International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 8 (2019), pp. 1293-1306
© International Research Publication House. http://www.irphouse.com

Fast Size-Invariant Binary Image Matching Through Dissimilarity Detection via Pixel Mapping

Adnan A. Y. Mustafa

Assistant Professor, Department of Mechanical Engineering, Kuwait University, Kuwait

ORCID ID: 0000-0001-5972-5733

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 and 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

d_H: Binary Hamming distance

D: Distinct-dissimilar images; $\gamma = 0$

DC: Detection confidence

DDM: Dissimilar Detection via Mapping

$E(\mathbf{A})$: Shannon entropy of **A**

$E(\mathbf{A},\mathbf{B})$: Joint Shannon entropy between **A** and **B**.

E[p]: Expected value of p

$\mathbf{I}_B$: Best image match

$\mathbf{I}_q$: Query image being matched

$\mathbf{I}_i$: Candidate image *i* from an image database

I: Image database consisting of *N* images

L_{map}: 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

N_d: Number of dissimilar images detected from the database

N_s: Number of similar images detected from the database

n: image size.

n_c: critical image size; image size at which $\eta_{D,x} = 1$

Pr: Probability of detecting dissimilarity between two images

P_o: Probability mass function

Q: Quasi-dissimilar images; $0 < \gamma < 1$

PMM: Probabilistic Matching Model

PMMBI: Probabilistic Matching Model for Binary Images

p: Number of mappings

R: Dissimilar images; $\gamma \neq 1$

r_x: Ratio of time taken to match a single pixel between method-*x* and *DDM*

S: Similar images; $\gamma = 1$

SADM: Sum of the absolute difference method

SMD: Simple matching distance (Sokal-Michener distance).

SSDM: Sum of the square difference method

$\mathbf{S}_D$: Sum of the absolute difference distance

$\mathbf{S}_S$: Sum of the square difference distance

$t_{p,d}$: Amount of time taken for *DDM* to map image pixels until pairs of images are determined to be dissimilar.

t^*_p: Amount of time taken to map pixels between two images until L_{map} is exhausted

t_p: Amount of time taken to map a single pixel.

T_p: Total detection time for *DDM* to compare a query image to all images in a database set

T_x: Detection time that method-*x* takes to match a query image to all images of a database to determine dissimilarity.

$\bar{t}_x$: Mean time taken by method-*x* to determine an image pair to be dissimilar

t_x: Matching time per pixel for method-*x*

β_D: Dissimilarity operation to be applied to the whole images.

β_S: 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_L: maximum expected similarity value (γ) at which images will be detected based on L_{map}

γ_μ: Mean value of the similarity between the query image and the detected dissimilar images

$\eta_{D,x}$: 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.

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 [11] are perhaps the two 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 β_S is applied to the whole images to find the best match (other methods may apply a dissimilarity operation β_D). The candidate image that produces the maximum similarity with β_S (or minimum dissimilarity for β_D) is selected as the best image match $\mathbf{I}_B$, i.e.,

$$\mathbf{I}_B(\mathbf{I}_q,\mathbf{I}) = \arg\max_i\left(\beta_S(\mathbf{I}_q,\mathbf{I}_i)\right) \qquad \forall \mathbf{I}_i \in \mathbf{I} \tag{1}$$

or

$$\mathbf{I}_B(\mathbf{I}_q,\mathbf{I}) = \arg\min_i\left(\beta_D(\mathbf{I}_q,\mathbf{I}_i)\right) \qquad \forall \mathbf{I}_i \in \mathbf{I} \tag{2}$$

where,

$$\mathbf{I} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_N\} \tag{3}$$

$\mathbf{I}_q$ is the query image being matched and $\mathbf{I}_i$ is the candidate image from an image database ($\mathbf{I}$) consisting of N images. Because β_S (or β_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,

$$\mathbf{I}_B(\mathbf{I}_o,\mathbf{I}) = \arg\min_i\left(\beta'_D(\mathbf{I}_o,\mathbf{I}_i)\right) \qquad \forall \mathbf{I}_i \in \mathbf{I} \tag{4}$$

β'_D differs from β_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 γ 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,

$$corr(\mathbf{I}_1,\mathbf{I}_2)=\frac{\mathrm{cov}(\mathbf{I}_1,\mathbf{I}_2)}{\sigma_{\mathbf{I}_1}\cdot\sigma_{\mathbf{I}_2}} \tag{5}$$

cov($\mathbf{I}_1$,$\mathbf{I}_2$) is the covariance matrix of images $\mathbf{I}_1$ and $\mathbf{I}_2$, and $\sigma_{\mathbf{I}}$ is the standard deviation of **I**. If *corr*($\mathbf{I}_1$,$\mathbf{I}_2$) = 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,

$$\mathbf{S}_D(\mathbf{I}_1,\mathbf{I}_2) = |\mathbf{I}_1 - \mathbf{I}_2| \tag{6}$$

$\mathbf{I}_1$ and $\mathbf{I}_2$ are the images to be compared. Other variations exist, e.g. the *Sum of the Square Difference Method* (*SSDM*),

$$\mathbf{S}_S(\mathbf{I}_1,\mathbf{I}_2) = (\mathbf{I}_1 - \mathbf{I}_2)^2 \tag{7}$$

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 ⊕ symbol– is computed by,

$$u\oplus v=(u+v)\cdot\overline{(u\cdot v)} \tag{8}$$

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,

$$d_H(\mathbf{I}_1,\mathbf{I}_2)=\sum(\mathbf{I}_1\oplus\mathbf{I}_2) \tag{9}$$

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,

$$SMD(\mathbf{I}_1,\mathbf{I}_2)=\frac{1}{n}d_H(\mathbf{I}_1,\mathbf{I}_2)=\frac{1}{n}\sum(\mathbf{I}_1\oplus\mathbf{I}_2) \tag{10}$$

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,

$$MI(\mathbf{I}_1,\mathbf{I}_2)=2-\frac{E(\mathbf{I}_1,\mathbf{I}_2)}{\max(E(\mathbf{I}_1),E(\mathbf{I}_2))} \tag{11}$$

E(**I**) is the Shannon entropy of **I**, and *E*($\mathbf{I}_1$,$\mathbf{I}_2$) denotes the joint entropy between $\mathbf{I}_1$ and $\mathbf{I}_2$. *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 γ binary similarity distance that is employed to measure similarity between images.

