Abstract

In this paper we present a fast size-invariant method for binary image matching. The method, called Dissimilar Detection via Mapping (DDM), is based on probabilistic matching models that can quickly detect dissimilarity between binary images regardless of image size. Dissimilarity detection is performed quickly by comparing only a few points (pixels) between the images. As a result, image matching can be performed fairly quickly. A complete detailed analysis of DDM and the mathematical proof of its superiority over other methods as images become big is given. We compare DDM to three of the most popular matching methods employed in the image processing arena: image correlation, sum of the absolute difference, mutual information. We show that DDM is magnitudes faster than these methods – if the images are not small. Furthermore, we show how DDM can be used as a pre-processor for other matching methods to speed up their matching speed. In particular, we use DDM with image correlation to enhance the latter’s performance. Test results are presented for real images varying in size from 16 kilo-pixels to 10 mega-pixels to show the quickness of the method and its size-invariance.

Keywords: Binary image, image matching, size invariance matching, image correlation, image mapping, big image, image retrieval, sum of the absolute difference, mutual information

NOMENCLATURE

corr: Pearson’s correlation coefficient
cov: the Covariance matrix
dH: Binary Hamming distance
D: Distinct-dissimilar images; γ = 0
DC: Detection confidence
DDM: Dissimilar Detection via Mapping
E(A): Shannon entropy of A
E(A,B): Joint Shannon entropy between A and B
E[p]: Expected value of p
Ib: Best image match
Iq: Query image being matched
Ic: Candidate image i from an image database
I: Image database consisting of N images
Lmap: Maximum number of mappings attempted to detect dissimilarity
MI: Mutual Information.
MRN: Number of mappings required to detect dissimilarity

N: Total number of images of a database
Nd: Number of dissimilar images detected from the database
Ns: Number of similar images detected from the database
n: image size.
n: critical image size; image size at which ηD = 1
P: Probability of detecting dissimilarity between two images
P: Probability mass function
Q: Quasi-dissimilar images; 0 < γ < 1
PMM: Probabilistic Matching Model
PMMBI: Probabilistic Matching Model for Binary Images
p: Number of mappings
R: Dissimilar images; γ ≠ 1
r: Ratio of time taken to match a single pixel between method-x and DDM
S: Similar images; γ = 1
SADM: Sum of the absolute difference method
SMD: Simple matching distance (Sokal-Michener distance).
SSDM: Sum of the square difference method
SSD: Sum of the absolute difference distance
S: Sum of the square difference distance
LP: Amount of time taken for DDM to map image pixels until pairs of images are determined to be dissimilar.
LP: Amount of time taken to map pixels between two images until Lmap is exhausted
lp: Amount of time taken to map a single pixel.
Ta: Total detection time for DDM to compare a query image to all images in a database set
Tm: Detection time that method-x takes to match a query image to all images of a database to determine dissimilarity.
Ti: Mean time taken by method-x to determine an image pair to be dissimilar
T: Matching time per pixel for method-x
βD: Dissimilarity operation to be applied to the whole images.
βS: Similarity operation to be applied to the whole images.
ζ: Percentage of highly similar image pairs found when comparing a query image to the images of a database
γ: Binary similarity distance
γ: maximum expected similarity value (γ) at which images will be detected based on Lmap
ηD: Mean value of the similarity between the query image and the detected dissimilar images
ηD: Ratio of the number of images processed by DDM for a single image processed by method-x to detect dissimilarity between two images.
σ: Standard deviation.

Adnan A. Y. Mustafa
Assistant Professor, Department of Mechanical Engineering, Kuwait University, Kuwait

ORCID ID: 0000-0001-5972-5733
I. INTRODUCTION

The problem of image matching arises frequently in the field of image analysis under many topics such as, image registration [1] [2], template matching [3] [4], object tracking [5] [6] and video processing [7] [8]. Matching methods can be classified as either feature-based that rely on some method of extracting key salient image features and then matching these features (e.g. SIFT [9]), or area-based methods (also referred to as direct or intensity methods) that are based on comparing pixel image intensity values directly without feature extraction (e.g. image correlation [10]). Attempting to extract features from binary images is difficult due to the fact that they have two intensity levels only, resulting in a limited amount of scene detail. This makes feature-based methods difficult to apply and area-based methods seem to be the method of choice. However, area-based methods suffer from a serious handicap; they become very slow when the images are big. Image correlation and image subtraction methods are the most popular area-based methods for image matching and suffer greatly from this handicap. As a result, much research has been devoted to improving the performance of these methods [12] [13] [14]. Nevertheless, the size dependency problem of these two methods is still an unsolved problem, and is the most serious obstacle facing the continuous use of these methods for big images.

In general, when matching a query image to a database of images, using image correlation—or any other intensity-based method—a similarity operation \( \beta_S \) is applied to the whole images to find the best match (other methods may apply a dissimilarity operation \( \beta_D \)). The candidate image that produces the maximum similarity with \( \beta_S \) (or minimum dissimilarity for \( \beta_D \)) is selected as the best image match \( I_B \), i.e.,

\[
I_B(I_q,I) = \arg \max_i \beta_S(I_q,I_i) \quad \forall I \in \mathbf{I}
\]  

(1)

or

\[
I_B(I_q,I) = \arg \min_i \beta_D(I_q,I_i) \quad \forall I \in \mathbf{I}
\]  

(2)

where,

\[ I = \{ I_1, I_2, \ldots, I_N \} \]

(3)

\( I_q \) is the query image being matched and \( I \) is the candidate image from an image database \( \mathbf{I} \) consisting of \( N \) images. Because \( \beta_S \) (or \( \beta_D \)) is applied to the whole image, these methods are image-size dependent; as size increases, more processing time is required. Today, with imaging applications producing high resolution images (e.g. 40 giga-pixel images [15]), these methods are not only computationally intensive, but are rather impractical. Even with today’s powerful computers, attempting to match such big images to databases consisting of tens of thousands of images, is simply impractical. The dependency on image size is a serious handicap to image matching that has not been fully addressed.

