Likelihood Map Fusion for Visual Object Tracking

Zhaozheng Yin† zyin@cse.psu.edu
Fatih Porikli‡ fatih@merl.com
Robert T. Collins† rcollins@cse.psu.edu

†The Pennsylvania State University
University Park, PA 16803

‡Mitsubishi Electric Research Labs
Cambridge, MA 02139

Abstract

Visual object tracking can be considered as a figure-ground classification task. In this paper, different features are used to generate a set of likelihood maps for each pixel indicating the probability of that pixel belonging to foreground object or scene background. For example, intensity, texture, motion, saliency and template matching can all be used to generate likelihood maps. We propose a generic likelihood map fusion framework to combine these heterogeneous features into a fused soft segmentation suitable for mean-shift tracking. All the component likelihood maps contribute to the segmentation based on their classification confidence scores (weights) learned from the previous frame. The evidence combination framework dynamically updates the weights such that, in the fused likelihood map, discriminative foreground/background information is preserved while ambiguous information is suppressed. The framework is applied here to track ground vehicles from thermal airborne video, and is also compared to other state-of-the-art algorithms.

1. Introduction

Visual object tracking has recently been addressed as a binary classification problem [1] [3] where object pixels must be discriminated from background pixels based on local image cues such as color or texture. To formulate object tracking as a discriminative figure-ground classification problem, two important issues need to be solved: what features to choose and how to build and combine classifiers. Furthermore, since the object and background change their appearance over time, on-line feature selection and classifier training are required to adapt the tracker to handle the appearance variations.

1.1 Related Work

Rather than predefining a specific feature for tracking, online feature selection [3] attempts to select the best distinguishing features that clearly separate object from background. In that work, linear combinations of (R, G, B) color space are mapped through a data-derived likelihood ratio feature to generate color feature candidates. The variance ratio is used to rank each candidate feature by how well it separates foreground and background color distributions. Features that maximize the separability are ranked most highly and are best suited to the tracking task.

The particle filter framework can be used to fuse diverse measurement sources. For example, in [10] stereo sound is fused with color for tele-conferencing and motion is fused with color for surveillance with a still camera. In both scenarios, the combination of cues proved to be more robust than any of the cues individually. The color likelihood function is generated by computing the Bhattacharyya similarity coefficient between a reference color histogram and hypothesized candidate color histogram. Motion information is represented by a histogram of motion amplitude. If the examined region contains no obvious movement, all the motion measurements will fall in the lower bins of the histogram. However it's not straightforward to construct a reference histogram for the motion measurement because the motion amplitude could spread over a large range.

The well-established ensemble methods [5] (sometimes called committee machines [12]) classify new samples (pixels) by taking a weighted combination of weak classifiers. The resulting strong classifier often performs better than any single weak classifier. For example, Avidan [1] constructs a feature vector including histogram of oriented gradients (HoG) [4] and RGB values for each pixel in the image. An ensemble of weak classifiers are trained to separate the object pixels from background pixels. Each weak classifier is formed by least squares regression of a hyperplane in the raw, multi-dimensional feature space. The final figure-ground separation is determined by a strong classifier trained via AdaBoost. In addition to AdaBoost used in [1], other classifier combination methods have been considered in [8]: product rule, sum rule, max rule, min rule, median rule and majority vote rule. The experimental comparison in [8] demonstrates that the sum rule outperforms other combination schemes.

