Online Selection of Discriminative Tracking Features

Robert T. Collins, *Senior Member*, *IEEE*, Yanxi Liu, *Senior Member*, *IEEE*, and Marius Leordeanu

Abstract—This paper presents an online feature selection mechanism for evaluating multiple features while tracking and adjusting the set of features used to improve tracking performance. Our hypothesis is that the features that best discriminate between object and background are also best for tracking the object. Given a set of seed features, we compute log likelihood ratios of class conditional sample densities from object and background to form a new set of candidate features tailored to the local object/background discrimination task. The two-class variance ratio is used to rank these new features according to how well they separate sample distributions of object and background pixels. This feature evaluation mechanism is embedded in a mean-shift tracking system that adaptively selects the top-ranked discriminative features for tracking. Examples are presented that demonstrate how this method adapts to changing appearances of both tracked object and scene background. We note susceptibility of the variance ratio feature selection method to distraction by spatially correlated background clutter and develop an additional approach that seeks to minimize the likelihood of distraction.

Index Terms—Computer vision, tracking, time-varying imagery, feature creation, feature evaluation and selection.

1 Introduction

Two decades of vision research have yielded an arsenal of powerful algorithms for object tracking. Multiple moving objects can be effectively tracked in real-time from stationary cameras using frame differencing or adaptive background subtraction combined with simple data association techniques [1], [6], [26]. These detect-then-track approaches can be generalized to situations where apparent camera motion is easily stabilized, including purely rotating and zooming cameras and aerial views where scene structure is approximately planar [14]. Modern appearance-based methods use gradient descent to incrementally follow a reference object model through video without prior knowledge of scene structure or camera motion. This includes the use of flexible template models [8], [21] and kernel-based methods that track nonrigid objects using viewpoint-insensitive histograms [7], [10]. Kalman filter extensions achieve more robust tracking of maneuvering objects by introducing statistical models of object and camera motion [3], [16]. Tracking through occlusion and clutter is achieved by reasoning over a state-space of multiple hypotheses [15], [23], [24].

Our experience with a variety of tracking methods can be summarized simply: Tracking success or failure depends primarily on how distinguishable an object is from its surroundings. If the object is very distinctive, we can use a simple tracker to follow it. If the object has low-contrast or is camouflaged, we will obtain robust tracking only by imposing prior knowledge about scene structure or expected motion, thus buying tracking success at the price of reduced generality.

The degree to which a tracker can discriminate object and background is directly related to the image features used. Surprisingly, most tracking applications are conducted using a fixed set of features, determined a priori. Sometimes, preliminary experiments are run to determine which fixed features to use—a good example is work on head tracking using skin color, where many papers evaluate different color spaces to find one in which pixel values for skin cluster most tightly, e.g., [30]. However, these approaches ignore the fact that it is the ability to distinguish between object and background that is most important and the background cannot always be specified in advance. Furthermore, both foreground and background appearance will change as the target object moves from place to place, so tracking features also need to adapt. Fig. 1 illustrates this observation with low contrast imagery of a car traveling through patches of sunlight and shadow. The best feature for tracking the car through sunlight performs poorly in shadow and vice versa.

A key issue addressed in this work is online, adaptive selection of appropriate features for tracking. Target tracking is cast as a local discrimination problem with two classes: foreground and background. Our insight is that the features that best distinguish between object and background are the best features for tracking. This point of view opens up a wide range of pattern recognition feature selection techniques that can be adapted for use in tracking. An interesting characteristic of target tracking is that foreground and background appearances are constantly changing, albeit gradually. Naturally, when class appearance varies, the most discriminating set of features

- *R.T. Collins is with the Computer Science Engineering Department, The Pennsylvania State University, University Park, PA 16802. E-mail: rcollins@cse.psu.edu.*
- *Y. Liu is with The Robotics Institute and the Center for Automated Learning and Discovery, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213. E-mail: yanxi@cs.cmu.edu*
- *M. Leordeanu is with The Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213. E-mail: mleordea@andrew.cmu.edu.*

Manuscript received 9 July 2004; revised 14 Dec. 2004; accepted 19 Jan. 2005; published online 11 Aug. 2005.
Recommended for acceptance by C. Taylor.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0343-0704.

0162-8828/05/$20.00 © 2005 IEEE Published by the IEEE Computer Society

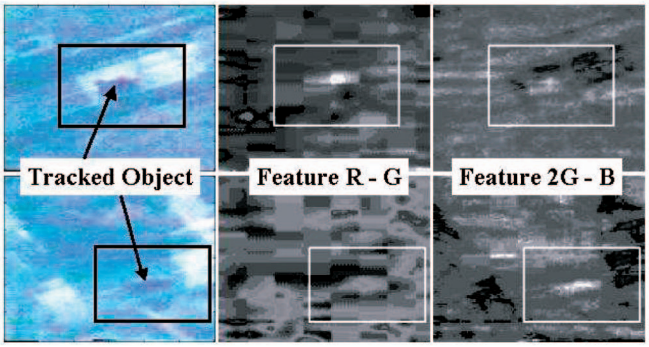

Fig. 1. Features used for tracking an object must be adapted as the appearance of the object and background changes. The source imagery (left column) is low contrast aerial video of a car on a road. The car travels between sunny patches (top row) and shadow (bottom row). The best feature for tracking the car in sunlight (R-G) performs poorly in shadow. Similarly, the best feature for tracking through shadow (2G-B) does not perform as well in sunlight.

also varies [19]. The issue of online feature selection has rarely been addressed in the literature, especially under the hard constraint of speed required for target tracking. The nearest relevant work is [27], which dynamically switches between five predetermined color spaces to improve face tracking performance.

Section 2 presents a brief look at offline discriminative feature selection in the field of pattern classification. Section 3 adapts these ideas to the task of target tracking. Since the goal is to perform online feature selection while tracking, efficiency must be favored over optimality. We develop an online feature selection mechanism using the two-class variance ratio to find features whose distributions best discriminate between the tracked object and the surrounding scene background. Examples are presented in Section 4 to illustrate how combining this feature selection mechanism with tracking facilitates adaptation to changing object and background appearance. In Section 5, we note that feature selection via the variance ratio may perform poorly in the presence of spatially-correlated background clutter, and we develop an alternate feature evaluation method that makes better use of spatial information to minimize distraction due to clutter. The new evaluation function favors features that minimize the influence of the maximum distractor, thus maximizing the likelihood of tracking the correct object in the next frame. Section 6 concludes with a discussion of issues raised by the approach presented in this paper.

2 Feature Selection

Feature selection is a technique for dimensionality reduction whereby a set of m features is chosen from a pool of n candidates, where usually $m << n$ [2]. This technique can improve classification performance by discarding irrelevant or redundant features.

The two main components in feature selection are the selection criterion function, which is a quantitative measure used to compare one feature subset against another, and the search strategy, which is a systematic procedure to enumerate candidate feature subsets and to decide when to stop. Criterion functions can be categorized by whether the evaluation process is data intrinsic (filters) or classifier-dependent (wrappers) [2]. For discrimination problems, the criterion involves evaluation of the discriminating power of the selected feature subset. There are many ways to evaluate the discriminative power of a feature. For example, augmented variance ratio (AVR) has been used for feature ranking as a preprocessing step for feature subset selection [18], [19], [20]. AVR is the ratio of the between-class variance of the feature to the within-class variance of the feature. Other measures of discriminative power include information gain and mutual information.

The goal in feature subset selection is to find m features that best complement each other for the classification task at hand. Since we usually do not know what the best subset size m should be, the search space for feature subsets is 2^n, where n is the total number of features. Existing heuristic search methods for feature selection provide a set of compromises between speed and optimality. For example, Sequential Forward Selection [2] has linear computational complexity in n, but it is a greedy strategy that can result in suboptimal feature sets. In biomedical imaging, a combination of feature ranking and feature subset selection has been shown to be effective for offline selection of discriminative subsets from thousands of feature candidates [20], [31]. However, to achieve online selection, we are forced to consider simplified selection criteria, nonexhaustive search spaces, and heuristic search strategies. In this work, we find the best m features individually, fully realizing that the best m individual features may not form the best feature subset of size m [28].

3 Feature Selection for Tracking

Our goal in this section is to develop an efficient method that continually evaluates and updates the set of features used for tracking. It is important to note that features used for tracking need only be *locally* discriminative, in that the object only needs to be clearly separable from its immediate surroundings. This is a much less restrictive assumption than is necessary for a tracker that uses a fixed set of features, since that set must, by necessity, be discriminative across a wide-range of imaging conditions. A tracker that swaps features in and out on the fly can instead use features that are finely tuned to provide good foreground/background discrimination, even if they are only locally, and temporarily, valid.