One approach to quick matching is to scan the images quickly and discard any dissimilar images, and then only match images that have not been discarded as dissimilar; this is the approach we present. In this paper, we present and analyze a fast method for detecting dissimilar binary images. To detect dissimilarity among images we apply,

\[
I_B(I_o,I) = \arg \min_i \beta_D(I_o,I_i) \quad \forall I \in \mathbf{I}
\]  

(4)

\( \beta_D \) differs from \( \beta_D \) in that it detects dissimilarity by searching in similarity space. This is performed by exploiting the amount of similarity that exists (or is absent) between the images probabilistically. As a result, only few points (i.e. pixels) are compared rather than the whole image. This is true regardless of image size. This method is called the Dissimilar Detection via Mapping (DDM) method and is based on two established probabilistic models: the Probabilistic Matching Model (PMM) for binary image matching [16] and the Probabilistic Matching Model for Binary Images (PMMBI) [17]. In this paper we show that when images are dissimilar, DDM will always outperform current image-size dependent state-of-the-art similarity methods and hence detect dissimilarity quicker. On the other hand, when images are highly similar (i.e. near-duplicate, near-similar or exact replica) and dissimilarity cannot be detected, we show that DDM can still be used to estimate the similarity between the images with high confidence. The relative performance of DDM to other methods increases as image size increases due to its size invariance; the number of points required to detect dissimilarity remains constant and is not a function of image size, but rather a function of the amount of similarity between the images. We compare the performance of DDM with other well-established fast matching methods and show its superiority in detecting dissimilar images. We also show that DDM can be used with other matching methods to speed up their matching speed performance. As an example, we show this by applying DDM as a quick and efficient pre-processor for image correlation and show how it can speed up the matching process considerably, particularly for big images.

After this brief introduction, the remainder of this paper is organized as follows: section 2 points out related literature by briefly presenting other matching methods, and reviews previous work related to our research, where PMM, PMMBI and the \( \gamma \) binary similarity measure are summarized. Section 3 presents the main theme of this paper and begins by explaining the strategy followed to detect dissimilarity quickly. Two algorithms are presented; one for dissimilarity detection and the other for using DDM with other matching methods for similarity detection. Section 4 introduces performance measures and presents a detailed analytical development and discussion of these measures. Section 5 presents tests conducted on real image sets of various sizes and compare DDM’s performance with other matching methods. We finally conclude our paper in section 8 and discuss where our future research is directed.

II. RELATED WORK

Image correlation is the most popular area-based method employed for image matching. Image correlation is based on Pearson’s correlation coefficient [18] and is calculated by,
\[ \text{corr}(I_1, I_2) = \frac{\text{cov}(I_1, I_2)}{\sigma_{I_1} \cdot \sigma_{I_2}} \] (5)

\( \text{cov}(I_1, I_2) \) is the covariance matrix of images \( I_1 \) and \( I_2 \), and \( \sigma \) is the standard deviation of \( I \). If \( \text{corr}(I_1, I_2) = 1 \) or -1 then the images are the same or inverted, respectively. A value of zero indicates that the images are completely different, and values in between reflect the amount of similarity between the images.

A simpler method to detect similarity/dissimilarity between images, which is also popular in the image processing community, is the Sum of the Absolute Difference Method (SADM) [11], which is to simply subtract the two images and take the absolute value of the result,

\[ S_\text{AD}(I_1, I_2) = |I_1 - I_2| \] (6)

\( I_1 \) and \( I_2 \) are the images to be compared. Other variations exist, e.g. the Sum of the Square Difference Method (SSDM),

\[ S_\text{SS}(I_1, I_2) = (I_1 - I_2)^2 \] (7)

In both cases, if the result is zero then the images are identical. Despite its simplicity, SADM is very powerful in detecting changes and is very popular in many applications, particularly in the time consuming application of motion detection in video scenes, promoting hardware implementation of it [19].

Another simple technique to detect differences in binary images is to logically exclusive-or (denoted by XOR) the images. For any two binary values \( u \) and \( v \), the XOR operation—denoted by the \( \oplus \) symbol—is computed by,

\[ u \oplus v = (u + v) \cdot \overline{(u \cdot v)} \] (8)

This produces a value of 1 when \( u \) and \( v \) are different and 0 otherwise. If this expression is applied to the whole image on a pixel basis and its values are summed, the Binary Hamming distance (\( d_H \)) [20] is produced,

\[ d_H(I_1, I_2) = \sum (I_1 \oplus I_2) \] (9)

However, values produced by this expression are image size dependent and hence this expression needs to be normalized to give meaningful results. Once normalized, the Sokal-Michener Distance [21], more commonly known as the Simple Matching Distance (SMD), is produced,

\[ SMD(I_1, I_2) = \frac{1}{n} d_H(I_1, I_2) = \frac{1}{n} \sum (I_1 \oplus I_2) \] (10)

\( n \) is the image size. A value of zero for SMD indicates complete similarity between two images, while a value of unity indicates perfect dissimilarity.

Measuring Mutual Information (MI) between two binary images can also be used for matching purposes [22] [23] [24]. From information theory [25], MI is computed by,

\[ MI(I_1, I_2) = 2 - \frac{E(I_1) + E(I_2)}{\max(E(I_1), E(I_2))} \] (11)

\( E(I) \) is the Shannon entropy of \( I \), and \( E(I_1, I_2) \) denotes the joint entropy between \( I_1 \) and \( I_2 \). MI is a metric with values \( 0 \leq M \leq 1 \); a value of \( MI = 1 \) implies that the images are the same. Unfortunately, computing entropy is computationally intensive because of the multiple calls to the \( \log \) function to compute \( E \). However, lookup tables for the \( \log \) function and hardware implementations of it [26] are two ways to improve the performance of systems employing MI.

Other area-based methods have also been developed based on a variety of principles over the last 40+ years. Mustafa et al. [27] matched images by minimizing the image intensity combinations between two images with excellent results. However, the method only works with multi-bit images and is not suited for binary images. Baudrier et al. [28] proposed a method to adaptively measure the local dissimilarities between two binary images using a modified Hausdorff distance producing measures that are richer than a single global measure. The local Hausdorff distance measures produce a local-dissimilarity map (LDMap) that contains the local distances and their spatial layout which are used to compare images. Tang et al. [29] presented an approach to accelerate multi-scale template matching by representing the template as a linear combination of a small number of Haar-like binary features that can easily adapt to template scale changes with negligible extra computation cost. Vidal et al. [30] matched a sequence of binary images by using mathematical morphology to establish correspondences between connected component sets from the images. Teshome et al. [31] proposed a simple method for binary matching that is a two way process of comparing pixel values between an image and a database image and then they count the number of hits to find the best candidate image from the image database.

