Differential EMD Tracking

Qi Zhao, Shane Brennan and Hai Tao
Department of Computer Engineering, University of California at Santa Cruz
1156 High Street, Santa Cruz, CA 95064
{zhaoqi,shanerb,tao}@soe.ucsc.edu

Abstract

Illumination changes cause object appearance to change drastically and many existing tracking algorithms lack the capability to handle this problem. The Earth Mover's Distance (EMD) is a similarity measure that is more robust against illumination changes. However, EMD is computationally expensive and we therefore propose the Differential EMD (DEMD) algorithm which computes the derivative of the EMD with respect to the object location so that the EMD does not need to be computed for every location in the tracking window. The fast differential formula is derived based on the sensitivity analysis of the simplex method as applied to the EMD formula. To further reduce the computation, signatures, i.e., variable-size descriptions of distributions, are employed as an object representation. The new algorithm models local background scenes as well as foreground objects to handle scale changes in a principled way. Extensive quantitative evaluation of the proposed algorithm has been carried out using benchmark sequences and the improvement over the standard Mean Shift tracker is demonstrated.

1. Introduction

Illumination changes are a commonly encountered phenomenon in visual tracking. Shading, inter-reflections and other lighting condition changes cause illumination to vary spatially both in intensity and in spectral composition, which may result in drastic changes of object appearance.

Traditional methods to approach this problem include using illumination-invariant features. The contour based methods such as snakes, balloons, and geodesic active contours all belong to this category. However, these methods may fail when the contours are not stable, or there are not many contours, e.g., for small objects. Image photometric, i.e., color, texture, etc., based tracking methods have gained popularity in the last decade. Unfortunately, these methods are sensitive to illumination changes due to their reliance on image photometric variables. One solution is to pre-process the image using some color constancy algorithms [10, 12, 15]. The drawback of such approaches is their degenerated performance under fixed illumination. Freedman and Turek [11] have recently proposed to compute illumination-invariant optical flow fields so as to utilize the photometric information under illumination changes, but the algorithm can be slow, as addressed by the authors.

Instead of using illumination-variant features, or applying transforms to make the features illumination-invariant, this paper approaches illumination changes using the EMD [18] as a similarity measure to match color distributions, which allows the features to vary under illumination changes. The problem with the EMD is its expensive computation, as each calculation requires solving a linear programming problem. This prohibits its application in real-time tracking systems.

The focus of this paper is to derive a gradient descent method to find the object location quickly using the EMD as a similarity measure. Since the objective of tracking using EMD is represented in the form of linear programming, direct differential methods [9, 13] cannot be applied. In this work, we conduct a two-phase analysis: firstly, we perform sensitivity analysis of the simplex method, i.e., an efficient algorithm to solve the EMD, to obtain the derivatives of the EMD with respect to the object colors. Secondly, in order to derive the derivatives of the object colors with respect to the location, we represent the statistical color feature in a kernel framework. By convolving the feature with an isotropic kernel, these derivatives can be calculated directly. Having the results of the two-phase analysis, the differential formula of the EMD with respect to the location is obtained.

To further reduce the computation, signatures [18] are employed as object representations. Unlike histograms, the structures of signatures are adjustable, in a sense that will be made precise in section 3. The use of color signatures significantly reduces the size of the EMD problem and consequently requires much less computation as the EMD has an exponential worst case running time.

Many existing tracking algorithms which consider foreground objects alone fail to estimate the object scale when

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

the object has a similar feature value for the entire object as well as parts of the object [7]. To cope with this problem, the proposed algorithm models both the objects and local background scenes, and the matching step considers the similarity measures for both of them. Discriminative tracking methods [3, 4, 8] have also utilized background information, but in a different manner than the generative way used in our approach. Those methods focused on discriminating the foreground objects from the background scenes and the scale adaptation problem was not explicitly handled.

The rest of the paper is organized as follows: section 2 describes related work. Section 3 discusses the details of the DEMD tracking algorithm. Section 4 proposes the DEMD tracking with background modeling to handle scale changes. Section 5 demonstrates promising comparative and quantitative results and section 6 concludes the paper.

2. Related Work

There are three areas of computer vision that bear on the work presented in this paper: object representations, similarity measures, and optimization techniques for kernel based tracking. We briefly review the most relevant literature in each case.

The literature on object representations is vast. In this paper, we use color distributions as our representation due to their simplicity, efficiency and robustness to rotation, scaling and partial occlusions. The early work of Swain and Ballard [19] employed color histogram as a global visual feature, demonstrating that color can be exploited as a useful feature for rapid detection. Later, methods such as the Mean Shift [9] and the CAMShift [6] algorithms were proposed using this representation for visual tracking.

Similarity measures between histograms broadly fall into two categories: the bin-by-bin similarity measures that only compare contents of corresponding histogram bins; and the cross-bin similarity measures that also compare non-corresponding bins. In practice, most existing histogram based tracking algorithms use bin-by-bin similarity measures such as the Bhattacharyya coefficient based distance [14] and the SSD [13]. These approaches tend to break down under color variations as no ground distances with different bins are used and thereby a small amount of deviation is treated the same way as a large difference as long as the color falls into a different bin. The cross-bin similarity measures are rarely employed for tracking largely due to their computational complexity. Ling and Okada [16] proposed to reduce the computation of EMD using EMD-L1 and applied it to image feature matching, but the method is limited by using the L1 distance as a ground distance.

