The Generalized Laplacian Distance and its Applications for Visual Matching

Elhanan Elboher[1] Michael Werman[1] Yacov Hel-Or[2]

[1]School of Computer Science,
The Hebrew University of Jerusalem,
Jerusalem, 90914, Israel
{elhanan.elboher@mail,werman@cs}.huji.ac.il

[2]School of Computer Science,
The Interdisciplinary Center,
Kanfey Nesharim St., Herzliya, 46150, Israel
toky@idc.ac.il

Abstract

The graph Laplacian operator, which originated in spectral graph theory, is commonly used for learning applications such as spectral clustering and embedding. In this paper we explore the Laplacian distance, a distance function related to the graph Laplacian, and use it for visual search. We show that previous techniques such as Matching by Tone Mapping (MTM) are particular cases of the Laplacian distance. Generalizing the Laplacian distance results in distance measures which are tolerant to various visual distortions. A novel algorithm based on linear decomposition makes it possible to compute these generalized distances efficiently. The proposed approach is demonstrated for tone mapping invariant, outlier robust and multimodal template matching.

1. Introduction

Distance and similarity measures are fundamental in computer vision. Functions with different invariance and robustness properties are required in order to overcome different visual distortions depending on the application and acquisition conditions. Common distortions are noise, illumination change, different camera parameters, occlusion and geometric transformation. In many cases efficient algorithms are also required for fast computation of the desired distance functions.

In this paper we introduce the Generalized Laplacian Distances (GLDs), a family of distance functions related to the graph Laplacian operator. The graph Laplacian, which is studied in spectral graph theory [3], has been used for machine learning problems such as spectral clustering [13, 10, 15] and dimensionality reduction [1, 11]. These have been applied in computer vision for image segmentation [13] and face recognition [6]. However, so far, the Laplacian operator has not been used for the design of sophisticated distance functions.

The Laplacian distance $LD(v,x)$ is a property of the graph Laplacian that can be interpreted as an asymmetric cross distance between two vectors $\mathbf{v},\mathbf{x}\in\mathbb{R}^M$. The graph edge weights are determined by $\mathbf{v}$, $w_{ij}=w(v_i,v_j)$; then the Laplacian distance of $\mathbf{x}$ from $\mathbf{v}$ is defined as

$$LD(\mathbf{v},\mathbf{x})=\frac{1}{2}\sum_{ij}w(v_i,v_j)(x_i-x_j)^2 \qquad (1)$$

The Laplacian distance is small where small squared differences $(x_i-x_j)^2$ correspond to large weights $w(v_i,v_j)$. Assigning large weights to pairs (v_i,v_j) with close values (e.g. $e^{-(v_i-v_j)^2}$), the Laplacian distance is small iff (x_i,x_j) are close wherever (v_i,v_j) are close. In this sense, the Laplacian distance captures the pairwise proximity of $\mathbf{v}$. An illustration is shown in Figure 1.

We extend the concept of the Laplacian distance to develop a rich family of distance functions, Generalized Laplacian Distances (GLDs). In this family the squared difference used by LD (Equation 1) is replaced with a general dissimilarity function, f_{ij}, depending on (x_i,x_j) and possibly also (v_i,v_j).

The proposed generalization makes it possible to replace the squared difference with a more robust dissimilarity f_{ij} such as the absolute difference $|x_i-x_j|$. An even greater advantage can be achieved by changing both the weight function w_{ij} and the dissimilarity f_{ij}. By careful selection of w and f, GLDs can be designed to capture various types of similarity. For example, the GLD_{sign} distance (Section 5.1) captures the order in each pair. This distance is small for vectors $\mathbf{x},\mathbf{v}$ for which $x_i\le x_j$ iff $v_i\le v_j$; an illustration is shown in Figure 2.

We demonstrate the advantage of GLD by an application to template matching. A template $\mathbf{p}$ is sought in a large image which may be distorted by noise, captured in different illumination conditions, or even from different modality (e.g. visual vs. infrared). Usually the best match to $\mathbf{p}$ in the image is found by minimizing a distance

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.300

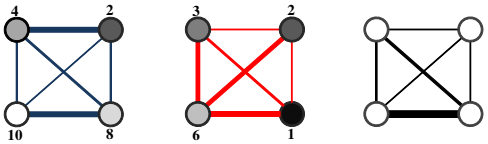


Figure 1. Illustration of a Laplacian distance between $\mathbf{v} = \mathbf{p}$ and $\mathbf{x} = \mathbf{q}$. Left: the graph $G(\mathbf{v} = \mathbf{p})$. The weights w_{ij} are large (thick lines) where p_i and p_j (intensity values) are similar. Middle: the dissimilarities f_{ij} are defined as $(q_i - q_j)^2$). Right: the weighted dissimilarity of each pair, $w_{ij} \cdot f_{ij}$. The Laplacian distance is the sum of these weighted dissimilarities, which preserves the ***pairwise proximity*** of $\mathbf{p}$.

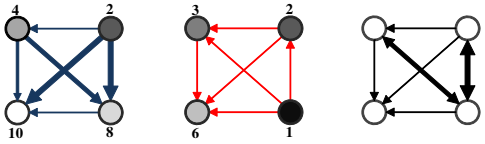


Figure 2. Illustration of a generalized Laplacian distance, GLD_{sign} (Section 5.1). Left: the arrows represent the order of (p_i, p_j). Large weights are assigned to pairs with distant values $|p_i - p_j|$. Middle: the orders of the corresponding pairs (q_i, q_j). Right: the dissimilarity f penalizes pairs (i, j) with opposite orders in $\mathbf{p}$ and $\mathbf{q}$ (double-end arrows). f_{ij} is multiplied by the edge weight $w(p_i, p_j)$. The distance $\text{GLD}_{sign}(\mathbf{p}, \mathbf{q})$ is the sum of these weighted dissimilarities.