Unfortunately, all of the methods mentioned above are image-size dependent, since they process the value of every pixel in the image to measure the amount of closeness between images. In fact, most—if not all matching methods reported in the image processing literature, whether they are feature-based or intensity based, are image-size dependent. Throughout the remainder of this paper, we refer to such matching methods as Image Size Based Matching Methods (ISBMM).

### III. Binary Dissimilarity Detection Models

In this section we review and then discuss the dissimilarity detection models on which our method is based upon. For completeness and clarity, we first present a brief description of the similarity classification between binary images that we follow and review the \( \gamma \) binary similarity distance that is employed to measure similarity between images.

#### III.1 Measuring Similarity between binary images

Closeness between binary images follows the classification discussed in [32]. Images are classified as either Similar images (S) or Dissimilar images (R). Similar images are further classified as either exact-similar images if all pixels mapped from one image to the other image have the same intensity values, or inverse-similar images if they have the inverted intensity values at each pixel. Dissimilar images are further classified as, either Distinct-dissimilar images (D) if the dissimilarity between two images is maximized, or quasi-dissimilar images (Q) if the dissimilarity is not maximized.
Using the $\gamma$ binary similarity distance described next, the amount of dissimilarity between two images can be easily measured and checked if it is maximized or not.

**III.II The binary similarity distance**

Let $\gamma$ be the binary similarity distance [32] which measures the amount of similarity and concurrence between two binary images $u$ and $v$ based on a pixel-to-pixel mapping defined by,

$$\gamma(u, v) = |1 - 2P_\delta((z = u \oplus v) = 0)|, \ Z \in \{0, 1\}$$

where $\oplus$ is the exclusive-or operation and $P_\delta$ denotes the probability mass function of the image intensities of $z = u \oplus v$. Hence, $\gamma$ is a quantitative measure of the closeness and has values in the range $0 \leq \gamma \leq 1$. Based on $\gamma$, the similarity between two images is classified as either Similar ($S$) if $\gamma = 1$, or Dissimilar ($R$) if $0 \leq \gamma < 1$. Furthermore, the similarity is classified as Distinct-dissimilar ($D$) for the special case of $\gamma = 0$, indicating that the two images are completely different (i.e. maximum dissimilarity). Quasi-dissimilar images ($Q$) have values of $\gamma$ in the range $0 < \gamma < 1$.

**III.III Probabilistic Matching Models: PMM and PMMBI**

The Probabilistic Matching Model (PMM) for binary image matching [16] is a model that describes how quick dissimilarity can be detected between completely different binary images. PMM states that by randomly mapping corresponding points (i.e. pixels) between any two distinct-dissimilar ($D$) binary images, the probability of the images being detected as dissimilar by the $p^{th}$ mapping, $P(D, p)$, is given by,

$$P(D, p) = 1 - \frac{1}{2^{p-1}} \quad p = 1, 2, \ldots$$

$P$ can be interpreted as the confidence level or detection confidence (DC) of detecting dissimilarity after $p$ mappings. This confidence increases rapidly as more pixels are mapped and approaches unity (e.g. $P(D, 3) = 0.75$, $P(D, 4) = 0.875$, $P(D, 5) = 0.9375$, ...). This implies that distinct-dissimilar images can be quickly detected. However, this model is only valid for distinct-dissimilar binary images.

On the other hand, the more general Probabilistic Matching Model for Binary Images (PMBMI) [17] is a model that describes how quick dissimilarity can be detected between any pair of binary images, not just distinct-dissimilar binary images. PMMBI states that the probability of detecting dissimilarity between any two binary images based on the number of mappings ($p$) and the amount of similarity ($\gamma$) between the images is given by,

$$Pr(\gamma, p) = 1 - \left(\frac{1}{2}(1 + \gamma)\right)^p \left(1 + \frac{1}{1 + \gamma}\right)^p$$

$$0 \leq \gamma \leq 1, \ p = 1, 2, \ldots$$

As before, the probability value obtained from this equation can be thought of as the detection confidence (DC) in detecting dissimilarity for a given value of $p$ and $\gamma$. The model shows that:

- As similarity between two images become larger, more mappings are required to detect dissimilarity.
- DC increases with increasing $p$ for all values of $\gamma$.
- With increasing values of $p$, DC reaches unity quicker for smaller values of $\gamma$ than larger values.

Perhaps the most important result of PMM and PMMBI is their invariance to image size; dissimilarity detection quickness is not dependent on image size. Table 1 tabulates values of the minimum number of mappings ($p$) required for a given detection confidence ($DC$) for several values of $\gamma$. For example, a confidence level of 95% exists of detecting dissimilarity when mappings 7 pixels between pairs of images with $\gamma = 0.25$. To attain the same confidence for pairs of images with higher similarity, say $\gamma = 0.90$, requires 59 mappings. To reach higher confidence levels requires additional mapping for any given value of $\gamma$.

The expected value of $p$ is the mean number of mappings required to detect dissimilarity for a given value of $\gamma$, and is given by [17].

$$E[p(\gamma)] = \frac{4}{1 - \gamma} - 1 \quad 0 \leq \gamma < 1$$

Table 2 tabulates the expected number of mappings required to detect dissimilarity for several values of $\gamma$. It can be seen that the number of mappings are small for small values of $\gamma$, but grow very quickly as $\gamma$ approaches unity.

The tabulated values shown in Tables 1 and 2 (and equations (14) and (15)) can be employed in an inverted manner; finding the values of $\gamma$ given $p$. For example, if the dissimilarity between two images were detected in 11 mappings, then there is a 95% confidence that the similarity between the two images is $\gamma = 0.5$, or a 75% confidence that the similarity between the two images is $\gamma = 0.75$. In fact, there is a confidence level attached to every $\gamma$ value as can be seen from Fig. 1, which plots the detection confidence (Pr) as function of $\gamma$ for a value of $p = 11$. The most probable value for this pair of images is that it has a similarity value of $\gamma = 0.817$, which corresponds to the expected value of $p = 11$ mappings (obtained from (15)).

**IV. DISSIMILARITY DETECTION VIA MAPPING (DDM)**

The Dissimilar Detection via Mapping (DDM) method is a probabilistic pixel mapping method where random image pixels are mapped until dissimilarity is detected. The quickness of detecting dissimilarity is governed by the two probabilistic models; PMM and PMMBI. To detect dissimilarity among images we apply (4). Hence only few points are compared rather than the whole image.

