Differential Earth Mover's Distance with Its Applications to Visual Tracking

Qi Zhao, *Student Member*, *IEEE*, Zhi Yang, *Student Member*, *IEEE*, and Hai Tao, *Senior Member*, *IEEE*

Abstract—The Earth Mover's Distance (EMD) is a similarity measure that captures perceptual difference between two distributions. Its computational complexity, however, prevents a direct use in many applications. This paper proposes a novel Differential EMD (DEMD) algorithm based on the sensitivity analysis of the simplex method and offers a speedup at orders of magnitude compared with its brute-force counterparts. The DEMD algorithm is discussed and empirically verified in the visual tracking context. The deformations of the distributions for objects at different time instances are accommodated well by the EMD, and the differential algorithm makes the use of EMD in real-time tracking possible. To further reduce the computation, signatures, i.e., variable-size descriptions of distributions, are employed as an object representation. The new algorithm models and estimates 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.

Index Terms—Earth mover's distance (EMD), gradient descent, real-time tracking.

1 Introduction

The EMD has proven to be perceptually consistent with human vision for comparing distributions. By taking into account the ground distances between noncorresponding bins and computing the similarity by solving a linear programming problem, the EMD provides a meaningful way of measuring distribution distances. While working effectively, the problem with the EMD is its expensive computation, which prohibits its applications in many vision problems such as real-time tracking, fast image retrieval [1], and contour matching [2]. This paper presents a general framework of differential EMD and focuses on the discussions on its application in visual tracking.

In visual tracking, image photometric, i.e., color, texture, etc., based statistic representations have gained popularity in the last decade. However, these methods are sensitive to appearance changes due to their reliance on image photometric variables, and the distributions could deform in many cases. For example, illumination changes are a commonly encountered phenomenon that can be caused by shading, interreflections, and other lighting condition changes, and may result in drastic changes of object appearance. One solution is to preprocess the image using some color constancy algorithms [3], [4], [5]. The drawback of such approaches is their degenerated performance under fixed illumination. Freedman and Turek [6] have recently proposed to compute illumination-invariant optical flow fields to utilize the photometric information under illumination changes, but the algorithm can be slow, as addressed by the authors. Further, these methods do not solve the general distribution deformation problem that may arise from many sorts of appearance changes including moderate pose changes and partial occlusions. In these settings, the use of EMD as a similarity measure to match color distributions provides a principled way of accommodating the feature deformations, thus allowing certain appearance variations.

The focus of this paper is to derive a gradient descent method using the EMD as a similarity measure. Since the objective function using EMD is normally represented in the form of linear programming, direct differential methods [7], [8] cannot be applied. In this work, we propose to approach this problem by decomposing the original problem into several subproblems using the chain rule and applying the sensitivity analysis of the simplex method, i.e., an efficient numerical algorithm to solve the EMD. Specifically, in the visual tracking context, the objective is to derive the derivative of EMD with respect to the location so as to locate objects fast. To achieve this goal, a two-phase analysis is conducted: First, we perform sensitivity analysis of the simplex method to obtain the derivatives of the EMD with respect to the object colors. Second, 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 using the chain rule.

To further reduce the computation, signatures [9] are employed as object representations. Signatures are abstract representations of distributions and are usually clustered versions of histograms. Unlike histograms, the structures of signatures are adjustable, which can cover part of the feature space with variable-size clusters. The concept of signature will be made precise in Section 3.5. The use of color signatures significantly reduces the size of the EMD problem and

• *The authors are with the School of Engineering, University of California, Santa Cruz, 1156 High Street, Santa Cruz, CA 95064. E-mail: {zhaoqi, yangzhi, tao}@soe.ucsc.edu.*

Manuscript received 19 Apr. 2008; revised 26 Sept. 2008; accepted 3 Dec. 2008; published online 16 Dec. 2008.
Recommended for acceptance by P. Fua.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-04-0227.
Digital Object Identifier no. 10.1109/TPAMI.2008.299.

consequently requires much less computation as the EMD has an exponential worst-case running time.

Many existing tracking algorithms that consider foreground objects alone fail to estimate the object scale when the object has similar feature values for the entire object and parts of the object [10]. 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. In addition, the local background region is aligned using optical flow techniques [11] so that the algorithm naturally handles dynamic background scenes as well as static scenes. Discriminative tracking methods [12], [13], [14] have also utilized background information, but in a different manner from 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 contribution of this paper is summarized as follows:

- We propose a gradient descent method to find the solution fast when EMD is employed as the similarity measure. The ideas of applying chain rule to decompose the differential problem into a series of subproblems and the sensitivity analysis to obtain the derivatives without closed-form formulations are adopted. The DEMD algorithm can be applied to many vision problems where complex objective function makes directly taking derivative difficult. In this paper, we focus our discussions on its application to visual tracking, which achieves a speedup at orders of magnitude.
- For the first time, EMD with signature is applied to tracking. The use of EMD as a similarity measure between color distributions accommodates distribution deformations caused by object appearance changes such as illumination variations. Further, the employment of signatures offers a principled way of reducing the size of the EMD problem. We use color as the object feature in this work, but the EMD with signature can be applied to any types of features.

The rest of the paper is organized as follows: Section 2 describes related work. Section 3 discusses the details of the DEMD tracking algorithm. The key idea of DEMD is described through the discussions of DEMD tracking. 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 four areas of computer vision that bear on the work presented in this paper: object representations, similarity measures, approximation algorithms of EMD, and optimization techniques for kernel-based tracking. We briefly review the most relevant literature in each area, especially the works related to visual tracking.

2.1 Object Representations

Object representations model the characteristics of the object being tracked, which is the combined outcome of the object's shape, pose, motion, reflectance properties, and illumination conditions. The literature on object representations is vast. Some of the approaches include template-based methods [15], image-statistics-based methods [7], contour-based methods [16], feature-based methods [17], and layer-based ones [18]. 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 [7] and the CAMShift [20] algorithms were proposed using this representation for visual tracking. Adding spatial information into the otherwise spatial-information-free color histograms improves their representation power. Along this line, color spatiograms [21] and color correlograms [22] are employed for tracking, but these methods normally achieve improved discrimination power at an expense of a more complex representation, therefore, a reduced efficiency. In contrast to the fixed-size descriptors of distributions, Rubner [9] proposed signatures where the dominant clusters are extracted from the original distribution using a clustering algorithm and are used to form its compressed representation. The signatures, as variable-size descriptors, are more flexible in summarizing the image contents.

2.2 Similarity Measures for Distributions

Similarity measures between distributions broadly fall into two categories: the bin-by-bin similarity measures [19], [23], [24] that only compare contents of corresponding histogram bins and the cross-bin similarity measures [25], [26] that also compare noncorresponding bins. In practice, most existing histogram-based tracking algorithms use bin-by-bin similarity measures such as the Bhattacharyya coefficient-based distance [27] and the Sum of Squared Distance (SSD) [8]. 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. Another drawback of bin-by-bin dissimilarity measures is their sensitivity to bin size. On the other hand, the cross-bin similarity measures always yield better results in these cases, as cross-bin information is incorporated to measure the similarity. The EMD [9] naturally considers the ground distances of noncorresponding bins and calculates the similarity by solving a linear programming problem. In addition, it can be applied to variable-size descriptors which most other cross-bin distances cannot. In general, cross-bin distances are rarely employed for tracking partly due to their computational complexity.

2.3 Approximation Algorithms of EMD

Various approximation algorithms have been proposed to speed up the computation of EMD. Ling and Okada [28] suggested to reduce the computation of EMD using EMD-L1 and applied it to shape recognition and interest point matching, but the method is limited by using the L1 distance

as a ground distance. In [1], [2], random embedding techniques were coupled with locality-sensitive hashing for fast nearest neighbor image retrieval [1] and contour matching [2]. Although they achieved linear time complexity, these methods lack deterministic error bounds. Recently, Shirdhonkar and Jacobs [29] propose a wavelet EMD and illustrated its superior performance over the embedding algorithms when applied to image retrieval. The "Fragtrack" [30] suggested the use of EMD for tracking. However, due to the computation cost of EMD, the authors simply selected 10 bins with maximal counts out of a 512 bin histogram to compute the EMD.

2.4 Optimization Techniques for Kernel-Based Tracking

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. The main idea behind kernel-based tracking methods is combining the statistical features with a stochastic gradient descend method for optimization. By convolving the features with an isotropic kernel and defining a spatially smooth similarity function, the object localization problem is then reduced to the optimization of this function. Furthermore, the smoothness of the similarity function allows the application of gradient descent methods to find the location of an object very efficiently. Some commonly used gradient descent approaches include the Mean Shift algorithm [7] and the Newton style minimization procedure [8]. Generally, the use of these techniques require the objective function to be written in a closed form.

3 Differential EMD (DEMD) Tracking

3.1 Introduction to the EMD

The EMD [9] 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. The properties of the EMD, in particular its advantages over other similarity measures for distributions in the context of tracking, are summarized as follows:

- The EMD matches perceptual similarity better than bin-by-bin distances for histogram matching. This is because the EMD extends the notion of a distance between corresponding elements to that of a distance between the entire distribution, where ground distance reflects the notion of nearness properly, which further avoids quantization and other binning problems typical of histograms.
- The EMD applies to signatures, which subsume histograms [9]. The greater compactness and flexibility of signatures is in itself an advantage, as described in Section 1, and having a distance measure that can handle these variable-size structures is important.
- If the ground distance is a metric and the total weights of two signatures are equal, the EMD is a true metric (See [9] for a proof).
- Computing the EMD is based on a solution to the well-known *transportation problem* [31] from linear optimization, for which efficient algorithms, e.g., simplex methods, are available.

