean or median distance over all patches, and choosing the class yielding the smaller value. *Majority voting* of sampled image patch classification decisions has also been tested. In our evaluations, all three operators produce similar results, although Median is sometimes slightly superior under very noisy conditions.

As an option, we can use the *Kernel Density Estimator* (KDE) [11] for segmentation-based tracking. In this case, rather than distances $\{d_p^F$ and $d_p^B\}$ we compute likelihood values m_p^F and m_p^B for aggregation and voting. This option becomes necessary when enforcing shape model constraint later.

Model Updating: In segmentation tracking, $\Omega^{F|B}$ normally has a complex multimodal distribution. If we perform resampling uniformly, as in the previous section, some modes of the appearance distribution may be mistakenly removed. Instead, we introduce an additional partitioning factor γ into the final "probability of survival" calculation. We first cluster $\Omega_{t+1}^{F'|B'}$ into several subgroups using the Kmeans algorithm [6]. Let n_c denote the number of members of cluster c. If p' falls in cluster c, we associate to it a factor $\gamma_{p'} = (1/n_c)^{\frac{1}{2}}$. The probability of survival for a patch p' is then

$$p_{p'}^F = \gamma_{p'} g(d_{p'}^F; \sigma'^F)/w \tag{7}$$

where σ'^F is the standard deviation of distance over all $p' \in \Omega_{t+1}'^F$ and $w = \sum_{p'} \gamma_{p'} g(d_{p'}^F; \sigma'^F)$. We do the same for the background model. Subsequent resampling proceeds as before.

Shape Model: We also make use of a weak shape model for segmentation based tracking. A weak shape model is expected to solve the ambiguity of indistinguishable figure/ground matching by pure appearance, but enforce only weak shape constraints that tolerate rapid motion (as shown in figure 6). For each video frame t, we place spatial (Gaussian) kernels over sampled patch locations weighted by their two-class KDE matching scores $\{m_p^F, m_p^B\}$, respectively. A *Kernel Density Estimator* from [11] is used to generate a shape density map for each of foreground and background. At the next frame $t + 1$, newly sampled image patches are classified using the product of their appearance KDE likelihood and their shape KDE likelihood. We refer to the product of these two likelihoods as the response map (*PRM*); an example shown in figure 7.

4. Experiments

We have tested both our location and segmentation tracking algorithms using dozens of videos from the past literature on tracking and video matting as well as our own data. Due space limitations, we illustrate selected results in figures 3,4, [1], figure 5 [12], figures 6,7,8 [4], figure 9. Refer to http://www.cs.jhu.edu/∼lelu/NonparametricTracking/ for more examples.

There are many types of applicable patch image fea-

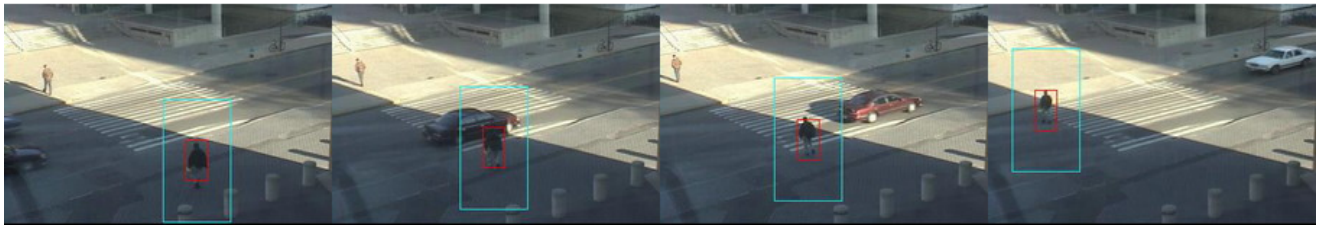

Figure 3. Walking person tracking in a low figure-ground contrast, low resolution, 120-frame long surveillance video [1] (*tr-ped1.mpg*) where the red/cyan rectangles represent the figure/ground regions respectively. We show example frames as 1, 31, 51, 120. Notice that the person's leg part has similar color to the ground; and the person's upper part is also similar to the passing car.

Figure 4. Two people captured with a moving camera. In this 141-frame long video [1] (*tr-ped2.mpg*) the red/cyan rectangles represent the figure/ground regions respectively. We show, as examples, frames 1, 22, 66, 140. Notice that there are dramatic figure/ground appearance pattern changes in this video.

tures. For the results demonstrated in this paper, we use *Color/Intensity + HOG* [15, 1] for location tracking (allowing direct comparison with prior results) and color-intensity vectors when performing segmentation tracking. Except where otherwise noted, we apply *PCA* to for dimensional reduction. For a more detailed discussion on comparisons between other features and other dimensional reduction methods, refer to [18].

Based on our experiments, algorithm 1 has performance that meets or exceed that of [1] (figures 3, 4 and another example *tr-car2.mpg*). These results are achieved while operating on the original video a full resolution instead of a three-layered Gaussian Pyramid [1]. It is also successfully employed for a 1145-frame "Dudek" sequence which was used to show the superior performance of the *WSL* algorithm [12].

