Object Tracking and Detection after Occlusion via Numerical Hybrid Local and Global Mode-seeking

Zhaozheng Yin and Robert T. Collins
Department of Computer Science and Engineering
The Pennsylvania State University, University Park, PA 16802
{zyin,rcollins}@cse.psu.edu

Abstract

Given an object model and a black-box measure of similarity between the model and candidate targets, we consider visual object tracking as a numerical optimization problem. During normal tracking conditions when the object is visible from frame to frame, local optimization is used to track the local mode of the similarity measure in a parameter space of translation, rotation and scale. However, when the object becomes partially or totally occluded, such local tracking is prone to failure, especially when common prediction techniques like the Kalman filter do not provide a good estimate of object parameters in future frames. To recover from these inevitable tracking failures, we consider object detection as a global optimization problem and solve it via Adaptive Simulated Annealing (ASA), a method that avoids becoming trapped at local modes and is much faster than exhaustive search. As a Monte Carlo approach, ASA stochastically samples the parameter space, in contrast to local deterministic search. We apply cluster analysis on the sampled parameter space to redetect the object and renew the local tracker. Our numerical hybrid local and global mode-seeking tracker is validated on challenging airborne videos with heavy occlusion and large camera motions. Our approach outperforms state-of-the-art trackers on the VIVID benchmark datasets.

1. Introduction

The goal of visual object tracking is to repeatedly localize an object in successive frames. Most object trackers search for the target locally in new frames, and consist of several key components: (1) an object representation (e.g. appearance model by color histogram [8], shape model by active contours [5], or bag of samples for classification [2]); (2) a similarity measure between the reference model and candidate targets (e.g. Bhattacharya coefficient [8], Earth Mover's Distance [24], or classifier scores); and (3) a local mode-seeking method for finding the most similar location in new frames (e.g. mean-shift [8] or Lucas-Kanade [4]).

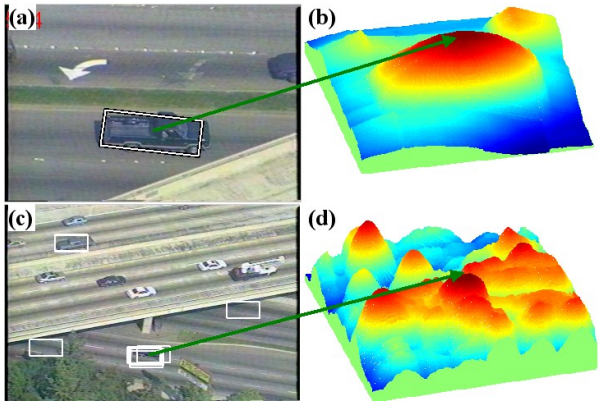

Figure 1. (a)-(b) Tracking during normal conditions is solved by local mode-seeking; (c)-(d) When the object becomes occluded, for example this vehicle passes under a bridge, global mode-seeking is needed to detect the object after occlusion and reinitialize the local tracker.

Efficient and robust local mode-seeking methods are of critical importance to the tracking problem, which is why the mean-shift hill climbing method has been popular for ten years. The original mean-shift tracker [8] uses color histograms as an object representation and Bhattacharya coefficient as a similarity measure. An isotropic kernel is used as a spatial mask to smooth a histogram-based appearance similarity function between model and target candidate regions. The mean-shift tracker climbs to a local mode of this smooth similarity surface to compute the translational offset of the target blob in each frame.

In the mean-shift framework, many efforts have considered tracking an object with changing scale and orientation. For example, multiple symmetric kernels can be used to track object parts and compute the whole object's orientation [15][13]. An anisotropic (elliptic) kernel mean-shift algorithm for image and video segmentation is proposed in [27]. The kernel associated with each pixel adapts to the local structure by adjusting its shape, scale and orientation. Anisotropic kernels and sample point density estimation both add complexity to the mean-shift procedure. As-

suming objects have ellipsoidal regions, Zivkovic and Krose [31] approximate the object shape by a local covariance matrix and update it using the EM algorithm. Yilmaz [29] presented an asymmetric kernel mean-shift algorithm to estimate object orientation and scale. The "asymmetric" property is achieved by introducing a level set kernel to represent a complex object shape, leading to better contour tracking. A look-up table that encodes the scale observed at each angle is created for generating the modified level set kernel.

During normal tracking conditions when the object is visible and moves predictably from frame to frame, local mode-seeking trackers are able to localize the object. However, when the object becomes partially or totally occluded, or motion is large and unpredictable, such local tracking is prone to failure. In Figure 1, the vehicle's scale and location change quickly in the image sequence after it passes under a bridge because the cameraman zoomed out to find it again. When common prediction techniques like the Kalman filter do not provide a good estimate of object parameters in the new frame, global object detection is required to recover from tracking failure. For example, Collins et al. [10] use peak difference to detect the object while avoiding false peaks. Avidan [2] proposes to spread particle filters in possible future locations to detect an object after occlusion. Shen et al. [25] use multi-bandwidth mean-shift to seek the global mode.