When comparing images, we distinguish between three primary objectives:
The objective is to measure the amount of closeness between a query image and a set of given images (say from an image database) based on some closeness or similarity criteria. The ultimate goal is to find the best match from among the given images.

**Detecting image dissimilarity**: The objective is to determine if images are similar.

**Detecting image similarity**: The objective is to determine if images are exactly similar.

**Image matching**: The objective is to measure the amount of closeness between a query image and a set of given images (say from an image database) based on some closeness or similarity criteria. The ultimate goal is to find the best match from among the given images.

To detect dissimilarity between two images DDM follows the mapping strategy outlined in [16]. As a result, DDM can be used to detect dissimilarity between binary images at a very fast rate with very high confidence (close to 100%), but it cannot detect similar images (S) with 100% confidence. However, if similar images are to be detected with 100% confidence, then DDM can be employed as an efficient pre-processor with other matching methods to increase their matching speed, as we show later in this paper. Two DDM based algorithms for dissimilarity and similarity detection, were developed for this purpose and are described in [34].

To show the strength of DDM, two tools for measuring the relative performance of DDM to other methods were developed. The first performance measure compares the performance of DDM with respect to other matching methods in detecting dissimilarity. The other performance measure assesses the advantage of using DDM as a pre-processor with other matching methods for similarity detection. We will use the general term ‘method-x’ to refer to any ISBMM.

Let N denote the total number of images of a given database.

**Table 1**: Minimum Number of Mappings (p) required for a Given Detection Confidence (DC) and similarity values (γ)

<table>
<thead>
<tr>
<th>γ</th>
<th>0.5000</th>
<th>0.7500</th>
<th>0.9000</th>
<th>0.9500</th>
<th>0.9900</th>
<th>0.9990</th>
<th>0.9999</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>0.2500</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>10</td>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>0.5000</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>11</td>
<td>16</td>
<td>24</td>
<td>31</td>
</tr>
<tr>
<td>0.7500</td>
<td>6</td>
<td>11</td>
<td>18</td>
<td>23</td>
<td>35</td>
<td>52</td>
<td>69</td>
</tr>
<tr>
<td>0.8500</td>
<td>9</td>
<td>18</td>
<td>30</td>
<td>39</td>
<td>60</td>
<td>89</td>
<td>119</td>
</tr>
<tr>
<td>0.9000</td>
<td>14</td>
<td>28</td>
<td>45</td>
<td>59</td>
<td>90</td>
<td>135</td>
<td>180</td>
</tr>
<tr>
<td>0.9500</td>
<td>28</td>
<td>55</td>
<td>91</td>
<td>119</td>
<td>182</td>
<td>273</td>
<td>362</td>
</tr>
<tr>
<td>0.9900</td>
<td>139</td>
<td>277</td>
<td>460</td>
<td>598</td>
<td>919</td>
<td>1379</td>
<td>1838</td>
</tr>
<tr>
<td>0.9990</td>
<td>1386</td>
<td>2772</td>
<td>4600</td>
<td>5980</td>
<td>9190</td>
<td>13790</td>
<td>18380</td>
</tr>
<tr>
<td>0.9999</td>
<td>13860</td>
<td>27720</td>
<td>46000</td>
<td>59800</td>
<td>91900</td>
<td>137900</td>
<td>183800</td>
</tr>
</tbody>
</table>

**Table 2**: The expected number of mappings (p) required to detect dissimilarity for given values of γ.

<table>
<thead>
<tr>
<th>γ</th>
<th>P</th>
<th>γ</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>3.00</td>
<td>0.8000</td>
<td>10.1</td>
</tr>
<tr>
<td>0.1000</td>
<td>3.04</td>
<td>0.8500</td>
<td>13.4</td>
</tr>
<tr>
<td>0.2000</td>
<td>3.17</td>
<td>0.9000</td>
<td>20</td>
</tr>
<tr>
<td>0.3000</td>
<td>3.40</td>
<td>0.9500</td>
<td>40</td>
</tr>
<tr>
<td>0.4000</td>
<td>3.76</td>
<td>0.9900</td>
<td>200</td>
</tr>
<tr>
<td>0.5000</td>
<td>4.33</td>
<td>0.9980</td>
<td>1000</td>
</tr>
<tr>
<td>0.6000</td>
<td>5.25</td>
<td>0.9990</td>
<td>2000</td>
</tr>
<tr>
<td>0.7000</td>
<td>6.84</td>
<td>0.9999</td>
<td>20000</td>
</tr>
</tbody>
</table>

**Fig. 1**: A plot of the detection confidence (Pr) vs. γ for a value of p = 11.
to which the query image is to be compared. Furthermore, let \( N_d \) and \( N_s \) denote the number of dissimilar images and similar images in the database, respectively, with respect to the query image -as determined by \( \text{DDM} \). Hence,

\[
N = N_d + N_s
\]

Define \( \zeta \) such that,

\[
\zeta = \frac{N_d}{N} \quad \text{and} \quad 1 - \zeta = \frac{N_s}{N} \tag{17}
\]

i.e., \( \zeta \) is the percentage of highly similar image pairs found when comparing a query image to the images of the database. The phrase ‘highly similar image pairs’ refers to image pairs that cannot be detected as being dissimilar by \( \text{DDM} \) (e.g. exact-similar images). \( \zeta \) is dependent on the maximum allowable mappings (\( L_{\text{map}} \)) for each pair of images. For example, using \( L_{\text{map}} = 1000 \) mappings results in \( DC = 0.993 \) for images with a similarity of up to \( \gamma = 0.99 \) (near-duplicate images). However, at this number of mappings, near-similar images (\( \gamma = 0.999 \)) have a low confidence level of \( DC = 0.394 \). Increasing \( L_{\text{map}} \) to 2000 mappings results in \( DC = 0.99996 \) for images with a similarity of up to \( \gamma = 0.99 \), and \( DC = 0.632 \) for near-similar images. Only at \( L_{\text{map}} = 4600 \) mappings does \( DC \) reach 0.9 for near-similar images. Clearly a trade-off exists between the amount of similarity between images not being detected, and the amount of time allowed to check for dissimilarity (\( L_{\text{map}} \)).