The following steps are taken in our approach. First, a set of candidate "seed" features are defined and the distributions of feature values for object and background classes are computed using samples taken from the most recently tracked frame. Second, the class conditional distributions for each feature are combined using a log likelihood ratio to produce a function that maps feature values associated with the object to positive values and feature values associated with the background to negative values. This important step can be interpreted as a nonlinear transformation of each seed feature into a new "tuned" feature that is tailored to the task of discriminating object from background in the current frame. Third, these tuned candidate features are evaluated using the two-class variance ratio to measure separability of the distributions they induce on object and background classes. Fourth, the most discriminative features are used to assign weight values to pixels in a new video frame, producing a weight image where object pixels have high values and background pixels have low values. Finally, the mean-shift algorithm is applied to this weight image surface to estimate the 2D location of the object in the current frame. Each of these steps is described in more detail in the sections below.

3.1 Seed Features

In principle, a wide range of features could be used for tracking, including color, texture, shape, and motion. Each potential feature set typically has dozens of tunable parameters and, therefore, the full number of potential features that could be used for tracking is enormous. In this work, we represent target appearance using histograms of color filter bank responses applied to R, G, B pixel values within local image windows. This representation is chosen since it is relatively insensitive to variations in target appearance due to viewpoint, occlusion, and nonrigidity. Although only color features are considered in this paper, the proposed approach can be extended easily to other cues represented as histograms of feature values.

The set of seed candidate features is composed of linear combinations of camera R, G, B pixel values. Specifically, for our experiments, we have chosen the following set of feature candidates:

$$\mathcal{F}_1 \equiv \{w_1 R + w_2 G + w_3 B \mid \; w_* \in [-2, -1, 0, 1, 2]\}, \qquad (1)$$

that is, linear combinations composed of integer coefficients between -2 and 2. The total number of such candidates would be 5^3, but, by pruning redundant coefficients where $(w_1', w_2', w_3') = k(w_1, w_2, w_3)$ and by disallowing $(w_1, w_2, w_3) = (0, 0, 0)$, we are left with a pool of 49 features. This set of seed features is chosen because: 1) the features are efficient to compute (only integer arithmetic is involved), 2) the features approximately uniformly sample the set of 1D subspaces of 3D RGB space, and 3) many common features from the literature are included in the candidate space, such as raw R, G, and B values, intensity R+G+B, approximate chrominance features such as R-B, and so-called *excess* color features such as 2G-R-B.

All features are normalized into the range 0 to 255 and further discretized into histograms of length 2^b, where b is the number of bits of resolution. We typically discretize to 5 or 6 bits, yielding feature histograms with 32 or 64 buckets. This discretization is performed for efficiency and for defeating the "curse of dimensionality" that occurs when trying to estimate feature distributions from small numbers of samples [2].

3.2 Creating Tuned Features

If both object and background were unicolored, then a plausible argument could be made that variation in apparent color of pixels would lead to Gaussian distributions in color space. In this case, Linear Discriminant Analysys (LDA) could be used to find the subspace projection yielding the least overlap (i.e., maximum separability) between object and background. However, we must be able to handle targets and backgrounds that have multimodal distributions of colors. These violate LDA's Gaussian assumption, invalidating its analytic solution.

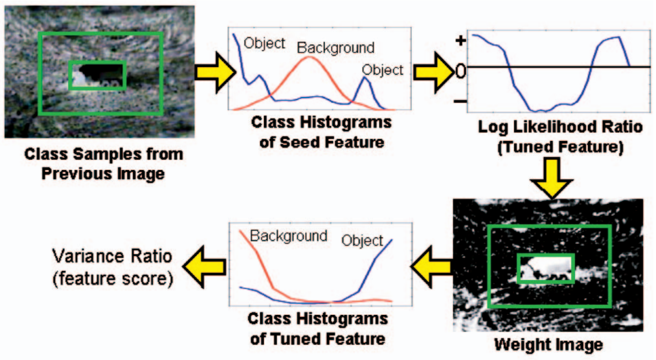
Class Samples from Previous Image
Object
Background
Object
Class Histograms of Seed Feature
+
0
–
Log Likelihood Ratio (Tuned Feature)
Weight Image
Background
Object
Class Histograms of Tuned Feature
Variance Ratio (feature score)

Fig. 2. Empirical tuning and evaluation of a candidate seed feature, demonstrated on an IR image of a truck. Sample histograms of (possibly multimodal) feature values for object and background are used to compute a log likelihood ratio function that maps object pixels to (unimodally) positive values and background pixels to negative values. When backprojected into image space, these function values form a 2D weight image that can be used to track the object. The variance ratio is used to evaluate how well the tuned feature induces separability between the two classes since separability correlates well with suitability of the weight image for tracking.

Our approach, illustrated in Fig. 2, transforms each seed feature based on the class-conditional distributions of its values. The transformation is computed as a log likelihood ratio of the feature value distributions for object versus background. This nonlinear transformation achieves two important goals. First, it creates a new feature that is "tuned" to discriminate between object and background pixels. Thresholding the value of this feature at zero is equivalent to using a maximum likelihood rule to classify object pixels from background. Second, for features with good discriminative power, this method collapses potentially multimodal object and background distributions into unimodal distributions. Simple methods for measuring separability of two Gaussian distributions are then applicable, including the variance ratio.

We use a "center-surround" approach to sampling pixels from object and background. A rectangular set of pixels covering the object is chosen to represent the object pixels, while a larger surrounding ring of pixels is chosen to represent the background. For an inner rectangle of dimensions $h \times w$ pixels, an outer margin of width $.75 * \max(h, w)$ pixels forms the background sample. This is a conservative strategy that leads to discriminative features that separate object from background regardless of which direction the object maneuvers in the image. Background appearance also could be sampled by biasing selection of pixels towards the area of the image where the object is predicted to be in the future, given its recent trajectory.

Given a feature f, let $H_{obj}(i)$ be a histogram of feature values for pixels on the object and $H_{bg}(i)$ be a histogram for pixels in the background sample, where i ranges from 1 to 2^b, the number of histogram buckets. We form an empirical discrete probability distribution $p(i)$ for the object, and $q(i)$ for the background, by normalizing each histogram by the number of elements in it.

For each seed feature, we create a new "tuned" feature tailored to better discriminate between object and background. This tuned feature is formed as the log likelihood ratio of the class conditional seed feature distributions. The log likelihood ratio of a feature value i is given by

$$L(i) = log\frac{\max\{p(i), \delta\}}{\max\{q(i), \delta\}}, \tag{2}$$

where δ is a small value (we set it to 0.001) that prevents dividing by zero or taking the log of zero. The nonlinear log likelihood ratio maps object/background distributions into positive values for colors distinctive to the object and negative for colors associated with the background. Colors that are shared by both object and background tend towards zero. Backprojecting these log likelihood ratio values into the image produces a weight image suitable for tracking (Fig. 2).

3.3 Evaluating Feature Discriminability

To summarize the development so far, for each seed feature, we estimate the distributions $p(i)$ and $q(i)$ of object and background pixels, respectively, and then create a tuned feature $L(i)$ as the log likelihood ratio of these two distributions. Now, we want to measure the separability that tuned feature $L(i)$ induces between object and background classes using the two-class *variance ratio*. We could proceed by reaccumulating new class conditional distributions for the tuned feature, as shown in Fig. 2, but for efficiency we can reuse the distributions $p(i)$ and $q(i)$ already computed for the seed feature. Using the equality $\mathrm{var}(x) = Ex^2 - (Ex)^2$, we compute the variance of $L(i)$ with respect to object class distribution $p(i)$ as

$$\mathrm{var}(L;p) = E[L^2(i)] - (E[L(i)])^2 \tag{3}$$

$$= \sum_i p(i)L^2(i) - \left[\sum_i p(i)L(i)\right]^2 \tag{4}$$

and similarly for background class distribution $q(i)$. The variance ratio of the log likelihood function can now be defined as

$$\mathrm{VR}(L;p,q) \equiv \frac{\mathrm{var}(L;(p+q)/2)}{[\mathrm{var}(L;p) + \mathrm{var}(L,q)]}, \tag{5}$$

which is the total variance of L over both object and background class distributions, divided by the sum of the within class variances of L for object and background treated separately. As in (2), the implementation of this equation avoids division by zero by taking the maximum of the denominator and a small epsilon value.