With the realization that tracking is an optimization problem given a suitable similarity measure (also called distance measure, cost/energy function, or objective function), we consider object tracking and detection in a generic numerical optimization framework. The optimization function, f, is a black-box function inside which the object representation and similarity measure can be of any type, even something that we can't take analytic derivatives of. Changing the object representation and similarity measure inside the black-box doesn't affect the outside numerical mode-seeking algorithms. Furthermore, although different kernels have been used under the mean-shift framework to find object location (u, v), scale s and orientation θ, the problem has not been attacked directly by basic numerical methods. For example, if we design a black-box function that can accept (u, v, s, θ) as input state x, and output a similarity value $f(x)$ between the model and candidate, then during normal tracking conditions the parameterized object state can be found directly by numerical local mode-seeking, i.e. solving $argmax_x f(x)$ without deliberately designing any specific kernel, shape map, or an extra ellipse-fitting step. When the local tracker loses the object, global mode-seeking can be applied with the same black-box function to detect the object and re-initialize the local tracker.

In Section 2 we formalize object tracking as a local numerical optimization problem in a parameter space representing the object's state. Section 3 presents object detection as a global optimization problem and solves it via Adaptive Simulated Annealing with cluster analysis. In Section 4, we validate our numerical hybrid local and global mode-seeking approach on challenging video sequences.

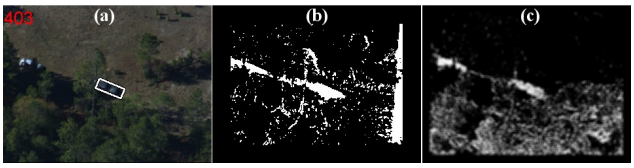

Figure 2. (a) Vehicles move through the woods and undergo large appearance change and heavy occlusion; (b) Motion detection by accumulated frame differencing; (c) Motion detection by forward/backward motion history images.

2. Tracking by Local Mode-seeking

In previous symmetric/asymmetric kernel mean-shift tracking, each pixel is assigned a weight that votes for the direction and magnitude of the mean-shift vector. Objects are tracked by computing the kernel motion iteratively in the form of a parametric transformation (e.g. translation (u, v), orientation θ, scaling s). Here, we directly seek the mode in a 4D parameter space by numerically maximizing the objective function $f(u, v, \theta, s)$ without specially designing an analytic kernel.

2.1. Objective Function (Feature and Measure)

Many types of object features have been used for tracking and detection. For example, color histograms are an efficient object representation for appearance-based tracking of nonrigid objects [8]. Histogram of oriented gradients (HoG) is a powerful texture representation for object detection [12]. When object appearance changes greatly and heavy occlusion exists (Figure 2(a)), motion is a good feature for moving object tracking because it is invariant to changes in object color, texture or shape. In [1], accumulated frame differencing is performed to give a rough estimate of the moving object boundary (Figure 2(b)), and a level-set based segmentation is applied to refine it. In [30], moving objects are detected by forward/backward motion history images, yielding a tighter boundary estimate around the target. The motion likelihood map that indicates each pixel's possibility of being a moving pixel is also useful for object tracking (Figure 2 (c)). All of these features can be used in our black-box objective function, either individually or jointly.

When a motion likelihood map, $M(i)$, is used, we try to find a local region inside which the majority of pixels are moving and few are stationary. The optimization objective function is

$$f(x) = \sum_{i \in R(x)} M(i) \tag{1}$$

where $R(x)$ is a region determined by a point x in the 4D parameter space (u, v, θ, s).

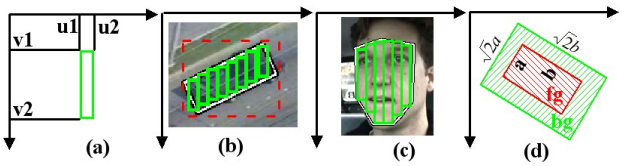

Figure 3. (a) Computing the sum or histogram inside the green rectangle is done efficiently by the integral image or integral histogram methods; (b) The sum or histogram inside the rotated white rectangle is computed from the piecewise union of green rectangles, each of which is performed using an integral image or integral histogram; (c) The piecewise integration idea can be generalized to any complex shape; (d) The sum or histogram inside the ring between two rotated rectangles can be computed by subtracting the inside rectangle from the outside one.