**IV.I Dissimilarity Detection Performance**

As earlier discussed, detecting dissimilar images refers to the goal of asserting that an image pair is dissimilar. Furthermore, dissimilar detection is a function of the number of mappings allowed, \( L_{\text{map}} \). Let \( \gamma_L \) denote the maximum expected similarity value at which images will be detected based on \( L_{\text{map}} \). This implies that image pairs with \( \gamma \geq \gamma_L \) will most likely not be detected as dissimilar. Hence, a value of \( L_{\text{map}} = 1000 \) implies that \( \gamma_L = 0.9980 \) (from Table 2), and image pairs with \( \gamma > 0.9980 \) will most likely not be detected as dissimilar. A value of \( L_{\text{map}} = 1000 \) is what is usually employed in our work.

Let \( t_{p,d} \) denote the amount of time taken for \( \text{DDM} \) to map image pixels until a pair of images are determined to be dissimilar. For image pairs that have great similarity and cannot be determined to be similar at a given \( L_{\text{map}} \) value, \( t_{p,d} \) is then,

\[
t_{p,d} = t^*_p = L_{\text{map}} \cdot t_p
\]

\( t^*_p \) is the amount of time taken to map points between two images until \( L_{\text{map}} \) is reached, and \( t_p \) is the amount of time taken to map a single point. As an example, on the system tested, it was found that \( t^*_p \approx 3.433 \text{ ms} \) for \( L_{\text{map}} = 1000 \). Then if \( T_p \) denotes the total detection time for \( \text{DDM} \) to compare the query image to all images in the database set, then \( T_p \) can be calculated by,

\[
T_p = N_d \cdot E[t_{p,d}] + N_s t^*_p
\]

\( \text{E}[t_{p,d}] \) denotes the expected time taken to map pixels until an image is determined to be dissimilar using \( \text{DDM} \). From \( \text{PMM} \),

\[
\text{E}[t_{p,d}] = \text{E}[\gamma_d] \cdot t_p
\]

Hence,

\[
\text{E}[t_{p,d}] = \text{MRN}(\gamma_d, DC) \cdot t_p
\]

MRN is the number of mappings required to detect dissimilarity which is a function of 1) the mean value of the similarity between the query image and the detected dissimilar images (\( \gamma_d \)), and 2) the desired detection confidence (DC). Hence, (19) becomes,

\[
T_p = N_d \cdot \text{MRN}(\gamma_d, DC) \cdot t_p + N_s \cdot L_{\text{map}} \cdot t_p
\]

If \( T_d \) denotes the detection time that \textit{method-x} takes to match the query image to all images of the database to determine dissimilarity, then,

\[
T_x = \bar{N} t_s
\]

where \( \bar{t}_s \) denotes the average time taken by \textit{method-x} to determine an image pair to be dissimilar which is only a function of image size (\( n \)),

\[
\bar{t}_s = n t_s
\]

\( t_s \) is the matching time per pixel for \textit{method-x}. Hence,

\[
T_x = N n t_s
\]

Let \( \eta_{D,x} \) be a performance comparison measure between \( \text{DDM} \) and \textit{method-x}. \( \eta_{D,x} \) is defined as the number of images processed by \( \text{DDM} \) for a single image processed by \textit{method-x} to detect dissimilarity between images, and is computed by the ratio of the detection time of \textit{method-x} to the detection time of \( \text{DDM} \),

\[
\eta_{D,x} = \frac{T_d}{T_p}
\]

Hence, a value of \( \eta_{D,x} > 1 \) indicates \( \text{DDM} \) outperforms \textit{method-x} (i.e. faster), while a value of \( \eta_{D,x} < 1 \) indicates \textit{method-x} outperforms \( \text{DDM} \) (i.e. \( \text{DDM} \) slower). Substituting (22) and (25) into (26) produces,

\[
\eta_{D,x} = \frac{N \cdot n \cdot \bar{t}_s}{(N_d \cdot \text{MRN}(\gamma_d, DC) + N_s \cdot L_{\text{map}}) \cdot t_p}
\]

Dividing by \( N \) and using (17) produces,

\[
\eta_{D,x}(n, \gamma_d, DC, \zeta, L_{\text{map}}, r_s) = \frac{n \cdot r_s}{(1 - \zeta) \cdot \text{MRN}(\gamma_d, DC) + \zeta L_{\text{map}}}
\]

\( r_s = \bar{t}_s / t_p \) is a constant dependent on \textit{method-x} and \( \text{DDM} \). Normalizing with respect to \( r_s \), produces,
\[ \eta_D(n, \gamma, DC, \zeta, L_{map}) = \eta_{D_1} / r_s \]
\[ = \frac{n}{(1 - \zeta) \cdot MRN(\gamma, DC) + \zeta L_{map}} \] (29)

This equation shows \( \eta_D \) to be a function of several variables. The sensitivity of \( \eta_D \) to these variables is discussed next.

**IV.I Sensitivity Analysis of \( \eta_D \)**

Eq. (29) shows \( \eta_D \) to be a function of the parameters: \( n, \zeta, L_{map} \), and \( MRN \), where \( MRN \) is a function of both \( \gamma \) and \( DC \). It can be seen that,

- \( \eta_D \) is directly proportional to \( n \) and inversely proportional to each of the three parameters: \( \zeta, MRN \) and \( L_{map} \).
- Since \( MRN \) increases with an increase in either \( \gamma \) or \( DC \), \( \eta_D \) is inversely proportional to both \( \gamma \) and \( DC \).
- A change in \( L_{map} \) may or may not cause changes in the values of \( \zeta \) and \( \gamma \) for a given set, and hence may or may not affect \( \eta_D \). For example, if an increase in \( L_{map} \) results in more images being detected as dissimilar, then \( \zeta \) will decrease while \( \gamma \) will increase. On the other hand, if an increase in \( L_{map} \) does not result in more images being detected, then \( \zeta \) and \( \gamma \) will not be affected (only more search time has been wasted).

1) **Sensitivity Analysis of \( \eta_D \) to \( n \)**

**Figure 2** shows curves of \( \eta_D \) versus \( n \) for several values of \( \zeta \) (for the sample case of \( \gamma = 0.2 \) and \( L_{map} = 1000 \)). Due to the dominance of the effect of \( n \) on \( \eta_D \), the curves are linear in \( n \); as \( n \) increases so does \( \eta_D \) and as \( n \to \infty \), then \( \eta_D \to \infty \). This can be demonstrated to be true from (29); since \( L_{map} \) has a finite value and by definition of \( MRN \),
\[ MRN(\gamma, DC) \leq L_{map} \] (30)

Hence, \( MRN(\gamma, DC) \) is also bounded and finite, resulting in the denominator of (29) also being finite, and hence,
\[ \lim_{n \to \infty} \eta_D = \infty \] (31)

Hence, as image size increases and becomes very large, DDM’s dissimilarity detection will be magnitudes faster than any ISBMM.

