# Fast Template Matching in Non-Linear Tone-Mapped Images

Yacov Hel-Or
IDC Herzelia
toky@idc.ac.il

Hagit Hel-Or
University of Haifa
hagit@cs.haifa.ac.il

Eyal David
University of Haifa
eyald@cs.haifa.ac.il

## Abstract

*We propose a fast pattern matching scheme termed* Matching by Tone Mapping (MTM) *which allows matching under non-linear tone mappings. We show that, when tone mapping is approximated by a piecewise constant function, a fast computational scheme is possible requiring computational time similar to the fast implementation of Normalized Cross Correlation (NCC). In fact, the MTM measure can be viewed as a generalization of the NCC for non-linear mappings and actually reduces to NCC when mappings are restricted to be linear. The MTM is shown to be invariant to non-linear tone mappings, and is empirically shown to be highly discriminative and robust to noise.*

## 1. Introduction

Template or pattern matching is a basic and fundamental image operation. In its simple form a given pattern is sought in an image, typically by scanning the image and evaluating a similarity measure between the pattern and every image window. Fast and reliable pattern matching is a basic building block in a vast range of applications, such as: image denoising, image re-targeting and summarization, image editing, super-resolution, object tracking and object recognition, just to name a few (e.g. [3, 16, 9, 1]).

In most cases, however, the input image is acquired in an uncontrolled environment, thus, the sought pattern may vary in tone-levels due to changes in illumination conditions, camera photometric parameters, viewing positions, different modalities, etc. [11]. Commonly, these changes can be modeled locally by a non-linear tone mapping - a functional mapping between the gray-levels of the sought pattern and those of the image pattern. In this paper we deal with pattern matching where gray-levels may be subject to some unknown, possibly non-linear, tone mapping.

Many distance measures for pattern matching have been suggested in the literature and the interested reader is referred to [4, 2] for excellent reviews. By far, the most common measure is the Euclidean distance. Assume the pattern $\mathbf{p}$ and the candidate window $\mathbf{w}$ are both vectors in $\mathcal{R}^m$, (e.g. by raster scanning the pixels). The Euclidean distance between $\mathbf{p}$ and $\mathbf{w}$ is denoted: $d_E(\mathbf{p},\mathbf{q}) = \|\mathbf{p}-\mathbf{w}\|_2$. Searching for the pattern over the entire image is performed by scanning the image and determining the minimal $d_E$ value. This search can be applied very fast using efficient convolution operations [5]. Nevertheless, although very common, the Euclidean distance assumes no tone mapping or, equivalently, that the tone mapping between $\mathbf{p}$ and $\mathbf{w}$ is the identity mapping. Clearly, the Euclidean distance is inadequate when the image undergoes tone deformations as demonstrated in Figure 1.

To overcome gray tone mapping effects in images, the *normalized cross correlation* (NCC) distance is often used [2]. Consider the pattern $\mathbf{p}$ and the candidate window $\mathbf{w}$ as random variables with samples $p_i$ and $w_i$, $i = 1..m$, respectively. The NCC is then defined as:

$$\rho(\mathbf{p},\mathbf{w}) = E\left[\left(\frac{\mathbf{p}-E[\mathbf{p}]}{\sqrt{\text{var}(\mathbf{p})}}\right)\left(\frac{\mathbf{w}-E[\mathbf{w}]}{\sqrt{\text{var}(\mathbf{w})}}\right)\right]$$

where for any vectors $\mathbf{x} \in \mathcal{R}^m$, $E[\mathbf{x}]$ and $\text{var}(\mathbf{x})$ denote the sample mean and variance. Due to the substraction of the mean and normalization by the st.d. in both $\mathbf{p}$ and $\mathbf{w}$, the NCC distance is invariant to linear tone mappings (Figure 1). The NCC distance can be applied very efficiently requiring little more than a single convolution on the input image [10]. However, such a distance will fail to detect patterns in cases where non-linear tone mappings have been applied (Figure 1 bottom row)[1].

Finally, when non-linear mapping is considered, the *Mutual Information* (MI) is commonly used, initially proposed for image registration [18]. MI measures the statistical dependency between two variables. Clearly, the statistical dependency is strong when gray-levels of one image result from a functional mapping of the gray-levels of the other image. Thus, MI can account for non-linear mappings (both monotonic and non-monotonic as in Figure 1).

In the context of pattern matching, MI measures the loss of entropy in the pattern $\mathbf{p}$ given a candidate window $\mathbf{w}$:

$$MI(\mathbf{p},\mathbf{w}) = H(\mathbf{w}) - H(\mathbf{w}|\mathbf{p}) = H(\mathbf{w}) + H(\mathbf{p}) - H(\mathbf{w},\mathbf{p})$$

[1] NCC often performs well even under monotonic non-linear mappings as these can often be assumed to be locally linear.