When a histogram of color or histogram of oriented gradients is used as a feature, we can use the Bhattacharyya coefficient as the similarity measure

$$f(x) = \sum_{k=1}^{m} \sqrt{p(x)} \times \sqrt{q} \tag{2}$$

where q is the m-bin reference histogram and $p(x)$ is the target candidate histogram computed in region $R(x)$, both normalized to sum to one so that they are m-bin probability mass functions. As a bin-by-bin similarity, Bhattacharyya coefficient only considers bins with the same index without using cross-bin information. A weighted form of cross-bin similarity is the negative quadratic form distance [14]

$$f(x) = -(p(x) - q)^T A(p(x) - q) \tag{3}$$

where $A = [a_{ij}]$ is a matrix and the weights a_{ij} denote similarity between bin i and j. For example, $a_{ij} = 1 - d_{ij}/d_{max}$ where d_{ij} is the ground distance between bin i and j and $d_{max} = max(d_{ij})$, i.e. A is a Toeplitz matrix. Another robust similarity metric between two distributions, Earth Mover's Distance (EMD) [24], can also be used here. It is based on the minimal cost that must be paid to transform one distribution into the other, and it is robust to color shift. All of these similarity measures can be applied inside our black-box objective function without changing the outside numerical optimization algorithms.

To reduce computational cost of evaluating an objective function, the integral image method [26] can be applied on motion likelihood maps, and the integral histogram method [23] can be applied to histogram calculations. As shown in Figure 3(a), once an integral image/histogram $H(u, v)$ in the current frame has been computed, the sum/histogram of any rectangular region with sides parallel to the image coordinates can be rapidly computed as a linear combination of four vectors:

$$h = H(u_2, v_2) - H(u_2, v_1) - H(u_1, v_2) + H(u_1, v_1) \tag{4}$$

Lienhart et al.[20] extend the original integral image to compute the sum within a 45° rotated rectangle (available in OpenCV).

To compute the sum/histogram of any rotated rectangular region (Figure 3(b)), or any region of arbitrary shape for that matter (Figure 3(c)), we break the shape into a piecewise union of rectangles with sides parallel to the image coordinates. The sum/histogram inside each of these oriented rectangles is calculated very quickly by Eq.4. Figure 3(d) shows how to compute the sum/histogram inside a ring. This process is fast because it does not assign different weights to pixels during different iterations, as the common kernel methods do. For each image, the integral image/histogram is computed only once at the start of the whole mode-seeking process. To compute histogram $p(x)$ or motion likelihood sum within an arbitrary region $R(x)$, we only need to center the region at location (u, v), orient it by θ, scale it by s, break it into piecewise axis-aligned rectangles, and then rapidly compute the sum/histogram using the piecewise integral image/histogram method.

When searching in scale space, it is important to consider background information to avoid the shrinkage problem [9]. Thus our objective function becomes

$$F(x) = f_{fg}(u, v, \theta, s) - f_{bg}(u, v, \theta, s) \tag{5}$$

When maximizing $F(x)$, we are actually seeking x to maximize the similarity between the model and foreground while minimizing the similarity between the model and background. This is consistent with the idea of discriminating between foreground and background regions [2][10].

2.2. Numerical Local Mode-seeking Methods

Remarkable progress has been made in the past forty years on the theory of unconstrained optimization of smooth functions. In general, Nocedal and Wright [22] summarize two fundamental strategies for moving from the current point x_k to a new iterate x_{k+1}: line search and trust region methods. Steepest ascent method, Newton's method, conjugate gradient method and BFGS method all fall within the line search framework as

$$x_{k+1} = x_k + \alpha_k d_k \tag{6}$$

where d_k is a search direction and α_k is a step size obtained by a one-dimensional search. For steepest ascent method,

$$d_k = g_k \tag{7}$$

where g_k is the gradient at the current point x_k, computed as a finite central difference. For Newton's method,

$$d_k = \frac{g_k}{\nabla^2 f(x_k)} \tag{8}$$

In the BFGS method, the Hessian matrix $\nabla^2 f(x_k)$ is approximated by a matrix B_k computed using the observed function values and gradient information. A search direction is then computed as

$$d_k = B_k^{-1} g_k \tag{9}$$

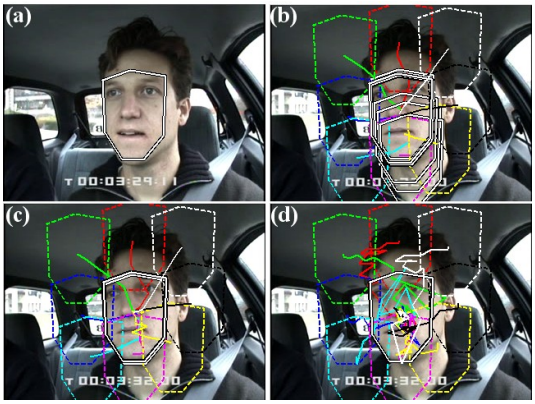

Figure 4. Dashed polygons represent different starting positions. Solid polygons are the modes found by local mode-seeking methods. The solid lines show the search paths. (a) The reference appearance model is represented by the color histogram inside the polygon; (b) Steepest ascent method; (c) Trust region algorithm; (d) Nelder-Mead simplex algorithm.

Local methods	Steepest	Trust region	Simplex
Comp. cost	96.0	206.3	52.8

Table 1. Average # of times the objective function f is evaluated.