2) **Sensitivity Analysis of \( \eta_D \) to \( \zeta \)**

A decrease in \( \zeta \) (i.e. less similar images in the database with respect to the query image) results in a shift of the curves upwards; smaller values of \( \zeta \) result in larger values of \( \eta_D \) and higher relative performance of DDM with respect to other methods. In the limit as \( \zeta \) becomes very small and \( \zeta \to 0 \), (29) reveals that \( \eta_D \) essentially becomes a function of \( n \),
\[ \lim_{\zeta \to 0} \eta_D = \frac{n}{MRN(\gamma, DC)} \] (32)

\( n \) dominates over the effect of \( MRN \) which is dependent on the two parameters, \( \gamma \) and \( DC \). Hence, \( \eta_D \) increases as image size increases. **Figures 3** shows curves of \( \eta_D \) versus \( n \) for different values of \( \gamma \) when \( \zeta = 0 \) (for the case \( L_{map} = 1000 \)). \( \eta_D \) increases with \( n \), but is inversely proportional to \( \gamma \), where an increase in \( \gamma \) shifts the curve downwards. The sensitivity of \( \eta_D \) to a change in the three parameters \( \gamma, DC \) and \( L_{map} \) is very minimal, where an increase in any of these parameters will result in a small shift of the curves downwards and vice versa.

It is important to note that when \( \zeta \neq 0 \), implies \( N_s = \zeta N \) images have not been determined to be dissimilar (inconclusive result), and are assumed to be similar up to a certain degree (with a predefined amount of confidence as earlier discussed).

**IV.II Critical Image Size Determination**

For DDM to outperform method-x, then,
\[ \eta_{D_1}(n, \gamma, DC, \zeta, L_{map}) > 1 \] (33)

must hold true. Then from (28),
\[ \frac{n \cdot r_s}{(1 - \zeta) \cdot MRN(\gamma, DC) + \zeta \cdot L_{map}} > 1 \] (34)

must also hold true. Solving for \( n \),
\[ n > \frac{1}{r_s} \left( (1 - \zeta) \cdot MRN(\gamma, DC) + \zeta \cdot L_{map} \right) \] (35)

When this condition is satisfied, then DDM will outperform any method-x. Let \( n_c \) denote the critical image size, defined as the image size at which \( \eta_{D_1} = 1 \), then,
\[ n_c = \frac{1}{r_s} \left( (1 - \zeta) \cdot MRN(\gamma, DC) + \zeta \cdot L_{map} \right) \] (36)

Thus, DDM outperforms any method-x when images of size \( n > n_c \) are matched. **Figures 4 and 5** show curves of \( n_c \) versus \( \gamma \) for different values of \( \zeta \) (with different values of \( L_{map} \) and \( DC \)). A value of \( r_s = 3.434 \cdot 10^3 \), was used to generate the plots, which was found to be the minimum value found for the methods tested. Graph regions above a given curve are where DDM will outperform other methods; the farther away upwards, the larger performance difference. From the plots, we see that \( n_c \) increases as \( \gamma \) increases. Also as \( \zeta \) increases, the curves shift upwards and hence \( n_c \) increases. Furthermore, as \( DC \) increases so does \( n_c \). Note that for highly similar images \( \gamma > 0.9 \) \( n_c \) increases rapidly.

**IV.III Similarity Detection Performance**

If the goal is to assert if a pair of images is similar with 100% confidence, then DDM cannot be used independently. Alternatively, using DDM can be used to boost the performance of any method-x. In this section, the performance of using DDM with other methods is analyzed.
IV.II.1 Performance Analysis

The time DDM expends for similarity detection consists of two parts: \( t_{p,d} \) and \( t_{p,x} \), where \( t_{p,d} \) is the same as that was presented earlier for dissimilarity detection, and \( t_{p,x} \) is the time taken to match a pair of images which were not determined to be dissimilar by DDM.

\( t_{p,x} \) consists of two parts: \( t_{p,x}^{\gamma_p} \); the time taken by DDM to map \( L_{map} \) points and the time taken by method-x,

\[
t_{p,x} = t_{p,x}^{\gamma_p} + n t_s
\]  

(37)

Using (18) this becomes,

\[
t_{p,x} = L_{map} t_p + n t_s
\]  

(38)

\( t_s \) is the same as before. Let \( T_{ps} \) denote the total matching time for DDM-x to detect similarity/dissimilarity between a query image and all images of the database, defined as,

\[
T_{ps} = N_c t_{p,d} + N_s t_{p,x}
\]  

(39)

Let \( \eta_{s,x} \) denote the matching performance of DDM-x with respect to method-x, measured by the number of images processed by DDM-x for a single image processed by method-x, and is computed by,
$$\eta_{S,x} = \frac{T_s}{T_{px}}$$

Substituting (21), (25), (38) and (39) produces,

$$\eta_{S,x} = \frac{N \cdot n \cdot t_x}{N_d \cdot MRN(\gamma_p) \cdot t_p + N_s(L_{map} \cdot t_p + n \cdot t_x)}$$

Dividing by $N$, and using (17),

$$\eta_{S,x} = \frac{n \cdot t_x}{(1 - \zeta) \cdot MRN(\gamma_p) \cdot t_p + \zeta(L_{map} \cdot t_p + n \cdot t_x)}$$

Rearranging the denominator,

$$\eta_{S,x} = \frac{n \cdot t_x}{MRN(\gamma_p) \cdot t_p + \zeta(L_{map} - MRN(\gamma_p) \cdot t_p + n \cdot t_x)}$$

Dividing the numerator and the denominator by $t_p$ and setting $r_s = t_x / t_p$ finally produces the desired result,

$$\eta_{S,x}(n, \gamma_p, DC, \zeta, L_{map}) = \frac{n \cdot r_s}{MRN(\gamma_p, DC) + \zeta(L_{map} - MRN(\gamma_p, DC) + n \cdot r_s)}$$

Equation (44) shows plots of $\eta_{S,x}$ curves versus $n$ for various values of $\zeta$ and different values of $L_{map}$, $\gamma_p$ and $DC$ (for $r_s = 3.434 \times 10^{-3}$). Figures 8 and 9 show typical plots of $\eta_{S,x}$ curves versus $n$ for different values of $\gamma_p$ (with different values of $L_{map}$, $\zeta$ and $DC$ values).