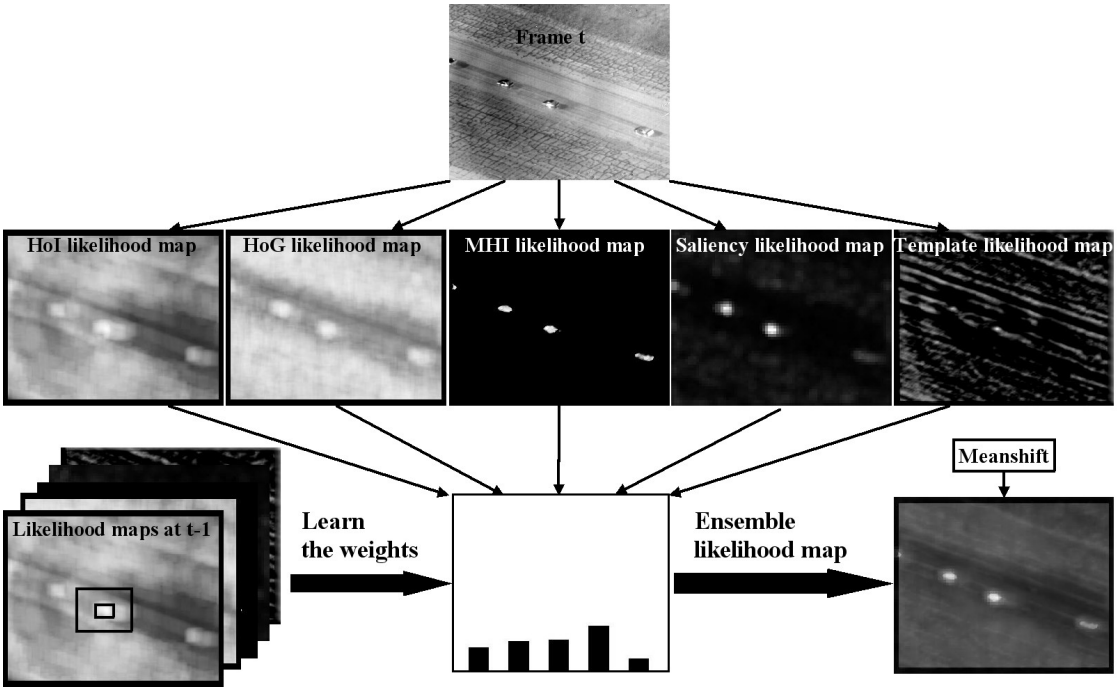

Figure 1: Five likelihood maps are generated based on (1) histogram of intensity (HoI), (2) histogram of oriented gradients(HoG), (3) motion detection, (4) saliency and (5) template matching. The confidence score (weight) of each likelihood map is computed from the previous frame by considering the foreground/background separability of the previous likelihood map. The ensemble of likelihood maps are integrated together by weighted linear fusion. Mean-shift tracking is performed on the fused likelihood map (a type of "soft segmentation") to localize the object.

1.2 Our Proposed Framework

Previous work can be seen as performing fusion at one of two levels. For example, in data level (or feature-level) fusion, Collins et al. [3] select the best discriminative feature as a projection from one feature space ($\{R, G, B\}$) to another ($\{\omega_1 R + \omega_2 G + \omega_3 B\}$ where ω_i are weights). Perez et al. [10] combine color, motion and sound localization cues in a particle filter framework. In decision level (or classifier-level) fusion, Avidan [1] integrates different weak classifiers via AdaBoost to yield a final figure-ground separation.

In this paper, we propose a novel likelihood map fusion framework that differs from feature selection or classifier training. As illustrated in Figure 1, different feature extraction mechanisms (e.g. intensity, texture, motion, saliency and template) are applied on the input frame to generate likelihood maps (also called confidence map in [1], weight image in [3] and likelihood function in [10]). A pixel value in the likelihood map is proportional to the probability that the pixel belongs to the object versus the background, i.e., pixels more likely to be on objects have higher likelihood and brighter value. Thus, each likelihood map makes a "soft" decision on the figure-ground separation. The ensemble of likelihood maps are combined into a single likelihood map by a weighted linear fusion. The confidence score (weight) of each current likelihood map is adaptively determined by measuring the foreground and background separability using the likelihood map from the previous frame. Mean-shift tracking [2] is then performed on the fused likelihood map to find the object location.