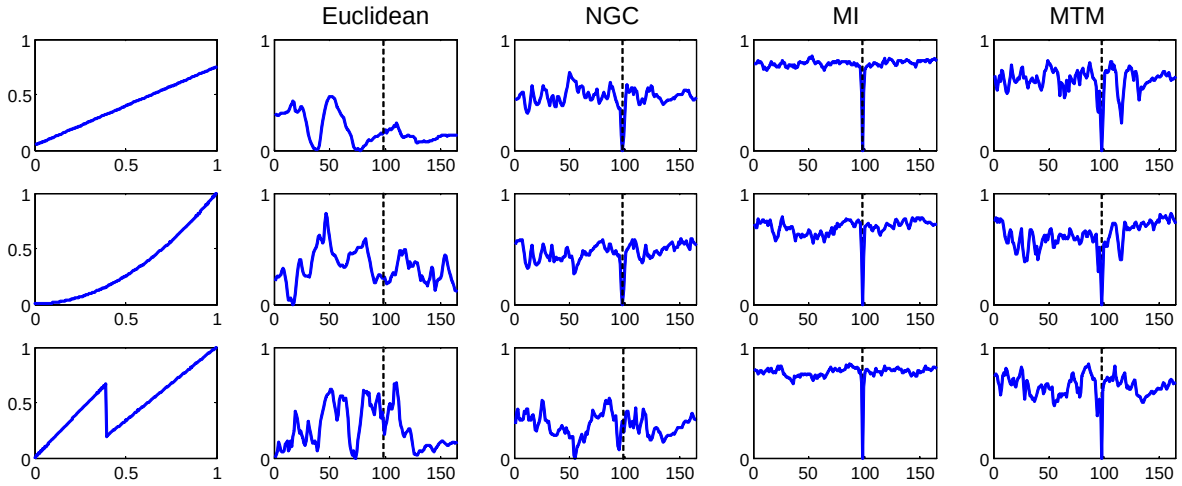


Figure 1. Pattern Matching under tone mapping. Euclidean, NCC, MI and MTM distance measures were used to evaluate distances between a pattern and all image windows for different tone mappings. To simplify visualization results are shown as the minimum distance value between pattern and image windows along image columns. Distance values have been normalized to the range $[0, 1]$ for easier comparison. Lower values indicate greater similarity of the window to the pattern. The dashed line marks the column of the correct match. Rows (top to bottom): Linear, Monotonic and Non-Monotonic tone mappings. Columns (left to right): Tone mapping function, followed by plots for Euclidean, NCC, MI (using bins of 20 tone-values) and MTM (using bins of 20 tone-values) distances.

where $H$ is the differential entropy.

Although MI is an effective distance measure that can account for non-linear mappings, it is hindered by computational issues in the context of pattern matching. First, it is computationally expensive as it requires the construction of the joint histogram (pattern vs. window) for each window to be matched. Although fast methods for evaluating histograms on running windows have been suggested [12, 19], fast methods for local *joint* histograms are yet a challenge. Additionally, entropy as well as MI is very sensitive to the the size of histogram bins used to estimate the joint density, especially when sparse value samples are given (small pattern size). Using kernel density estimation methods [17] rather than discrete histograms is, again, computationally expensive when dealing with joint probability, not to mention its sensitivity to the kernel width.

In this paper we propose a very fast pattern matching scheme based on a distance measure expressed as a minimization problem over all possible tone mappings, thus, we term the resulting measure *Matching by Tone Mapping (MTM)*. From its definition, the MTM is invariant to non-linear tone mappings and can be viewed as a generalization of the NCC for non-linear mappings and actually reduces to NCC when mappings are restricted to be linear [14]. In fact, for the general case, it can be shown that the MTM measure coincides with the statistical measure known as the correlation-ratio [6]. This measure was previously suggested as an image similarity measure in [13, 15]. We empirically show that the MTM is highly discriminative and robust to noise with comparable performance capability to that of the well performing Mutual Information. Thus, the MTM allows a pattern matching scheme on par with NCC in terms of computation time but with performance capability equivalent to that of the Mutual Information scheme.

## 2. Matching by Tone Mapping

In the proposed pattern matching scheme, we wish to evaluate the minimum distance between a pattern and a candidate window under all possible tone mappings. Since tone mapping is not necessarily a bijective mapping, two alternatives may be considered: i) tone mapping applied to the pattern, transforming it to be as similar as possible to the candidate window, and ii) tone mapping applied to the window, transforming it to be as similar as possible to the pattern. For each case we find the minimum normed distance over all possible tone mappings.

Let $\mathbf{p} \in \mathcal{R}^m$ be a pattern and $\mathbf{w} \in \mathcal{R}^m$ a candidate window to be compared. Denote by $\mathcal{M} : \mathcal{R} \rightarrow \mathcal{R}$ a tone mapping function. Thus, $\mathcal{M}(\mathbf{p})$ represents the tone mapping applied independently to each entry in $\mathbf{p}$. For the case of tone mapping applied to the pattern, the MTM distance is defined as follows:

$$D(\mathbf{p}, \mathbf{w}) = \min_{\mathcal{M}} \left\{ \frac{\| \mathcal{M}(\mathbf{p}) - \mathbf{w} \|^2}{m \text{ var}(\mathbf{w})} \right\} \quad (1)$$

Similarly, if the mapping is applied to the window rather than the pattern, we define:

$$D(\mathbf{w}, \mathbf{p}) = \min_{\mathcal{M}} \left\{ \frac{\| \mathcal{M}(\mathbf{w}) - \mathbf{p} \|^2}{m \text{ var}(\mathbf{p})} \right\} \quad (2)$$

The numerator in both cases is simply the norm distance after compensating for the tone mapping. The denominator

is a normalization factor enforcing the distance to be scale invariant. Thus $D(\mathbf{p},\mathbf{w}) = D(\mathbf{p},\alpha\mathbf{w})$ for any scalar $\alpha$. Additionally, it penalizes incorrect matching of $\mathbf{p}$ to smooth windows when using the constant mapping $\mathcal{M}(\mathbf{p}) = c$. Due to the tone mapping compensation, the MTM measure reflects the inherent *structural* difference between the pattern $\mathbf{p}$ and the window $\mathbf{w}$.