Different approaches have been tried to improve the optimization part of the mean-shift tracker. For example, Shen et al. [25] use an adjustable step size during mean-shift tracking,

$$x_{k+1} = x_k + \alpha_k \mathrm{MV}_h(x_k) \tag{10}$$

where $\mathrm{MV}_r(x_k)$ is the mean-shift vector with smooth kernel of bandwidth r. Yang et al. [28] propose a BFGS method to accelerate steepest-ascent mean-shift by setting

$$d_k = B_k^{-1} \mathrm{MV}_h(x_k) \tag{11}$$

In fact, standard mean-shift tracking is also a steepest ascent procedure, and can be considered as one example of the line-search framework with $d_k = \mathrm{MV}_r(x_k)$ and $\alpha_k = 1$.

In the second strategy, known as the trust region method, typically a quadratic function is constructed around the current point x_k to approximate the true function f, while restricting the search range to a trust region. Depending on the performance of the candidate solution, the trust region will expand or shrink. Liu et al. [21] show that trust-region tracking is more effective than line-search mean-shift tracking.

A shared assumption in the above local-mode seeking methods is that the objective function f is smooth and its gradient g is available. However, for vision-based tracking applications, our objective function could be quite noisy. One popular derivative-free optimization method is the Nelder-Mead simplex method. In each iteration, the vertex with the worst function value is removed and replaced with a new point with a better value. The new point is obtained by reflecting, expanding or contracting the simplex.

We compare the classic steepest ascent (mean-shift), trust region [6] and simplex algorithms [19] via a face contour alignment experiment (Figure 4). The color histogram within a polygon in the first frame (Figure 4(a)) is used as the reference model. Different initial points are chosen in the test image. In this experiment, the steepest ascent method (Figure 4(b)) and trust region method (Figure 4(c)) have straightforward search paths, while the simplex algorithm (Figure 4(d)) has a more zig-zag-like path because the simplex is reflecting, expanding or contracting in each iteration. However, steepest ascent suffers from being trapped at local modes. Since no gradient information is needed in the simplex method, the objective function is evaluated the fewest number of times among the compared methods (Table 1), which could be an important difference if the objective function is expensive to compute.

3. Detection by Global Mode-seeking

When the tracked object becomes partially or totally occluded, local mode-seeking is prone to failure. To allow a local tracker to track through occlusions, motion prediction techniques like the Kalman filter are often used. However, when the prediction is far away from the true position, a global detection is required to recover from tracking failure. In this section, we consider a successful global optimization technique, simulated annealing [18], and adapt it to the problem of visual object detection.

3.1. Adaptive Simulated Annealing

The term simulated annealing derives from the physical process of heating and then slowly cooling a substance until the system settles to a minimum energy configuration. The initial temperature T_0 is set high enough to avoid being trapped at local modes. Enough perturbation at each temperature in the cooling process is needed to arrive at thermal equilibrium. Since typical annealing schedules for temperature T at annealing time k, like Boltzmann annealing ($T = T_0/lnk$) and Cauchy annealing ($T = T_0/k$), are slow, we adopt Adaptive Simulated Annealing [16], which exponentially decreases temperature in D-dimensional space

$$T = T_0 e^{-ck^{\frac{1}{D}}} \tag{12}$$

where c is some constant. In ASA the exponential annealing schedules permit resources to be spent adaptively on reannealing and on pacing the convergence in all dimensions, ensuring ample global searching in the first phase of search and ample quick convergence in the final phase [16].

Theoretically, if the temperature decreases extremely slowly, simulated annealing will find the global mode starting from any initial point, but the time to achieve such a solution would be unacceptable. Realizing that ASA relies on Monte Carlo importance-sampling in the parameter space, i.e. the Metropolis algorithm iteratively visits those

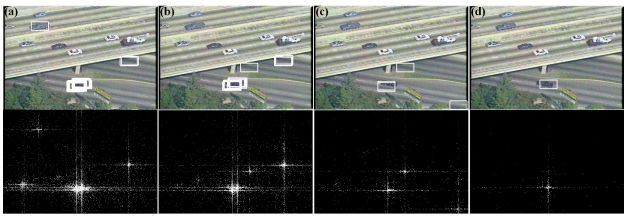

Figure 5. Detecting an object in 2D translation space for a given (θ, s). The top row shows the detection results with white boxes representing the modes found by ASA. The bottom row shows corresponding sampling maps where the bright pixels represent the sampled points with their objective function values. From left to right: ASA detection restarted from 25, 16, 4 and 1 different initial points, respectively.

# of initial pts	25	16	9	4	1
Comp. cost	4983	3945	2315	1135	288

Table 2. # of times the objective function f is evaluated.

points in the parameter space that have low cost (i.e. high similarity), we allow the annealing temperature to decrease fast to save computation time but restart the simulated annealing from different initial points uniformly distributed in the parameter space. This forces the restarted ASA to "ergodically" search the entire parameter space. Although the parameter space is not exhaustively searched, most of the "reasonably probable" points are sampled because of the Monte Carlo importance sampling.