3.2 The EMD as a Similarity Measure

In this paper, we formulate the EMD in the specific context of visual tracking, where 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 uth 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 uth cluster in the model signature and the vth cluster in the candidate signature as d_{uv}, and the flow (amount of transported earth) between them as f_{uv}. The goal is to find the 2D image coordinate $\mathbf{y} = (x, y)^T$ for the candidate signature that corresponds to the smallest EMD

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

In (2), the inner optimization is to find the EMD for each location, and the outer one is to obtain the best object location. In the following, we use the superscript M to denote the object *model* and C for the object *candidate*. $w_v^{(C)}$ is the weight of the vth cluster in the candidate signature and $w_u^{(M)}$ the weight of the uth cluster in the model signature. $m^{(C)}$ and $m^{(M)}$ are the numbers of clusters in the candidate and model signatures, respectively. According to the definition of EMD [9], Z in (2) is formulated as

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

subject to

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

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

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

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

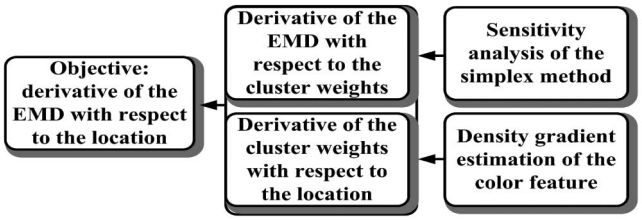


Fig. 1. Overview of the DEMD tracking algorithm.

3.3 Overview of the DEMD Tracking Algorithm

The main theoretical contribution of the paper is the derivation of a differential formula to compute the derivative of the EMD. In the visual tracking context, the goal is to obtain the derivative of the EMD with respect to the location so as to locate the objects fast. Since the formulation of the EMD is a linear programming problem, derivative of the EMD cannot be directly computed. To overcome this difficulty, we propose a two-phase algorithm, as depicted in Fig. 1.

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

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

where $w_v^{(C)}(\mathbf{y})$ is the weight of the vth cluster in the candidate signature and $m^{(C)}$ the number of clusters in the candidate signature.

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

3.4 Simplex Method and Sensitivity Analysis

The linear programming problem can be considered in geometric terms as finding an optimum in a closed convex polytope. In the problem presented in this work, the polytope is defined by intersecting $m^{(M)} + m^{(C)} + 1$ half spaces in an $m^{(M)} \times m^{(C)}$-dimensional euclidean space. The simplex method essentially works by searching the boundary of the polytope for an optimum. As illustrated in Fig. 2, the simplex algorithm begins at a starting vertex and moves along the edges of the polytope until it reaches the vertex of the optimum solution. A detailed description on simplex method is included in Appendix A. Sensitivity analysis aims to obtain the derivatives of the optimum with respect to the right-hand side (RHS) of the constraints. Intuitively, changing the RHS changes the intercept of the boundary lines. In the following, we express the linear programming problem in a matrix form and perform sensitivity analysis of the simplex method.

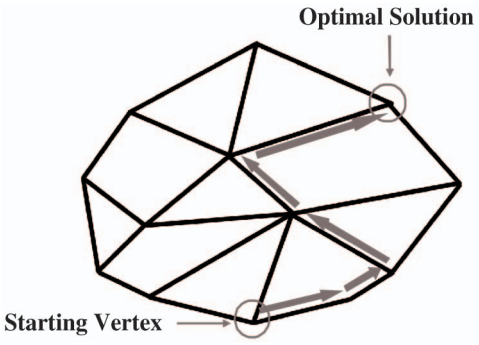


Fig. 2. An illustration of the simplex method.

3.4.1 Simplex Method in Matrix Form

To perform the sensitivity analysis, the problem in (2) is first represented in a matrix form. The starting matrix is then transformed to an optimal form based on the simplex algorithm so that 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 $f_{uv}(\mathbf{y})$ and $m^{(M)} \times m^{(C)}$ constants d_{uv} in (2), we use column vectors $\mathbf{f}(\mathbf{y})$ and $\mathbf{d}$, both of size $m^{(M)} \times m^{(C)}$, to represent the flow and the ground distance as

$$\mathbf{f}(\mathbf{y}) = [f_{11}(\mathbf{y}) \cdots f_{1m^{(C)}}(\mathbf{y}) \cdots f_{m^{(M)}1}(\mathbf{y}) \cdots f_{m^{(M)}m^{(C)}}(\mathbf{y})]^T$$

and

$$\mathbf{d} = [d_{11} \cdots d_{1m^{(C)}} \cdots d_{m^{(M)}1} \cdots d_{m^{(M)}m^{(C)}}]^T.$$

Stacking the first three equations of the constraints in (2), the coefficients c_{uv} ($1 \le u \le m^{(M)}, 1 \le v \le m^{(C)}$), which are either 1 or 0, can form a 2D matrix of $m^{(M)} + m^{(C)} + 1$ rows and $m^{(M)} \times m^{(C)}$ columns. Denoting this coefficient matrix as $\mathbf{H}$ and representing $[(\mathbf{w}^{(C)}(\mathbf{y}))^T (\mathbf{w}^{(M)})^T 1]^T$ as $\mathbf{b}(\mathbf{y})$ yield

$$\mathbf{H} = \begin{bmatrix} c_{11} & 0 \cdots & 0 & \cdots & c_{m^{(M)}1} & 0 \cdots & 0 \\ & & & \vdots & & & \\ 0 & \cdots 0 & c_{1m^{(C)}} & \cdots & 0 & \cdots 0 & c_{m^{(M)}m^{(C)}} \\ c_{11} & \cdots & c_{1m^{(C)}} & \cdots & 0 & \cdots & 0 \\ & & & \vdots & & & \\ 0 & \cdots & 0 & \cdots & c_{m^{(M)}1} & \cdots & c_{m^{(M)}m^{(C)}} \\ c_{11} & \cdots & c_{1m^{(C)}} & \cdots & c_{m^{(M)}1} & \cdots & c_{m^{(M)}m^{(C)}} \end{bmatrix}$$

and

$$\mathbf{b}(\mathbf{y}) = \begin{bmatrix} w_1^{(C)}(\mathbf{y}) \cdots w_{m^{(C)}}^{(C)}(\mathbf{y}) & w_1^{(M)} \cdots w_{m^{(M)}}^{(M)} & 1 \end{bmatrix}^T.$$

Then, we have the matrix form of (2) as

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

subject to

$$\mathbf{H}\mathbf{f}(\mathbf{y}) = \mathbf{b}(\mathbf{y}),$$

$$\mathbf{f}(\mathbf{y}) \ge \mathbf{0}.$$

TABLE 1
Starting Tableau

Z	$\mathbf{f}_B$	$\mathbf{f}_{NB}$	RHS
1	$-(\mathbf{d}_B^T)^{(S)}$	$-(\mathbf{d}_{NB}^T)^{(S)}$	0
$\mathbf{0}$	$(\mathbf{H}_B)^{(S)}$	$(\mathbf{H}_{NB})^{(S)}$	$(\mathbf{b})^{(S)}$

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, we can always formulate $m^{(M)} + m^{(C)} + 1$ basic variables, i.e., variables of nonzero value, and $m^{(M)} \times m^{(C)} - (m^{(M)} + m^{(C)} + 1)$ nonbasic variables, as shown in Appendix A.2. Grouping all of the basic variables together and all of the nonbasic 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 nonbasic variables. The ground distance vector $\mathbf{d}$ is similarly divided as $[\mathbf{d}_B^T\ \mathbf{d}_{NB}^T]^T$ and $\mathbf{H} = [\mathbf{H}_B\ \mathbf{H}_{NB}]$. Thus, the starting tableau for the simplex method is written as in Table 1.

   In this table, RHS denotes the right-hand side of the equations. The second row corresponds to the objective function of (4), and the third row is a vector representation of the constraints in (4). The superscript S denotes starting tableau.
2. Applying the simplex algorithm yields the following optimal tableau, where the sets of basic variables and nonbasic variables change, and so are all their coefficients. After matrix transformations ((17)-(19) in Appendix A.1), the optimal tableau can be reformulated as shown in Table 2.

   This reformulated optimal tableau is important for the sensitivity analysis, as will be discussed in the next section.

3.4.2 Sensitivity Analysis

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

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

$$\begin{aligned} Z' = \mathbf{d}_B^T \mathbf{H}_B^{-1} \mathbf{b}' &= \mathbf{d}_B^T \mathbf{H}_B^{-1} \mathbf{b} + \mathbf{d}_B^T \mathbf{H}_B^{-1} [0..0 \Delta b_i 0..0]^T \\ &= \mathbf{d}_B^T \mathbf{H}_B^{-1} \mathbf{b} + k_i \Delta b_i, \end{aligned}$$

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

TABLE 2
Reformulated Optimal Tableau

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

(a) (b) (c)

Fig. 3. An example of a color signature. (a) Original image (from the *MIT Pedestrian Data set* [32]). (b) Rendered image using a 16-cluster signature. (c) Color signature.

Therefore,

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

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

$$\frac{\partial Z}{\partial b_i} = k_i - \sum_{j \ne i} k_j \frac{b_j}{\sum_{j \ne i} b_j}, i = 1, ..., m^{(C)}. \tag{6}$$

The proof is given in Appendix B. The intuition of this equation is the projection of the k_i (5) from an $m^{(C)}$-dimensional space to an $m^{(C)} - 1$-dimensional space, where the "-1" is imposed by the constraint of $\sum_{i=1}^{m^{(C)}} b_i = 1$. Equation (6) provides an explicit formula of how the EMD would change with respect to the color changes, considering all the effective constraints.