Real-time tracking imposes rigorous requirements on the algorithm speed. Instead of a brute-force search, kernel based objective functions allow the use of optimization techniques to find the optimal object state quickly. Some commonly used approaches include the Mean Shift algorithm [9] and the Newton style minimization procedure [13]. Generally, the use of these techniques require the objective function to be written in an closed form.

3. Differential EMD (DEMD) Tracking

3.1. The EMD as A Similarity Measure

The EMD [18] gains its name from the intuition that given two distributions, one can be seen as a mass of earth properly spread in space, the other as a collection of holes in that same space. The EMD measures the least amount of work needed to fill all of the holes with all of the earth, where a unit of work corresponds to transporting a unit of earth by a unit of ground distance.

In this paper, the EMD is employed to compare the color distributions of the object model and that of the object candidate. The distributions are represented in the form of signatures. Formally, a signature represents a set of feature clusters and is defined as

$$\mathbf{s} = \{s_u\}_{u=1,..,m}, \ \ s_u = (a_u, w_u), \tag{1}$$

where m is the number of clusters in the signature, a_u is the mean of the u-th cluster and w_u the weight of the cluster.

Representing the model distribution as model signature, and the candidate distribution as candidate signature, we denote the ground distance between the u-th cluster in the model signature and the v-th cluster in the candidate signature as d_{uv}, and the flow (amount of transported earth) between them as $f_{uv}(y)$. The goal is to find the location y that corresponds to the smallest EMD

$$\arg\min_{y}(\min_{f_{uv}} Z(f_{uv}(y))). \tag{2}$$

In Eq.2 the inner optimization is to find the EMD for each location, and the outer one is to obtain the best object location. According to the definition of EMD [18], Z in Eq.2 is formulated as

$$Z(f_{uv}(y)) = \sum_{u=1}^{m^M} \sum_{v=1}^{m^C} d_{uv} f_{uv}(y),$$

subject to

$$\sum_{u=1}^{m^M} f_{uv}(y) = w_v^C(y), \quad 1 \le v \le m^C$$

$$\sum_{v=1}^{m^C} f_{uv}(y) = w_u^M(y), \quad 1 \le u \le m^M$$

$$\sum_{u=1}^{m^M} \sum_{v=1}^{m^C} f_{uv}(y) = 1$$

$$f_{uv}(y) \ge 0, \qquad 1 \le u \le m^M, 1 \le v \le m^C.$$

In these equations, the superscript M denotes the object *model* and C is for the object *candidate*. w_v^C is the weight of the v-th cluster in the candidate signature and w_u^M the weight of the u-th cluster in the model signature. m^C and m^M are the numbers of clusters in the candidate and model signatures, respectively.

3.2. Overview of the DEMD Algorithm

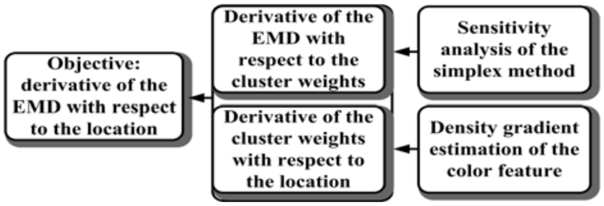

Objective: derivative of the EMD with respect to the location
Derivative of the EMD with respect to the cluster weights
Derivative of the cluster weights with respect to the location
Sensitivity analysis of the simplex method
Density gradient estimation of the color feature

Figure 1. Diagram of the DEMD algorithm.

The main theoretical contribution of the paper is the derivation of a differential formula to compute the derivative of the EMD with respect to the location so as to locate the object quickly. Since the formulation of the EMD is a linear programming problem, derivative of the EMD can not be directly computed. To overcome this difficulty, we propose a two-phase algorithm, as depicted in Figure 1.

Specifically, we formulate the gradient descent representation of the EMD with respect to the location as $\nabla_y Z(y)$, which can be expressed using the change of the EMD with respect to each cluster weight ($\frac{\partial Z(y)}{\partial w_v^C(y)}$) and the derivative of the cluster weight with respect to the location ($\nabla_y w_v^C(y)$) as

$$\nabla_y Z(y) = \sum\nolimits_{v=1}^{m^C} \frac{\partial Z(y)}{\partial w_v^C(y)} \nabla_y w_v^C(y), \tag{3}$$

where $w_v^C(y)$ is the weight of the v-th cluster in the candidate signature and m^C the number of clusters in the candidate signature.

In the following two subsections, we first calculate $\frac{\partial Z(y)}{\partial w_v^C(y)}$ through a sensitivity analysis of the simplex method, followed by a density gradient estimation of the color feature to obtain $\nabla_y w_v^C(y)$.