$d(\mathbf{p}, \mathbf{q})$ over all possible candidate windows $\mathbf{q}$ in the image. An appropriate distance should be tolerant to the possible distortions between the template and the searched image. Additionally, the location of the best match should be computed efficiently.

Matching a template over an image can be performed efficiently by common measures such as the Euclidean distance, the sum of absolute differences (SAD) or the normalized cross correlation (NCC, a.k.a. the correlation coefficient) [8, 2, 9, 12]. However, treating combinations of distortions such as illumination change and outliers, or matching noisy mutlimodal data, is more difficult.

Recently, Hel-Or et al. proposed the Matching by Tone Mapping (MTM) technique [7]. This is an efficient template matching algorithm based on a distance function invariant to an arbitrary tone mapping of the image values. In Section 2.4 we show that MTM is a particular case of a Laplacian distance. Moreover, by replacing the dissimilarity function f we define in Section 4 GLD_{abs} which is more robust than MTM. GLD_{abs} is computed efficiently using a novel technique based on a linear decomposition of non Euclidean distance functions. The advantage of GLD_{abs} is demonstrated experimentally in Section 6; An example is shown in Figure 3.

In Section 5 we propose C_{sign}, a robust variant of the normalized cross correlation (NCC). Maximizing C_{sign} is equivalent to minimizing GLD_{sign}, a GLD which preserves the order property for the informative pairs in the template (see Figure 2). C_{sign} is invariant to monotonic intensity transformations and robust to outliers. Combining both advantages, C_{sign} is successful also in cases where outlier robust methods (e.g. SAD) and methods invariant to affine illumination transform (e.g. NCC) tend to fail, as shown by our experimental analysis (Section 6).

2. The Laplacian Distance (LD)

2.1. Definition

Let $G(V, W)$ be a weighted graph represented by a matrix W containing the edge weights w_{ij} for each node pair (v_i, v_j). The Laplacian of G is defined as

$$L = D - W \tag{2}$$

where D is a diagonal matrix, $d_i = D_{ii} = \sum_j W_{ij}$.

Given a vector $\mathbf{x} \in \mathbb{R}^M$, the Laplacian has the following property [3]:

$$\mathbf{x}^T L \mathbf{x} = \mathbf{x}^T (D - W)\mathbf{x} = \frac{1}{2} \sum_{ij} w_{ij}(x_i - x_j)^2 \tag{3}$$

There are several variants of normalizing the graph Laplacian. In this paper we use the left normalized Laplacian [3], $\tilde{L} = D^{-1}L$,

$$\mathbf{x}^T \tilde{L} \mathbf{x} = \mathbf{x}^T (I - D^{-1}W)\mathbf{x} = \frac{1}{2} \sum_{ij} \frac{w_{ij}}{d_i}(x_i - x_j)^2 \tag{4}$$

Suppose that the graph G depends on a parameter vector $\mathbf{v} \in \mathbb{R}^M$, i.e., the edge weights w_{ij} are determined by a weight function $w(v_i, v_j)$ (e.g. Gaussian weights $\exp(-\frac{(v_i - v_j)^2}{2\sigma^2})$). Then Equation 3 defines the following non symmetric function between $\mathbf{v}$ and $\mathbf{x}$:

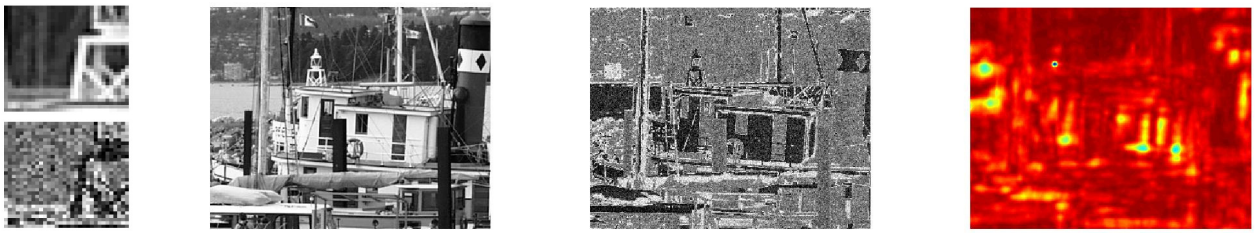
$$LD(\mathbf{v}, \mathbf{x}) = \mathbf{x}^T L(\mathbf{v})\mathbf{x} = \frac{1}{2} \sum_{ij} w(v_i, v_j)(x_i - x_j)^2 \tag{5}$$

We refer to $LD(\mathbf{v}, \mathbf{x})$ as the *Laplacian distance* between $\mathbf{v}$ and $\mathbf{x}$. Similarly, the normalized Laplacian distance $NLD(\mathbf{v}, \mathbf{x})$ is defined by dividing each $w(v_i, v_j)$ by d_i.