**IV.III. Sensitivity Analysis of $\eta_{S,x}$**

A striking difference between the curves of $\eta_{S,x}$ and $\eta_{D,x}$ is that the $\eta_{S,x}$ curves do not increase indefinitely with $n$, but reach a steady state value. Furthermore,

- $\eta_{S,x}$ only increases for small $n$ values before reaching its steady state values.
- As $\zeta$ or $\gamma_p$ increases, $\eta_{S,x}$ decreases.

$n$ and $\zeta$ are the two most important factors affecting $\eta_{S,x}$ and are discussed in more detail next.

1. **Sensitivity Analysis of $\eta_{S,x}$ to $n$**

$\eta_{S,x}$ increases as $n$ increases until it reaches its steady state value of $1/\zeta$. This is due to the fact that for large $n$,

$$L_{map} - MRN(\gamma_p, DC) << n \cdot r_s$$

Thus in the limit as $n \to \infty$, (44) reduces to,

$$\lim_{n \to \infty} \eta_{S,x} = \frac{n \cdot r_s}{\zeta(n \cdot r_s)} = \frac{1}{\zeta}$$

Equation (46) shows plots of $\eta_{S,x}$ curves versus $n$ for various values of $\zeta$ ($L_{map} = 1000$, $\gamma_p = 0.2$, $DC = 0.9$).
2. Sensitivity Analysis of $\eta_{S,x}$ to $\zeta$

As $\zeta$ increases, $\eta_{S,x}$ decreases. In the limit when $\zeta$ is very small,

$$\lim_{\zeta \to 0} \eta_{S,x} = \frac{n \cdot r_x}{MRN(\gamma_D, DC)}$$

which is an identical result to (32). Hence,

$$\lim_{\zeta \to 0} \eta_{S,x} = \lim_{\zeta \to 0} \eta_{D,x}$$

i.e. i.e., when a query image is highly dissimilar with respect to the images of the database, employing DDM independently or with any method-$x$ will produce the same result. This should not come as a surprise; when the image set contains no highly similar images, then DDM-$x$ will detect dissimilarity by DDM only and method-$x$ is never invoked. In a similar fashion, as $\zeta \to 0$ in (46), then,

$$\lim_{n \to \infty} \eta_{S,x} = \infty$$

This result agrees with (31). Hence,

$$\lim_{n \to \infty} \eta_{S,x} = \lim_{n \to \infty} \eta_{D,x}$$

This implies that, when the query image is big and highly dissimilar with respect to the database images, then both DDM and DDM-$x$ will produce the same result.

V. DISCUSSION

In this section we present our test results conducted on real images. The performance of DDM is compared to three intensity-based binary image matching methods:

1. The sum of the absolute difference method (SADM).
2. Image Correlation (Corr).
3. Mutual information (MI).

Seven image sets of different sizes with size ranging from 16 kp to 8 Mp were employed. The $\gamma$ statistics for the image sets are shown in Table 3. A value of $L_{\text{map}} = 1000$ was employed for DDM.

V.I Detecting Dissimilar Images with $\zeta = 0$

The first series of tests are for image sets with $\zeta = 0$; images are different with no similar or near-similar images in these sets. The performance curves for these tests are shown in Fig. 10 and tabulated in Table 4. DDM clearly outperformed all methods by detecting dissimilarity faster than any other method. In general, $\eta_{D,x}$ increased as size increased for all methods,

- **DDM performance with respect to correlation ($\eta_{D,\text{corr}}$)**
  $\eta_{D,\text{corr}}$ had a performance range from 152 to 141301, as image size increased from 16 kp to 10 Mp.

- **DDM performance with respect to SADM ($\eta_{D,\text{SADM}}$)**
  $\eta_{D,\text{SADM}}$ had a performance range from 450 to 227550, as image size increased from 16 kp to 10 Mp.

- **DDM performance with respect to MI ($\eta_{D,\text{MI}}$)**
  $\eta_{D,\text{MI}}$ had a performance range from 3409 to 373743, as image size increased from 16 kp to 10 Mp.
We observe, after conjunction with another method to image sets, the performance of methods in comparison with correlation, where \( \eta_{D,corr} = 1 \) (i.e. employing either DDM or correlation yields the same performance at this particular image size). But as image size increases, DDM outperforms correlation once again. In general,

- **DDM performance with respect to correlation** \( (\eta_{D,corr}) \)
  - \( \eta_{D,corr} \) had a performance range from 1 to 1097, as image size increased from 16 kp to 10 Mp.

- **DDM performance with respect to SADM** \( (\eta_{D,SADM}) \)
  - \( \eta_{D,SADM} \) had a performance range from 3 to 1766, as image size increased from 16 kp to 10 Mp.

- **DDM performance with respect to MI** \( (\eta_{D,MI}) \)
  - \( \eta_{D,MI} \) had a performance range from 26 to 2901, as image size increased from 16 kp to 10 Mp.

Despite the decrease in the performance of \( \eta_{D,corr} \) for all methods in comparison with the previous case \( (\zeta = 0) \) due to the presence of a significant percentage of similar images in the image sets, nevertheless the performance of \( \eta_{D,corr} \) is greater than unity for all cases tested and increases as image size increases. The fact that \( \eta_{D,corr} \) produced large values for big images \( (\eta_{D,corr} = 1097, \eta_{D,SADM} = 1766 \text{ and } \eta_{D,MI} = 2901) \) despite a significant portion of the images being matched are similar, shows the strength of DDM.

**V.III Improving the Performance of Image Correlation with DDM**

In this section, we present the results of employing DDM in conjunction with another method to see the advantage of using DDM to speed up its performance. Since correlation was found to be the 2nd fastest method in detecting dissimilarity after DDM, correlation was selected to be tested with DDM to detect similarity, denoted by DDM-corr. In this test 70 image pairs with \( \zeta = 0.4 \) were tested using a value of \( L_{map} = 1000 \). Table 6 shows values of \( T_{pc}, T_r \) and \( \eta_{pc,corr} \) for the image sets, and a plot of \( T_r \) and \( T_{pc} \) against image size is shown in Fig. 12. We observe,

- For the two smallest image sets, the 16 kp and 64 kp image sets, \( T_r < T_{pc} \) and thus \( \eta_{pc,corr} < 1 \). Hence, DDM outperforms correlation using DDM-corr for small images. There is no advantage of using DDM as a pre-processor.
- For the remaining larger sets \( T_r > T_{pc}, \eta_{pc,corr} > 1 \). Thus DDM provides a great advantage to the speed of correlation as images become big.