The intuition behind the variance ratio is that we would like log likelihood values of pixels on both the object and background to be tightly clustered (low within class variance), while the two clusters should ideally be spread apart as much as possible (high total variance). The denominator enforces that the within class variances should be small for both object and background classes, while the numerator rewards cases where values associated with object and background are widely separated. Note the similarity to the Fisher discriminant used in the computation of LDA, where the squared difference between the mean values of the two classes is used as an alternative measure of total variance. To reemphasize an earlier point, while LDA and the variance ratio are not appropriate for measuring separability of the multimodal class distributions induced by seed features, after mapping through the log likelihood ratio to produce tuned features, class distributions should be more unimodal and, thus, use of variance ratio for measuring discriminative power of tuned features is appropriate.

3.4 Ranked Weight Images

If a feature's two-class log likelihood function from the previous step is used to label pixels in a new video frame, the result is a weight image where, ideally, object pixels contain positive values and background pixels contain negative values. For a perfect discriminating feature, this weight image would be an indicator function, with value 1 at pixels corresponding to the object and -1 everywhere else. In this ideal case, tracking could be achieved simply by thresholding at zero and computing the object center and rough shape using the method of moments (see also [4]). In practice, object and background color distributions will overlap and perfect separation is not achievable. Instead, we settle for ranking the features by separability and choosing the top N.

Fig. 3 shows a sample object and the set of weight images produced by all 49 candidate features, after rank-ordering the features based on the two-class variance ratio measure. The weight image for the most discriminative feature is at the upper left and the image for least discriminative feature is at the lower right. We observe a very high correlation between variance-ratio ranking and suitability of the weight image for localizing the object in the next frame.

Fig. 4 shows other sample images with labeled object and background pixels, along with the weight images associated with the tuned features having highest, median, and lowest variance ratio values, corresponding to the best, median, and worst features, respectively, in terms of object/background separability. Again, we see good agreement between these rankings and our intuitive preference regarding which weight images to use for tracking.

3.5 Tracking

The above feature ranking mechanism is embedded in a tracking system as shown in Fig. 5. Object and background pixels are sampled from the previous frame, given the previous location of the tracked object. Potential tracking features are ranked using the variance ratio to determine how well each feature distinguishes object from background. The top N most discriminative individual features are used to compute weight images for the current frame. Due to the continuous nature of video, the distribution of object and background features in the current frame should remain similar to the previous frame and the most discriminative features should still be valid.

A local mean-shift process is initialized in each of the N new weight images. These processes perform gradient ascent to find the nearest local mode in their respective weight images. These mean-shift processes converge to N estimates of the 2D location of the object in the current frame, which are combined to yield a final estimate of object location. In our implementation, we use a *naive median* estimator with $\hat{x} = \mathrm{median}(x_1, \ldots, x_n)$ and $\hat{y} = \mathrm{median}(y_1, \ldots, y_n)$. The median is chosen rather than the mean in an attempt to add robustness

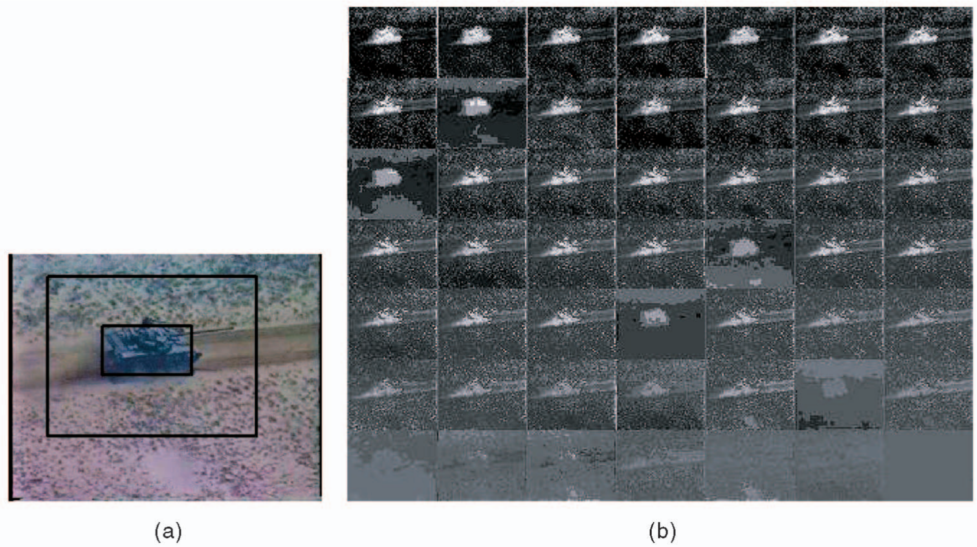

Fig. 3. (a) A sample image with concentric boxes delineating object and background samples. (b) Weight images produced by all 49 tuned candidate features, rank-ordered by the two-class variance ratio measure. The weight image for the most discriminative feature (hypothesized as best for tracking) is at the upper left. The image for least discriminative feature (worst for tracking) is at the lower right.

against any single mean-shift process yielding a bad estimate of object location that corrupts the pooled estimate.

The algorithm iterates through each subsequent frame of the video, extracting new samples of object and background pixels and choosing new sets of discriminative features. In this way, both the features used for tracking and the appearance models of object and background classes evolve together over time. Adaptively updating appearance models in this manner raises the specter of *model drift*, a classic problem in adaptive tracking. Model drift builds up gradually over time as misclassified background pixels start to "pollute" the foreground model, leading to further misclassification and eventual tracking failure. To avoid this problem, we compute the empirical object feature distribution at each frame by pooling pixel samples from the previously tracked image together with the labeled object pixels from the original training sample in the first frame, which is assumed to be uncontaminated. The estimated feature distribution is therefore a straightforward average of the initial and current feature distributions. Forming a pooled estimate allows the object appearance model to adapt to current conditions while keeping the overall distribution anchored to the original training appearance of the object. This heuristic approach assumes that the initial color histogram remains representative of object appearance throughout the entire tracking sequence.

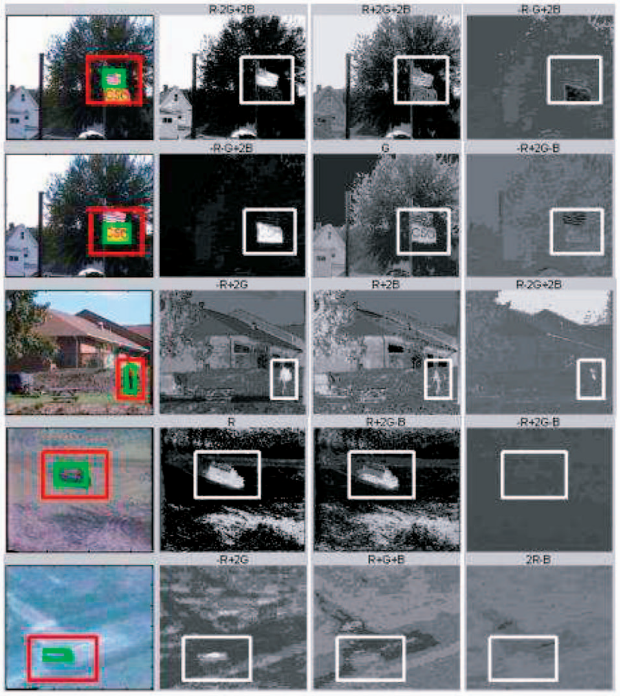

Fig. 4. Sample video frames with ranked weight images. Left column: Frame with labeled object (green box) and background pixels (red box). Second through fourth columns: weight images corresponding to the tuned features with highest, median, and lowest variance ratio scores, respectively. Rank ordering features by the two-class variance ratio correlates well with intuition regarding which features would be best to use for tracking the object.

4 Experiments

This section presents three challenging tracking examples that illustrate the benefits of combining online feature selection with object tracking. Specifically, these benefits are enhanced ability to track low contrast objects, ability to adapt to changing background and illumination conditions, and ability to avoid distraction by automatically emphasizing appearance characteristics that are distinctive to the object.

