Best-Buddies Similarity—Robust Template Matching Using Mutual Nearest Neighbors

Shaul Oron, Tali Dekel, Tianfan Xue, William T. Freeman, and Shai Avidan

Abstract—We propose a novel method for template matching in unconstrained environments. Its essence is the Best-Buddies Similarity (BBS), a useful, robust, and parameter-free similarity measure between two sets of points. BBS is based on counting the number of Best-Buddies Pairs (BBPs)—pairs of points in source and target sets that are mutual nearest neighbours, i.e., each point is the nearest neighbour of the other. BBS has several key features that make it robust against complex geometric deformations and high levels of outliers, such as those arising from background clutter and occlusions. We study these properties, provide a statistical analysis that justifies them, and demonstrate the consistent success of BBS on a challenging real-world dataset while using different types of features.

Index Terms—Best buddies, mutual nearest neighbors, template matching, point set similarity, non-rigid matching

1 Introduction

Finding a template patch in a target image is a core component in a variety of computer vision applications such as object detection, tracking, image stitching and 3D reconstruction. In many real-world scenarios, the template—a bounding box containing a region of interest in the source image—undergoes complex deformations in the target image: the background can change and the object may undergo nonrigid deformations and partial occlusions.

Template matching methods have been used with great success over the years but they still suffer from a number of drawbacks. Typically, all pixels (or features) within the template and a candidate window in the target image are taken into account when measuring their similarity. This is undesirable in some cases, for example, when the object of interest is partially occluded or when the background behind it changes between the template and the target image (see Fig. 1). In such cases, the dissimilarities between pixels from different backgrounds may be arbitrary, and accounting for them may lead to false detections of the template (see Fig. 1b).

In addition, many template matching methods assume a specific parametric deformation model between the template and the target image (e.g., rigid, affine transformation, etc.). This limits the type of scenes that can be handled, and may require estimating a large number of parameters when complex deformations are considered.

In order to address these challenges, we introduce a novel similarity measure termed *Best-Buddies Similarity (BBS)*, and show how it can be applied successfully to template matching *in the wild*. BBS measures the similarity between two point sets in $\mathbb{R}^d$. A key feature of this measure is that it relies only on a subset (usually small) of pairs of points—the *Best-Buddies Pairs (BBPs)*. A pair of points is considered a BBP if the points are mutual nearest neighbours, i.e., each point is the nearest neighbour of the other in the corresponding point set. BBS is then taken to be the fraction of BBPs out of all the points in the set. To apply BBS for template matching, we represent both the template patch and candidate query patches as point sets in $\mathbb{R}^d$ and directly measure the BBS between these point sets.

Albeit simple, this measure turns out to have important and non-trivial properties. Because BBS counts only the pairs of points that are best buddies, it is robust to significant amounts of outliers. Another, less obvious, property is that the BBS between two point sets is maximal when the points are drawn from the same distribution, and drops sharply as the distributions diverge. In other words, if two points are BBP, they were likely drawn from the same distribution. We provide a statistical formulation of this observation, and analyze it numerically in the 1D case for point sets drawn from distinct Gaussian distributions (often used as a simplified model for natural images).

In addition, we prove a connection between BBS and the Chi-Square (χ^2) distance, in the 1D case. χ^2 is typically used in computer vision to measure distance between histograms. Specifically, we show that for sufficiently large sets, BBS converges to the χ^2 distance between two distributions. However, unlike χ^2, computing BBS is done directly on the raw data without the need to construct histograms. This in turn alleviates the need to choose the histogram bin size. Moreover, BBS is able to work with high dimensional representation, such as deep features, for which constructing histograms is not tractable.

We apply BBS to template matching using a sliding window over a query image. Both the template and each of the candidate image regions are represented as point sets in a joint location-appearance space. We use normalized patch

- *S. Oron and S. Avidan are with the Department of Electrical Engineering, Tel-Aviv University, Tel Aviv-Yafo 69102, Israel. E-mail: shauloro@post.tau.ac.il, avidan@eng.tau.ac.il.*
- *T. Dekel, T. Xue, and W.T. Freeman are with the MIT Computer Science and Artificial Intelligence Lab, Google, Cambridge, MA 02139. E-mail: tdekel@google.com, {tfxue, billf}@mit.edu.*

Manuscript received 6 June 2016; revised 26 May 2017; accepted 30 July 2017. Date of publication 8 Aug. 2017; date of current version 11 July 2018. (Corresponding author: Shaul Oron.) Recommended for acceptance by V. Ferrari. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TPAMI.2017.2737424

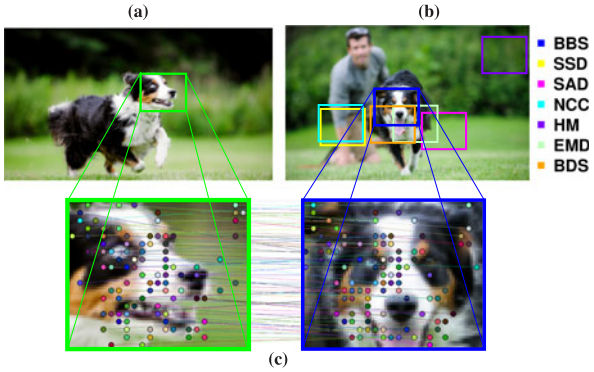


Fig. 1. *Best-Buddies Similarity (BBS) for template matching:* (a) The template, marked in green, contains an object of interest against a background. (b) The object in the target image undergoes complex deformation (background clutter and large geometric deformation); the detection results using different similarity measures are marked on the image (see legend); our result is marked in blue. (c) The Best-Buddies Pairs (BBPs) between the template and the detected region are mostly found the object of interest and not on the background; each BBP is connected by a line and marked in a unique color.

coordinates as our spatial descriptor, and experiment with both color as well as deep features for capturing appearance information (although BBS is not restricted to these specific choices). Once the template and candidate windows are converted to point-sets BBS is used to measure the similarity between them. The aforementioned properties of BBS now readily apply to template matching. That is, pixels on the object of interest in both the template and the candidate patch can be thought of as originating from the same underlying distribution. These pixels in the template are likely to find best buddies in the candidate patch, and hence would be considered as inliers. In contrast, pixels that come from different distributions, e.g., pixels from different backgrounds, are less likely to find best buddies, and hence would be considered outliers (see Fig. 1c). Given this important property, BBS bypasses the need to explicitly model the underlying object appearance and deformation.

To summarize, the main contributions of this paper are: (a) introducing BBS—a useful, robust, parameter-free measure for template matching in unconstrained environments, (b) analysis providing theoretical justification of its key features and linking BBS with the χ^2 distance, and (c) extensive evaluation on challenging real data, using different feature representations, and comparing BBS to a number of commonly used template matching methods. A preliminary version of this paper appeared in CVPR 2015 [1].

2 Related Work

Template matching algorithms depend heavily on the similarity measure used to match the template and a candidate window in the target image. Various similarity measures have been used for this purpose. The most popular are the Sum of Squared Differences (SSD), Sum of Absolute Differences (SAD) and Normalized Cross-Correlation (NCC), mostly due to their computational efficiency [2]. Different variants of these measures have been proposed to deal with illumination changes and noise [3], [4], [5].

Another family of measures is composed of robust error functions such as M-estimators [6], [7] or Hamming-based distance [8], [9], which are less affected by additive noise and 'salt and paper' outliers than cross correlation related methods. However, all the methods mentioned so far assume a strict rigid geometric deformation (only translation) between the template and the target image, as they penalize pixel-wise differences at corresponding positions in the template and the query region.

A number of methods extended template matching to deal with parametric transformations (e.g., [10], [11]). Korman et al. [12] introduced a template matching algorithm under 2D affine transformation that guarantees an approximation to the globally optimal solution. Likewise, Tian and Narasimhan [13] find a globally optimal estimation of nonrigid image distortions. However, these methods assume a one-to-one mapping between the template and the query region for the underlying transformation. Thus, they are prone to errors in the presence of many outliers, such as those caused by occlusions and background clutter. Furthermore, these methods assume a parametric model for the distortion geometry, which is not required in the case of BBS.

Measuring the similarity between color histograms, known as Histogram Matching (HM), offers a non-parametric technique for dealing with deformations and is commonly used in visual tracking [14], [15]. However, HM completely disregards geometry, which is a powerful cue. Other tracking methods have been proposed to deal with cluttered environments and partial occlusions [16], [17]. But unlike tracking, we are interested in detection in a single image, which lacks the redundant temporal information given in videos.

Olson [18] formulated template matching in terms of maximum likelihood estimation, where an image is represented in a 3D location-intensity space. Taking this approach one step further, Oron et al. [19] use $xyRGB$ space and reduced template matching to measuring the EMD [20] between two point sets. Unlike EMD, BBS does not require $1:1$ matching (does not have to account for all the data when matching), which makes it more robust to outliers.

In the context of image matching, another widely used measure is the Hausdorff distance [21]. To deal with occlusions or degradations, Huttenlocher et al. [21] proposed a fractional Hausdorff distance in which the Kth farthest point is taken instead of the most farthest one. Yet, this measure highly depends on K that needs to be tuned.

The BBS is a bi-directional measure. The importance of such two-side agreement has been demonstrated by the Bidirectional similarity (BDS) in [22] for visual summarization. Specifically, the BDS was used as a similarity measure between two images, where an image is represented by a set of patches. The BDS sums over the distances between each patch in one image to its nearest neighbor in the other image, and vice versa. In contrast, the BBS is based on a *count* of the Best-Buddies Pairs–pairs of points in source and target sets that are mutual nearest neighbours (each point is the nearest neighbour of the other), and makes only implicit use of their actual distance.

The concept of mutual nearest neighbours criterion is not new and has been used in various tasks in computer vision. For example, it has been used to select reliable matches between keypoints in stereo images [23], affine registration [24], and image matching [25]. Similar ideas are also

proposed in classification of images [26] and natural language data [27]. All these methods use the mutual nearest neighbours only as a post processing step as a mean to filter out outliers in the data. In contrast, our work makes a direct use of mutual nearest neighbours as the core similarity measure. In image retrieval, a number of methods have been proposed to estimate a metric that maximizes the number of images with mutual nearest neighbours in a given image collection [28], [29], [30]. Here too, mutual nearest neighbours have not been used directly as a similarity measure.

On the theoretical side, we show a connection between BBS and the Chi-Square (χ^2) distance used as a distance measure between distributions (or histograms). Chi-Square distance comes from the χ^2 test-statistic [31] where it is used to test the fit between a distribution and observed frequencies. χ^2 was successfully applied to a wide range of computer vision tasks such as texture and shape classification [32], [33], local descriptors matching [34], and boundary detection [35] to name a few.

It is worth mentioning that the term *Best Buddies* was used by Pomeranz et al. [36] in the context of solving jigsaw puzzles. Specifically, they used a metric similar to ours in order to determine if a pair of pieces are compatible with each other.