We further compare algorithm 1 with [1] in the following two aspects. First, algorithm 1 is an open framework and any classifier can be integrated for the binary figure/ground classification. To illustrate the effect of different classifiers, we run Dudek sequence 10 times using LDA [6] and SVM [3], respectively. Due to randomness in our sampling and resampling process, the tracking results are similar but not exactly the same. LDA succeeds in all 10 trials, but SVM fails once (out of 10) at frame 594 when illumination becomes darker. Through, this is a special case, it is desirable to make the tracker compatible with all off-the-shelf classification techniques. Second, the effectiveness of our nonparametric model updating method is further proved in figures 5 and 6 by its capacity to model unique, long-term appearances history for re-acquisition. It is thus more flexible than the fixed-term appearance model of [1].

To illustrate algorithm 2, we present our video cutout results using three sequences of [4]. From figure 6,7,8, our algorithm outputs quite reasonable foreground/background masks under challenging conditions of smoking, rapid out-of-plane head rotations and hand motions, and a rapidly panning camera. Typically, the underlying image segmentation algorithm generates natural image partitioning boundaries which is important to our segment-wise figure/ground labeling. In cases where the image segments span the boundary of foreground and background, labeling errors are unavoidable. However the mislabeling artifacts does not appear to influence the robustness of our tracker. The reason is that the model matching and updating processes are performed at the fine-grained image patch level, which produces and works on smooth probability response maps (as shown in figure 7). By maintaining robust patch-based appearance models, our tracker can tolerate local segmentation artifacts.

Finally, our algorithms are focusing on visual appearance (with a weak shape model for algorithm 2) based tracking, thus are not designed to solve all types of tracking problems. In figure 9, our tracker will locate one of the modes in the confidence response map when two visually very similar cars appearing in the target window. The selected mode is not guaranteed to be the right target. In this case, a simple motion smoothness constraint (such as constant-velocity) can be used to predict the original car easily.

5. Conclusion and Discussion

We have presented a framework for tracking and segmenting target regions with a complex, changing appearance and dynamically changing backgrounds. The frame-

Figure 5. Face tracking in an office environment from a 1145-frame long video [12] (*tr-dudek.mpg*) where the red/cyan rectangles represent figure/ground regions, respectively. We show example frames 2, 93, 210, 364, 446, 567, 680, 751, 962, 1145. Besides the interesting appearance variations listed in [12], 19 image frames were corrupted (from 554# to 572#) while downloading. Our model automatically rejects the corrupted image patch samples due to their visual unfamiliarity, and the tracker locks the face when it appears again within the search region.

Figure 6. Human segmentation tracking in a static background from a 176-frame long video [4] (*vm-adam.mpg*). We show example frames 1, 40, 43, 110, 154. Notice that the subject has large out-of-plane rotation and rapid hand motion (as from 40# to 43#) through the example video. Our model also demonstrates its long-term appearance modeling capacity. For instance, the image patch samples from the smoky regions captured around frame 43# can be temporally propagated and maintained to recognize other smoky occurrences 60 frames later (around 110#).

work employs sampled "bags of patches" to represent appearance. These models are updated using robust KNN statistics and employs a novel bidirectional consistency check to ensure model updates are performed consistently. Our experimental results with the method are compelling, and have also shown that the overall framework is relatively insensitive to many choice of parameters and/or classification method.

Finally, to our knowledge, algorithm 2 provides the first fully-automatic video cutout [4, 23, 16, 17, 25] method once provided with the annotation of the first frame. Given this initial segmentation labels, there are many methods (such as pairwise random field model [10, 14], multi-level image segmentation [10], supervised segmentation hypotheses [14] or matting [4, 17, 17, 23]) to produce improved figure/ground image boundaries; we leave this as future work.

References

[1] S. Avidan, Ensemble Tracking, CVPR 2005.

[2] Y. Boykov and M. Jolly, Interactive Graph Cuts for Optimal boundary and Region Segmentation of Objects in n-d Images, ICCV, 2001.

[3] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines, http://www.csie.ntu.edu.tw/∼cjlin/libsvm, 2001.

[4] Y.-Y Chuang, A. Agarwala, B. Curless, D. Salesin and R. Szeliski, Video Matting of Complex Scenes, Proceedings of *SIGGRAPH* 2002, San Antonio.

[5] D. Comaniciu and P. Meer, Mean shift: A robust approach toward feature space analysis. *IEEE Trans. PAMI*, 2002.

[6] R. Duda, P. Hart and D. Stork, Pattern Classification (2nd ed.), Wiley Interscience, 2001.

[7] P. Felzenszwalb and D. Huttenlocher, Efficient Graph-Based Image Segmentation, *IJCV*, 2004.

[8] B. Han and L. Davis, On-Line Density-Based Appearance Modeling for Object Tracking, ICCV 2005.

[9] D. Hoiem, A. Efros and M. Hebert, Geometric Context from a Single Image, ICCV 2005.

[10] D. Hoiem, A. Efros and M. Hebert, Recovering Surface Layout from an Image, *IJCV*, 2006.