During the restarted ASA processes, all the sampled points are stored in a single sampling map. Note that we are considering all samples from all intermediate states of the ASA process, not just the final sample points to which it converges. This is different from mean-shift mode analysis [7], where the mean shift procedure is started from multiple randomly sampled locations and the location/height of each mode and how many points converged to each mode are analyzed. The intermediate points of mean-shift mode seeking are ignored, since they just lie on a gradient path from the start point up to the local mode.

Since we anneal fast, we sacrifice the global convergence of ASA, and false peaks are detected as shown in Figure 5. However, most intermediate sampled points in the parameter space are around the global mode because of importance sampling. The sampling property is quite similar to the particle filter framework [17] where the particles around the modes of an estimated density get more weight, and thus more particles will gather around those modes. When ASA is restarted from many different initial points, more points in the parameter space are evaluated (Table 2) and the cluster of points around the global mode gets bigger. For a 240×360 pixel image, exhaustive search evaluates f 86400 times to find the global mode, while Table 2 shows that ASA is much faster.

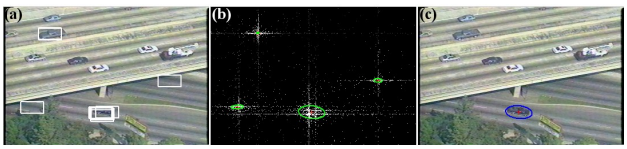

Figure 6. (a) ASA detection in 2D translation space restarted from 9 different initial points; (b) Cluster analysis on the sampling map; (c) The object location is determined by the weighted mean of the most confident cluster.

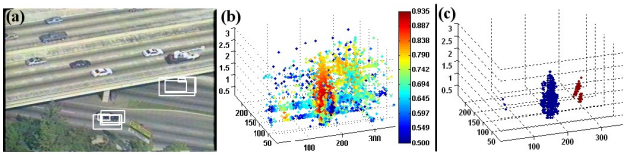

Figure 7. (a) ASA detection in 3D (translation plus scale space) restarted from 9 different initial points; (b) Sampling map in the 3D space; (c) Two clusters are generated by K-means.

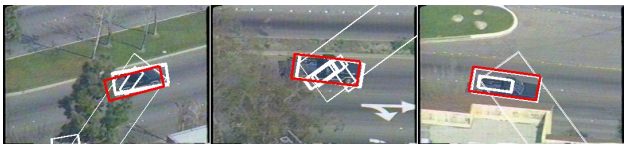

Figure 8. White rectangles represent the ASA detection results in 4D space (translation, rotation and scale) from 9 different initial points. The red rectangle represents the weighted mean of the most confident cluster refined by local mode-seeking.

3.2. Cluster Analysis

Previous efforts on Simulated Annealing consider how to statistically guarantee finding an optimal solution. However, if we relax the interest in global convergence and instead consider how the parameter space is sampled, this sampling map is useful for visual object detection. Instead of choosing the largest similarity value in the sampled space as the final detection result, we perform cluster analysis on the sampled space first. K-means is run on those samples with high similarity scores. During the K-means process, clusters that are close in the parameter space are joined together. The weighted mean of the most confident cluster is calculated as the global mode, which tolerates noise in the sampled parameter space and thus is more robust. The confidence of each cluster is determined by the sum of all its samples' similarity scores. Figure 6 and 7 show the ASA detection results and cluster analysis on a 2D and 3D space.

For clarity, we summarize our entire algorithm in Table 3. When the objective function value by local mode-seeking is lower than some threshold t, global mode-seeking is applied to redetect the object. The threshold for switching between local and global mode-seeking is learned by recording the previous objective function scores, fitting a Gaussian, and detecting a drop if the current score is 3σ away from the mean. The threshold can also be predefined [2]. To accelerate the object detection process, we enforce discrete optimization in Adaptive Simulated Annealing with $\theta \in \{0, \pi/4, \pi/2, 3\pi/4\}$ and $s \in \{0.5, 1, 2\}$. In

Table 3. Hybrid local and global mode-seeking.

Initialization: $k = 0$; Detection = false; select object in frame I_0 by hand and get x_0^*
for each frame in the sequence
  $k = k + 1$;
  if !Detection [Local mode-seeking]
    $x_k^* = \text{argmax} F(x)$ with initial point x_{k-1}^*;
  if $F(x_k^*) < t$ or Detection
    Detection = true;
    $x_k^{'} = ASA(I_k)$; [Global mode-seeking]
    if $F(x_k^{'}) > t$
      Detection = false;
      $x_k^* = \text{argmax} F(x)$ with initial point $x_k^{'}$;

other words, the object shape (polygon) is allowed to rotate around its center every 45 degrees and expand or shrink twice in scale around the initial scale estimate. This allows ASA to run at around 0.23 seconds per detection (C code on a common desktop PC). Starting from the roughly detected global mode, local mode-seeking is performed to get a refined mode estimate. Figure 8 shows three examples of ASA-based global object detection in a 4D space. The global mode is roughly detected by cluster analysis on a 4D ASA sampling map and then refined by local mode-seeking.