This framework can be adapted to use any feature for which we can generate a likelihood map (soft segmentation). When such heterogeneous features rely on different image information, they are complementary and the fused likelihood map gains some desirable properties. For example, useful discriminative information is preserved while unwanted ambiguous information is suppressed; when one feature fails (e.g. motion features on a static object), the framework can rely on more relevant features. Since the individual likelihood maps are generated independently and their confidence scores are computed independently, the "front-end" of the framework can be executed in parallel. This is different from the AdaBoost algorithm, which trains weak classifiers sequentially. It is usually not straightforward to build a single feature vector for each pixel to embed different modalities represented by heterogeneous features. This trouble is avoided here because no classifier training is required.

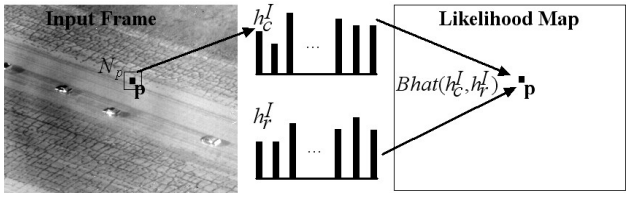

Figure 2: The intensity appearance likelihood is computed at pixel p as the Bhattacharyya similarity coefficient between the reference and candidate histograms of intensity.

2 Likelihood Maps

As a generic framework, the proposed approach can fuse likelihood maps produced by several heterogeneous information sources. To illustrate the approach, here we use likelihood maps generated by information sources that use color/intensity, texture, motion detection, saliency and template matching.

2.1 HoI Likelihood Map

Color or intensity histograms are widely used to represent appearance of rigid or non-rigid objects during visual tracking [2]. The tracked object or region of interest can have any complex shape or boundary. To quickly calculate the histogram in a new input image, we adopt an axis-aligned rectangular box to model the object shape. As shown in Figure 2, the candidate histogram located at pixel p, h_c^I, is computed within its surrounding box N_p. Given the object reference histogram of intensity (HoI), h_r^I, the likelihood at pixel p is determined by the Bhattacharyya similarity coefficient

$$\text{Bhat}(h_c^I, h_r^I) = \sum_{i=1}^{B} \sqrt{h_{i,c}^I h_{i,r}^I} \quad (1)$$

where B is the number of histogram bins.

For real-time tracking purposes, we only compute the likelihood map within a gating region (subimage in Figure 3). Subsampling the image can also be applied to reduce the computational cost [10]. We use the integral histogram method [11] to speed up our histogram-based likelihood map computation. Once an integral histogram $H(u, v)$ in the subimage has been computed, the histogram of any rectangular region with sides parallel to the image coordinates can be easily computed as a linear combination of four vectors. For example, in Figure 3, the histogram of the small box around the motorcycle can be computed as

$$h_c^I = H(u_2, v_2) - H(u_2, v_1) - H(u_1, v_2) + H(u_1, v_1) \quad (2)$$

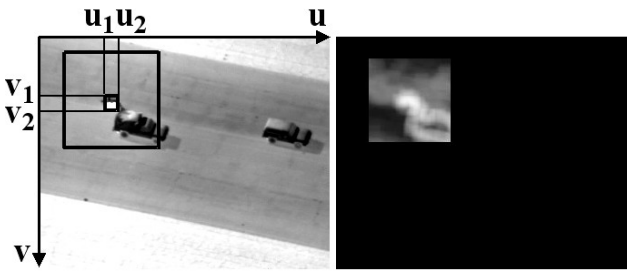

Figure 3: Histogram-based likelihood map computation is accelerated by using a gating region and the integral histogram methods.