3.5 Density Gradient Estimation of the Color Feature

3.5.1 Representing Objects Using Color Signatures

In this paper, we use color information as the feature, and the object is represented by its p.d.f. in the color space, which can be estimated using kernel density estimation. Signatures are used to represent the objects, due to its compactness. Fig. 3 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. Signatures are very efficient and effective for distributions with sparse structures, which is the case for object tracking where the target of interest normally has limited numbers of major clusters in the feature space.

By convolving the color signatures with isotropic kernels, the smoothness of the feature density allows gradient estimation of the representation. Fig. 4a corresponds to one cluster of the color signature of an image patch, where the yellow grids correspond to the pixels whose color fall into a certain cluster and the green grids correspond to the opposite. Fig. 4b is an isotropic kernel and Fig. 4c shows the smooth feature density after the convolution.

Without loss of generality, the object model is considered as centered at the spatial location $\mathbf{0}$ and a kernel-based representation [7] is defined according to (1) as

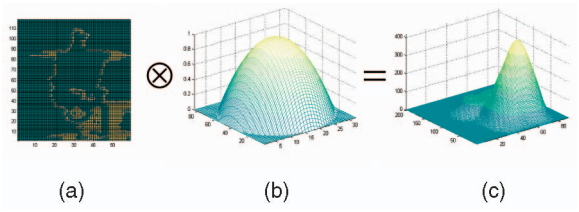


Fig. 4. An example of kernel-based statistical features. (a) Statistical color feature. (b) An isotropic kernel. (c) Resulting feature density.

$$\mathbf{s}^{(M)} = \{S_u^{(M)}\}, u = 1, .., m^{(M)}, \tag{7}$$

where

$$S_u^{(M)} = \left(a_u^{(M)}, w_u^{(M)}\right)$$

and

$$w_u^{(M)} = \beta \sum_{n=1}^{N} K\left(\frac{\mathbf{x}_n}{h}\right) \delta[c(\mathbf{x}_n) - u]. \tag{8}$$

In this equation, the density in the feature space is clustered into $m^{(M)}$ clusters, where the superscript M denotes object *model*. Correspondingly, we use superscript C for object *candidate* in later definitions, $\mathbf{x}_n$ denotes 2D image coordinates, the number of pixels is N, c is a function that associates the pixel at location $\mathbf{x}$ to the cluster which is the nearest to the color of that pixel, and $K(\mathbf{x})$ is an isotropic kernel that 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, given by

$$\beta = \frac{1}{\sum_{n=1}^{N} K\left(\frac{\mathbf{x}_n}{h}\right)}.$$

Similarly, the object candidate is defined at location $\mathbf{y}$ as

$$\mathbf{s}^{(C)}(\mathbf{y}) = \{S_v^{(C)}(\mathbf{y})\}, v = 1, .., m^{(C)}, \tag{9}$$

where

$$S_v^{(C)}(\mathbf{y}) = \left(a_v^{(C)}(\mathbf{y}), w_v^{(C)}(\mathbf{y})\right)$$

and

$$w_v^{(C)}(\mathbf{y}) = \gamma \sum_{n=1}^{N} K\left(\frac{\mathbf{x}_n - \mathbf{y}}{h}\right) \delta[c(\mathbf{x}_n) - v]. \tag{10}$$

In (10), the feature space has $m^{(C)}$ clusters and γ normalizes the feature as

$$\gamma = \frac{1}{\sum_{n=1}^{N} K\left(\frac{\mathbf{x}_n - \mathbf{y}}{h}\right)}.$$

3.5.2 Estimation of the Density Gradient

Taking the gradient of the cluster weights (10), we have the density gradient of the color feature as

$$\nabla_{\mathbf{y}} w_v^{(C)}(\mathbf{y}) = \frac{2\gamma}{h^2} \sum_{n=1}^{N} (\mathbf{x}_n - \mathbf{y}) g\left(\left\|\frac{\mathbf{y} - \mathbf{x}_n}{h}\right\|^2\right) \delta[c(\mathbf{x}_n) - v]. \tag{11}$$

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

3.6 Closed-Form DEMD Tracking

Recall from (3) that the gradient descent representation of the EMD is

$$\nabla_{\mathbf{y}} Z(\mathbf{y}) = \sum_{v=1}^{m^{(C)}} \frac{\partial Z(\mathbf{y})}{\partial w_v^{(C)}(\mathbf{y})} \nabla_{\mathbf{y}} w_v^{(C)}(\mathbf{y}).$$

Substituting the RHS of (6) for $\frac{\partial Z(\mathbf{y})}{\partial w_v^{(C)}(\mathbf{y})}$ and the RHS of (11) for $\nabla_{\mathbf{y}} w_v^{(C)}(\mathbf{y})$ yields

$$\nabla_{\mathbf{y}} Z(\mathbf{y}) = \frac{2\gamma}{h^2} \sum_{n=1}^{N} (\mathbf{x}_n - \mathbf{y}) g\left(\left\|\frac{\mathbf{y} - \mathbf{x}_n}{h}\right\|^2\right) \pi_n. \tag{12}$$

In (12), the weight of each pixel $\mathbf{x}_n$ is

$$\pi_n = \sum_{v=1}^{m^{(C)}} \left(k_v - \sum_{j \neq v} k_j \frac{b_j}{\sum_{j \neq v} b_j}\right) \delta[c(\mathbf{x}_n) - v], \tag{13}$$

where

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

Thus, the distance minimization can be efficiently achieved based on (12), using the following algorithm:

Algorithm 1. Fast *Differential EMD (DEMD)* Tracking Procedure

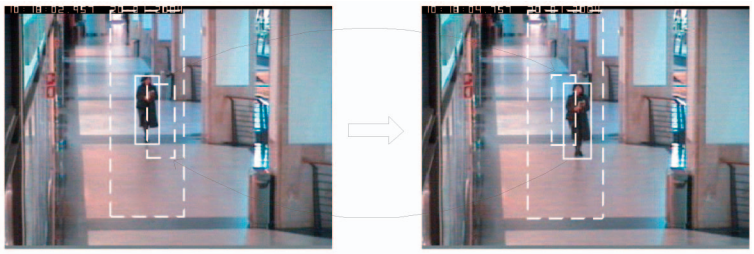
Input: Object center of the previous frame $\mathbf{y}_0 = \mathbf{y}^{(i-1)}$
Output: Initialized object center for the current frame $\mathbf{y}_0^{(i)}$

- Initialize the location of the object in the current frame with $\mathbf{y}_0$. Evaluate $EMD(\mathbf{y}_0)$ using (2).
- Compute the weights $\{\pi_n\}_{n=1,..,N}$ for the pixels in the tracking window according to (13).
- Compute the gradient $\nabla_{\mathbf{y}} Z(\mathbf{y}_0)$ based on (12).
- Move the object along the gradient vector to one of its 8 neighboring pixels $\mathbf{y}_1$. Evaluate $EMD(\mathbf{y}_1)$ using (2).
- If $EMD(\mathbf{y}_1) > EMD(\mathbf{y}_0)$, set $\mathbf{y}_0^{(i)} \leftarrow \mathbf{y}_0$ and stop; otherwise, set $\mathbf{y}_0 \leftarrow \mathbf{y}_1$ and go to the 1^{st} step.

4 DEMD Tracking with Background Modeling

4.1 DEMD Tracking with Background Modeling

The DEMD tracking algorithm provides accurate results under most scenarios. However, the method may be insufficient in certain scenarios such as scale changes and background clutter. When the feature of the entire object is similar with those of object parts, modeling the object region and evaluating its similarity could cause the tracker to lock onto part of the object. In [10], a scale kernel is applied with the spatial kernel to deal with the problem. With the Epanechnikov kernel used by the authors, the mean shift iterations to find the best scale is equivalent to averaging different scales in the scale space. As illustrated in Algorithm 2, the first step of evaluating different scales is similar to [10]. In addition, we observe that, in many cases



(a) (b)

Fig. 5. DEMD tracking with background modeling. (a) The $(t-1)$th frame. (b) The tth 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 tth 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 tth frame, should match the same area of the previous frame.

such as scales changes, the trackers' awareness of what is not included in the tracking window is as important as their knowledge of what is in the window. This observation naturally motivates the modeling of local background scenes, as well as the foreground regions. As long as the object feature is not exactly the same as background features, the modeling and estimation of local background scenes prevent the tracker locking onto part of the objects. In addition, it alleviates the drifting problem in that the similarity measure of the background scenes adds an additional cue to estimate the final object state.

Specifically, 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 $\mathbf{y}$ and scale σ corresponding to the smallest sum of the EMD for the foreground object and the EMD for the local background scene:

$$\arg\min_{\mathbf{y},\sigma}\left(\min_{f_{uv}} Z(f_{uv}(\mathbf{y},\sigma)) + \min_{f_{uv}^{(Bg)}} Z^{(Bg)}\left(f_{uv}^{(Bg)}(\mathbf{y},\sigma)\right)\right), \quad (14)$$

where the superscript Bg denotes the local background scenes. The formulations for Z are addressed in (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 tracking 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 (14). Fig. 5 illustrates the method and the detailed algorithm for the adjustment step is given as follows:

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

Input: Object center $\mathbf{y}_0 = \mathbf{y}_0^{(i)}$ returned by Algorithm 1
Object scale from the previous frame $\sigma_0 = \sigma^{(i-1)}$.