[11] A. Ihler, Kernel Density Estimation Matlab Toolbox, http://www.ics.uci.edu/∼ihler/code/kde.shtml, 2004.

[12] A. Jepson, D. Fleet and T. El-Maraghi, Robust Online Appearance Models for Visual Tracking, CVPR 2001.

[13] N. Jojic, J. Winn and L. Zitnick, Escaping Local Minima through Hierarchical Model Selection: Automatic Object Discovery, Segmentation, and Tracking in Video, CVPR 2006.

[14] S. Kumar and M. Hebert, A Hierarchical Field Framework for Unified Context-Based Classification, ICCV, 2005.

Figure 7. Human segmentation tracking in a moving background from a 91-frame long video [4] (*vm-amira.mpg*). We show example frames 3, 35, 60, 89. For 89, we show the associated probability response map (PRM). The map is cubically interpolated from the semi-dense responses of sampled random visual patches in the image coordinates. The response strength is coded as color, blue (cold-color) represents foreground and red (warm-color) represents background. Notice the probability map clearly shows the figure/ground classification separation and the blurry responses around the hair area. Both figure and ground have multimodal appearance distributions, but are well-separated into classes. More accurate PRMs generated from our method could be used to replace the trimap (masks of figure, ground and boundary) as the input to image and video matting algorithms [4, 23, 16] for more appealing visual effects.

Figure 8. Human segmentation tracking in a rapidly panning background from a 111-frame long video [4] (*vm-kim.mpg*). We show two pairs of example frames as 19 and 105. In each pair, the original image is shown on the left; the cutout figure image regions from classification is displayed on the right. Our approach demonstrates its ability to successfully adapt with the complex, changing background while preserving the complete foreground visual patterns.

[15] K. Levi and Y. Weiss, Learning Object Detection from a Small Number of Examples: The Importance of Good Features, CVPR, 2004.

[16] Y. Li, J. Sun, C.-K. Tang and H.-Y. Shum, Lazy Snapping, *SIGGRAPH*, 2004.

[17] Y. Li, J. Sun and H.-Y. Shum. Video Object Cut and Paste, *SIGGRAPH*, 2005.

[18] L. Lu and G. Hager, Dynamic Foreground/Background Extraction from Images and Videos using Random Patches, NIPS 2006.

[19] A. Mittal and N. Paragios, Motion-based Background Substraction using Adaptive Kernel Density Estimation, CVPR, 2004.

[20] H.T. Nguyen and A. W.M. Smeulders, Robust Tracking using Foreground-Background Texture Discrimination, *International Journal on Computer Vision*, 2006.

[21] R. Pless, T. Brodský and Y. Aloimonos, Detection Independent Motion: The Statistics of Temporal Continuity, *IEEE Trans PAMI*, 2000.

[22] X. Ren and J. Malik, Learning a classification model for segmentation, ICCV, 2003.

[23] C. Rother, V. Kolmogorov and A. Blake, Interactive Foreground Extraction using Iterated Graph Cuts, *SIGGRAPH*, 2004.

[24] Yaser Sheikh and Mubarak Shah, Bayesian Object Detection in Dynamic Scenes, CVPR, 2005.

[25] J. Wang, P. Bhat, A. Colburn, M. Agrawala and M. Cohen, Interactive Video Cutout. *SIGGRAPH*, 2005.

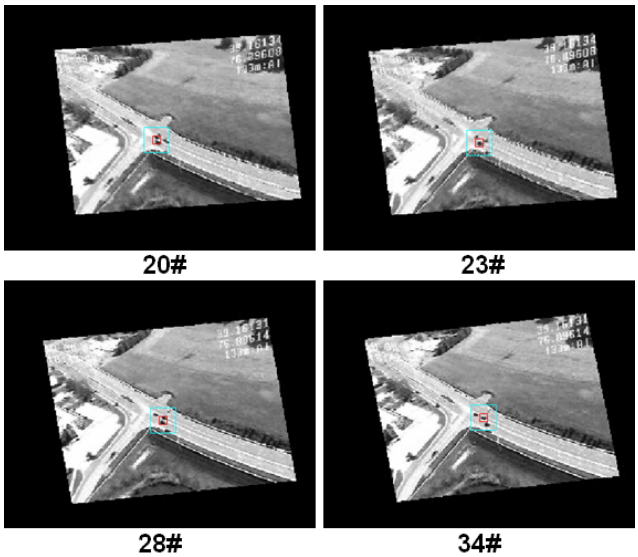

Figure 9. An example of appearance mode-shifting in an aerial vehicle tracking video (*tr-aerial-seq1.mpg*) . The target car is moving from left to right and passing two very similar-looking cars driving in the opposite direction. Because the indistinguishable visual appearance, the class-conditional response map has (unsurprisingly) two peaks. From the previous target position, mean-shift [5] algorithm converges to the closest peak. This explains the reason why the tracker locks the correct target in the first passing (20#, 23#), but fails in the second (28#, 34#). A simple motion dynamic model (such as constant-velocity) can be employed to solve this ambiguity easily.