The distance of $\mathbf{x}$ from $\mathbf{v}$ is small where the following property is satisfied: pairs (i, j) with large weights, usually indicating that v_i and v_j are close, have close values in $\mathbf{x}$ (i.e. small difference $(x_i - x_j)^2$). For pairs (i, j) with small weights, the proximity of x_i and x_j is irrelevant. The normalization by D^{-1} makes the contribution of each v_i equal, since the sum of the normalized weights $\sum_j w(v_i, v_j)$ is 1 for each i.

2.2. Laplacian distance for template matching

Assume a template (pattern) $\mathbf{p}$ of M pixels is sought in an image. One can represent $\mathbf{p}$ by a weighted graph $G(\mathbf{p})$ in



(a) $\mathbf{p}$ and $\mathbf{q}^*$ (b) original image (c) tone-mapped image (d) GLD_{abs}

Figure 3. Template matching with non-monotonic intensity transformation, additive noise $\sigma = 30/255$ and 30% outliers (random values in random locations). (a) The template $\mathbf{p}$ (32×32) and the correct match $\mathbf{q}^*$. (b) the mapped image. (c) GLD_{abs} distance map.

which each node is associated with a pixel, and the weights w_{ij} depend on grayscale values p_i, p_j.

Given $G(\mathbf{p})$, the best match to $\mathbf{p}$ in the image can be defined as the window $\mathbf{q}^*$,

$$\mathbf{q}^* = \arg\min_{\mathbf{q}} \sum_{ij} w(p_i, p_j)(q_i - q_j)^2 \tag{6}$$

i.e., the window $\mathbf{q}$ that minimizes $LD(\mathbf{p}, \mathbf{q})$. We call this approach pattern-based (PB), since the graph weights are determined by the grayscale values of the template $\mathbf{p}$.

Alternatively, a window based approach (WB) can also be defined:

$$\mathbf{q}^* = \arg\min_{\mathbf{q}} \sum_{ij} w(q_i, q_j)(p_i - p_j)^2 \tag{7}$$

In this case the graph weights are determined the grayscale values of $\mathbf{q}$, and the best match $\mathbf{q}^*$ minimizes $LD(\mathbf{q}, \mathbf{p})$.

2.3. Efficiency and weights decomposition

The weight function w strongly influences what is considered a good match. Moreover, the computational complexity strongly depends on choosing w which can be computed efficiently.

A naive computation of Equations 6 and 7 requires $O(M^2)$ operations for each candidate window $\mathbf{q}$ which is time consuming when applying to the entire image. This can be reduced significantly using low rank decomposition as follows.

Let $w(x, y)$ be a real function defined on a discrete bounded domain, $w : [1, 2, ...m] \times [1, 2, ...m] \rightarrow \mathbb{R}$. One can represent w by an $m \times m$ matrix $H \in \mathbb{R}^{m \times m}$, where $H_{x,y} = w(x, y)$.

Consider the rank-K approximation of H by singular value decomposition, $H = U\Sigma V^T$ where $U, V \in \mathbb{R}^{m \times K}$ and $\Sigma \in \mathbb{R}^{K \times K}$. Given two integers $(x, y) \in [1..m]$, it follows that

$$w(x, y) = H_{x,y} \approx \sum_{k=1}^{K} \sigma_k U_k[x] V_k[y] \tag{8}$$

where σ_k is the kth singular value, and U_k, V_k are the corresponding eigenvectors. Storing σ_k, U_k and V_k in cache, Equation 8 can be computed using lookup tables.

Based on Equation 8, the Laplacian distance can be rewritten as

$$LD(\mathbf{p}, \mathbf{q}) = \mathbf{q}^T D \mathbf{q} - \sum_{k=1}^{K} \sigma_k \Big(U_k[\mathbf{p}]^T \mathbf{q} \Big) \Big(V_k[\mathbf{p}]^T \mathbf{q} \Big) \tag{9}$$

where $U_k[\mathbf{p}] = \big(U_k[p_1]...U_k[p_m]\big)$. The terms $U_k[\mathbf{p}]^T \mathbf{q}$ and $V_k[\mathbf{p}]^T \mathbf{q}$ are computed by $2K$ convolutions. Note that D is a diagonal matrix depending on $\mathbf{p}$, which can be pre-computed. The term $\mathbf{q}^T D \mathbf{q} = \sum_i d_i q_i^2$ is computed by a single convolution between $\text{diag}(D)$ and the squared image values. Thus, computing $LD(\mathbf{p}, \mathbf{q})$ requires $2K + 1$ convolutions, or $K + 1$ convolutions if w is symmetric (U=V). The time complexity in the general case is therefore $O(KN \log M)$. In specific cases the complexity can be further reduced (Section 2.4).

The computation of the WB variant, $LD(\mathbf{q}, \mathbf{p})$ is similar but now D depends on $\mathbf{q}$ and varies between different image windows. The WB Laplacian distance is

$$\begin{aligned} LD(\mathbf{q}, \mathbf{p}) = &\sum_{k=1}^{K} \sigma_k \Big(U_k[\mathbf{q}]^T \mathbf{p}^2 \Big) \Big(V_k[\mathbf{q}]^T \mathbf{1} \Big) \\ &- \sum_{k=1}^{K} \sigma_k \Big(U_k[\mathbf{q}]^T \mathbf{p} \Big) \Big(V_k[\mathbf{q}]^T \mathbf{p} \Big) \end{aligned} \tag{10}$$

where $\mathbf{p}^2$ denote the squared template, and $\mathbf{1}$ is a vector of ones (box filter). Equation 10 is computed as follows: for each $k = 1...K$ the entire image is mapped by U_k (denote I_k^U) and by V_k (denote I_k^V). The first sum is computed by convolving I_k^U with $\mathbf{p}^2$, summing I_k^V in each window using a box filter, and multiplying the results point-wise. The second sum is computed by convolving both I_k^U and I_k^V with $\mathbf{p}$, and multiplying the results point-wise. Thus, K box filters and $3K$ convolutions are required (or $2k$ convolutions if the weights are symmetric).