Output: Object center $\mathbf{y}^{(i)}$ and scale $\sigma^{(i)}$ of the current frame

- Initialize the object location with $\mathbf{y}_0$, vary σ_0 by $+/-10\%$ and evaluate which scale is the best using (14).
- If the scale with the smallest distance σ_1 equals σ_0, set $\sigma^{(i)} \leftarrow \sigma_0, \mathbf{y}^{(i)} \leftarrow \mathbf{y}_0$ and stop; otherwise, set $\sigma_0 \leftarrow \sigma_1$, and run a numerical gradient algorithm, which calculates the sum of EMDs (14) only on two neighbors of the current pixel, to obtain the new location $\mathbf{y}_1$.
- If $\mathbf{y}_1$ equals $\mathbf{y}_0$, set $\sigma^{(i)} \leftarrow \sigma_0$, $\mathbf{y}^{(i)} \leftarrow \mathbf{y}_0$ and stop; otherwise, set $\mathbf{y}_0 \leftarrow \mathbf{y}_1$ and go to the first step.

4.2 Background Alignment

When the background is static, features in 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 [11]. 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.

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 of these experiments, a simple "*divide and recombine*" strategy [9] is applied to compute color signatures of the image regions. In fact, a nice property of the EMD is its insensitivity to the clustering algorithm that is used to find the significant clusters. In the clustering procedure, 16-cluster signatures are empirically proved to be sufficient, and 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. Fig. 6 shows the results of the standard Mean Shift (MS) tracker that 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

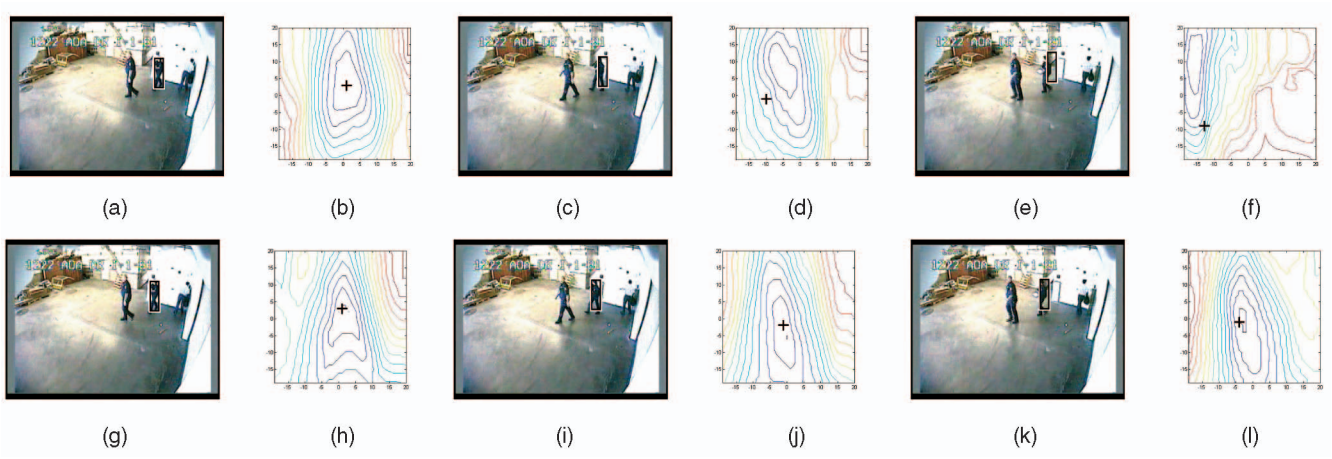


Fig. 6. Comparison between the Bhattacharyya-coefficient-based distance and the EMD. Frames 1, 11, and 21 from the *Pedestrian* sequence are shown. (a), (c), and (e) Tracking results of the MS tracker. (b), (d), and (f) Bhattacharyya-coefficient-based distances for a 40×40 region. "+" indicates the ground truth object location. (g), (i), and (k) Tracking results of the DEMD tracker. (h), (j), and (l) EMDs.

distances. Figs. 6j and 6l 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 Figs. 6d and 6f, 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.

Fig. 7 shows the number of iterations for the DEMD tracker on the *Pedestrian* sequence. The average number of iterations is 3.01 iterations per frame. In comparison, using brute-force search, the search range would be at least 10×10 to cover the interframe translation, which leads to a 100 iterations. The differential method requires a much smaller 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 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. Figs. 8 and 9 illustrate the tracking results. Fig. 8a reveals the MS tracker's incapability in dealing with illumination changes caused by the shadows, while in Fig. 8b, the DEMD tracker keeps track of the car into and out of strong shadows. Fig. 9a is another example where the MS tracker loses track of the object fast. However, despite the shadows and the fast motion, the DEMD tracker maintains a secure focus on the object throughout the sequence (Fig. 9b).

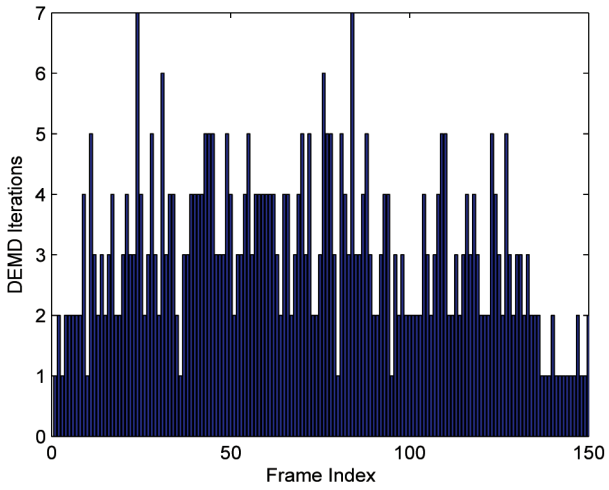


Fig. 7. The number of DEMD iterations versus frame index for the *Pedestrian* sequence.

The fourth experiment is performed on the OTCBVS benchmark data [33]. The tracking results using the DEMD tracker with 16-cluster color signatures are presented in Fig. 10a. It can be observed that the tracker outputs quite accurate object locations. However, due to the lack of any scale adaptation mechanism, the performance degenerates in cases of large-scale changes. Eventually, the tracker loses the object when it passes other pedestrians with similar colors.

5.2 Examples of the DEMD Tracking with Background Modeling

As shown in Fig. 10b, for the same OTCBVS sequence, the DEMDB tracker keeps tight track of the object due to the consideration of local background scenes, as discussed in Section 4. Thereby, the DEMDB tracker is more robust against background distraction.

In the fifth experiment, we track a vehicle in the *RedTeam* sequence from the PETS '05 data set, where the background is dynamic. Using the background alignment technique introduced in Section 4.2, the DEMDB tracker accommodates to the dynamic background, and Fig. 16b shows that the tracker deals with scale changes and moderate camera motion reliably.

5.3 Quantitative Results

We have conducted a quantitative evaluation of the DEMD algorithm and 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 $+/-10$ percent and choosing the one with the smallest distance [7], is implemented. The

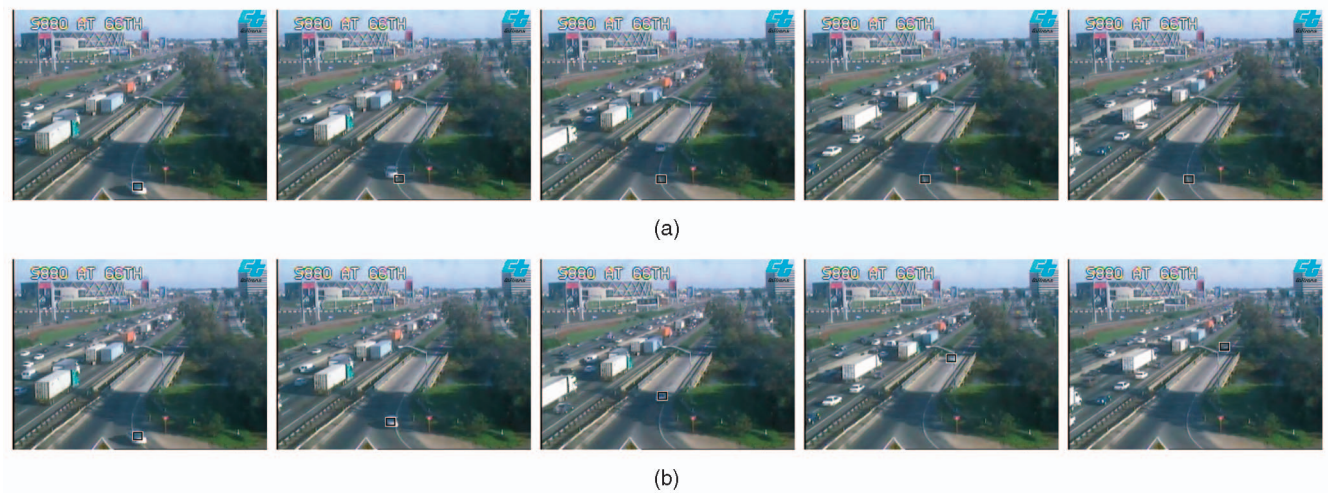


Fig. 8. Frames 1, 13, 35, 86, and 110 from the *Highway-Car* sequence. (a) The MS tracker starts to wander when the car is entering the strong shadow and fails around frame 13. (b) The DEMD tracker successfully follows the car into and out of strong shadows.