The first tracking example uses low-contrast aerial footage of a car driving through patches of sunlight and shadow (Fig. 6). Watching the video frame-by-frame, it is challenging even for a human observer to delineate the position of the car when it passes through shadow regions. Despite the difficulties, the tracker presented here smoothly tracks the car through the changing illumination conditions and through partial occlusion caused by trees lining the

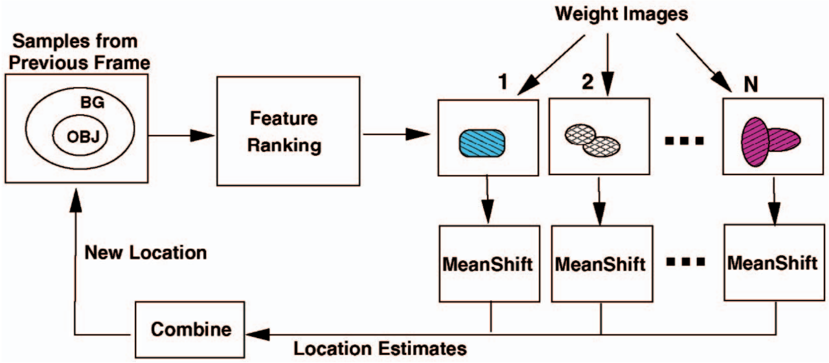

Fig. 5. Overview of tracking system with online, adaptive feature selection. Samples of object and background pixels in the previous frame guide evaluation of candidate features, leading to a rank ordering of features based on discriminative ability. The top N best features are applied to the current frame to compute N weight images. A mean-shift process is applied to each weight image to compute a 2D location estimate. These N estimates are combined to determine the best location of the object in the current frame and the procedure iterates.

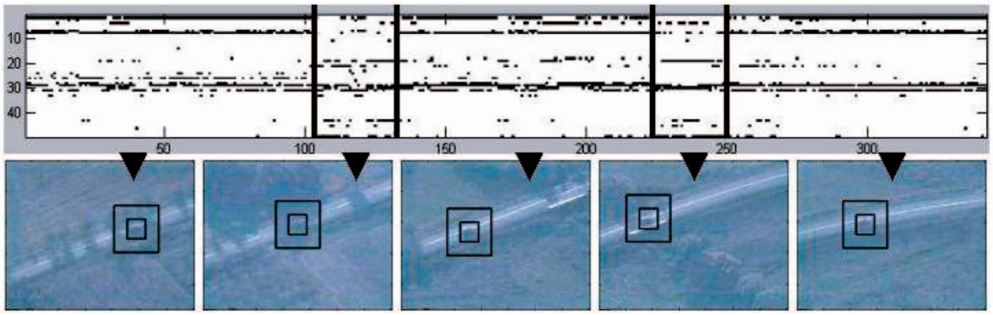

Fig. 6. Trace of features selected to track a car through a hazy aerial sequence. The car is tracked successfully through shadows and partial occlusion by trees lining the road. See text for details.

road. Fig. 6 presents a trace showing which five features out of 49 were chosen as most discriminative for each frame of the tracked sequence. We see that many of the same features are selected through most of the video (horizontal bars in the picture represent the same features being chosen again and again) and many features were never selected (empty rows). At a coarse level of description, the feature history can be broken into five blocks of frames, where roughly the same set of features were chosen consistently within each block and the discontinuity between blocks is marked by a switch to a different set of features. Fig. 6 also shows representative frames from within each of these five coarsely segmented time blocks. For the first, middle, and last block, the car is predominantly driving through sunny road or dappled patches of shadow. The second block delineates a subsequence where the car plunges into an area of deep, extended shadow. The fourth block denotes a subsequence where the car travels over a small bridge that has color properties similar to the car.

Further analysis shows that the five features chosen most often when the car is in sunlight are R-G, 2G-R, 2G-B-R, 2G+B-2R, and 2R-G-B. The five features chosen most often in shadow are 2G-B, 2G-R, G, 2G+B-R, and 2G+B. The most chosen features under each condition, R-G in sunlight and 2G-B in shadow, were compared in Fig. 1 on sample sunlight and shadow images. It is not easy to intuitively explain why these particular features were chosen most often. In fact, that is the point of this paper: Features should be chosen based on an objective function that measures their discriminability, rather than on subjective human intuition.

Fig. 7a illustrates failure of a standard mean-shift tracker [7] on one section of the video. When the car passes over a small bridge, the color of the top of the bridge rail is nearly identical to the color of the specular highlight on top of the car. The mean-shift tracker gets sidetracked by this similar color, leading to tracking failure. Fig. 7b shows results from our adaptive tracker. Since the tracker maintains a model of both object AND background color distributions, it detects that a color in the background is similar to a color in the model and automatically down-weights those pixels. The tracker is therefore not attracted to the bridge railing and tracking proceeds.

This ability to adaptively emphasize different object characteristics to avoid distraction is illustrated more clearly by a video tracking example shown in Fig. 8. Here, a red car with a white roof is tracked on a busy highway. Fig. 8 shows three representative frames from the sequence, along with the weight image produced by the top-ranked (most discriminative) feature. Against the dark tarmac, the white roof of the vehicle provides an excellent, high contrast target for tracking. However, when the car passes near other white vehicles, there is danger of distraction. In this case, the feature selection process automatically shifts to a feature that emphasizes the red color, thus causing the distracting white vehicles to disappear from the weight image (Fig. 8b). After the distractors have been passed, the feature selection process again shifts back to emphasize the

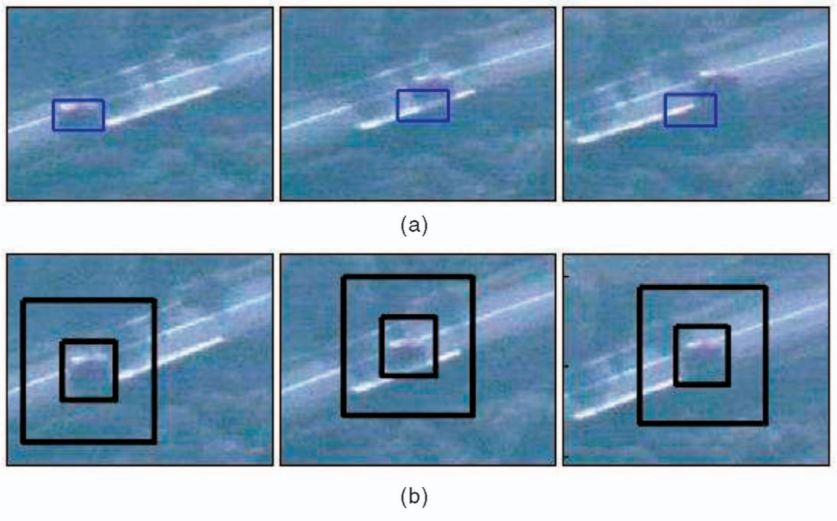

Fig. 7. (a) The traditional mean-shift tracker is attracted to background pixels that have the same color as part of the tracked car, leading to tracking failure. (b) By modeling both object AND background color distributions, our tracking approach automatically down-weights shared colors, thus avoiding distraction.

high contrast white roof. We stress that this fortunate shifting of color emphasis to avoid distraction happens automatically, as a byproduct of selecting features that maximize separability between object and background. This example also illustrates the importance of sampling object pixels from both the previous frame AND the original set of labeled pixels (both red and white) in the first frame. If only pixels from the previous tracked location were sampled to determine object appearance in this example, the method would drift to a model containing either only red or only white pixels, losing the ability to adapt to future distractors.

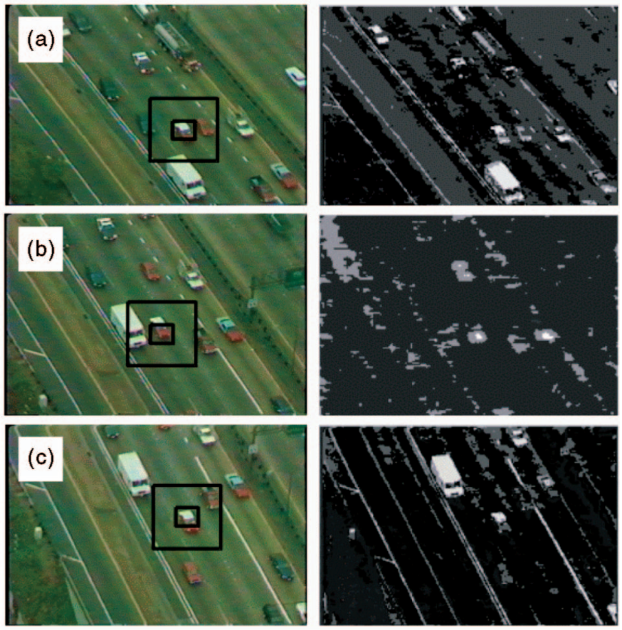

Fig. 8. Example of feature adaptation to avoid distractors. Left column: video frame with object/background windows overlaid. Right column: weight image from top-ranked tracking feature. (a) A red and white car is being tracked. (b) When the car passes a large white truck, the top-ranked tracking feature adapts to emphasize the red color, causing the white truck to "disappear" (compare weight image (b) with the other two weight images). (c) When the car is alone against the dark road, the top-ranked tracking feature empasizes the high-contrast white roof.