3 Best-Buddies Similarity

Our goal is to match a template to a given image, in the presence of high levels of outliers (i.e., background clutter, occlusions) and nonrigid deformation of the object of interest. We follow the traditional sliding window approach and compute the Best-Buddies Similarity between the template and every window (of the size of the template) in the image. In the following, we give a general definition of BBS and demonstrate its key features via simple intuitive toy examples. We then statistically analyze these features in Section 4.

General Definition. *BBS measures the similarity between two sets of points $P = \{p_i\}_{i=1}^{N}$ and $Q = \{q_i\}_{i=1}^{N}$, where $p_i, q_i \in \mathbb{R}^d$ (throughout the paper we assume the size of P and Q is equal). The BBS is the fraction of* Best-Buddies Pairs *(BBPs) between the two sets. Specifically, a pair of points $\{p_i \in P, q_j \in Q\}$ is a BBP if p_i is the nearest neighbor of q_j in the set Q, and vice versa. Formally,*

$$bb(p_i, q_j, P, Q) = \begin{cases} 1 & \mathrm{NN}(p_i, Q) = q_j \wedge \mathrm{NN}(q_j, P) = p_i \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where, $\mathrm{NN}(p_i, Q) = argmin_{q \in Q} d(p_i, q)$, and $d(p_i, q)$ is some distance measure. The BBS between the point sets P and Q is given by

$$\mathrm{BBS}(P, Q) = \frac{1}{N} \cdot \sum_{i=1}^{N} \sum_{j=1}^{N} bb(p_i, q_j, P, Q). \tag{2}$$

The key properties of the BBS are: (i) it relies only on a (usually small) subset of matches i.e., pairs of points that are BBPs, whereas the rest are considered as outliers. (ii) BBS finds the bi-directional inliers in the data without any prior knowledge on the data or its underlying deformation. (iii) BBS uses *rank*, i.e., it counts the number of BBPs, rather than using the actual distance values.

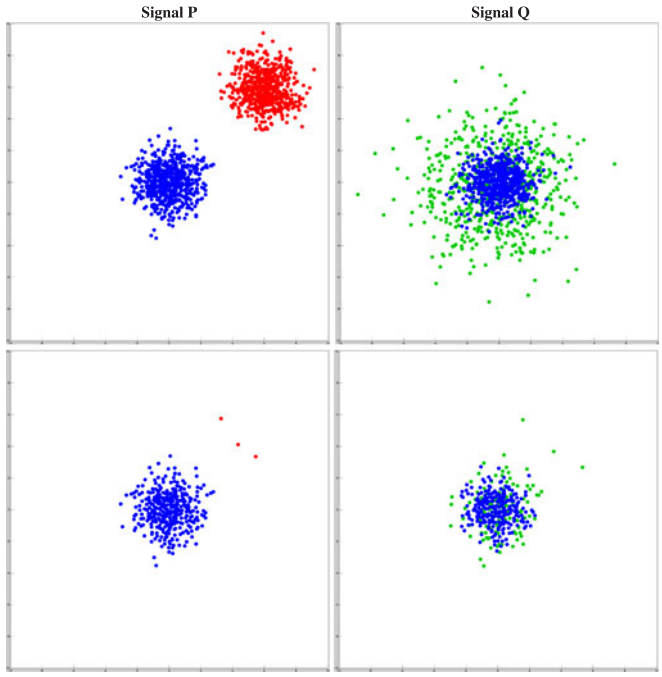


Fig. 2. *Best-Buddies Pairs (BBPs) between 2D Gaussian signals:* First row, Signal P consists of "foreground" points drawn from a normal distribution, $N(\mu_1, \sigma_1)$, marked in blue; and "background" points drawn from $N(\mu_2, \sigma_2)$, marked in red. Similarly, the points in the second signal Q are drawn from the same distribution $N(\mu_1, \sigma_1)$, and a different background distribution $N(\mu_3, \sigma_3)$. The color of points is for illustration only, i.e., BBS does not know which point belongs to which distribution. Second row, only the BBPs between the two signals which are mostly found between foreground points.

To understand why these properties are useful, let us consider a simple 2D case of two point sets P and Q. The set P consist of 2D points drawn from two different normal distributions, $N(\mu_1, \Sigma_1)$, and $N(\mu_2, \Sigma_2)$. Similarly, the points in Q are drawn from the same distribution $N(\mu_1, \Sigma_1)$, and a different distribution $N(\mu_3, \Sigma_3)$ (see first row in Fig. 2). The distribution $N(\mu_1, \Sigma_1)$ can be treated as a *foreground* model, whereas $N(\mu_2, \Sigma_2)$ and $N(\mu_3, \Sigma_3)$ are two different *background* models. As can be seen in Fig. 2, the BBPs are mostly found between the foreground points in P and Q. For set P, where the foreground and background points are well separated, 95 percent of the BBPs are foreground points. For set Q, despite the significant overlap between foreground and background, 60 percent of the BBPs are foreground points.

This example demonstrates the robustness of BBS to high levels of outliers in the data. BBS captures the foreground points and does not force the background points to match. In doing so, BBS sidesteps the need to model the background/foreground parametrically or have a prior knowledge of their underlying distributions. This shows that a pair of points $\{p, q\}$ is more likely to be BBP if p and q are drawn from the same distribution. We formally prove this general argument for the 1D case in Section 4. With this observations in hand, we continue with the use of BBS for template matching.

3.1 BBS for Template Matching

To apply BBS to template matching, one needs to convert each image patch to a point set in $\mathbb{R}^d$. BBS, as formulated in Eq. (2), can be computed for any arbitrary feature space and

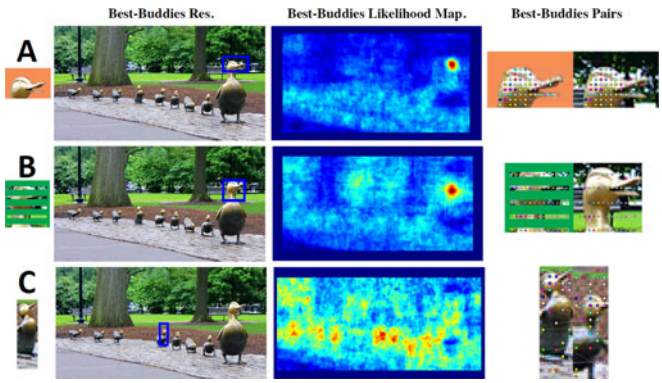


Fig. 3. *BBS template matching results.* Three toys examples are shown: (A) cluttered background, (B) occlusions, (C) nonrigid deformation. The template (first column) is detected in the target image (second column) using the BBS; the results using BBS are marked in a blue. The likelihood maps (third column) show well-localized distinct modes. The BBPs are shown in last column. See text for more details.

for any distance measure between point pairs. In this paper we focus on a joint location-appearance space which was shown to be useful for template matching [19]. Specifically, for the location we use normalized patch coordinates, and for the appearance, we consider two specific representations: (i) color features, and (ii) using deep features taken from a pre-trained neural net. Using such deep features is motivated by recent success in applying features taken from deep neural nets to different applications [37], [38]. A detailed description of each of these feature spaces is given in Section 5.

Following the intuition presented in the 2D Gaussian example (see Fig. 2), the use of BBS for template matching allows us to overcome several significant challenges such as background clutter, occlusions, and nonrigid deformation of the object. This is demonstrated in three synthetic examples shown in Fig. 3. The templates A and B include the object of interest in a cluttered background, and under occlusions, respectively. In both cases the templates are successfully matched to the image despite the high level of outliers. As can be seen, the BBPs are found only on the object of interest, and the BBS likelihood maps have a distinct mode around the true location of the template. In the third example, the template C is taken to be a bounding box around the forth duck in the original image, which is removed from the searched image using inpainting techniques. In this case, BBS matches the template to the fifth duck, which can be seen as a nonrigid deformed version of the template. Note that the BBS does not aim to solve the pixel correspondence. In fact, the BBPs are not necessarily semantically correct (see third row in Fig. 3), but rather pairs of points that likely originated from the same distribution. This property, which we next formally analyze, helps us deal with complex visual and geometric deformations in the presence of outliers.

4 Analysis

So far, we have shown some empirical evidence demonstrating that the BBS is robust to outliers, and results in well-localized modes. In what follows, we give a statistical analysis that justifies these properties, and explains why using the count of the BBP is a good similarity measure. Additionally, we show that for sufficiently large sets BBS converges to the well known χ^2 distance, which provides additional insight into the way BBS handles outliers.

4.1 Expected Value of BBS

We begin with a simple mathematical model in 1D, in which an "image" patch is modeled as a set of points drawn from a general distribution. Using this model, we derive the expectation of BBS between two sets of points, drawn from two given distributions $f_P(p)$ and $f_Q(q)$, respectively. We then analyze numerically the case in which $f_P(p)$, and $f_Q(q)$ are two different normal distributions. Finally, we relate these results to the multi-dimentional case. We show that the BBS distinctively captures points that are drawn from similar distributions. That is, we prove that the likelihood of a pair of points being BBP, and hence the expectation of the BBS, is maximal when the points in both sets are drawn from the same distribution, and drops sharply as the distance between the two normal distributions increases.

4.1.1 One-Dimensional Case

Following Eq. (2), the expectation BBS(P, Q), over all possible samples of P and Q is given by

$$E[\mathrm{BBS}(P,Q)] = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} E[bb_{i,j}(P,Q)], \tag{3}$$

where $bb_{i,j}(P,Q)$ is defined in Eq. (1). We continue with computing the expectation of a pair of points to be BBP, over all possible samples of P and Q, denoted by E_{BBP}. That is

$$E_{\mathrm{BBP}} = \iint_{P,Q} bb_{i,j}(P,Q)\Pr\{P\}\Pr\{Q\}dPdQ, \tag{4}$$

This is a multivariate integral over all points in P and Q. However, assuming each point is independent of the others this integral can be simplified as follows.

Claim.

$$E_{\mathrm{BBP}} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(F_Q(p^-)+1-F_Q(p^+)\right)^{N-1}\cdot \left(F_P(q^-)+1-F_P(q^+)\right)^{N-1} f_P(p)f_Q(q)dpdq, \tag{5}$$

where, $F_P(x)$, and $F_Q(x)$ denote the CDFs of P and Q, respectively. That is, $F_P(x) = \Pr\{p \le x\}$. And, $p^- = p - d(p,q)$, $p^+ = p + d(p,q)$, and q^+, q^- are similarly defined.