III.I 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 γ *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 γ 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,

$$\gamma(\mathbf{u},\mathbf{v}) = |1 - 2P_o((\mathbf{z} = \mathbf{u}\oplus\mathbf{v}) = Z)|, \quad Z \in \{0,1\} \qquad (12)$$

where ⊕ is the *exclusive-or* operation and P_o denotes the probability mass function of the image intensities of $\mathbf{z} = \mathbf{u}\oplus\mathbf{v}$. Hence, γ is a quantitative measure of the closeness and has values in the range $0 \le \gamma \le 1$. Based on γ, the similarity between two images are classified as either *Similar* (*S*) if $\gamma = 1$, or *Dissimilar* (*R*) if $0 \le \gamma < 1$. 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 γ 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,

$$\mathrm{P}(D, p) = 1 - \frac{1}{2^{p-1}} \qquad p = 1, 2, \ldots \qquad (13)$$

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* (*PMMBI*) [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 (γ) between the images is given by,

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

$$0 \le \gamma \le 1, p = 1, 2, \ldots \qquad (14)$$

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 γ. 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 γ.
- With increasing values of *p*, *DC* reaches unity quicker for smaller values of γ 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 γ. For example, a confidence level of 95% exists of detecting dissimilarity when mappings 7 pixels between pairs of images with γ = 0.25. To attain the same confidence for pairs of images with higher similarity, say γ = 0.90, requires 59 mappings. To reach higher confidence levels requires additional mapping for any given value of γ.

The expected value of *p* is the mean number of mappings required to detect dissimilarity for a given value of γ, and is given by [17],

$$E[p(\gamma)] = \frac{4}{1-\gamma^2} - 1 \qquad 0 \le \gamma < 1 \qquad (15)$$

Table 2 tabulates the expected number of mappings required to detect dissimilarity for several values of γ. It can be seen that the number of mappings are small for small values of γ, but grow very quickly as γ 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 γ 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 γ = 0.5, or a 75% confidence that the similarity between the two images is γ = 0.75. In fact, there is a confidence level attached to every γ value as can be seen from **Fig. 1**, which plots the detection confidence (Pr) as function of γ for a value of *p* = 11. The most probable value for this pair of images is that it has a similarity value of γ = 0.817, which corresponds to the expected value of *p* = 11 mappings (obtained from (15)).

IV. DISSIMILAR 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:

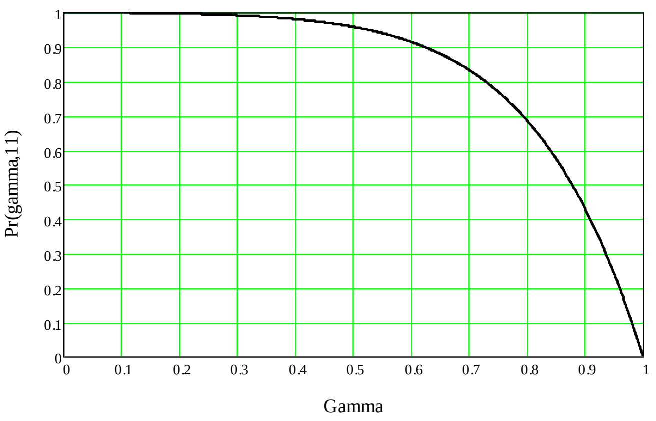

Fig. 1. A plot of the detection confidence (Pr) vs. γ for a value of p = 11.

a) **Detecting image dissimilarity**: The objective is to determine if images are dissimilar.
b) **Detecting image similarity**: The objective is to determine if images are exactly similar.
c) **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.

In this paper we show that *DDM* can be used for objective ‘a’ and ‘b’. Using *DDM* to address objective ‘c’ was briefly discussed in [16]. However, we do show in our later discussion how the amount of similarity between two images can be estimated with confidence if dissimilarity is not detected. An approach to binary similarity estimation between binary images, i.e. binary image matching, was presented in [33].

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)

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

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

γ	*p*	γ	*p*
0.0000	3.00	0.8000	10.1
0.1000	3.04	0.8500	13.4
0.2000	3.17	0.9000	20
0.3000	3.40	0.9500	40
0.4000	3.76	0.9900	200
0.5000	4.33	0.9980	1000
0.6000	5.25	0.9990	2000
0.7000	6.84	0.9999	20000

to which the query image is to be compared to. 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 *DDM*. Hence,

$$N = N_d + N_s \tag{16}$$

Define ζ such that,

$$\zeta = \frac{N_s}{N} \qquad \text{and} \qquad 1-\zeta = \frac{N_d}{N} \tag{17}$$

i.e., ζ 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 *DDM* (e.g. *exact-similar images*). ζ is dependent on the maximum allowable mappings (L_{map}) for each pair of images. For example, using L_{map} = 1000 mappings results in *DC* = 0.993 for images with a similarity of up to γ = 0.99 (*near-duplicate images*). However, at this number of mappings, near-similar images (γ = 0.999) have a low confidence level of *DC* = 0.394. Increasing L_{map} to 2000 mappings results in *DC* = 0.99996 for images with a similarity of up to γ = 0.99, and *DC* = 0.632 for near-similar images. Only at L_{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_{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_{map}. Let γ_L denote the maximum expected similarity value at which images will be detected based on L_{map}. This implies that image pairs with $\gamma > \gamma_L$ will most likely not be detected as dissimilar. Hence, a value of L_{map} = 1000 implies that γ_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_{map} = 1000 is what is usually employed in our work.