A third video example is depicted in Fig. 9. The object being tracked is a flag, blowing nonrigidly in the wind. The camera viewpoint continually changes, causing the scene background to vary. The flag is sometimes seen as a bright object against dark trees and sometimes seen as a darker object backlit by the bright sky. Nonetheless, the tracker successfully follows the flag through the entire minute-long sequence. Fig. 9 presents a trace showing which five features out of 49 were chosen as most discriminative for each frame of the tracked sequence. Again, we see that many of the same features are selected through most of the video. However, we also note that these are different features than the ones chosen in the earlier car tracking example. There is a lot of variation in background clutter and illumination conditions throughout this sequence, and coarsely segmenting the feature selection trace into time blocks, as was done in the earlier example, is difficult. Instead, we show a few sample frames from the tracked sequence, with an indication of where they occur.

5 Distractor-Resistant Feature Selection

5.1 Drawback of the Variance Ratio

The intuition behind using the variance ratio as a feature selection method is that we would like feature values of pixels on both the object and background to be tightly clustered (low within class variance), while the two clusters should ideally be spread apart as much as possible (high total variance). Variance ratio is computationally efficient and does a good job at selecting features that maximize the overall contrast between the foreground object and the surrounding background. However, it is best suited to backgrounds that are relatively homogeneous and is not necessarily the best method to use when there are nearby distractors in the neighborhood of the tracked object. This is so because it maximizes the average contrast between the log likelihood of foreground pixels and the entire set of

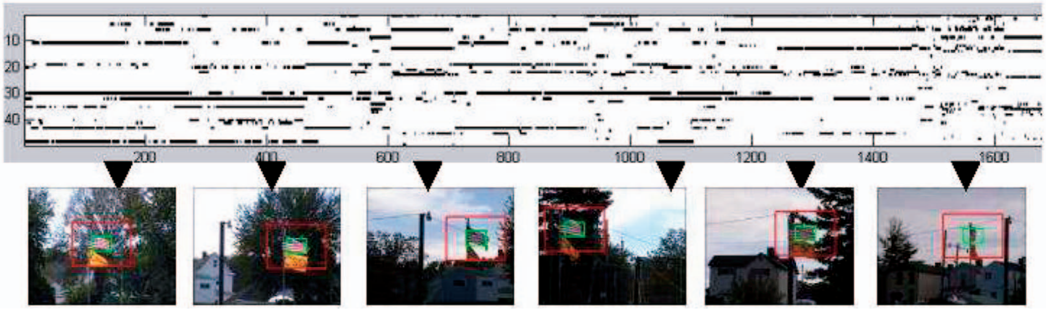

Fig. 9. Trace of selected features over a one-minute long tracking sequence. The object tracked is a flag waving nonrigidly in the breeze. The camera motion leads to a wide range of changing background and illumination conditions, all of which are handled successfully by the tracker.

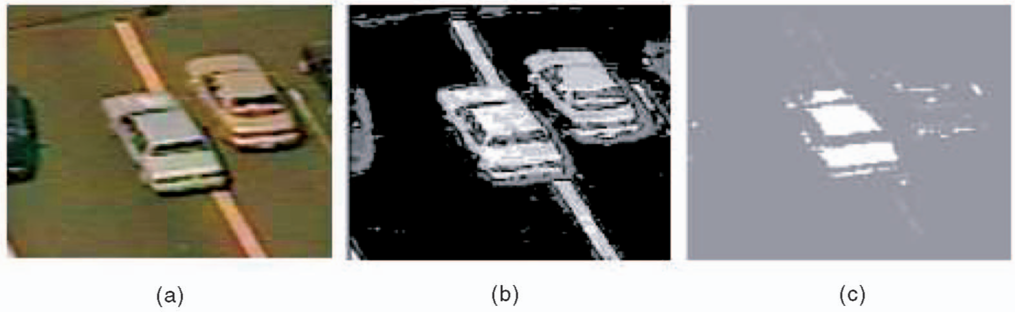

Fig. 10. (a) Car passing another car of a similar color. (b) Weight image chosen by the variance ratio. (c) Weight image chosen by the method developed in this section. The variance ratio favors features that produce weight images where the object has high contrast with respect to the average background, even though there may be an equally high contrast distractor nearby. We prefer the weight image at the far right for tracking—despite the poor contrast between object and background, there are no nearby distractors to tempt the tracker.

background pixels without taking into account spatial clustering of high likelihood values in the background associated with a potential distractor.

This problem is illustrated in Fig. 10. The feature chosen by the variance ratio (Fig. 10b) yields a weight image with high contrast between object and the "average" background. However, there is also an equally likely (high contrast) distractor nearby that could easily attract the mean-shift window and cause tracking failure. In this case, the weight image in Fig. 10c is a better image to use for tracking—even though the object has lower contrast with respect to the background using this feature, there are no nearby distractors that could cause the tracker to fail. Unfortunately, the average contrast between foreground and background is less in this image, and it receives a lower variance ratio ranking than the weight image in Fig. 10b.

Although an example in the last section showed the algorithm successfully downweighting potential distractors, that example worked due to the two-color nature of the object being tracked, allowing the algorithm to discretely switch from one color to the other. This section considers a more general solution to the problem of avoiding spatially correlated background distractions.

5.2 Quantifying Distraction

The key to distractor-resistant feature selection is spatial reasoning about peaks in the weight image. To form an accurate picture of distractors in the neighborhood of a tracked object, we examine nearby image regions of similar size to the object. Each such region is a potential distractor, with strength characterized by the sum of weights within its image area. To minimize the likelihood of distraction, we seek features that minimize the maximum sum of weights within any potential distractor region. This strategy is related to the concept of maximizing the "margin" in pattern classification—in our case, we are trying to maximize tracking success by minimizing the probability of misclassifying the distractor as the object. A more formal procedure description follows.

Recall that, given a candidate feature f with values i ranging from 0 to 2^b, we use samples of pixels from the object and background classes to form empirical discrete probability distribution $H_{obj}(i)/n_{obj}$ for object and $H_{bg}(i)/n_{bg}$ for background, with n_{obj} and n_{bg} being the number of object and background sample pixels, respectively. To facilitate reasoning about the spatial layout of feature values, we define two likelihood images indexed by pixel location x

$$P(x|obj) = p(x) = H_{obj}(f(x))/n_{obj}, \tag{6}$$

$$P(x|bg) = q(x) = H_{bg}(f(x))/n_{bg}, \tag{7}$$

where $f(x)$ denotes the value of feature f at pixel x. The weight image $L(x)$ is then formed from the log likelihood ratio values as

$$L(x) = log\frac{p(x)}{q(x)}, \tag{8}$$

where, for simplicity, we have left out the modifications in (2) that prevent dividing by zero or taking the log of zero.