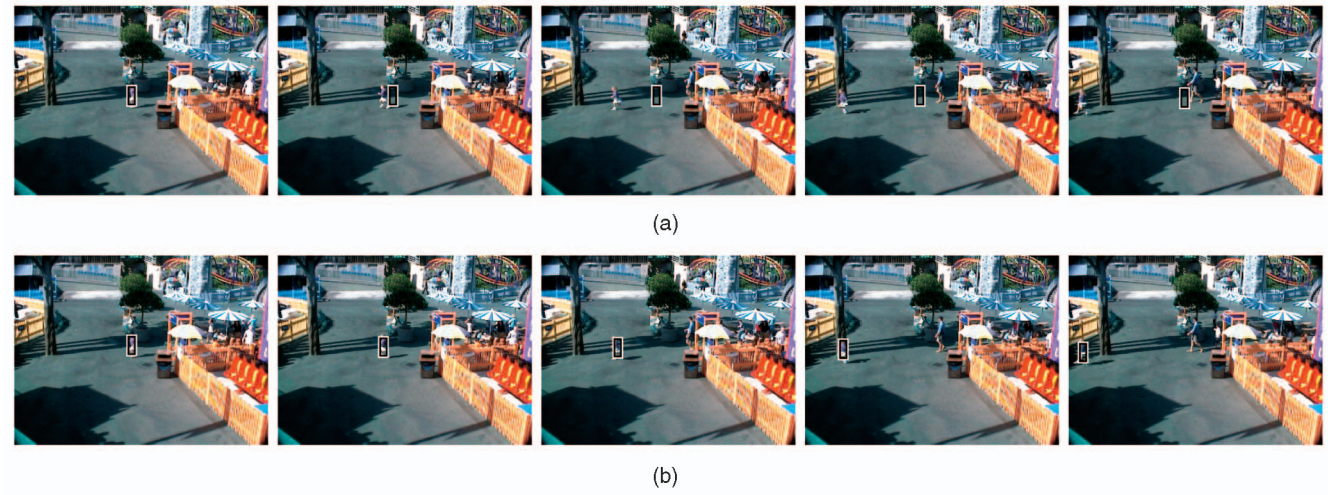


Fig. 9. Frames 1, 10, 30, 51, and 65 from the *Running-Girl* sequence. (a) The MS tracker has lost track of the girl by frame 10. (b) The DEMD tracker maintains a secure focus on the object throughout the sequence though there are shadows all along the path and the object is moving fast.

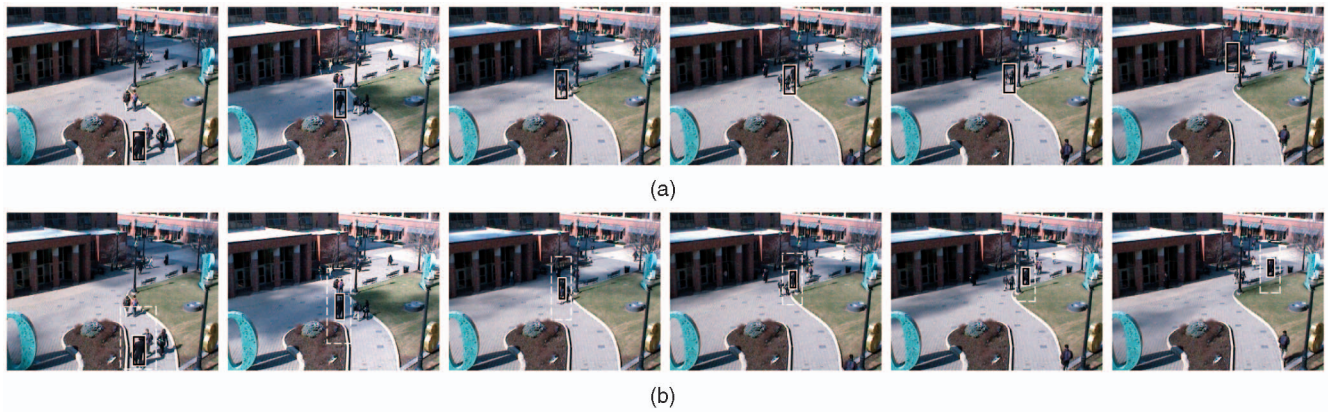


Fig. 10. Frames 1, 161, 289, 354, 405, and 486 from the *OTCBVS* sequence. (a) The performance of the DEMD tracker degenerates as the object becomes smaller and eventually loses track of it. (b) The DEMDB tracker keeps tight track of the object.

same-scale adaptation scheme is employed for the DEMD method without background modeling for a fair comparison. We use six sequences taken from the public PETS '01, PETS '04, and PETS '05 data sets [34], where ground truth data are available. Quantitative results are shown in Table 3 and Figs. 11, 12, 13, 14, 15, and 16 display sample frames of the tracking results. All of 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

TABLE 3
Quantitative Results for Several Public Data of the Proposed DEMD Tracker, DEMD Tracker with Background Modeling (DEMDB) and Their Comparison with the Standard MS Tracker

Source Dataset	Description / File Name	Frames Tracked			Position Error			Size Error		
		MS	DEMD	DEMDB	MS	DEMD	DEMDB	MS	DEMD	DEMDB
PETS'01	Red-Coat Female - Cam2	577/651	651/651	651/651	0.284	0.180	**0.133**	0.264	0.197	**0.136**
PETS'01	White Van - Cam1	135/260	149/260	260/260	0.229	0.241	**0.090**	0.312	0.353	**0.152**
PETS'04	Female - Front View	81/162	162/162	162/162	0.260	0.169	**0.156**	0.135	0.126	**0.120**
PETS'04	Female - Corridor View	381/381	381/381	381/381	0.047	0.046	**0.040**	**0.057**	0.060	0.072
PETS'04	Male - Corridor View	550/550	550/550	550/550	0.097	**0.089**	0.102	0.133	0.130	**0.122**
PETS'05*	RedTeam	1918/1918	1918/1918	1918/1918	0.170	0.117	**0.056**	0.282	0.227	**0.108**

In data sets with $*$, ground truth was provided every 10 frames and we count only these frames with ground truth for comparison. Others have ground truth for each frame.

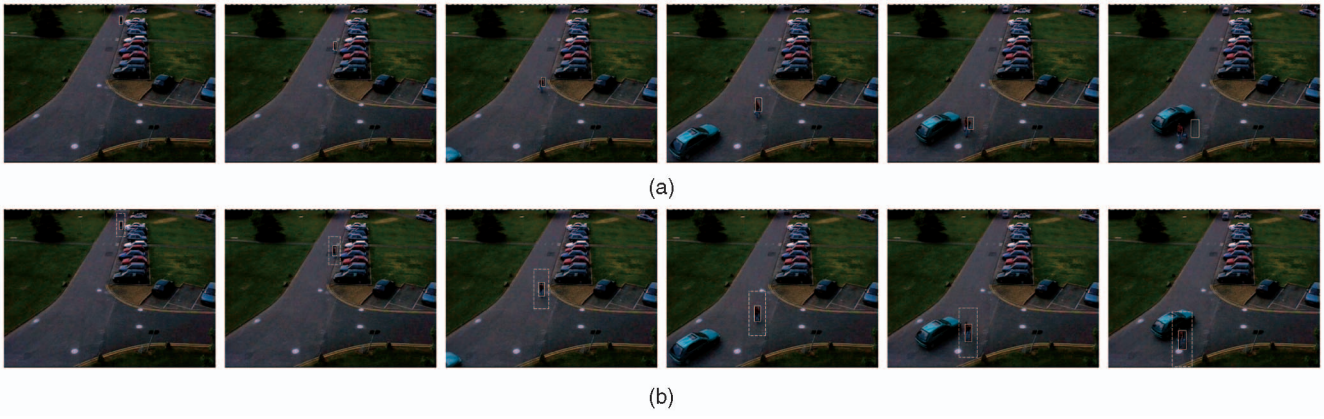


Fig. 11. Frames 1, 279, 480, 540, 574, and 588 from the *Red-Coat* sequence. (a) The MS tracker does not keep tight track of the pedestrian, and finally, loses track of it. (b) The DEMDB tracker tracks the pedestrian reliably throughout the sequence.

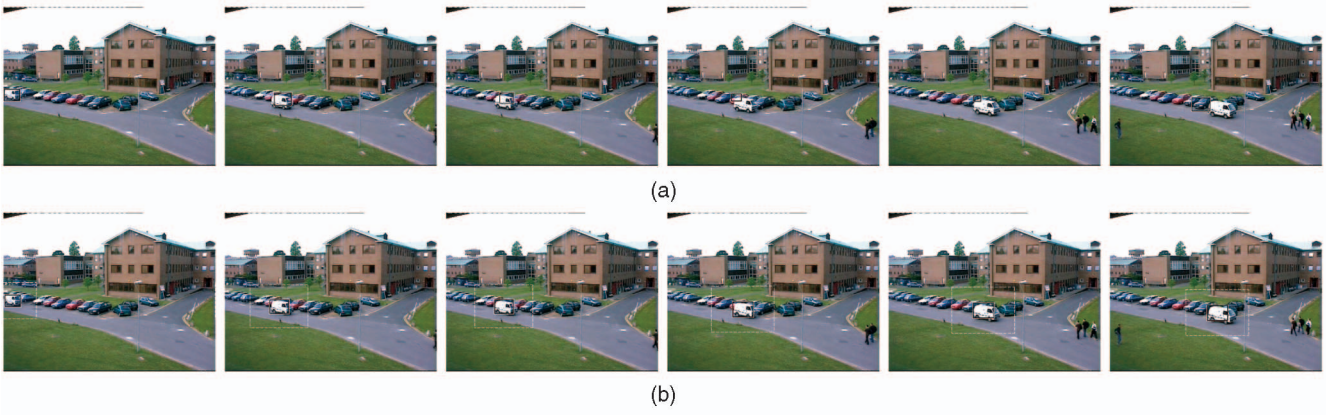


Fig. 12. Frames 1, 64, 67, 89, 114, and 133 from the *White Van-Cam1* sequence. (a) The tracking window of the MS tracker tends to shrink as the van is of uniform color. (b) The DEMDB tracker prevents the window shrinkage by considering the background consistency.

calculated as the euclidean 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 euclidean distance between the two (height and width) vectors, normalized by the ground truth length of the object's diagonal.