Fig. 13 shows a plot of \( \eta_{pc,corr} \) for the results obtained experimentally. A plot of the theoretical equation developed for \( \eta_{pc,corr} \) (i.e. (44)) is also plotted. We see that experimental data agree very well with the theory developed with very little discrepancy. We also observe that as \( n \) becomes large, \( \eta_{pc,corr} \) approaches its steady state value of 2.5 in agreement with (46). Hence, using DDM with correlation boosted its detection rate by 250% for big images.

<table>
<thead>
<tr>
<th>Image Set</th>
<th>Image size</th>
<th>Range</th>
<th>mean</th>
<th>std</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 kp</td>
<td>128 x 128</td>
<td>0.000 – 0.364</td>
<td>0.140</td>
<td>0.128</td>
</tr>
<tr>
<td>64 kp</td>
<td>256 x 256</td>
<td>0.093 – 0.284</td>
<td>0.196</td>
<td>0.068</td>
</tr>
<tr>
<td>256 kp</td>
<td>521 x 512</td>
<td>0.024 – 0.206</td>
<td>0.102</td>
<td>0.056</td>
</tr>
<tr>
<td>300 kp</td>
<td>640 x 480</td>
<td>0.002 – 0.256</td>
<td>0.145</td>
<td>0.090</td>
</tr>
<tr>
<td>3 Mp</td>
<td>2048 x 1536</td>
<td>0.029 – 0.129</td>
<td>0.069</td>
<td>0.039</td>
</tr>
<tr>
<td>8 Mp</td>
<td>3264 x 2448</td>
<td>0.122 – 0.479</td>
<td>0.297</td>
<td>0.130</td>
</tr>
<tr>
<td>10 Mp</td>
<td>3648 x 2736</td>
<td>0.026 – 0.363</td>
<td>0.170</td>
<td>0.135</td>
</tr>
</tbody>
</table>

**Table 6:** Values of \( \eta_{D,x} \) for the methods tested \( (\zeta = 0.4) \)

<table>
<thead>
<tr>
<th>Image Set</th>
<th>Image size</th>
<th>( T_{pc} ) (ms)</th>
<th>( T_r ) (ms)</th>
<th>( \eta_{pc,corr} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 kp</td>
<td>128 x 128</td>
<td>2.1</td>
<td>13.6</td>
<td>0.15</td>
</tr>
<tr>
<td>64 kp</td>
<td>256 x 256</td>
<td>10.1</td>
<td>17</td>
<td>0.6</td>
</tr>
<tr>
<td>256 kp</td>
<td>512 x 512</td>
<td>85.8</td>
<td>48.2</td>
<td>1.78</td>
</tr>
<tr>
<td>300 kp</td>
<td>640 x 480</td>
<td>107</td>
<td>56.4</td>
<td>1.9</td>
</tr>
<tr>
<td>3 Mp</td>
<td>2048 x 1536</td>
<td>1238</td>
<td>511.1</td>
<td>2.42</td>
</tr>
<tr>
<td>8 Mp</td>
<td>3264 x 2448</td>
<td>3393</td>
<td>1371</td>
<td>2.47</td>
</tr>
<tr>
<td>10 Mp</td>
<td>3648 x 2736</td>
<td>5026</td>
<td>2024</td>
<td>2.48</td>
</tr>
</tbody>
</table>

**Table 2:** Values of \( \eta_{D,a} \) for the methods tested \( (\zeta = 0) \)

**Table 3:** \( \gamma \) Statistics for the image sets

**Table 4:** Values of \( \eta_{D,a} \) for the methods tested \( (\zeta = 0) \)

**Table 5:** Values of \( \eta_{D,a} \) for the methods tested \( (\zeta = 0.4) \)

**Table 6:** Comparison of Time and Performance of DDM and DDM-corr
VI. CONCLUSION

In this paper we have presented a fast method for detecting dissimilar binary images. The method is called the Dissimilar Detection via Mapping method (DDM) and is based on two established probabilistic models: the Probabilistic Matching Model (PMM) for binary image matching and the Probabilistic Matching Model for Binary Images (PMMBI). The method requires the comparison of only a few points to detect dissimilar images regardless of image size. DDM was compared to other matching methods such as, image correlation, sum of the absolute difference (SADM), and mutual information (MI). Results show that DDM is magnitudes faster than all of these methods, but is dependent on the amount of similarity between the query image and the database images, \( \zeta \). For example, for 10 mega-pixel images, it was found that DDM was more than 140000 times faster than image correlation in detecting dissimilarity when matching a query image to a database of images containing highly dissimilar images (\( \zeta = 0 \)). When compared to other methods, the performance of DDM was even greater; DDM was more than 220000 times faster than SADM, and more than 370000 times faster than MI. If the images of the database consists of highly similar or near duplicate images of the query image (more than 99% similar), then the performance of DDM decreases somewhat, but still outperforms other methods tested. For example, employing 10 mega-pixel images, it was found that if the image database contains 40% highly similar images to the query image (\( \zeta = 0.4 \)), then DDM was more than 1000 times faster than image correlation, and more than 1700 times faster than SADM, and more than 2900 times faster than MI.

A limitation of DDM is that, while it can detect dissimilarity extremely fast regardless of image size, it cannot be used to detect similarity 100%. However, DDM can be used as an efficient pre-processor with other matching methods to increase their matching performance. This increased performance has a limiting value of \( 1/\zeta \). For example, using DDM with image correlation increased the time performance of correlation by 2.5 times for image sets containing 40% highly similar image pairs (\( \zeta = 0.4 \)).

Our future work is headed towards the application of DDM to image registration and template matching. We hope to show the advantage of using DDM to reduce the computational cost for such applications.

![Fig. 10. Plots of \( \eta_{D,x} \) against image size for the various methods tested (\( \zeta = 0.0 \)).](image1)

![Fig. 11. Plots of \( \eta_{D,x} \) against image size for the various methods tested (\( \zeta = 0.4 \)).](image2)
REFERENCES