Searching for the pattern in the entire input image requires calculating the optimal tone mapping for each possible window in the image. Although seemingly a computationally expensive process, we show in the following sections that in fact this distance can be calculated very efficiently requiring an order of a single convolution with the input image!

In the next section we introduce the *Slice Transform* [8]. This transform enables the representation of tone mappings in a linear form allowing a closed form solution for the defined MTM distance.

### 2.1. The Slice Transform (SLT)

The *Slice Transform (SLT)* was first introduced in [8] in the context of Image Denoising. In this paper we exploit the SLT to represent a mapping function using a linear sum of basis functions. Consider an image segment represented as a column vector $\mathbf{x} = [x_1, \cdots, x_m]^T$ with values in the half open interval $[a, b)$. The interval is divided into $k$ bins with boundary values $q_1 \cdots q_{k+1}$ such that:

$$a = q_1 < q_2 < \ldots < \quad q_{k+1} = b$$

Any value $v \in [a, b)$ is naturally associated with a single bin $\pi(v) \in \{1 \cdots k\}$:

$$\pi(v) = i \quad \text{if} \quad v \in [q_i, q_{i+1})$$

Given the bins defined by $\{q_i\}$, the vector $\mathbf{x}$ can be decomposed into a collection of binary *slices*: Slice $\mathbf{x}^i = [x^i_1, \cdots, x^i_m]$ is an indicator function over the vector $\mathbf{x}$ representing the entries of $\mathbf{x}$ associated with the $i$-th bin.

$$x^i_j = \begin{cases} 1 & \text{if } \pi(x_j) = i \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The vector $\mathbf{x}$ can then be approximated as a linear combination of slice images:

$$\mathbf{x} \approx \sum_{i=1}^{k} \alpha_i \mathbf{x}^i \tag{4}$$

where the weights $\{\alpha_i\}_{i=1}^k$ are the values assigned for each bin (e.g $\alpha_i = q_i$ or $\alpha_i = (q_i + q_{i+1})/2$). The approximation is in fact a quantization of the values of $\mathbf{x}$ into the bins represented by $\{\alpha_i\}$. The greater the number of bins the better the approximation of the original image. Specifically, if $\mathbf{x}$ values are discrete and all values are in $\{\alpha_i\}_{i=1}^k$ then $\mathbf{x} = \sum_{i=1}^k \alpha_i \mathbf{x}^i$.
Collecting the slices $\mathbf{x}^i$ in columns, we define the *SLT matrix* of $\mathbf{x}$:

$$S(\mathbf{x}) = [\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^k] \tag{5}$$

Then Equation 4 can be rewritten in matrix form:

$$\mathbf{x} \approx S(\mathbf{x})\alpha \tag{6}$$

where we define $\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_k]^T$. Note, that since the slices are mutually disjoint, the columns of $S(\mathbf{x})$ are mutually orthogonal, satisfying:

$$\mathbf{x}^i \cdot \mathbf{x}^j = |\mathbf{x}^i|\, \delta_{i,j} \tag{7}$$

where $'\cdot'$ is the vectorial inner product, $|\mathbf{x}|$ counts the non-zero enteries in $\mathbf{x}$ and $\delta_{i,j}$ is the Kronecker's delta. The SLT matrix enables the representation of any piece-wise constant mapping of $\mathbf{x}$. In fact, substituting the vector $\alpha$ in Equation 6 with a different vector $\beta$, we obtain

$$\mathbf{y} = S(\mathbf{x})\beta \tag{8}$$

Image $\mathbf{y}$ is a piecewise constant tone mapping of $\mathbf{x}$ s.t. all pixels of $\mathbf{x}$ with values in the $j$-th bin are mapped to $\beta_j$. Thus, the columns of $S(\mathbf{x})$ form an orthogonal basis spanning the space of all images that can be produced by applying a piecewise constant tone mapping on $\mathbf{x}$. Figure 2 illustrates an SLT matrix and a piecewise mapping of a 5-pixel signal with 3 gray-level values. Figure 3 shows an example of linearly combining image slices to form the original (quantized) image and to form a tone mapped version.

In the context of this paper, we use the SLT for tone mapping approximation. A mapping applied to pattern $\mathbf{p}$ is approximated by a piecewise constant mapping:

$$\mathcal{M}(\mathbf{p}) \approx S(\mathbf{p})\beta$$

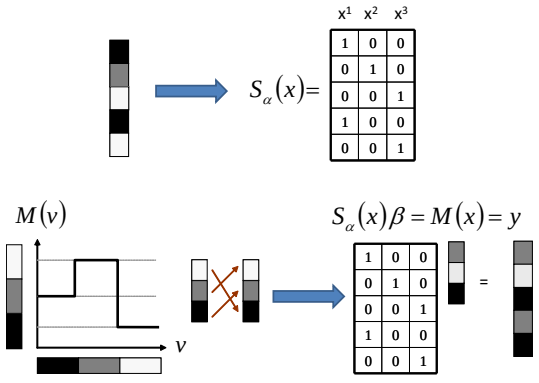


Figure 2. Top: the SLT matrix for a 5-pixel vector having 3 gray values. Bottom: a piecewise constant mapping and its representation using the SLT matrix.