3.3. Simplex Method and Sensitivity Analysis

3.3.1 Simplex Method in Matrix Form

To perform the sensitivity analysis the problem in Eq.2 is first represented in a matrix form. The starting matrix is then transformed to an optimal form where the change of EMD with respect to the changes of the cluster weights are expressed in an explicit way.

Specifically, since there are $m^M \times m^C$ variables $f_{uv}(y)$ and $m^M \times m^C$ constants d_{uv} in Eq.2, we use column vectors $\mathbf{f}(y)$ and $\mathbf{d}$, both of size $m^M \times m^C$, to represent the flow and the ground distance. Stacking the first three equations of the constraints in Eq.2, the coefficients, which are either 1 or 0, can form a 2-dimensional matrix of $m^M + m^C + 1$ rows and $m^M \times m^C$ columns. Denoting this coefficient matrix as H and representing $[(\mathbf{w^C}(y))^T\ (\mathbf{w^M})^T\ 1]^T$ as $\mathbf{b}(y)$, we have the matrix form of Eq.2 as

$$\arg\min_y (\min_{\mathbf{f}} Z = \mathbf{d}^T \mathbf{f}(y)), \tag{4}$$

subject to

$$H\mathbf{f}(y) = \mathbf{b}(y)$$
$$\mathbf{f}(y) \geq \mathbf{0}.$$

1. To perform matrix transformations, the matrix is reformulated. Since there are $m^M \times m^C$ variables and $m^M + m^C + 1$ constraints in the problem, there are $m^M + m^C + 1$ basic variables, i.e., variables of nonzero value, and $m^M \times m^C - (m^M + m^C + 1)$ non-basic variables. Grouping all the basic variables together and all the non-basic variables together we split the flow vector $\mathbf{f}$ into $[\mathbf{f}_B^T\ \mathbf{f}_{NB}^T]^T$ where the subscript B denotes basic variables and NB is for non-basic variables. The ground distance vector $\mathbf{d}$ is similarly divided as $[\mathbf{d}_B^T\ \mathbf{d}_{NB}^T]^T$, and $H = [H_B\ H_{NB}]$. Thus the starting tableau for the simplex method is written as

Z	$\mathbf{f}_B$	$\mathbf{f}_{NB}$	RHS
1	$-\mathbf{d}_B^T$	$-\mathbf{d}_{NB}^T$	0
$\mathbf{0}$	H_B	H_{NB}	$\mathbf{b}$

Table 1. Starting Tableau

In this table, RHS denotes the right hand side of the equations. The second row corresponds to the objective function of Eq.4, and the third row is a vector representation of all the constraints in Eq.4.

2. Apply matrix transformations (details omitted due to space limitations), the optimal tableau is

Z	$\mathbf{f}_B$	$\mathbf{f}_{NB}$	RHS
1	0	$-\mathbf{d}_{NB}^T + \mathbf{d}_B^T H_B^{-1} H_{NB}$	$\mathbf{d}_B^T H_B^{-1}\mathbf{b}$
$\mathbf{0}$	I	$H_B^{-1} H_{NB}$	$H_B^{-1}\mathbf{b}$

Table 2. Optimal Tableau

3.3.2 Sensitivity Analysis

Based on the optimal tableau we analyze the sensitivity of the EMD to a change in the cluster weights of the color signature. Note that sensitivity analysis can only be performed on the $\mathbf{w^C}(y)$ part, i.e., the cluster weights corresponding to the object candidate.

From the second row of Table 2, we have $Z = \mathbf{d}_B^T H_B^{-1}\mathbf{b}$. Assume $\mathbf{b}$ is changed to $\mathbf{b}^{'}$, where in $\mathbf{b}^{'}$, $b_i^{'} = b_i + \Delta b_i$, ($1 \leq i \leq m^C$), i.e., the weight of the ith cluster changes, and b_j ($j \neq i$) remain the same. The optimal solution becomes

$$Z^{'} = \mathbf{d}_B^T H_B^{-1}\mathbf{b}^{'} = \mathbf{d}_B^T H_B^{-1}\mathbf{b} + \mathbf{d}_B^T H_B^{-1}[0..0\ \Delta b_i\ 0..0]^T$$

$$= \mathbf{d}_B^T H_B^{-1}\mathbf{b} + k_i \Delta b_i,$$

where $k_i = \sum_{l=1}^{m^M+m^C} (\mathbf{d}_B)_l (H_B^{-1})_{li}$.

Therefore,

$$\frac{\partial Z}{\partial b_i} = lim_{\Delta b_i \to 0}\frac{\Delta Z}{\Delta b_i} = \frac{k_i \Delta b_i}{\Delta b_i} = k_i. \qquad (5)$$

As the sum of the cluster weights of the candidate signature is 1, the change of color in one cluster causes a change to the value of the other clusters due to a normalization procedure. Considering this constraint leads to

$$\frac{\partial Z}{\partial b_i} = k_i - \sum\nolimits_{j\neq i} k_j \frac{b_j}{\sum_{j\neq i} b_j}, \quad i = 1, ..., m^C. \qquad (6)$$

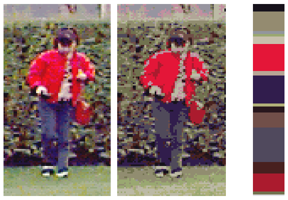
The proof is given in the Appendix A. The intuition of this equation is the projection of the k_i (Eq.5) from a m^C dimensional space to a $m^C - 1$ dimensional space, where the "-1" is imposed by the constraint of $\sum_{i=1}^{m^C} b_i = 1$.