When w depends only on the value difference $(x-y)$, a 1D frequency decomposition is also possible. This approach was suggested in [5] for fast image filtering. The function $w(x-y)$ is expressed as a linear combination of cosine frequencies $\cos(k(x-y))$, the discrete cosine transform (DCT) of w. Following the identity $\cos(k(x-y)) = \cos(kx)\cos(ky) + \sin(kx)\sin(ky)$, w

can be approximated in the same form as Equation 8 by a linear combination of cosine and sine functions (vectors) weighted by the DCT coefficients. We found that this approximation is better than SVD in practice (see Section 4).

2.4. MTM as a Laplacian distance

In [7], Hel-Or et al. proposed a template matching method which is closely related to the Laplacian distance. This method, called Matching by Tone Mapping (MTM), is robust to an intensity transform (tone mapping) of the sought template. The MTM method has two variants, pattern-to-window (P2W) and window-to-pattern (W2P). In the following we show that these variants are particular cases of the pattern based (PB) and the window based (WB) Laplacian distance, respectively.

The P2W variant of MTM is defined as follows:

$$MTM(\mathbf{p},\mathbf{q}) = \min_{\beta} \frac{\|S(\mathbf{p})\beta - \mathbf{q}\|^2}{\text{var}(\mathbf{q})} \tag{11}$$

Here the grayscale range (e.g. $[0..255]$) is divided into K bins, and $S(\mathbf{p}) \in \{0,1\}^{M\times K}$ is a matrix that indicates for each p_i whether it falls to the kth bin, $k = 1...K$. Denoting by $b(x)$ the bin in which x falls, $S(i,k) = \delta\big(b(p_i) - k\big)$.

$S(\mathbf{p})\beta$ applies a piecewise constant tone mapping to $\mathbf{p}$ where the optimization parameter, $\beta \in \mathbb{R}^K$, is a vector that represents piecewise constant tone mapping. Thus, MTM is the minimal squared Euclidean distance between $\mathbf{q}$ and $\tilde{\mathbf{p}} = T(\mathbf{p})$ over all possible piecewise constant tone mappings $T(\cdot)$. Due to the difference between different windows $\mathbf{q}$, the distance is normalized by the variance of $\mathbf{q}$.

The authors of [7] derived a closed form solution for Equation 11,

$$MTM(\mathbf{p},\mathbf{q}) = \frac{\mathbf{q}^T\mathbf{q} - \sum_k \frac{1}{n_k}\Big(S_k(\mathbf{p})^T\mathbf{q}\Big)^2}{\text{var}(\mathbf{q})} \tag{12}$$

where n_k is the number of pixels p_i in the kth bin, and $S_k(\mathbf{p})$ is the kth template slice – i.e., the kth column of $S(\mathbf{p})$, which is a binary mask. In [7], MTM is computed efficiently using sparse convolutions [16]. This is possible since the slices $S_k(\mathbf{p})$ are binary and non overlapping. Computing $MTM(\mathbf{p},\mathbf{q})$ requires K sparse convolutions, which sum up to a single convolution. A similar computation scheme is described in [7] also for the W2P variant, $MTM(\mathbf{q},\mathbf{p})$.

In the following we rewrite the MTM distance (Equation 12) in Laplacian form. The graph weights w_{ij} are determined by the delta weight function:

$$w(x,y) = H_{x,y} = \delta\Big(b(x) - b(y)\Big) \tag{13}$$

H is an $m\times m$ matrix of rank K and an exact decomposition $H = UU^T$, where $U_k[x] = \delta\big(b(x) - k\big)$. Thus the kth template slice, $S_k(\mathbf{p})$, is the result of applying U_k as a mapping (lookup table) over $\mathbf{p}$ values.

We also note that if $b(p_i) = k$ then

$$d_i = \sum_j w(p_i,p_j) = \sum_j \delta\big(b(p_j) - k\big) = n_k \tag{14}$$

Rearranging Equation 12 and plugging in the delta weights and the normalization factors $d_i = n_k$, we get the following equivalent formulation for MTM:

$$MTM(\mathbf{p},\mathbf{q}) = \frac{1}{2}\sum_{ij} \frac{w(p_i,p_j)}{d_i} \cdot \frac{(q_i - q_j)^2}{\text{var}(\mathbf{q})} \tag{15}$$

Hence, $MTM(\mathbf{p},\mathbf{q})$ is a pattern-based (PB), left-normalized Laplacian distance (Equation 4). This distance is defined by delta weights depending on $\mathbf{p}$ and squared differences depending on $\mathbf{q}/\text{std}(\mathbf{q})$.

By replacing $\mathbf{p}$ and $\mathbf{q}$ in Equations 11–15 it follows also that the W2P variant of MTM is a window-based (WB) Laplacian distance with delta weights depending on $\mathbf{q}$ and squared differences depending on $\mathbf{p}$.