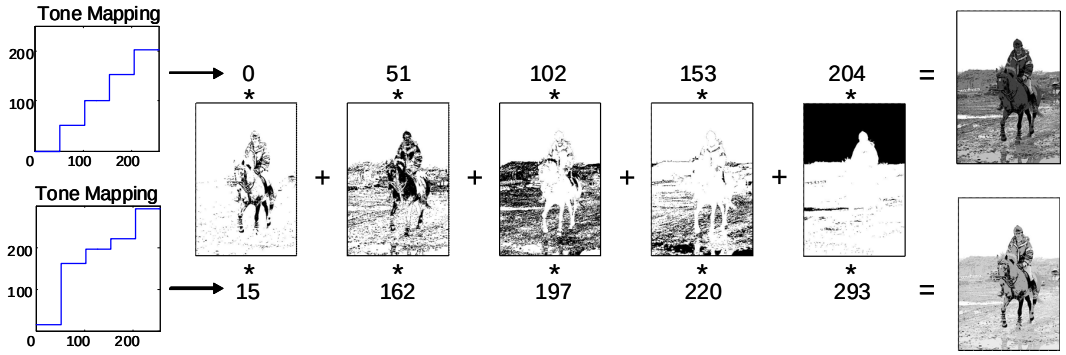


Figure 3. Linear Combination of image slices. The SLT transform was applied to an image using 5 bins defined by $\alpha = [0, 51, 102, 153, 204, 256]$. Top: Reconstructing the original images using $\alpha$ values as weights in the linear combination. Bottom: Using weights other than $\alpha$ produces a tone mapped version of the original image. Slice images are shown inverted (1=black, 0=white).

Consequently, the distance measures as defined in Equations 1-2 can be rewritten using the SLT:

$$D(\mathbf{p}, \mathbf{w}) = \min_{\beta} \frac{\| S(\mathbf{p})\beta - \mathbf{w} \|^2}{m \text{ var}(\mathbf{w})} \tag{9}$$

or alternatively

$$D(\mathbf{w}, \mathbf{p}) = \min_{\beta} \frac{\| S(\mathbf{w})\beta - \mathbf{p} \|^2}{m \text{ var}(\mathbf{p})} \tag{10}$$

where $S(\mathbf{p})$ and $S(\mathbf{w})$ are the SLT matrices as defined in Equation 5. In the sections below we elaborate on the two possible alternative distances defined above. We show that solving for $D$ for each image window can be applied in a very efficient computation scheme. In fact, computing $D$ for the entire image requires on the order of a single image convolution.

## 2.2. MTM Distance Measure using SLT

The SLT scheme allows a closed form solution for the minimizations defined in Equations 1 and 2. To introduce the matching process, we first consider the case where a pattern $\mathbf{p}$ is to be matched against a candidate window $\mathbf{w}$. Thus, the distance measure used is that given in Equation 9. To simplify notation, we henceforth denote the SLT matrix $S(\mathbf{p})$ as $S$. The solution for $\beta$ that minimizes Equation 9 is given by:

$$\hat{\beta} = \arg\min_{\beta} \|S\beta - \mathbf{w}\|^2 = S^{\dagger}\ \mathbf{w}$$

where $S^{\dagger} = (S^T\ S)^{-1}\ S^T$ is the Moore-Penrose pseudo-inverse. Substituting into Equation 9 we obtain:

$$D(\mathbf{p}, \mathbf{w}) = \frac{\| S\hat{\beta} - \mathbf{w}\|^2}{m \text{ var}(\mathbf{w})} = \frac{\|S(S^TS)^{-1}S^T\mathbf{w} - \mathbf{w}\|^2}{m \text{ var}(\mathbf{w})}$$

Due to the orthogonality of $S$, we have that $S^TS = G$ is a diagonal matrix with the histogram of $\mathbf{p}$ along its diagonal: $G(i,i) = |\mathbf{p}^i|$ where $\mathbf{p}^i$ is the pattern slice associated with the $i$-th bin as defined in Equation 3. Expanding the numerator it is easy to verify that:

$$\|S(S^TS)^{-1}S^T\mathbf{w} - \mathbf{w}\|^2 = \|\mathbf{w}\|^2 - \|G^{-1/2}S^T\mathbf{w}\|^2$$

Since $S = \left[\mathbf{p}^1, \mathbf{p}^2, \cdots, \mathbf{p}^k\right]$ and exploiting the diagonality of $G$, the above expression can be re-written using a sum of inner-products:

$$\|\mathbf{w}\|^2 - \|G^{-1/2}S^T\mathbf{w}\|^2 = \|\mathbf{w}\|^2 - \sum_j \frac{1}{|\mathbf{p}^j|}(\mathbf{p}^j \cdot \mathbf{w})^2$$

As a result, the overall MTM distance $D(\mathbf{p}, \mathbf{w})$ reads:

$$D(\mathbf{p}, \mathbf{w}) = \frac{1}{m \text{ var}(\mathbf{w})} \left[ \|\mathbf{w}\|^2 - \sum_j \frac{1}{|\mathbf{p}^j|}(\mathbf{p}^j \cdot \mathbf{w})^2 \right] \tag{11}$$

When matching is applied by mapping $\mathbf{w}$ towards $\mathbf{p}$, we use Equation 10 and exchange the role of $\mathbf{w}$ and $\mathbf{p}$ to obtain a symmetric expression:

$$D(\mathbf{w}, \mathbf{p}) = \frac{1}{m \text{ var}(\mathbf{p})} \left[ \|\mathbf{p}\|^2 - \sum_j \frac{1}{|\mathbf{w}^j|}(\mathbf{w}^j \cdot \mathbf{p})^2 \right] \tag{12}$$

The expression in brackets in Eq. 11 can be shown to be related to the conditional variance $E(\text{var}(\mathbf{w}|\mathbf{p}))$. When dividing by $\text{var}(\mathbf{w})$ the expression coincides with the definition of the correlation ratio [6]. The same is applicable to Equation 12.