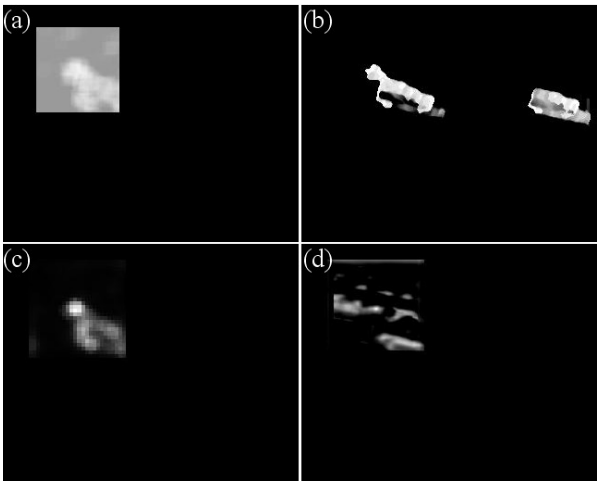

Figure 4: (a) HoG likelihood map; (b) MHI likelihood map; (c) Saliency likelihood map; (d) template matching likelihood map.

2.2 HoG Likelihood Map

The histogram of oriented gradients (HoG) representation [4] has been successfully applied to pedestrian detection. We use HoG here to capture local object texture information that is insensitive to color/intensity changes. Each pixel's gradient orientation is quantized to determine a histogram bin, into which the pixel's gradient magnitude is added. Similar to HoI likelihood map computation, the HoG likelihood at any pixel is determined by the Bhattacharyya similarity coefficient between reference and candidate HoGs. The method of integral histograms is also applied here to speed up the computation. Figure 4(a) is an example of the HoG likelihood map computed in a gating region.

2.3 MHI Likelihood Map

To get object motion information from a moving camera, we first must compensate for the camera motion. After camera motion compensation, frame differencing between two

consecutive frames usually indicates where the object is in the current frame and where the object was in the previous frame. To compute a more accurate motion mask in the current frame, we adopt the forward/backward motion history image (MHI) method, which combines information over a larger temporal window [13]. Figure 4(b) is an example of a MHI likelihood map computed over the whole image. The motion likelihood value is normalized into the range $[0, 1]$.

2.4 Saliency Likelihood Map

The ability of the human vision system to detect salient objects in complex scenes in real time has inspired several computational models of visual attention. For example, Itti et al. proposed a conceptually simple computational model for saliency-driven visual focus of attention[7]. A bottom-up saliency map is built where only locations that locally "jump out" from their surroundings can persist. In other words, the saliency map reflects object conspicuity, i.e. an object distinct from its surrounding background deserves visual attention. We adopt a simple and fast saliency detection method based on image spectrum analysis [6]. The assumption is that statistical singularities in the spectrum may be responsible for anomalous regions in the image, i.e. the spectral residual indicates the innovation in the image. Figure 4(c) is an example of a saliency likelihood map computed in a subimage, with values normalized into the range $[0, 1]$.

2.5 Template Likelihood Map

Correlation with the color/intensity template contained within the object's rectangular bounding box could also be used as a feature to compute a likelihood map. Figure 4(d) is an example of a template matching likelihood map computed in a subimage. The template matching is executed by normalized cross-correlation and has an output value range of $[-1, 1]$. We set the negative values to zero in the template matching likelihood map, while keeping the positive values unchanged.

3 Likelihood Map Fusion

Our tracking goal is to infer state variable x (e.g. object location) from input image I, i.e. solving $P(x|I)$. Given a set of likelihood maps $P(x|L_i)$ based on different feature extraction mechanisms, we apply Bayesian inference to obtain

$$P(x|I) = \int P(x|L)P(L|I)dL \approx \sum_{i=1}^{M} w_i P(x|L_i) \quad (3)$$

where M denotes the number of likelihood maps, and $w_i = P(L_i|I)$ represents our degree of belief about the validity of each individual likelihood map and sums to one over the range $i = 1, \ldots, M$. Thus $P(x|I)$ is approximated by an ensemble of likelihood maps.