3. Generalized Laplacian Distances (GLDs)

In the Laplacian distance the dissimilarity between x_i and x_j is expressed by the squared difference $(x_i - x_j)^2$ (Equation 5). We generalize the concept of the Laplacian distance by replacing the squared difference with an arbitrary dissimilarity function f_{ij} depending on x_i and x_j, and possibly also on v_i and v_j. We also omit the factor $\frac{1}{2}$ which makes no sense in the pairwise i,j formulation. The resulting, *generalized Laplacian distance* (GLD) is therefore

$$GLD(\mathbf{v},\mathbf{x}) = \sum_{ij} w_{ij} f_{ij} \tag{16}$$

where $w_{ij} = w(v_i,v_j)$ and $f_{ij} = f(x_i,x_j;v_i,v_j)$.

GLD is more general than LD. First, with GLD one can define dissimilarities, e.g. the absolute difference $|x_i - x_j|$ that is more robust than the squared difference. Second, it is possible to design GLDs that preserve properties other than value proximity, e.g. the sign of the pairwise difference Δ_{ij}. Both these advantages are used below to design novel distance functions with various robustness properties.

From a computational viewpoint, GLD is more challenging than LD. The squared difference $(x_i - x_j)^2$ used by LD is decomposed to $x_i^2 + x_j^2 - 2x_ix_j$ for fast computation (Equations 9,10). Such a decomposition is not available for the dissimilarities f_{ij} used by GLDs. In the next sections (4,5) we compute GLDs efficiently by decomposing the dissimilarity function f using the spectral methods described in Section 2.3 (SVD and DCT).

4. Robust, Tone Mapping Invariant GLD_{abs}

Based on the Laplacian interpretation of MTM (Equation 15) we define the following distance:

$$GLD_{abs}(\mathbf{p},\mathbf{q}) = \frac{\sum_{ij} d_i^{-1} w(p_i,p_j)|q_i - q_j|}{\sum_{ij}|q_i - q_j|} \tag{17}$$

Here the weight function remains as in the original MTM (Equation 15), i.e. delta weights and $d_i = n_k$ (where $b(p_i) = k$). The only change is replacing the squared differences $(q_i - q_j)^2$ by absolute differences $|q_i - q_j|$, both in the nominator and denominator (note that $\text{var}(\mathbf{q}) = \frac{1}{2M^2}\sum_{ij}(q_i - q_j)^2$). The proposed change significantly improves the performance in noisy scenarios, as demonstrated in Section 6. Here too, naively computing GLD_{abs} for the entire image requires $O(M^2N)$ operations. In the following we present efficient algorithms for both the PB and WB variants of GLD_{abs}. We will show that in both cases only few convolutions with the image are required. The total complexity is $O(RMN)$ where R is a small approximation rank.

4.1. Pattern based GLD_{abs}

We use the technique described in Section 2.3 in order to decompose the function $h(x, y) = |x - y|$. Since the absolute difference is symmetric, Either the SVD or the frequency decomposition are of the form

$$|x - y| \approx \sum_{r=1}^{R} \lambda_r U_r[x] U_r[y] \tag{18}$$

where λ_r are either singular values or DCT coefficients.

Decomposing the delta weights (Equation 13) and the absolute difference (Equation 18), and using the equality $d_i = n_k$ (Equation 14), the nominator of GLD_{abs} (Equation 17) can be rewritten as

$$\sum_{ij} \frac{1}{d_i} w(p_i, p_j)|q_i - q_j| \tag{19}$$

$$\approx \sum_{ij} \left(\sum_{k=1}^{K} \frac{1}{n_k} S_k[p_i] S_k[p_j] \right) \left(\sum_{r=1}^{R} \lambda_r U_r[q_i] U_r[q_j] \right)$$

$$= \sum_{k=1}^{K} \frac{1}{n_k} \sum_{r=1}^{R} \lambda_r \left(\sum_{ij} S_k[p_i] U_r[q_i] S_k[p_j] U_r[q_j] \right)$$

$$= \sum_{k=1}^{K} \sum_{r=1}^{R} \frac{\lambda_r}{n_k} \Big(S_k(\mathbf{p})^T U_r(\mathbf{q}) \Big)^2$$

where $S_k(\mathbf{p})$ is the kth column of $S(\mathbf{p})$, and $U_r(\mathbf{q})$ is the result of assigning the value $U_r[q_i]$ for each value q_i. i.e., U_r is applied as a lookup table on the image values.

Computing Equation 19 requires KR convolutions. However, sparse convolutions can be used since the template slices $S_k(\mathbf{p})$ are sparse and non overlapping. Thus the computational cost is $O(RM)$ operations for each image window, or $O(RMN)$ totally, which is equivalent to R full convolutions. This is very useful since increasing the number of slices K does not increase the execution time. Additionally, the template size is usually small which means that direct convolution is faster than many (K) FFT based convolutions. In our experiments we used $R = 4$ vectors; in practice, the DCT decomposition was found to perform better than the SVD (see an illustration of this approximation in the supplementary material).

The denominator of GLD_{abs} (Equation 17) is approximated using the same decomposition,

$$\sum_{ij} |q_i - q_j| \approx \sum_{ij} \sum_{r=1}^{R} \lambda_r U_r[q_i] U_r[q_j] \tag{20}$$

$$= \sum_{r=1}^{R} \lambda_r \Big(\sum_{i} U_r[q_i] \Big)^2$$

Each of the sums $\sum_i U_r[q_i]$ over all the image window $\mathbf{q}$ can be computed by a single box filter [4, 14] with complexity $O(N)$, where N is the image size. Thus computing the denominator for all the image windows requires $O(RN)$ operations.