In Table 3, the MS, DEMD, and DEMDB trackers track throughout three, five, and six out of the six sequences, respectively. The DEMD tracker outperforms the MS tracker in terms of accuracy, especially when illumination changes happen, e.g., some frames in the "*Female-Front View*" (Fig. 13) and "*Male-Corridor View*" (Fig. 15) sequences. Additionally, the DEMDB tracker outperforms the other two algorithms, due to its 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 objects become larger [10]. The "*White Van*" sequence (Fig. 12) clearly illustrates this point. Both the MS and DEMD trackers lose track of the van since the trackers get confused on the object scale due to the van's mostly uniform color, and finally, drifts away. Generally, the scale adaptation scheme for the MS and DEMD trackers suffer less when the object is becoming smaller, e.g., they achieve performance comparable to the DEMDB tracker for

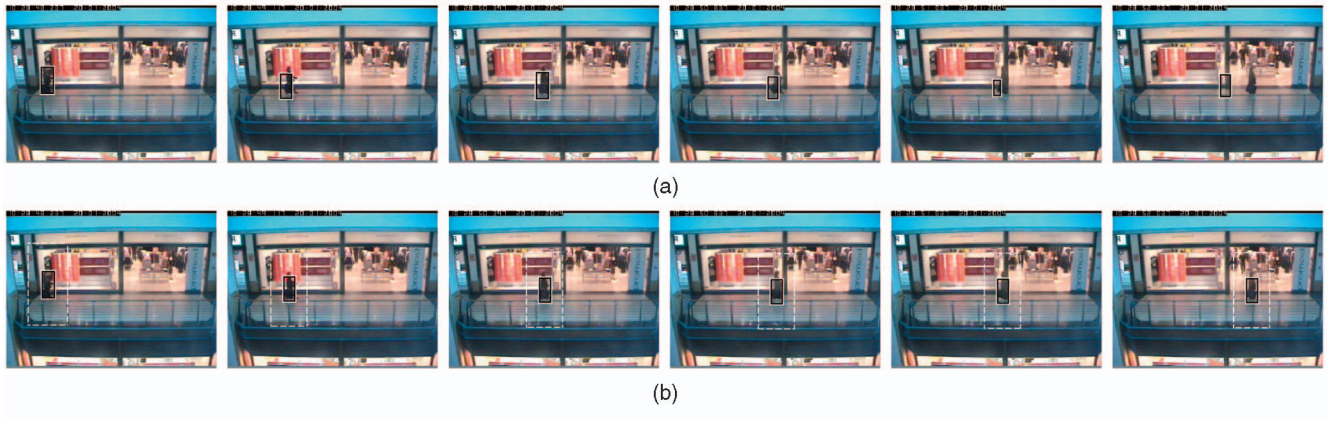


Fig. 13. Frames 1, 23, 55, 66, 71, and 96 from the *Female-Front View* sequence. (a) The MS tracker performs poorly under the illumination changes due to reflection and eventually locks track onto the background scene. (b) The DEMDB tracker shows its better capability in dealing with illumination changes and background clutter.

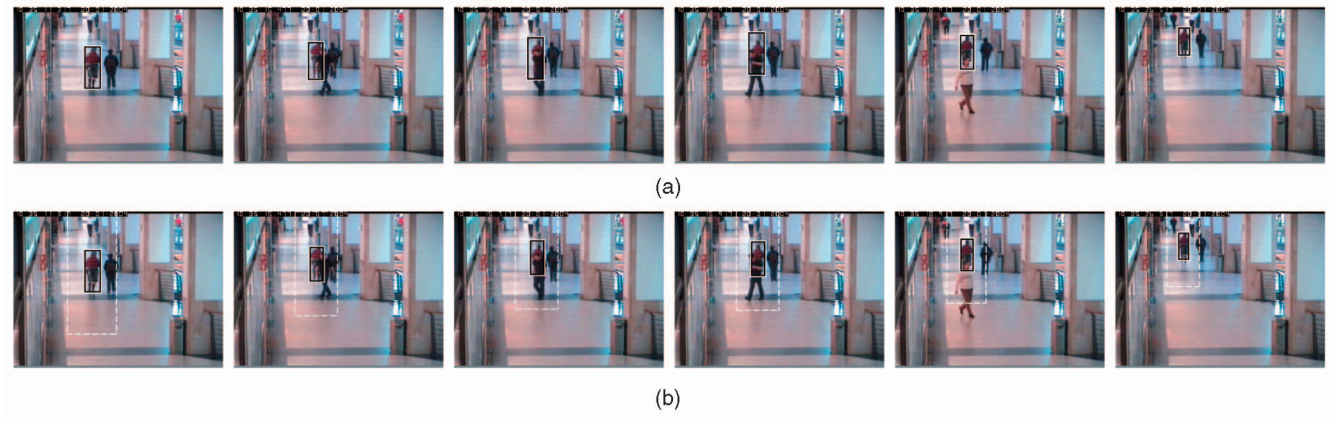


Fig. 14. Frames 1, 104, 111, 116, 190, and 381 from the *Female-Corridor View* sequence. Both the MS and DEMDB trackers keep track of the women well, even under heavy occlusion.

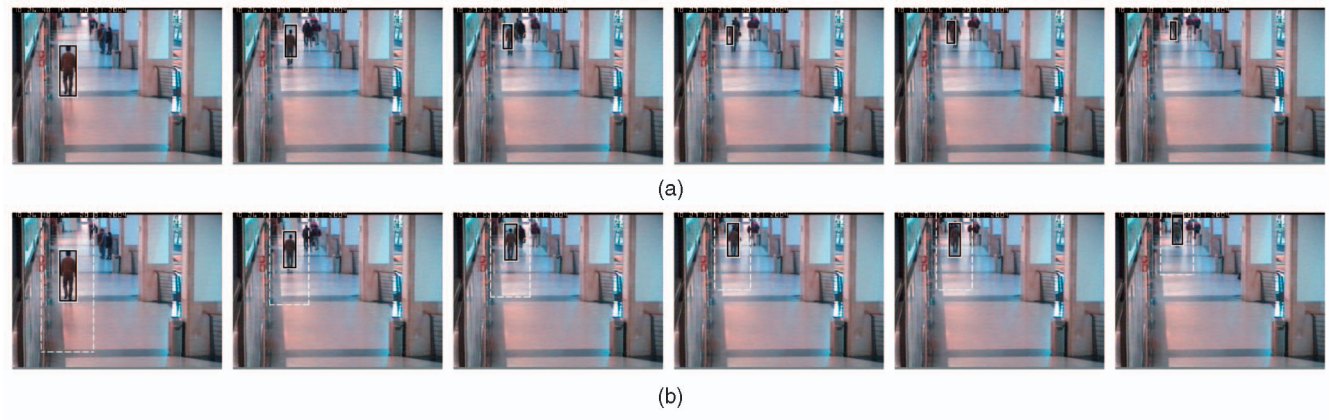


Fig. 15. Frames 1, 242, 356, 402, 459, and 549 from the *Male-Corridor View* sequence. (a) The MS tracker tends to lock track onto parts of the pedestrian. (b) The performance of the DEMDB tracker degenerates when the background is similar to the foreground object, as shown in the last two sample frames.

the "*Female-Corridor View*" (Fig. 14) and "*Male-Corridor View*" (Fig. 15) sequences, when the pedestrians walk away from the camera. However, for those where the objects become larger, e.g., the red-coat female (Fig. 11) and the vehicle in the "*RedTeam*" sequences (Fig. 16), the DEMDB tracker outputs more accurate results due to its principled way of estimating object position and scale. Please refer to the supplemental video, which can be found at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2008.299, for video results.

6 Discussions

The main theoretical contribution of this work is the development of a fast differential EMD algorithm which significantly reduces the number of running iterations of

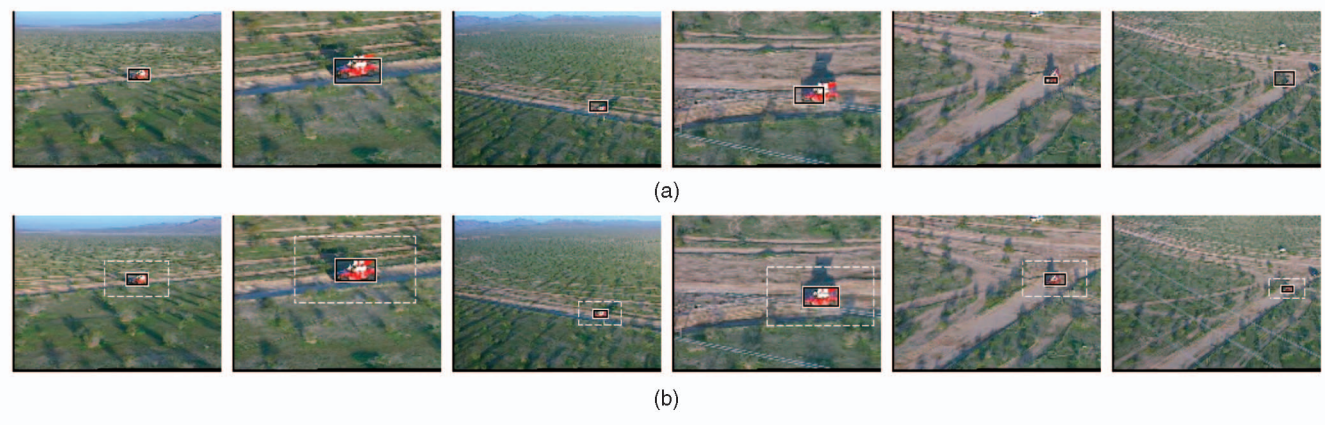


Fig. 16. Frames 1, 538, 1,005, 1,619, 1,835, and 1,890 from the *RedTeam* sequence, recorded with a moving camera. (a) The MS tracker fails to estimate the object scale reliably. (b) The DEMDB keeps tight track of the object under dynamic background.

calculating the EMD. The EMD was previously used in the image retrieval literature, yet its complex computation prohibits its use in many applications that require a fast performance. The general differential EMD algorithm that is based on the chain rule and the sensitivity analysis is applicable to many vision problems where the objective function is expressed in a nonclosed form.