Proof. Due to the independence between the points, the integral in Eq. (4) can be decoupled as follows:

$$E_{\mathrm{BBP}} = \int_{p_1}\cdots\int_{p_N}\int_{q_1}\cdots\int_{q_N} bb_{i,j}(P,Q)\prod_{k=1}^{N} f_P(p_k)\prod_{l=1}^{N} f_Q(q_l)dPdQ. \tag{6}$$

With abuse of notation, we use $dP = dp_1 \cdot dp_2 \cdots dp_N$, and $dQ = dq_1 \cdot dq_2 \cdots dq_N$. Let us consider the function $bb_{i,j}(P,Q)$ for a given realization of P and Q. By definition, this indicator function equals 1 when p_i and q_j are nearest neighbors of each other, and zero otherwise. This can be expressed in terms of the distance between the points as follows:

$$bb_{i,j}(P,Q) = \prod_{k\neq i,k=1}^{N}\mathbb{I}[d(p_k,q_j) > d(p_i,q_j)]\prod_{l\neq j,l=1}^{N}\mathbb{I}[d(q_l,p_i) > d(p_i,q_j)], \tag{7}$$

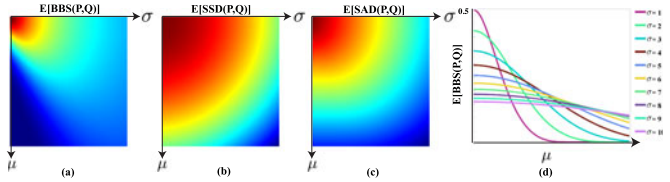


Fig. 4. *The expectation of BBS in the 1D Gaussian case:* Two point sets, P and Q, are generated by sampling points from $N(0,1)$, and $N(\mu,\sigma)$, respectively. (a) the approximated expectation of BBS(P,Q) as a function of σ (*x*-axis), and μ (*y*-axis). (b)-(c) the expectation of SSD(P,Q), and SAD(P,Q), respectively. (d) the expectation of BBS as a function of μ plotted for different σ.

where $\mathbb{I}$ is an indicator function. It follows that for a given value of p_i and q_j, the contribution of p_k to the integral in Eq. (6) can be decoupled. Specifically, we define

$$Cp_k = \int_{-\infty}^{\infty} \mathbb{I}[d(p_k, q_j) > d(p_i, q_j)] f_P(p_k) dp_k. \tag{8}$$

Assuming $d(p,q) = \sqrt{(p-q)^2} = |p-q|$, the latter can be written as

$$Cp_k = \int_{-\infty}^{\infty} \mathbb{I}[p_k < q_j^- \vee p_k > q_j^+] f_P(p_k) dp_k, \tag{9}$$

where $q_j^- = q_j - d(p_i, q_j)$, $q_j^+ = q_j + d(p_i, q_j)$. Since $q_j^- < q_j^+$, it can be easily shown that Cp_k can be expressed in terms of $F_P(x)$, the CDF of P

$$Cp_k = F_P(q_j^-) + 1 - F_P(q_j^+). \tag{10}$$

The same derivation hold for computing Cq_l, the contribution of q_l to the integral in Eq. (6), given p_i, and q_j. That is,

$$Cq_l = F_Q(p_i^-) + 1 - F_Q(p_i^+), \tag{11}$$

where p_i^-, p_i^+ are similarly defined and $F_Q(x)$ is the CDF of Q. Note that Cp_k and Cq_l depends only on p_i and q_j and on the underlying distributions. Therefore, Eq. (6) results in

$$\begin{aligned} E_{\text{BBP}} &= \iint_{p_i,q_j} dp_i dq_j f_P(p_i) f_Q(q_j) \prod_{k=1,k\neq i}^{N} Cp_k \prod_{l=1,l\neq j}^{N} Cq_l \\ &= \iint_{p_i,q_j} dp_i dq_j f_P(p_i) f_Q(q_j) Cp_k^{N-1} Cq_l^{N-1}. \end{aligned} \tag{12}$$

Substituting the expressions for Cp_k and Cq_l in Eq. (12), and omitting the subscripts i, j for simplicity, result in Eq. (5), which completes the proof.

In general, the integral in Eq. (5) does not have a closed form solution, but it can be solved numerically for selected underlying distributions. To this end, we proceed with Gaussian distributions, which are often used as simple statistical models of image patches. We then use Monte-Carlo integration to approximate E_{BBP} for discrete choices of parameters μ and σ of Q in the range of [0, 10] while fixing the distribution of P to have $\mu = 0, \sigma = 1$. We also fixed the number of points to $N = 100$. The resulting approximation for E_{BBP} as a function of the parameters μ, σ is shown in Fig. 4, on the left. As can be seen, E_{BBP} is the highest at $\mu = 0, \sigma = 1$, i.e., when the points are drawn from the same distribution, and drops rapidly as the the underlying distribution of Q deviates from $N(0,1)$.

Note that E_{BBP} does not depends on p and q (because of the integration, see Eq. (5). Hence, the expected value of the BBS between the sets (Eq. (3)) is given by

$$E[\text{BBS}(P,Q)] = N \cdot E_{\text{BBP}}. \tag{13}$$

We can compare the BBS to the expectation of SSD, and SAD. The expectation of the SSD has a closed form solution given by

$$E[\text{SSD(P, Q)}] = \iint_{-\infty}^{\infty} (p-q)^2 f_P(p) f_Q(q|k) dp dq = 1 + \mu^2 + \sigma^2. \tag{14}$$

Replacing $(p-q)^2$ with $|p-q|$ results in the expression of the SAD. In this case, the expected value reduces to the expectation of the Half-Normal distribution and is given by

$$E[\text{SAD(P,Q)}] = \frac{1}{\sqrt{2\pi}} \sigma_K \exp^{-\mu^2/(2\sigma^2)} + \mu(1 - 2f_P(-\mu/\sigma)). \tag{15}$$

Figs. 4b and 4c shows the maps of the expected values for $1 - \text{SSD}_n(P,Q)$, and $1 - \text{SAD}_n(P,Q)$, where $\text{SSD}_n, \text{SAD}_n$ are the expectation of SSD and SAD, normalized to the range of [0,1]. As can be seen, the SSD and SAD results in a much wider spread around their mode. Thus, we have shown that the likelihood of a pair of points to be a BBP (and hence the expectation of the BBS) is the highest when P and Q are drawn from the same distribution and drops sharply as the distance between the distributions increases. This makes the BBS a robust and distinctive measure that results in well-localized modes.

4.1.2 Multi-Dimensional Case

With the result of the 1D case in hand, we can bound the expectation of BBS when P and Q are sets of multi-dimensional points, i.e., $p_i, q_j \in R^d$.

If the d-dimensions are uncorrelated (i.e., the covariance matrices are diagonals in the Gaussian case), a sufficient (but not necessary) condition for a pair of points to be BBP is that the point would be BBP in each of the dimensions. In this case, the analysis can be done for each dimension independently similar to what was done in Eq. (5). The expectation of the BBS in the multi-dimensional case is then bounded by the product of the expectations in each of the dimensions. That is

$$E_{\text{BBS}} \geq \prod_{i=1}^{d} E_{\text{BBS}}^i, \tag{16}$$

where E_{BBS}^i denote the expectation of BBS in the ith dimension. This means that the BBS is expected to be more distinctive, i.e., to drop faster as d increases. Note that if a pair of points is not a BBP in one of the dimensions, it does not necessarily imply that the multi-dimentional pair is not BBP. Thus, this condition is sufficient but not necessary.

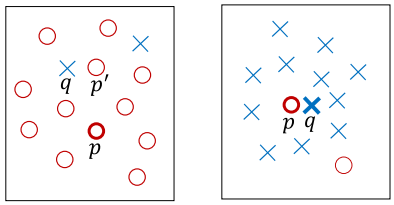


Fig. 5. *Finding a Best-Buddy:* We illustrate how the underlying density functions affect the probability that a point p (bold red circle) has a best buddy. (a) Points from set P (red circles) are dense but points from set Q (blue cross) are sparse. Although q is the nearest neighbor of p in Q, p is not the nearest neighbor of q in P (p' is closer). (b) Points from set Q are dense and points from set P are sparse. In this case, p and q are best buddies, as p is the closest point to q.

4.2 BBS and Chi-Square

Chi-Square (χ^2) is often used to measure the distance between histograms of two sets of features. For example, in face recognition, χ^2 is used to measure the similarity between local binary patterns (LBP) of two faces [39], and it achieves superior performance relative to other distance measures. In this section, we prove a connection between this well known statistical distance measure and BBS in the 1D case. Specifically, we show that, for sufficiently large point sets, BBS converges to the χ^2 distance.

We assume, as before, that point sets P and Q are drawn i.i.d. from 1D distribution functions $f_P(p)$ and $f_Q(q)$ respectively. We begin by considering the following lemma:

Lemma 1. *Let $Pr[bb(p_i = p; P, Q)]$ be the probability that a point $p_i = p \in P$ has a best buddy in Q. Then when the limit (w.r.t. the number of points) of that probability is given by*

$$\lim_{N \to +\infty} Pr[bb(p_i = p; P, Q)] = \frac{f_Q(p)}{f_P(p) + f_Q(p)}, \tag{17}$$

The proof of this lemma in given in Appendix A. Intuitively, if there are many points from P in the vicinity of point p, but only few points from Q, i.e., $f_P(p)$ is large but $f_Q(p)$ is small. It is then hard to find a best buddy in Q for p, as illustrated in Fig. 5a. Conversely, if there are few points from P in the vicinity of p but many points from Q, i.e., $f_P(p)$ is small and $f_Q(p)$ is large. In that case, it is easy for p to find a best buddy, as illustrated in Fig. 5b.

A synthetic example illustrating Lemma 1 is shown in Fig. 6. We consider two point sets P and Q, each is sampled from a different Gaussian mixture (red and blue in (Fig. 6a). Each mixture model consists of two modes. We then empirically calculate the probability that a certain point $p_i \in P$ has a best buddy in set Q for different set sizes, ranging from 10 to 10,000 points, Fig. 6b. As the sets size increases, the empirical probability converges to the analytical value given by Lemma 1, marked by the dashed black line. Note how the results agree with our intuition. For example, at $p = 0$, $f_P(p)$ is very large but $f_Q(p)$ is almost 0, such that $Pr[bbs(p_i; P, Q)]$ is almost 0. At $p = 5$, however, $f_P(p)$ is very small and $f_Q(p)$ is almost 0, so $Pr[bbs(p_i; P, Q)]$ is almost 1.

Lemma 1 assumes the value of the point p_i is fixed. However, we need to consider that p_i itself is also sampled from the distribution $f_P(p)$, in which case the probability this point has a best buddy is

$$\begin{aligned} &Pr[bb(p_i; P, Q)] \\ &= \int_{p=-M}^{M} f_P(p) \cdot Pr(p_i = p; P, Q) dp. = \int_{p=-M}^{M} \frac{f_Q(p) f_P(p)}{f_P(p) + f_Q(p)} dp. \end{aligned} \tag{18}$$

Where we assume both density functions are defined on the closed interval $[-M, M]$.