Let $t_{p,d}$ denote the amount of time taken for *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_{map} value, $t_{p,d}$ is then,

$$t_{p,d} = t^*_p = L_{map} \cdot t_p \tag{18}$$

t^*_p is the amount of time taken to map points between two images until L_{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 $t^*_p \cong 3.433$ ms for L_{map} = 1000. Then if T_p denotes the total detection time for *DDM* to compare the query image to all images in the database set, then T_p can be calculated by,

$$T_P = N_d \mathrm{E}[t_{p,d}] + N_s t_p^* \tag{19}$$

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

$$\mathrm{E}[t_{p,d}] = \mathrm{E}[\gamma_\mu] \cdot t_p \tag{20}$$

Hence,

$$\mathrm{E}[t_{p,d}] = MRN(\gamma_\mu, DC) \cdot t_p \tag{21}$$

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 (γ_μ), and 2) the desired detection confidence (*DC*). Hence, (19) becomes,

$$\begin{aligned} T_P &= N_d \cdot MRN(\gamma_\mu, DC) \cdot t_p + N_s \cdot L_{map} \cdot t_p \\ &= (N_d \cdot MRN(\gamma_\mu, DC) + N_s L_{map}) t_p \end{aligned} \tag{22}$$

If T_x denotes the detection time that *method-x* takes to match the query image to all images of the database to determine dissimilarity, then,

$$T_x = N\bar{t}_x \tag{23}$$

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

$$\bar{t}_x = n t_x \tag{24}$$

t_x is the matching time per pixel for *method-x*. Hence,

$$T_x = N n t_x \tag{25}$$

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

$$\eta_{\mathrm{D,x}} = \frac{T_x}{T_p} \tag{26}$$

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

$$\eta_{D,x} = \frac{N \cdot n \cdot t_x}{(N_d \cdot MRN(\gamma_\mu, DC) + N_s \cdot L_{map}) t_p} \tag{27}$$

Dividing by N and using (17) produces,

$$\eta_{D,x}(n, \gamma_\mu, DC, \zeta, L_{map}, r_x) = \frac{n \cdot r_x}{(1-\zeta) \cdot MRN(\gamma_\mu, DC) + \zeta \cdot L_{map}} \tag{28}$$

$r_x = t_x / t_p$, is a constant dependent on *method-x* and *DDM*. Normalizing with respect to r_x produces,

$$\eta_D(n,\gamma_\mu,DC,\zeta,L_{map}) = \eta_{D,x}/r_x$$
$$= \frac{n}{(1-\zeta)\cdot MRN(\gamma_\mu,DC)+\zeta\cdot L_{map}} \quad (29)$$

This equation shows η_D to be a function of several variables. The sensitivity of η_D to these variables is discussed next.

IV.I.I Sensitivity Analysis of η_D

Eq. (29) shows η_D to be a function of the parameters: *n*, ζ, L_{map} and *MRN*, where *MRN* is a function of both γ_μ and *DC*. It can be seen that,

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

1) Sensitivity Analysis of η_D to n

Figure 2 shows curves of η_D versus *n* for several values of ζ (for the sample case of γ_μ = 0.2 and L_{map} = 1000). Due to the dominance of the effect of *n* on η_D, the curves are linear in *n*; as *n* increases so does η_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*,

$$MRN(\gamma_\mu,DC) \le L_{map} \quad (30)$$

Hence *MRN*(γ_μ, *DC*) is also bounded and finite, resulting in the denominator of (29) also being finite, and hence,

$$\lim_{n\to\infty}\eta_D = \infty \quad (31)$$

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

2) Sensitivity Analysis of η_D to ζ

A decrease in ζ (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 ζ result in larger values of η_D and higher relative performance of *DDM* with respect to other methods. In the limit as ζ becomes very small and $\zeta \to 0$, (29) reveals that η_D essentially becomes a function of *n*,

$$\lim_{\zeta\to 0}\eta_D = \frac{n}{MRN(\gamma_\mu,DC)} \quad (32)$$

n dominates over the effect of *MRN* which is dependent on the two parameters, γ_μ and *DC*. Hence, η_D increases as image size increases. **Figures 3** shows curves of η_D versus *n* for different values of γ_μ when ζ = 0 (for the case L_{map} = 1000). η_D increases with *n*, but is inversely proportional to γ_μ, where an increase in γ_μ shifts the curve downwards. The sensitivity of η_D to a change in the three parameters γ_μ, *DC* and L_{map} is very minimal, where an increase in any of these parameter 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.I.II** Critical Image Size Determination*

For *DDM* to outperform *method-x*, then,

$$\eta_{D,x}(n,\gamma_\mu,DC,\zeta,L_{map}) > 1 \quad (33)$$

must hold true. Then from (28),

$$\frac{n\cdot r_x}{(1-\zeta)\cdot MRN(\gamma_\mu,DC)+\zeta\cdot L_{map}} > 1 \quad (34)$$

must also hold true. Solving for *n*,

$$n > \frac{1}{r_x}\left((1-\zeta)\cdot MRN(\gamma_\mu,DC)+\zeta\cdot L_{map}\right) \quad (35)$$

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 $\eta_{D,x}$ = 1, then,

$$n_c = \frac{1}{r_x}\left((1-\zeta)\cdot MRN(\gamma_\mu,DC)+\zeta\cdot L_{map}\right) \quad (36)$$

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 γ_μ for different values of ζ (with different values of L_{map} and *DC*). A value of $r_x = 3.434\times10^{-3}$, 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 γ_μ increases. Also as ζ increases, the curves shift upwards and hence n_c increases. Furthermore, as *DC* increases so does n_c. Note that for highly similar images ($\gamma_\mu > 0.9$) n_c increases rapidly.