Eq.6 provides an explicit formula of how the EMD would change with respect to the color changes.

3.4. Density Gradient Estimation of the Color Feature

3.4.1 Representing Objects using Color Signatures

In this paper we use color signatures instead of color histograms to represent the objects due to their compactness. Figure 2 illustrates an example of using a 16-cluster signature to represent the image. Though the cluster number is small, the color of the image is well preserved.



(a) (b) (c)

Figure 2. An example of a color signature. (a) Original image (from the *MIT Pedestrian Dataset* [17]). (b) Rendered image using a 16-cluster signature. (c) Color signature.

Without loss of generality, the object model is considered as centered at the spatial location 0 and a kernel based representation [9] is defined according to Eq.1 as $\mathbf{s}^M = \{s_u^M\}, u = 1, .., m^M$, where $s_u^M = (a_u^M, w_u^M)$ and

$$w_u^M = \beta \sum\nolimits_{n=1}^{N} K(\|\frac{x_n}{h}\|^2)\delta[c(x_n) - u]. \qquad (7)$$

In this equation the density in the feature space is clustered into m^M clusters. x_n denotes 2D image coordinates, and the number of pixels is N. c is a function which associates the pixel at location x to the cluster which is the nearest to the color of that pixel. $K(x)$ is an isotropic kernel which assigns a smaller weight to the locations that are farther from the center of the object. The summations are performed over a local window around the object center, with h being the window radius. δ is the Kronecker delta function and β is the normalization factor.

Similarly, the object candidate is defined at location y as $\mathbf{s}^C(y) = \{s_v^C(y)\}, v = 1, .., m^C$, where $s_v^C(y) = (a_v^C(y), w_v^C(y))$ and

$$w_v^C(y) = \gamma \sum\nolimits_{n=1}^{N} K(\|\frac{x_n - y}{h}\|^2)\delta[c(x_n) - v]. \qquad (8)$$

Here, the feature space has m^C clusters and γ normalized the feature.

3.4.2 Estimation of the Density Gradient

Take the gradient of the cluster weights (Eq.8), we have the density gradient of the color feature as

$$\nabla_y w_v^c(y) = \frac{2\gamma}{h^2} \sum\nolimits_{n=1}^{N} (x_n - y) g(\|\frac{y - x_n}{h}\|^2)\delta[c(x_n) - v]. \qquad (9)$$

In this formula, $g(x) = -k^{'}(x)$, where k is the profile of kernel K and is defined as $k : [0, +\infty) \rightarrow R$ such that $k(\|x\|^2) = K(x)$.

3.5. Closed-Form DEMD Tracking

Recall from Eq.3 that the gradient descent representation of the EMD is $\nabla_y Z(y) = \sum_{v=1}^{m^C} \frac{\partial Z(y)}{\partial w_v^C(y)} \nabla_y w_v^C(y)$.

Substituting the RHS of Eq.6 for $\frac{\partial Z(y)}{\partial w_v^C(y)}$ and the RHS of Eq.9 for $\nabla_y w_v^C(y)$ yields

$$\nabla_y Z(y) = \frac{2\gamma}{h^2} \sum\nolimits_{n=1}^{N} (x_n - y) g(\|\frac{y - x_n}{h}\|^2)\pi_n. \qquad (10)$$

In Eq.10, the weight of each pixel x_n is

$$\pi_n = \sum\nolimits_{v=1}^{m^C} (k_v - \sum\nolimits_{j\neq v} k_j \frac{b_j}{\sum_{j\neq v} b_j})\delta[c(x_n) - v] \qquad (11)$$

where $k_v = \sum_{l=1}^{m^M+m^C} (\mathbf{d}_B)_l (H_B^{-1})_{lv}$.

Thus, the distance minimization can be efficiently achieved based on Eq.10, using the following algorithm:

Algorithm 1 Fast *Differential EMD (DEMD)* Procedure

```
Input: Object center of the previous frame $y_0 = y^{i-1}$
Output: Initialized object center for the current frame $y_0^i$

• Initialize the location of the object in the current frame
  with $y_0$. Evaluate $EMD(y_0)$ using Eq.2.
```

- Compute the weights $\{\pi_n\}_{n=1,..,N}$ for the pixels in the tracking window according to Eq.11.
- Compute the gradient $\nabla_x Z(y_0)$ based on Eq.10.
- Move the object along the gradient vector to one of its 8 neighboring pixels y_1. Evaluate $EMD(y_1)$ using Eq.2.
- If $EMD(y_1) > EMD(y_0)$, set $y_0^i \leftarrow y_0$ and stop; otherwise, set $y_0 \leftarrow y_1$ and go to the 1^{st} step.