The particular problem now at hand is how to derive the proper weighting value w_i, i.e. how much we can trust each likelihood map $P(x|L_i)$. Under the squared loss function, the optimal w_i in the current frame can be solved by minimizing the minimum mean square error with respect to the true probability $P_t(x|I)$.

$$\begin{aligned} \text{MMSE} &= E[(\sum_{i=1}^{M} w_i P(x|L_i) - P_t(x|I))^2] \\ &= E[(\sum_{i=1}^{M} w_i (P(x|L_i) - P_t(x|I)))^2] \end{aligned}$$

Note that $\sum_{i=1}^{M} w_i = 1$ and $P_t(x|I)$ is deterministic. Denoting $\bar{w} = [w_1, \ldots, w_M]$ and $C_{ij} = E[(P(x|L_i) - P_t(x|I))(P(x|L_j) - P_t(x|I))]$, the above MMSE equals $\bar{w}^T C \bar{w}$, where C is a symmetric positive definite covariance matrix with elements C_{ij}. The solution to the optimization problem is (see Appendix for the derivation)

$$\bar{w}_{opt} = \frac{C^{-1}\bar{\mathbf{1}}}{\bar{\mathbf{1}}^T C^{-T} \bar{\mathbf{1}}} \quad (4)$$

where $\bar{\mathbf{1}} = [1, \ldots, 1]$. If the likelihood maps are uncorrelated, i.e. $C_{ij} = 0 \; \forall \; i \neq j$, and $E[P(x|L_i)] = P_t(x|I)$ (i.e. unbiased estimation), the optimal w_i's have a simple formulation

$$w_i = \frac{C_{ii}^{-1}}{\sum_{j=1}^{M} C_{jj}^{-1}} \quad (5)$$

Unfortunately, we don't know the true probability $P_t(x|I)$ for the current frame, and it is therefore infeasible to get w_i directly by Eq. 5. One possible way to overcome this problem is to perform unsupervised parameter learning on the current unlabeled pixel data, but that would be very challenging. With the realization that w_i is related to the corresponding likelihood map variance, we may approximate w_i in the previous frame using the variance ratio [3]:

$$\text{VR}(L_i; p, q) = \frac{var(L_i; (p+q)/2)}{var(L_i; p) + var(L_i; q)} \quad (6)$$

$$w_i \approx \frac{\text{VR}(L_i; p, q)}{\sum_{j=1}^{M} \text{VR}(L_j; p, q)} \quad (7)$$

where p represents the object distribution and q represents the surrounding background distribution in the likelihood map L_i. Here w_i is inversely proportion to the within-class likelihood map variance and proportion to the between-class likelihood map variance. If likelihood values of pixels on both the object and background are tightly clustered (low

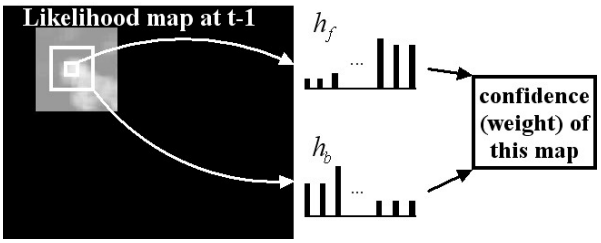

Figure 5: The confidence score of each likelihood map is computed based on separability of histograms of likelihood values on the object versus values from the surrounding background region, measured from the likelihood map of the previous frame.

within-class variance) and the two classes are well separated (high between-class variance), this likelihood map is good for tracking and deserves a high confidence score (weight).

In [9], the best set of weights for classifiers are obtained using an exhaustive search method by changing the weights incrementally between zero and one. In contrast, Figure 5 illustrates our process of computing the confidence score (weight) of each map based on separability. For each likelihood map in the previous frame, we extract two histograms of likelihood values: one within the bounding box thought to contain the object, and one from a surrounding ring of pixels that presumably contain mostly background. The confidence of the likelihood map is measured by the variance ratio between the foreground histogram and the surrounding background histogram [3]. The larger the variance ratio is, the more we can trust the likelihood map.