IV.II 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.

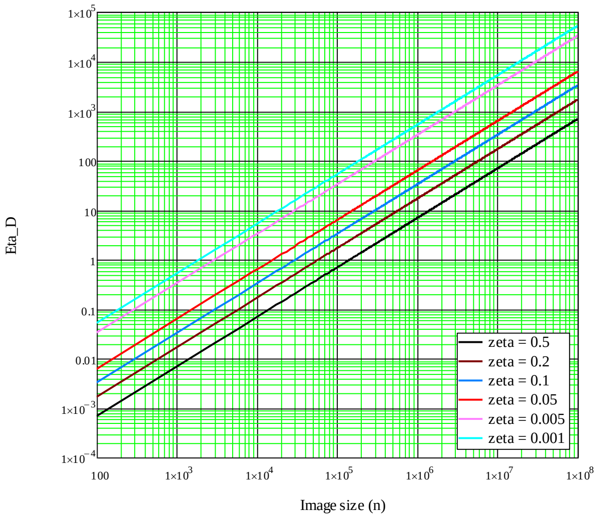

Fig. 2. Curves of η_D versus n for different values of ζ ($\gamma_\mu = 0.2$, $L_{map} = 1000$).

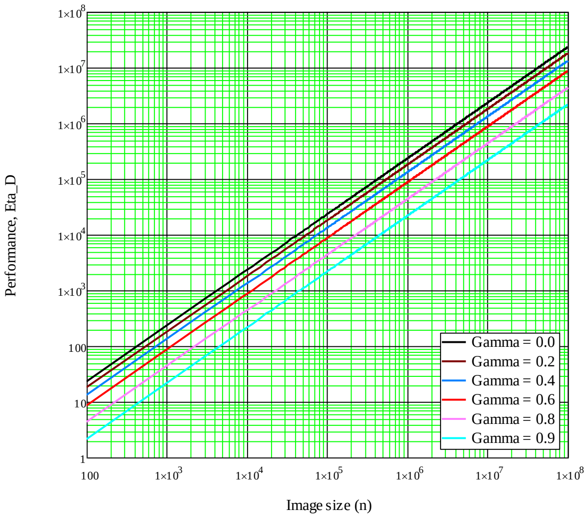

Fig. 3. Curves of η_D versus n for different values of γ_μ ($\zeta = 0$, $L_{map} = 1000$).

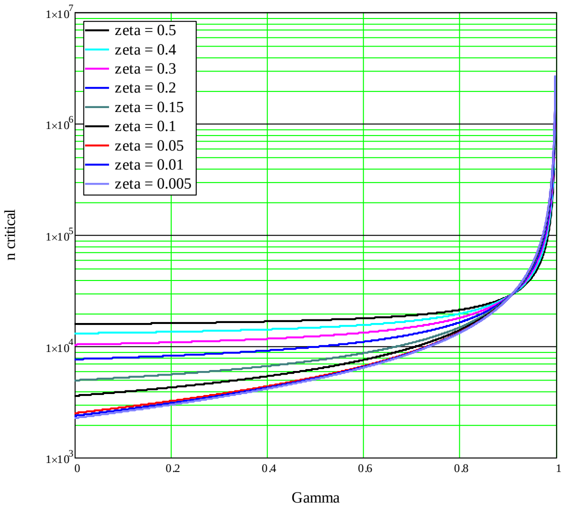

Fig. 4. Curves of critical image size (n_c) vs. γ_μ for various values of ζ ($L_{map} = 1000$, $DC = 0.99$).

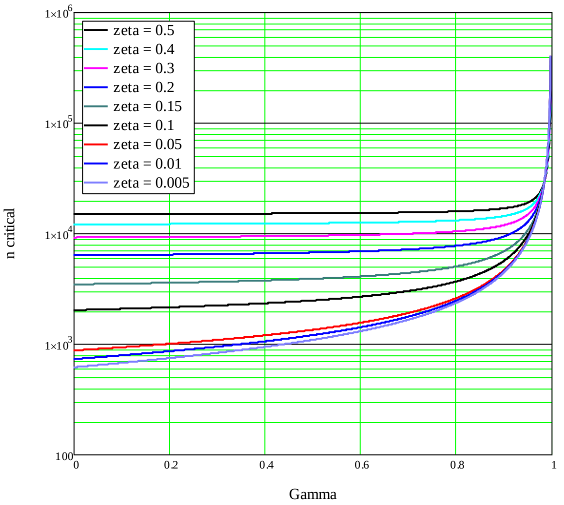

Fig. 5. Curves of n_c vs. γ_μ for various values of ζ ($L_{map} = 1000$, $DC = 0.5$).

IV.II.I 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: the time taken by *DDM* to map L_{map} points and the time taken by *method-x*,

$$t_{p,x} = t^*_p + nt_x \tag{37}$$

Using (18) this becomes,

$$t_{p,x} = L_{map} \cdot t_p + nt_x \tag{38}$$

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

$$T_{px} = N_d \bar{t}_{p,d} + N_s t_{p,x} \tag{39}$$

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,

$$\eta_{S,x} = \frac{T_x}{T_{px}} \tag{40}$$

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

$$\eta_{S,x} = \frac{N \cdot n \cdot t_x}{N_d \cdot MRN(\gamma_\mu) \cdot t_p + N_s(L_{map} \cdot t_p + n \cdot t_x)} \tag{41}$$

Dividing by N, and using (17),

$$\eta_{S,x} = \frac{n \cdot t_x}{(1-\zeta) \cdot MRN(\gamma_\mu) \cdot t_p + \zeta(L_{map} \cdot t_p + n \cdot t_x)} \tag{42}$$