## 2.3. Calculating MTM Distance Over an Image

Equations 11 and 12 provide a method for computing the structural difference between $\mathbf{p}$ and $\mathbf{w}$ using two complementary distances. For pattern matching, this computation must be performed on each candidate window of a given input image. Naively applying the above expressions to each image window is highly time consuming and impractical.

In the following we show that, in fact, computing $D(\mathbf{p}, \mathbf{w})$ or $D(\mathbf{w}, \mathbf{p})$ over an entire image can be calculated very efficiently. We first describe the pattern-to-window mapping case, and then detail the window-to-pattern case.

Let $\mathbf{F}$ be a 2D image with $n$ pixels in which the pattern $\mathbf{p}$ with $m$ pixels is sought. Denote by $\mathbf{w}_r$ the $r$-th window of $\mathbf{F}$, $r \in \{1, \cdots, n\}$. Consider the pattern-to-window (P2W) scheme where the distance given in Equation 11 is used. For each window $\mathbf{w}_r \in F$ two terms must be calculated, namely the numerator and the denominator of Equation 11:

$$d_1(r) = \|\mathbf{w}_r\|^2 - \sum_j \frac{1}{|\mathbf{p}^j|}(\mathbf{p}^j \cdot \mathbf{w}_r)^2 \quad \text{and} \quad d_2(r) = m \text{ var}(\mathbf{w}_r)$$

Note, that $\text{var}(\mathbf{w}_r) = E\left[\mathbf{w}_r^2\right] - E^2\left[\mathbf{w}_r\right]$, thus

$$d_2(r) = m \text{ var}(\mathbf{w}_r) = \mathbf{1} \cdot (\mathbf{w}_r \odot \mathbf{w}_r) - (\mathbf{1} \cdot \mathbf{w}_r)^2/m$$

where $\mathbf{1}$ is an m-vector of 1's (box filter), $\mathbf{x} \cdot \mathbf{y}$ is the inner product, and $\mathbf{x} \odot \mathbf{y}$ is the elementwise multiplication of vectors $\mathbf{x}$ and $\mathbf{y}$. Algorithm 1 gives the pseudo-code for calculating the P2W MTM distance between pattern $\mathbf{p}$ and each window in $F$. Code can be found in [7]. In the pseudo-code '$*$' denotes image convolution[2], $\odot$ and $\oslash$ denote elementwise multiplication and division, respectively. We denote by upper-case letters arrays of size similar to that of the image $F$, and by lower-case letters scalar variables. Vectors are denoted by bold lower-case letters.

**Algorithm 1** MTM - Pattern-to-Window

{**Input**: pattern $\mathbf{p}$, image $F$.}
{**Output**: image $D$ of MTM distances.}
$W_1 := \mathbf{1} * F$ {window's sum}
$W_2 := \mathbf{1} * (F \odot F)$ {window's sum of squares}
$D_2 := W_2 - (W_1 \odot W_1)/m$ {calc $d_2$ (denominator)}
Generate $\{\mathbf{p}^j\}$, for $j = 1..k$
$D_1 := 0$ {will accumulate the numerator}
**for** $j := 1$ to $k$ **do**
  $n_j = \mathbf{1} \cdot \mathbf{p}^j$, {calc $|\mathbf{p}^j|$}
  $T := flip(\mathbf{p}^j) * F$ {convolve image with slice $j$}
  $T := (T \odot T)/n_j$
  $D_1 := D_1 + T$
**end for**
$D := (W_2 - D_1) \oslash D_2$
**return** D

Prior to the loop, two convolutions with a box filter are calculated, each of which can be applied efficiently (with a separable 1D box filter) requiring a total of 4 additions per pixel. Within the loop there are $k$ convolutions with the pattern slices $\{\mathbf{p}^j\}_{j=1}^k$. Since each slice $\mathbf{p}^j$ is sparse, convolving it with an image requires only $|\mathbf{p}^j|$ additions per pixel using a sparse convolution scheme [20]. Additionally, since all pattern slices are mutually disjoint the number of additions sum up to a total of $m$ additions per pixel. All other operations sum to $O(k)$ operations per pixel, thus, the algorithm requires a total of $O(mn + kn)$ operations which is comparable in complexity to a single convolution! Memory requirement is also economized. Distance value for each image window is accumulated in place, thus the required memory is on the order of the image size.

Consider now the window-to-pattern (W2P) scheme using the distance given in Equation 12. For each window $\mathbf{w}_r \in F$, the expressions to be calculated are:

$$d_1(r) = \|\mathbf{p}\|^2 - \sum_j \frac{1}{|\mathbf{w}_r^j|}(\mathbf{w}_r^j \cdot \mathbf{p})^2 \quad \text{and} \quad d_2 = m \text{ var}(\mathbf{p})$$

$d_2$ and the first term of $d_1$ are constant for all windows and are calculated only once. The second term in $d_1$ differs for each window. Algorithm 2 gives the pseudo-code for calculating the W2P distance over the entire image. In this algorithm $F^j$ denotes the $j$-th image slice, i.e. $F^j(x, y) = 1$ iff $\pi(F(x, y)) = j$. Since each $F^j$ is a sparse image, convolution can be applied efficiently in this case as well. Since $\{F^j\}$ are mutually disjoint, the operations required for the $k$ sparse convolutions sum to $O(mn)$ operations. As in the P2W case, the entire algorithm requires $O(mn + kn)$ operations, which is on the order of a single image convolution. Memory requirement is also economical and is on an order order of the image size.