4. DEMD Tracking with Background Modeling

4.1. DEMD Tracking with Background Modeling

The DEMD algorithm provides accurate tracking results under most scenarios. However, the method may be insufficient in cases of scale changes, background clutter, etc. To determine the object scale and position in a principled way, we model local background scenes as well as foreground objects and consider the similarities of both components to determine the object state. Using the notations in section 3, the goal is to find the object position y and scale σ corresponding to the smallest sum of the EMD for the foreground object and the EMD for the local background scene

$$\arg\min_{y,\sigma}(\min_{f_{uv}} Z(f_{uv}(y,\sigma)) + \min_{f_{uv}^{Bg}} Z^{Bg}(f_{uv}^{Bg}(y,\sigma))), \quad (12)$$

where the superscript Bg denotes the local background scenes. The formulations for Z are addressed in Eq.2 and Z^{BG} is formulated in the same way. The linear combination is found to be simple and effective in balancing the influence of the foreground objects and background scenes.

To achieve real-time performance the initial object location for the current frame is obtained by the fast DEMD algorithm, as discussed in section 3. This offers a good initialization for the following steps where the scale and position of the object are adjusted iteratively according to Eq.12. Figure 3 illustrates the method and the detailed algorithm for the adjustment step is given in Appendix B.

4.2. Background Alignment

When the background is static, features on the same background region, i.e., a rectangle with two overlapping holes, in two consecutive frames are compared to estimate the background similarity. With a dynamic background it is not reasonable to compare the same regions in consecutive frames. This paper allows for dynamic backgrounds by aligning the background in consecutive frames by solving the optical flow equations using the direct method [5]. In this way, background similarity can be obtained by comparing the background candidate region in the aligned image with the background model in the previous frame.

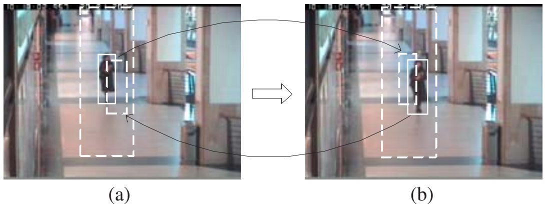


Figure 3. DEMD tracking with background modeling (a) The $(t-1)^{th}$ frame (b) The t^{th} frame. Pixels within the solid-line rectangle belong to the object, pixels outside the solid-line rectangle and within the larger dashed-line rectangle belong to the local background. For the ideal object scale and position in the t^{th} frame, the object should conform to its model; besides, the local background region, i.e., the area outside the two foreground regions and within the background region of the t^{th} frame, should match the same area of the previous frame.

5. Experimental Results

Extensive and comparative experiments are carried out and reported in this section. We first show examples of the DEMD tracking on foreground objects only and then the DEMD tracking with background modeling, followed by quantitative results. In all these experiments a simple "*divide and recombine*" strategy [18] is applied to compute 16-cluster color signatures of the image regions, and the Euclidean distance in RGB color space is used as the ground distance.

5.1. Examples of the DEMD Tracking

In the first experiment we compare the Bhattacharyya coefficient based distance with the EMD under color variations. Figure 4 shows the results of the standard Mean Shift (MS) tracker which employs the Bhattacharyya coefficient based distance, and the proposed DEMD tracker using EMD of color signatures on an indoor *Pedestrian* sequence. The color of the pedestrian is changing due to reflections. The figures beside the actual frames show the values of the two distances. Figures 4(j) and 4(l) illustrate that the minimum in the error surface is very close to the actual object location, which indicates that the color variation does not cause the EMD to change significantly. However, the color change makes a difference for the Bhattacharyya coefficient, causing the expected minimal distance to be large, as shown in Figures 4(d) and 4(f), thereby the tracker starts to drift away from the pedestrian. The MS tracker loses the object quickly while the DEMD tracker manages to track the pedestrian throughout the entire sequence.

The average number of iterations for the DEMD tracker on the *Pedestrian* sequence is 3.01 iterations per frame. The differential method requires only a small number of iterations to find the location of the object which is critical for a real-time tracking system.