Rearranging the denominator,

$$\eta_{S,x} = \frac{n \cdot t_x}{MRN(\gamma_\mu) \cdot t_p + \zeta(L_{map} \cdot t_p + nt_x - MRN(\gamma_\mu) \cdot t_p)} \tag{43}$$

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

$$\eta_{S,x}(n, \gamma_\mu, DC, \zeta, L_{map}) = \frac{n \cdot r_x}{MRN(\gamma_\mu, DC) + \zeta(L_{map} - MRN(\gamma_\mu, DC) + n \cdot r_x)} \tag{44}$$

Figures 6 and **7** show plots of $\eta_{S,x}$ curves versus n for various values of ζ and different values of L_{map}, γ_μ and DC (for $r_x = 3.434 \times 10^{-3}$). **Figures 8** and **9** show typical plots of $\eta_{S,x}$ curves versus n for different values of γ_μ (with different values of L_{map}, ζ and DC values).

IV.II.II 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 ζ or γ_μ increases, $\eta_{S,x}$ decreases.

n and ζ 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,

$$L_{map} - MRN(\gamma_\mu, DC) << n \cdot r_x \tag{45}$$

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

$$\lim_{n\to\infty} \eta_{S,x} = \frac{n \cdot r_x}{\zeta(n \cdot r_x)} = \frac{1}{\zeta} \tag{46}$$

Hence, as image size increases, $\eta_{S,x}$ approaches a performance limit equal to the reciprocal of the percentage of very similar images to the query image in the set. Since (46) is not a function of r_x then this is true for the performance of *DDM-x* with respect to any *ISBMM*. With $\zeta \leq 1$, then *DDM-x* will outperform any image sized based matching method or in the worst case will have the same performance (when all images of the database are highly similar or near duplicate to the query image).

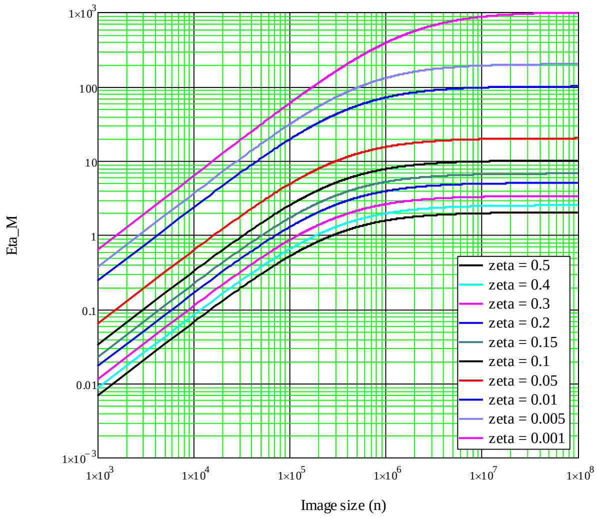

Fig. 6. Curves of η_S vs. image size for various values of ζ (L_{map} = 1000, γ_μ = 0.2, DC = 0.9).

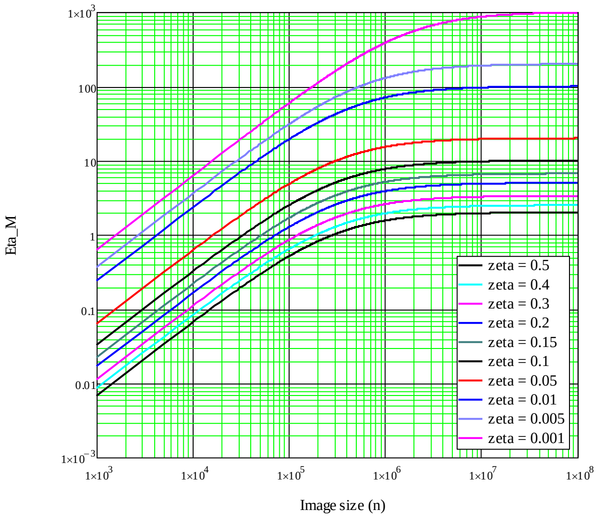

Fig. 7. Curves of η_S vs. image size for various values of ζ (L_{map} = 1000, γ_μ = 0.2, DC = 0.9).

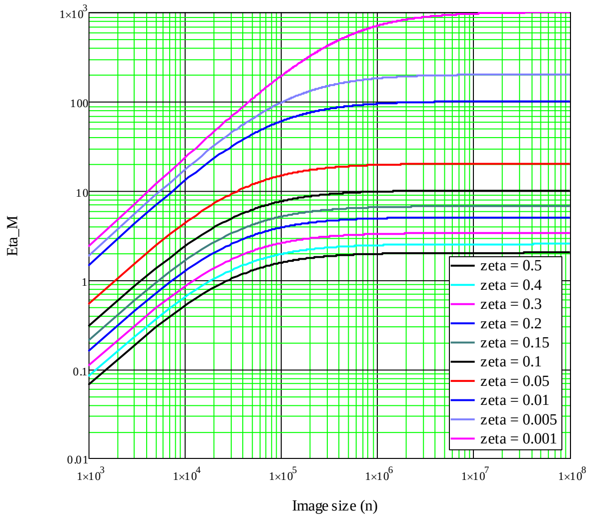

Fig. 8. Curves of η_M vs. image size for various values of ζ (L_{map} = 100, γ_μ = 0.2, DC = 0.5).