4.2. Window based $\text{GLD}_{\mathbf{abs}}$

Computing the window-based GLD_{abs} is slightly different. Replacing $\mathbf{p}$ and $\mathbf{q}$ in Equation 17, the denominator becomes $\sum_{ij} |p_i - p_j|$ which is constant for all the possible windows $\mathbf{q}$. Hence, minimizing $GLD_{abs}(\mathbf{q}, \mathbf{p})$ requires only the computation of the nominator, that is

$$\sum_{k=1}^{K} \sum_{r=1}^{R} \frac{\lambda_r}{n_k(\mathbf{q})} \Big(S_k(\mathbf{q})^T U_r(\mathbf{p}) \Big)^2 \tag{21}$$

Here $U_r(\mathbf{p})$ is the result of applying U_r to the template values, while $S_k(\mathbf{q})$ is the kth slice of the image (1 for pixels q_i which fall in the kth bin, and 0 otherwise). Again, KR sparse convolutions are required which are equivalent to R full convolutions. The terms $n_k(\mathbf{q})$, denoting the number of pixels in $\mathbf{q}$ which fall in the kth bin, are computed by applying a box filter to each of the image slices in $O(RN)$.

5. Robust GLD Based Correlation

One of the most used similarities in computer vision is the normalized cross correlation (a.k.a. the correlation coefficient) which is defined as

$$NCC(\mathbf{p}, \mathbf{q}) = \frac{\sum_i (p_i - \mu_{\mathbf{p}})(q_i - \mu_{\mathbf{q}})}{\sigma_{\mathbf{p}} \sigma_{\mathbf{q}}} \tag{22}$$

where $\mu_p, \mu_{\mathbf{q}}$ and $\sigma_{\mathbf{p}}, \sigma_{\mathbf{q}}$ are the means and the standard deviations of $\mathbf{p}$ and $\mathbf{q}$, respectively. NCC is invariant to affine illumination change both of $\mathbf{p}$ and $\mathbf{q}$, and tolerant to additive Gaussian noise. It is also tolerant to general monotonic intensity transform, as can be seen from the formulation:

$$NCC(\mathbf{p}, \mathbf{q}) = \frac{\sum_{ij} (p_i - p_j)(q_i - q_j)}{M^2 \sigma_{\mathbf{p}} \sigma_{\mathbf{q}}} \tag{23}$$

which is equivalent to Equation 22. This means that NCC does not change a lot by a monotonic increasing transform,

since the signs of all the differences $(q_i - q_j)$ are not changed.

On the other hand, NCC is not robust to outlier noise or partial occlusions of the sought template. Usually, other distances are used in this case such as sum of absolute differences (SAD) or M-estimators. However, these distance functions are sensitive to changes in the illumination.

One way to overcome both difficulties is to treat $\mathbf{p}$ and $\mathbf{q}$ asymmetrically. This is reasonable in the context of template matching since the given template is usually clean and informative, while the image might be distorted. Therefore we define the following non symmetric sign correlation:

$$C_{sign}(\mathbf{p},\mathbf{q}) = \sum_{ij}(p_i - p_j)\text{sign}(q_i - q_j) \qquad (24)$$

This is a robust modification of Equation 23. Due the use of the sign function, C_{sign} has two advantages. First, it is invariant to monotonic increasing transform of the image values q_i. Second, it is robust to outlier noise in the image.

5.1. C_{sign} as a GLD

An interesting perspective of C_{sign} is its close relation to the generalized Laplacian distance. We define the pattern-based distance GLD_{sign} by the weights

$$w_{ij} = |p_i - p_j| \qquad (25)$$

and dissimilarity function

$$f_{ij} = 1 - \text{sign}(p_i - p_j)\text{sign}(q_i - q_j) \qquad (26)$$

The dissimilarity f_{ij} penalizes opposite orders of (p_i, p_j) and (q_i, q_j). The weights w_{ij} imply that pairs with distant values in the template p_i, p_j are more important than pairs with similar values. It can be shown that

$$\arg\min_{\mathbf{q}} GLD_{sign}(\mathbf{p},\mathbf{q}) = \arg\max_{q} C_{sign}(\mathbf{p},\mathbf{q}) \qquad (27)$$

5.2. Efficient computation

The C_{sign} measure is computed efficiently by decomposing the sign function using the low rank technique described in Section 2.3. This makes it possible to rewrite Equation 24 as

$$\sum_{k=1}^{K} \sigma_k \Big(\sum_i p_i U_k[q_i] \Big) \Big(\sum_j V_k[q_j] \Big) \qquad (28)$$

$$- \sum_{k=1}^{K} \sigma_k \Big(\sum_j p_j V_k[q_j] \Big) \Big(\sum_i U_k[q_i] \Big)$$

This expression is computed by $2K$ convolutions (for the summations over $p_i U_k[q_i]$ and $p_j V_k[q_j]$) and $2K$ box filters (for the summations over $V_k[q_j]$ and $U_k[q_i]$).