This paper discusses the general idea of DEMD and then focuses on its application to real-time visual tracking. The employment of the EMD as a similarity measure accommodates various appearance changes happening at different time instances and the DEMD algorithm makes the use of EMD at a real-time performance possible. Further, the signatures are employed to reduce the size of the EMD problem significantly. To the best of our knowledge, this is the first work using the EMD and signatures in visual tracking. To handle scale changes in a principled way, both foreground objects and background scenes are modeled and estimated in the tracking procedure. Extensive experiments have demonstrated 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.

APPENDIX A
SIMPLEX METHOD

The simplex method invented by Dantzig is an efficient procedure for generating an optimal solution of the linear programming problem. As a general description of a linear programming problem [35]

$$\arg\min_{\mathbf{x}} Z = \mathbf{d}^T\mathbf{x}, \tag{15}$$

subject to

$$\mathbf{H}\mathbf{x} = \mathbf{b},$$

where $\mathbf{H} \in \mathbb{R}_{m\times n}(m \le n)$, $\mathbf{x} \in \mathbb{R}_{n\times 1}$, $\mathbf{d} \in \mathbb{R}_{n\times 1}$, and $\mathbf{x} \ge \mathbf{0}$. A matrix description of (15) is

$$\begin{bmatrix} 1 & -\mathbf{d}^T \\ \mathbf{O} & \mathbf{H} \end{bmatrix}\begin{bmatrix} Z \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{b} \end{bmatrix}. \tag{16}$$

In the following, the optimization procedure of the simplex method is first illustrated with the assumption that the rows of $\mathbf{H}$ are linearly independent, followed by the discussions of basic variable selection to start the simplex procedure for arbitrarily ranked $\mathbf{H}$.

A.1 The Simplex Method Procedure

With the assumption that the rows of $\mathbf{H}$ are linearly independent, $rank(\mathbf{H}) = m$. For the convenience of discussion and without loss of generality, the first m columns are assumed to be linearly independent and denoted as $\mathbf{H}_B = [\mathbf{H}_{(1,:)}\ \mathbf{H}_{(2,:)} \dots \mathbf{H}_{(m,:)}]$; the rest columns of $\mathbf{H}$ are denoted as $\mathbf{H}_{NB} = [\mathbf{H}_{(m+1,:)}\ \mathbf{H}_{(m+2,:)} \dots \mathbf{H}_{(n,:)}]$. Let $\mathbf{x} = [\mathbf{x}_B^T\ \mathbf{x}_{NB}^T]^T$ with $\mathbf{x}_B = [x_1\ x_2 \dots x_m]^T$ and

$$\mathbf{x}_{NB} = [x_{m+1}\ x_{m+2} \dots x_n]^T,$$

where $\mathbf{x}_B$ and $\mathbf{x}_{NB}$ are called the basic and nonbasic variables, respectively. Further, let $\mathbf{d} = [\mathbf{d}_B^T\ \mathbf{d}_{NB}^T]^T$, where $\mathbf{d}_B = [d_1\ d_2 \dots d_m]^T$ and $\mathbf{d}_{NB} = [d_{m+1}\ d_{m+2} \dots d_n]^T$, then (16) becomes

$$\begin{bmatrix} 1 & -\mathbf{d}_B^T & -\mathbf{d}_{NB}^T \\ \mathbf{O} & \mathbf{H}_B & \mathbf{H}_{NB} \end{bmatrix}\begin{bmatrix} Z \\ \mathbf{x}_B \\ \mathbf{x}_{NB} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{b} \end{bmatrix}. \tag{17}$$

To obtain a solution to (17) and evaluate whether the solution is optimal or not, left multiply (17) with

$$\begin{bmatrix} 1 & -\mathbf{d}_B^T \\ \mathbf{O} & \mathbf{H}_B \end{bmatrix}^{-1} = \begin{bmatrix} 1 & \mathbf{d}_B^T\mathbf{H}_B^{-1} \\ \mathbf{O} & \mathbf{H}_B^{-1} \end{bmatrix}. \tag{18}$$

After the left multiplication, the objective function Z becomes uncorrelated with the first linearly independent m columns, as shown in the matrix form:

$$\begin{bmatrix} 1 & 0 & \mathbf{d}_B^T\mathbf{H}_B^{-1}\mathbf{H}_{NB} - \mathbf{d}_{NB}^T \\ \mathbf{O} & \mathbf{I}_m & \mathbf{H}_B^{-1}\mathbf{H}_{NB} \end{bmatrix}\begin{bmatrix} Z \\ \mathbf{x}_B \\ \mathbf{x}_{NB} \end{bmatrix} = \begin{bmatrix} \mathbf{d}_B^T\mathbf{H}_B^{-1}\mathbf{b} \\ \mathbf{H}_B^{-1}\mathbf{b} \end{bmatrix}, \tag{19}$$

where $\mathbf{I}_m$ is the identity matrix of rank m. A solution to (16) is $\mathbf{x} = [\mathbf{x}_B^T\ \mathbf{x}_{NB}^T]^T$, where $\mathbf{x}_B = \mathbf{H}_B^{-1}\mathbf{b}$ and $\mathbf{x}_{NB} = [0 \dots 0]^T$. As shown in [35], the sufficient conditions which lead to the conclusion that the solution is global optimal are

- $\mathbf{H}_B^{-1}\mathbf{b} \geq \mathbf{0}$;
- $\mathbf{d}_B^T\mathbf{H}_B^{-1}\mathbf{H}_{NB} - \mathbf{d}_{NB}^T \leq \mathbf{0}$.

Theorem 1. *The condition that the solution* $\mathbf{x} = [\mathbf{x}_B^T\ \mathbf{x}_{NB}^T]^T$ ($\mathbf{x}_B = \mathbf{H}_B^{-1}\mathbf{b}$ *and* $\mathbf{x}_{NB}^T = [0 \dots 0]^T$) *to (15) is feasible, i.e.,* $\mathbf{H}_B^{-1}\mathbf{b} \geq \mathbf{0}$, $\mathbf{d}_B^T\mathbf{H}_B^{-1}\mathbf{H}_{NB} - \mathbf{d}_{NB}^T \leq \mathbf{0}$ *is a sufficient condition to a global minimum of* Z. *Particularly, there is only one global optimal solution if* $\mathbf{d}_B^T\mathbf{H}_B^{-1}\mathbf{H}_{NB} - \mathbf{d}_{NB}^T < \mathbf{0}$ *holds.*

Proof. Suppose there is a different solution $\mathbf{y}$, $\mathbf{y} \neq \mathbf{x}$ and $\mathbf{y} > \mathbf{0}$ satisfying $\mathbf{H}\mathbf{y} = \mathbf{b}$.

By definition, we have

$$\mathbf{b} = \mathbf{H}_B\mathbf{x}_B \tag{20}$$

and

$$\mathbf{b} = \mathbf{H}\mathbf{y} = \mathbf{H}_B\big[\mathbf{I}_m \quad \mathbf{H}_B^{-1}\mathbf{H}_{NB}\big]\mathbf{y}. \tag{21}$$

Combining (20) and (21) yields

$$\mathbf{H}_B\mathbf{x}_B = \mathbf{H}_B\big[\mathbf{I}_m \quad \mathbf{H}_B^{-1}\mathbf{H}_{NB}\big]\mathbf{y}. \tag{22}$$

Since $\mathbf{H}_B$ is nonsingular, (22) becomes

$$\mathbf{x}_B = \big[\mathbf{I}_m \quad \mathbf{H}_B^{-1}\mathbf{H}_{NB}\big]\mathbf{y}. \tag{23}$$

Given the condition that $\mathbf{d}_B^T\mathbf{H}_B^{-1}\mathbf{H}_{NB} - \mathbf{d}_{NB}^T \leq \mathbf{0}$, we obtain

$$\mathbf{d}_B^T\mathbf{x}_B = \big[\mathbf{d}_B^T\mathbf{I}_m \quad \mathbf{d}_B^T\mathbf{H}_B^{-1}\mathbf{H}_{NB}\big]\mathbf{y} \leq \big[\mathbf{d}_B^T \quad \mathbf{d}_{NB}^T\big]\mathbf{y} = \mathbf{d}^T\mathbf{y}. \tag{24}$$

Equation (24) proves that $\mathbf{d}_B^T\mathbf{H}_B^{-1}\mathbf{H}_{NB} - \mathbf{d}_{NB}^T \leq \mathbf{0}$ is an sufficient condition to a global minimum of Z. In addition, the solution $\mathbf{x}$ becomes the only optimal solution if $\mathbf{d}_B^T\mathbf{H}_B^{-1}\mathbf{H}_{NB} < \mathbf{d}_{NB}^T$, as for any other solution $\mathbf{y}$, $\mathbf{y} \neq \mathbf{x}$, there is $\mathbf{d}_B^T\mathbf{x}_B < \mathbf{d}^T\mathbf{y}$ according to (24). This completes the proof. □

It is possible that the procedure described in (17)-(19) generates a solution that meets the two requirements for global optimal, but the two requirements are not necessarily guaranteed. The first requirement that $\mathbf{H}_B^{-1}\mathbf{b} \geq \mathbf{0}$ is equivalent to stating that $\mathbf{x} \geq \mathbf{0}$ as $\mathbf{x}_B = \mathbf{H}_B^{-1}\mathbf{b}$. It is usually satisfied by formulating the problem so that $\mathbf{H}_B$ contains all the columns of $\mathbf{I}_m$ by introducing artificial variables [35]. The second requirement usually requires an iterative procedure of switching basic and nonbasic variables.