With Lemma 1 in hand, we are now ready to proof that BBS converges to Chi-Square,

Theorem 1. *Suppose both density functions are defined on a close interval $[-M, M]$, non-zero and Lipschitz continuous.*[1] *That is,*

1) $\forall p, q, f_P(p) \neq 0, f_Q(q) \neq 0$
2) $\exists A > 0, \forall p, q, h, \ s.t. \ |f_P(p+h) - f_P(p)| < A|h|$ *and* $|f_Q(q+h) - f_Q(q)| < A|h|$,

then we have,

$$\begin{aligned} \lim_{N \to +\infty} E[BBS(P, Q)] &= \int_{p=-M}^{M} \frac{f_P(p) f_Q(p)}{f_P(p) + f_Q(p)} dp \\ &= \frac{1}{2} - \frac{1}{4} \chi^2(f_p, f_q), \end{aligned} \tag{19}$$

where $\chi^2(f_p, f_q)$ is the Chi-Square distance between two distributions.

To see why this theorem holds, consider the BBS measure between two sets, P and Q. When the two sets have the same size, the BBS measure equals to the fraction of points in P that have a best buddy, that is $BBS(P, Q) = \frac{1}{N} \sum_{i=1}^{N} bbs(p_i; P, Q)$. Taking expectation on both sides of the equation, we get

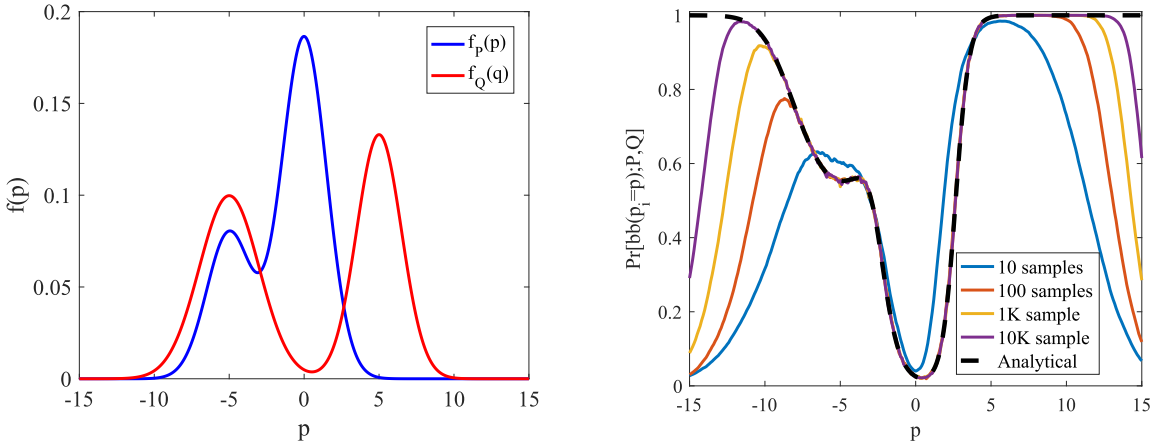
$$\begin{aligned} E[BBS(P, Q)] &= \frac{1}{N} \sum_{i=1}^{N} E[bbs(p_i; P, Q)] \\ &= \frac{1}{N} \cdot N \cdot E[bbs(p_i; P, Q)] \\ &= \int_{p=-M}^{M} \frac{f_Q(p) f_P(p)}{f_P(p) + f_Q(p)} dp. \end{aligned} \tag{20}$$

Where for the last equality we used lemma 1. This completes the proof of Theorem 1.

The theorem helps illustrate why BBS is robust to outliers. To see this, consider the signals in Fig. 6a. As can be seen f_P and f_Q are both Gaussian mixtures. Let us assume that the Gaussian with mean -5 represents the foreground (in both signals), i.e., $\mu_{fg} = -5$, and that the second Gaussian in each mixture represents the background, i.e., $\mu_{bg1} = 0$ and $\mu_{bg2} = 5$. Note how, $f_P(p)$ is very close to zero around μ_{bg2} and similarly $f_Q(q)$ is very close to zero around μ_{bg1}. This means that the background distributions will make very little contribution to the χ^2 distance, as the numerator $f_P(p) f_Q(q)$ of Eq. (19) is very close to 0 in both cases.

We note that using BBS has several advantages compared to using χ^2. One such advantage is that BBS does not

1. Note that most of density functions, like the density function of a Gaussian distribution, are non-zero and Lipschitz continuous in their domain.



(a) Density functions of two Gaussian mixtures

(b) The probability that a point in P has a best buddy

Fig. 6. *Illustrating Lemma 1:* Point sets P and Q are sampled iid from the two Gaussian mixtures shown in (a). The probability that a point in set P has a best buddy in set Q is empirically computed for different set sizes (b). When the size of the sets increase, the empirical probability converges to the analytical solution in Lemma 1 (dashed black line).

require binning data into histograms. It is not trivial to set the bin size, as it depends on the distribution of the features. A second advantage is the ability to use high dimensional feature spaces. The computational complexity and amount of data needed for generating histograms quickly explodes when the feature dimension becomes large. On the contrary, the nearest neighbor algorithm used by BBS can easily scale to high-dimensional features, like Deep features.

5 Feature Spaces

A joint spatial-appearance representation is used in order to convert both template and candidate windows into point sets. For the spatial component, normalized xy coordinates within the windows are used. For the appearance descriptor, we experiment with color features as well as deep features.

Color features. For building our color feature we break the template and candidate windows into $k \times k$ distinct patches. Each such $k \times k$ patch is represented by its $3 \cdot k^2$ color channel values and xy location of the central pixel, relative to the patch coordinate system. For all our toy examples and qualitative experiments RGB color space is used. However, for our quantitative evaluation HSV was used as it was found to produce better results. Both spatial and appearance channels were normalized to the range $[0, 1]$. The point-wise distance measure used with our color features is

$$d(p_i, q_j) = ||p_i^{(A)} - q_j^{(A)}||_2^2 + \lambda ||p_i^{(L)} - q_j^{(L)}||_2^2, \qquad (21)$$

where superscripts A and L denote the appearance and location descriptor, respectively. The parameter $\lambda = 0.25$ was chosen empirically (see Fig. 11) and was fixed in all of our experiments.

Deep features. As our deep feature [40] descriptor, we used the VGG-Deep-Net [41] pretrained on ImageNet. Specifically, we extract features from two layers of the network, conv 1_2 (64 features) and conv 3_4 (256 features). The feature maps from conv 1_2 are down-sampled twice, using max-pooling, to reach the size of the conv 3_4 which is down-sampled by a factor of $1/4$ with respect to the original image. In this case we treat every pixel in the down-sampled feature maps as a point. Each such point is represented by its xy location in the down-sampled window and its appearance is given by the 320 feature channels.

We found it is important to normalize the features prior to computing the point-wise distances. Therefore each feature channel is independently normalized to have zero mean and unit variance over the window.

The deep feature has much higher dimensionality than our color feature and we found the cosine distance works better in this case. Therefore, our point-wise distance when using deep features is

$$d(p_i, q_j) = < p_i^{(A)}, q_j^{(A)} > + exp(-\lambda ||p_i^{(L)} - q_j^{(L)}||_2^2), \qquad (22)$$

where $< \cdot, \cdot >$ denotes the inner product operator between feature vectors. Unlike the color features we now want to maximize d rather then minimize it (we can always minimize $-d$). The parameter $\lambda = 0.5$ was chosen empirically (see Fig. 11) and was fixed in all of our experiments.

6 Complexity

Computing the BBS between two point sets $P, Q \in \mathbb{R}^d$, requires computing the distance between each pair of points. That is, constructing a distance matrix D where $[D]_{i,j} = d(p_i, q_j)$. Given D, the nearest neighbour of $p_i \in Q$, i.e., $NN(p_i, Q)$, is the minimal element in the ith row of D. Similarly, $NN(q_j, P)$ is the minimal element in the jth column of D. The BBS is then computed by counting the number of mutual nearest neighbours (divided by a constant). We first analyze the computational complexity of computing BBS exhaustively for every window in a query image. We then propose a caching scheme, allowing extensive computation reuse which dramatically reduces the computational complexity, trading it off with increased memory complexity.

Naive implementation. Consider a target window P of size $w \times h$ and a query image I of size $W \times H$. We want to exhaustively measure the similarity between P and every $w \times h$ window Q in image I. For brevity, we interchangeably refer to P and Q as both patches and point sets. Both target window and query image are represented in a g-dimensional feature space.

We start by considering each pixel in our target window as a point in our target point set P and similarly every pixel in some query window is considered as a point in the query point set Q. In this case, $|P| = |Q| = w \cdot h \triangleq l$ and our distance matrices D are of size $l \times l$. Assuming some arbitrary image padding, we have $W \cdot H \triangleq L$ query windows for which BBS has to be computed. Computing all the L distance matrices requires $O(Ll^2g)$. For each such distance matrix we need to find the minimal element in every row and column. The minimum computation for a single row or column is done in $O(l)$ and for the entire matrix in $O(l^2)$. Therefore, the complexity of computing BBS naively for all query windows of image I is,

$$O(Ll^4g). \tag{23}$$

This is a high computational load compared to simpler methods such as sum-of-square-difference that require only $O(Llg)$.

Distance computation reuse. Assuming our sliding window works column by column. We cache all the distance matrices computed along the first image column. Staring from the second matrix in the second column, we now have to compute the distance between just *one* new candidate pixel and the target window. This means we only have to recompute one column of D, which requires only $O(l)$. Assuming $W, H >> w, h$ the majority of distance matrices can be computed in $O(l)$, instead of $O(l^2)$. This means that computing BBS for the entire image I would now require

$$O(Ll^3g). \tag{24}$$

Minimum operator load reduction. For the majority of query windows we only have *one* new column of D, as discussed above. In this case, for all other $l-1$ columns we already know the minimal element. Therefore, we can obtain the minimum over all columns in just $O(l)$. Along the rows there are two cases to consider. First, that the minimal value, for a certain row, was in the column that was pushed out of D. In which case we have to find the minimum value for that row in $O(l)$. The second option is that the minimal value of the row was not pushed out and we know where it is. In such a case we only have to compare the new element added to the row (by the new column in D) relative to the previous minimum. This operation requires only $O(1)$. Assuming the position of the minimal value along a row is uniformly distributed, on average, there will be only one row where the minimum value needs to be recomputed. To see this consider a set of random variables $\{X_i\}_{i=1}^{l}$ such that $X_i = 1$ if and only if the minimal value in the i'th row of D was pushed out of the matrix when a new column was introduced. Assuming a uniform distribution $X_i \sim Bernoulli(1/l)$. The number of rows for which the minimum has to be recomputed is given by $m = \sum_{i=1}^{l} X_i$, and the expected number of such rows is,

$$E[m] = E\left[\sum_{i=1}^{l} X_i\right] = \sum_{i=1}^{l} E[X_i] = \sum_{i=1}^{l} \frac{1}{l} = 1. \tag{25}$$

This means that, on average, there will be only one row for which the minimum has to be computed in $O(l)$. In which case we are able to find the minimum for all rows and columns in D, in $O(l)$ instead on $O(l^2)$. Combining these results with what we have so far yields an overall BBS complexity over the entire image of,

$$O(Ll^2g). \tag{26}$$

Additional load reduction. When using color features, the complexity of BBS is actually lower due to the use of non-overlapping $k \times k$ patches (instead of individual pixels). In this case the image, candidate window and distance matrices are $\frac{1}{k^2}$ smaller, and the feature space is k^2 larger. Overall this gives a complexity of

$$O\left(\frac{Ll^2g}{k^4}\right). \tag{27}$$

Regarding the spatial distance matrix used in Eqs. (21) and (22). We note that it is fixed for a given patch size and thus only has to be computed once for each query image.