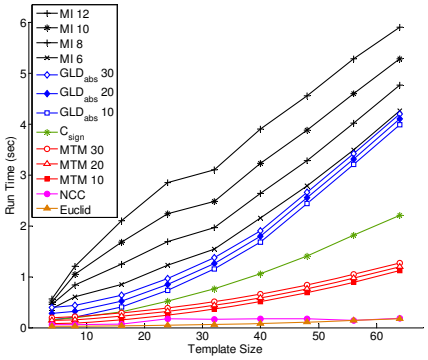


Figure 5. Average execution times for template matching using Euclidean distance, NCC, MTM, C_{sign}, GLD$_{abs}$ and mutual information (MI). We note the number of bins (K) used by each method.

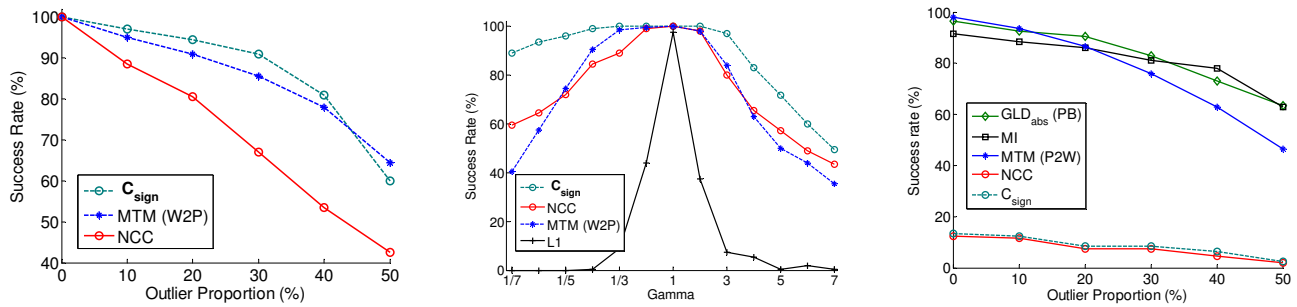
6. Results

We present experimental evaluation of the proposed GLD based template matching methods, GLD$_{abs}$ and C_{sign}, compared with other distance and similarity measures commonly used for template matching. The experiments have been carried out in various conditions of intensity change and noise distortion. In order to compare different methods properly we start with an evaluation of their execution times. Then we compare the performance of the relevant methods[1] for different intensity changes ranging from monotonic affine transformation to multimodal resources. For each of these conditions we also examine the effect of additive and outlier noise.

6.1. Speed

Figure 5 compares the runtime of the examined methods while seeking templates with different sizes in a 512×512 image. The most efficient methods are Euclidean distance, l_1 (SAD) and NCC for which fast algorithms have been proposed in the literature (see e.g. [8, 9, 12]). Computing MTM requires slightly more time. Note that the effect of the parameter K (number of used bins / slices) is almost negligible, since the total cost is a single full convolution. C_{sign} is also efficient but the parameter R (number of eigenvectors) affects the runtime as $2R$ FFT convolutions are required (we set $R = 5$). The GLD$_{abs}$ method requires $R = 4$ full convolutions. As in MTM, the number of bins K has almost no effect on the execution time. The slowest examined method is mutual information (MI) which utilizes the joint distribution of $\mathbf{p}$ and $\mathbf{q}$ values, thus needs the computation of their joint histogram. This requires K^2 sparse convolutions or, equivalently, K full convolutions. As shown by Figure 5, computing MI with a small number of bins K is slower than GLD$_{abs}$ with any number of bins.

[1] The full comparisons are available in the supplementary material.



(a) Affine transformation + outlier noise (b) Gamma correction + 10% Outliers (c) General transformation + outlier noise

Figure 4. Template matching with intensity transformations and outlier noise. The template size is fixed (24×24). (a) Affine transformations with increasing amounts of outliers (random values in random locations). (b) Gamma corrections $f(x) = x^{\gamma}, \gamma = \frac{1}{7}...7$, with 10% outliers. (c) Random non-monotonic tone mapping with increasing amounts of outliers.

The time gap is especially significant for small template size, e.g. 32×32, which is common in template matching.

6.2. Accuracy

Affine transformation. We compared the tolerance of various methods to monotonic affine transformation. 200 template locations were sampled randomly in an image dataset as well as affine transformation parameters. In order to examine only informative templates we selected locations with a sum of the absolute values of the Hessian matrix (second partial image derivatives) larger than a threshold.

Where the image is noiseless, all the compared methods perform well except the Euclidean norm and l_1 which are affine variant. While adding Gaussian noise, NCC is the best performing method and C_{sign} is less affected than the MTM variants (graph is provided in the supplementary material). However, when adding outliers (random values in random locations) C_{sign} performs much better than all the compared methods (Figure 4(a)).

Monotonic transformation. An important non linear distortion is the Gamma correction $f(x) = x^{\gamma}$. Figure 4 shows the success rate of finding the correct match in images distorted by γ values ranging from $1/7$ to 7.

In the case of distorted images with small additive Gaussian noise (std $\sigma = 15/255$) the best performing method is NCC with a small gap from C_{sign} (graph is provided in the supplementary material). Although NCC is not invariant to Gamma correction, this result can be explained by the pairwise interpretation of NCC (Equation 23) since the signs of the pairs $(p_i - p_j), (q_i - q_j)$ do not change. When 10% outliers are added, C_{sign} is the best performing method with a significant gap from NCC and the other methods (Figure 4(b). This corresponds to the invariance and robustness properties of C_{sign} as discussed in Section 5.

General transformation. We compare the performance of GLD_{abs} to MTM and MI, which have been shown in [7] to be very tolerant to general continuous intensity transformations. 200 tone mapping functions and informative template locations were sampled randomly and used for three experiments.