Consider a target of known size, whose appearance model scores most highly for two regions (sets) of pixels X_0 and X_1

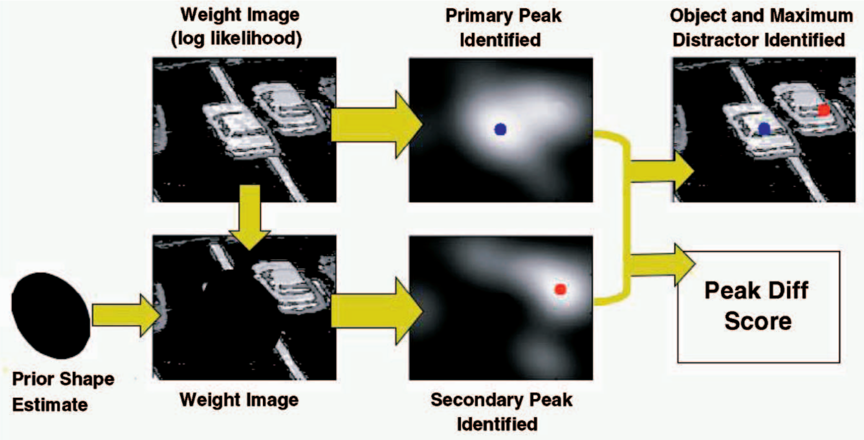

Fig. 11. Deriving the peak-difference score for a given weight image. Top row: The weight image is smoothed and the primary peak location and value is identified. Bottom row: A prior shape estimated is used to mask out the object pixels in the weight image. This masked weight image is then smoothed and the secondary peak location and value is identified. This secondary peak represents the estimated maximum (worst) distractor. The difference between primary and secondary peak values yields the peak difference score.

in the current frame. Let c_0 be the class label of the pixels in X_0 and c_1 be the class label of the pixels in X_1. Since the target is at only one of the two locations, we consider two events, $A \equiv \{c_0 = \text{obj}, c_1 = \text{bg}\}$ and $B \equiv \{c_0 = \text{bg}, c_1 = \text{obj}\}$. Because we have previously tracked the object into the current frame, we know that region X_0 actually contains the object and that region X_1 is therefore a distractor. We now want to find a feature that maximizes the probability that we will deduce the correct state of affairs, A, rather than the alternative, B, given the observed regions X_0 and X_1. Assuming independence of the observed regions given their class labels, we thus want to find a feature that maximizes

$$\frac{P(A|X_0,X_1)}{P(B|X_0,X_1)} = \frac{P(c_0 = \text{obj}, c_1 = \text{bg}|X_0,X_1)}{P(c_0 = \text{bg}, c_1 = \text{obj}|X_0,X_1)} \tag{9}$$

$$\downarrow \text{apply Bayes rule, with priors denoted by } \pi \tag{10}$$

$$= \frac{P(X_0,X_1|c_0 = \text{obj}, c_1 = \text{bg})\,\pi(c_0 = \text{obj}, c_1 = \text{bg})}{P(X_0,X_1|c_0 = \text{bg}, c_1 = \text{obj})\,\pi(c_0 = \text{bg}, c_1 = \text{obj})} \tag{11}$$

$$\downarrow \text{class conditional independence; replace constant prior by C} \tag{12}$$

$$= C\ \frac{P(X_0|c_0 = \text{obj})P(X_1|c_1 = \text{bg})}{P(X_0|c_0 = \text{bg})P(X_1|c_1 = \text{obj})} \tag{13}$$

$$\downarrow \text{independence over pixels in region} \tag{14}$$

$$= C\ \prod_{X_0}\frac{P(x|\text{obj})}{P(x|\text{bg})}\prod_{X_1}\frac{P(x|\text{bg})}{P(x|\text{obj})} \tag{15}$$

$$\downarrow \text{substitute empirical distributions from (7)} \tag{16}$$

$$= C\ \prod_{X_0}\frac{p(x)}{q(x)}\prod_{X_1}\frac{q(x)}{p(x)}. \tag{17}$$

Dropping the constant prior term C, we maximize the log of (17):

$$\log\left(\prod_{X_0}\frac{p(x)}{q(x)}\prod_{X_1}\frac{q(x)}{p(x)}\right) = \log\left(\prod_{X_0}\frac{p(x)}{q(x)}\Big/\prod_{X_1}\frac{p(x)}{q(x)}\right) \tag{18}$$

$$= \sum_{X_0} log\frac{p(x)}{q(x)} - \sum_{X_1} log\frac{p(x)}{q(x)} \tag{19}$$

$$= \sum_{X_0} L(x) - \sum_{X_1} L(x). \tag{20}$$

To summarize, given a specific object region X_0 and potential distractor region X_1 discovered in the previous frame, we can minimize the likelihood of misclassifying distractor X_1 as object by choosing the feature that minimizes the difference between the sum of weight image pixels over regions X_0 and X_1. Features that minimize this difference should also be good features for minimizing that distraction in the weight image computed for the current frame.

5.3 Minimizing the Maximum Distraction

The derived formula for measuring the feature-specific severity of a distractor relies on knowing the region X_1 that contains the distractor. Although in theory we want to minimize over all distractions, in practice we minimize with respect to just the single, worst distractor. This still requires a search for the maximal distractor region X_1, over all potential distractor regions X_*. Finding the maximum distractor region is performed efficiently as follows (see Fig. 11):

Step 1. Smooth the candidate feature weight image with an isotropic, separable Gaussian kernel related to the current size of the object region X_0. The value at each pixel in the convolved image is a weighted sum of pixels in a circular region surrounding it, normalized by the total weight pixels in that region. Convolution with a Gaussian is thus a fast, approximate method for computing the region sum of (20) over circular regions centered at every pixel. Note also the theoretical connection between convolution

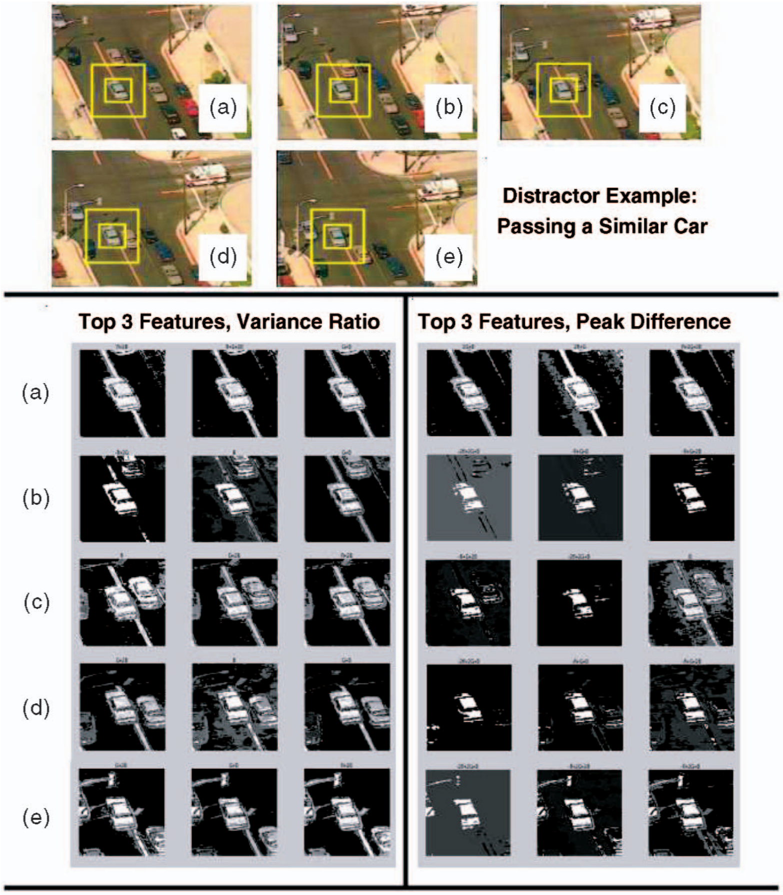

Fig. 12. Comparison of variance ratio and peak difference feature selection for an example sequence where a tracked car passes another with a similar color. Top: Five frames from the sequence with object/background windows overlaid. Bottom: Top three features chosen within the region of interest, using variance ratio score (left) and peak difference score (right). Peak difference chooses features that minimize the distractor object.

with a Gaussian and using the mean-shift algorithm with a Gaussian kernel [5]. As a result, the smoothed weight image represents the actual surface that the mean-shift algorithm performs hill-climbing on and spatial reasoning about peaks in this image is relevant to determining whether mean-shift will converge to the correct mode.

Step 2. Extract the central object peak from the smoothed image. We want to maximize the difference between the height of this peak and the largest distractor.

Step 3. Find the next highest peak after removing the central object peak. This peak represents the most likely distractor object. The mean-shift tracker *may* be attracted to this incorrect position. Whether it *will* be attracted to this position and, therefore, potentially lose the tracked object, depends on where in the weight image the mean-shift tracker is initialized and whether the gradient at that position points toward the true (central) peak or this incorrect distractor peak. To find the second highest peak, we mask out the object pixels in the current weight image using a coarse estimate of object shape. The current shape estimate can be as simple as the rectanglular region used to sample object pixels. More sophisticated elliptical shape estimates can be computed via the EM algorithm from the previous weight image (see also [33]).

Step 4. Evaluate feature quality as the difference between these two peak heights. By choosing the feature that makes the true object peak most prominent as compared against the most likely distractor, we seek to minimize the maximum distractor and, thus, minimize the possibility of distraction in the next frame.