The reuse schemes presented above cannot be used with our deep features because we normalize the features differently, with respect to each query window. The above analysis ignores the complexity of extracting the deep features themselves. Additionally, some of the above techniques, and specifically using non-overlapping patches, benefits all methods, e.g., SSD, and not only BBS.

As can be seen in the above analysis BBS runtime depends on both image and template size. In practice, typical runtime, of our unoptimized Matlab code, for a 360×480 image with a 30×40 template is ~ 1 sec when using color feature, and ~ 3 sec for deep features.

7 Results

We perform qualitative as well as extensive quantitative evaluation of our method on real world data. We compare the BBS with the following similarity measures commonly used for template matching: (1) Sum-of-Square-Difference, (2) Sum-of-Absolute-Difference, (3) Normalized-Cross-Correlation, (4) color Histogram Matching using the χ^2 distance, (5) Bidirectional Similarity [22] (BDS) and (6) Kernelized Correlation Filters [42] (KCF). All methods use the same appearance-location space as BBS (location is only used where relevant).

7.1 Qualitative Evaluation

Four template-image pairs taken from the Web are used for qualitative evaluation. The templates, which were manually chosen, and the target images are shown in Figs. 1a and 1b, and in Fig. 7. In all examples, the template drastically changes its appearance due to large geometric deformation, partial occlusions, and change of background.

Detection results, using color features with RGB color space, are presented in Figs. 1a and 1b, and in Fig. 7b, and compared to the above mentioned methods as well as to the Earth Movers Distance (EMD) [20]. BBS is the only method successfully matching the template in all these challenging examples. The confidence maps of BBS, presented in Fig. 7c, show distinct and well-localized modes compared to other methods.[2] The BBPs for the first example are shown in

2. Our data and code are publicly available at: http://people.csail.mit.edu/talidekel/Best-Buddies Similarity.html

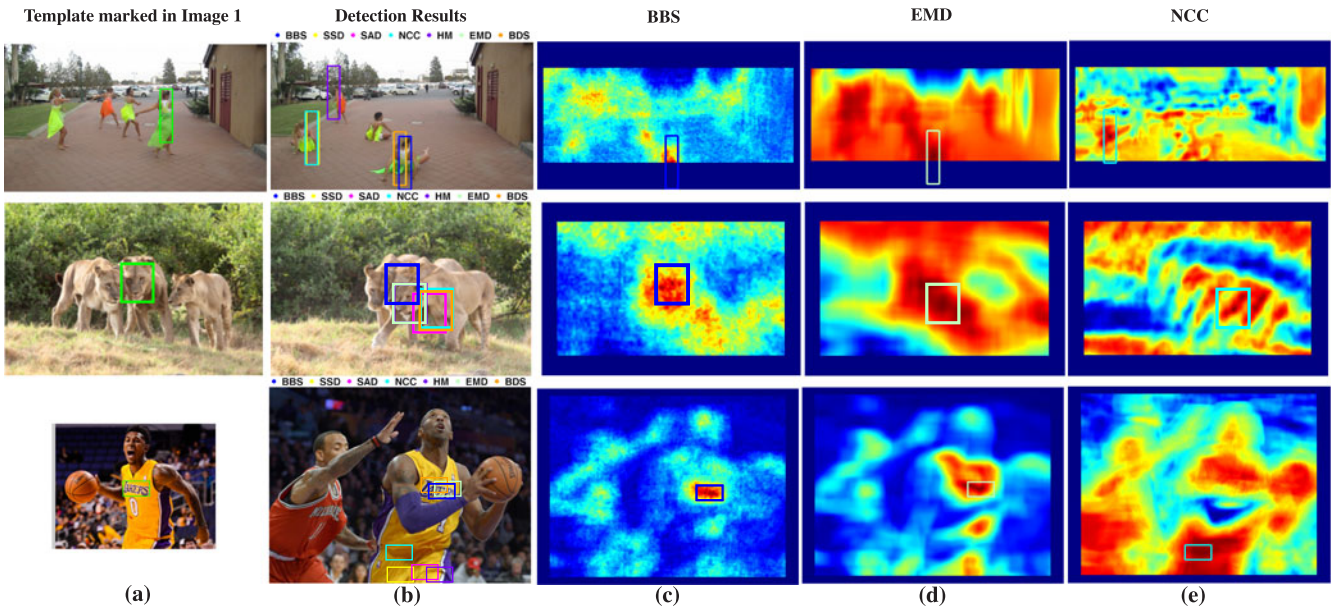


Fig. 7. *BBS results on real data:* (a) the templates are marked in green over the input images. (b) the target images marked with the detection results of 6 different methods (see text for more details). BBS results are marked in blue. (c)-(e) the resulting likelihood maps using BBS, EMD and NCC, respectively; each map is marked with the detection result, i.e., its global maxima. BBS produces well localized modes with respect to other methods and is able to indicate the correct target location in all these examples.

Fig. 1c. As discussed in Section 3, BBS captures the bidirectional inliers, which are mostly found on the object of interest. Note that the BBPs, as discussed, are not necessarily true physical corresponding points.

7.2 Quantitative Evaluation

The data for our quantitative experiments was generated using 100 annotated video sequences (color and gray-scale) taken from the Visual Tracking Benchmark[3] introduced by Wu et al. [43]. These videos capture a wide range of challenging scenes in which the objects of interest are diverse and typically undergo nonrigid deformations, photometric changes, motion blur, in/out-of-plane rotation, and occlusions.

Three template matching datasets were randomly sampled from the annotated videos. Each dataset is comprised of template-image pairs, where each such pair consists of frames f and $f+df$, where f was randomly chosen. For each dataset a different value of df was used (25, 50 or 100). The ground-truth annotated bounding box in frame f is used as the template, while frame $f+df$ is used as the query image. This random choice of frames creates a challenging benchmark with a wide baseline in both time and space (see examples in Figs. 9 and 10). For $df = 25, 50$ the data sets consist of 270 pairs and for $df = 100$ there are 254 pairs.

The ground-truth annotations were used for quantitative evaluation. Specifically, we measure the accuracy of the top match as well as the top k-ranked matches, using the common intersection over union (IoU) measure between bounding boxes

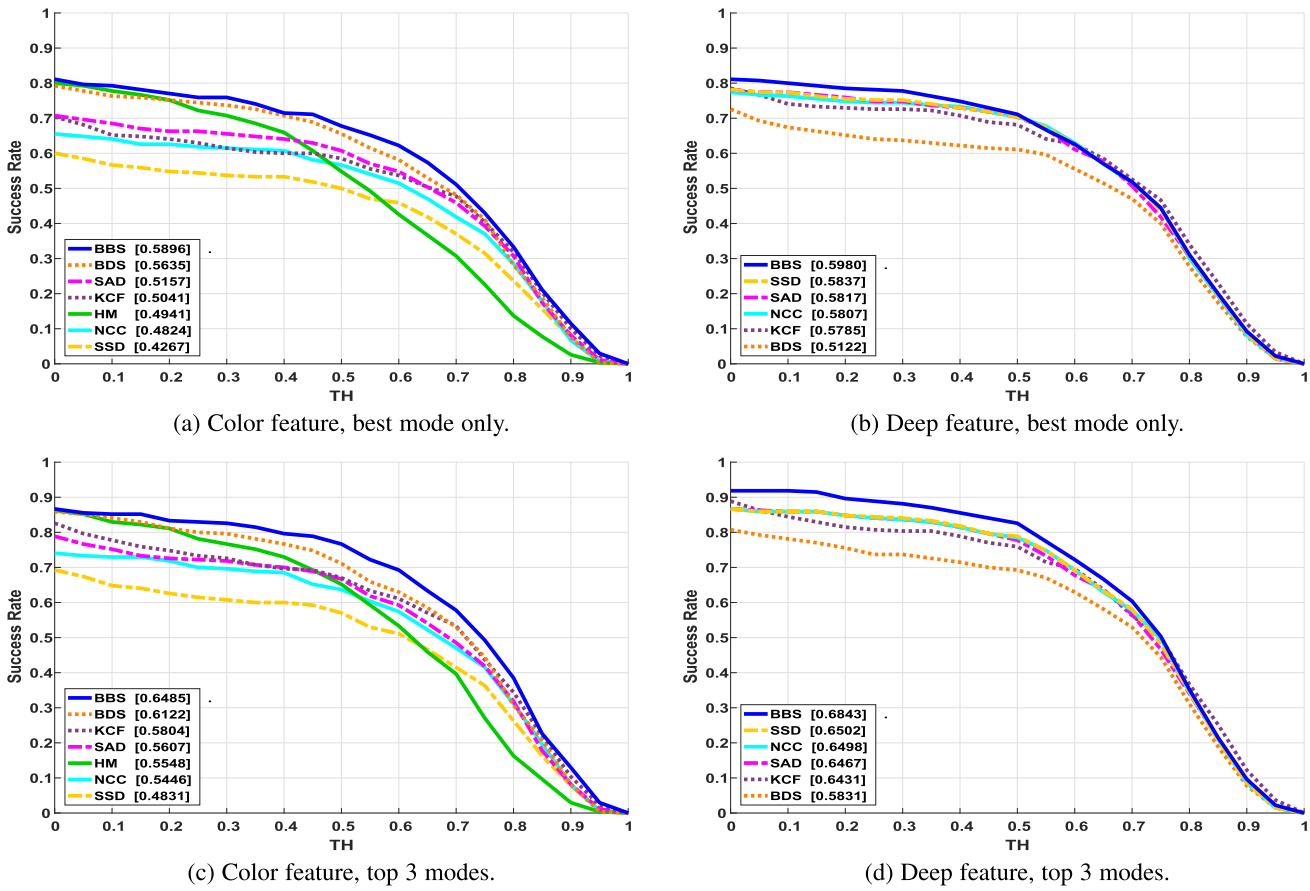
$$Acc. = \frac{\text{area}(B_e \cap B_g)}{\text{area}(B_e \cup B_g)}, \tag{28}$$

where B_e and B_g are the estimated and ground truth bounding boxes, respectively. The ROC curves show the fraction of examples with overlap larger than a threshold ($TH \in [0,1]$). Mean average precision (mAP) is taken as the area-under-curve (AUC).