First, we added small Gaussian noise ($\sigma = 15/255$) and examined the performance for different template sizes. The performance of MTM and GLD_{abs} is similar and slightly better than MI (graph is provided in the supplementary material).

Second, templates of size 24×24 were sought in images with increasing amounts of additive Gaussian noise. Here the performance of MTM and GLD_{abs} is also similar, and significantly better than MI with the same execution time (graph is provided in the supplementary material).

Third, templates of size 24×24 were sought in images with increasing amounts of outliers, i.e. random values in random locations. The performance of GLD_{abs} and MI are similar and better than MTM, as shown in Figure 4(c). An example of template matching with both additive and outlier noise is shown in Figure 3.

In all these experiments, faster methods such as NCC and C_{sign} totally fail since they are not designed to overcome non monotonic intensity change. The P2W variant of MTM and the PB variant of GLD_{abs} perform better than the W2P and WB variants (shown in the supplementary material). The reason is that in the above experiments there exists a tone mapping T which maps the template to the correct match (up to noise), but there might be no injective mapping from the mapped image back to the template.

Multimodal template matching. Finding a small template in an image from a different source (modality) is a difficult task. Usually, alignment of two large multimodal images is performed by maximizing the mutual information (MI) between them. Template matching is more challenging since the joint statistics of a small template and candidate windows are less reliable.

In our experiment, 400 image-template pairs were sampled randomly from a dataset of aligned multimodal

(a) CT (b) Visual (c) Optic

(d) MRI (e) IR (f) SAR

Figure 6. Sample images from the multimodal image dataset.

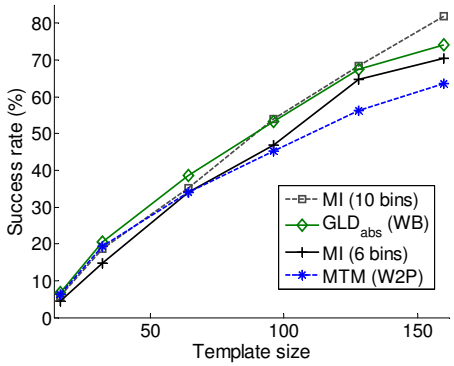


Figure 7. Results of multimodal template matching. The success rate of the proposed generalized Laplacian distance (GLD$_{abs}$) is comparable with mutual information (MI) in less time (see Figure 5) .

images (see Figure 6). We compare the performance of GLD$_{abs}$ with those of MTM and MI. The results are shown in Figure 7. GLD$_{abs}$ has better performance than MI with 6 bins which has the same runtime, and comparable performance with MI with 10 bins that is much slower (see Figure 5).

In this experiment the W2P variant of MTM and the WB variant of GLD$_{abs}$ perform better than P2W and PB (see the supplementary material). Although there is no injective mapping either from the template to the image or vice versa, fitting the candidates to the template (as done by W2P and WB) is more successful than fitting the template to each candidate window (as done by P2W and PB). Intuitively, fitting the template to an incorrect window (e.g. flat area) by a degenerate tone mapping (e.g. a constant mapping) is more probable than fitting an incorrect candidate to an informative template where mapping is rarely available.

7. Summary

We introduced the Generalized Laplacian Distances (GLDs), a distance family related to the graph Laplacian. The GLD framework makes it possible to capture different types of visual similarity, which is demonstrated for template matching in challenging conditions such as complicated intensity transformations, outlier noise and multimodal matching. For fast computation of GLDs we developed a novel algorithm based on linear decomposition of distance functions. We plan to extend the use of GLD also for geometrical distortions, as well as for other computer vision applications.

Acknowledgments. This research was supported in part by the Israel Ministry of Science and Technology, the Israel Science Foundation and the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI).

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *NIPS*, 14:585–591, 2001.

[2] R. Brunelli. *Template matching techniques in computer vision*. Wiley, 2009.

[3] F. Chung. *Spectral graph theory*. The American Mathematical Society, 1997.

[4] F. Crow. Summed-area tables for texture mapping. *Computer Graphics*, 18(3):207–212, 1984.

[5] E. Elboher and M. Werman. Cosine integral images for fast spatial and range filtering. In *ICIP*, 2011.

[6] X. He, S. Yan, Y. Hu, et al. Face recognition using laplacianfaces. *PAMI*, 27(3):328–340, 2005.

[7] Y. Hel-Or, H. Hel-Or, and E. David. Fast template matching in non-linear tone-mapped images. In *ICCV*, pages 1355–1362, 2011.

[8] J. P. Lewis. Fast normalized cross-correlation. *Vision Interface*, 10:120–123, 1995.

[9] A. Mahmood and S. Khan. Exploiting transitivity of correlation for fast template matching. *Trans. on Image Processing*, 19(8):2190–2200, 2010.

[10] A. Ng, M. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. *NIPS*, 2:849–856, 2002.

[11] P. Niyogi. Locality preserving projections. *NIPS*, 16:153–160, 2004.

[12] W. Ouyang, F. Tombari, S. Mattoccia, et al. Performance evaluation of full search equivalent pattern matching algorithms. *PAMI*, 34(1):127–143, 2012.

[13] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000.

[14] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *CVPR*, 1:511–518, 2001.

[15] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[16] M. Zibulevski. Adjoint convolution 2d convolution, efficient mex implementation: Version 29.11.2010. Code at `http://ie.technion.ac.il/~mcib/`.