The example illustrated in Fig. 12 demonstrates that this new approach to distractor-resistent feature selection can outperform the original method based on variance ratio when distractors are present. To make it easier to find cases of distractors, we use just 3 bits of resolution in all color histograms (eight buckets), greatly reducing the ability of the color features to separate object from background. The example shows a tracked car passing another of similar color. Before the passing begins (Fig. 12a), both the variance ratio and the peak difference method select similar features. However, while the tracked car is close to the other vehicle (Figs. 12b, 12c, 12d, and 12e), the weight images selected by the variance ratio are poor candidates to use for tracking since the passed car remains as a highly-visible distractor, and there is danger that the mean-shift tracker may incorrectly jump to follow it instead. In contrast, the top-ranked features produced by the peak difference method produce weight images where the target car can be safely tracked since the other vehicle presents only a minimal distraction in these images.

6 Discussion

6.1 Summary

Although object tracking based on color histogram appearance models can achieve efficient tracking through partial occlusion and pose variation, tracking success or failure depends primarily on how distinguishable the object is from its surroundings. Surprisingly, most tracking applications use a fixed set of features, determined a priori (a notable exception is [27]). These approaches ignore the fact that it is the ability to distinguish between object and background that is most important and that appearance of both object and background will change as the target object moves.

This paper presents an effective method for continuously evaluating multiple features while tracking and for selecting a set of features that improve tracking performance. We develop an online feature ranking mechanism based on applying the two-class variance ratio to log likelihood distributions computed for a given feature from samples of object and background pixels. This feature ranking mechanism is embedded in a tracking system that adaptively selects top-ranked features for tracking. The result is a system in which the features used for tracking and the appearance models of object and background coevolve over time. The experimental results demonstrate successful tracking performance even on challenging video sequences.

Although the variance ratio is a computationally efficient mechanism for selecting tracking features, it does not take into account the spatial distribution of background values in the weight image and, thus, does not appropriately penalize features that produce spatially-correlated background clutter or strong distractors. We have presented an additional feature selection method that performs spatial reasoning over potential distractor regions, seeking features that maximize the difference between the sum of weights within the object versus the maximum sum of weights over any distractor region. This method chooses features that minimize the potential for distraction in the next frame.

6.2 Issues and Future Work

6.2.1 How Many Features to Select

This paper presents methods for ranking and selecting the N best features for tracking an object. However, we have left open how to choose the value N, i.e., how many features to use in the tracking system described in Section 3.5. In our experiments, we typically choose N to be either 1, 3, or 5. There is little difference in tracking results when using either three or five features. There is a cost to choosing a higher number of features because more computation time must be spent during tracking (N runs of mean-shift must be performed per frame). When using the variance ratio for selection, choosing only a single feature is dangerous because, as we have discussed, the variance ratio sometimes gives high rank to a feature that does not discriminate well between the object and background clutter. However, the peak difference selection method is powerful enough that the single best feature found by that algorithm is often sufficient for successful tracking.

More principled methods for choosing the number of tracking features N could be devised by referring back to our original insight that selecting features for tracking is related to the problem of selecting discriminative features for classifying foreground from background pixels. The pattern recognition literature describes both exhaustive and heuristic methods for searching over both size and composition of the best subset of features [2]. Searching the space of feature subsets is far too expensive to run during online tracking. However, given a training set containing samples of the types of objects one wants to track and the types of environment one will be tracking them through, we can imagine an offline process for determining how many features should be used on average to maintain a specified level of performance, and even a coarse prior ranking of which features might be best for which object in which environment.

A more principled method for choosing sets of features would also take into account the degree of independence between features. Weight images produced by two high-ranking features are often highly correlated and, therefore, not much new information is introduced by adding the second feature. Discovering such correlations between features is not addressed in our current work. Finally, one could explore more sophisticated ways to combine the information from multiple features [20]. Here, we have treated each feature as an independent information source used to run an entire mean-shift process with pooling of information happening at the end by combining the end-result location estimates. One can imagine combining the information from multiple features at the weight-image level to produce a single, more refined weight image where the foreground object is more clearly distinguished from the background than in any individual feature weight image. See [12] for an exploration of this approach.

6.2.2 What Type of Features to Use

This paper uses 49 linear combinations of RGB color space as a simple yet concrete example of a set of candidate seed features. Features derived from other color spaces, such as HSV or YUV, could be used instead of or in addition to this set of features. For example, the work of Stern and Efros adaptively selects between five color spaces RG, rg, HS, YQ, and CbCr for face tracking [27]. The approach presented here can be easily extended to include histograms formed from other types of features. These include texture features computed by, e.g., Gabor filters [17], edge orientation histograms [11], motion features computed via optical flow or background subtraction [9], and joint spatial-feature models such as color correlograms [13]. Each of these spaces has tunable parameters such as scale, orientation, or discretization resolution of the histogram. Therefore, the space of potential features that can be used is enormous if one considers also selecting among differently parameterized and quantized versions of the same base features. Of course, adding more features means that computation time for feature selection also rises, particularly if all features are evaluated each time a selection is made. This may be prohibitively expensive for real-time tracking. As discussed above, one possible remedy is to use training data of expected object and background appearances in an offline search for a smaller set of online candidate features that typically do well in those conditions. Online feature selection during tracking then needs to consider only the smaller set of feature candidates that have shown prior promise of utility.

6.2.3 Combining Feature Selection Mechanisms

Since we have described two feature selection evaluation functions in this paper, variance ratio and peak difference, it is natural to consider whether and how they could be combined. For example, perhaps we could use the variance ratio until a distractor is noticed and then switch to using peak-difference until the distractor has safely disappeared. There would be some benefit to doing this from a computational standpoint (variance ratio is less expensive to compute than peak difference). However, in our experience, the peak difference method works well when variance ratio does not and, additionally, works better in clutter situations, so from a tracking performance standpoint, one would do well to just use peak difference and dispense with the variance ratio method altogether, rather than try to combine them. It is our opinion that effort is better spent creating improved feature selection evaluation functions, and better features to apply them to, rather than designing methods to combine multiple feature selection techniques.

6.2.4 Feature Selection as Needed

In this paper, it has been assumed that all features are evaluated and the best N selected at every frame. This ideally provides maximal responsiveness to rapidly changing background and illumination conditions. There is a heavy computational cost to considering all features at every frame, as well as the potential cost of inconsistent localization caused by switching between features that emphasize different portions of the tracked object. However, the best features to use are a function of both object appearance and background appearance, and if both these appearances are slowly varying, then the features used for tracking do not need to be updated frequently.

One strategy is to invoke feature selection only periodically during tracking. For example, we have implemented C versions of the variance ratio and peak difference algorithms in this paper that select from the 49 candidate seed features only at every 10th frame. Running time is 17 frames per second for the variance ratio method and 15 frames per second for peak difference. These runtimes were measured on an Intel Pentium4, 2.5 GHz machine with 1GB RAM. The run times include image file reading as well as graphical display of the 720×480 color images overlaid with the current object bounding box. In future work, we will explore methods that efficiently monitor tracking quality using the current set of features and invoke the full feature selection process only when that quality degrades too far. For example, the strength of the maximum distractor using a current set of features could be monitored to determine whether it is time to initiate the full feature selection computation.

6.2.5 Tracking Initialization

Although tracking initialization is beyond the scope of this paper, it is essential to address it in a real system. The experiments in this paper were initialized by hand by drawing a bounding box around the object to track. In a practical system, object tracking could be initialized by automatically detecting moving objects. For a stationary camera, this is easily achieved via background subtraction [6]. When the camera is in motion, moving objects can still be detected based on motion stabilization and frame differencing [32] or motion segmentation [25].

A more subtle issue is how to ensure that the initial color distribution selected in the first frame is representative of the appearance of the object in subsequent frames. Recall that, to avoid model drift, we "anchor" the current color distribution by pooling it with a reference distribution from the first frame. Anchoring to a reference frame is also used in [22] to avoid drift when updating intensity templates. However, if the first frame were to contain an unusual specular reflection, for example, it would corrupt all subsequent distributions. When we outline objects by hand, this can largely be avoided, however, a system that automatically initializes bounding boxes for tracking might not be so fortunate. For prototypical objects like heads, color models have been bootstrapped using prior knowledge of object shape and movement [29]. In our case, we could track an object initially using motion detection methods while accumulating an initial histogram appearance model over several frames (see also [12]). This would avoid forming a corrupt reference color distribution from a singularly poor initial frame. Ultimately, the approach of maintaining a reference distribution needs to be discarded, as it limits the amount of variation that can be tolerated as the object appearance evolves. More work is needed to solve the twin problems of robust model initialization and drift-free model update.