Fig. 8 shows the success rates for BBS and the six similarity measures mentioned above for $df = 25$, using both color and deep features (see Section 5). We evaluated the performance considering only the global maximum (best mode) prediction (Figs. 8a and 8b) and the best out of the top 3 modes, computed using non-maximum suppression (Figs. 8c and 8d). As can be seen, BBS outperforms competing methods regardless of the feature space used. Using color features and considering only the top mode Fig. 8a, BBS outperforms competing methods with a margin ranging from 4.6 percent compared to BDS, and over 30 percent compared to SSD. When considering the top 3 modes (Fig. 8c) the performance of all methods improves, however the margin of BBS over competing methods increases as well (BBS reaches mAP of 0.648 compared to 0.589 with only the top mode). This increase in performance suggests that there are cases where BBS is able to produce a mode at the correct target position however this mode may not be the global maximum of the likelihood map.

Results using using deep feature and considering only the top mode are shown in figures Fig. 8b. We note that HM was not evaluated in this case due to the high dimensionality of the feature space requiring large amounts of samples to generate meaningful histograms for the template and candidate. We observe that BBS outperforms the second best methods by only a small margin of 2.4 percent. Considering the top 3 modes allows BBS to reach mAP of 0.684 increasing its margin relative to competing methods. For example the margin relative to the second best method (SSD) is now 5.2 percent. It is interesting to see that BDS which was the runner up when color features were used comes in last when using deep features. This demonstrates the robustness of BBS which is able to successfully use different features. Additionally, we see that the performance of BBS with deep features

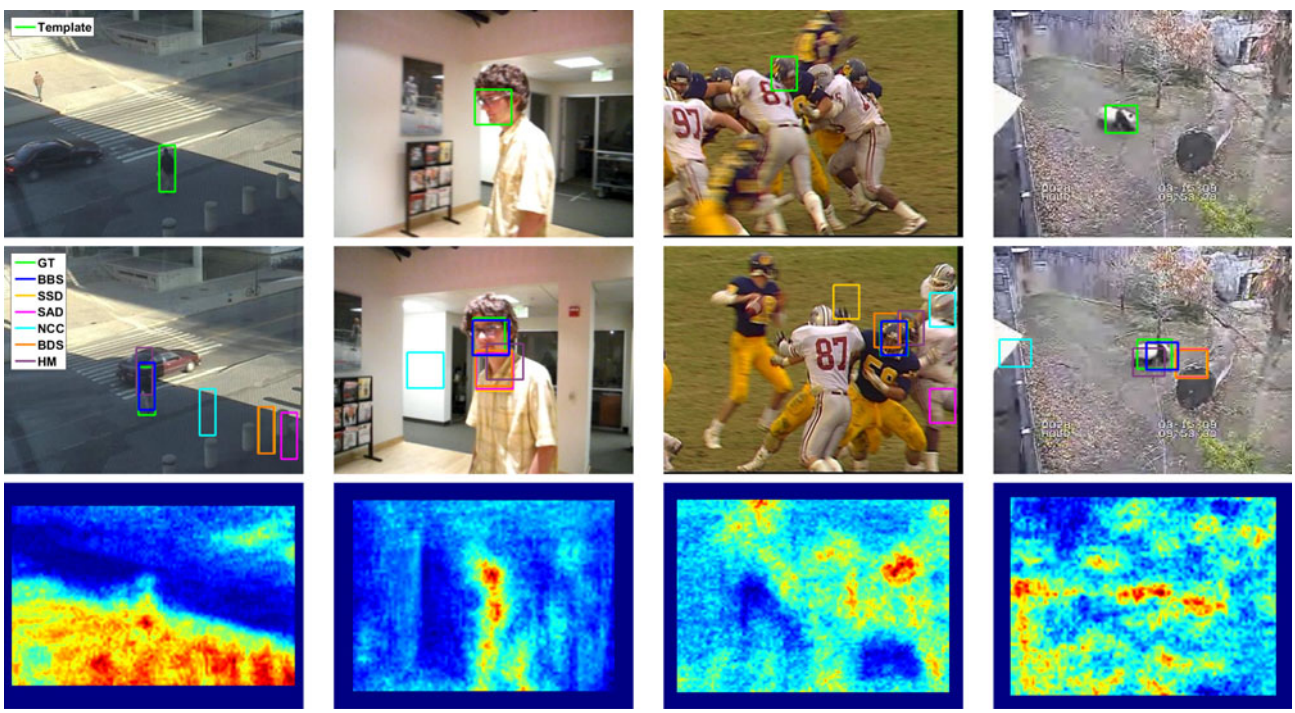
3. https://sites.google.com/site/benchmarkpami/



(a) Color feature, best mode only. (b) Deep feature, best mode only.

(c) Color feature, top 3 modes. (d) Deep feature, top 3 modes.

Fig. 8. *Template matching accuracy:* Evaluation of method performance using 270 template-image pairs with $df = 25$. BBS outperforms competing methods as can be seen in ROC curves showing fraction of examples with overlap greater than threshold values in [0,1]. Top: only best mode is considered. Bottom: best out of top 3 modes is taken. Left: Color features. Right: Deep features. Mean-average-precision (mAP) values taken as area-under-curve are shown in the legend. Best viewed in color.



(a) (b) (c) (d)

Fig. 9. *Example results using color features*. Top, input images with annotated template marked in green. Middle, target images and detected bounding boxes (see legend); ground-truth (GT) marked in green (our results in blue). Bottom, BBS likelihood maps. BBS successfully match the template in all these examples.

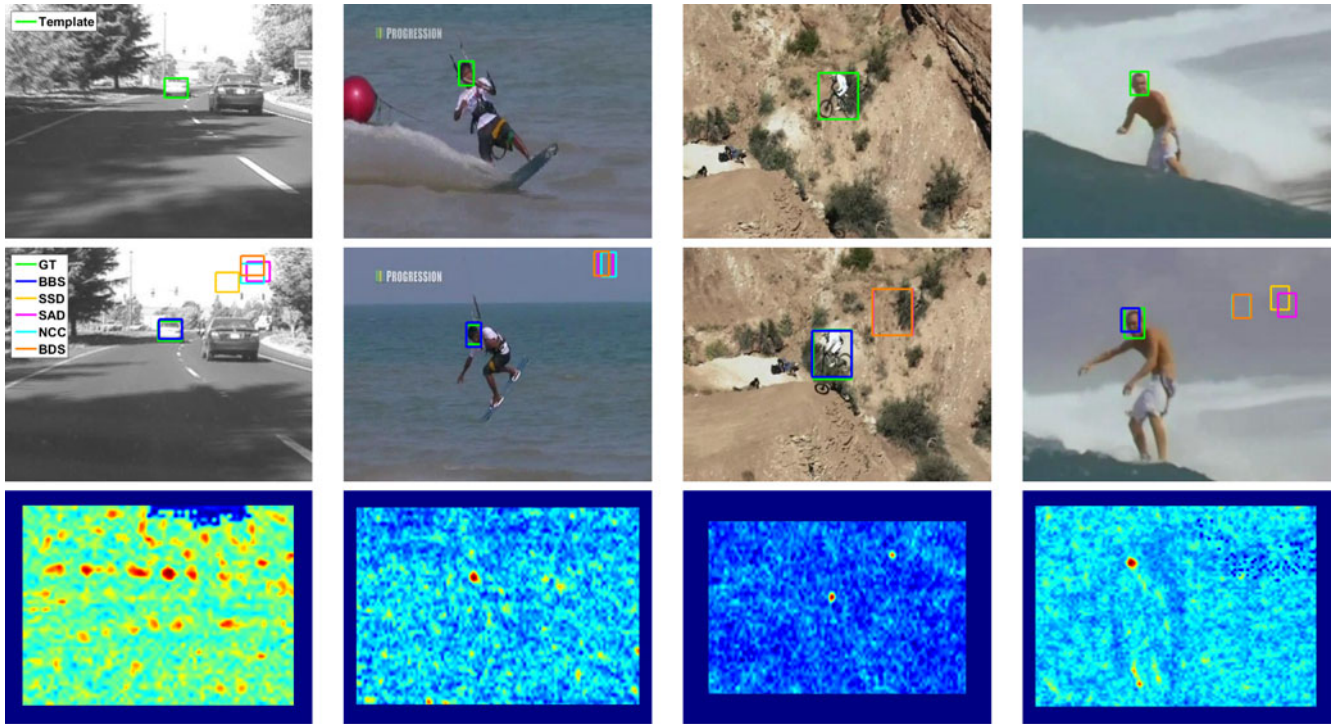


Fig. 10. *Example results using deep features.* Top, input images with annotated template marked in green. Middle, target images and detected bounding boxes (see legend); ground-truth (GT) marked in green (our results in blue). Bottom, BBS likelihood maps. BBS successfully match the template in all these examples.

improves (a margin of 5.5 percent with top 3 modes). However, this performance gain requires a significant increase in computational load.

Finally, we note that, when using the color features BBS outperforms HM which uses the χ^2 distance. Although BBS converges to χ^2 for large sets there are clear benefits for using BBS over χ^2. Computing BBS does not require modeling the distributions (i.e., building normalized histograms) and can be performed on the raw data itself. This alleviates the need to choose the histogram bin size which is known to be a delicate issue. Moreover, BBS can be performed on high dimensional data, such as our deep features, for which modeling the underlying distribution is challenging.