We then perform comparative experiments of the DEMD

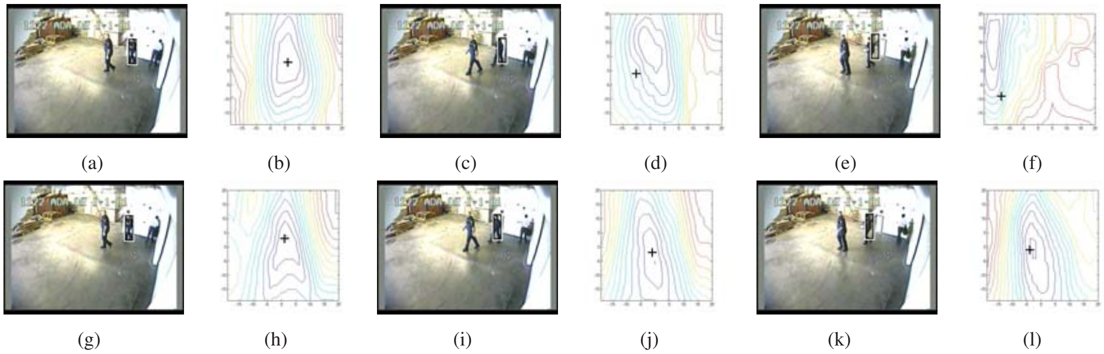


Figure 4. Comparison between the Bhattacharyya coefficient based distance and the EMD. Frames 1, 11, 21 from the *Pedestrian* sequence are shown. (a)(c)(e) Tracking results of the MS tracker. (b)(d)(f) Bhattacharyya coefficient based distances for a 40×40 region. "+" indicates the ground truth object location. (g)(i)(k) Tracking results of the DEMD tracker. (h)(j)(l) EMDs.

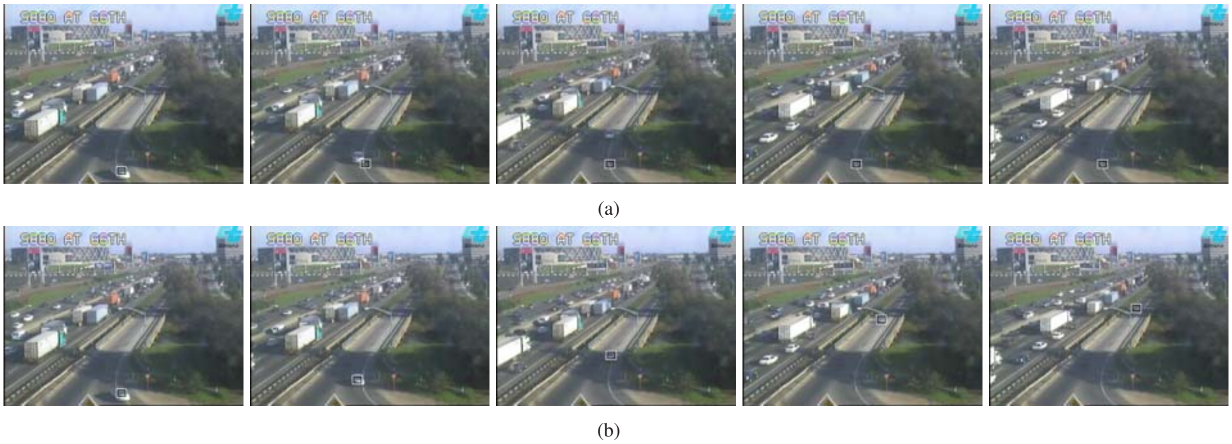


Figure 5. Frames 1, 13, 35, 86, 110 from the *Highway-Car* sequence. (a) The MS tracker starts to wander when the car is entering the strong shadow and fails around frame 13. (b) The DEMD tracker successfully follows the car into and out of strong shadows.

tracker and the MS tracker on two outdoor sequences, where the moving objects undergo severe appearance changes due to the sunshine and the strong shadows. Figure 5 and Figure 6 illustrate the tracking results.

The fourth experiment is performed on the OTCBVS benchmark data [1]. The tracking results using the DEMD tracker with 16-cluster color signatures are presented in Figure 7(a). It can be observed that the tracker provides quite accurate location of objects. However, due to the lack of any scale adaptation mechanism, the performance degenerates in cases of large scale changes.

5.2. Examples of the DEMD Tracking with Background Modeling (DEMDB)

As shown in Figure 7(b), for the same OTCBVS sequence, the DEMDB tracker keeps tight track of the object thereby it is more robust against background distraction.

In the fifth experiment we track a vehicle in the *RedTeam* sequence from the PETS'05 dataset, where the background is dynamic. From the results shown in Figure 8, we see that the DEMDB tracker deals with scale changes reliably.

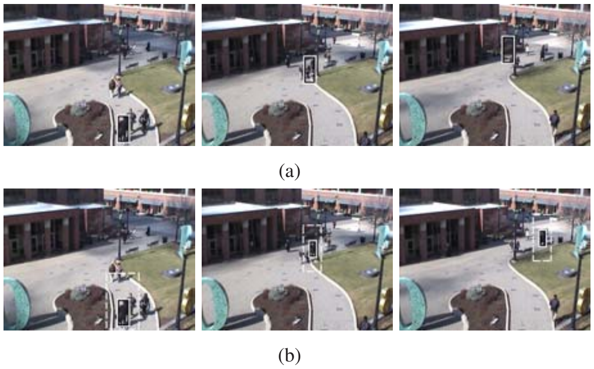
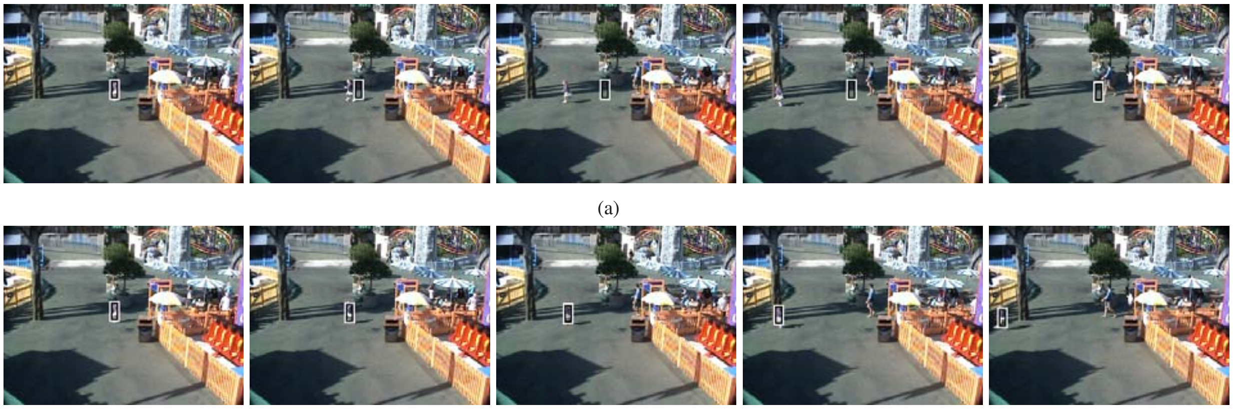