The feature confidence scores computed from the previous frame are applied at the current frame to fuse all likelihood maps. Only likelihood maps with high confidence make a significant contribution to the final decision. Since the confidence scores are updated at each frame, the framework can adapt to various situations (e.g. a moving object that suddenly stops). Figure 6 shows a comparison between weighted fusion and the sum-rule result. Sum-rule [8] is a special case corresponding to all weights being equal. Our confidence-weighted fusion produces a higher contrast distribution between the foreground object blob location and the surrounding background.

4 Experimental Results

We test our likelihood map fusion framework on a set of airborne thermal sequences for which purely color-based tracking approaches are not feasible. Figure 7 shows four examples. In the first video, a small vehicle passes several similar cars. In the second video, a truck becomes partially occluded by trees and suddenly stops. The illumination and

Figure 6: Left: weighted fusion; Right: sum-rule fusion.

pose change sharply in this sequence. In the third video, a motorcycle becomes partially occluded by nearby vehicles. In the fourth video, a vehicle is static while the camera moves around it. For these sequences, a mean-shift tracker works successfully on the fused likelihood maps. However, it is clear to see that some individual likelihood maps are not discriminative enough, and a tracker would drift easily if only using likelihood maps based on HoI or HoG features.

We compare the results with on-line feature selection [3] and ensemble tracking [1]. The intensity channel of a thermal image is copied to three (R, G, B) channels to run the on-line feature selection, but in fact, only one intensity feature can be selected here. As shown in Figure 7, the algorithm drifts to nearby similar background regions in the first three sequences but works successfully in the fourth sequence. The on-line selection could work better if more features were available. The ensemble tracker drifts very quickly in the first three sequences and locks on the vehicle's shadow in the fourth sequence. The original ensemble tracker uses images at three scales (full-size, half-size and quarter-size) to train weak classifiers. It seems that the ensemble tracker works better with bigger objects, as the quarter-scale image can make small objects into 1-2 pixel regions that are very sensitive to outliers. When the object bounding box contains many background pixels (outliers), the tracker fails more quickly. We changed the implementation to use full-size images only and used a suitable bounding box, but the tracker still drifted as shown in Figure 8.

Unfortunately, there are also some failure cases for all three trackers. As shown in Figure 9, when the small motorcycle is totally occluded by a truck and some parts of the truck have similar features with the motorcycle, the mean-shift tracker will drift to another local modes. After the motorcycle passes the truck, there are two modes appearing in the fused likelihood map: one corresponding to the motorcycle and the other belonging to the truck. To prevent the mean-shift tracker from locking onto a local false mode, particle filters could be applied to find the global mode [1]. Constant velocity motion prediction and multiple object tracking could also be used to assist the tracking