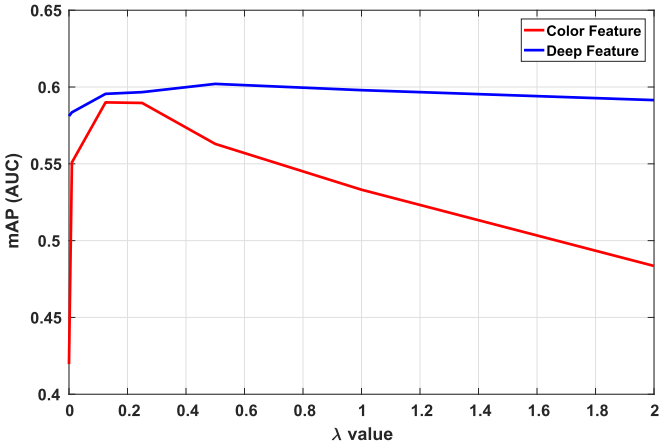


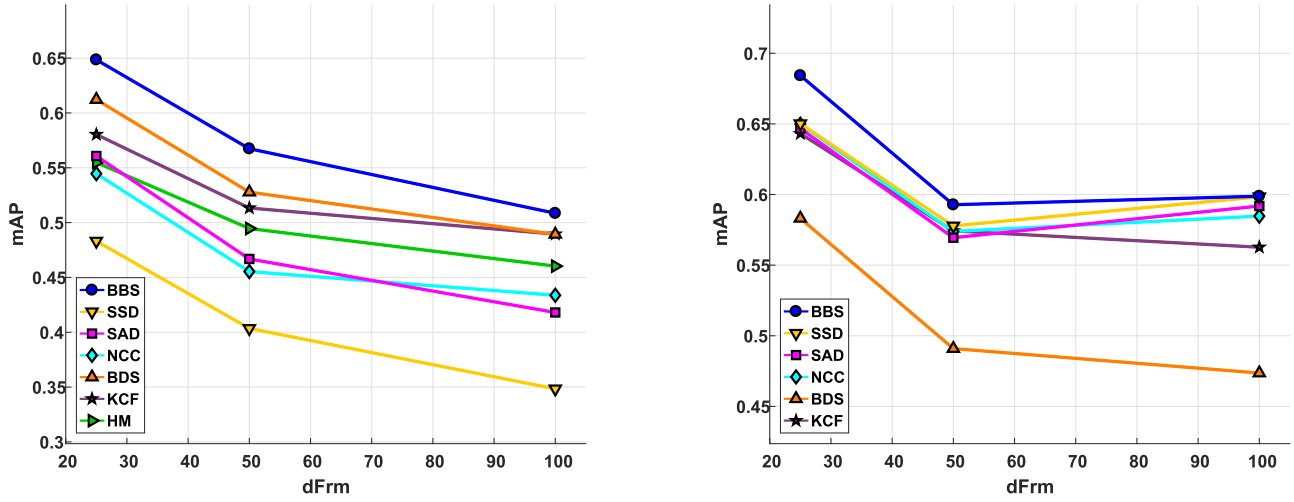
Fig. 11. *Effect of choice of λ on BBS performance:* BBS is not very sensitive to choice of λ (Results shown are for $df = 25$).

Some successful matching results, along with the likelihood maps produced by BBS using color features are shown in Fig. 9, and using the deep features in Fig. 10. Typical failure cases are also presented in Figs. 13 and 14.

The deep features are not sensitive to illumination variations and can capture both low level information as well as higher level object semantics. As can be seen, the combination of using deep features and BBS can deliver superior results due to its ability to explain non-rigid deformations. Note how when using the deep feature, we can correctly match the bike rider in Fig. 10c for which color features failed (Fig. 13 bottom row). BBS with deep features produce very well localized and compact modes compared to when color features are used.

Most of the failure cases using the color features, Fig. 13, can be attributed to either distracting objects with a similar appearance to the target (top row), illumination variations (middle row), or cases were BBS matches the background or occluding object rather than the target (bottom row). This usually happens when the target is heavily occluded or when the background region in the target window is very large. As for deep features most of the failure cases, Fig. 14, are due to distracting objects with a similar appearance (top and middle rows) or cases where BBS matches the background or occluding object (bottom row).

Effect of λ. The only parameter that has to be tuned for BBS is λ which affects the point-wise distances, weighing between appearance and spatial information. As can be seen in Fig. 11 BBS is not very sensitive to the choice of lambda. When $\lambda = 0$, performance degrades but the method does not breakdown. Using deep features is less



(a) Color features (b) Deep features

Fig. 12. *Effect of space time baseline:* Methods performance evaluated for data sets with different space-time baseline, $df = 25, 50$ and 100. Left: Color features, Right: Deep features. BBS outperforms competing methods for both feature choices and for all df values. Best viewed in color.

sensitive to the choice of lambda. We believe this is because deep features already contain some spatial information stored in the appearance channels due to the receptive field of the different neurons which have some overlap.

The space time baseline. Effect on performance was examined using data-sets with different df values (25, 50, 100). Fig. 12 shows mAP of competing methods for different values of df. Results using color features are shown on the left and using deep features on the right. All results were analyzed taking the best out of the top 3 modes. It can be seen that BBS outperforms competing methods for the different df values with the only exception being deep feature with $df = 100$ in which case BBS and SSD produce similar results reaching mAP of 0.6.

8 Conclusions

We have presented a novel similarity measure between sets of objects called the Best-Buddies Similarity. BBS leverages statistical properties of mutual nearest neighbors and was shown to be useful for template matching in the wild. Key features of BBS were identified and analyzed demonstrating its ability to overcome several challenges that are common in real life template matching scenarios. It was also shown, that for sufficiently large point sets, BBS converges to the Chi-Square distance. This result provides interesting insights into the statistical properties of mutual nearest neighbors, and the advantages of using BBS over χ^2 were discussed.

Extensive qualitative and quantitative experiments on challenging data were performed and a caching scheme

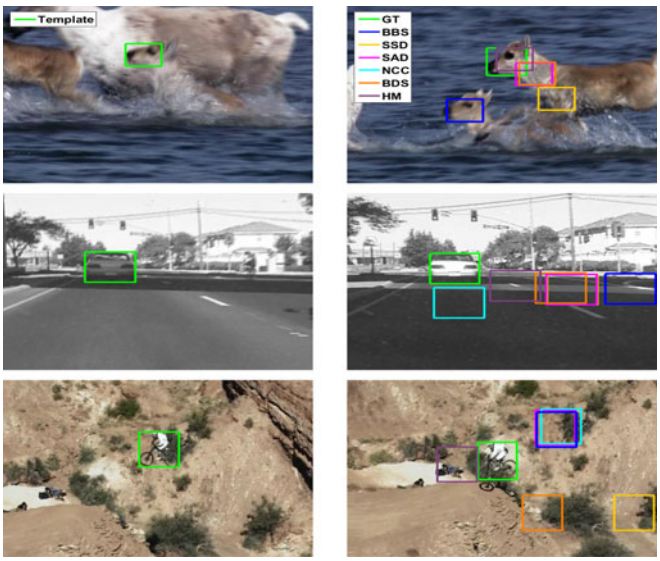


Fig. 13. *Example of failure cases using color features*. Left, input images with annotated template marked in green. Right, target images and detected bounding boxes (see legend); ground-truth (GT) marked in green (our results in blue). As can be seen, some common failure causes are illumination changes, similar distracting targets or locking onto the background.

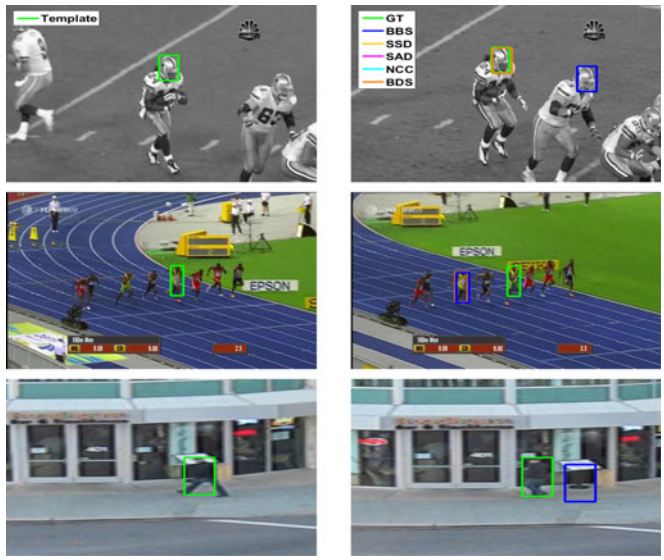


Fig. 14. *Example of failure cases using Deep features*. Left, input images with annotated template marked in green. Right, target images and detected bounding boxes (see legend); ground-truth (GT) marked in green (our results in blue). Some common failure causes are similar distracting targets or locking onto the background.

allowing for an efficient computation of BBS was proposed. BBS was shown to outperform commonly used template matching methods such as normalized cross correlation, histogram matching and bi-directional similarity. Different types of features can be used with BBS, as was demonstrated in our experiments, where superior performance was obtained using both color features as well as Deep features.

Our method may fail when the template is very small compared to the target image, when similar targets are present in the scene or when the outliers (occluding object or background clutter) cover most of the template. In some of these cases it was shown that BBS can predict the correct position (produce a mode) but non necessarily give it the highest score.

Finally, we note that since BBS is generally defined between sets of objects it might have additional applications in computer-vision or other fields that could benefit from its properties. A natural future direction of research is to explore the use of BBS as an image similarity measure, for object localization or even for document matching.

Appendix A
Proof of Lemma 1

Because of independent sampling, all points in Q have equal probability being the best buddy of p. From this we have

$$\begin{aligned} Pr[bb(p_i = p; P, Q)] &= \sum_{i=1}^{N} Pr(bb(p, q_i; P, Q) = 1) \\ &= N \cdot Pr(bb(p, q; P, Q)), \end{aligned} \tag{29}$$

where q is a point from Q and subscript is dropped for ease of description.

The probability that two points are best buddies is given by

$$\begin{aligned} &Pr(bb(p_i = p, q; P, Q)) \\ &= (F_Q(p^-) + 1 - F_Q(p^+))^{N-1} (F_P(q^-) + 1 - F_P(q^+))^{N-1}. \end{aligned} \tag{30}$$

where $F_P(x)$ and $F_Q(x)$ denote CDFs of these two distributions, that is, $F_P(x) = \Pr\{p \leq x\}$. And, $p^- = p - |p - q|$, $p^+ = p + |p - q|$, and q^+, q^- are similarly defined. Combining Eqs. (29) and (30), the probability that p_i has a best buddy equals to

$$\begin{aligned} &\lim_{N \to +\infty} N \int_{q=-M}^{M} (F_Q(p^-) + 1 - F_Q(p^+))^{N-1} \\ &\cdot (F_P(q^-) + 1 - F_P(q^+))^{N-1} f_Q(q) dq. \end{aligned} \tag{31}$$

We denote the signed distance between two points by $m = p - q$. Intuitively, because the density function are non-zero at any place, when N goes to infinity, the probability that two points $p \in P, q \in Q$ are BBP decreases rapidly as m increases. Therefore, we only need to consider the case when the distance between p and q is very small. Formally, for any positive $\overline{m}$, changing the integration limits in Eq. (31) from $\int_{p=-M}^{M}$ to $\int_{q=p-\overline{m}}^{p+\overline{m}}$ does not change the result (see Claim 2 in the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2017.2737424).