Figure 7. Frames 1, 354, 486 from the *OTCBVS* sequence. (a) The performance of the DEMD tracker degenerates as the object becomes smaller and eventually loses track of it. (b) The DEMDB tracker keeps tight track of the object.



(a)

(b)

Figure 6. Frames 1, 10, 30, 51, 65 from the *Running-Girl* sequence. (a) The MS tracker has lost track of the girl by frame 10. (b) The DEMD tracker maintains a secure focus on the object throughout the sequence though there are shadows all along the path and the object is moving fast.

Source Dataset	Description / File Name	Object Type	Frames Tracked		Position Error		Size Error	
			MS	DEMDB	MS	DEMDB	MS	DEMDB
PETS'01	Red-Coat Female - Cam2	Person	577/651	651/651	0.284	**0.133**	0.264	**0.136**
PETS'01	White Van - Cam1	Vehicle	135/260	260/260	0.229	**0.090**	0.312	**0.152**
PETS'04	Female - Front View	Person	81/162	162/162	0.260	**0.156**	0.135	**0.120**
PETS'04	Female - Corridor View	Person	381/381	381/381	0.047	**0.040**	**0.057**	0.072
PETS'04	Male - Corridor View	Person	550/550	550/550	**0.097**	0.102	0.133	**0.122**
PETS'05*	RedTeam	Vehicle	1918/1918	1918/1918	0.170	**0.056**	0.282	**0.108**

Table 3. Quantitative results for several public data of the proposed DEMD tracker with background modeling (DEMDB) and its comparison with the standard Mean Shift (MS) Tracker. In datasets with $*$, ground truth was provided every 10 frames and we count only those frames with ground truth for comparison. Others have ground truth for each frame.

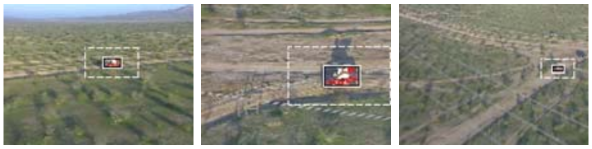


Figure 8. DEMDB tracking with a moving camera. Frames 60, 1650, 1895 from the *RedTeam* sequence are shown.

5.3. Quantitative Results

We have conducted a quantitative evaluation of the DEMD algorithm with background modeling. We carry out comparisons with the MS tracking method, where the conventional scheme for scale adaptation, i.e., varying the object size by $+/-10\%$ and choosing the one with smallest distance [9], is implemented. We use 6 sequences taken from the public PETS'01, PETS'04 and PETS'05 datasets [2], where ground truth data are available. Quantitative results are shown in Table 3. All the objects are initialized using ground truth data. Tracking is deemed to fail if the tracker-identified bounding box has no overlap with the ground truth bounding box. The object centroid position error is calculated as the Euclidian distance between the centroids of the bounding boxes of the ground truth and the tracking results on frames of successful tracking. To prevent errors in frames with larger object scales from dominating the averaged error, the centroid error is normalized with respect to the ground truth length of the object's diagonal. Similarly, the size error is defined as the Euclidian distance between the two (height, width) vectors, normalized by the ground truth length of the object's diagonal.

In Table 3, the MS tracker tracks throughout three of the sequences, while the DEMDB tracker succeeds in tracking throughout all the six sequences. Additionally, the DEMDB tracker outperforms the MS tracker in terms of accuracy. This is due to the robustness of the DEMDB against illumination changes and the tracker's capability in accurately estimating the object scale even when the objects are of mostly uniform-color, where algorithms considering only the matching score of the foreground objects have no "force" to keep the window expanded as the object becomes larger [7].