Denote z_j as the jth element in $\mathbf{d}_B^T\mathbf{H}_B^{-1}\mathbf{H}_{NB} - \mathbf{d}_{NB}^T$ and assume $z_j > 0$. Clearly, increasing the corresponding element (the jth nonbasic variable, denoted as x_j) in solution $\mathbf{x}$ should linearly reduce the objective function Z. On the other hand, the increase of the x_j is limited by the constraints that all the basic variables should be nonnegative. A switching of basic variable x_i and nonbasic variable x_j happens if a basic variable denoted as x_i reduces to zero (the rest of the basic variables are still nonnegative).

After the pair of basic and nonbasic variables is switched, reperform the procedure described in (17)-(19) and check if the second constraint holds. If it satisfies the requirement that $\mathbf{d}_B^T\mathbf{H}_B^{-1}\mathbf{H}_{NB} - \mathbf{d}_{NB}^T \leq \mathbf{0}$, a global optimal solution is reached. Otherwise, keep switching another pair of basic and nonbasic variables, until a global optimal solution appears.

A.2. Selection of Basic Variables

Before the simplex procedures, it is required to select a set of m basic variables to start. The selection should satisfy the first constraint that $\mathbf{H}_B^{-1}\mathbf{b} \geq \mathbf{0}$. Also, (17)-(19) are performed based on the assumption that $rank(\mathbf{H}) = m$. If $rank(\mathbf{H}) < m$, $\mathbf{H}_B$ becomes singular and does not have its inverse form, which is the case of this work using the EMD formulation (the rows of the constraint matrix described in (4) are linearly dependent). Identifying and deleting the redundant constraints is a possible approach; however, it is inefficient given the large number of constraints. A technique of using artificial variables is described in [35], which facilitates the selection of the m basic variables as well as solving the problem caused by redundant constraints.

Appendix B
Proof of (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

$$\frac{\partial Z}{\partial b_i} = lim_{\Delta b_i^* \to 0} \frac{k_i \Delta b_i^* + \sum_{j \neq i} k_j \Delta b_j}{\Delta b_i^*}, i = 1, \dots, m^{(C)}, \tag{26}$$

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

$$\Delta b_j = -\frac{b_j}{\sum_{j \neq i} b_j} \Delta b_i^*. \tag{27}$$

Substituting (27) into (26) results in

$$\frac{\partial Z}{\partial b_i} = k_i - \sum_{j \neq i} k_j \frac{b_j}{\sum_{j \neq i} b_j}, i = 1, \dots, m^{(C)}. \tag{28}$$

Acknowledgments

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

References

[1] P. Indyk and N. Thaper, "Fast Image Retrieval via Embeddings," *Proc. Third Int'l Workshop Statistical and Computational Theories of Vision*, 2003.

[2] K. Grauman and T. Darrell, "Fast Contour Matching Using Approximate Earth Mover's Distance," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. I-220-I-227, 2004.

[3] D. Forsyth, "A Novel Approach to Color Constancy," *Int'l J. Computer Vision*, vol. 5, no. 1, pp. 5-36, Aug. 1990.

[4] B. Funt and G. Finlayson, "Color Constant Color Indexing," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 5, pp. 522-529, May 1995.
[5] E. Land and J. McCann, "Lightness and Retinex Theory," *J. Optical Soc. of Am.,* vol. 61, no. 1, pp. 1-11, 1971.
[6] D. Freedman and M. Turek, "Illumination-Invariant Tracking via Graph Cuts," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 10-17, 2005.
[7] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 5, pp. 564-577, May 2003.
[8] G. Hager, M. Dewan, and C. Stewart, "Multiple Kernel Tracking with SSD," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. I-790-I-797, 2004.
[9] Y. Rubner, "Perceptual Metrics for Image Database Navigation," PhD dissertation, Stanford Univ., 1999.
[10] R. Collins, "Mean-Shift Blob Tracking through Scale Space," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 234-240, 2003.
[11] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical Model-Based Motion Estimation," *Proc. European Conf. Computer Vision,* pp. 237-252, 1992.
[12] S. Avidan, "Support Vector Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 8, pp. 1064-1072, Aug. 2004.
[13] S. Avidan, "Ensemble Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 2, pp. 261-271, Feb. 2007.
[14] R. Collins, Y. Liu, and M. Leordeanu, "Online Selection of Discriminative Tracking Features," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 10, pp. 1631-1643, Oct. 2005.
[15] G. Hager and P. Belhumeur, "Efficient Region Tracking with Parametric Models of Geometry and Illumination," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 10, pp. 1025-1039, Oct. 1998.
[16] M. Isard and A. Blake, "Condensation—Conditional Density Propagation for Visual Tracking," *Int'l J. Computer Vision,* vol. 29, no. 1, pp. 5-28, 1998.
[17] J. Shi and C. Tomasi, "Good Features to Track," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 593-600, 1994.
[18] H. Tao, H. Sawhney, and R. Kumar, "Object Tracking with Bayesian Estimation of Dynamic Layer Representations," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 1, pp. 75-89, Jan. 2002.
[19] M. Swain and D. Ballard, "Color Indexing," *Int'l J. Computer Vision,* vol. 7, no. 1, pp. 11-32, 1991.
[20] G. Bradski, "Computer Vision Face Tracking for Use in a Perceptual User Interface," *Proc. IEEE Workshop Applications of Computer Vision,* pp. 214-219, 1998.
[21] S. Birchfield and R. Sriram, "Spatiograms versus Histograms for Region-Based Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 1158-1163, 2005.
[22] Q. Zhao and H. Tao, "Object Tracking Using Color Correlogram," *Proc. IEEE Workshop Performance Evaluation of Tracking and Surveillance,* pp. 263-270, 2005.
[23] T. Cover and J. Thomas, *Elements of Information Theory.* John Wiley and Sons, 1991.
[24] J. Puzicha, T. Hofmann, and J. Buhmann, "Non-Parametric Similarity Measures for Unsupervised Texture Segmentation and Image Retrieval," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 267-272, 1997.
[25] W. Niblack et al., "Querying Images by Content, Using Color, Texture, and Shape," *Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases,* pp. 173-187, 1993.
[26] M. Werman, S. Peleg, and A. Rosenfeld, "A Distance Metric for Multi-Dimensional Histograms," *Computer, Vision, Graphics, and Image Processing,* vol. 32, pp. 328-336, 1985.
[27] T. Kailath, "The Divergence and Bhattacharyya Distance Measures in Signal Selection," *IEEE Trans. Comm. Technology,* vol. 15, no. 1, pp. 52-60, Feb. 1967.
[28] H. Ling and K. Okada, "An Efficient Earth Mover's Distance Algorithm for Robust Histogram Comparison," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 5, pp. 840-853, May 2007.
[29] S. Shirdhonkar and D. Jacobs, "Approximate Earth Mover's Distance in Linear Time," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, 2008.
[30] A. Adam, E. Rivlin, and I. Shimshoni, "Robust Fragments-Based Tracking Using the Integral Histogram," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 798-805, 2006.
[31] F. Hitchcock, "The Distribution of a Product from Several Sources to Numerous Localities," *J. Math. and Physics,* vol. 20, pp. 224-230, 1941.
[32] C. Papageorgiou and T. Poggio, "Trainable Pedestrian Detection," *Proc. IEEE Int'l Conf. Image Processing,* vol. 4, pp. 35-39, 1999.
[33] http://www.cse.ohio-state.edu/otcbvs-bench/, 2009.
[34] http://www.cvg.rdg.ac.uk/slides/pets.html, 2009.
[35] G. Dantzig and M. Thapa, *Linear Programming: 1: Introduction.* Springer, Jan. 1997.

Qi Zhao received the BS degree in computer science from Zhejiang University, China, in 2004. She is currently working toward the PhD degree in the Computer Engineering Department at the University of California, Santa Cruz. During the summers of 2007 and 2008, she was a research intern with Microsoft Research, Redmond, Washington, and Google Research, New York City, respectively. Her research interests include computer vision, pattern recognition, machine learning, computational neuroscience, multimedia systems, and bio-signal processing and biological image analysis. She is a student member of the IEEE.

Zhi Yang received the BS degree in electrical engineering from Zhejiang University, China, in 2004. Since 2005, he has been with the University of California, Santa Cruz, where he received the MS degree in electrical engineering in 2007 and where he is now working toward the PhD degree. His research interests include mathematical modeling, biophysics, biomedical signal processing, analog circuit design, wireless power and data telemetry, neural engineering, neural prosthesis, computation neuroscience, computer vision, and machine learning. He is a student member of the IEEE.

Hai Tao received the BS and MS degrees in automation from Tsinghua University in 1991 and 1993, respectively, the MS degree in electrical engineering from Mississippi State University in 1995, and the PhD degree in electrical engineering from the University of Illinois at Urbana-Champaign in 1999. From 1999 to 2001, he was a member of the technical staff in the Vision Technology Laboratory at Sarnoff Corp., New Jersey. Since July 2001, he has been with the Department of Computer Engineering at the University of California, Santa Cruz, where he is now an associate professor. His research interests include image and video processing, computer vision, vision-based computer graphics, and human-computer interaction. He has published more than 50 technical papers and holds 12 US patents. In 2004, he received the US National Science Foundation Faculty Early Career Development (CAREER) Award. He is a senior member of the IEEE and currently serves as an associate editor for the journals *Machine Vision and Applications* and *Pattern Recognition*.