Then let us break down $F_P(\cdot)$ and $F_Q(\cdot)$ in Eq. (31). Given that the density functions $f_P(p)$ and $f_Q(q)$ are Lipschitz continuous (Condition 2 in Theorem 1), we can assume that they take a constant value in the interval $[p^-, p^+]$, and $[q^-, q^+]$. That is,

$$\begin{aligned} f_P(p^-) \approx f_P(p^+) \approx f_P(p) \\ f_Q(q^-) \approx f_Q(q^+) \approx f_Q(q). \end{aligned} \tag{32}$$

And thus, the expression $F_Q(p^+) - F_Q(p^-)$ can be approximated as follows:

$$\begin{aligned} &F_Q(p^+) - F_Q(p^-) \\ &= \int_{p^-}^{p^+} f_Q(q) dq \approx f_Q(q) \cdot (p^+ - p^-) = 2|m| \cdot f_Q(p). \end{aligned} \tag{33}$$

Similarly, $F_p(q^+) - F_P(q^-) \approx 2|m| \cdot f_P(q)$. Note that this approximation can also be obtained using Taylor expansion on $F_p(q^+)$ and $F_p(q^-)$. At last, since p and q are very close to each other, we assume

$$f_Q(q) \approx f_Q(p). \tag{34}$$

Plugging all these approximations (Eqs. (33) and (34)) to Eq. (31) and replacing q by m, we get

$$\begin{aligned} \text{Eq. (31)} = \lim_{N \to +\infty} N \int_{m=-\overline{m}}^{\overline{m}} &(1 - 2|m| f_Q(p))^{N-1} \\ &\cdot (1 - 2|m| f_P(p))^{N-1} f_Q(p) dq \end{aligned} \tag{35}$$

$$\begin{aligned} &= f_Q(p) \lim_{N \to +\infty} N \int_{m=-\overline{m}}^{\overline{m}} \Big(1 - 2(f_P(p) + f_Q(p))|m| \\ &\quad + 4 f_P(p) f_Q(p) m^2\Big)^{N-1} dm \end{aligned} \tag{36}$$

$$= f_Q(p) \lim_{N \to +\infty} N \int_{m=-\overline{m}}^{\overline{m}} \Big(1 - 2(f_P(p) + f_Q(p))m\Big)^{N-1} dm. \tag{37}$$

It is worth mentioning that the approximated equality in Eqs. (33) and (34) becomes restrict equality when N goes to infinity (for the proof see Claim 3 in the supplementary material, available online). Also, since the distance between two points m is very small, the second order term $4 f_P(p) f_Q(p) m^2$ in Eq. (36) is negligible and is dropped in Eq. (37) (for full justification see Claim 4 in the supplementary material, available online).

At last, $\lim_{N \to +\infty} N \int_{m=-\overline{m}}^{\overline{m}} (1 - a|m|)^{N-1} dm = \frac{2}{a}$ (see Claim 1 in supplementary material, available online). Thus Eq. (37) equals to

$$\frac{f_Q(p)}{f_P(p) + f_Q(p)}, \tag{38}$$

which completes the proof of Lemma 1.

ACKNOWLEDGMENTS

This work was supported in part by an Israel Science Foundation grant 1917/2015, National Science Foundation Robust Intelligence 1212849 Reconstructive Recognition, and a grant from Shell Research.

REFERENCES

[1] T. Dekel, S. Oron, S. Avidan, M. Rubinstein, and W. Freeman, "Best buddies similarity for robust template matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 2021–2029.

[2] W. Ouyang, F. Tombari, S. Mattoccia, L. Di Stefano, and W.-K. Cham, "Performance evaluation of full search equivalent pattern matching algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 127–143, Jan. 2012.

[3] Y. Hel-Or, H. Hel-Or, and E. David, "Matching by tone mapping: Photometric invariant template matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 317–330, Feb. 2014.

[4] E. Elboher and M. Werman, "Asymmetric correlation: A noise robust similarity measure for template matching," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3062–3073, Aug. 2013.

[5] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog*, 2007, pp. 1–8.

[6] J.-H. Chen, C.-S. Chen, and Y.-S. Chen, "Fast algorithm for robust template matching with m-estimators," *IEEE Trans. Signal Process.*, vol. 51, no. 1, pp. 230–243, Jan. 2003.

[7] A. Sibiryakov, "Fast and high-performance template matching method," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1417–1424.

[8] B. G. Shin, S.-Y. Park, and J. J. Lee, "Fast and robust template matching algorithm in noisy image," in *Proc. Int. Conf. Control Autom. Syst.*, 2007, pp. 6–9.

[9] O. Pele and M. Werman, "Robust real-time pattern matching using Bayesian sequential hypothesis testing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 8, pp. 1427–1443, Aug. 2008.

[10] D.-M. Tsai and C.-H. Chiang, "Rotation-invariant pattern matching using wavelet decomposition," *Pattern Recogn. Lett.*, vol. 23, no. 1–3, pp. 191–201, Jan. 2002.

[11] H. Y. Kim and S. A. De Araújo, "Grayscale template-matching invariant to rotation, scale, translation, brightness and contrast," in *Proc. Pacific-Rim Symp. Image Video Technol.*, 2007, pp. 100–113.

[12] S. Korman, D. Reichman, G. Tsur, and S. Avidan, "Fast-match: Fast affine template matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog*, 2013, pp. 2331–2338.

[13] Y. Tian and S. G. Narasimhan, "Globally optimal estimation of nonrigid image distortion," *Int. J. Comput. Vis.*, vol. 98, no. 3, pp. 279–302, Jul. 2012.

[14] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2000, pp. 142–149.

[15] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. 7th Eur. Conf. Comput. Vis.-Part I*, 2002, pp. 661–675.

[16] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 1830–1837.

[17] X. Jia, H. Lu, and M. Yang, "Visual tracking via adaptive structural local sparse appearance model," *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 1822–1829.

[18] C. F. Olson, "Maximum-likelihood image matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 853–857, Jun. 2002. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2002.1008392

[19] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," *Int. J. Comput. Vis.*, vol. 11, no. 2, pp. 213–228, 2015.

[20] Y. Rubner, C. Tomasi, and L. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, 2000.

[21] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, Sep. 1993.

[22] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing visual data using bidirectional similarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[23] A. Baumberg, "Reliable feature matching across widely separated views," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2000, pp. 774–781.

[24] M. Perdoch, J. Matas, and S. Obdrzalek, "Stable affine frames on isophotes," *Comput. Vis., 2007. ICCV 2007. IEEE 11th Int. Conf.*, pp. 1–8, 2007.

[25] T.-T. Li, B. Jiang, Z.-Z. Tu, B. Luo, and J. Tang, *Intelligent Computation in Big Data Era*. Berlin, Heidelberg: Springer. 2015, pp. 276–283. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-46248-5_34

[26] H. Liu, S. Zhang, J. Zhao, X. Zhao, and Y. Mo, "A new classification algorithm using mutual nearest neighbors," in *Proc. 9th Int. Conf. Grid Cloud Comput.*, Nov. 2010, pp. 52–57.

[27] K. Ozaki, M. Shimbo, M. Komachi, and Y. Matsumoto, "Using the mutual k-nearest neighbor graphs for semi-supervised classification of natural language data," in *Proc. 15th Conf. Comput. Natural Language Learn.*, 2011, pp. 154–162. [Online]. Available: http://dl.acm.org/citation.cfm?id=2018936.2018954

[28] H. Jegou, H. Harzallah, and C. Schmid, "A contextual dissimilarity measure for accurate and efficient image search," *Comput. Vis. Pattern Recog., 2007. CVPR'07. IEEE Conf.*, pp. 1–8, 2007.

[29] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool, "Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 777–784.

[30] A. Delvinioti, H. Jégou, L. Amsaleg, and M. E. Houle, "Image retrieval with reciprocal and shared nearest neighbors," in *Proc. Int. Conf. Comput. Vis. Theory Appl.*, 2014, pp. 321–328.

[31] G. Snedegor and W. G. Cochran, "Statistical methods," *Statistical methods.*, no. 6th ed, Ames IA, USA: Iowa State University Press, 1967.

[32] M. Varma and A. Zisserman, "A statistical approach to material classification using image patch exemplars," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 11, pp. 2032–2047, Nov. 2009.

[33] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[34] P.-E. Forssén and D. G. Lowe, "Shape descriptors for maximally stable extremal regions," in *Proc. 11th IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[35] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 530–549, May 2004.

[36] D. Pomeranz, M. Shemesh, and O. Ben-Shahar, "A fully automated greedy square jigsaw puzzle solver," in *Proc. 24th IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 9–16. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2011.5995331

[37] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3074–3082.

[38] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3119–3127.

[39] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, Dec. 2006, pp. 2037–2041.

[40] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. 24th IEEE Conf. Comput. Vis. Pattern Recog. Workshop*, 2014, pp. 512–519.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[42] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[43] Y. Wu, J. Lim, and M. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 2411–2418.

Shaul Oron received the BSc and MSc degree in electrical engineering and physics all from the Tel-Aviv University. He is working toward the PhD degree in electrical engineering with the Tel-Aviv University under supervision of Prof. Shai Avidan. His main research interests include computer vision and image processing focusing mainly on visual tracking and template matching of deformable and non-rigid objects.

Tali Dekel received the PhD degree in the School of Electrical Engineering, Tel-Aviv University, under the supervision of Prof. Shai Avidan, and Prof. Yael Moses. She has recently joined Google as a research scientist, working on developing computer vision and computer graphics algorithms. Before Google, she was a postdoctoral associate in the Computer Science and Artificial Intelligence Lab (CSAIL), MIT, working with Prof. William T. Freeman. Her PhD focused on the use of multi-camera systems to solve classic and innovative tasks in computer vision and computer graphics including 3D structure and 3D motion estimation, content-geometry aware stereo retargeting, and photo sequencing (recovering temporal order of distributed image set). In her postdoc studies, she has been working on developing new algorithms that detect and visualize imperfections/irregularities in a single image. Her research interests include computer vision and graphics, geometry, 3D reconstruction, motion analysis, and image visualization.

Tianfan Xue received the BE degree from Tsinghua Universtiy, and the MPhil degree from the Chinese University of Hong Kong. He is working toward the PhD degree with MIT CSAIL. His research interests include computer vision, image processing, and machine learning.

William T. Freeman is the Thomas and Gerd Perkins professor of Electrical Engineering and Computer Science, MIT, and a member of the Computer Science and Artificial Intelligence Laboratory (CSAIL) there. He was the Associate Department Head from 2011-2014. His current research interests include machine learning applied to computer vision, Bayesian models of visual perception, and computational photography. He received outstanding paper awards at computer vision or machine learning conferences in 1997, 2006, 2009 and 2012, and test-of-time awards for papers from 1990 and 1995. Previous research topics include steerable filters and pyramids, orientation histograms, the generic viewpoint assumption, color constancy, computer vision for computer games, and belief propagation in networks with loops. He is active in the program or organizing committees of computer vision, graphics, and machine learning conferences. He was the program co-chair for ICCV 2005, and for CVPR 2013.

Shai Avidan received the PhD degree from the Hebrew University, Jerusalem, Israel, in 1999. He is an associate professor in the School of Electrical Engineering, Tel-Aviv University, Israel. Later, he was a postdoctoral researcher at Microsoft Research, a Project Leader at MobilEye, a Research Scientist at Mitsubishi Electric Research Labs (MERL), and a senior researcher at Adobe. He published extensively in the fields of object tracking in video and 3-D object modeling from images. Recently, he has been working on Computational Photography. He was an Associate Editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* as well as a program committee member of multiple conferences and workshops in the fields of Computer Vision and Computer Graphics.