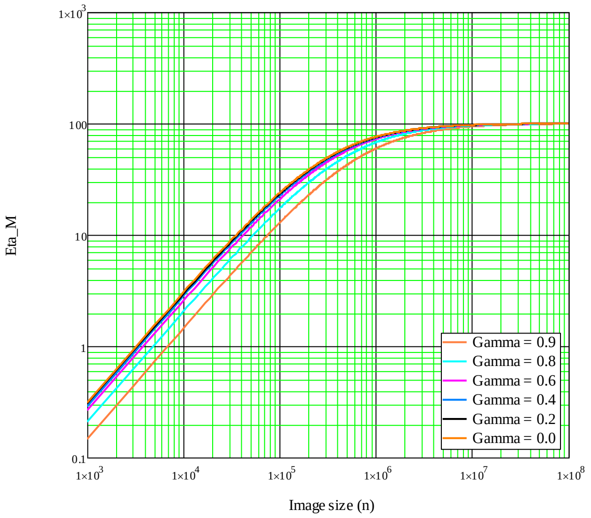

Fig. 9. Curves of η_M vs. image size for various values of γ_μ (ζ = 0.01, L_{map} = 1000, DC = 0.5).

2. Sensitivity Analysis of $\eta_{S,x}$ to ζ

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

$$\lim_{\zeta\to 0}\eta_{S,x} = \frac{n\cdot r_x}{MRN(\gamma_\mu, DC)} \tag{47}$$

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

$$\lim_{\zeta\to 0}\eta_{S,x} = \lim_{\zeta\to 0}\eta_{D,x} \tag{48}$$

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,

$$\lim_{\substack{n\to\infty \\ \zeta\to 0}}\eta_{S,x} = \infty \tag{49}$$

This result agrees with (31). Hence,

$$\lim_{\substack{n\to\infty \\ \zeta\to 0}}\eta_{S,x} = \lim_{n\to\infty}\eta_{D,x} \tag{50}$$

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 γ statistics for the image sets are shown in Table 3. A value of L_{map} = 1000 was employed for *DDM*.

V.I Detecting Dissimilar Images with ζ = 0

The first series of tests are for image sets with ζ = 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 ($\eta_{D,corr}$)**
  $\eta_{D,corr}$ had a performance range from 152 to 141301, as image size increased from 16 kp to 10 Mp.
- ***DDM* performance with respect to *SADM* ($\eta_{D,SADM}$)**
  $\eta_{D,SADM}$ had a performance range from 450 to 227550, as image size increased from 16 kp to 10 Mp.
- ***DDM* performance with respect to *MI* ($\eta_{D,MI}$)**
  $\eta_{D,MI}$ had a performance range from 3409 to 373743, as image size increased from 16 kp to 10 Mp.

The values obtained for $\eta_{D,x}$ are impressive; *DDM* is magnitudes faster than other methods. Values of $\eta_{D,x}$ continue to grow with increased image size. Most impressive are the values obtained for the largest image size tested (10Mp); $\eta_{D,corr} = 141301$, $\eta_{D,SADM} = 227550$ and $\eta_{D,MI} = 373743$. This implies that *DDM* is 141301 times faster than using correlation! At this rate, we expect *DDM* to be a million times faster than *corr* for 100 Mp images, and well over 15 million times faster than *corr* for 1 Gp images.

V.II Detecting Dissimilar Images with $\zeta \neq 0$

The second series of tests are for image sets with $\zeta = 0.4$; there are *similar* images in these sets (that *DDM* won't be able to detect as being dissimilar). The performance curves for $\eta_{D,x}$ for all methods are shown in **Fig. 11** and tabulated in **Table 5**. We see as a result of the increase in ζ that the $\eta_{D,x}$ curves have shifted down in comparison with **Fig.** 9, in agreement with our earlier analytical discussion. Once again, $\eta_{D,x} > 1$ for all image sizes tested for all methods, except for 16 kp images using 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* ($\eta_{D,SADM}$)**
  $\eta_{D,corr}$ had 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 $\eta_{D,x}$ 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 $\eta_{D,x}$ is greater than unity for all cases tested and increases as image size increases. The fact that $\eta_{D,x}$ produced large values for big images ($\eta_{D,corr} = 1097$, $\eta_{D,SADM} = 1766$ 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 $T_{p,c}$, T_c and $\eta_{S,corr}$ for the image sets, and a plot of T_c and T_{pc} against image size is shown in **Fig. 12**. We observe,

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

Fig. 13 shows a plot of $\eta_{S,corr}$ for the results obtained experimentally. A plot of the theoretical equation developed for $\eta_{S,corr}$ (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, $\eta_{S,corr}$ 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 3: γ Statistics for the image sets

Image Set	Image size	γ		
		Range	μ	σ
16 kp	128 x 128	0.000 – 0.364	0.140	0.128
64 kp	256 x 256	0.093 – 0.284	0.196	0.068
256 kp	521 x 512	0.024 – 0.206	0.102	0.056
300 kp	640 x 480	0.002 – 0.256	0.145	0.090
3 Mp	2048 x 1536	0.029 – 0.129	0.069	0.039
8 Mp	3264 x 2448	0.122 – 0.479	0.297	0.130
10 Mp	3648 x 2736	0.026 – 0.363	0.170	0.135

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

Image Set	Image size	***Method***		
		SADM	***corr***	***MI***
16 kp	128 x 128	450	152	3409
64 kp	256 x 256	1691	662	5114
256 kp	521 x 512	6784	4032	13677
300 kp	640 x 480	7264	4368	14119
3 Mp	2048 x 1536	87847	53474	149075
8 Mp	3264 x 2448	138044	88140	223181
10 Mp	3648 x 2736	227550	141301	373743

Table 5: Values of ηD,x for the methods tested ($\zeta = 0.4$)

Image Set	Image Dimensions	**Method (*x*)**		
		SADM	***corr2***	***MI***
16 kp	128 × 128	3	1	26
64 kp	256 × 256	13	5	40
256 kp	512 × 512	51	31	104
300 kp	640 × 480	56	34	108
3 Mp	2048 × 1536	660	402	1120
8 Mp	3264 × 2448	1163	743	1880
10 Mp	3648 × 2736	1766	1097	2901

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

Image Set	Image Dimensions	T_c (ms)	T_{pc} (ms)	$\eta_{S,corr}$
16 kp	128×128	2.1	13.6	0.15
64 kp	256×256	10.1	17	0.6
256 kp	512×512	85.8	48.2	1.78
300 kp	640×480	107	56.4	1.9
3 Mp	2048×1536	1238	511.1	2.42
8 Mp	3264×2448	3393	1371	2.47
10 Mp	3648×2736	5026	2024	2.48

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, ζ. 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 (ζ = 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 (ζ = 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/ζ. 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 (ζ = 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.