Acknowledgments

This work was supported in part by US Defense Advanced Research Projects Agency contract N00014-00-1-0915, NBCHC020090, and NBCH1030013, by US National Institutes of Health contract AG05133, DA015900-01, and N01-CO-07119, and by PA DOH grant ME01-738.

References

[1] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association.* Academic Press, 1988.

[2] C. Bishop, *Neural Networks for Pattern Recognition.* Clarendon Press, 1997.

[3] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems.* Artech House, 1999.

[4] G. Bradski, "Computer Vision Face Tracking for Use in a Perceptual User Interface," *Proc. IEEE Workshop Applications of Computer Vision,* pp. 214-219, 1998.

[5] Y. Cheng, "Mean-Shift, Mode Seeking, and Clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 8, pp. 790-799, Aug. 1995.

[6] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for Cooperative Multi-Sensor Surveillance," *Proc. IEEE,* vol. 89, no. 10, pp. 1456-1477, Oct. 2001.

[7] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 5, pp. 564-577, May 2003.

[8] T. Cootes, G. Edwards, and C. Taylor, "Active Appearance Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 6, pp. 681-685, June 2001.

[9] J. Davis, "Recognizing Movement Using Motion Histograms," Technical Report No. 487, MIT, Mar. 1999.

[10] A. Elgammal, R. Duraiswami, and L. Davis, "Probabilistic Tracking in Joint Feature-Spatial Spaces," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 781-788, June 2003.

[11] W. Freeman and M. Roth, "Orientation Histograms for Hand Gesture Recognition," *Proc. IEEE Workshop Automatic Face and Gesture Recognition,* June 1995.

[12] E. Hayman and J.O. Eklundh, "Probabilistic and Voting Approaches to Cue Integration for Figure-Ground Segmentation," *Proc. European Conf. Computer Vision,* pp. 469-486, 2002.

[13] S. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih, "Spatial Color Indexing and Applications," *Int'l J. Computer Vision,* vol. 35, no. 3, pp. 245-268, 1999.

[14] M. Irani and P. Anandan, "A Unified Approach to Moving Object Detection in 2D and 3D Scenes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 6, pp. 577-589, June 1998.

[15] M. Isard and A. Blake, "CONDENSATION—Conditional Density Propagation for Visual Tracking," *Int'l J. Computer Vision,* vol. 29, no. 1, pp. 5-28, 1998.

[16] D. Koller, K. Daniilidis, and H. Nagel, "Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes," *Int'l J. Computer Vision,* vol. 10, no. 3, pp. 257-281, 1993.

[17] B. Manjunath and W. Ma, "Texture Features for Browsing and Retrieval of Image Data," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 8, pp. 837-842, Aug. 1996.

[18] Y. Liu, K. Schmidt, J. Cohn, and S. Mitra, "Facial Asymmetry Quantification for Expression Invariant Human Identification," *Computer Vision and Image Understanding,* vol. 91, nos. 1/2, pp. 138-159, July 2003.

[19] Y. Liu and J. Palmer, "A Quantified Study of Facial Asymmetry in 3D Faces," *Proc. IEEE Int'l Workshop Modeling of Faces and Gestures,* pp. 222-229, Oct. 2003.

[20] Y. Liu et al., "Discriminative MR Image Feature Analysis for Automatic Schizophrenia and Alzheimer's Disease Classification," *Proc. Seventh Int'l Conf. Medical Image Computing and Computer Aided Intervention,* pp. 393-401, Oct. 2004.

[21] I. Matthews and S. Baker, "Active Appearance Models Revisited," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 135-164, Nov. 2004.

[22] I. Matthews, T. Iashikawa, and S. Baker, "The Template Update Problem," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 6, pp. 810-815, June 2004.

[23] C. Rasmussen and G. Hager, "Joint Probabilistic Techniques for Tracking Multi-Part Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 6, pp. 560-576, June 2001.

[24] D. Reid, "An Algorithm for Tracking Multiple Targets," *IEEE Trans. Aerospace and Electronic Systems,* vol. 17, pp. 122-130, Jan. 1981.

[25] S. Smith and J. Brady, "ASSET-2: Real-Time Motion Segmentation and Shape Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 8, pp. 814-820, Aug. 1995.

[26] C. Stauffer and W.E.L. Grimson, "Learning Patterns of Activity Using Real-Time Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 747-757, Aug. 2000.

[27] H. Stern and B. Efros, "Adaptive Color Space Switching for Face Tracking in Multi-Colored Lighting Environments," *IEEE Int'l Conf. Automatic Face and Gesture Recognition,* pp. 249-254, May 2002.

[28] G. Toussaint, "Note on Optimal Selection of Independent Binary-Valued Features for Pattern Recognition," *IEEE Trans. Information Theory,* vol. 17, no. 5, pp. 618-618, 1971.

[29] K. Toyama and Y. Wu, "Bootstrap Initialization of Nonparameteric Texture Models for Tracking," *Proc. European Conf. Computer Vision,* vol. 2, pp. 119-133, 2000.

[30] B. Zarit, B. Super, and F. Quek, "Comparison of Five Color Models in Skin Pixel Classification," *Proc. Int'l Workshop Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems,* pp. 58-63, Sept. 1999.

[31] J. Zhang and Y. Liu, "Cervical Cancer Detection Using SVM-Based Feature Screening," *Proc. Seventh Int'l Conf. Medical Image Computing and Computer Aided Intervention,* pp. 873-880, Oct. 2004.

[32] X. Zhou, R. Collins, T. Kanade, and P. Metes, "A Master-Slave System to Acquire Biometric Imagery of Humans at a Distance," *Proc. ACM SIGGM Int'l Workshop Video Surveillance,* pp. 113-120, Nov. 2003.

[33] Z. Zivkovic and B. Krose, "An EM-Like Algorithm for Color Histogram Based Object Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 798-803, June 2004.

Robert T. Collins received the PhD degree in computer science in 1993 from the University of Massachusetts at Amherst for work on recovering scene models using stochastic projective geometry. He is an associate professor in the Computer Science Engineering Department of Pennsylvania State University. His research interests include video scene understanding, automated surveillance, human activity modeling, and real-time tracking. He was coeditor of the August 2000 special issue of *IEEE Transactions on Pattern Analysis and Machine Intelligence* on the topic of video surveillance. In 1999, he was area chair for the computational geometry section of the IEEE Computer Vision and Pattern Recognition Conference. He routinely serves on the program committees for IEEE workshops on visual surveillance, performance evaluation of tracking and surveillance, and detection and recognition of events in video. He has been a reviewer for all of the major computer vision journals and conferences and serves on the US National Science Foundation review panels in the area of computer vision. He is a senior member of the IEEE and the IEEE Computer Society.

Yanxi Liu received the PhD degree in computer science from the University of Massachusetts at Amherst, where she studied group theory applications in robotics. Her postdoctoral training was performed in INRIA, Grenoble, France. She also won a research-education fellowship from the US National Science Foundation and spent one year at DIMACS (NSF center for Discrete Mathematics and Theoretical Computer Science). She is currently an associate research professor affiliated with both the Robotics Institute (RI) and the Center for Automated Learning and Discovery (CALD) of Carnegie Mellon University and an adjunct associate professor in the Radiology Department of the University of Pittsburgh. Her current research interests center on seeking discriminative feature subspaces in large image sets or videos with an emphasis on medical applications and developing a group theoretical computational framework for near-regular pattern/texture analysis and synthesis. Dr. Liu is the director of the Biomedical Image Analysis Lab. She is the chair for the ICCV 2005 workshop on the topic of "Computer Vision for Biomedical Image Applications: Current Techniques and Future Trends." She is a senior member of the IEEE and a member of the IEEE Computer Society.

Marius Leordeanu received the BA degree in both mathematics and computer science from Hunter College of The City University of New York, after he transferred from the Technical University of Cluj Napoca, Romania. From 2002 to 2003, he worked in the vision lab at Hunter College in the area of 3D registration and modeling. Since 2003, he has been a PhD student at the Robotics Institute of Carnegie Mellon University. At CMU, he works on tracking, learning, and object recognition.