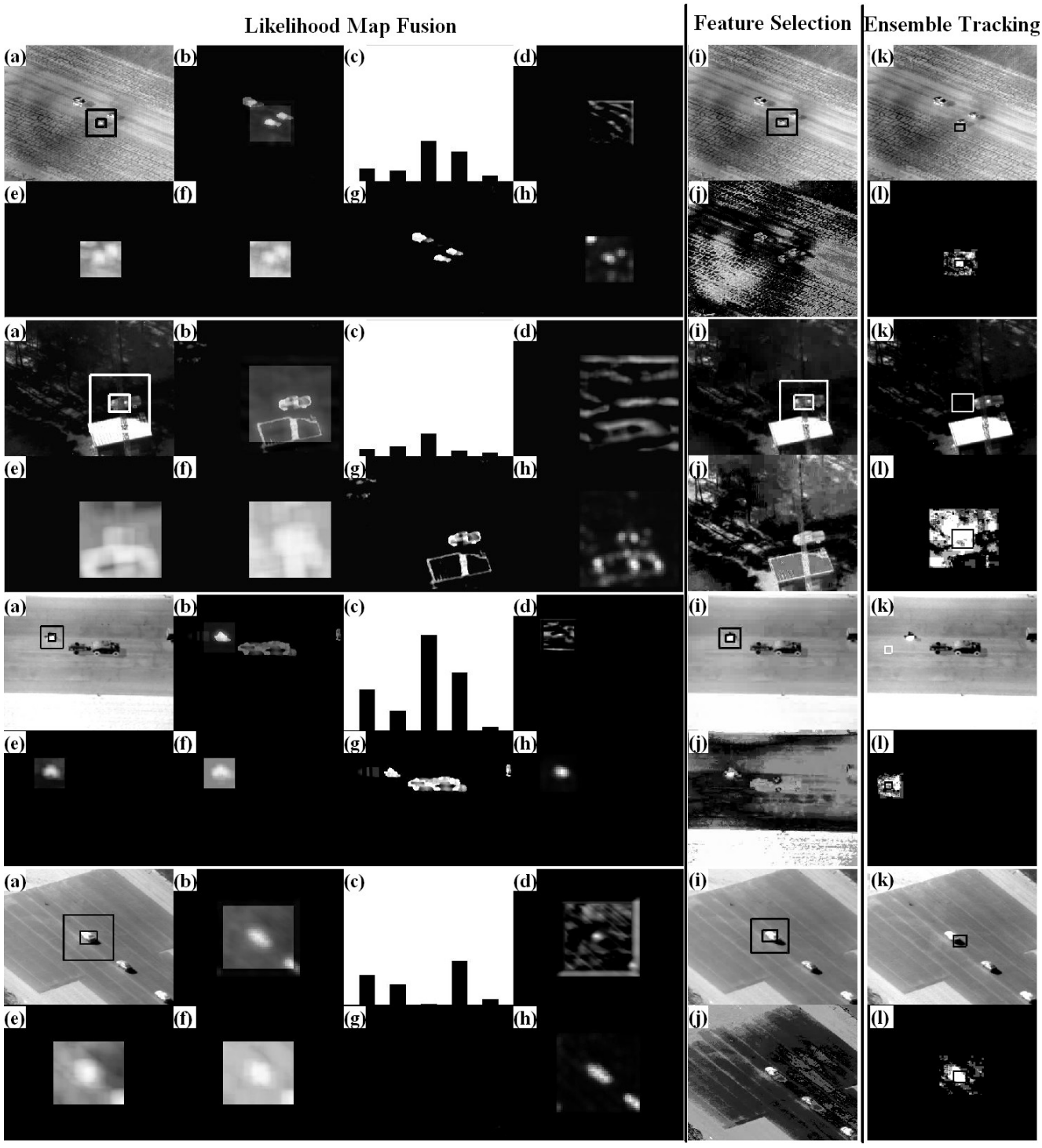

Figure 7: (a) Tracking result overlaid on the input frame; (b) fused likelihood map; (c) the confidence scores shown in the bar plot correspond, from left to right, to: HoI likelihood map (e), HoG likelihood map (f), MHI likelihood map (g), saliency likelihood map (h) and template matching likelihood map (d) separately; (i) on-line feature selection tracking; (j) weight image of (i); (k) ensemble tracking; (l) confidence map of (k) in a gating region.

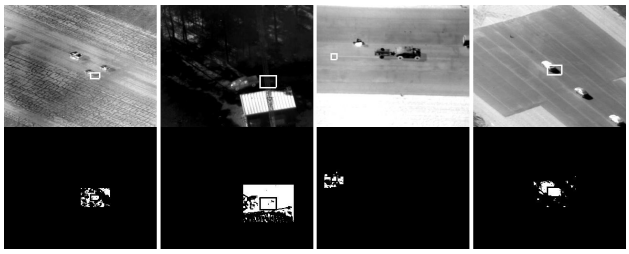

Figure 8: Ensemble tracker on the full size only.