4. Experiments

First, we validate our numerical local mode-seeking approach by tracking a face in intensity images [1] using a rectangular box shape. We also demonstrate tracking a complicated polygonal shape by tracking a hand in color images (Figure 9).

The hybrid numerical local and global mode-seeking approach is evaluated on challenging airborne videos with large camera motion and heavy occlusion. In the car chase videos (Figure 11(a-b)), color histograms are used as an object feature and Bhattacharya coefficient is used as the similarity measure. Although the vehicles are often occluded partially or totally and their scale/orientation changes greatly, our tracker successfully tracks the targets and detects them after occlusion. In the VIVID benchmark dataset [11], we chose two challenging sequences on which all previously evaluated trackers only manage to track 17% or less of the way through the whole sequence due to heavy occlusion. Since the object appearances change a lot (Figure 11(c-d)), we chose motion likelihood maps for tracking. Since there are multiple moving vehicles in the scene, if the target is totally occluded while some other vehicle is moving, the original ASA global mode-seeking will detect the other vehicle and restart the local tracker from the wrong location. To avoid such data association problems, we adopt Kalman filter motion prediction to hypothesize an extended trajectory during occlusion (a method also used by other

[1] http://www.cs.toronto.edu/~fleet/research/data.html

Figure 9. Top row: the face is represented by HoG and tracked by the Simplex method using EMD distance. Bottom row: the hand is represented by color histogram and tracked by the Simplex method using Bhattacharya coefficient.

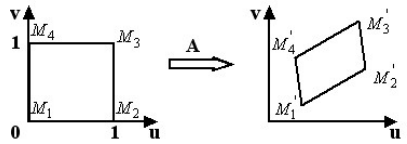

Figure 10. Warping a unit square by an affine transformation.

benchmark dataset competitors). The ASA detection result is accepted only if it is not too far away from the Kalman filter's predicted position estimate. Table 4 shows the comparison of our results with other state-of-the-art trackers on these datasets.

Mean-shift in scale space [9] and our local mode-seeking in scale space can be used to estimate an object's scale between consecutive frames. When the object becomes occluded and reappears at a very different scale (Figure 11(b) frame 505-593), global object detection by restarted ASA needs to search a large scale range (e.g. in Figure 7, the scale range is [0.1, 3]). Finding an object's scale while being blind to rapidly changing camera parameters is challenging, and unnecessary. In fact, if the change is caused by rapid camera zoom, we can roughly estimate the scale change via estimation of the background camera motion. Assuming an affine transformation between two consecutive frames, we stabilize the two frames and get the affine warping matrix $A = [a_{ij}]$. A unit square matrix is warped to the next frame as a parallelogram (Figure 10), and the square root of area change between the two quadrilaterals represents the camera scale change. Since the area of a parallelogram is the magnitude of the cross product of two adjacent edge vectors, we have

$$A(M_1'M_2'M_3'M_4') = |(M_2' - M_1') \times (M_4' - M_1')| \quad (13)$$

where

$$M_1^{'} = \begin{pmatrix} a_{13} \\ a_{23} \\ 1 \end{pmatrix} M_2^{'} = \begin{pmatrix} a_{11} + a_{13} \\ a_{21} + a_{23} \\ 1 \end{pmatrix} M_4^{'} = \begin{pmatrix} a_{12} + a_{13} \\ a_{22} + a_{23} \\ 1 \end{pmatrix}$$

This becomes

$$A(M_1'M_2'M_3'M_4') = \begin{vmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{vmatrix} \quad (14)$$

Chaining consecutive camera scale change estimates over several frames gives us a rough estimate of the object scale after a long occlusion. This approach is used in the sequence of Figure 11(b) to estimate a large object scale change that occurs while the object is occluded.

5. Conclusion

We handle object tracking during normal conditions by numerical local mode-seeking and attack the hard problem of object detection after occlusion via numerical global mode-seeking. Since our numerical optimization methods work on any black-box objective function that accepts a vector of parameters as input and returns a scalar similarity value, we achieve great flexibility to design the objective function without affecting the numerical optimization methods outside the black-box function. We show that the object location, orientation and scale can be found directly via searching a 4D parameter space of translation, rotation and scale by numerical mode-seeking. Different features like histogram of color, HoG or motion, and different similarity measures like Battacharya coefficient or EMD distance, can be used in the black-box objective function. We use piecewise integral image/histogram methods to accelerate the objective function evaluation. All the classic local mode-seeking methods like the Simplex method can be exploited for numerical tracking. We also have introduced a method for global object detection by performing cluster analysis on the sampling map generated by Adaptive Simulated Annealing. Our hybrid numerical local and global mode-seeking approach is validated on airborne videos, and it outperforms state-of-the-art trackers on two challenging sequences from the VIVID benchmark dataset.