**Algorithm 2** MTM - Window-to-Pattern

{**Input**: pattern $\mathbf{p}$, image $F$.}
{**Output**: image $D$ of MTM distances.}
$p_1 := \mathbf{1} \cdot \mathbf{p}$ {pattern's sum}
$p_2 := \mathbf{1} \cdot (\mathbf{p} \odot \mathbf{p})$ {pattern's sum of squares}
$d_2 := p_2 - p_1^2/m$ {computation of $d_2$ (denominator)}
Generate $\{F^j\}$, for $j = 1..k$
$D_1 := 0$ {will accumulate the numerator}
**for** $j := 1$ to $k$ **do**
  $N_j := \mathbf{1} * F^j$ {calc $|\mathbf{F}^j|$}
  $T := flip(\mathbf{p}) * F^j$ {convolve image slice with $\mathbf{p}$}
  $T := (T \odot T) \oslash N_j$
  $D_1 := D_1 + T$
**end for**
$D := (p_2 - D_1)/d_2$
**return** D

## 3. Results

The MTM was tested in the context of pattern matching. Performance was compared with the Euclidean distance, Normalized Cross Correlation (NCC), and Mutual Information (MI). In this section we show that even under extreme

[2] In fact, correlations rather than convolutions are required, thus, a flipped kernel is used when needed.

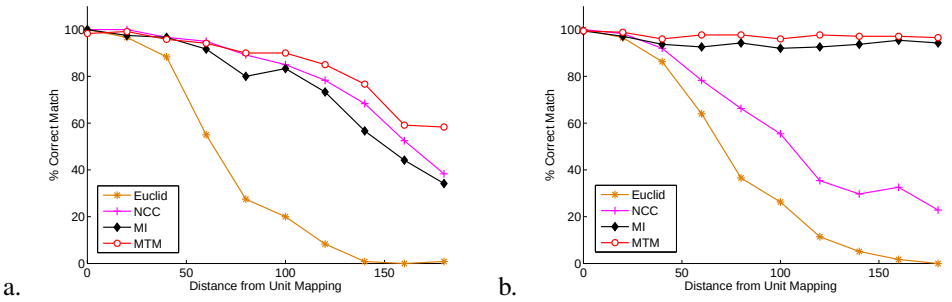


Figure 4. Pattern detection performance vs. extremity of the tone mapping for monotonic mapping (a) and non-monotonic mapping (b).

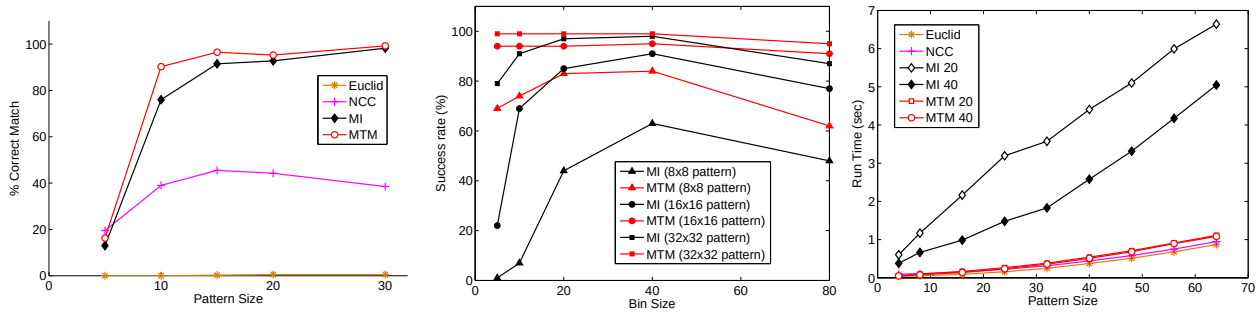


Figure 5. Left: Pattern detection performance as a function of pattern size. Middle: Performance comparison of MI and MTM as a function of bin size. Right: Algorithm run time for pattern matching under various schemes (and different bin sizes) for different pattern sizes.

tone mappings and under heavy noise conditions, the MTM approach successfully and efficiently detects the sought patterns, performing significantly better than the compared methods with run times on par with the fast NCC.

Pattern matching was applied on a large set of randomly selected image-pattern pairs under various conditions. The image set consisted of 100 natural images selected and downloaded from Flickr. The image set was selected to include urban, nature, people, and indoor scenes. For each input image, a pattern of a given size was selected at a random location. To prevent smooth or "non-interesting" patterns, these were selected from among patterns with saliency above a specific threshold. Saliency was measured at each pixel as the direct sum of the structure tensor's eigen-values (a 2x2 matrix of partial derivatives) calculated for a small window about the pixel location. Given an image and a selected pattern, a random tone mapping was applied to the image (with possible additive noise) and the original selected pattern was then sought in the mapped image. Distances were calculated for all possible locations in the mapped image, and the window associated with the minimal distance was considered the *matched window*. If the matched window was detected at the correct position the match is considered a *correct detection*.