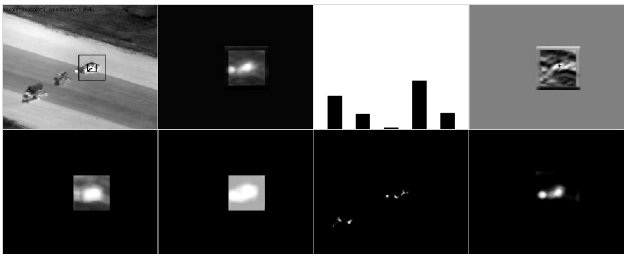

Figure 9: A failure case: the small object becomes totally occluded by another object. We have not used motion prediction or particle filter extensions to the basic tracker.

in such cases of occlusion.

5. Conclusion

This paper presents a generic framework to fuse different likelihood maps based on their confidence. There is no need to embed heterogeneous features into a single vector for classifier training. Instead, each likelihood map makes a "soft" decision about the proper figure-ground segmentation. The confidence scores of all likelihood maps are updated on-line to adapt continuously to changing lighting, pose and motion conditions. The experimental results verify that the fused likelihood map preserves discriminative information while suppressing ambiguous information.

Appendix

To minimize $\bar{w}^T C \bar{w}$ subject to $\bar{w}^T \bar{\mathbf{1}} = 1$, we use the Lagrangian function

$$J = \bar{w}^T C \bar{w} - 2\lambda(\bar{w}^T \bar{\mathbf{1}} - 1)$$

Setting the gradient

$$\frac{\partial J}{\partial \bar{w}} = 2C\bar{w} - 2\lambda\bar{1}$$

to zero, we get

$$\bar{w} = \lambda C^{-1} \bar{\mathbf{1}}$$

Substituting $\bar{w}$ back into the constraint, we have

$$\lambda \bar{\mathbf{1}}^T C^{-T} \bar{\mathbf{1}} = 1$$

Thus $\lambda = (\bar{\mathbf{1}}^T C^{-T} \bar{\mathbf{1}})^{-1}$, and

$$\bar{w}_{opt} = \frac{C^{-1}\bar{\mathbf{1}}}{\bar{\mathbf{1}}^T C^{-T} \bar{\mathbf{1}}}$$

References

[1] S. Avidan, "Ensemble Tracking," IEEE Trans. Pattern Analysis and Machine Intelligence, 29(2): 261-271, 2007

[2] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-Based Object Tracking," IEEE Trans. Pattern Analysis and Machine Intelligence, 25(5): 564-577 May 2003.

[3] R. Collins, Y. Liu and M. Leordeanu, "On-line Selection of Discriminative Tracking Features," IEEE Trans. Pattern Anal. and Machine Intell., 27(10): 1631-1643 Oct. 2005.

[4] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," CVPR2005

[5] T. Dietterich, "Ensemble methods in machine learning," In J. Kittler and F. Roli (Ed.) First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science (pp. 1-15). New York: Springer Verlag.

[6] X. Hou and L. Zhang, "Saliency Detection: A Spectral Residual Approach," CVPR2007

[7] L. Itti, C. Koch and E. Niebur, "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis," IEEE Trans. Pattern Anal. and Machine Intell., 20(11): 1254-1259, 1998.

[8] J. Kittler et al, "On Combining Classifiers," IEEE Trans. Pattern Analysis and Machine Intelligence, 20(3): 226-239 March 1998.

[9] J. Kittler and S. A. Hojjatoleslami, "A Weighted of Classifiers Employing Shared and Distinct Representations," CVPR1998

[10] P. Perez, J. Vermaak and A. Blake, "Data fusion for visual tracking with particles," Proc. IEEE, 92(3):495-513, 2004.

[11] F. Porikli, "Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces," CVPR2005.

[12] V. Tresp, "Committee Machine," In Y. Hu and J. Hwang (Ed.) Handbook for Neural Network Signal Processing, CRC Press 2001.

[13] Z. Yin and R. Collins, "Moving Object Localization in Thermal Imagery by Forward-backward MHI," Third IEEE Workshop on Object Tracking and Classification in and Beyond the Visible Spectrum (OTCBVS'06), June 2006.