Acknowledgments

We thank David Capel for his helpful discussions. This work was funded under the NSF Computer Vision program via grant IIS-0535324 on Persistent Tracking.

References

[1] S. Ali and M. Shah, "COCOA-Tracking in Aerial Imagery," Demo at ICCV 2005.

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

[3] V. Badrinarayanan et al., "Probabilistic Color and Adaptive Multi-feature Tracking with Dynamically Switched Priority between Cues," ICCV 2007

[4] S. Baker and I. Matthews, "Lucas-Kanade 20 Years On: A Unifying Framework," Int. J. of Computer Vision, 56(3):221-255, 2004

[5] A.Blake and M.Isard. Active Contours. Springer1998.

[6] T. Coleman and Y. Li, "An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds," SIAM J. on Opt. 6(2):418-445, 1996.

[7] D. Comaniciu and P. Meer, "Mean shift: A Robust Approach Toward Feature Space Analysis," IEEE Trans. Patt. Anal. and Mach. Intell., 24(5):603-619, 2002.

[8] D. Comaniciu et al., "Kernel-Based Object Tracking," IEEE Trans. Patt. Anal. and Mach. Intell., 25(5):564-577, 2003.

[9] R. Collins, "Mean-shift Blob Tracking through Scale Space," CVPR2003.

[10] R. Collins et al., "On-line Selection of Discriminative Tracking Features," IEEE Trans. Pattern Anal. and Machine Intell., 27(10): 1631-1643, 2005.

[11] R. Collins et al., "An Open Source Tracking Testbed and Evaluation Web Site," IEEE Workshop on Performace Evaluation of Tracking and Surveillance, 2005.

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

[13] Z. Fan, Y. Wu and M. Yang, "Multiple Collaborative Kernel Tracking," CVPR2005.

[14] J.Hafner et al, "Efficient Color Histogram Indexing for Quadratic Form Distance Functions," IEEE Trans. Patt. Anal. and Machi. Intell., 17(7): 729-736, 1995

[15] G. Hager, M. Dewan and C. Stewart, "Multiple Kernel Tracking with SSD," CVPR2004.

[16] L. Ingber, "Simulated Annealing: Practice Versus Theory," J. of Mathematical and Computer Modeling 18(11), 29-57, 1993

[17] M. Isard and A. Blake, "CONDENSATION-conditional density propagation for visual tracking," Int. J. Computer Vision, 29(1): 5-28, 1998

[18] S. Kirkpatrick et al., "Optimization by simulated annealing," Science 220: 671-680, 1983

[19] J. Lagarias et al., "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," SIAM J. of Opt., 9(1): 112-147, 1998.

[20] R. Lienhart et al., "Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection," MRL Tech. Report, Intel Labs, May 2002,

[21] T. Liu and H. Chen, "Real-Time Tracking Using Trust-Region Methods," IEEE Trans. Pattern Anal. and Machine Intell., 26(3): 397-402, 2004.

[22] J. Nocedal and S. J. Wright. Numerical Optimization. Springer, 2006.

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

[24] Y. Rubner, C. Tomasi and L. Guibas, "The Earth Mover's distance as a Metric for Image Retrieval," Int. J. of Computer Vision, 40(2), 99-121, 2000.

[25] C. Shen et al.,, "Fast Global Kernel Density Mode Seeking: Applications to Localization and Tracking," IEEE Trans. on Image Proc., 16(5):1457-1469, 2007.

[26] P. Viola and M. Jones, "Robust Real-Time Face Detection," Int. J. of Comp. Vis., 57(2): 137-154, 2004