Figure 4 displays the detection rate as a function of the extremity of the tone mapping applied to the image for the 4 distance measures. Extremity of tone mapping was measured as the mean squared distances between the original range of values ( $[0..255]$) and the mapped tone values. Results are shown separately for monotonic mappings (Figure 4a) and for non-monotonic mappings (Figure 4b). Each data point represents the detection rate (in percentages) over 2000 randomly selected image-pattern pairs. Images were of size $200 \times 200$ and patterns of size $20 \times 20$. Tone mappings were generated by randomly selecting six new tone values serving as the mapping values for six equally spaced source tone values (in the range $[0..255]$). The tone mapping was defined as a piecewise linear function passing through the selected values. For monotonic mappings the randomly selected tone values were sorted in increasing order prior to the construction of the tone mapping. Additive Gaussian noise with s.t.d. of 15 gray-values was added to each mapped image before pattern matching was performed.

We note an important caveat with respect to the mappings and their extremity measure: in the case of monotonic mappings, the monotonicity constrains the extreme mappings and typically produces deeply convex or concave mappings which imply loss of spatial details in certain image regions. This in turn increases the difficulty of correctly detecting a given pattern. Non-monotonic mappings on the other hand, produce false contours but typically maintain the image structure (preserve original edges though possibly increasing or decreasing their contrast). Thus, as will be seen, performance under monotonic mappings is often

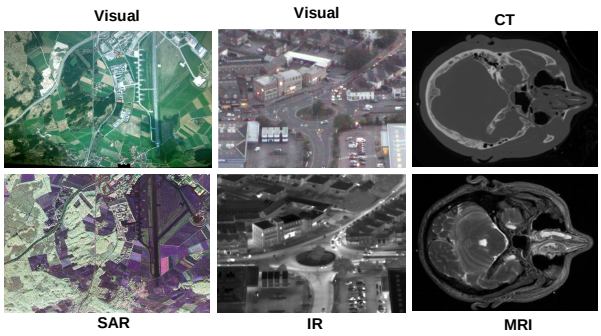


Figure 6. Examples from the tested set of images from different modalities.

degraded compared to non-monotonic mappings.

Figure 4 shows that performance using the Euclidean distance degrades very fast with mapping extremity. The NCC is expected to fail in both monotonic and non-monotonic mappings, however in the monotonic case, mapping is smooth and can often be approximated locally as linear. Thus, NCC performs relatively well under monotonic mappings compared to the non-monotonic mappings. The MTM and MI results are shown for bin size of 20 gray tones. It can be seen that the MTM performs very well and on par with the MI approach. Both, MTM and MI perform better under non-monotonic mappings compared with monotonic mappings due to the caveat described above.

We examined the robustness of the mentioned distance measures to additive noise and their performance under different pattern sizes. Figure 5-left shows the detection rate for various pattern sizes under a specific non-linear mapping. All images were contaminated with Gaussian noise with s.t.d. of 15. A set of 2000 pattern-image pairs were randomly selected using the procedure described above. Both the MTM and MI methods used bin size of 20. It can be seen that for small patterns (under $10 \times 10$ pixels) detection rates are very low in all methods. This behavior stems from the fact that histogram bins of small sized patterns are sparsely populated if at all. This may produce an underdetermined minimization or an over-fitting phenomena. For this reason, techniques using a low number of free parameters are preferable and outperform other methods, as long as they can model accurately all possible tone mappings (e.g. NCC for monotonic or linear mappings).

## 4. MTM v.s. MI

It can be shown that the MTM and MI approaches are similar in spirit. While MI maximizes the entropy reduction in $\mathbf{w}$ given $\mathbf{p}$, MTM maximizes the variance reduction in $\mathbf{w}$ given $\mathbf{p}$. The entropy and variance are two different measures of uncertainty. While variance is a quantitative measure preferring a compact distribution of samples, the entropy is a qualitative measure disregarding bin rearrangements. The use of variance rather than entropy is advantageous when a small number of samples are available. This is demonstrated in Figure 5-middle where detection rates are displayed for different pattern and bin sizes. Compared to MTM, MI shows larger sensitivity to bin size with a decrease in performance for smaller bin sizes. Results are shown for different pattern sizes ($8\times8$, $16\times16$ and $32\times32$). Each data point results from 200 randomly selected image-pattern pairs. For every image-pattern pair, a random monotonic mapping was generated, within extremity range of 40-60, and added Gaussian noise (std = 20).

A major, significant, advantage of MTM over MI is its computational efficiency. Figure 5 displays run times of pattern matching using different schemes under varying pattern sizes. For MI and MTM, run times are shown for different bin sizes as well. Run times shown are the average over 10 runs. Since MI requires the computation of the joint histogram for every image window pair, it is computationally more expensive than MTM and other approaches. Furthermore, run time for MI increases with number of bins. On the other hand, run time of the MTM scheme is of the order equivalent to a single image convolution (Section 2.3) and thus on par with the NCC and Euclidean approaches. Bin size has very little effect on the run time of MTM.

Although MTM demonstrates superior performance with respect to run time and stability under sparse samples, it relies on functional dependency between $\mathbf{p}$ and $\mathbf{w}$. When this assumption is violated, e.g. in multi-modal images (between which functional mapping does not necessarily exist), MI often outperforms MTM, although at the expense of longer running time.

To compare the capabilities of the MTM and MI in multi modality scenarios we evaluated the performance of pattern matching and image alignment between image pairs originating from different modalities, including: visual - SAR, visual - InfraRed, CT - MRI and CT - PET. Examples from the tested set are shown in Figure 6. These images display tone changes between their counterparts which are one-to-many, thus, cannot be regarded as a mapping function.