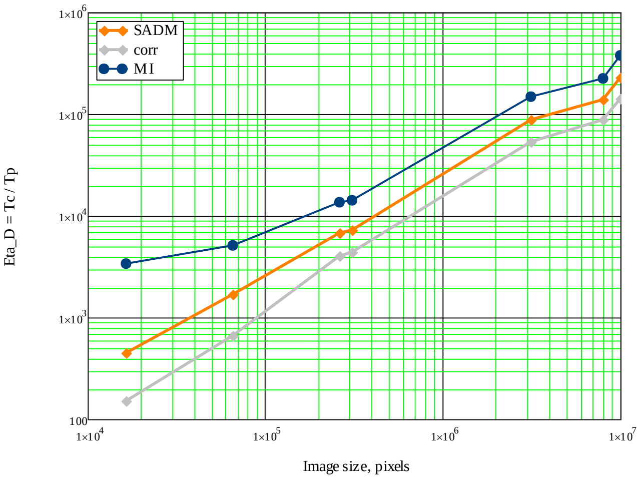

Fig. 10. Plots of $\eta_{D,x}$ against image size for the various methods tested (ζ = 0.0).

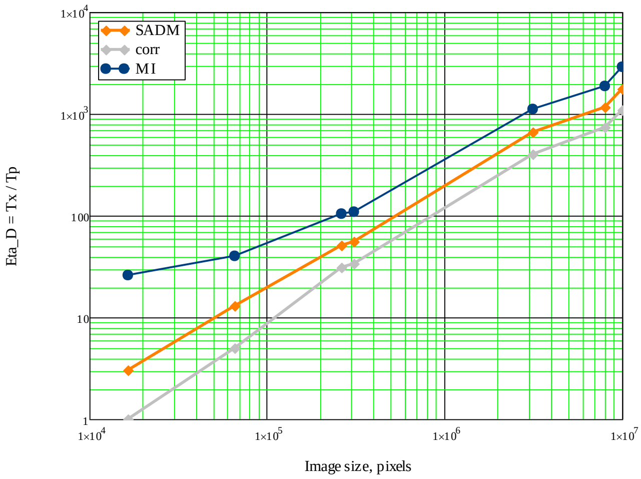

Fig. 11. Plots of $\eta_{D,x}$ against image size for the various methods tested (ζ = 0.4).

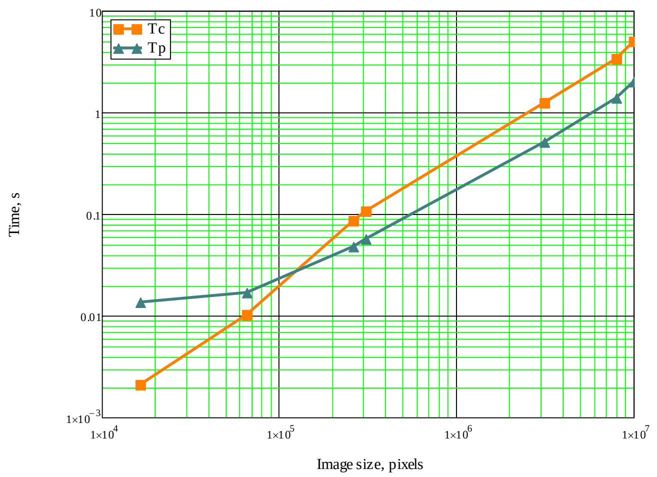

Fig. 12. Plots of T_c and T_{pc} against image size.

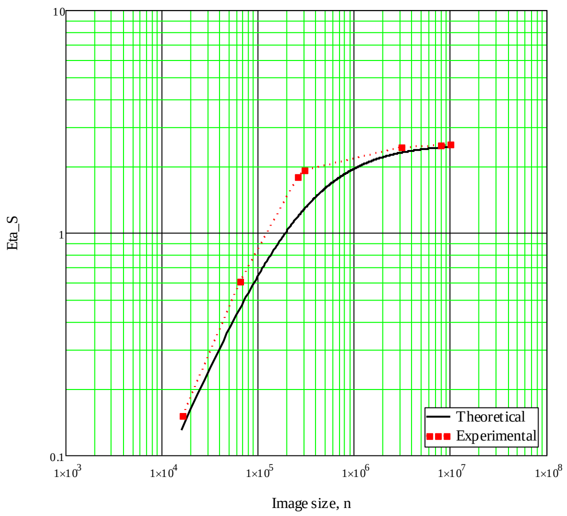

Fig. 13. Plots of $\eta_{S,corr}$ against image size: theoretical and experimental results.

REFERENCES

[1] Zitova, B. and Flusser, J., 2003, "Image registration methods: a survey." Image and vision computing 21, (11), pp. 977-1000.

[2] Le Moigne, J., Netanyahu, N., & Eastman, R. (Eds.), 2011. Image Registration for Remote Sensing. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511777684

[3] Briechle, K., and Hanebeck, U., 2001, "Template matching using fast normalized cross correlation." Optical Pattern Recognition XII. Vol. 4387. International Society for Optics and Photonics.