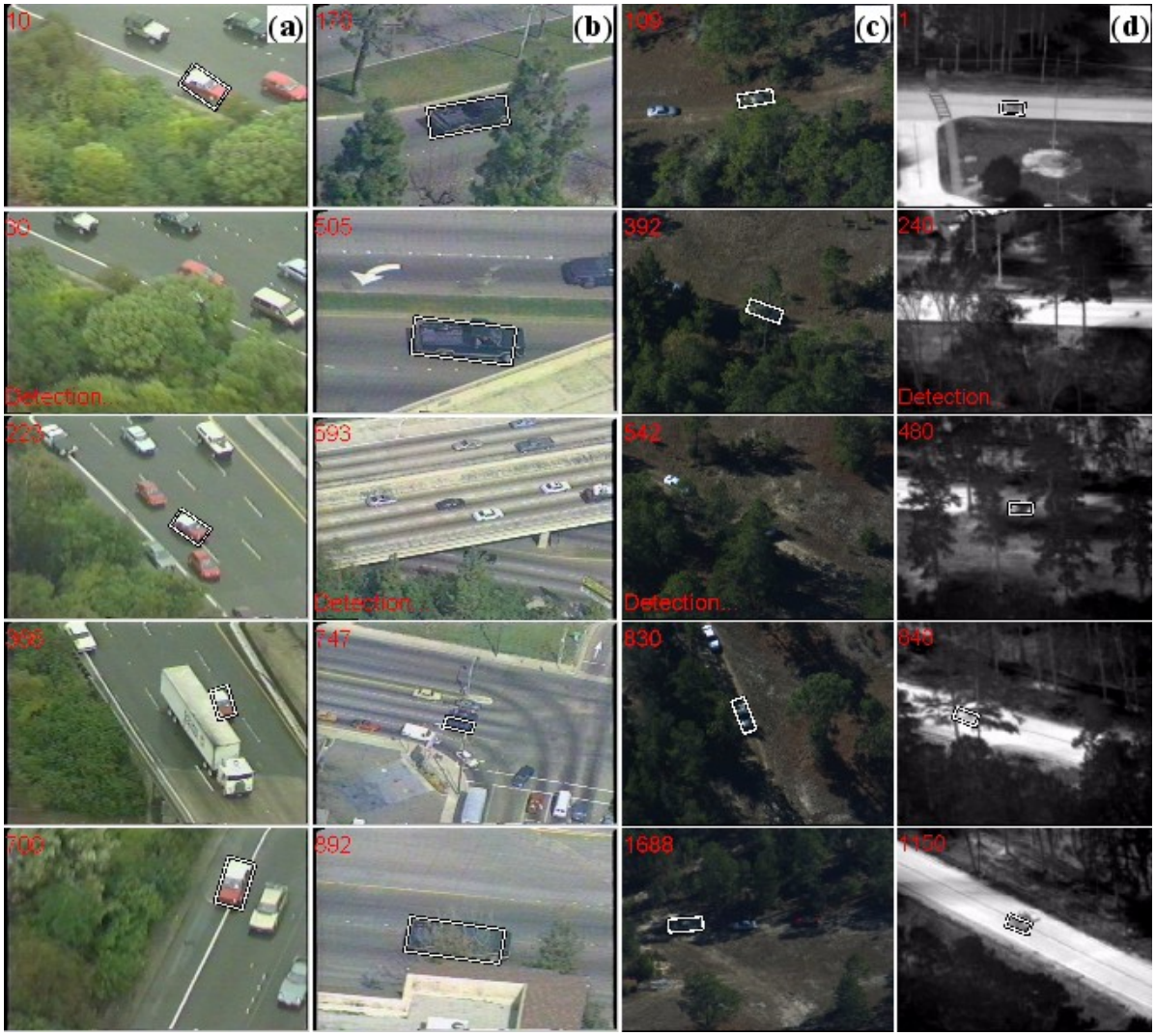

Figure 11. (a)-(b): airborne car chase videos; (c) VIVID benchmark dataset EgTest05; (d) VIVID benchmark dataset PkTest01.

Algorithm	% Dataset Tracked		Avg % overlap BB		Avg % overlap BM		Avg DT(US to GT)		Avg DT(GT to US)	
	PkTest01	EgTest05	PkTest01	EgTest05	PkTest01	EgTest05	PkTest01	EgTest05	PkTest01	EgTest05
Graphcut	14.48	2.27	54.99	40.00	44.54	68.21	inf	9.69	inf	9.94
PMCT [3]	8.97	13.64	84.88	89.33	83.09	70.88	0.7	2.93	0.09	0.3
TemplateMatch	10.34	17.61	91.06	79.82	72.96	59.82	0.77	inf	0.43	inf
VarianceRatio	8.97	13.64	80.95	86.46	74.16	85.12	0.52	1.11	0.72	0.84
FgBgRatio	12.41	13.64	77.75	88.75	66.96	71.12	1.42	0.14	1.07	0.95
Meanshift	8.97	13.64	76.32	94.58	76.58	84.02	1.16	0.49	0.61	0.42
PeakDiff	12.41	13.64	74.79	86.98	62.63	69.90	1.78	0.09	1.86	1.39
Ours	**100**	**100**	95.18	94.55	63.57	54.47	0.61	1.95	0.13	0.53

Table 4. Evaluation on VIVID benchmark datasets. See http://www.vividevaluation.ri.cmu.edu/.

[27] J. Wang, et al., "Image and video segmentation by anisotropic kernel mean-shift," ECCV2004.

[28] C. Yang et al, "Mean-shift Analysis using Quasi-Newton Methods," ICIP2003.

[29] A. Yilmaz, "Object Tracking by Asymmetric Kernel mean-shift with Automatic Scale and Orientation Selection," CVPR2007.

[30] Z. Yin and R. Collins, "Moving Object Localization in Thermal Imagery by Forward-backward MHI," CVPR workshop on OTCBVS 2006.

[31] Z. Zivkovic and B. Krose, "An Em-like Algorithm for Color-histogram-based Object Tracking," CVPR2004.