Figure 7 displays the detection rates of pattern matching between multi-modal images for various pattern sizes. The recorded performance is an average over five multi-modal image pairs where 100 (randomly selected) pattern-window cases were tested for each pair. It is demonstrated that MI outperforms MTM and NGC for almost all pattern sizes. While NGC severely fails in almost all pattern sizes, MTM presents reasonable performance although inferior to MI.

With respect to multi-modal image registration, both MI and MTM present comparable accuracies since the matched areas are extremely large. To demonstrate this, we evaluated the distance between an image and its modality counterpart under different translation parameters. The demonstrated performance is similar for other transformation parameters. Figure 8 displays distance maps between the visible and IR

image pair shown in Figure 6-middle. Distance maps for NGC, MTM and MI are shown (left to right). The minimum distance in the MI and MTM maps correspond to the correct translation with a deep and global minima. It can be seen that MTM and MI are comparable in their accuracy of alignment whereas NGC largely fails. Similar performance is observed for other image modalities as well.

Note that, searching for the correct transformation parameter in multi-modal alignment using MTM, can be implemented very efficiently: image slices need to be computed only once for the reference image of the pair, while transformed image requires only resampling and pointwise multiplication with the image slices. In contrast, searching for the transformation parameters using MI, requires computation of the joint histogram of the image pairs for each candidate parameter - a time consuming process.

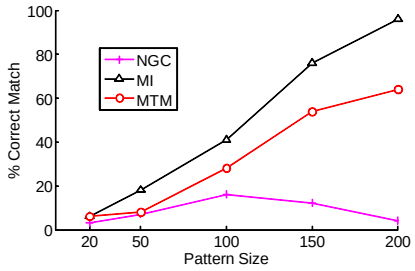


Figure 7. Detection rates of pattern matching between multi-modal images for various pattern sizes.

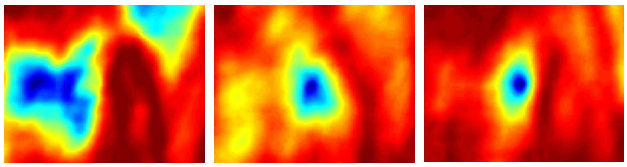

Figure 8. Distance maps between the visible and IR image pair shown in Figure 6-middle. The distance maps are for NGC (left), MTM (middle) and MI (right).

## 5. Conclusion

In this paper, a fast pattern matching scheme called *Matching by Tone Mapping (MTM)* was introduced. The distance measure used is expressed as a minimization problem over all possible tone mappings. Thus by definition, the MTM is invariant to non-linear tone mappings (both monotonic and non-monotonic). Furthermore, the MTM is shown to be a generalization of the NCC for non-linear mappings and actually reduces to NCC when mappings are restricted to be linear [14]. An efficient computation of the MTM is proposed requiring computation time similar to the fast implementation of Normalized Cross Correlation (NCC). The MTM is shown to be highly discriminative and robust to noise with performance capability equivalent to that of the Mutual Information scheme and requiring significantly lower computational time.

## References


[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. In *SIGGRAPH '09*, 2009. 1

[2] R. Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley, 2009. 1

[3] A. Buades and B. Coll. A non-local algorithm for image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 60–65, 2005. 1

[4] G. Cox. Review: Template matching and measures of match in image processing. *University of Cape Town, TR*, 1995. 1

[5] R. Gonzalez and R. Woods. *Digital Image Processing*. Prentice-Hall, Inc., 2006. 1

[6] C. Goulden. *Methods Of Statistical Analysis*. John Wiley and Sons, 1939. 2, 4

[7] H. Hel-Or. Matlab code available at "http://www.faculty.idc.ac.il/toky/software/software.htm". 5

[8] Y. Hel-Or and D. Shaked. A discriminative approach for wavelet denoising. *IEEE Trans. on Image Processing*, 17(4):443–457, 2008. 3

[9] N. Jojic, B. J. Frey, and A. Kannan. Epitomic analysis of appearance and shape. In *IEEE International Conference on Computer Vision*, volume 1, pages 34–41, 2003. 1

[10] J.P. Lewis. Fast normalized cross-correlation. In *Vision Interface*, pages 120–123, 1995. 1

[11] S. Kagarlitsky, Y. Moses, and Y. Hel-Or. Piecewise-consistent color mappings of images acquired under various conditions. In *The 12th IEEE International Conference on Computer Vision*, Kyoto, Japan, Sept 2009. 1

[12] F. Porikli. Integral histogram:a fast way to extract histograms in cartesian spaces. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 829–836, 2005. 2

[13] A. Roche, G. Malandain, X. Pennec, and N. Ayache. The correlation ratio as a new similarity measure for multimodal image registration. *Lecture Notes in Computer Science*, 1496:1115–1124, 1998. 2

[14] S. Ross. *A First Course in Probability*, chapter 7. Prentice Hall, fifth edition, 1998. 2, 8

[15] A. Sibiryakov. Statistical template matching under geometric transformations. In *international conference on Discrete geometry for computer imagery*, pages 225–237, 2008. 2

[16] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani. Summarizing visual data using bidirectional similarity. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 1

[17] T. Hastie and R. Tibshirani and J.H. Friedman. *The Elements of Statistical Learning*. Springer, 2003. 2

[18] P. Viola and W. Wells. Alignment by maximization of mutual information. *International journal of computer vision*, 24(2):137–154, 1977. 1

[19] Y. Wei and L. Tao. Efficient histogram-based sliding window. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3003–3010, 2010. 2

[20] M. Zibulevski. Code at "http://ie.technion.ac.il/∼mcib". 5