[4] Xia, H., et al., 2018, "Fast template matching based on deformable best-buddies similarity measure." Multimedia Tools and Applications pp. 1-21.

[5] Yilmaz, A., Javed, O. and Shah, M., 2006, "Object tracking: A survey." ACM computing surveys (CSUR) 38.4: 13.

[6] Gong, S., et al., 2019, "Visual object tracking." Advanced Image and Video Processing Using MATLAB. Springer, Cham, pp. 391-428.

[7] Tekalp, A., 2015, Digital video processing. Prentice Hall Press.

[8] Kastrinaki, V., Zervakis, M., and Kalaitzakis, K., 2003, "A survey of video processing techniques for traffic applications." Image and vision computing 21.4, pp. 359-381.

[9] Lowe, D., 1999, "Object recognition from local scale-invariant features", the proceedings of the *seventh IEEE international conference on Computer vision*, Vol. 2.

[10] Anuta, P., 1970, "Spatial Registration of Multispectral and Multitemporal Digital Imagery Using Fast Fourier Transform Techniques", *IEEE Trans. on Geoscience Electronics*, GE-8, N 4, pp. 353-368.

[11] Barnea, D. and Silverman, H., 1972, "A Class of Algorithms for Fast Digital Image Registration", *IEEE Transactions on Computers*, 21(2), pp. 179-186.

[12] Yoo, J. and Han, T., 2009. "Fast normalized cross-correlation". Circuits, systems and signal processing, 28(6), pp. 8-19.

[13] Mukherji, S., 2005, Fast Algorithms for Binary Cross-correlation, *proceedings of Geoscience and Remote Sensing Symposium*, V.1, pp. 4-11.

[14] Lewis, J., Fast Template Matching, *Vision Interface*, 120-123, (1995).

[15] Kovalev, V.A., Kalinovsky, A.A. and Liauchuk, V.A. Deep Learning in Big Image Data: Histology Image Classification for Breast Cancer Diagnosis, *Int. Conference on Big Data and Advanced Analytics*, Minsk, Belarus, 15-17 June, 2016.

[16] Mustafa A. A., 2015, "Probabilistic model for quick detection of dissimilar binary images", *J. Electron. Imaging*. 24(5), 053024. http://dx.doi.org/10.1117/1.JEI.24.5.053024.

[17] Mustafa, A. A., 2017, "A Probabilistic Model for Random Binary Image Mapping", WSEAS Transactions on Systems and Control, Volume 12, Art. #34, pp. 317-331.

[18] Montgomery, D., and Runger G., 2014, *Applied Statistics & Probability for Engineers*, 6th Edition, John Wiley.

[19] Vanne, J., Aho, E., Hämäläinen, T., and Kuusilinna, K., 2006, "A High-Performance Sum of Absolute

Difference Implementation for Motion Estimation:, *IEEE Transactions on Circuits and Systems for Video Technology*, 16(7), pp. 876-883.

[20] Hamming, R. W., 1950, “Error detecting and error correcting codes”, *Bell System Technical Journal*, 29(2), pp. 147–160.

[21] Sokal, R., and Michener, C., 1958, “A statistical method for evaluating systematic relationships”, *Bulletin of the Society of University of Kansas*, 38, pp. 1409-1438.

[22] Maes, F., et al., 1997, “Multimodality Image Registration by Maximization of Mutual Information,” *IEEE Transactions on Medical Imaging*, 16(2), pp. 187-198.

[23] Tomaževič, D., Likar, B. and Pernuš, F., 2012, “Multi-Feature Mutual Information Image Registration”, *Image Anal Stereol*, 31, pp. 43-53.

[24] Josien, P., et al., 2003, “Mutual-Information-Based Registration of Medical Images: A Survey”, *IEEE Transactions on Medical Imaging*, 22(8).

[25] Cover, T. M., and Joy, A., 2012, Thomas, *Elements of information theory*, John Wiley & Sons.

[26] Castro-Pareja, C. and Shekhar, R., 2005, "Hardware acceleration of mutual information-based 3D image registration." Journal of Imaging Science and Technology 49, no. 2, pp. 105-113.

[27] Mustafa, A. and Ganter, M., 1995, “An Efficient Image Registration Method by Minimizing Intensity Combinations”, *Research in Computer and Robot Vision*, Archibald, C. and Kwok, P. (Eds.), World Scientific Press, Singapore, pp. 247-268.

[28] Baudrier, E., Nicolier, F., Millon, G. and Ruan, S., 2008, “Binary-image comparison method with local-dissimilarity quantification”, *Pattern Recognition*, 41, pp. 1461-1478.

[29] Tang, F. and Tao, H., 2007, “Fast multi-scale template matching using binary features”, *8th IEEE Workshop on Applications of Computer Vision*, pp. 36-39.

[30] Vidal, J. and Crespo, J., 2008, “Sets Matching in Binary Images Using Mathematical Morphology”, *International Conference of the Chilean Computer Science Society*, pp. 110-115.

[31] Teshome, M., Zerubabe, L. and Yoon, K., 2009, “A Simple Binary Image Similarity Matching Method Based on Exact Pixel Matching”, *International Conference on Computer Engineering and Applications*, pp. 12-15.

[32] Mustafa, A. A., 2018, “A Probabilistic Binary Similarity Distance for Quick Image Matching”. IET Journal on Image Processing, 12 (10), pp. 1844-1856.

[33] Mustafa, A. A., 2018, “Quick Similarity Measurement of Binary Images via Probabilistic Pixel Mapping”. World Academy of Science, Engineering and Technology, International Science Index 137, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 12 (5), pp. 297 - 301.

[34] Mustafa, A. A., 2016, “Two Algorithms for Detecting Dissimilar and Similar Binary Images”, Kuwait University, Department of Mechanical Engineering, Technical Report: KUME-AAM-2016-3.