6. Conclusions

Illumination changes make image photometric based trackers unreliable. This paper employs the EMD as a similarity measure to approach this problem. To the best of our knowledge, this is the first work using the EMD and signatures in visual tracking. The main theoretical contribution in this work is the development of a fast differential algorithm based on the sensitivity analysis of the simplex method. The gradient descent technique and the use of signature significantly reduce the computation of the EMD based tracker and make real-time processing of video streams possible - the tracker runs comfortably at 30 fps on a PIV 3.20GHz PC. Experiments demonstrate the advantage of the EMD over other commonly used metrics under varying illuminations, and the importance of knowing local background scenes in estimating the object scales.

7. Appendix

A. Proof of Eq.6 Due to the constraint that $\sum_{i=1}^{m^C} b_i = 1$, the increase/decrease of b_i would decrease/increase b_j ($j \neq i$) after normalization. Therefore, the partial derivative of Z with respect to b_i is written as

$$\frac{\partial Z}{\partial b_i} = lim_{\Delta b_i^* \to 0} \frac{k_i \Delta b_i^* + \sum_{j \neq i} k_j \Delta b_j}{\Delta b_i^*}, \;\; i = 1, ..., m^C, \tag{13}$$

where Δb_i^* is the change of b_i after normalization, and Δb_j is the change of b_j. $\partial Z / \partial b_i$ can be solved considering the following two conditions:

Condition 1: $\Delta b_j / b_j = Const.$ for all $j \neq i$.

This is justified by the fact that the b_j are unchanged without the normalization procedure, therefore they simply scale down/up to satisfy the constraint.

Condition 2: $\Delta b_i^* + \sum_{j \neq i} \Delta b_j = 0$.

From the two conditions, we obtain $\Delta b_j = -\frac{b_j}{\sum_{j \neq i} b_j} \Delta b_i^*$. Substituting this into Eq.13 results

$$\frac{\partial Z}{\partial b_i} = k_i - \sum_{j \neq i} k_j \frac{b_j}{\sum_{j \neq i} b_j}, \;\; i = 1, ..., m^C. \tag{14}$$

B. Algorithm to Adjust the Object Scale and Position

Algorithm 2 Algorithm to Adjust Object Scale and Position with Both Foreground and Background Cues

Input: Object center $y_0 = y_0^i$ returned by Algorithm 1
Object scale from the previous frame $\sigma_0 = \sigma^{i-1}$.
Output: Object center y^i and scale σ^i of the current frame

- Initialize the object location with y_0, vary σ_0 by $+/-$ 10% and evaluate which scale is the best using Eq.12.
- If the scale with the smallest distance σ_1 equals σ_0, set $\sigma^i \leftarrow \sigma_0, y^i \leftarrow y_0$ and stop; otherwise, set $\sigma_0 \leftarrow \sigma_1$, and run a numerical gradient algorithm to obtain the new location y_1.
- If y_1 equals y_0, set $\sigma^i \leftarrow \sigma_0, y^i \leftarrow y_0$ and stop; otherwise, set $y_0 \leftarrow y_1$ and go to the 1^{st} step.

Acknowledgement

This work was supported in part by National Science Foundation (NSF) Grant IIS-0348020.

References

[1] http://www.cse.ohio-state.edu/otcbvs-bench/.

[2] http://www.cvg.rdg.ac.uk/slides/pets.html.

[3] S. Avidan. Support vector tracking. *PAMI*, 26(8):1064–1072, August 2004.

[4] S. Avidan. Ensemble tracking. *PAMI*, 29(2):261–271, February 2007.

[5] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV*, pages 237–252, 1992.

[6] G. Bradski. Computer vision face tracking for use in a perceptual user interface. In *WACV*, pages 214–219, 1998.

[7] R. Collins. Mean-shift blob tracking through scale space. In *CVPR*, pages II: 234–240, 2003.

[8] R. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *PAMI*, 27(10):1631–1643, October 2005.

[9] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *PAMI*, 25(5):564–577, May 2003.

[10] D. Forsyth. A novel approach to color constancy. *IJCV*, 5(1):5–36, August 1990.

[11] D. Freedman and M. Turek. Illumination-invariant tracking via graph cuts. In *CVPR*, pages II: 10–17, 2005.

[12] B. Funt and G. Finlayson. Color constant color indexing. *PAMI*, 17(5):522–529, May 1995.

[13] G. Hager, M. Dewan, and C. Stewart. Multiple kernel tracking with SSD. In *CVPR*, pages I: 790–797, 2004.

[14] T. Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE Trans. Commun. Tech.*, 15(1):52–60, February 1967.

[15] E. Land and J. McCann. Lightness and retinex theory. *J. Opt. Soc. Am.*, 61(1):1–11, 1971.

[16] H. Ling and K. Okada. Emd-L1: An efficient and robust algorithm for comparing histogram-based descriptors. In *ECCV*, pages III: 330–343, 2006.

[17] C. Papageorgiou and T. Poggio. Trainable pedestrian detection. In *ICIP*, pages IV:35–39, 1999.

[18] Y. Rubner. Perceptual metrics for image database navigation. In *Ph.D. dissertation, Stanford University*, 1999.

[19] M. Swain and D. Ballard. Color indexing. *IJCV*, 7(1):11–32, 1991.